**SPRING ASSIGNMENT 2**

**web.xml**

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

<web-app xmlns=*"http://java.sun.com/xml/ns/javaee"*

version=*"2.5"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"*>

<display-name>Spring Web Application</display-name>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/spring.xml</param-value>

</context-param>

</web-app>

**spring.xml**

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:tx=*"http://www.springframework.org/schema/tx"* xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<context:component-scan base-package=*"com.books.app.dao"* />

<context:component-scan base-package=*"com.books.app.service"* />

<bean id=*"bookAppService"* class=*"com.books.app.service.BookAppServiceImpl"*></bean>

<bean id=*"bookAppDAO"* class=*"com.books.app.dao.BookAppDAOImpl"*></bean>

</beans>

**dispatcher-servlet.xml**

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*" http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<mvc:annotation-driven />

<context:component-scan base-package=*"com.books.app"* />

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/pages/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

<mvc:resources location=*"/resources/"* mapping=*"/resources/\*\*"*></mvc:resources>

</beans>

**pom.xml**

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>books.app</groupId>

<artifactId>springmvc</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>springmvc Maven Webapp</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<spring.version>4.0.0.RELEASE</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<resources>

<resource>

<directory>src/main/resources</directory>

</resource>

</resources>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>${java-version}</source>

<target>${java-version}</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<configuration>

<port>9090</port>

<path>/</path>

</configuration>

</plugin>

</plugins>

</build>

</project>

**index.jsp**

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"* %>

<html>

<body>

<h2>Book Shop Application</h2>

<a href=*"view"*>View all books</a><br>

<a href=*"addBook"*>Add Book</a><br>

<a href=*"searchBookForm"*>Search Book</a><br>

<a href=*"searchSubjectForm"*>Search Subject</a><br>

<a href=*"deleteBook"*>Delete Book</a><br>

<a href=*"deleteSubject"*>Delete Subject</a><br>

<a href=*"exit"*>Exit</a><br>

</body>

</html>

**addBook.jsp**

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<html>

<body>

<h2>Add Book</h2>

<form:form method=*"POST"* action=*"saveBook"*>

Enter Book ID : <form:input type=*"text"* path=*"bookId"*></form:input>

<br>

Enter Book title : <form:input type=*"text"* path=*"title"*></form:input>

<br>

Enter Book price : <form:input type=*"text"* path=*"price"*></form:input>

<br>

Enter Book volume : <form:input type=*"text"* path=*"volume"*></form:input>

<br>

Enter Book published Date : <form:input type=*"text"* path=*"publistDt"*></form:input>

<br>

Enter Subject title : <form:input type=*"text"* path=*"subtitle"*></form:input>

<br>

<input type=*"submit"* value=*"Add Book"*></input>

</form:form>

</body>

</html>

**deleteBook.jsp**

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<html>

<body>

<h2>Delete Book</h2>

<form:form method=*"POST"* action=*"delBook"*>

Enter Book title : <form:input type=*"text"* path=*"title"*></form:input>

<input type=*"submit"* value=*"Delete Book"*></input>

</form:form>

<br><br>

<a href=*""*>Go back to menu</a><br>

</body>

</html>

**deleteSubject.jsp**

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<html>

<body>

<h2>Delete Subject</h2>

<form:form method=*"POST"* action=*"delSubject"*>

Enter Subject title : <form:input type=*"text"* path=*"subtitle"*></form:input>

<input type=*"submit"* value=*"Delete Subject"*></input>

</form:form>

<br><br>

<a href=*""*>Go back to menu</a><br>

</body>

</html>

**searchBookForm.jsp**

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<html>

<body>

<h2>Search Book</h2>

<form:form method=*"POST"* action=*"searchBook"*>

Enter Book title : <form:input type=*"text"* path=*"title"*></form:input>

<input type=*"submit"* value=*"Search Book"*></input>

</form:form>

<br><br>

<a href=*""*>Go back to menu</a><br>

</body>

</html>

**searchBook.jsp**

<html>

<body>

<h2>Book search results </h2>

<table border=*"1"* width=*"70%"* cellpadding=*"2"*>

<tr>

<th>Subject title</th>

<th>Book id</th>

<th>Book title</th>

<th>Book price</th>

<th>Book volume</th>

</tr>

<tr>

<td>${subtitle}</td>

<td>${bookId}</td>

<td>${title}</td>

<td>${price}</td>

<td>${volume}</td>

</tr>

</table>

<br><br>

<a href=*""*>Go back to menu</a><br>

</body>

</html>

**searchSubjectForm.jsp**

<%@ taglib prefix=*"form"* uri=*"http://www.springframework.org/tags/form"*%>

<html>

<body>

<h2>Search Subject</h2>

<form:form method=*"POST"* action=*"searchSubject"*>

Enter Subject title : <form:input type=*"text"* path=*"subtitle"*></form:input>

<input type=*"submit"* value=*"Search Subject"*></input>

</form:form>

<br><br>

<a href=*""*>Go back to menu</a><br>

</body>

</html>

**searchSubject.jsp**

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"* %>

<a href=*""*>Go back to menu</a><br>

<h2>Subject search results</h2>

<table border=*"1"* width=*"70%"* cellpadding=*"2"*>

<tr>

<th>Subject title</th>

<th>Book id</th>

<th>Book title</th>

<th>Book price</th>

<th>Book volume</th>

</tr>

<c:forEach var=*"book"* items=*"*${booksbysubject}*"*>

<tr>

<td>${book.subtitle}</td>

<td>${book.bookId}</td>

<td>${book.title}</td>

<td>${book.price}</td>

<td>${book.volume}</td>

</tr>

</c:forEach>

</table>

**view.jsp**

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/core"* prefix=*"c"*%>

<%@ taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"* %>

<a href=*""*>Go back to menu</a><br>

<h2>Books List</h2>

<table border=*"1"* width=*"70%"* cellpadding=*"2"*>

<tr>

<th>Subject title</th>

<th>Book id</th>

<th>Book title</th>

<th>Book price</th>

<th>Book volume</th>

</tr>

<c:forEach var=*"book"* items=*"*${booklist}*"*>

<tr>

<td>${book.subtitle}</td>

<td>${book.bookId}</td>

<td>${book.title}</td>

<td>${book.price}</td>

<td>${book.volume}</td>

</tr>

</c:forEach>

</table>

**exit.jsp**

<html>

<h2>Thanks for Shopping !!</h2>

</html>

**BookController.java**

**package** com.books.app.controller;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Set;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

**import** org.springframework.validation.BindingResult;

**import** org.springframework.web.bind.annotation.ModelAttribute;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**import** com.books.app.service.IBookAppService;

@Controller

**public** **class** BookController {

@Autowired

IBookAppService bookAppService;

@RequestMapping("/")

**public** String showHome() {

**return** "index";

}

@RequestMapping("/exit")

**public** String exitApp() {

**return** "exit";

}

@RequestMapping("/view")

**public** ModelAndView showAllBooks() {

System.***out***.println("View all method");

List<Book> booklist = **null**;

**try** {

booklist = bookAppService.viewAllBooks();

} **catch** (BookAppException e) {

e.printStackTrace();

}

**return** **new** ModelAndView("view", "booklist", booklist);

}

@RequestMapping("/addBook")

**public** ModelAndView showBookForm() {

**return** **new** ModelAndView("addBook", "command", **new** Book());

}

@RequestMapping(value = "/saveBook", method = RequestMethod.***POST***)

**public** ModelAndView saveBook(@ModelAttribute("book") Book book, BindingResult result, Model model) {

System.***out***.println("Add book method");

**try** {

System.***out***.println("Add book method----- Inside try/catch");

bookAppService.add(book, book.getSubtitle());

} **catch** (BookAppException e) {

e.printStackTrace();

}

System.***out***.println("Book title " + book.getTitle());

**return** **new** ModelAndView("redirect:/view");

}

@RequestMapping("/searchBookForm")

**public** ModelAndView searchBookForm() {

**return** **new** ModelAndView("searchBookForm", "command", **new** Book());

}

@RequestMapping("/searchBook")

**public** ModelAndView searchBook(@ModelAttribute("book") Book book, BindingResult result, Model model) {

**try** {

Book sbook = bookAppService.search(book.getTitle());

model.addAttribute("subtitle", sbook.getSubtitle());

model.addAttribute("bookId", sbook.getBookId());

model.addAttribute("title", sbook.getTitle());

model.addAttribute("price", sbook.getPrice());

model.addAttribute("volume", sbook.getVolume());

} **catch** (BookAppException e) {

e.printStackTrace();

}

**return** **new** ModelAndView("searchBook");

}

@RequestMapping("/searchSubjectForm")

**public** ModelAndView searchSubjectForm() {

**return** **new** ModelAndView("searchSubjectForm", "command", **new** Book());

}

@RequestMapping("/searchSubject")

**public** ModelAndView searchSubject(@ModelAttribute("book") Book book, BindingResult result, Model model) {

List<Book> booksbysubject = **new** ArrayList<Book>();

**try** {

Subject subj = bookAppService.searchSubject(book.getSubtitle());

Set<Book> books = subj.getReferences();

**for** (Book sbook : books) {

booksbysubject.add(sbook);

}

} **catch** (BookAppException e) {

e.printStackTrace();

}

**return** **new** ModelAndView("searchSubject", "booksbysubject", booksbysubject);

}

@RequestMapping("/deleteBook")

**public** ModelAndView deleteBookForm() {

**return** **new** ModelAndView("deleteBook", "command", **new** Book());

}

@RequestMapping("/delBook")

**public** ModelAndView delBook(@ModelAttribute("book") Book book, BindingResult result, Model model) {

List<Book> booklist = **null**;

**try** {

bookAppService.delete(book.getTitle());

booklist = bookAppService.viewAllBooks();

} **catch** (BookAppException e) {

e.printStackTrace();

}

**return** **new** ModelAndView("view", "booklist", booklist);

}

@RequestMapping("/deleteSubject")

**public** ModelAndView deleteSubjectForm() {

**return** **new** ModelAndView("deleteSubject", "command", **new** Book());

}

@RequestMapping("/delSubject")

**public** ModelAndView delSubject(@ModelAttribute("book") Book book, BindingResult result, Model model) {

List<Book> booklist = **null**;

**try** {

bookAppService.deleteSubject(book.getSubtitle());

booklist = bookAppService.viewAllBooks();

} **catch** (BookAppException e) {

e.printStackTrace();

}

**return** **new** ModelAndView("view", "booklist", booklist);

}

}

**IBookAppService.java**

**package** com.books.app.service;

**import** java.util.List;

**import** java.util.Map;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**public** **interface** IBookAppService {

String add(Book book, String subtitle) **throws** BookAppException;

String addSubject(Subject subject) **throws** BookAppException;

**boolean** delete(String booktitle) **throws** BookAppException;

**boolean** deleteSubject(String subtitle) **throws** BookAppException;

Book search(String booktitle) **throws** BookAppException;

Subject searchSubject(String subtitle) **throws** BookAppException;

List<Book> viewAllBooks() **throws** BookAppException;

Map<String, Subject> view() **throws** BookAppException;

}

**BookAppServiceImpl.java**

**package** com.books.app.service;

**import** java.util.List;

**import** java.util.Map;

**import** com.books.app.dao.BookAppDAOImpl;

**import** com.books.app.dao.IBookAppDAO;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

@Service

**public** **class** BookAppServiceImpl **implements** IBookAppService {

@Autowired

**private** IBookAppDAO bookAppDao;

@Override

**public** String add(Book book, String subtitle) **throws** BookAppException {

**return** bookAppDao.add(book, subtitle);

}

@Override

**public** String addSubject(Subject subject) **throws** BookAppException {

**return** bookAppDao.addSubject(subject);

}

@Override

**public** **boolean** delete(String booktitle) **throws** BookAppException {

**return** bookAppDao.delete(booktitle);

}

@Override

**public** **boolean** deleteSubject(String subtitle) **throws** BookAppException {

**return** bookAppDao.deleteSubject(subtitle);

}

@Override

**public** Book search(String booktitle) **throws** BookAppException {

**return** bookAppDao.search(booktitle);

}

@Override

**public** Subject searchSubject(String subtitle) **throws** BookAppException {

**return** bookAppDao.searchSubject(subtitle);

}

@Override

**public** List<Book> viewAllBooks() **throws** BookAppException {

**return** bookAppDao.viewAll();

}

@Override

**public** Map<String, Subject> view() **throws** BookAppException{

**return** bookAppDao.view();

}

}

**IBookAppDAO.java**

**package** com.books.app.dao;

**import** java.util.List;

**import** java.util.Map;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**public** **interface** IBookAppDAO {

String add(Book book, String subtitle) **throws** BookAppException;

String addSubject(Subject subject) **throws** BookAppException;

**boolean** delete(String booktitle) **throws** BookAppException;

**boolean** deleteSubject(String subtitle) **throws** BookAppException;

Book search(String booktitle) **throws** BookAppException;

Subject searchSubject(String subtitle) **throws** BookAppException;

List<Book> viewAll() **throws** BookAppException;

Map<String, Subject> view() **throws** BookAppException;

}

**BookAppDAOImpl.java**

**package** com.books.app.dao;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.io.ObjectInputStream;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.Set;

**import** java.util.TreeMap;

**import** com.books.app.exception.BookAppException;

**import** com.books.app.model.Book;

**import** com.books.app.model.Subject;

**public** **class** BookAppDAOImpl **implements** IBookAppDAO {

**public** **static** **final** String ***DATA\_STORE\_FILE\_NAME*** = "C:\\Users\\Admin\\Desktop\\bookapp.dat";

**private** Map<String, Subject> subjects;

**public** BookAppDAOImpl() **throws** BookAppException {

File file = **new** File(***DATA\_STORE\_FILE\_NAME***);

**if** (!file.exists()) {

subjects = **new** TreeMap<String, Subject>();

System.***out***.println("BookApp map is newly instantiated");

} **else** {

**try** (ObjectInputStream fin = **new** ObjectInputStream(**new** FileInputStream(***DATA\_STORE\_FILE\_NAME***))) {

Object obj = fin.readObject();

**if** (obj **instanceof** Map) {

subjects = (TreeMap<String, Subject>) obj;

System.***out***.println("books successfully loaded");

} **else** {

**throw** **new** BookAppException("Not a valid BookApp");

}

} **catch** (IOException | ClassNotFoundException exp) {

System.***err***.println(exp);

**throw** **new** BookAppException("Loading Data Failed");

}

}

}

@Override

**public** String add(Book book, String subtitle) **throws** BookAppException {

String bookTitle = **null**;

Set<Book> books = **null**;

**if** (book != **null**) {

bookTitle = book.getTitle();

**if** (subjects == **null** || subjects.isEmpty() || !subjects.containsKey(subtitle)) {

Subject subj = **new** Subject();

subj.setSubtitle(subtitle);

addSubject(subj);

}

**if** (subjects != **null** && !subjects.isEmpty()) {

**if** (subjects.containsKey(subtitle)) {

**if** (subjects.get(subtitle).getReferences() != **null**) {

books = subjects.get(subtitle).getReferences();

books.add(book);

} **else** {

books = **new** HashSet<Book>();

books.add(book);

subjects.get(subtitle).setReferences(books);

}

subjects.get(subtitle).setReferences(books);

}

}

}

**return** bookTitle;

}

@Override

**public** String addSubject(Subject subject) **throws** BookAppException {

String subjectTitle = **null**;

**if** (subject != **null**) {

subjectTitle = subject.getSubtitle();

subjects.put(subjectTitle, subject);

}

**return** subjectTitle;

}

@Override

**public** **boolean** delete(String booktitle) **throws** BookAppException {

**if** (subjects != **null**) {

**for** (Map.Entry<String, Subject> entry : subjects.entrySet()) {

Subject subj = entry.getValue();

**if** (subj != **null**) {

Set<Book> books = subj.getReferences();

**for** (Book book : books) {

**if** (booktitle.equalsIgnoreCase(book.getTitle())) {

books.remove(book);

**return** **true**;

}

}

}

}

}

**return** **false**;

}

@Override

**public** **boolean** deleteSubject(String subtitle) **throws** BookAppException {

**if** (subjects != **null**) {

**if** (subjects.containsKey(subtitle)) {

subjects.remove(subtitle);

**return** **true**;

}

}

**return** **false**;

}

@Override

**public** Book search(String booktitle) **throws** BookAppException {

**if** (subjects != **null**) {

**for** (Map.Entry<String, Subject> entry : subjects.entrySet()) {

Subject subj = entry.getValue();

**if** (subj != **null**) {

Set<Book> books = subj.getReferences();

**for** (Book book : books) {

**if** (booktitle.equalsIgnoreCase(book.getTitle())) {

**return** book;

}

}

}

}

}

**return** **null**;

}

@Override

**public** Subject searchSubject(String subtitle) **throws** BookAppException {

**if** (subjects != **null**) {

**if** (subjects.containsKey(subtitle)) {

**return** subjects.get(subtitle);

}

}

**return** **null**;

}

@Override

**public** List<Book> viewAll() **throws** BookAppException {

List<Book> bookList = **new** ArrayList<Book>();

**if** (subjects != **null** && !subjects.isEmpty()) {

**for** (Map.Entry<String, Subject> entry : subjects.entrySet()) {

Subject subj = entry.getValue();

**if** (subj != **null**) {

Set<Book> books = subj.getReferences();

**for** (Book book : books) {

bookList.add(book);

}

}

}

}

**return** bookList;

}

@Override

**public** Map<String, Subject> view() **throws** BookAppException {

**if** (subjects != **null** && !subjects.isEmpty()) {

**return** subjects;

}

**return** **null**;

}

}

**BookAppException.java**

**package** com.books.app.exception;

@SuppressWarnings("serial")

**public** **class** BookAppException **extends** Exception {

**public** BookAppException(String errMsg) {

**super**(errMsg);

}

}

**Subject.java**

**package** com.books.app.model;

**import** java.io.Serializable;

**import** java.util.Set;

**public** **class** Subject **implements** Serializable {

**private** **long** subjectId;

**private** String subtitle;

**private** **int** durationInHours;

**private** Set<Book> references;

**public** long getSubjectId() {

**return** subjectId;

}

**public** void setSubjectId(long subjectId) {

**this**.subjectId = subjectId;

}

**public** String getSubtitle() {

**return** subtitle;

}

**public** void setSubtitle(String subtitle) {

**this**.subtitle = subtitle;

}

**public** int getDurationInHours() {

**return** durationInHours;

}

**public** void setDurationInHours(int durationInHours) {

**this**.durationInHours = durationInHours;

}

**public** Set<Book> getReferences() {

**return** references;

}

**public** void setReferences(Set<Book> references) {

**this**.references = references;

}

}

**Book.java**

**package** com.books.app.model;

**import** java.io.Serializable;

**import** java.time.LocalDate;

**public** **class** Book **implements** Serializable {

**private** **long** **bookId**;

**private** String title;

**private** **double** price;

**private** **int** volume;

**private** LocalDate publistDt;

**private** String subtitle;

**public** String getSubtitle() {

**return** subtitle;

}

**public** **void** setSubtitle(String subtitle) {

**this**.subtitle = subtitle;

}

**public** **long** getBookId() {

**return** bookId;

}

**public** **void** setBookId(**long** bookId) {

**this**.bookId = bookId;

}

**public** String getTitle() {

**return** title;

}

**public** **void** setTitle(String title) {

**this**.title = title;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** **int** getVolume() {

**return** volume;

}

**public** **void** setVolume(**int** volume) {

**this**.volume = volume;

}

**public** LocalDate getPublistDt() {

**return** publistDt;

}

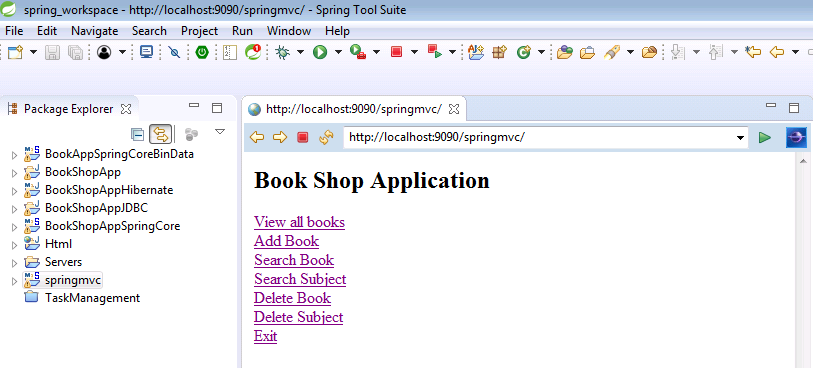
**public** **void** setPublistDt(LocalDate publistDt) {

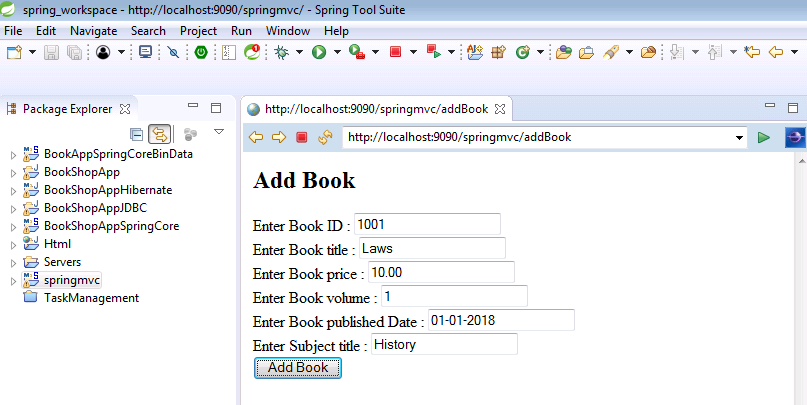
**this**.publistDt = publistDt;

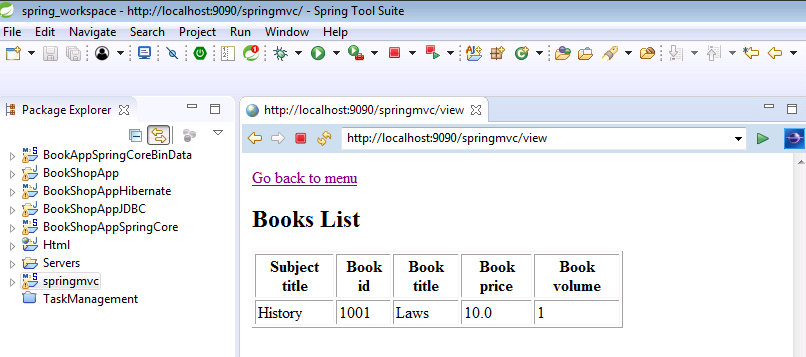
}

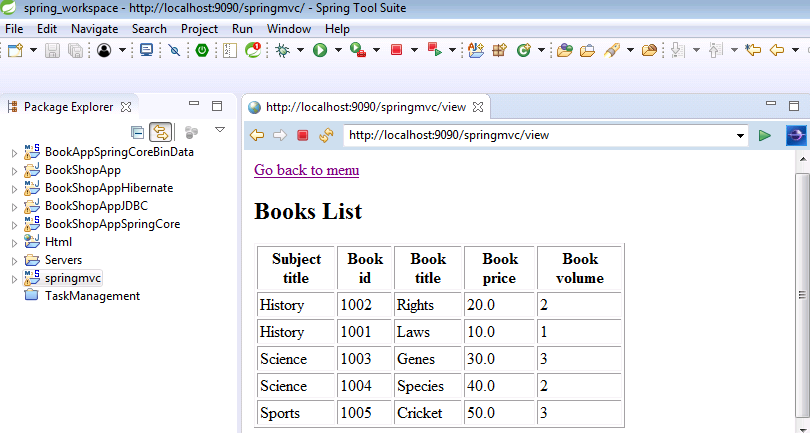
}

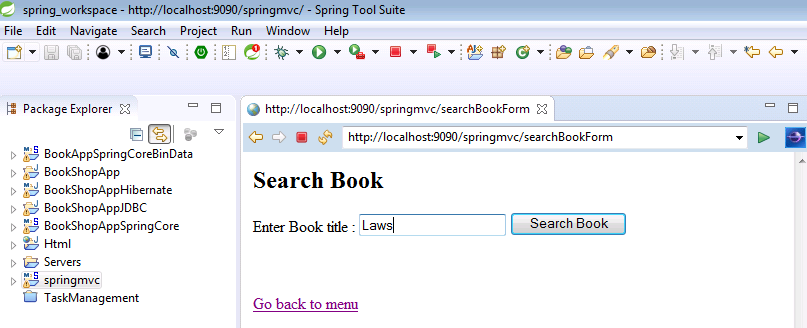
**RESULTS :**

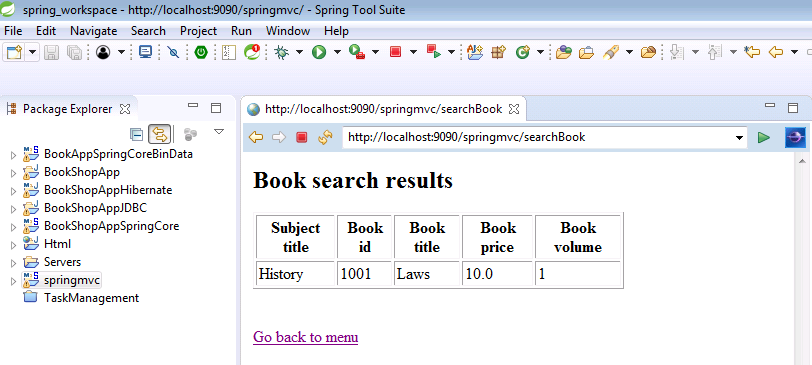
****

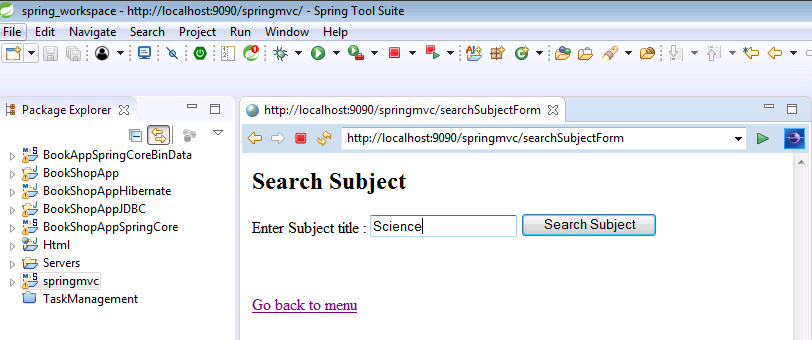
****

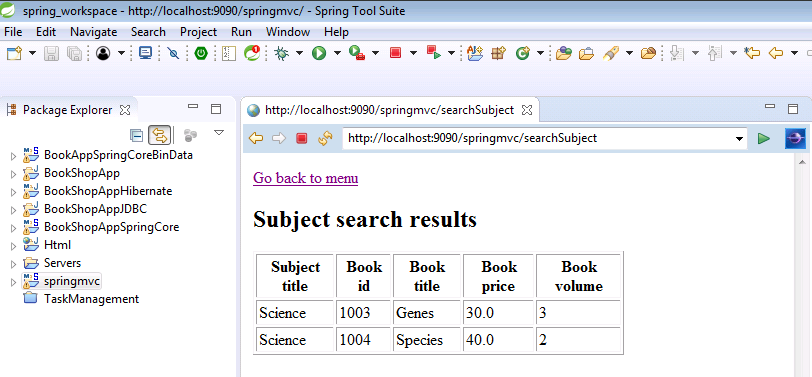
****

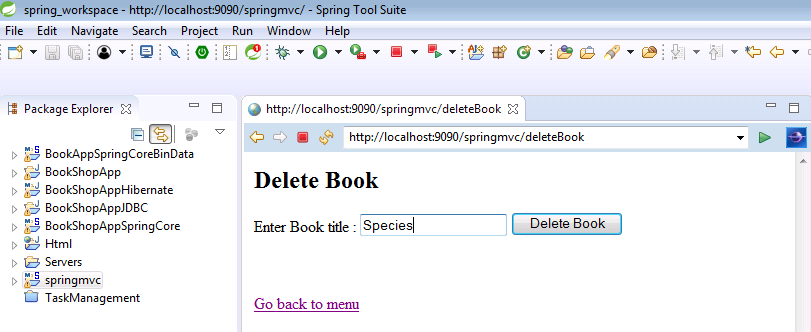
****

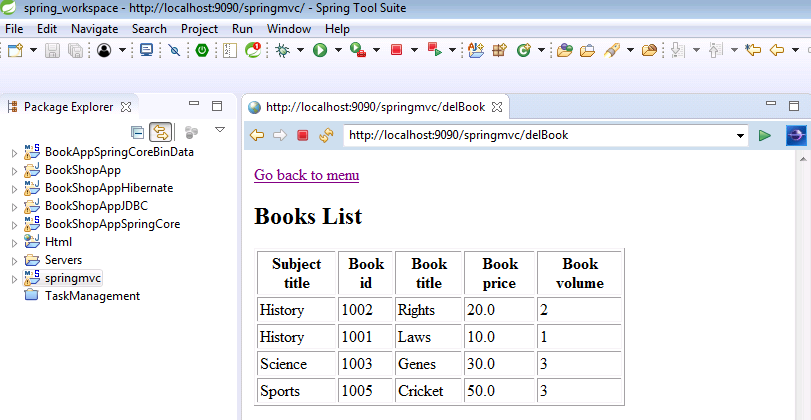
****

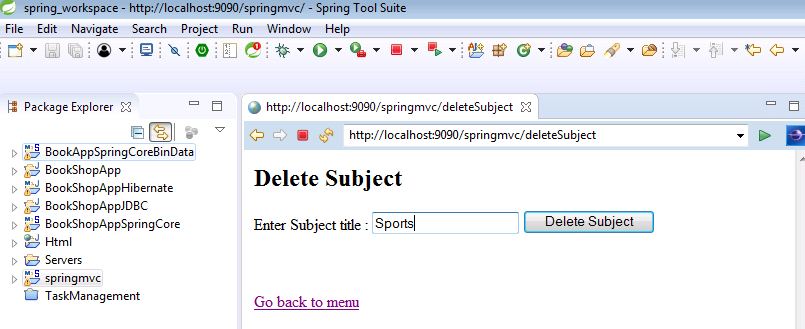
****

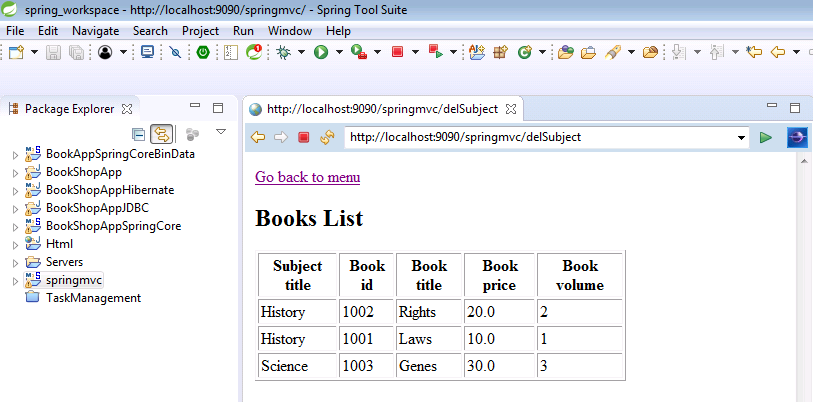
****

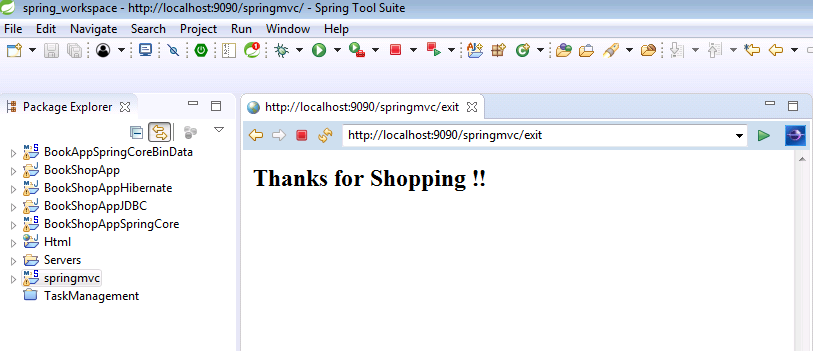
****

****

****

****

****

****