**SPRING ASSIGNMENT 3**

**web.xml**

<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 MVC Hibernate 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>

**persistence.xml**

<persistence xmlns=*"http://xmlns.jcp.org/xml/ns/persistence"*

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

xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence\_2\_1.xsd"*

version=*"2.1"*>

<persistence-unit name=*"PERSISTENCE"*>

<provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

<properties>

<property name=*"javax.persistence.jdbc.driver"* value=*"com.mysql.jdbc.Driver"* />

<property name=*"javax.persistence.jdbc.url"* value=*"jdbc:mysql://localhost:3306/bookshop"* />

<property name=*"javax.persistence.jdbc.user"* value=*"root"* />

<property name=*"javax.persistence.jdbc.password"* value=*"root"* />

<property name=*"hibernate.show\_sql"* value=*"true"* />

<property name=*"hibernate.hbm2ddl.auto"* value=*"update"* />

</properties>

</persistence-unit>

</persistence>

**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>springmvchibernate</artifactId>

<packaging>war</packaging>

<version>0.0.1-SNAPSHOT</version>

<name>springmvchibernate 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>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.44</version>

</dependency> <!-- Hibernate 5.2.6 Final -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.2.6.Final</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>springmvchibernate</finalName>

</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=*"subject.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 Id : <form:input type=*"text"* path=*"bookId"*></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 Id : <form:input type=*"text"* path=*"bookId"*></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>${stitle}</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.subject.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.subject.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.view();

} **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) {

**try** {

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

} **catch** (BookAppException e) {

e.printStackTrace();

}

**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.getBookId());

model.addAttribute("stitle", sbook.getSubject().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** Subject());

}

@RequestMapping("/searchSubject")

**public** ModelAndView searchSubject(@ModelAttribute("subject") Subject subject, BindingResult result, Model model) {

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

**try** {

Subject subj = bookAppService.searchSubject(subject.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.getBookId());

booklist = bookAppService.view();

} **catch** (BookAppException e) {

e.printStackTrace();

}

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

}

@RequestMapping("/deleteSubject")

**public** ModelAndView deleteSubjectForm() {

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

}

@RequestMapping("/delSubject")

**public** ModelAndView delSubject(@ModelAttribute("subject") Subject subject, BindingResult result, Model model) {

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

**try** {

bookAppService.deleteSubject(subject.getSubtitle());

booklist = bookAppService.view();

} **catch** (BookAppException e) {

e.printStackTrace();

}

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

}

}

**IBookAppService.java**

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

**import** java.util.List;

**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(**long** bookid) **throws** BookAppException;

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

Book search(**long** bookid) **throws** BookAppException;

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

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

}

**BookAppServiceImpl.java**

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

**import** java.util.List;

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

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

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

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

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

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

**private** IBookAppDAO bookAppDao;

**public** BookAppServiceImpl() **throws** BookAppException {

bookAppDao = **new** BookAppHibernateDAOImpl();

}

**public** IBookAppDAO getBookAppDAO() {

**return** 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(**long** bookid) **throws** BookAppException {

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

}

@Override

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

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

}

@Override

**public** Book search(**long** bookid) **throws** BookAppException {

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

}

@Override

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

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

}

@Override

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

**return** bookAppDao.view();

}

}

**IBookAppDAO.java**

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

**import** java.util.List;

**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(**long** bookid) **throws** BookAppException;

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

Book search(**long** bookid) **throws** BookAppException;

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

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

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

}

**BookAppHibernateDAOImpl.java**

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

**import** java.util.List;

**import** javax.persistence.EntityManager;

**import** javax.persistence.EntityTransaction;

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

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

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

**import** com.books.app.util.JPAUtil;

**public** **class** BookAppHibernateDAOImpl **implements** IBookAppDAO {

**public** BookAppHibernateDAOImpl() **throws** BookAppException {

}

@Override

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

String bookTitle = **null**;

Subject subject = **null**;

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

List<Subject> subjects = viewAll();

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

**for** (Subject iterSubject : subjects) {

**if** (iterSubject != **null** && iterSubject.getSubtitle().equalsIgnoreCase(subtitle)) {

subject = iterSubject;

}

}

}

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

subject = **new** Subject();

subject.setSubtitle(subtitle);

addSubject(subject);

}

book.setSubject(subject);

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

EntityTransaction trnx = em.getTransaction();

trnx.begin();

em.persist(book);

trnx.commit();

bookTitle = book.getTitle();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book is not inserted");

}

}

**return** bookTitle;

}

@Override

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

String subjectTitle = **null**;

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

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

EntityTransaction trnx = em.getTransaction();

trnx.begin();

em.persist(subject);

trnx.commit();

subjectTitle = subject.getSubtitle();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject is not inserted");

}

}

**return** subjectTitle;

}

@Override

**public** **boolean** delete(**long** bookid) **throws** BookAppException {

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

EntityTransaction trnx = em.getTransaction();

trnx.begin();

Book book = em.find(Book.**class**, bookid);

em.remove(book);

trnx.commit();

**return** **true**;

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book is not deleted");

}

}

@Override

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

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

EntityTransaction trnx = em.getTransaction();

trnx.begin();

Subject subj = em.find(Subject.**class**, subtitle);

em.remove(subj);

trnx.commit();

**return** **true**;

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject is not deleted");

}

}

@Override

**public** Book search(**long** bookid) **throws** BookAppException {

Book book = **null**;

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

book = em.find(Book.**class**, bookid);

} **catch** (Exception exp) {

**throw** **new** BookAppException("Book not found");

}

**return** book;

}

@Override

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

Subject subj = **null**;

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

subj = em.find(Subject.**class**, subtitle);

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subject not found");

}

**return** subj;

}

@Override

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

List<Subject> subjList = **null**;

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

subjList = em.createQuery(IQueryMapper.***GET\_ALL\_SUBJECTS\_QRY***).getResultList();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Subjects are not retrieved");

}

**return** subjList;

}

@Override

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

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

**try** {

EntityManager em = JPAUtil.*getEntityManagerFactory*().createEntityManager();

bookList = em.createQuery(IQueryMapper.***GET\_ALL\_BOOKS\_QRY***).getResultList();

} **catch** (Exception exp) {

**throw** **new** BookAppException("Books are not retrieved");

}

**return** bookList;

}

}

**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.util.Set;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.Id;

**import** javax.persistence.OneToMany;

**import** javax.persistence.Table;

@Entity

@Table(name="subjects")

**public** **class** Subject {

@Id

@Column(name="subtitle")

**private** String subtitle;

@Column(name="subjectid")

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

@Column(name="durationInHours")

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

@OneToMany(fetch = FetchType.***EAGER***,mappedBy="subject",orphanRemoval=**true**)

**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.time.LocalDate;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.Id;

**import** javax.persistence.JoinColumn;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.Table;

**import** org.springframework.format.annotation.DateTimeFormat;

@Entity

@Table(name="books")

**public** **class** Book {

@Id

@Column(name="bookid")

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

@Column(name="title")

**private** String title;

@Column(name="price")

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

@Column(name="vol")

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

@Column(name="pdate")

@DateTimeFormat(pattern ="dd-MM-yyyy")

**private** LocalDate publistDt;

@ManyToOne(fetch=FetchType.***EAGER***)

@JoinColumn(name="stitle" , nullable=**false**)

**private** Subject subject;

**public** Subject getSubject() {

**return** subject;

}

**public** **void** setSubject(Subject subject) {

**this**.subject = subject;

}

**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;

}

}

**JPAUtil.java**

**package** com.books.app.util;

**import** javax.persistence.EntityManagerFactory;

**import** javax.persistence.Persistence;

**public** **class** JPAUtil {

**private** **static** **final** String ***PERSISTENCE\_UNIT\_NAME*** = "PERSISTENCE";

**private** **static** EntityManagerFactory *factory*;

**public** **static** EntityManagerFactory getEntityManagerFactory() {

**if** (*factory* == **null**) {

*factory* = Persistence.*createEntityManagerFactory*(***PERSISTENCE\_UNIT\_NAME***);

}

**return** *factory*;

}

**public** **static** **void** shutdown() {

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

*factory*.close();

}

}

}

**IQueryMapper.java**

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

**public** **interface** IQueryMapper {

**public** **static** **final** String ***GET\_ALL\_BOOKS\_QRY*** =

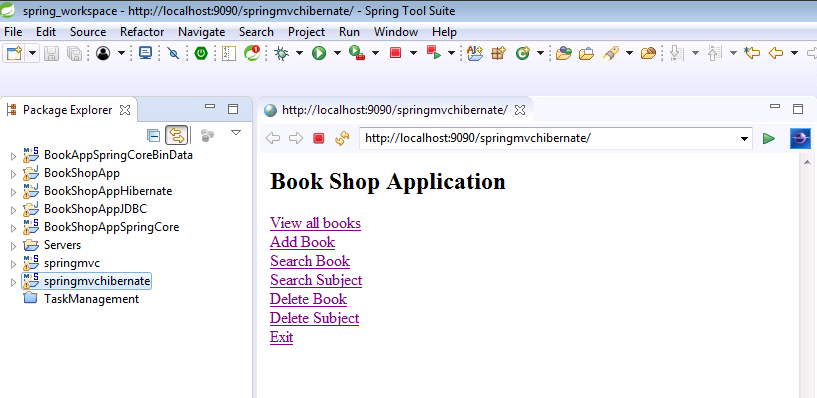
"FROM Book";

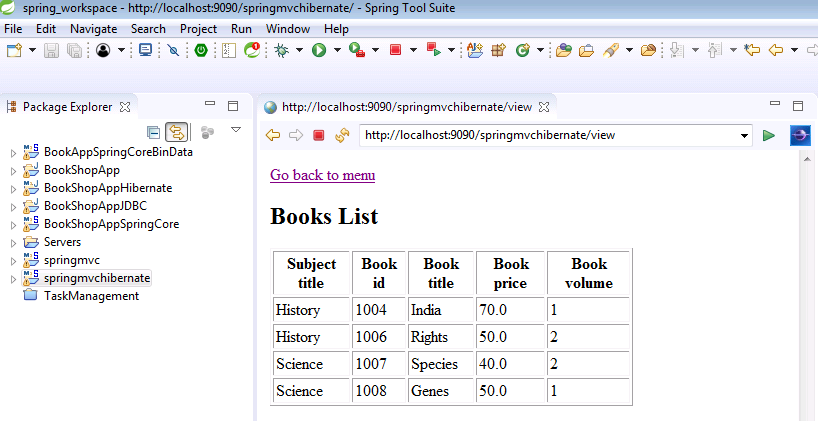
**public** **static** **final** String ***GET\_ALL\_SUBJECTS\_QRY*** =

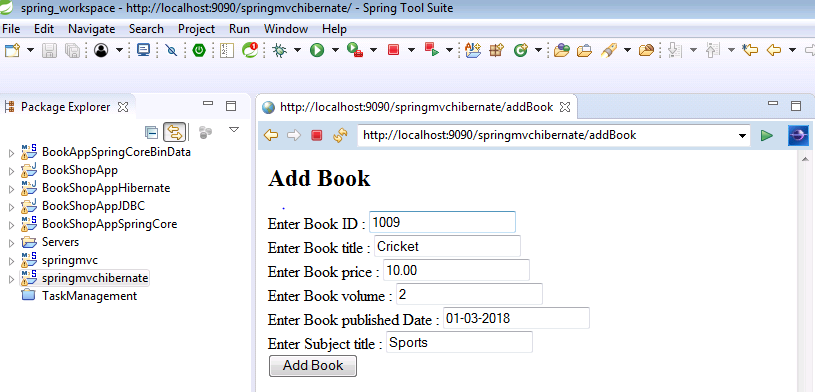
"FROM Subject";

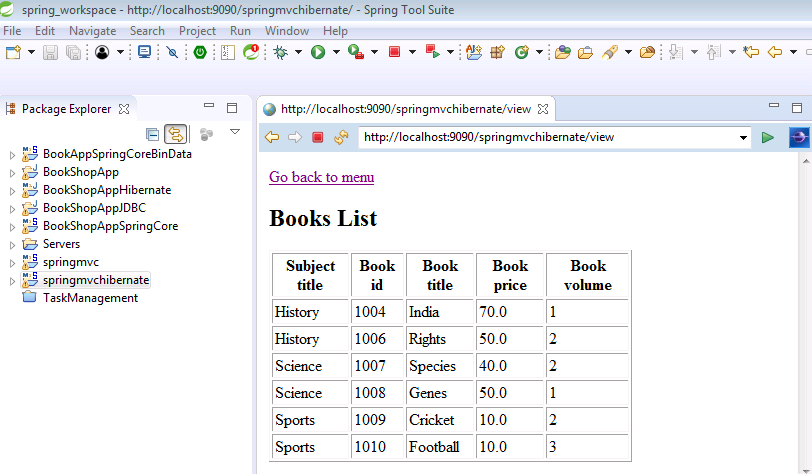
}

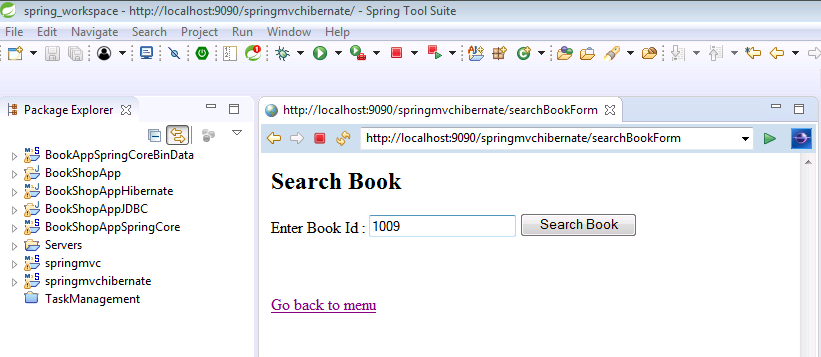
**RESULTS :**

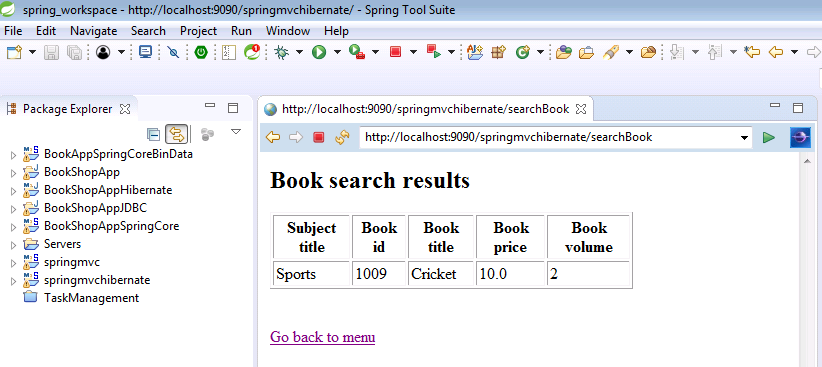
****

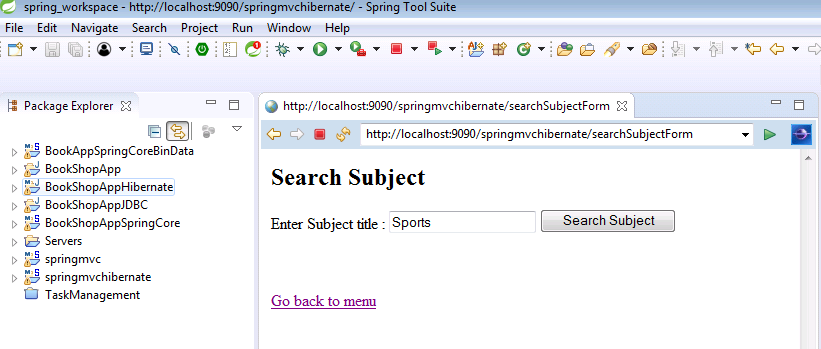
****

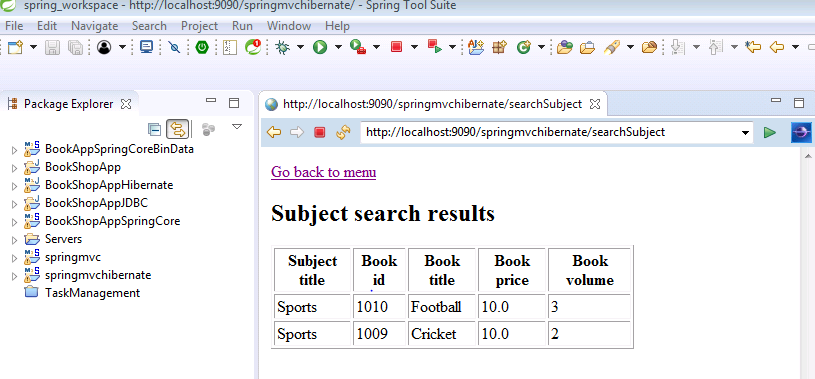
****

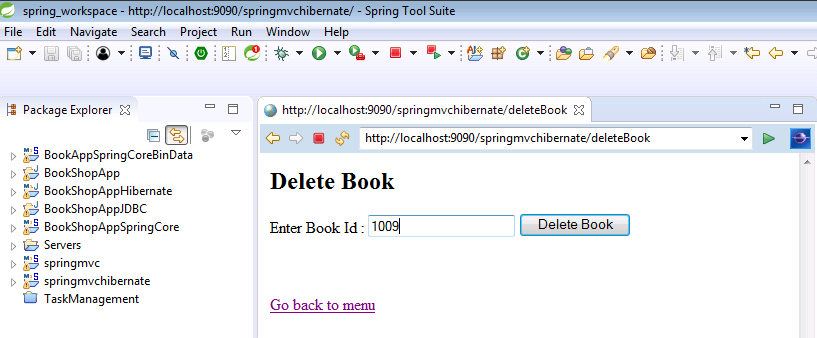
****

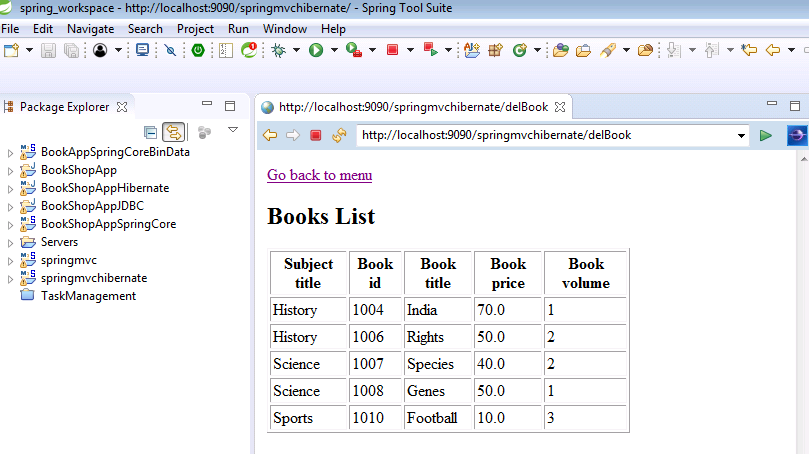
****

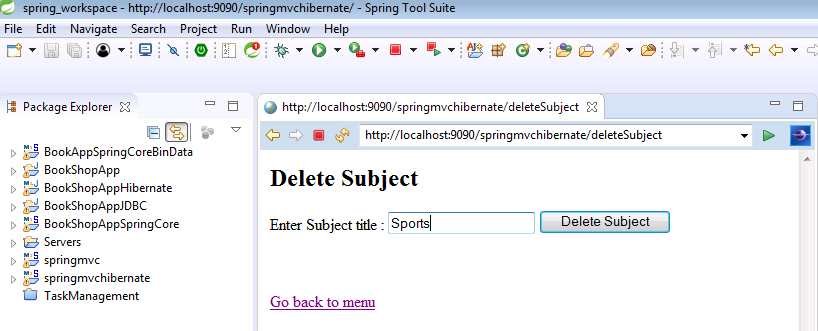
****

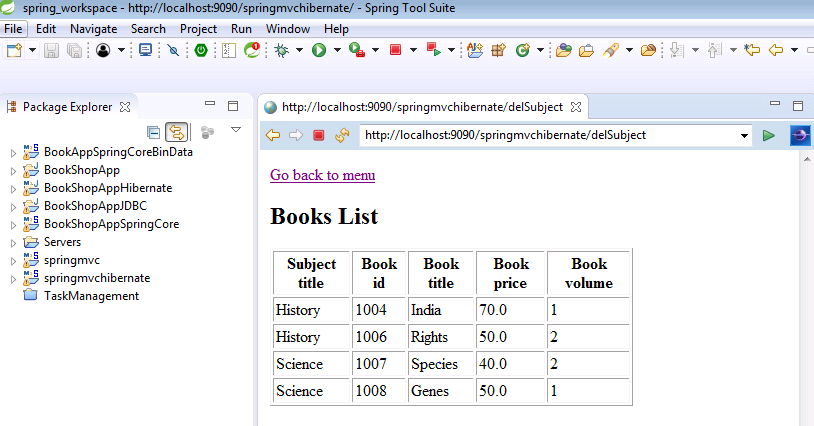
****

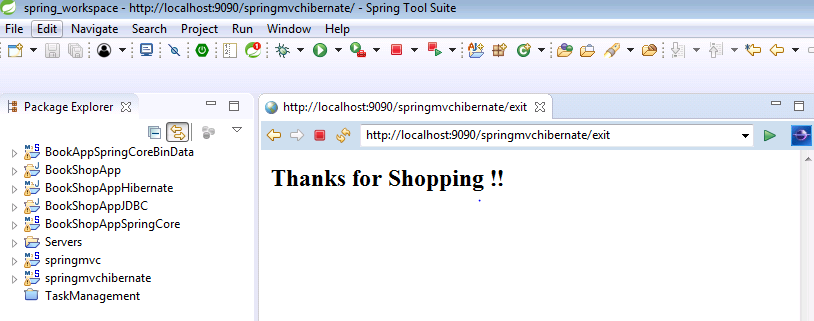
****

****

****

****

****

****