

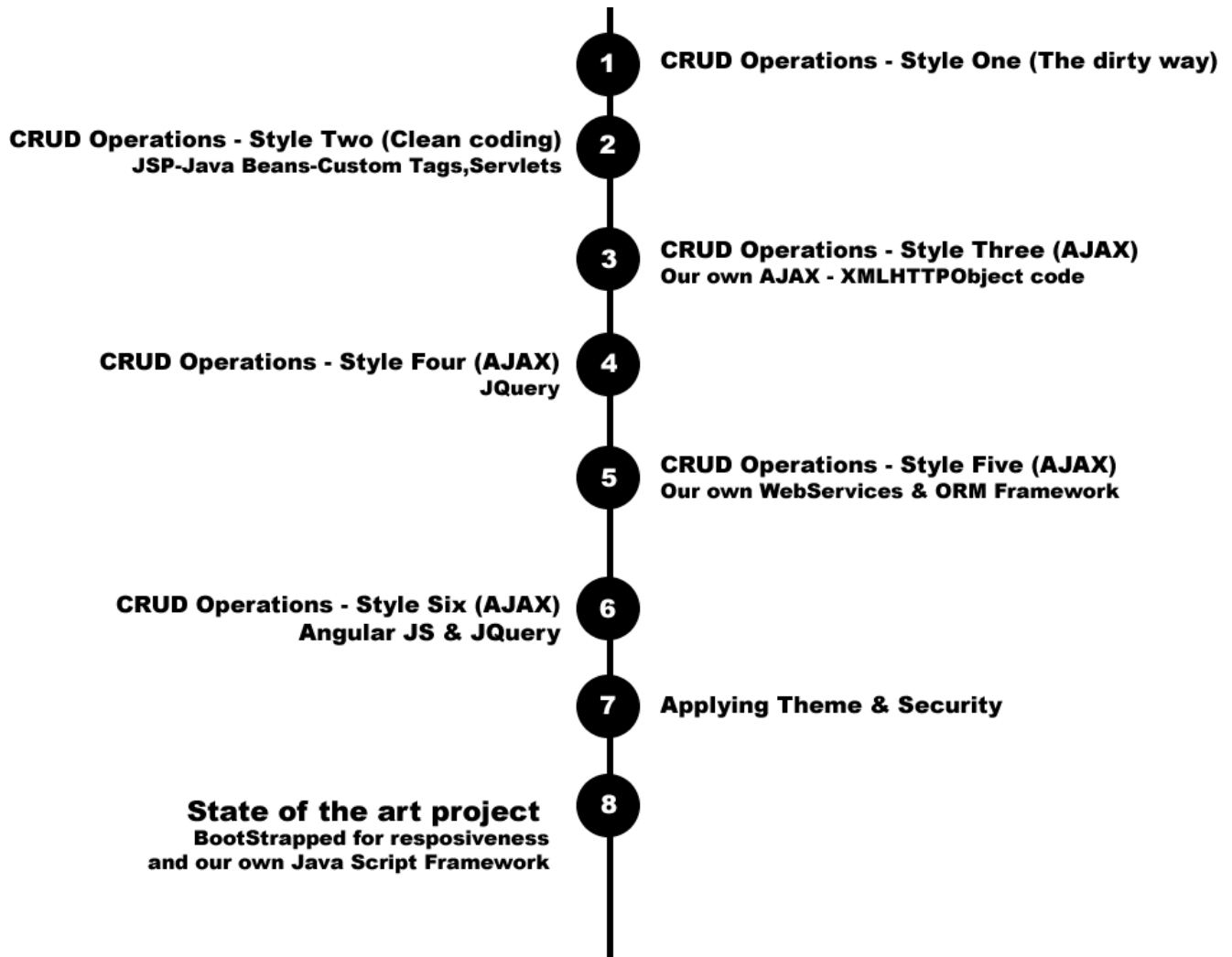
J2EE Web Application Development

This is the kind of work that most of you will get,
when it comes to Indian Service Industry

I will try to take you beyond that.

The learning curve is going to be very steep.

**To create magic,
me
as
well as you
need to work hard**



Download Apache Tomcat 8 from Apache's site (Download the zip version of binary file)

Unzip it and copy the extracted folder to c:\ and rename it to tomcat8

Note : The tomcat8 folder contains the bin, lib, webapps and some more folders.
The jar files to be included into classpath are in the lib folder and the context name (site folder) should be kept in the webapps folder.

Now edit to c:\tomcat8\bin\startup.bat file and insert the following 2 lines at the top

```
set JAVA_HOME=c:\jdk1.8  
set CATALINA_HOME=c:\tomcat8
```

Note : those who are working on LINUX don't need to do anything as we have already configured tomcat8 for linux. You just have to replace ; with : and \ with / when it comes to specifying classpath.

Now create a shortcut to c:\tomcat8\bin\startup.bat named as tomcat8. (For windows users)

Double click the shortcut to start the server

Start the browser and type the url <http://localhost:8080> and you should see the Apache Tomcat default home page.

We will be developing the CRUD Modules using various techniques.

I have created a mysql database named as mydb2017 and I will be connecting using the username as root and password as kelkar. I have already taught you to create database, create user and how to grant rights of the database to the use. You can create a database/user account of your choice.

Login and create table using the following SQL Statement

```
create table author  
(  
code int primary key auto_increment,  
name char(35) not null unique  
)Engine=InnoDB;  
create table book  
(  
code int primary key auto_increment,  
title char(35) not null unique,  
author_code int not null,  
category char(35) not null,  
price int not null,  
foreign key (author_code) references author(code)  
)Engine=InnoDB;
```

CRUD One

Dirty Programming, but it is very essential to understand the techniques involved.

Issues : Refresh and Back button Problem, at some places I have purposely made the mistake of writing incorrect style.

These will be solved in CRUD Two

Download & See it in Action

<http://tm-certificates.com/CRUDOne.wmv>

Create the following structure in webapps folder

crudone – WEB-INF - classes

crudone - WEB-INF - lib

crudone - images

The lib folder should contain mysql.jar file

The classes folder will contain the necessary packages with compiled classes

The image folder should contain the following image files



delete_icon.png (16x16)



edit_icon.png (16x16)



logo.png (whatever is your requirement)

You can download zip file with these images - CRUDOneResources.zip

The Data Layer (To compile, use javac *.java)

```
package com.thinking.machines.library.dl;
public class DAOException extends Exception
{
public DAOException(String message)
{
super(message);
}
}
```

```
package com.thinking.machines.library.dl;
import java.sql.*;
public class DAOConnection
{
private DAOConnection()
{
}
public static Connection getConnection() throws DAOException
{
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection connection;
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb2017","root","kelkar");
return connection;
}catch(ClassNotFoundException classNotFoundException)
```

```
{
throw new DAOException("Driver class : com.mysql.jdbc.Driver missing in classpath");
}
catch(SQLException sqlException)
{
throw new DAOException("Unable to connect : "+sqlException.getMessage());
}
}
}
}
}
}

package com.thinking.machines.library.dl;
public interface AuthorInterface extends java.io.Serializable,Comparable<AuthorInterface>
{
public void setCode(int code);
public int getCode();
public void setName(String name);
public String getName();
}

package com.thinking.machines.library.dl;
public class Author implements AuthorInterface
{
private int code;
public String name;
public Author()
{
this.code=0;
this.name="";
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
return this.code;
}
public void setName(String name)
{
this.name=name;
}
public String getName()
{
return this.name;
}
public boolean equals(Object object)
{
if(!(object instanceof Author)) return false;
```

```
Author otherAuthor=(Author)object;
return this.code==otherAuthor.code;
}
public int compareTo(AuthorInterface otherAuthor)
{
return this.code-otherAuthor.getCode();
}
public int hashCode()
{
return this.code;
}
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
public interface AuthorDAOInterface
{
public void add(AuthorInterface author) throws DAOException;
public void update(AuthorInterface author) throws DAOException;
public void remove(int code) throws DAOException;
public AuthorInterface getByCode(int code) throws DAOException;
public AuthorInterface getByName(String name) throws DAOException;
public LinkedList<AuthorInterface> getAll() throws DAOException;
public long getCount() throws DAOException;
}
}
```

```
package com.thinking.machines.library.dl;
import java.util.*;
import java.sql.*;
public class AuthorDAO implements AuthorDAOInterface
{
public void add(AuthorInterface author) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where name=?");
preparedStatement.setString(1,author.getName());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Author : "+author.getName()+" exists.");
}
}
}
```

```

preparedStatement=connection.prepareStatement("insert into author (name)
values(?)",Statement.RETURN_GENERATED_KEYS);
preparedStatement.setString(1,author.getName());
preparedStatement.executeUpdate();
resultSet=preparedStatement.getGeneratedKeys();
resultSet.next();
author.setCode(resultSet.getInt(1));
resultSet.close();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to add : "+sqlException.getMessage());
}
}
public void update(AuthorInterface author) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where name=? and
code <=?");
preparedStatement.setString(1,author.getName());
preparedStatement.setInt(2,author.getCode());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Author : "+author.getName()+" exists.");
}
preparedStatement=connection.prepareStatement("update author set name=? where code=?");
preparedStatement.setString(1,author.getName());
preparedStatement.setInt(2,author.getCode());
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
}catch(SQLException sqlException)
{
throw new DAOException("Unable to update : "+sqlException.getMessage());
}
}

public void remove(int code) throws DAOException

```



```
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists==false)
{
connection.close();
throw new DAOException("Author code : "+code+"does not exist.");
}
preparedStatement=connection.prepareStatement("select 1 as result from book where author_code=?");
preparedStatement.setInt(1,code);
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Cannot delete author as book exists against it.");
}
preparedStatement=connection.prepareStatement("delete from author where code=?");
preparedStatement.setInt(1,code);
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
} catch(SQLException sqlException)
{
throw new DAOException("Unable to delete : "+sqlException.getMessage());
}
}
}
public AuthorInterface getByCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from author where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
```

```
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Author code : "+code+" does not exist.");
}
AuthorInterface author=new Author();
author.setCode(resultSet.getInt("code"));
author.setName(resultSet.getString("name").trim());
resultSet.close();
preparedStatement.close();
connection.close();
return author;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}

public AuthorInterface getByName(String name) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from author where name=?");
preparedStatement.setString(1,name);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Author : "+name+" does not exist.");
}
AuthorInterface author=new Author();
author.setCode(resultSet.getInt("code"));
author.setName(resultSet.getString("name").trim());
resultSet.close();
preparedStatement.close();
connection.close();
return author;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}
```

```
public LinkedList<AuthorInterface> getAll() throws DAOException
{
    LinkedList<AuthorInterface> authors;
    try
    {
        Connection connection=DAOConnection.getConnection();
        Statement statement=connection.createStatement();
        ResultSet resultSet=statement.executeQuery("select * from author order by name ");
        if(resultSet.next()==false)
        {
            resultSet.close();
            statement.close();
            connection.close();
            throw new DAOException("No authors");
        }
        authors=new LinkedList<AuthorInterface>();
        AuthorInterface author;
        do
        {
            author=new Author();
            author.setCode(resultSet.getInt("code"));
            author.setName(resultSet.getString("name").trim());
            authors.add(author);
        }while(resultSet.next());
        resultSet.close();
        statement.close();
        connection.close();
        return authors;
    }catch(SQLException sqlException)
    {
        throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
    }
}

public long getCount() throws DAOException
{
    try
    {
        Connection connection=DAOConnection.getConnection();
        Statement statement=connection.createStatement();
        ResultSet resultSet=statement.executeQuery("select count(*) as cnt from author");
        resultSet.next();
        long count=resultSet.getLong("cnt");
        resultSet.close();
        statement.close();
        connection.close();
        return count;
    }catch(SQLException sqlException)
```

```
{  
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());  
}  
}  
}  
}  


---

  
package com.thinking.machines.library.dl;  
public interface BookInterface extends java.io.Serializable,Comparable<BookInterface>  
{  
public void setCode(int code);  
public int getCode();  
public void setTitle(String title);  
public String getTitle();  
public void setAuthorCode(int authorCode);  
public int getAuthorCode();  
public void setCategory(String category);  
public String getCategory();  
public void setPrice(int price);  
public int getPrice();  
}  


---

  
package com.thinking.machines.library.dl;  
public class Book implements BookInterface  
{  
private int code;  
public String title;  
private int authorCode;  
private String category;  
private int price;  
public Book()  
{  
this.code=0;  
this.title="";  
this.authorCode=0;  
this.category="";  
this.price=0;  
}  
public void setCode(int code)  
{  
this.code=code;  
}  
public int getCode()  
{  
return this.code;  
}  
public void setTitle(String title)  
{  
this.title=title;
```

```
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(int price)
{
this.price=price;
}
public int getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(!(object instanceof Book)) return false;
Book otherBook=(Book)object;
return this.code==otherBook.code;
}
public int compareTo(BookInterface otherBook)
{
return this.code-otherBook.getCode();
}
public int hashCode()
{
return this.code;
}
}

package com.thinking.machines.library.dl;
import java.util.*;
```

```
public interface BookDAOInterface
{
public void add(BookInterface book) throws DAOException;
public void update(BookInterface book) throws DAOException;
public void remove(int code) throws DAOException;
public BookInterface getByCode(int code) throws DAOException;
public BookInterface getByTitle(String title) throws DAOException;
public LinkedList<BookInterface> getAll() throws DAOException;
public LinkedList<BookInterface> getAllByAuthorCode(int authorCode) throws DAOException;
public boolean containsBookWithAuthorCode(int code) throws DAOException;
public long getCount() throws DAOException;
}
}

package com.thinking.machines.library.dl;
import java.util.*;
import java.sql.*;
public class BookDAO implements BookDAOInterface
{
public void add(BookInterface book) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where title=?");
preparedStatement.setString(1,book.getTitle());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Book with title : "+book.getTitle()+" exists.");
}
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,book.getAuthorCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Invalid author code : "+book.getAuthorCode());
}
preparedStatement=connection.prepareStatement("insert into book (title,author_code,category,price)
```

```

values(?,?,?,?)",Statement.RETURN_GENERATED_KEYS);
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getAuthorCode());
preparedStatement.setString(3,book.getCategory());
preparedStatement.setInt(4,book.getPrice());
preparedStatement.executeUpdate();
resultSet=preparedStatement.getGeneratedKeys();
resultSet.next();
book.setCode(resultSet.getInt(1));
resultSet.close();
preparedStatement.close();
connection.close();
} catch(SQLException sqlException)
{
throw new DAOException("Unable to add : "+sqlException.getMessage());
}
}
public void update(BookInterface book) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where code=?");
preparedStatement.setInt(1,book.getCode());
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Book code : "+book.getCode()+" does not exist.");
}
preparedStatement=connection.prepareStatement("select 1 as result from book where title=? and
code<>?");
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(exists)
{
connection.close();
throw new DAOException("Book title : "+book.getTitle()+" exists.");
}
}
}

```

```
preparedStatement=connection.prepareStatement("select 1 as result from author where code=?");
preparedStatement.setInt(1,book.getAuthorCode());
resultSet=preparedStatement.executeQuery();
exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Author code : "+book.getAuthorCode()+" does not exist.");
}
preparedStatement=connection.prepareStatement("update book set
title=?,author_code=?,category=?,price=? where code=?");
preparedStatement.setString(1,book.getTitle());
preparedStatement.setInt(2,book.getAuthorCode());
preparedStatement.setString(3,book.getCategory());
preparedStatement.setInt(4,book.getPrice());
preparedStatement.setInt(5,book.getCode());
preparedStatement.executeUpdate();
preparedStatement.close();
connection.close();
} catch(SQLException sqlException)
{
throw new DAOException("Unable to update : "+sqlException.getMessage());
}
}
```

```
public void remove(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select 1 as result from book where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean exists=resultSet.next();
resultSet.close();
preparedStatement.close();
if(!exists)
{
connection.close();
throw new DAOException("Book code : "+code+" does not exist.");
}
preparedStatement=connection.prepareStatement("delete from book where code=?");
preparedStatement.setInt(1,code);
preparedStatement.executeUpdate();
}
```



```
preparedStatement.close();
connection.close();
} catch(SQLException sqlException)
{
throw new DAOException("Unable to delete : "+sqlException.getMessage());
}
}
public BookInterface getByCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from book where code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet;
resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Book code : "+code+" does not exist.");
}
BookInterface book;
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
resultSet.close();
preparedStatement.close();
connection.close();
return book;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}
public BookInterface getByTitle(String title) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement;
preparedStatement=connection.prepareStatement("select * from book where title=?");
```

```
preparedStatement.setString(1,title);
ResultSet resultSet;
resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("Book title : "+title+" does not exist.");
}
BookInterface book;
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
resultSet.close();
preparedStatement.close();
connection.close();
return book;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch record : "+sqlException.getMessage());
}
}

public LinkedList<BookInterface> getAll() throws DAOException
{
LinkedList<BookInterface> books;
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select * from book order by title");
if(resultSet.next()==false)
{
resultSet.close();
statement.close();
connection.close();
throw new DAOException("No books");
}
books=new LinkedList<BookInterface>();
Book book;
do
{
book=new Book();
```

```
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
books.add(book);
}while(resultSet.next());
resultSet.close();
statement.close();
connection.close();
return books;
}catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
}
public LinkedList<BookInterface> getAllByAuthorCode(int authorCode) throws DAOException
{
LinkedList<BookInterface> books;
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement=connection.prepareStatement("Select * from book where
author_code=?");
preparedStatement.setInt(1,authorCode);
ResultSet resultSet=preparedStatement.executeQuery();
if(resultSet.next()==false)
{
resultSet.close();
preparedStatement.close();
connection.close();
throw new DAOException("No books");
}
books=new LinkedList<BookInterface>();
Book book;
do
{
book=new Book();
book.setCode(resultSet.getInt("code"));
book.setTitle(resultSet.getString("title").trim());
book.setAuthorCode(resultSet.getInt("author_code"));
book.setCategory(resultSet.getString("category").trim());
book.setPrice(resultSet.getInt("price"));
books.add(book);
}while(resultSet.next());
resultSet.close();
preparedStatement.close();
```

```
connection.close();
return books;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public boolean containsBookWithAuthorCode(int code) throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
PreparedStatement preparedStatement=connection.prepareStatement("select 1 as result from book
where author_code=?");
preparedStatement.setInt(1,code);
ResultSet resultSet=preparedStatement.executeQuery();
boolean contains=resultSet.next();
resultSet.close();
preparedStatement.close();
connection.close();
return contains;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
public long getCount() throws DAOException
{
try
{
Connection connection=DAOConnection.getConnection();
Statement statement=connection.createStatement();
ResultSet resultSet=statement.executeQuery("select count(*) as cnt from book");
resultSet.next();
long count=resultSet.getLong("cnt");
resultSet.close();
statement.close();
connection.close();
return count;
} catch(SQLException sqlException)
{
throw new DAOException("Unable to fetch records : "+sqlException.getMessage());
}
}
}
```

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
</head>
<body style="background:#F5F5F5">
<div style="background:#DBDBDB;width:100%;border:2px">
<br>
</div>
<br>
<br>
<h3>Author</h3>
<ul>
<li><a href="/crudone/AuthorAddForm.html">Add</a></li>
<li><a href="/crudone/AuthorEditForm.html">Edit</a></li>
<li><a href="/crudone/AuthorDeleteForm.html">Delete</a></li>
<li><a href="/crudone/getAuthors">List</a></li>
</ul>
<h3>Book</h3>
<ul>
<li><a href="/crudone/BookAddForm.html">Add</a></li>
<li><a href="/crudone/getBooks">Edit/Delete/List</a></li>
</ul>
<div style="background:#DBDBDB;width:100%;border:2px;padding:10px">
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>

```

The Author Add Module Starts
AuthorAddForm.html (Location – (crudone) folder)

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();

```

```

var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
</script>
</head>
<body style="background:#F5F5F5">
<div style="background:#DBDBDB;width:100%;border:2px">
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</tr>
</table>
</div>
<br>
<br>
<h2>Add Author </h2>
<form action='/crudone/addAuthor' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>

```

```

<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>

```

web.xml (Location – (WEB-INF) folder)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
```

```

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
version="3.1"
metadata-complete="true">

```

```
<description>
  CRUD Operations - Style One
</description>
<display-name>CRUD Operations - Style One</display-name>
```

```
<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>
```

```
<servlet>
<servlet-name>EditAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditAuthor</servlet-name>
<url-pattern>/editAuthor</url-pattern>
</servlet-mapping>
```

```
<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>
```

```
<servlet>
<servlet-name>GetAuthorForDeletion</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthorForDeletion</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthorForDeletion</servlet-name>
<url-pattern>/getAuthorForDeletion</url-pattern>
</servlet-mapping>
```

```
<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
```



```
<servlet-mapping>  
<servlet-name>DeleteAuthor</servlet-name>  
<url-pattern>/deleteAuthor</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
<servlet-name>GetAuthors</servlet-name>  
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>GetAuthors</servlet-name>  
<url-pattern>/getAuthors</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
<servlet-name>BookAddForm</servlet-name>  
<servlet-class>com.thinking.machines.library.servlets.BookAddForm</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>BookAddForm</servlet-name>  
<url-pattern>/BookAddForm.html</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
<servlet-name>AddBook</servlet-name>  
<servlet-class>com.thinking.machines.library.servlets.AddBook</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>AddBook</servlet-name>  
<url-pattern>/addBook</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
<servlet-name>GetBooks</servlet-name>  
<servlet-class>com.thinking.machines.library.servlets.GetBooks</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>GetBooks</servlet-name>  
<url-pattern>/getBooks</url-pattern>  
</servlet-mapping>
```

```
<servlet>  
<servlet-name>DeleteBook</servlet-name>  
<servlet-class>com.thinking.machines.library.servlets.DeleteBook</servlet-class>  
</servlet>  
<servlet-mapping>  
<servlet-name>DeleteBook</servlet-name>
```

```

<url-pattern>/deleteBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditBook</servlet-name>
<url-pattern>/editBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateBook</servlet-name>
<url-pattern>/updateBook</url-pattern>
</servlet-mapping>
</web-app>

```

To compile the servlets

On Windows

```
javac -classpath c:\tomcat8\lib\*;c:\tomcat8\webapps\crudone\WEB-INF\classes;. *.java
```

On Linux

```
replace c:\tomcat8 with your apache-tomcat folder path, replace all ; with : and \ with /
```

AddAuthor.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{

```

```

authorDAO.add(author);
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author added successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add more authors</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");

```

```
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim());");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var");
pw.println("validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .!\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))!=-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Author </h2>");
```

```

pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/addAuthor' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='36' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;   </span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}
}

```

The Author Edit Module

AuthorEditForm.html (Location – (crudone) folder)

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">

```

```
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
</script>
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</tr>
</table>
</div>
<br>
<br>
<h2>Edit Author </h2>
<form action='/crudone/editAuthor' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
```

```
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Edit</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040
</center>
</div>
</body>
</html>
```

EditAuthor.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class EditAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
```

```
author=authorDAO.getByName(name);
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim());");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required'");");
pw.println("frm.name.focus());");
pw.println("return false;");");
pw.println("}");
pw.println("var");
validCharacters='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .';");
pw.println("var e=0;");");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Invalid characters in name of author'");");
pw.println("frm.name.focus());");
pw.println("return false;");");
pw.println("}");
pw.println("e++;");");
pw.println("}");
pw.println("return true;");");
pw.println("}");");
pw.println("</script>");
pw.println("</head>");");
pw.println("<body style='background:#F5F5F5'>");");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");");
pw.println("<table width='100%'>");");
pw.println("<tr>");");
pw.println("<td>");");
pw.println("<img src='/crudone/images/logo.png'><br>");");
pw.println("</td>");");
pw.println("<td align='right' style='padding:5px'>");");
pw.println("<a href='/crudone/'>Home</a>");");
pw.println("</td>");");
```



```

pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<form action='/crudone/updateAuthor' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+">");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code</td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+">");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;  </span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");

```

```
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required'");");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var");
validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Invalid characters in name of author'");");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
```

```

pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/editAuthor' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Edit</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}

```

UpdateAuthor.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

```

```
public class UpdateAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
AuthorInterface author;
AuthorDAO authorDAO=new AuthorDAO();
try
{
author=authorDAO.getByCode(code);
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>That author cannot be updated as someone working in parallel deleted the author while
you were trying to update it.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorEditForm.html'>");
pw.println("<Button type='submit'>Edit other author</button>");
pw.println("</form>");
```

```
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
author.setName(name);
try
{
authorDAO.update(author);
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author updated successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorEditForm.html'>");
pw.println("<Button type='submit'>Edit more authors</button>");
pw.println("</form>");
pw.println("</div>");
```

```
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required'");
pw.println("frm.name.focus()");
pw.println("return false;");
pw.println("}");
pw.println("var");
validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\");");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Invalid characters in name of author'");
pw.println("frm.name.focus()");
pw.println("return false;");
pw.println("}");
pw.println("e++;");
pw.println("}");
pw.println("return true;");
pw.println("}");
pw.println("</script>");
```

```
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/updateAuthor' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+">");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code</td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+">");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
```

```

pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}
}

```

The Author Delete Module AuthorDeleteForm.html

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<script>
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ.";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
}
}

```



```

e++;
}
return true;
}
</script>
</head>
<body style='background:#F5F5F5'>
<div style='background:#DBDBDB;width:100%;border:2px'>
<table width='100%'>
<tr>
<td>
<img src='/crudone/images/logo.png'><br>
</td>
<td align='right' style='padding:5px'>
<a href='/crudone/'>Home</a>
</td>
</tr>
</table>
</div>
<br>
<br>
<h2>Delete Author </h2>
<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)' method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' style='color:red'>&nbsp;  </span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>
<center>
&copy; Thinking Machines 2017-2040

```

```
</center>
</div>
</body>
</html>
```

GetAuthorForDeletion.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class GetAuthorForDeletion extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorDAO authorDAO=new AuthorDAO();
AuthorInterface author;
try
{
author=authorDAO.getByName(name);
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(author.getCode());
if(bookWithAuthorExists)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required';");
```

```

pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("{}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{}");
pw.println("if(validCharacters.indexOf(name.charAt(e))!=-1)");
pw.println("{}");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("{}");
pw.println("e++;");
pw.println("{}");
pw.println("return true;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>Cannot delete author as book exists against it.</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");

```

```

pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
} catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim());");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required'");");
pw.println("frm.name.focus());");
pw.println("return false;");");
pw.println("}");");
pw.println("var
validCharacters='abcdefghijklmnopqrstuvwxzyABCDEFGHIJKLMNOPQRSTUVWXYZ .'");");

```

```

pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{}");
pw.println("if(validCharacters.indexOf(name.charAt(e))!=-1)");
pw.println("{}");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("{}");
pw.println("e++;");
pw.println("{}");
pw.println("return true;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<form action='/crudone/deleteAuthor' onsubmit='return validateForm(this)'");
pw.println("method='post'>");
pw.println("<input type='hidden' name='code' id='code' value='"+author.getCode()+">");
pw.println("<input type='hidden' name='name' id='name' value='"+author.getName()+">");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Code : </td>");
pw.println("<td>");
pw.println("<b>"+author.getCode()+"</b>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Name of author : </td>");
pw.println("<td>");
pw.println(author.getName());
pw.println("</td>");
pw.println("</tr>");

```

```

pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById('nameErrorSection');");
pw.println("nameErrorSection.innerHTML='';");
pw.println("if(name.length==0)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Required'");");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("}");
pw.println("var");
validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\");");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{");
pw.println("if(validCharacters.indexOf(name.charAt(e))==-1)");
pw.println("{");
pw.println("nameErrorSection.innerHTML='Invalid characters in name of author'");");
pw.println("frm.name.focus();");

```

```
pw.println("return false;");
pw.println("{}");
pw.println("e++;");
pw.println("{}");
pw.println("return true;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
```

```

pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}

```

DeleteAuthor.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class DeleteAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
int code=Integer.parseInt(request.getParameter("code"));
String name=request.getParameter("name");
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(code);
if(bookWithAuthorExists)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");

```



```
pw.println("{}");
pw.println("var name=frm.name.value.trim();");
pw.println("var nameErrorSection=document.getElementById(\"nameErrorSection\");");
pw.println("nameErrorSection.innerHTML=\"\";");
pw.println("if(name.length==0)");
pw.println("{}");
pw.println("nameErrorSection.innerHTML=\"Required\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("{}");
pw.println("var
validCharacters=\"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ .\";");
pw.println("var e=0;");
pw.println("while(e<name.length)");
pw.println("{}");
pw.println("if(validCharacters.indexOf(name.charAt(e))!=-1)");
pw.println("{}");
pw.println("nameErrorSection.innerHTML=\"Invalid characters in name of author\";");
pw.println("frm.name.focus();");
pw.println("return false;");
pw.println("{}");
pw.println("e++;");
pw.println("{}");
pw.println("return true;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Delete Author </h2>");
pw.println("<div style='color:red'>Cannot delete author as book exists against it.</div>");
pw.println("<form action='/crudone/getAuthorForDeletion' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
```

```

pw.println("<td>Name of author</td>");
pw.println("<td>");
pw.println("<input type='text' name='name' id='name' maxlength='35' size='31' value='"+name+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='right'>");
pw.println("<span id='nameErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Delete</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
return;
}
} catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.remove(code);
} catch(DAOException daoException)
{
}
}
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");

```

```

pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Author deleted successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorDeleteForm.html'>");
pw.println("<Button type='submit'>Delete more authors</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}

```

The Author List Module GetAuthor.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

```

```

import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Authors </h2>");
pw.println("<table style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Author</th>");
pw.println("<th>Code</th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(AuthorInterface author:authors)

```

```

{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+author.getName()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+author.getCode()+"</td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");

```

```

pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Authors </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}
}

```

The Book Add Module BookAddForm.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class BookAddForm extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");

```

```
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
try
{
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode===-1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
```

```
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null);");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0);");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length);");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))==-1);");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null);");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Book</h2>");
```



```

pw.println("<form action='/crudone/addBook' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
pw.println("<option value='"+author.getCode()+"'+author.getName()+"</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
pw.println("<option value='Science fiction'>Science fiction</option>");
pw.println("<option value='Satire'>Satire</option>");
pw.println("<option value='Drama'>Drama</option>");
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
pw.println("<option value='Mystery'>Mystery</option>");
pw.println("<option value='Horror'>Horror</option>");
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");

```

```

pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;  </span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");

```

```

pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Add Book </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added, cannot add book without adding authors.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add author</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
}
}
}
}
}
}
}
}

```

AddBook.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
String title=request.getParameter("title");
int authorCode=Integer.parseInt(request.getParameter("authorCode"));

```

```
String category=request.getParameter("category");
int price=Integer.parseInt(request.getParameter("price"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
BookInterface book=new Book();
book.setTitle(title);
book.setAuthorCode(authorCode);
book.setCategory(category);
book.setPrice(price);
try
{
bookDAO.add(book);
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'> ");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Notification</h2>");
pw.println("<div style='font-size:24pt'>");
pw.println("<b>Book added successfully</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/BookAddForm.html'>");
pw.println("<Button type='submit'>Add more books</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'> ");
```

```

pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
try
{
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode== -1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");

```

```
pw.println("if(errorComponent==null)");
pw.println("{}");
pw.println("errorComponent=frm.authorCode;");
pw.println("{}");
pw.println("{}");
pw.println("if(category=='none')");
pw.println("{}");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{}");
pw.println("errorComponent=frm.category;");
pw.println("{}");
pw.println("{}");
pw.println("if(price.length==0)");
pw.println("{}");
pw.println("frm.price.value=\"0\";");
pw.println("{}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{}");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{}");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{}");
pw.println("errorComponent=frm.price;");
pw.println("{}");
pw.println("break;");
pw.println("{}");
pw.println("i++;");
pw.println("{}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
```

```

pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Add Book</h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/addBook' onsubmit='return validateForm(this)' method='post'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");

```

```
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
{
pw.println("<option selected value='Horror'>Horror</option>");
}
}
```



```

else
{
pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value='' +price+'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Add</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException2)
{
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");

```

```

pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Add Book </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No authors added, cannot add book without adding authors.</b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/AuthorAddForm.html'>");
pw.println("<Button type='submit'>Add author</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
}
} catch(Exception exception)
{
}
}
}
}
}
}
}

```

The Book List With Edit/Delete Options

GetBooks.java

```
package com.thinking.machines.library.servlets;
```

```
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetBooks extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
BookDAO bookDAO=new BookDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
LinkedList<BookInterface> books;
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang=\"en\">");
pw.println("<head>");
pw.println("<meta charset=\"utf-8\">");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name=\"description\" content=\"The Whatever Corporation\">");
pw.println("<meta name=\"author\" content=\"Thinking Machines\">");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
```

```

pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<table style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface book:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+book.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==book.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
}
pw.println("<td align='right' style='border:1px solid black'>"+book.getCategory()+"</td>");

```

```

pw.println("<td align='right' style='border:1px solid black'>"+book.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+book.getCode()+"><img src='/crudone/images/edit_icon.png'
style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+book.getCode()+",\""+book.getTitle()+"')><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");

```

```

pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No books added </b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}
}

```

EditBook.java

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class EditBook extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
int code=Integer.parseInt(request.getParameter("code"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookInterface book;
BookDAO bookDAO=new BookDAO();

```

```
try
{
book=bookDAO.getByCode(code);
String title=book.getTitle();
int authorCode=book.getAuthorCode();
String category=book.getCategory();
int price=book.getPrice();
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim()");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim()");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML=\"Required\"");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode== -1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML=\"Required\"");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
```

```
pw.println("{}");
pw.println("{}");
pw.println("if(category=='none')");
pw.println("{}");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{}");
pw.println("errorComponent=frm.category;");
pw.println("{}");
pw.println("{}");
pw.println("if(price.length==0)");
pw.println("{}");
pw.println("frm.price.value=\"0\";");
pw.println("{}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{}");
pw.println("if(v.indexOf(price.charAt(i))==-1)");
pw.println("{}");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{}");
pw.println("errorComponent=frm.price;");
pw.println("{}");
pw.println("break;");
pw.println("{}");
pw.println("i++;");
pw.println("{}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("{}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
```



```

pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Book</h2>");
pw.println("<form action='/crudone/updateBook' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' id='code' name='code' value='"+code+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+">" +author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+">" +author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;&nbsp;&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");

```

```
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
{
pw.println("<option selected value='Horror'>Horror</option>");
}
else
{
```

```

pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value='"+price+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
LinkedList<BookInterface> books;

```

```

books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<table style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");

```

```

pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface b:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+b.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==b.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+b.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+b.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+b.getCode()+"'><img src='/crudone/images/edit_icon.png' style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook(\""+b.getCode()+"','\""+b.getTitle()+"')'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");

```

```
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
} catch (DAOException daoException2)
{
// this case won't arise
}
}
} catch (Exception exception)
{
}
}
}
```

UpdateBook.java

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class UpdateBook extends HttpServlet
{
public void doPost(HttpServletRequest request, HttpServletResponse response)
{
try
{
int code=Integer.parseInt(request.getParameter("code"));
String title=request.getParameter("title");
int authorCode=Integer.parseInt(request.getParameter("authorCode"));
String category=request.getParameter("category");
int price=Integer.parseInt(request.getParameter("price"));
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
BookDAO bookDAO=new BookDAO();
BookInterface book=new Book();
book.setCode(code);
book.setTitle(title);
book.setAuthorCode(authorCode);
book.setCategory(category);
book.setPrice(price);
try
{
bookDAO.update(book);
LinkedList<AuthorInterface> authors;
AuthorDAO authorDAO=new AuthorDAO();
```

```

authors=authorDAO.getAll();
LinkedList<BookInterface> books;
books=bookDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<h3>Book title : "+title+" updated.</h3>");
pw.println("<table style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");

```

```

pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface b:books)
{
sno++;
if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border: 1px solid black'>"+sno+"</td>");
pw.println("<td style='border: 1px solid black'>"+b.getTitle()+"</td>");
pw.println("<td align='right' style='border: 1px solid black'>"+b.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==b.getAuthorCode())
{
pw.println("<td align='right' style='border: 1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border: 1px solid black'>"+b.getCategory()+"</td>");
pw.println("<td align='right' style='border: 1px solid black'>"+b.getPrice()+"</td>");
pw.println("<td align='right' style='border: 1px solid black'><a href='/crudone/editBook?
code="+b.getCode()+"'><img src='/crudone/images/edit_icon.png' style='padding:5px'></a></td>");
pw.println("<td align='right' style='border: 1px solid black'><a
href='javascript:deleteBook("+b.getCode()+",\""+b.getTitle()+"')'><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");

```



```

pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='POST'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
} catch (DAOException daoException)
{
try
{
AuthorDAO authorDAO=new AuthorDAO();
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("<script>");
pw.println("function validateForm(frm)");
pw.println("{");
pw.println("var title=frm.title.value.trim();");
pw.println("var titleErrorSection=document.getElementById('titleErrorSection');");
pw.println("titleErrorSection.innerHTML='&nbsp;';");
pw.println("var authorCode=frm.authorCode.value;");
pw.println("var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');");
pw.println("authorCodeErrorSection.innerHTML='&nbsp;';");
pw.println("var category=frm.category.value;");
pw.println("var categoryErrorSection=document.getElementById('categoryErrorSection');");
pw.println("categoryErrorSection.innerHTML='&nbsp;';");
pw.println("var price=frm.price.value.trim();");
pw.println("var priceErrorSection=document.getElementById('priceErrorSection');");
pw.println("priceErrorSection.innerHTML='&nbsp;';");
pw.println("var errorComponent=null;");
pw.println("var valid=true;");
pw.println("if(title.length==0)");
pw.println("{");
pw.println("titleErrorSection.innerHTML='Required'");
pw.println("valid=false;");
pw.println("errorComponent=frm.title;");
pw.println("}");
pw.println("if(authorCode== -1)");
pw.println("{");
pw.println("authorCodeErrorSection.innerHTML='Required'");

```

```
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.authorCode;");
pw.println("}");
pw.println("}");
pw.println("if(category=='none')");
pw.println("{");
pw.println("categoryErrorSection.innerHTML=\"Required\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.category;");
pw.println("}");
pw.println("}");
pw.println("if(price.length==0)");
pw.println("{");
pw.println("frm.price.value=\"0\";");
pw.println("}");
pw.println("v=\"0123456789\";");
pw.println("var i=0;");
pw.println("while(i<price.length)");
pw.println("{");
pw.println("if(v.indexOf(price.charAt(i))== -1)");
pw.println("{");
pw.println("priceErrorSection.innerHTML=\"Invalid\";");
pw.println("valid=false;");
pw.println("if(errorComponent==null)");
pw.println("{");
pw.println("errorComponent=frm.price;");
pw.println("}");
pw.println("break;");
pw.println("}");
pw.println("i++;");
pw.println("}");
pw.println("if(!valid) errorComponent.focus();");
pw.println("return valid;");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
```

```

pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<h2>Edit Book</h2>");
pw.println("<div style='color:red'>"+daoException.getMessage()+"</div>");
pw.println("<form action='/crudone/updateBook' onsubmit='return validateForm(this)'
method='post'>");
pw.println("<input type='hidden' id='code' name='code' value='"+code+"'>");
pw.println("<table border='0'>");
pw.println("<tr>");
pw.println("<td>Title</td>");
pw.println("<td>");
pw.println("<input type='text' name='title' id='title' maxlength='35' size='31' value='"+title+"'>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='titleErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Author</td>");
pw.println("<td>");
pw.println("<select id='authorCode' name='authorCode'>");
pw.println("<option value='-1'>&lt;Select&gt;</option>");
for(AuthorInterface author:authors)
{
if(author.getCode()==authorCode)
{
pw.println("<option selected value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
else
{
pw.println("<option value='"+author.getCode()+"'>"+author.getName()+"</option>");
}
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='authorCodeErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");

```

```
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Category</td>");
pw.println("<td>");
pw.println("<select id='category' name='category'>");
pw.println("<option value='none'>&lt;Select&gt;</option>");
if(category.equals("Science fiction"))
{
pw.println("<option selected value='Science fiction'>Science fiction</option>");
}
else
{
pw.println("<option value='Science fiction'>Science fiction</option>");
}
if(category.equals("Satire"))
{
pw.println("<option selected value='Satire'>Satire</option>");
}
else
{
pw.println("<option value='Satire'>Satire</option>");
}
if(category.equals("Drama"))
{
pw.println("<option selected value='Drama'>Drama</option>");
}
else
{
pw.println("<option value='Drama'>Drama</option>");
}
if(category.equals("Action and Adventure"))
{
pw.println("<option selected value='Action and Adventure'>Action and Adventure</option>");
}
else
{
pw.println("<option value='Action and Adventure'>Action and Adventure</option>");
}
if(category.equals("Mystery"))
{
pw.println("<option selected value='Mystery'>Mystery</option>");
}
else
{
pw.println("<option value='Mystery'>Mystery</option>");
}
if(category.equals("Horror"))
```

```

{
pw.println("<option selected value='Horror'>Horror</option>");
}
else
{
pw.println("<option value='Horror'>Horror</option>");
}
pw.println("</select>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='categoryErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td>Price</td>");
pw.println("<td>");
pw.println("<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right'
value=''+price+'">");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td></td><td>");
pw.println("<span id='priceErrorSection' style='color:red'>&nbsp;</span>");
pw.println("</td>");
pw.println("</tr>");
pw.println("<tr>");
pw.println("<td colspan='2' align='center'>");
pw.println("<button type='submit'>Update</button>");
pw.println("</td>");
pw.println("</tr>");
pw.println("</table>");
pw.println("</form>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}catch(DAOException daoException2)
{
// this case won't arise

```



```

pw.println("<script>");
pw.println("function deleteBook(bookCode,bookTitle)");
pw.println("{");
pw.println("var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');");
pw.println("if(c)");
pw.println("{");
pw.println("var deleteBookForm=document.getElementById('deleteBookForm');");
pw.println("deleteBookForm.code.value=bookCode;");
pw.println("deleteBookForm.title.value=bookTitle;");
pw.println("deleteBookForm.submit();");
pw.println("}");
pw.println("}");
pw.println("</script>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<h3>Book : "+title+" deleted.</h3>");
pw.println("<table style='border:1px solid black'>");
pw.println("<thead>");
pw.println("<tr>");
pw.println("<th>S.No.</th>");
pw.println("<th>Title</th>");
pw.println("<th>Code</th>");
pw.println("<th>Author</th>");
pw.println("<th>Category</th>");
pw.println("<th>Price</th>");
pw.println("<th></th>");
pw.println("<th></th>");
pw.println("</tr>");
pw.println("</thead>");
pw.println("<tbody>");
int sno=0;
for(BookInterface book:books)
{
sno++;

```

```

if(sno%2==0)
{
pw.println("<tr style='background:#DDDDDD'>");
}
else
{
pw.println("<tr>");
}
pw.println("<td align='right' style='border:1px solid black'>"+sno+"</td>");
pw.println("<td style='border:1px solid black'>"+book.getTitle()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getCode()+"</td>");
for(AuthorInterface author:authors)
{
if(author.getCode()==book.getAuthorCode())
{
pw.println("<td align='right' style='border:1px solid black'>"+author.getName()+"</td>");
break;
}
}
pw.println("<td align='right' style='border:1px solid black'>"+book.getCategory()+"</td>");
pw.println("<td align='right' style='border:1px solid black'>"+book.getPrice()+"</td>");
pw.println("<td align='right' style='border:1px solid black'><a href='/crudone/editBook?
code="+book.getCode()+"><img src='/crudone/images/edit_icon.png'
style='padding:5px'></a></td>");
pw.println("<td align='right' style='border:1px solid black'><a
href='javascript:deleteBook("+book.getCode()+",\""+book.getTitle()+"')><img
src='/crudone/images/delete_icon.png' style='padding:5px'></a></td>");
pw.println("</tr>");
}
pw.println("</tbody>");
pw.println("</table>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("<form id='deleteBookForm' action='/crudone/deleteBook' method='post'>");
pw.println("<input type='hidden' name='code' id='code'>");
pw.println("<input type='hidden' name='title' id='title'>");
pw.println("</form>");
pw.println("</body>");
pw.println("</html>");
} catch(DAOException daoException)
{

```



```

System.out.println(daoException); // don't forget to remove this after testing
pw.println("<!doctype html>");
pw.println("<html lang='en'>");
pw.println("<head>");
pw.println("<meta charset='utf-8'>");
pw.println("<title>Whatever Corporation</title>");
pw.println("<meta name='description' content='The Whatever Corporation'>");
pw.println("<meta name='author' content='Thinking Machines'>");
pw.println("</head>");
pw.println("<body style='background:#F5F5F5'>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px'>");
pw.println("<table width='100%'>");
pw.println("<tr>");
pw.println("<td>");
pw.println("<img src='/crudone/images/logo.png'><br>");
pw.println("</td>");
pw.println("<td align='right' style='padding:5px'>");
pw.println("<a href='/crudone/'>Home</a>");
pw.println("</td>");
pw.println("</table>");
pw.println("</div>");
pw.println("<br>");
pw.println("<h2>Books </h2>");
pw.println("<br>");
pw.println("<div style='font-size:24pt;color:red'>");
pw.println("<b>No books added </b>");
pw.println("<br><br>");
pw.println("<form action='/crudone/'>");
pw.println("<Button type='submit'>Ok</button>");
pw.println("</form>");
pw.println("</div>");
pw.println("<br>");
pw.println("<br>");
pw.println("<br>");
pw.println("<div style='background:#DBDBDB;width:100%;border:2px;padding:10px'>");
pw.println("<center>");
pw.println("&copy; Thinking Machines 2017-2040");
pw.println("</center>");
pw.println("</div>");
pw.println("</body>");
pw.println("</html>");
}
} catch(Exception exception)
{
}
}
}

```

}

--- CRUD Two Complete ---

CRUD Two

JSP – Java Beans – TagSupport – Servlets

Bridging the gap between a designer and a programmer.

Download & See it in Action

<http://tm-certificates.com/CRUDTwo.wmv>

Create folder named as crudtwo in webapps

The following folder structures need to be created

crudtwo – WEB-INF
crudtwo – WEB-INF – classes
crudtwo – WEB-INF – lib
crudtwo – WEB-INF – tlds
crudtwo – images
crudtwo – js
crudtwo – css

all the JSP files should be kept in crudtwo folder
copy the images from crudone\images to crudtwo\images

Create the necessary folders for packages

crudtwo – WEB-INF – classes – com – thinking – machines – library – dl
crudtwo – WEB-INF – classes – com – thinking – machines – library – beans
crudtwo – WEB-INF – classes – com – thinking – machines – library – tags
crudtwo – WEB-INF – classes – com – thinking – machines – library – servlets

copy the contents of the crudone\WEB-INF\classes\com\thinking\machines\library\dl to
crudtwo\WEB-INF\classes\com\thinking\machines\library\dl
We won't be rewriting the data layer classes, they are going to stay same.

The CustomTags.tld file should be placed in crudtwo\WEB-INF\tlds folder (the second char is l for lion and not digit 1)

The web.xml should be placed in crudtwo\WEB-INF folder
The mysql.jar file should be in the crudtwo\WEB-INF\lib folder

The files with extension js should be placed in the crudtwo\js folder
the files with extension css should be placed in the crudtwo\css folder

The Backbone

**We are wearing the shoes of a
Programmer and know nothing about
designing.**

web.xml (Location WEB-INF folder)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
```

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
```

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"  
  version="3.1"  
  metadata-complete="true">
```

```
  <description>
```

```
    CRUD Operations - Style One
```

```
  </description>
```

```
  <display-name>CRUD Operations - Style One</display-name>
```

```
<servlet>
```

```
  <servlet-name>AddAuthor</servlet-name>
```

```
  <servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
```

```
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditAuthor</servlet-name>
<url-pattern>/editAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateAuthor</servlet-name>
<url-pattern>/updateAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetAuthorForDeletion</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthorForDeletion</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthorForDeletion</servlet-name>
<url-pattern>/getAuthorForDeletion</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>DeleteAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteAuthor</servlet-name>
<url-pattern>/deleteAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>AddBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddBook</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>AddBook</servlet-name>
<url-pattern>/addBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>DeleteBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.DeleteBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DeleteBook</servlet-name>
<url-pattern>/deleteBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>EditBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.EditBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>EditBook</servlet-name>
<url-pattern>/editBook</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>UpdateBook</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.UpdateBook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>UpdateBook</servlet-name>
<url-pattern>/updateBook</url-pattern>
</servlet-mapping>

</web-app>
```

CustomTags.tld (location WEB-INF\tlds folder)

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
```

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

-->

```
<!DOCTYPE taglib
  PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
  "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">
```

```
<taglib>
```

```
<tlib-version>1.0</tlib-version>
```

```
<jsp-version>1.2</jsp-version>
```

```
<short-name>CustomTags</short-name>
```

```
<description>
```

```
A tag library for this project
```

```
</description>
```

```
<tag>
```

```
<name>SetEnvironmentVariables</name>
```

```
<tag-class>com.thinking.machines.library.tags.EnvironmentVariablesTagHandler</tag-class>
```

```
<description> Environment Variables Tag Handler </description>
```

```
</tag>
```

```
<tag>
```

```
<name>FormId</name>
```

```
<tag-class>com.thinking.machines.library.tags.FormIdTagHandler</tag-class>
```

```
<description> Form Id. Generator Tag </description>
```

```
</tag>
```

```
<tag>
```

```
<name>FormResubmitted</name>
```

```
<tag-class>com.thinking.machines.library.tags.FormResubmittedTagHandler</tag-class>
```

```
<description> Process body if form is resubmitted </description>
```

```
</tag>
```

```
<tag>
```

```
<name>IfModule</name>
```

```
<tag-class>com.thinking.machines.library.tags.IfModuleTagHandler</tag-class>
```

```
<description> Tag to identify module page </description>
```

```
</tag>
```

```
<tag>
```

```
<name>If</name>
```

```
<tag-class>com.thinking.machines.library.tags.IfTagHandler</tag-class>
```

```
<description> Tag to identify module page </description>
```

```
<attribute>
```

```
<name>expression</name>
```

```
<required>true</required>
```

```
<rtexprvalue>true</rtexprvalue>
```

```
</attribute>
```

```
</tag>
```



```

<tag>
  <name>Author</name>
  <tag-class>com.thinking.machines.library.tags.AuthorTagHandler</tag-class>
  <tei-class>com.thinking.machines.library.tags.AuthorTEI</tei-class>
  <description> Author Tag Handler </description>
</tag>
<tag>
  <name>Book</name>
  <tag-class>com.thinking.machines.library.tags.BookTagHandler</tag-class>
  <tei-class>com.thinking.machines.library.tags.BookTEI</tei-class>
  <description> Book Tag Handler </description>
  <attribute>
    <name>authorCode</name>
    <required>false</required>
    <rtexprvalue>>true</rtexprvalue>
  </attribute>
</tag>
</taglib>

```

To compile the java files

```
javac -classpath c:\tomcat8\lib\*;c:\tomcat8\webapps\crudtwo\WEB-INF\classes;. *.java
```

Note : You know what to do in case of LINUX

Data Layer classes (Same as crudone)

```

package com.thinking.machines.library.beans;
public class NotificationBean implements java.io.Serializable
{
private String message;
private String actionText;
private String actionURL;
public NotificationBean()
{
this.message="";
}
public void setMessage(String message)
{
this.message=message;
}
public String getMessage()
{
return this.message;
}
public void setActionText(String actionText)
{
this.actionText=actionText;
}
}

```

```
public String getActionText()
{
return this.actionText;
}
public void setActionURL(String actionURL)
{
this.actionURL=actionURL;
}
public String getActionURL()
{
return this.actionURL;
}
}
```

```
package com.thinking.machines.library.beans;
public class ErrorBean implements java.io.Serializable
{
private String errorMessage;
public ErrorBean()
{
this.errorMessage="";
}
public void setErrorMessage(String errorMessage)
{
this.errorMessage=errorMessage;
}
public String getErrorMessage()
{
return this.errorMessage;
}
}
```

```
package com.thinking.machines.library.beans;
public class AuthorBean implements java.io.Serializable,Comparable<AuthorBean>
{
private int code;
public String name;
public AuthorBean()
{
this.code=0;
this.name="";
}
public void setCode(int code)
{
this.code=code;
}
public int getCode()
{
```



```
{
return this.code;
}
public void setTitle(String title)
{
this.title=title;
}
public String getTitle()
{
return this.title;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public void setCategory(String category)
{
this.category=category;
}
public String getCategory()
{
return this.category;
}
public void setPrice(int price)
{
this.price=price;
}
public int getPrice()
{
return this.price;
}
public boolean equals(Object object)
{
if(!(object instanceof BookBean)) return false;
BookBean otherBookBean=(BookBean)object;
return this.code==otherBookBean.code;
}
public int compareTo(BookBean otherBookBean)
{
return this.code-otherBookBean.getCode();
}
public int hashCode()
{
```

```
return this.code;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
public class IfTagHandler extends TagSupport
{
private boolean expression;
public void setExpression(boolean expression)
{
this.expression=expression;
}
public boolean getExpression()
{
return this.expression;
}
public int doStartTag()
{
if(expression) return EVAL_BODY_INCLUDE;
return SKIP_BODY;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
public class IfModuleTagHandler extends TagSupport
{
public int doStartTag()
{
HttpServletRequest request=(HttpServletRequest)pageContext.getRequest();
if(request.getRequestURI().endsWith("/") || request.getRequestURI().endsWith("/index.jsp")) return
SKIP_BODY;
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
```

```
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class FormIdTagHandler extends TagSupport
{
public int doStartTag()
{
try
{
String formId=java.util.UUID.randomUUID().toString();
pageContext.setAttribute(formId,formId,pageContext.SESSION_SCOPE);
pageContext.getOut().write("<input type='hidden' id='formId' name='formId' value='"+formId+"'>");
} catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.beans.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public class FormResubmittedTagHandler extends TagSupport
{
public int doStartTag()
{
try
{
HttpServletRequest request=(HttpServletRequest)pageContext.getRequest();
String formId=request.getParameter("formId");
if(formId==null)
{
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("You should not refresh / resubmit data.");
notificationBean.setActionText("Ok");
notificationBean.setActionURL("/");
}
```

```

pageContext.setAttribute("notificationBean",notificationBean,PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
HttpSession session=request.getSession();
if(session.getAttribute(formId)==null)
{
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("You should not refresh / resubmit data.");
notificationBean.setActionText("Ok");
notificationBean.setActionURL("/");
pageContext.setAttribute("notificationBean",notificationBean,PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
session.removeAttribute(formId);
} catch(Exception exception)
{
System.out.println(exception); // dont' forget to remove after testing
}
return SKIP_BODY;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}

```

```

package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
public class EnvironmentVariablesTagHandler extends TagSupport
{
public int doStartTag()
{
pageContext.setAttribute("contextName","crudtwo",PageContext.REQUEST_SCOPE);
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}

```

```

package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;

```

```
import javax.servlet.jsp.tagext.*;
public class AuthorTagHandler extends BodyTagSupport
{
private LinkedList<AuthorInterface> authors;
private int index;
private int serialNumber;
private void reset()
{
this.index=0;
this.serialNumber=0;
}
public int doStartTag()
{
this.index=0;
this.serialNumber=0;
try
{
AuthorDAO authorDAO=new AuthorDAO();
authors=authorDAO.getAll();
} catch (DAOException daoException)
{
return SKIP_BODY;
}
if(authors.size()==0) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_INCLUDE;
}
public int doAfterBody()
{
index++;
if(index==authors.size()) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_AGAIN;
}
public int doEndTag()
{
reset();
return EVAL_PAGE;
}
private void setScriptingVariables()
{
AuthorInterface author;
author=authors.get(index);
pageContext.setAttribute("serialNumber",new Integer(serialNumber));
pageContext.setAttribute("code",new Integer(author.getCode()));
}
```



```
pageContext.setAttribute("name",author.getName());
pageContext.setAttribute("evenRow",new Boolean(serialNumber%2==0));
pageContext.setAttribute("oddRow",new Boolean(serialNumber%2!=0));
}
}
```

```
package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class AuthorTEI extends TagExtraInfo
{
public VariableInfo[] getVariableInfo(TagData tagData)
{
VariableInfo[] variables=new VariableInfo[5];
variables[0]=new VariableInfo("serialNumber","java.lang.Integer",true,VariableInfo.NESTED);
variables[1]=new VariableInfo("code","java.lang.Integer",true,VariableInfo.NESTED);
variables[2]=new VariableInfo("name","java.lang.String",true,VariableInfo.NESTED);
variables[3]=new VariableInfo("evenRow","java.lang.Boolean",true,VariableInfo.NESTED);
variables[4]=new VariableInfo("oddRow","java.lang.Boolean",true,VariableInfo.NESTED);
return variables;
}
}
```

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
public class BookTagHandler extends BodyTagSupport
{
private LinkedList<BookInterface> books;
private LinkedList<AuthorInterface> authors;
private int index;
private int serialNumber;
private int authorCode;
private void reset()
{
this.index=0;
this.serialNumber=0;
this.authorCode=0;
}
public void setAuthorCode(int authorCode)
{
this.authorCode=authorCode;
}
public int getAuthorCode()
{
return this.authorCode;
}
public int doStartTag()
```

```
{
this.index=0;
this.serialNumber=0;
try
{
AuthorDAO authorDAO=new AuthorDAO();
authors=authorDAO.getAll();
BookDAO bookDAO=new BookDAO();
if(authorCode==0)
{
books=bookDAO.getAll();
}
else
{
books=bookDAO.getAllByAuthorCode(authorCode);
}
} catch(DAOException daoException)
{
return SKIP_BODY;
}
if(books.size()==0) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_INCLUDE;
}
public int doAfterBody()
{
index++;
if(index==books.size()) return SKIP_BODY;
serialNumber++;
setScriptingVariables();
return EVAL_BODY_AGAIN;
}
public int doEndTag()
{
reset();
return EVAL_PAGE;
}
private void setScriptingVariables()
{
BookInterface book;
book=books.get(index);
pageContext.setAttribute("serialNumber",new Integer(serialNumber));
pageContext.setAttribute("code",new Integer(book.getCode()));
pageContext.setAttribute("title",book.getTitle());
pageContext.setAttribute("authorCode",new Integer(book.getAuthorCode()));
pageContext.setAttribute("authorName",new String(getAuthor(book.getAuthorCode()).getName()));
```

```

pageContext.setAttribute("category",book.getCategory());
pageContext.setAttribute("price",new Integer(book.getPrice()));
pageContext.setAttribute("evenRow",new Boolean(serialNumber%2==0));
pageContext.setAttribute("oddRow",new Boolean(serialNumber%2!=0));
}
private AuthorInterface getAuthor(int code)
{
for(AuthorInterface author:authors)
{
if(author.getCode()==code) return author;
}
return null; // this case won't arise
}
}
}
}

package com.thinking.machines.library.tags;
import javax.servlet.jsp.tagext.*;
public class BookTEI extends TagExtraInfo
{
public VariableInfo[] getVariableInfo(TagData tagData)
{
VariableInfo[] variables=new VariableInfo[9];
variables[0]=new VariableInfo("serialNumber","java.lang.Integer",true,VariableInfo.NESTED);
variables[1]=new VariableInfo("code","java.lang.Integer",true,VariableInfo.NESTED);
variables[2]=new VariableInfo("title","java.lang.String",true,VariableInfo.NESTED);
variables[3]=new VariableInfo("authorCode","java.lang.Integer",true,VariableInfo.NESTED);
variables[4]=new VariableInfo("authorName","java.lang.String",true,VariableInfo.NESTED);
variables[5]=new VariableInfo("category","java.lang.String",true,VariableInfo.NESTED);
variables[6]=new VariableInfo("price","java.lang.Integer",true,VariableInfo.NESTED);
variables[7]=new VariableInfo("evenRow","java.lang.Boolean",true,VariableInfo.NESTED);
variables[8]=new VariableInfo("oddRow","java.lang.Boolean",true,VariableInfo.NESTED);
return variables;
}
}
}

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");

```

```

AuthorInterface author=new Author();
author.setName(authorBean.getName());
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
authorBean.setCode(author.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Author : "+authorBean.getName()+" added successfully.");
notificationBean.setActionText("Add more authors");
notificationBean.setActionURL("/AuthorAddForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
} catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorAddForm.jsp");
requestDispatcher.forward(request,response);
}
} catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
}

```

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class EditAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
AuthorBean authorBean;
authorBean=(AuthorBean)request.getAttribute("authorBean");
AuthorInterface author;
AuthorDAO authorDAO=new AuthorDAO();

```



```
BookDAO bookDAO=new BookDAO();
try
{
boolean bookWithAuthorExists=bookDAO.containsBookWithAuthorCode(authorBean.getCode());
if(bookWithAuthorExists)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage("Cannot delete author as book exists against it.");
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteForm.jsp");
requestDispatcher.forward(request,response);
return;
}
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
try
{
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.remove(authorBean.getCode());
}catch(DAOException daoException)
{
System.out.println(daoException); // don't forget to remove after testing
}
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Author : "+authorBean.getName()+" deleted successfully.");
notificationBean.setActionText("Delete more authors");
notificationBean.setActionURL("/AuthorDeleteForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
```

```
}  
package com.thinking.machines.library.servlets;  
import com.thinking.machines.library.beans.*;  
import com.thinking.machines.library.dl.*;  
import javax.servlet.http.*;  
import javax.servlet.*;  
public class GetAuthorForDeletion extends HttpServlet  
{  
public void doPost(HttpServletRequest request,HttpServletResponse response)  
{  
try  
{  
AuthorBean authorBean;  
authorBean=(AuthorBean)request.getAttribute("authorBean");  
AuthorInterface author;  
AuthorDAO authorDAO=new AuthorDAO();  
try  
{  
author=authorDAO.getByName(authorBean.getName());  
authorBean.setCode(author.getCode());  
RequestDispatcher requestDispatcher;  
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteConfirmationForm.jsp");  
requestDispatcher.forward(request,response);  
}catch(DAOException daoException)  
{  
ErrorBean errorBean=new ErrorBean();  
errorBean.setErrorMessage(daoException.getMessage());  
request.setAttribute("errorBean",errorBean);  
RequestDispatcher requestDispatcher;  
requestDispatcher=request.getRequestDispatcher("/AuthorDeleteForm.jsp");  
requestDispatcher.forward(request,response);  
}  
}catch(Exception exception)  
{  
System.out.println(exception); // don't forget to remove after testing  
}  
}  
}
```

```
package com.thinking.machines.library.servlets;  
import com.thinking.machines.library.beans.*;  
import com.thinking.machines.library.dl.*;  
import javax.servlet.http.*;  
import javax.servlet.*;  
public class AddBook extends HttpServlet  
{  
public void doPost(HttpServletRequest request,HttpServletResponse response)
```



```
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book=new Book();
book.setTitle(bookBean.getTitle());
book.setAuthorCode(bookBean.getAuthorCode());
book.setCategory(bookBean.getCategory());
book.setPrice(bookBean.getPrice());
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.add(book);
bookBean.setCode(book.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" added successfully.");
notificationBean.setActionText("Add more books");
notificationBean.setActionURL("/BookAddForm.jsp");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Notification.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookAddForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
}
}

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class EditBook extends HttpServlet
{
```

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book;
BookDAO bookDAO=new BookDAO();
try
{
book=bookDAO.getByCode(bookBean.getCode());
bookBean.setTitle(book.getTitle());
bookBean.setAuthorCode(book.getAuthorCode());
bookBean.setCategory(book.getCategory());
bookBean.setPrice(book.getPrice());
NotificationBean notificationBean;
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookUpdateForm.jsp");
requestDispatcher.forward(request,response);
} catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}
} catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class UpdateBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
```

```
bookBean=(BookBean)request.getAttribute("bookBean");
BookInterface book;
BookDAO bookDAO=new BookDAO();
try
{
book=bookDAO.getByCode(bookBean.getCode());
book.setTitle(bookBean.getTitle());
book.setAuthorCode(bookBean.getAuthorCode());
book.setCategory(bookBean.getCategory());
book.setPrice(bookBean.getPrice());
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage("That book cannot be updated as someone working in parallel deleted the
author while you were trying to update it.");
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
return;
}
try
{
bookDAO.update(book);
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" updated successfully.");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/BookUpdateForm.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
```

```

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.beans.*;
import com.thinking.machines.library.dl.*;
import javax.servlet.http.*;
import javax.servlet.*;
public class DeleteBook extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
BookBean bookBean;
bookBean=(BookBean)request.getAttribute("bookBean");
BookDAO bookDAO=new BookDAO();
try
{
bookDAO.remove(bookBean.getCode());
NotificationBean notificationBean;
notificationBean=new NotificationBean();
notificationBean.setMessage("Book : "+bookBean.getTitle()+" deleted successfully.");
request.setAttribute("notificationBean",notificationBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}catch(DAOException daoException)
{
ErrorBean errorBean=new ErrorBean();
errorBean.setErrorMessage(daoException.getMessage());
request.setAttribute("errorBean",errorBean);
RequestDispatcher requestDispatcher;
requestDispatcher=request.getRequestDispatcher("/Books.jsp");
requestDispatcher.forward(request,response);
}
}catch(Exception exception)
{
System.out.println(exception); // don't forget to remove after testing
}
}
}
}
}

```

The following Java Script Files should be in crudtwo\js folder

AuthorAddForm.js

```

function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";

```

```
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorEditForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorUpdateForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
return false;
}
e++;
}
return true;
}
```

AuthorDeleteForm.js

```
function validateForm(frm)
{
var name=frm.name.value.trim();
var nameErrorSection=document.getElementById("nameErrorSection");
nameErrorSection.innerHTML="";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
frm.name.focus();
return false;
}
var validCharacters="abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ .";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
frm.name.focus();
}
```

```
return false;
}
e++;
}
return true;
}
```

BookAddForm.js

```
function validateForm(frm)
{
var title=frm.title.value.trim();
var titleErrorSection=document.getElementById('titleErrorSection');
titleErrorSection.innerHTML='&nbsp;';
var authorCode=frm.authorCode.value;
var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');
authorCodeErrorSection.innerHTML='&nbsp;';
var category=frm.category.value;
var categoryErrorSection=document.getElementById('categoryErrorSection');
categoryErrorSection.innerHTML='&nbsp;';
var price=frm.price.value.trim();
var priceErrorSection=document.getElementById('priceErrorSection');
priceErrorSection.innerHTML='&nbsp;';
var errorComponent=null;
var valid=true;
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
valid=false;
errorComponent=frm.title;
}
if(authorCode===-1)
{
authorCodeErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.authorCode;
}
}
if(category=='none')
{
categoryErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.category;
}
}
}
```

```
if(price.length==0)
{
frm.price.value="0";
}
v="0123456789";
var i=0;
while(i<price.length)
{
if(v.indexOf(price.charAt(i))!=-1)
{
priceErrorSection.innerHTML="Invalid";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.price;
}
break;
}
i++;
}
if(!valid) errorComponent.focus();
return valid;
}
```

BookUpdateForm.js

```
function validateForm(frm)
{
var title=frm.title.value.trim();
var titleErrorSection=document.getElementById('titleErrorSection');
titleErrorSection.innerHTML='&nbsp;';
var authorCode=frm.authorCode.value;
var authorCodeErrorSection=document.getElementById('authorCodeErrorSection');
authorCodeErrorSection.innerHTML='&nbsp;';
var category=frm.category.value;
var categoryErrorSection=document.getElementById('categoryErrorSection');
categoryErrorSection.innerHTML='&nbsp;';
var price=frm.price.value.trim();
var priceErrorSection=document.getElementById('priceErrorSection');
priceErrorSection.innerHTML='&nbsp;';
var errorComponent=null;
var valid=true;
if(title.length==0)
{
titleErrorSection.innerHTML="Required";
valid=false;
errorComponent=frm.title;
}
}
```



```
if(authorCode==-1)
{
authorCodeErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.authorCode;
}
}
if(category=='none')
{
categoryErrorSection.innerHTML="Required";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.category;
}
}
if(price.length==0)
{
frm.price.value="0";
}
v="0123456789";
var i=0;
while(i<price.length)
{
if(v.indexOf(price.charAt(i))==-1)
{
priceErrorSection.innerHTML="Invalid";
valid=false;
if(errorComponent==null)
{
errorComponent=frm.price;
}
break;
}
i++;
}
if(!valid) errorComponent.focus();
return valid;
}
```

Books.js

```
function deleteBook(bookCode,bookTitle)
{
var c=confirm('Delete Book : '+bookTitle+'('+bookCode+')');
if(c)
```

```

{
var deleteBookForm=document.getElementById('deleteBookForm');
deleteBookForm.code.value=bookCode;
deleteBookForm.title.value=bookTitle;
deleteBookForm.submit();
}
}

```

MasterPageTopSection.js

```

function initializeBody()
{
var history_api = typeof history.pushState !== 'undefined';
if(location.hash!=='#library') location.hash='#library';
if ( location.hash === '#library' ) {
if ( history_api )
history.pushState(null, '#librarian');
else
location.hash = '#librarian';
window.onhashchange = function() {
if ( location.hash === '#library' ) {
if ( history_api )
history.pushState(null, '#librarian');
else
location.hash = '#librarian';
}
};
}
}

```

```

window.addEventListener("load",initializeBody);

```

Back Bone Complete The UI Part Starts

Now we are wearing the shoes of a person who doesn't know how to write programming logics, but is well aware of html, designing and know how to follow guidelines.

styles.css (This file should be placed in crudtwo\css folder)

```

body {
background: #F5F5F5;
}

```

```

.home

```

```
{
padding: 5px;
}

.header {
background: #DBDBDB;
width: 100%;
}

.header table
{
width: 100%;
}

.footer {
background: #DBDBDB;
width: 100%;
padding: 10px;
text-align: center;
}

.error {
color: red;
}

.evenRow
{
background: #DDDDDD;
}

.oddRow
{
background: #F5F5F5;
}

.author-list {
border: 1px solid black;
}

.author-list tr td {
border: 1px solid black;
text-align: right;
}

.author-list tr td + td {
border: 1px solid black;
text-align: left;
```

```
}  
  
.author-list tr td + td + td {  
border: 1px solid black;  
text-align: right;  
}  
  
.book-list {  
border: 1px solid black;  
}  
  
.book-list tr td {  
border: 1px solid black;  
text-align: right;  
}  
  
.book-list tr td + td {  
border: 1px solid black;  
text-align: left;  
}  
  
.book-list tr td + td + td {  
border: 1px solid black;  
text-align: right;  
}  
  
.book-list tr td + td + td + td {  
border: 1px solid black;  
text-align: left;  
}  
  
.book-list tr td + td + td + td + td {  
border: 1px solid black;  
text-align: left;  
}  
  
.book-list tr td + td + td + td + td + td {  
border: 1px solid black;  
text-align: right;  
}  
  
.book-list tr td + td + td + td + td + td + td {  
border: 1px solid black;  
text-align: center;  
}  
  
.book-list tr td + td + td + td + td + td + td + td {
```

```
border: 1px solid black;
text-align: center;
}
```

MasterPageTopSection.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
</head>
<body>
<div class='header'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp

```
<div class='footer'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

index.jsp

```
<jsp:include page='/MasterPageTopSection.jsp' />
<h3>Author</h3>
<ul>
<li><a href='/${contextName}/AuthorAddForm.jsp'>Add</a></li>
<li><a href='/${contextName}/AuthorEditForm.jsp'>Edit</a></li>
<li><a href='/${contextName}/AuthorDeleteForm.jsp'>Delete</a></li>
<li><a href='/${contextName}/Authors.jsp'>List</a></li>
</ul>
```

```

<h3>Book</h3>
<ul>
<li><a href='/${contextName}/BookAddForm.jsp'>Add</a></li>
<li><a href='/${contextName}/Books.jsp'>Edit/Delete/List</a></li>
</ul>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

AuthorAddForm.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorAddForm.js'></script>
<h2>Add Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/AddAuthor.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;  </span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

AddAuthor.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='authorBean' scope='request'

```

```
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/addAuthor' />
```

AuthorEditForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorEditForm.js'></script>
<h2>Edit Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/EditAuthor.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;  </span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Edit</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

EditAuthor.jsp

```
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/editAuthor' />
```

AuthorUpdateForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
```



```
<jsp:forward page='/updateAuthor' />
```

AuthorDeleteForm.jsp

```
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/AuthorDeleteForm.js'></script>
<h2>Delete Author </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/GetAuthorForDeletion.jsp' onsubmit='return validateForm(this)'
method='post'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='nameErrorSection' class='error'>&nbsp;  </span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

GetAuthorForDeletion.jsp

```
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/getAuthorForDeletion' />
```

AuthorDeleteConfirmationForm.jsp

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<h2>Delete Author </h2>
```

```

<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/DeleteAuthor.jsp' onSubmit='return validateForm(this)'
method='post'>
<tm:FormId/>
<input type='hidden' id='code' name='code' value='${authorBean.code}' >
<input type='hidden' id='name' name='name' value='${authorBean.name}' >
<table border='0'>
<tr>
<td>Code</td>
<td>
${authorBean.code}
</td>
</tr>
<tr>
<td>Name of author</td>
<td>
${authorBean.name}
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Delete</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

DeleteAuthor.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:setProperty name='authorBean' property='*' />
<jsp:forward page='/deleteAuthor' />

```

Authors.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<h2>Authors</h2>
<table class='author-list'>
<thead>
<tr>

```

```

<th>S.No.</th>
<th>Author</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tm:Author>
<tm:If expression='${evenRow}'>
<tr class='evenRow'>
</tm:If>
<tm:If expression='${oddRow}'>
<tr class='oddRow'>
</tm:If>
<td>${serialNumber}</td>
<td>${name}</td>
<td>${code}</td>
</tr>
</tm:Author>
</tbody>
</table>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

BookAddForm.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/BookAddForm.js'></script>
<h2>Add Book </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/AddBook.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<table border='0'>
<tr>
<td>Title</td>
<td>
<input type='text' name='title' id='title' maxlength='35' size='31' value='${bookBean.title}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='titleErrorSection' class='error'>&nbsp;  </span>
</td>
</tr>

```

```

<tr>
<td>Author</td>
<td>
<select id='authorCode' name='authorCode'>
<option value='-1'>&lt;Select&gt;</option>
<tm:Author>
<tm.If expression='${bookBean.authorCode==code}'>
<option value='${code}' selected>${name}</option>
</tm.If>
<tm.If expression='${bookBean.authorCode!=code}'>
<option value='${code}'>${name}</option>
</tm.If>
</tm:Author>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='authorCodeErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction' ${bookBean.category=="Science fiction"}?"selected":"">Science
fiction</option>
<option value='Satire' ${bookBean.category=="Satire"}?"selected":"">Satire</option>
<option value='Drama' ${bookBean.category=="Drama"}?"selected":"">Drama</option>
<option value='Action and Adventure' ${bookBean.category=="Action and
Adventure"}?"selected":"">Action and Adventure</option>
<option value='Mystery' ${bookBean.category=="Mystery"}?"selected":"">Mystery</option>
<option value='Horror' ${bookBean.category=="Horror"}?"selected":"">Horror</option>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='categoryErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right' value='${
bookBean.price}'>

```

```

</td>
</tr>
<tr>
<td></td><td>
<span id='priceErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Add</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

AddBook.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/addBook' />

```

Books.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:useBean id='notificationBean' scope='request'
class='com.thinking.machines.library.beans.NotificationBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<jsp:include page='/MasterPageTopSection.jsp' />
<script src='/${contextName}/js/Books.js'></script>
<h2>Books</h2>
<tm:If expression='${notificationBean.message!=""}'>
<h3>${notificationBean.message}</h3>
</tm:If>
<div class='error'>${errorBean.errorMessage}</div>
<table class='book-list'>
<thead>
<tr>
<th>S.No.</th>
<th>Title</th>
<th>Code</th>
<th>Author</th>
<th>Category</th>

```

```

<th>Price</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tm:Book>
<tm:If expression='${evenRow}'>
<tr class='evenRow'>
</tm:If>
<tm:If expression='${oddRow}'>
<tr class='oddRow'>
</tm:If>
<td>${serialNumber}</td>
<td>${title}</td>
<td>${code}</td>
<td>${authorName}</td>
<td>${category}</td>
<td>${price}</td>
<td><a href='/${contextName}/EditBook.jsp?code=${code}'><img src='/${
contextName}/images/edit_icon.png' style='padding:5px'></a></td>
<td><a href='javascript:deleteBook(${code},"${title}")'><img src='/${
contextName}/images/delete_icon.png' style='padding:5px'></a></td>
</tr>
</tm:Book>
</tbody>
</table>
<form id='deleteBookForm' action='/${contextName}/DeleteBook.jsp' method='POST'>
<tm:FormId/>
<input type='hidden' name='code' id='code'>
<input type='hidden' name='title' id='title'>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

EditBook.jsp

```

<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/editBook' />

```

BookUpdateForm.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/js/BookUpdateForm.js'></script>

```

```

<h2>Edit Book </h2>
<div class='error'>${errorBean.errorMessage}</div>
<form action='/${contextName}/UpdateBook.jsp' onsubmit='return validateForm(this)' method='post'>
<tm:FormId/>
<input type='hidden' id='code' name='code' value='${bookBean.code}' >
<table border='0'>
<tr>
<td>Title</td>
<td>
<input type='text' name='title' id='title' maxlength='35' size='31' value='${bookBean.title}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='titleErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Author</td>
<td>
<select id='authorCode' name='authorCode'>
<option value='-1'>&lt;Select&gt;</option>
<tm:Author>
<tm:If expression='${bookBean.authorCode==code}'>
<option value='${code}' selected>${name}</option>
</tm:If>
<tm:If expression='${bookBean.authorCode!=code}'>
<option value='${code}'>${name}</option>
</tm:If>
</tm:Author>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='authorCodeErrorSection' class='error'>&nbsp;</span>
</td>
</tr>
<tr>
<td>Category</td>
<td>
<select id='category' name='category'>
<option value='none'>&lt;Select&gt;</option>
<option value='Science fiction' ${bookBean.category=="Science fiction"}?"selected":"">Science
fiction</option>
<option value='Satire' ${bookBean.category=="Satire"}?"selected":"">Satire</option>
<option value='Drama' ${bookBean.category=="Drama"}?"selected":"">Drama</option>

```

```

<option value='Action and Adventure' ${bookBean.category=="Action and
Adventure")?"selected":""}>Action and Adventure</option>
<option value='Mystery' ${bookBean.category=="Mystery")?"selected":""}>Mystery</option>
<option value='Horror' ${bookBean.category=="Horror")?"selected":""}>Horror</option>
</select>
</td>
</tr>
<tr>
<td></td><td>
<span id='categoryErrorSection' class='error'>&nbsp;&nbsp;&nbsp;</span>
</td>
</tr>
<tr>
<td>Price</td>
<td>
<input type='text' name='price' id='price' maxlength='5' size='6' style='text-align:right' value='${
bookBean.price}'>
</td>
</tr>
<tr>
<td></td><td>
<span id='priceErrorSection' class='error'>&nbsp;&nbsp;&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='submit'>Update</button>
</td>
</tr>
</table>
</form>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

UpdateBook.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>
<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/updateBook' />

```

DeleteBook.jsp

```

<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:FormResubmitted>

```



```

<jsp:forward page='/Notification.jsp' />
</tm:FormResubmitted>
<jsp:useBean id='bookBean' scope='request' class='com.thinking.machines.library.beans.BookBean' />
<jsp:setProperty name='bookBean' property='*' />
<jsp:forward page='/deleteBook' />

```

Notification.jsp

```

<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='notificationBean' scope='request'
class='com.thinking.machines.library.beans.NotificationBean' />
<h2>Notification</h2>
<div style='font-size:24pt'>
<b>${notificationBean.message}</b>
<br><br>
<form action='/${contextName}${notificationBean.actionURL}'>
<Button type='submit'>${notificationBean.actionText}</button>
</form>
</div>
<br>
<br>
<br>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

Note : We will be changing the Books List to incorporate feature to filter list of books by author in the classroom session.

--- CRUD Two Complete ---

Before going on to CRUD Three, we need to learn how to implement AJAX

We will be learning 3 examples

- a) using xmlhttpObject
- b) using JQuery
- c) building a wrapper like JQuery \$.ajax

I will roll all into one web application

create a web app folder named as ajax.com

create the following servlet

```

package thinking.machines;
import java.io.*;
import java.sql.*;
import javax.servlet.*;

```

```

import javax.servlet.http.*;
public class GetSquareAndFactorial extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
    try
    {
        PrintWriter pw=response.getWriter();
        int num=Integer.parseInt(request.getParameter("num"));
        int s=num*num;
        int e,f;
        e=f=1;
        while(e<=num)
        {
            f=f*e;
            e++;
        }
        pw.print(s+", "+f);
    } catch(Exception ex)
    {
        System.out.println(ex);
    }
}
}
}

```

Following is the servlet mapping that needs to be done in the deployment descriptor.

```

<servlet>
<servlet-name>GetSquareAndFactorial</servlet-name>
<servlet-class>thinking.machines.GetSquareAndFactorial</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetSquareAndFactorial</servlet-name>
<url-pattern>/GetSquareAndFactorial</url-pattern>
</servlet-mapping>
</web-app>

```

index.html

```

<html>
<head>
<title>
</title>
<script language='JavaScript'>
var xmlhttpObject=null;
function setupXMLHttpRequest()
{
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttpObject=new XMLHttpRequest();

```

```
    }
else
    { // code for IE6, IE5
    xmlhttpObject=new ActiveXObject("Microsoft.XMLHTTP");
    }
}

function computeSquareAndFactorial()
{
if(xmlHttpObject==null)
{
setupXMLHttpObject()
}
var num=document.getElementById("myForm").num.value;
xmlHttpObject.onreadystatechange=jawabAaGaya;
xmlHttpObject.open("GET","GetSquareAndFactorial?num="+num,true);
xmlHttpObject.send();
}

function jawabAaGaya()
{
if (xmlHttpObject.readyState==4 )
{
if(xmlHttpObject.status==200)
{
var res=xmlHttpObject.responseText.split(",");
document.getElementById("squareDivision").innerHTML="Square : "+res[0];
document.getElementById("factorialDivision").innerHTML="Factorial : "+res[1];
}
}
else
{
alert("Invalid request or server not ready");
}
}
}

</script>
</head>
<body>
<h2>AJAX Example One</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>
<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
```

```

</body>
<br/>
<div id='factorialDivision'>
</div>
<br>
<br>
<br>
<a href='jqueryajax.html'>AJAX Example Two</a><br>
<a href='tmjsajax.html'>AJAX Example Three</a><br>
</html>

```

Download jquery as discussed in the classroom session, and place the folder containing the jquery.js file in the ajax.com folder, in the following html, replace the jquery.js file name with the one that you have. I have highlighted it.

jqueryajax.html

```

<html>
<head>
<title>
</title>
<script src='jquery3.1.1/jquery-3.1.1.min.js'></script>
<script>
function computeSquareAndFactorial()
{
$.ajax({
  type: 'GET',
  url: 'GetSquareAndFactorial',
  data: {
    num : $("#num").val()
  },
  success: function(data) {
    var res=data.split(",");
    document.getElementById("squareDivision").innerHTML="Square : "+res[0];
    $("#factorialDivision").html("Factorial : "+res[1]);
  },
  error: function() {
    alert("Unable to send request, server not ready.....");
  }
});
}
</script>
</head>
<body>
<h2>AJAX Example</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>

```

```

<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
</body>
<br/>
<div id='factorialDivision'>
</div>
<br><br><br>
<a href='index.html'>AJAX Example One</a><br>
<a href='tmjsajax.html'>AJAX Example Three</a><br>

</html>

```

Now create a folder named as tmjs-1
in it create the following file tmjs-1.js as follows

tmjs-1.js

```

function setupXMLHttpRequest()
{
var xmlhttpObject=null;
if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttpObject=new XMLHttpRequest();
    }
else
    { // code for IE6, IE5
    xmlhttpObject=new ActiveXObject("Microsoft.XMLHTTP");
    }
return xmlhttpObject;
}
function jsonToQueryString(json)
{
return '?' +
Object.keys(json).map(function(key) {
return encodeURIComponent(key) + '=' +
encodeURIComponent(json[key]);
}).join('&');
}

function AjaxHandler()
{
var xmlhttpObject; // a private property
this.ajax=function(object)
{
if(xmlhttpObject==null) xmlhttpObject=setupXMLHttpRequest();
xmlhttpObject.onreadystatechange=function(){

```

```

if (xmlHttpRequest.readyState===4 )
{
if(xmlHttpRequest.status===200)
{
object.success(xmlHttpRequest.responseText);
}
else
{
if(object.error) object.error();
}
}
};
if(!object.type) object.type='GET';
if(!object.url) return;
if(object.data && object.type=='GET')
{
var qs=jsonToQueryString(object.data);
xmlHttpRequest.open(object.type,object.url+qs,true);
}
if(!object.data && object.type=='GET')
{
xmlHttpRequest.open(object.type,object.url,true);
}
// many more if conditions to handle other types of request
xmlHttpRequest.send();
};
}

```

```
var $=new AjaxHandler();
```

tmjsajax.html

```

<html>
<head>
<title>
</title>
<script src='tmjs-1/tmjs-1.js'></script>
<script>
function computeSquareAndFactorial()
{
$.ajax({
type: 'GET',
url: 'GetSquareAndFactorial',
data: {
num : document.getElementById("num").value
},
success: function(data) {
var res=data.split(",");

```

```
document.getElementById("squareDivision").innerHTML="Square : "+res[0];
document.getElementById("factorialDivision").innerHTML="Square : "+res[1];
},
error: function() {
    alert('Unable to send request, server not ready.....');
}
});
}
</script>
</head>
<body>
<h2>AJAX Example</h2>
<form id='myForm' name='myForm'>
Enter number
<input type='text' name='num' id='num'>
<input type='button' value='Compute' onclick='computeSquareAndFactorial()'>
</form>
<div id='squareDivision'>
</div>
</body>
<br/>
<div id='factorialDivision'>
</div>
</html>
```

CRUD Three

AJAX

Zero libraries

The learning will be incremental fashion

This is the first learning phase, where we will try to understand how to create UI for performing all the CRUD Operations in AJAX Style without using any third party library.

I am assuming that for this example, we know the resolution of the clients monitor and it won't change. As if we are writing the web application to be used in an organization for its process management.

Create a webapplication folder named as crudthree
in it create folders named as css, images, js, tm-js-1,WEB-INF

tm-js-1.css (in tm-js-1 folder)

```
.progressModalMask
{
position:absolute;
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/progressMask.png');
visibility:hidden;
opacity: 50%;
text-align:center;
}
.progressIndicator
{
position: absolute;
top:0;
bottom: 0;
left: 0;
right: 0;
margin: auto;
}
.forModal
{
visibility: hidden;
}
.modalMask
{
position:absolute;
left:0;
top:0;
width:100%;
height:100%;
background: url('../tm-js-1/mask.png');
visibility:hidden;
opacity: 50%;
}
.modal
```

```
{
border : 1px solid black;
height: 100px;
width: 100px;
position: absolute;
left:50%;
top: 10%;
z-index: 10000;
background: #FFFFFF;
border-radius: 0px;
box-shadow: 5px 5px 3px grey;
}
.modalHeader
{
background: #2E2E2E;
margin: 1px;
padding: 5px;
color: #FFFFFF;
font-size: 10pt;
height: 18px;
}
.modalClose
{
float: right;
cursor: pointer;
}
.modalTitle
{
float: left;
}
.modalFooter
{
background: #DFDFDF;
margin: 1px;
padding: 6px;
font-size: 8pt;
height: 10px;
text-align: center;
vertical-align: middle;
}
.modalContainer
{
margin: 1px;
padding: 10px;
font-size: 12pt;
overflow: auto;
```

```
}
```

tm-js-1.js (in tm-js-1 folder)

```
function getXMLHttpRequest()
{
var xmlhttpObject=null;
if (window.XMLHttpRequest)
  { // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttpObject=new XMLHttpRequest();
  }
else
  { // code for IE6, IE5
  xmlhttpObject=new ActiveXObject("Microsoft.XMLHTTP");
  }
return xmlhttpObject;
}
function jsonToQueryString(json)
{
return '?' +
Object.keys(json).map(function(key) {
return encodeURIComponent(key) + '=' +
encodeURIComponent(json[key]);
}).join('&');
}

function AjaxHandler()
{
var xmlhttpObject; // a private property
this.ajax=function(object)
{
if(xmlhttpObject===null) xmlhttpObject=getXMLHttpRequest();
xmlhttpObject.onreadystatechange=function(){
if (xmlhttpObject.readyState===4 )
{
if(xmlhttpObject.status===200)
{
object.success(xmlhttpObject.responseText);
}
else
{
if(object.error) object.error();
}
}
};
if(!object.type) object.type=object.type.toUpperCase();
if(!object.type) object.type='GET';
if(!object.url) return;
```

```

if(object.data && object.type=='GET')
{
var qs=jsonToQueryString(object.data);
xmlHttpRequest.open(object.type,object.url+qs,true);
xmlHttpRequest.send();
} else if(!object.data && object.type=='GET')
{
xmlHttpRequest.open(object.type,object.url,true);
xmlHttpRequest.send();
} else if(!object.data && object.type=='POST')
{
xmlHttpRequest.open(object.type,object.url, true);
xmlHttpRequest.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpRequest.send();
} else if(object.data && object.type=='POST')
{
xmlHttpRequest.open(object.type,object.url, true);
xmlHttpRequest.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlHttpRequest.send(jsonToQueryString(object.data).substring(1));
}
};
}
function showProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="visible";
}
function hideProgressModal()
{
var pmm=document.getElementsByClassName("progressModalMask")[0];
pmm.style.visibility="hidden";
}
function showModal(divisionToBeFixedInModal,width,height,title,footer)
{
document.body.scrollTop=document.documentElement.scrollTop=0;
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapper=document.getElementById("modalWrapper");
modalWrapper.style.width=width;
modalWrapper.style.height=height;
var widthValue=parseInt(modalWrapper.style.width);
var widthUOM=modalWrapper.style.width.substring(modalWrapper.style.width.length-2);
modalWrapper.style.marginLeft=((widthValue/2)*(-1))+"px";
var heightValue=parseInt(modalWrapper.style.height);
var heightUOM=modalWrapper.style.height.substring(modalWrapper.style.height.length-2);
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
modalWrapperContainer.style.height=(heightValue-18-18-38)+heightUOM; // 18 header, 18 footer, 38
shadow

```

```
var modalWrapperCloseIcon=document.getElementById("modalWrapperCloseIcon");
if(title)
{
var modalWrapperTitle=document.getElementById("modalWrapperTitle");
modalWrapperTitle.innerHTML=title;
}
if(footer)
{
var modalWrapperFooter=document.getElementById("modalWrapperFooter");
modalWrapperFooter.innerHTML=footer;
}
var divToFix=document.getElementById(divisionToBeFixedInModal);
divToFix.remove();
var clone=divToFix.cloneNode(true);
clone.style.visibility="visible";
modalWrapperContainer.append(clone);
modalWrapperContainer.fixedComponent=divToFix;
modalWrapperContainer.clonedComponent=clone;
modalWrapperMask.style.visibility="visible";
}
function disposeModal()
{
var modalWrapperMask=document.getElementById('modalWrapperMask');
var modalWrapperContainer=document.getElementById("modalWrapperContainer");
var divToUnfix=modalWrapperContainer.clonedComponent;
divToUnfix.remove();
modalWrapperMask.style.visibility="hidden";
document.body.appendChild(modalWrapperContainer.fixedComponent);
}
function setupModals()
{
var progressModalWrapperMask=document.createElement("div");
progressModalWrapperMask.className="progressModalMask";
var img=document.createElement("img");
img.className='progressIndicator';
img.src="tm-js-1/loading.gif"
progressModalWrapperMask.appendChild(img);
document.getElementsByTagName('body')[0].appendChild(progressModalWrapperMask);
progressModalWrapperMask.style.zIndex = "2000"
// Following is what we created and appended it to body of the page
/*<div id='progressModalWrapperMask' class='progressModalMask'>
<img src='images/loading.gif' class='progressIndicator'></img>
</div>*/
var modalWrapperMask=document.createElement("div");
modalWrapperMask.className="modalMask";
modalWrapperMask.id="modalWrapperMask";
var modalWrapper=document.createElement("div");
```

```

modalWrapper.id="modalWrapper";
modalWrapper.className="modal";
modalWrapperMask.appendChild(modalWrapper);
var modalWrapperHeader=document.createElement("div");
modalWrapperHeader.className="modalHeader";
modalWrapperHeader.id="modalWrapperHeader";
modalWrapper.appendChild(modalWrapperHeader);
var modalWrapperTitle=document.createElement("div");
modalWrapperTitle.className="modalTitle";
modalWrapperTitle.id="modalWrapperTitle";
modalWrapperHeader.appendChild(modalWrapperTitle);
var modalWrapperCloseIcon=document.createElement("div");
modalWrapperCloseIcon.className="modalClose";
modalWrapperCloseIcon.id="modalWrapperCloseIcon";
modalWrapperCloseIcon.onclick=disposeModal;
var closeIconImage=document.createElement("img");
closeIconImage.src='tm-js-1/close_modal_icon.png';
modalWrapperCloseIcon.appendChild(closeIconImage);
modalWrapperHeader.appendChild(modalWrapperCloseIcon);
var modalWrapperContainer=document.createElement("div");
modalWrapperContainer.className="modalContainer";
modalWrapperContainer.id="modalWrapperContainer";
modalWrapper.appendChild(modalWrapperContainer);
var modalWrapperFooter=document.createElement("div");
modalWrapperFooter.className="modalFooter";
modalWrapperFooter.id="modalWrapperFooter";
modalWrapper.appendChild(modalWrapperFooter);
document.getElementsByTagName('body')[0].appendChild(modalWrapperMask);
modalWrapperMask.style.zIndex = "1000"
// Following is what we created and appended it to body of the page
/*
<div id='modalWrapperMask' class='modalMask'>
<div id='modalWrapper' class='modal'>
<div id='modalWrapperHeader' class='modalHeader'>
<div id='modalWrapperTitle' class='modalTitle'></div>
<div id='modalWrapperCloseIcon' class='modalClose' onClick='disposeModal()'>
<img src='tm-js-1/close_modal_icon.png'></img>
</div>
</div>
<div id='modalWrapperContainer' class='modalContainer'>
</div>
<div id='modalWrapperFooter' class='modalFooter'>
</div>
</div>
</div>*/
}

```

```
// following will setup the function to be called on window load event  
window.onload=setupModals();
```

```
// following is the declaration of a global variable named as dollar ($)  
// The name dollar ($) is just to give you an insight of how JQuery must have done things  
// Note : My code is nowhere near to what JQuery creators have done, it is just an introduction
```

```
var $=new AjaxHandler();
```

Download tm-js-resources.zip from <http://tm-certificates.com/tm-js-resources.zip>. Unzip and place the images in tm-js-1 folder

Place delete_icon.png,edit_icon.png and logo.png in images folder (Refer crudone)

styles.css (place it in css folder)

```
body {  
background: #F5F5F5;  
}  
  
.home  
{  
padding: 5px;  
}  
  
.header {  
background: #DBDBDB;  
width:100%;  
}  
  
.header table  
{  
width: 100%;  
}  
  
.footer {  
background: #DBDBDB;  
width:100%;  
padding:10px;  
text-align: center;  
}  
  
.error {  
color: red;  
}
```

```
.evenRow
{
background: #DDDDDD;
}
```

```
.oddRow
{
background: #F5F5F5;
}
```

```
.author-list {
border: 1px solid black;
}
```

```
.author-list tr td {
border: 1px solid black;
text-align: right;
}
```

```
.author-list tr td + td {
border: 1px solid black;
text-align: left;
}
```

```
.author-list tr td + td + td {
border: 1px solid black;
text-align: right;
}
```

```
.book-list {
border: 1px solid black;
}
```

```
.book-list tr td {
border: 1px solid black;
text-align: right;
}
```

```
.book-list tr td + td {
border: 1px solid black;
text-align: left;
}
```

```
.book-list tr td + td + td {
border: 1px solid black;
text-align: right;
}
```



```

}

.book-list tr td + td + td + td {
border: 1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td {
border: 1px solid black;
text-align: left;
}

.book-list tr td + td + td + td + td + td {
border: 1px solid black;
text-align: right;
}

.book-list tr td + td + td + td + td + td + td {
border: 1px solid black;
text-align: center;
}

.book-list tr td + td + td + td + td + td + td + td {
border: 1px solid black;
text-align: center;
}

```

MasterPageTopSection.js (in js folder)

```

function initializeBody()
{
    var history_api = typeof history.pushState !== 'undefined';
    if(location.hash !== '#library') location.hash = '#library';
    if ( location.hash === '#library' ) {
        if ( history_api )
            history.pushState(null, "", '#librarian');
        else
            location.hash = '#librarian';
    }
    window.onhashchange = function() {
        if ( location.hash === '#library' ) {
            if ( history_api )
                history.pushState(null, "", '#librarian');
            else
                location.hash = '#librarian';
        }
    };
}

```

```
}

```

```
window.addEventListener("load",initializeBody);
```

MasterPageTopSection.jsp (in crudthree folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<tm:SetEnvironmentVariables/>
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Whatever Corporation</title>
<meta name="description" content="The Whatever Corporation">
<meta name="author" content="Thinking Machines">
<link rel='stylesheet' type='text/css' href='/${contextName}/css/styles.css' >
<link rel='stylesheet' type='text/css' href='/${contextName}/tm-js-1/tm-js-1.css' >
<script src='/${contextName}/js/MasterPageTopSection.js'></script>
</head>
<body>
<div class='header'>
<table>
<tr>
<td>
<img src='/${contextName}/images/logo.png'><br>
</td>
<td align='right' class='home'>
<tm:IfModule>
<a href='/${contextName}/'>Home</a>
</tm:IfModule>
</td>
</table>
</div>
<br>
```

MasterPageBottomSection.jsp (in crudthree folder)

```
<div class='footer'>
&copy; Thinking Machines 2017-2040
</div>
</body>
</html>
```

index.jsp (in crudthree folder)

```
<jsp:include page='/MasterPageTopSection.jsp' />
<ul>
<li><a href='/${contextName}/Authors.jsp'>Authors</a></li>
<li><a href='/${contextName}/Books.jsp'>Books</a></li>
</ul>
<jsp:include page='/MasterPageBottomSection.jsp' />
```

Authors.jsp (in crudthree folder)

```
<%@ taglib uri='/WEB-INF/tlds/CustomTags.tld' prefix='tm' %>
<jsp:include page='/MasterPageTopSection.jsp' />
<jsp:useBean id='authorBean' scope='request'
class='com.thinking.machines.library.beans.AuthorBean' />
<jsp:useBean id='errorBean' scope='request' class='com.thinking.machines.library.beans.ErrorBean' />
<script src='/${contextName}/tm-js-1/tm-js-1.js'></script>
<script>
function clearAuthors()
{
var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
while(authorsListBody.rows.length>0)
{
authorsListBody.deleteRow(0);
}
var row=authorsListBody.insertRow(authorsListBody.length);
var cell1=row.insertCell(0);
cell1.innerHTML="&nbsp;";
var cell2=row.insertCell(1);
cell2.innerHTML="&nbsp;";
var cell3=row.insertCell(2);
cell3.innerHTML="&nbsp;";
var cell4=row.insertCell(3);
cell4.innerHTML="&nbsp;";
}
function loadAuthors()
{
showProgressModal();
clearAuthors();
$.ajax({
type: 'GET',
url: 'getAuthors',
success: function(data) {
var spl=data.split(",")
var authorsListBody=document.getElementById("authorsList").getElementsByTagName('tbody')[0];
while(authorsListBody.rows.length>0) authorsListBody.deleteRow(0);
var i=0;
var serialNumber=0;
while(i<spl.length)
{
serialNumber++;
var row=authorsListBody.insertRow(authorsListBody.length);
var cell1=row.insertCell(0);
var cellText=document.createTextNode(serialNumber+".");
cell1.appendChild(cellText);
var cell2=row.insertCell(1);
cellText=document.createTextNode(spl[i+1]);
cell2.appendChild(cellText);

```

```
var cell3=row.insertCell(2);
cellText=document.createTextNode("Edit");
cell3.appendChild(cellText);
var cell4=row.insertCell(3);
cellText=document.createTextNode("Delete");
cell4.appendChild(cellText);
i+=2;
}
hideProgressModal();
},
error: function() {
hideProgressModal();
alert('Unable to send request, server not ready.....');
}
});
}
function addAuthor()
{
var authorAddForm=document.getElementById("authorAddForm");
var name=authorAddForm.name.value.trim();
var nameErrorSection=document.getElementById("authorAddFormNameErrorSection");
nameErrorSection.innerHTML="&nbsp;";
if(name.length==0)
{
nameErrorSection.innerHTML="Required";
authorAddForm.name.focus();
return;
}
var validCharacters="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ.";
var e=0;
while(e<name.length)
{
if(validCharacters.indexOf(name.charAt(e))==-1)
{
nameErrorSection.innerHTML="Invalid characters in name of author";
authorAddForm.name.focus();
return;
}
e++;
}
showProgressModal();
$.ajax({
type: 'POST',
url: 'addAuthor',
data: {
name : name
},
},
```

```
success: function(data) {
var spl=data.split(",");
if(spl[0]=== "true")
{
hideProgressModal();
disposeModal();
loadAuthors();
}
else
{
hideProgressModal();
var formErrorSection=document.getElementById("authorAddFormErrorSection");
formErrorSection.innerHTML=spl[1];
}
},
error: function() {
hideProgressModal();
alert('Unable to send request, server not ready.....');
}
});
}
window.addEventListener('load',loadAuthors);
</script>
<style>
.actionIcon
{
cursor:pointer;
}
.listContainer
{
text-align:right;
}
.authorsListView
{
border: 1px solid black;
border-spacing: 0px;
border-collapse: collapse;
width: 600px
font-size: 12pt;
}
.authorsListView thead
{
background-color: lightgray;
}
.authorsListView th
{
border: 1px solid black;
```

```

}
.authorsListView td
{
border: 1px solid black;
text-align:right;
width: 100px;
}
.authorsListView td+td
{
border: 1px solid black;
width:300px;
text-align:left;
}
.authorsListView td+td+td
{
border: 1px solid black;
text-align: center;
width:100px;
}
.authorsListView td+td+td+td
{
border: 1px solid black;
width:100px;
}
.authorsListLoader
{
text-align: center;
}
</style>
<h2>Authors</h2>
<table class='listContainer'>
<tr>
<td>
<img src='/${contextName}/images/add_icon.png' class='actionIcon'
onclick='showModal("authorAddFormContainer","400px","200px","Add author","")'></img>
</td>
</tr>
<tr>
<td>
<table id='authorsList' class='authorsListView'>
<thead>
<tr>
<th>S.No.</th>
<th>Author</th>
<th>Edit</th>
<th>Delete</th>
</tr>

```

```

</thead>
<tbody>
</tbody>
</table>
</td>
</tr>
</table>
<br>
<br>
<br>
<!-- Module Container For Modal Window Start -->
<div id='authorAddFormContainer' class='forModal'>
<span id='authorAddFormErrorSection' class='error'>&nbsp;&nbsp;&nbsp;</span>
<form id='authorAddForm'>
<table border='0'>
<tr>
<td>Name of author</td>
<td>
<input type='text' name='name' id='name' maxlength='35' size='31' value='${authorBean.name}'>
</td>
</tr>
<tr>
<td colspan='2' align='right'>
<span id='authorAddFormNameErrorSection' class='error'>&nbsp;&nbsp;&nbsp;</span>
</td>
</tr>
<tr>
<td colspan='2' align='center'>
<button type='button' onclick='addAuthor()'>Add</button>
</td>
</tr>
</table>
</form>
</div>
<jsp:include page='/MasterPageBottomSection.jsp' />

```

The WEB-INF folder should have tlds, lib and classes folder

the lib folder should contain mysql.jar

CustomTags.tld (in tlds folder)

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with

```

the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

-->

```
<!DOCTYPE taglib
  PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
  "http://java.sun.com/j2ee/dtd/web-jsptaglibrary_1_2.dtd">

<taglib>

  <tlib-version>1.0</tlib-version>
  <jsp-version>1.2</jsp-version>
  <short-name>CustomTags</short-name>
  <description>
    A tag library for this project
  </description>
  <tag>
    <name>SetEnvironmentVariables</name>
    <tag-class>com.thinking.machines.library.tags.EnvironmentVariablesTagHandler</tag-class>
    <description> Environment Variables Tag Handler </description>
  </tag>
  <tag>
    <name>FormId</name>
    <tag-class>com.thinking.machines.library.tags.FormIdTagHandler</tag-class>
    <description> Form Id. Generator Tag </description>
  </tag>
  <tag>
    <name>FormResubmitted</name>
    <tag-class>com.thinking.machines.library.tags.FormResubmittedTagHandler</tag-class>
    <description> Process body if form is resubmitted </description>
  </tag>
  <tag>
    <name>IfModule</name>
    <tag-class>com.thinking.machines.library.tags.IfModuleTagHandler</tag-class>
    <description> Tag to identify module page </description>
  </tag>
  <tag>
    <name>If</name>
    <tag-class>com.thinking.machines.library.tags.IfTagHandler</tag-class>
    <description> Tag to identify module page </description>
    <attribute>
```



```

    <name>expression</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
</tag>
<tag>
  <name>Author</name>
  <tag-class>com.thinking.machines.library.tags.AuthorTagHandler</tag-class>
  <tei-class>com.thinking.machines.library.tags.AuthorTEI</tei-class>
  <description> Author Tag Handler </description>
</tag>
<tag>
  <name>Book</name>
  <tag-class>com.thinking.machines.library.tags.BookTagHandler</tag-class>
  <tei-class>com.thinking.machines.library.tags.BookTEI</tei-class>
  <description> Book Tag Handler </description>
  <attribute>
    <name>authorCode</name>
    <required>false</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
</tag>
</taglib>

```

web.xml should contain following servlet mappings (in WEB-INF folder)

```

<servlet>
<servlet-name>AddAuthor</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.AddAuthor</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AddAuthor</servlet-name>
<url-pattern>/addAuthor</url-pattern>
</servlet-mapping>

<servlet>
<servlet-name>GetAuthors</servlet-name>
<servlet-class>com.thinking.machines.library.servlets.GetAuthors</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>GetAuthors</servlet-name>
<url-pattern>/getAuthors</url-pattern>
</servlet-mapping>

```

create the following structure in classes folder com\thinking\machines\library

in classes, create servlets, dl, beans and tags folder

copy all the classes from crudtwo for dl and beans folder

copy all the classes from crudtwo for tags folder and make one small change in the following class

```
package com.thinking.machines.library.tags;
import com.thinking.machines.library.dl.*;
import java.util.*;
import javax.servlet.jsp.tagext.*;
import javax.servlet.jsp.*;
public class EnvironmentVariablesTagHandler extends TagSupport
{
public int doStartTag()
{
pageContext.setAttribute("contextName","crudthree",PageContext.REQUEST_S
COPE);
return EVAL_BODY_INCLUDE;
}
public int doEndTag()
{
return EVAL_PAGE;
}
}
```

Following are the servlets to be created in the servlets folder

```
package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class GetAuthors extends HttpServlet
{
public void doGet(HttpServletRequest request,HttpServletResponse response)
{
try
{
Thread.sleep(1500); // remove this line after testing, this is just to Stimulate that the server is taking
time to fetch records
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
AuthorDAO authorDAO=new AuthorDAO();
try
{
LinkedList<AuthorInterface> authors;
authors=authorDAO.getAll();
boolean applyComma=false;
for(AuthorInterface author:authors)
```

```
{
if(applyComma) pw.print(",");
pw.print(author.getCode()+" "+author.getName());
applyComma=true;
}
} catch(DAOException daoException)
{
pw.print("false,No authors");
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
}

package com.thinking.machines.library.servlets;
import com.thinking.machines.library.dl.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class AddAuthor extends HttpServlet
{
public void doPost(HttpServletRequest request,HttpServletResponse response)
{
try
{
Thread.sleep(1500);
PrintWriter pw=response.getWriter();
response.setContentType("text/html");
String name=request.getParameter("name");
AuthorInterface author=new Author();
author.setName(name);
AuthorDAO authorDAO=new AuthorDAO();
try
{
authorDAO.add(author);
pw.print("true,Author added");
} catch(DAOException daoException)
{
pw.print("false,"+daoException.getMessage());
}
} catch(Exception exception)
{
System.out.println(exception); // remove after testing
}
}
}
```

}

Problems pending

The table has no scroll bars and it will become endless.

The newly added row doesn't get selected

On every addition operation, the table is being populated from zero

Assignment : Replace the words Delete and Edit with appropriate icons and add the necessary functionality against it as discussed in the classroom session.