**13\_SpringBoot\_Assignment1**

pom.xml

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

<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/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.cognizant.fsd</groupId>

<artifactId>13\_SpringBoot\_Assignment1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>13\_SpringBoot\_Assignment1</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.1.1.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<properties>

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

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- <dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency> -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

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

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

<!-- JSTL -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Application.java

package com.cognizant.fsd.springbootassignment1;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

@SpringBootApplication

public class Application extends SpringBootServletInitializer {

@Override

protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

return application.sources(Application.class);

}

public static void main(String[] args) {

SpringApplication.run(Application.class, args);

}

}

SpringDataConfig.java

package com.cognizant.fsd.springbootassignment1.config;

import java.util.HashMap;

import java.util.Properties;

import javax.sql.DataSource;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.core.env.Environment;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.orm.jpa.JpaTransactionManager;

import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;

import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

@EnableTransactionManagement

@EnableJpaRepositories("com.cognizant.fsd.springbootassignment1.repository" )

@PropertySource("classpath:mysql.properties")

public class SpringDataConfig {

private final String PROPERTY\_DRIVER = "jdbc.driverClassName";

private final String PROPERTY\_URL = "jdbc.url";

private final String PROPERTY\_USERNAME = "jdbc.user";

private final String PROPERTY\_PASSWORD = "jdbc.pass";

private final String PROPERTY\_SHOW\_SQL = "hibernate.show\_sql";

private final String PROPERTY\_DIALECT = "hibernate.dialect";

@Autowired

Environment environment;

/\* @Bean

LocalContainerEntityManagerFactoryBean entityManagerFactory() {

LocalContainerEntityManagerFactoryBean lfb = new LocalContainerEntityManagerFactoryBean();

lfb.setDataSource(dataSource());

//lfb.setPersistenceProviderClass(HibernatePersistence.class);

lfb.setPackagesToScan("com.cognizant.fsd.springbootassignment1.model");

lfb.setJpaProperties(hibernateProps());

return lfb;

}\*/

@Bean

public LocalContainerEntityManagerFactoryBean entityManagerFactory() {

LocalContainerEntityManagerFactoryBean em

= new LocalContainerEntityManagerFactoryBean();

em.setDataSource(dataSource());

em.setPackagesToScan(

new String[] { "com.cognizant.fsd.springbootassignment1.model" });

HibernateJpaVendorAdapter vendorAdapter = new HibernateJpaVendorAdapter();

em.setJpaVendorAdapter(vendorAdapter);

HashMap<String, Object> properties = new HashMap<>();

properties.put("hibernate.hbm2ddl.auto",

environment.getProperty("hibernate.hbm2ddl.auto"));

properties.put("hibernate.dialect",

environment.getProperty("hibernate.dialect"));

properties.put("hibernate.show\_sql",

true);

em.setJpaPropertyMap(properties);

return em;

}

@Bean

DataSource dataSource() {

DriverManagerDataSource ds = new DriverManagerDataSource();

ds.setUrl(environment.getProperty(PROPERTY\_URL));

ds.setUsername(environment.getProperty(PROPERTY\_USERNAME));

ds.setPassword(environment.getProperty(PROPERTY\_PASSWORD));

ds.setDriverClassName(environment.getProperty(PROPERTY\_DRIVER));

return ds;

}

Properties hibernateProps() {

Properties properties = new Properties();

properties.setProperty(PROPERTY\_DIALECT, environment.getProperty(PROPERTY\_DIALECT));

properties.setProperty(PROPERTY\_SHOW\_SQL, environment.getProperty(PROPERTY\_SHOW\_SQL));

return properties;

}

@Bean

JpaTransactionManager transactionManager() {

JpaTransactionManager transactionManager = new JpaTransactionManager();

transactionManager.setEntityManagerFactory(entityManagerFactory().getObject());

return transactionManager;

}

}

**com.cognizant.fsd.springbootassignment1.controller.HomeController**

package com.cognizant.fsd.springbootassignment1.controller;

import java.util.Set;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

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.bind.annotation.RequestParam;

import com.cognizant.fsd.springbootassignment1.model.Book;

import com.cognizant.fsd.springbootassignment1.model.Subject;

import com.cognizant.fsd.springbootassignment1.service.BookService;

import com.cognizant.fsd.springbootassignment1.service.SubjectService;

@Controller

public class HomeController {

@Autowired

private SubjectService subjectService;

@Autowired

private BookService bookService;

@InitBinder

public void dataBinding(WebDataBinder binder) {

FsdCollectionEditor fsdCollectionEditor = new FsdCollectionEditor(Set.class, true);

fsdCollectionEditor.setBookService(bookService);

//binder.registerCustomEditor(Set.class, "references", fsdCollectionEditor);

}

@RequestMapping(path = { "/" }, method = RequestMethod.GET)

public String index(Model model) {

return "index";

}

@RequestMapping(value = "/addSubject", method = RequestMethod.GET)

public String addSubject(Model model) {

System.out.println("add Subject");

Subject subject = new Subject();

model.addAttribute("subject", subject);

model.addAttribute("BOOK\_LIST", bookService.fetchAllBook());

return "add-subject";

}

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

public String saveSubject(@ModelAttribute final Subject subject, Model model) {

System.out.println("save Subject ");

subject.getBookList().stream()

.filter(bookId->bookId>0)

.forEach(bookId->{

subject.getReferences().add(bookService.searchBook(bookId));

});

model.addAttribute("ENTITY\_NAME", "Subject");

subjectService.addSubject(subject);

model.addAttribute("ENTITY\_ID", subject.getSubjectId());

return "success";

}

@RequestMapping(value = "/addBook", method = RequestMethod.GET)

public String addBook(Model model) {

System.out.println("add Book");

model.addAttribute("book", new Book());

return "add-book";

}

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

public String saveBook(@ModelAttribute Book book, Model model) {

System.out.println("save book " + book.getPublishDate());

model.addAttribute("ENTITY\_NAME", "Book");

book = bookService.addBook(book);

model.addAttribute("ENTITY\_ID", book.getBookId());

return "success";

}

@RequestMapping(value = "/deleteBook", method = RequestMethod.GET)

public String deleteBook(Model model) {

System.out.println("delete Book");

return "delete-book";

}

@RequestMapping(value = "/performDeleteBook", method = RequestMethod.GET)

public String performDeleteBook(@RequestParam("bookId") String bookId, Model model) {

System.out.println("delete Book " + bookId);

boolean isSuccess = bookService.deleteBook(Long.parseLong(bookId));

model.addAttribute("ENTITY\_NAME", "Book");

model.addAttribute("ENTITY\_ID", bookId);

model.addAttribute("isSuccess", isSuccess);

return "delete-success";

}

@RequestMapping(value = "/deleteSubject", method = RequestMethod.GET)

public String deleteSubject(Model model) {

System.out.println("delete Book");

return "delete-subject";

}

@RequestMapping(value = "/performDeleteSubject", method = RequestMethod.GET)

public String performDeleteSubject(@RequestParam("subjectId") String subjectId, Model model) {

System.out.println("delete subjectId " + subjectId);

boolean isSuccess = subjectService.deleteSubject(Long.parseLong(subjectId));

model.addAttribute("ENTITY\_NAME", "Subject");

model.addAttribute("isSuccess", isSuccess);

model.addAttribute("ENTITY\_ID", subjectId);

return "delete-success";

}

@RequestMapping(value = "/search", method = RequestMethod.GET)

public String search(Model model) {

System.out.println("search Subject");

return "search";

}

@RequestMapping(value = "/performSearch", method = RequestMethod.GET)

public String performSearch(@RequestParam("id") String id, @RequestParam("entityName") String entityName,

Model model) {

System.out.println(entityName + "Perform Search" + id);

Subject subject = null;

Book book = null;

String name = "";

if (entityName.equals("subject")) {

subject = subjectService.searchSubject(Long.parseLong(id));

System.out.println("subject "+subject.getSubjectId());

name = "subject";

}

if (entityName.equals("book")) {

book = bookService.searchBook(Long.parseLong(id));

name = "book";

}

model.addAttribute("ENTITY\_NAME", name);

model.addAttribute("book", book);

model.addAttribute("subject", subject);

return "result";

}

@RequestMapping(value = "/exit", method = RequestMethod.GET)

public String exit(Model model) {

System.out.println("exit");

return "exit";

}

}

com.cognizant.fsd.springbootassignment1.controller.FsdCollectionEditor

package com.cognizant.fsd.springbootassignment1.controller;

import java.util.Collection;

import org.springframework.beans.propertyeditors.CustomCollectionEditor;

import com.cognizant.fsd.springbootassignment1.model.Book;

import com.cognizant.fsd.springbootassignment1.service.BookService;

public class FsdCollectionEditor extends CustomCollectionEditor {

private BookService bookService;

public FsdCollectionEditor(Class<? extends Collection> collectionType, boolean nullAsEmptyCollection) {

super(collectionType, nullAsEmptyCollection);

}

protected Object convertElement(Object element) {

System.out.println("element : "+element);

Book book = bookService.searchBook((Long)element);

return book;

}

public BookService getBookService() {

return bookService;

}

public void setBookService(BookService bookService) {

this.bookService = bookService;

}

}

com.cognizant.fsd.springbootassignment1.model.Book

/\*\*

\*

\*/

package com.cognizant.fsd.springbootassignment1.model;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

import javax.persistence.Transient;

import org.springframework.format.annotation.DateTimeFormat;

/\*\*

\* @author Santi Chowhury

\*

\*/

@Entity

@Table(name="book")

public class Book implements Serializable{

@Transient

private static final long serialVersionUID = 3667779253735136971L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

//private Long bid;

@Column(name="BOOKID")

private Long bookId;

@Column(name="TITLE")

private String title;

@Column(name="PRICE")

private Double price;

@Column(name="VOLUME")

private Integer volume;

@Column(name="PUBLISHDATE")

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

private LocalDate publishDate;

@ManyToMany(mappedBy = "references",fetch=FetchType.EAGER)

private Set<Subject> references = new HashSet<>();

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 Integer getVolume() {

return volume;

}

public void setVolume(Integer volume) {

this.volume = volume;

}

public LocalDate getPublishDate() {

return publishDate;

}

public void setPublishDate(LocalDate publishDate) {

this.publishDate = publishDate;

}

public Set<Subject> getReferences() {

return references;

}

public void setReferences(Set<Subject> references) {

this.references = references;

}

@Override

public String toString() {

return "Book [bookId=" + bookId + ", title=" + title + ", price=" + price + ", volume=" + volume

+ ", publishDate=" + publishDate + "]";

}

}

com.cognizant.fsd.springbootassignment1.model.Subject

package com.cognizant.fsd.springbootassignment1.model;

import java.io.Serializable;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

import javax.persistence.Transient;

/\*\*

\* @author Santi Chowdhury

\*

\*/

@Entity

@Table(name="subject")

public class Subject implements Serializable{

@Transient

private static final long serialVersionId = 1L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

//private Long sid;

@Column(name="SUBJECTID")

private Long subjectId;

@Column(name="SUBTITLE")

private String subTitle;

@Column(name="DURATIONINHOURS")

private Integer durationInHours;

@ManyToMany(fetch=FetchType.EAGER)

private Set<Book> references=new HashSet<Book>();

@Transient

private List<Long> bookList;

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 Integer getDurationInHours() {

return durationInHours;

}

public void setDurationInHours(Integer durationInHours) {

this.durationInHours = durationInHours;

}

public Set<Book> getReferences() {

return references;

}

public void setReferences(Set<Book> references) {

this.references = references;

}

@Override

public String toString() {

return "Subject [subjectId=" + subjectId + ", subTitle=" + subTitle + ", durationInHours=" + durationInHours

+ ", references=" + references + "]";

}

public List<Long> getBookList() {

return bookList;

}

public void setBookList(List<Long> bookList) {

this.bookList = bookList;

}

}

com.cognizant.fsd.springbootassignment1.service.BookService

package com.cognizant.fsd.springbootassignment1.service;

import java.util.List;

import com.cognizant.fsd.springbootassignment1.model.Book;

public interface BookService {

public Book addBook(Book book);

public boolean deleteBook(long bookId);

public Book searchBook(long bookId);

public List<Book> fetchAllBook();

}

com.cognizant.fsd.springbootassignment1.service.BookServiceImpl

package com.cognizant.fsd.springbootassignment1.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.fsd.springbootassignment1.model.Book;

import com.cognizant.fsd.springbootassignment1.repository.BookRepository;

@Service("bookService")

@Transactional

public class BookServiceImpl implements BookService {

@Autowired

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

@Override

public Book addBook(Book book) {

return bookRepository.save(book);

}

@Override

public boolean deleteBook(long bookId) {

bookRepository.deleteById(bookId);

return true;

}

@Override

public Book searchBook(long bookId) {

return bookRepository.findById(bookId).get();

}

@Override

public List<Book> fetchAllBook() {

return (List<Book>) bookRepository.findAll();

}

}

com.cognizant.fsd.springbootassignment1.service.SubjectService

package com.cognizant.fsd.springbootassignment1.service;

import java.util.List;

import com.cognizant.fsd.springbootassignment1.model.Subject;

public interface SubjectService {

public Subject addSubject(Subject subject);

public boolean deleteSubject(long subjectId);

public Subject searchSubject(long subjectId);

public List<Subject> fetchAllSubject();

}

com.cognizant.fsd.springbootassignment1.service.SubjectServiceImpl

package com.cognizant.fsd.springbootassignment1.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.fsd.springbootassignment1.model.Subject;

import com.cognizant.fsd.springbootassignment1.repository.SubjectRepository;

@Service("subjectService")

@Transactional

public class SubjectServiceImpl implements SubjectService{

@Autowired

private SubjectRepository subjectRepository;

public void setSubjectRepository(SubjectRepository subjectRepository) {

this.subjectRepository = subjectRepository;

}

@Override

public Subject addSubject(Subject subject) {

return subjectRepository.save(subject);

}

@Override

public boolean deleteSubject(long subjectId) {

subjectRepository.deleteById(subjectId);

return true;

}

@Override

public Subject searchSubject(long subjectId) {

return subjectRepository.findById(subjectId).get();

}

@Override

public List<Subject> fetchAllSubject() {

return (List<Subject>) subjectRepository.findAll();

}

}

**mysql.properties**

# jdbc

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/spring\_assignment\_13?createDatabaseIfNotExist=true

jdbc.user=root

jdbc.pass=pass@word1

# hibernate

hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

hibernate.show\_sql=true

hibernate.hbm2ddl.auto=update

**application.properties**

server.port=9000

spring.mvc.view.prefix=/WEB-INF/views/

spring.mvc.view.suffix=.jsp

**JSPS**

add-book.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<table style="width: *100%*">

<form:form method=*"POST"* modelAttribute=*"book"* action=*"*${pageContext.request.contextPath}*/saveBook"*>

<tr>

<td><form:label path=*"bookId"*>Book Id :</form:label></td>

<td><form:input path=*"bookId"* /></td>

</tr>

<tr>

<td><form:label path=*"title"*>Book Title :</form:label></td>

<td><form:input path=*"title"* /></td>

</tr>

<tr>

<td><form:label path=*"price"*>Book Price :</form:label></td>

<td><form:input path=*"price"* /></td>

</tr>

<tr>

<td><form:label path=*"volume"*>Book Volume :</form:label></td>

<td><form:input path=*"volume"* /></td>

</tr>

<tr>

<td><form:label path=*"publishDate"*>Publish Date (yyyy-MM-dd):</form:label></td>

<td><form:input path=*"publishDate"* /></td>

</tr>

<tr>

<td colspan=*"2"*><input type=*"submit"* value=*"save"*/></td>

</tr>

</form:form>

</table>

</body>

</html>

add-subject.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<table style="width:*100%*">

<form:form method=*"POST"* modelAttribute=*"subject"* action=*"*${pageContext.request.contextPath}*/saveSubject"* >

<tr>

<td><form:label path=*"subjectId"*>Subject Id :</form:label></td>

<td><form:input path=*"subjectId"*/></td>

</tr>

<tr>

<td><form:label path=*"subTitle"*>Subject Title :</form:label></td>

<td><form:input path=*"subTitle"*/></td>

</tr>

<tr>

<td><form:label path=*"durationInHours"*>Duration In Hours :</form:label></td>

<td><form:input path=*"durationInHours"*/></td>

</tr>

<tr>

<td><form:label path=*"references"*>Book References :</form:label></td>

<td>

<form:select path=*"bookList"*>

<form:option value=*"0"* label=*"----- Select -----"*/>

<form:options items=*"*${BOOK\_LIST}*"* itemLabel=*"title"* itemValue=*"bookId"*/>

</form:select>

</td>

</tr>

<tr>

<td colspan=*"2"*><input type=*"submit"* value=*"save"*/></td>

</tr>

</form:form>

</table>

</body>

</html>

delete-book.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>enter the book id to delete :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performDeleteBook"*>

<input id=*"bookId"* name=*"bookId"*/>

<input type=*"submit"* value=*"delete"*>

</form>

</body>

</html>

delete-subject.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>enter the subject id to delete :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performDeleteSubject"*>

<input id=*"subjectId"* name=*"subjectId"*/>

<input type=*"submit"* value=*"delete"*>

</form>

</body>

</html>

delete-success.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>The ${ENTITY\_NAME} with id (${ENTITY\_ID}) has been deleted successfully</h1>

</body>

</html>

error.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>Something went wrong, Please try again</h1>

</body>

</html>

exit.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>Thanks for using this application</h1>

</body>

</html>

index.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

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

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

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

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Spring 4 MVC - Hello World Example</title>

</head>

<body>

<h1>Welcome to Spring Boot with Spring Data(JPA)</h1>

<a href=*"*<spring:url value=*"/addSubject"*/>*"*>Add Subject</a><br/>

<a href=*"*<spring:url value=*"/addBook"*/>*"*>Add Book</a><br/>

<a href=*"*<spring:url value=*"/deleteSubject"*/>*"*>Delete Subject</a><br/>

<a href=*"*<spring:url value=*"/deleteBook"*/>*"*>Delete Book</a><br/>

<a href=*"*<spring:url value=*"/search"*/>*"*>Search</a><br/>

<a href=*"*<spring:url value=*"/exit"*/>*"*>Exit</a><br/>

</body>

</html>

result.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a>

<br />

<hr />

<c:choose>

<c:when test=*"*${ENTITY\_NAME **eq** 'subject'}*"*>

<table>

<tr>

<td>Subject Id :</td>

<td>${subject.subjectId}</td>

</tr>

<tr>

<td>Subject Title :</td>

<td>${subject.subTitle}</td>

</tr>

<tr>

<td>Duration In Hours :</td>

<td>${subject.durationInHours}</td>

</tr>

<tr>

<td>Reference :</td>

<td>

<ol>

<c:forEach items=*"*${subject.references}*"* var=*"book"* varStatus=*"counter"*>

<li>${book.title} <b>(${book.bookId})</b></li>

</c:forEach>

</ol>

</td>

</tr>

</table>

</c:when>

<c:when test=*"*${ENTITY\_NAME **eq** 'book'}*"*>

<table style="width: *100%*">

<tr>

<td>Book Id :</td>

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

</tr>

<tr>

<td>Book Title :</td>

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

</tr>

<tr>

<td>Price :</td>

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

</tr>

<tr>

<td>Volume :</td>

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

</tr>

<tr>

<td>Publish Date (yyyy-MM-dd) :</td>

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

</tr>

</table>

</c:when>

<c:otherwise>

</c:otherwise>

</c:choose>

</body>

</html>

search.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>please choose entity & enter the id :</h1>

<form method=*"get"* action=*"*${pageContext.request.contextPath}*/performSearch"*>

<input type=*"radio"* id=*"entityName1"* name=*"entityName"* value=*"subject"* checked> Subject<br>

<input type=*"radio"* id=*"entityName2"* name=*"entityName"* value=*"book"*> Book<br>

<input type=*"text"* id=*"id"* name=*"id"*/>

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

</form>

</body>

</html>

success.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

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

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

</head>

<body>

<a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>

<hr/>

<h1>The ${ENTITY\_NAME} with id (${ENTITY\_ID}) has been saved successfully</h1>

</body>

</html>

com.cognizant.fsd.springbootassignment1.repository.AbstractDao

package com.cognizant.fsd.springbootassignment1.repository;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

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

public abstract class AbstractDao {

@Autowired

private SessionFactory sessionFactory;

protected Session getSession() {

return sessionFactory.getCurrentSession();

}

public void persist(Object entity) {

getSession().saveOrUpdate(entity);

}

public void delete(Object entity) {

getSession().delete(entity);

}

}

com.cognizant.fsd.springbootassignment1.repository.BookRepository

package com.cognizant.fsd.springbootassignment1.repository;

import org.springframework.data.repository.CrudRepository;

import com.cognizant.fsd.springbootassignment1.model.Book;

public interface BookRepository extends CrudRepository<Book, Long>{

}

com.cognizant.fsd.springbootassignment1.repository.SubjectRepository

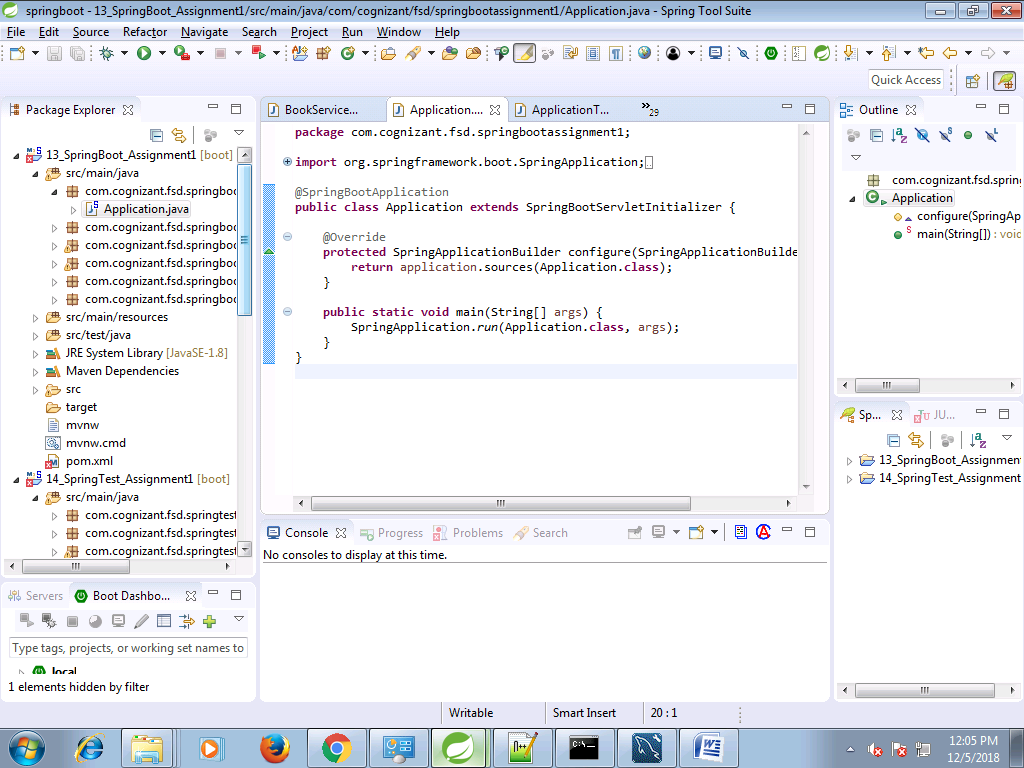
package com.cognizant.fsd.springbootassignment1.repository;

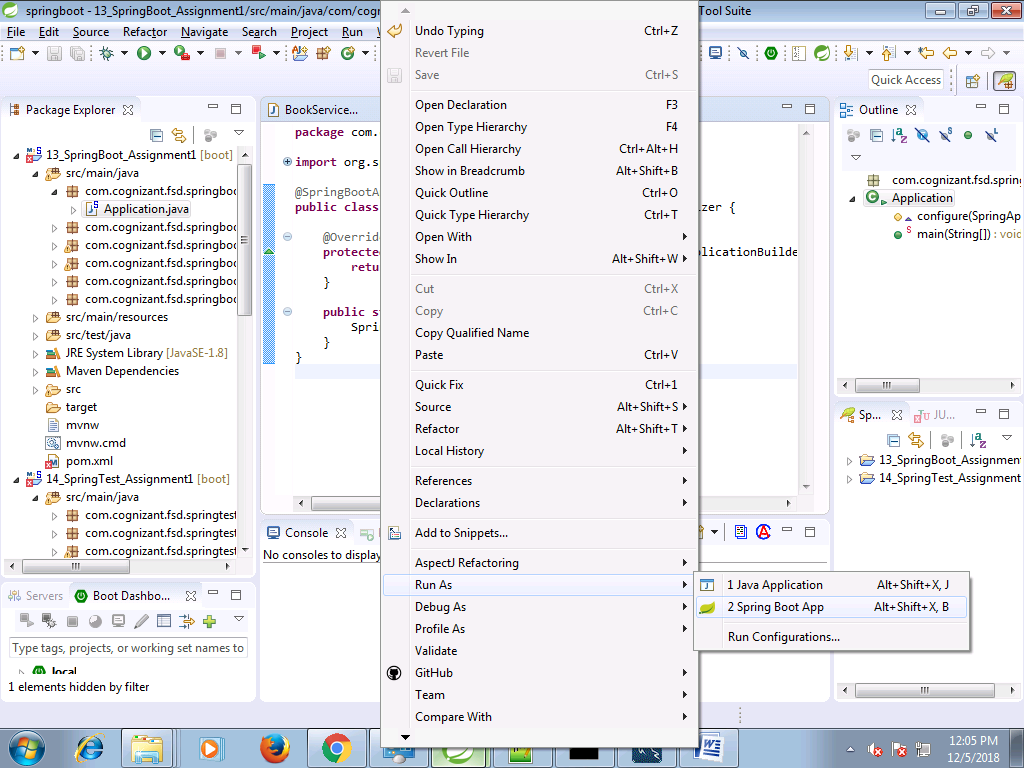
import org.springframework.data.repository.CrudRepository;

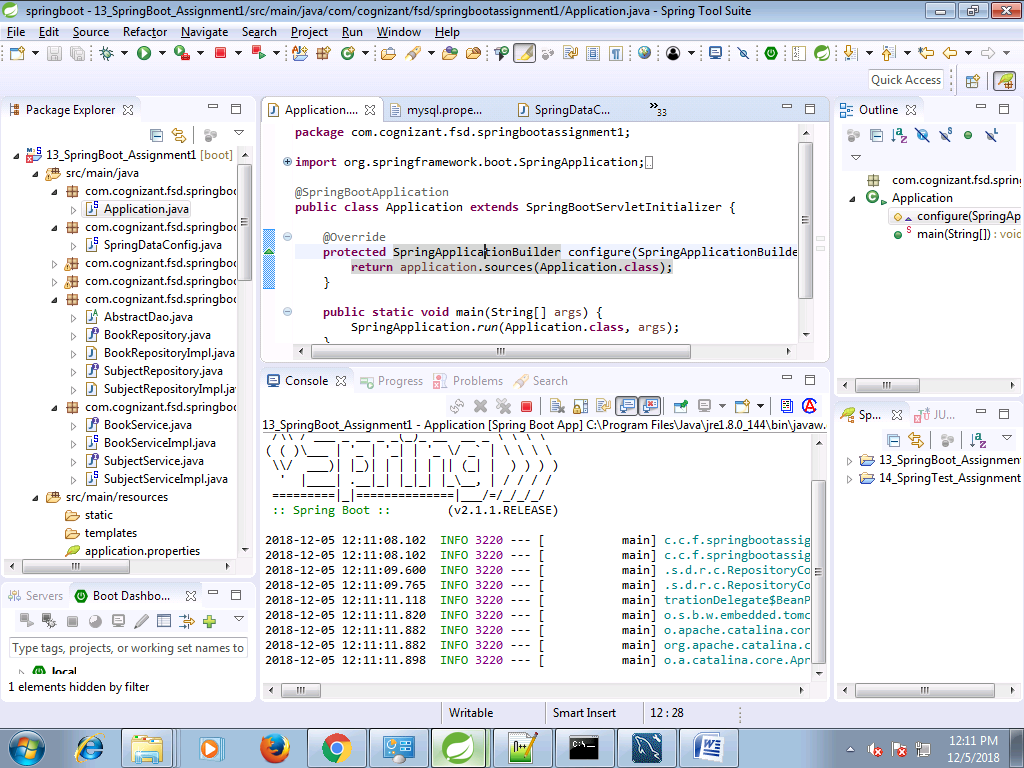
import com.cognizant.fsd.springbootassignment1.model.Subject;

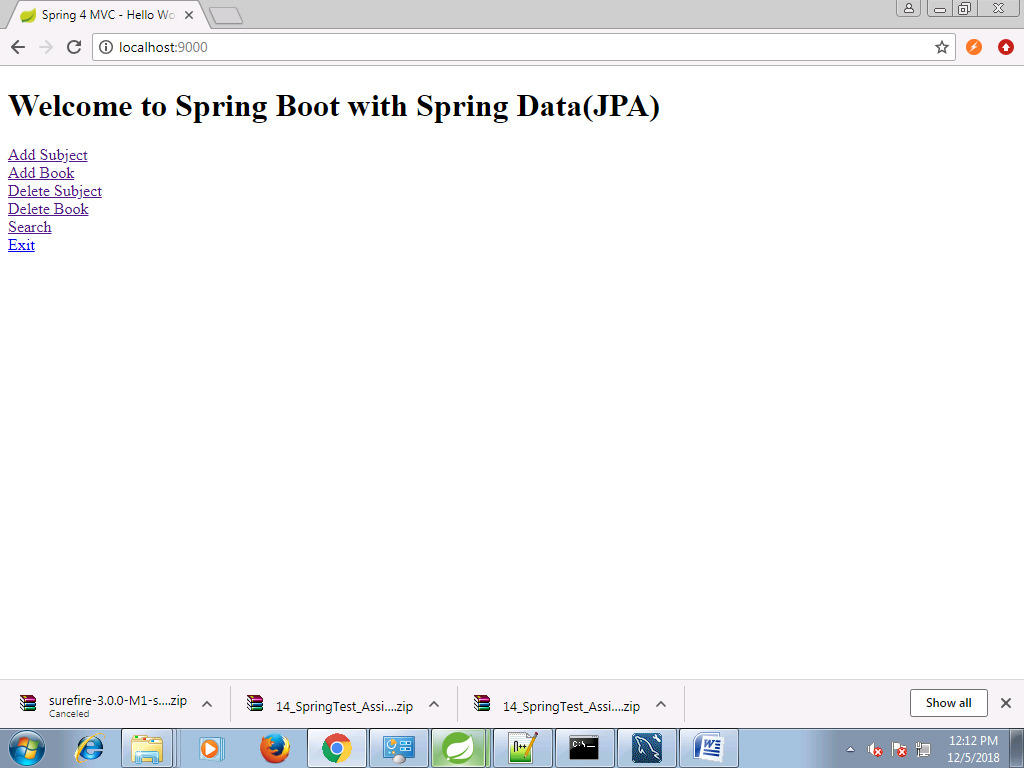
public interface SubjectRepository extends CrudRepository<Subject, Long>{

}

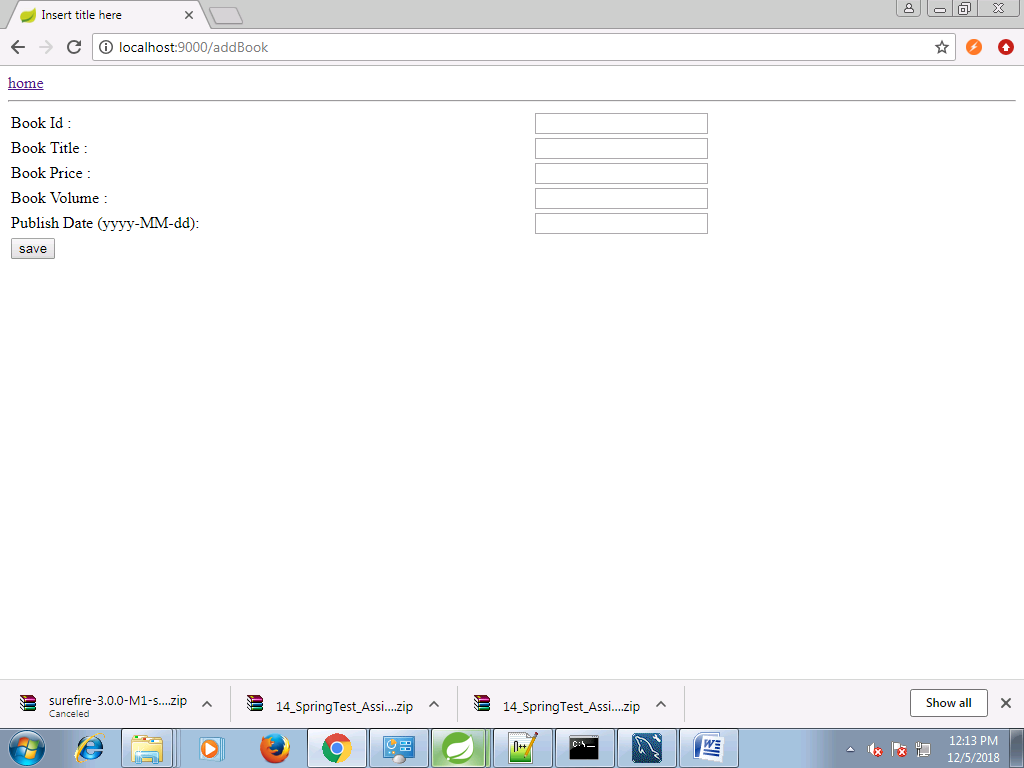


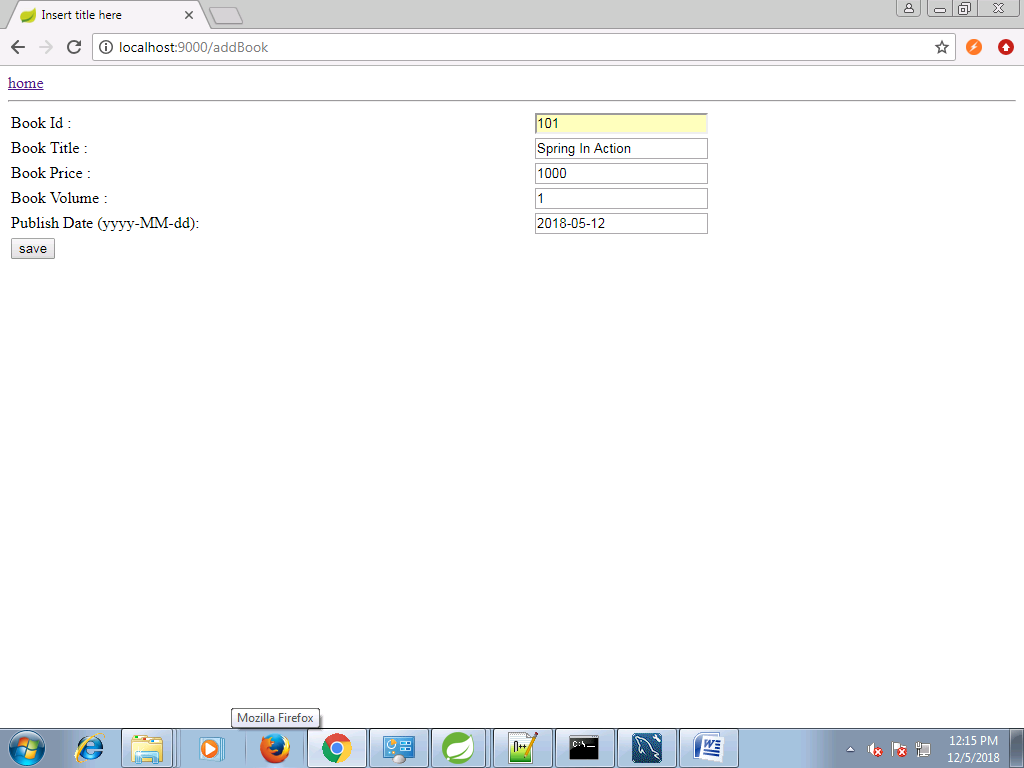


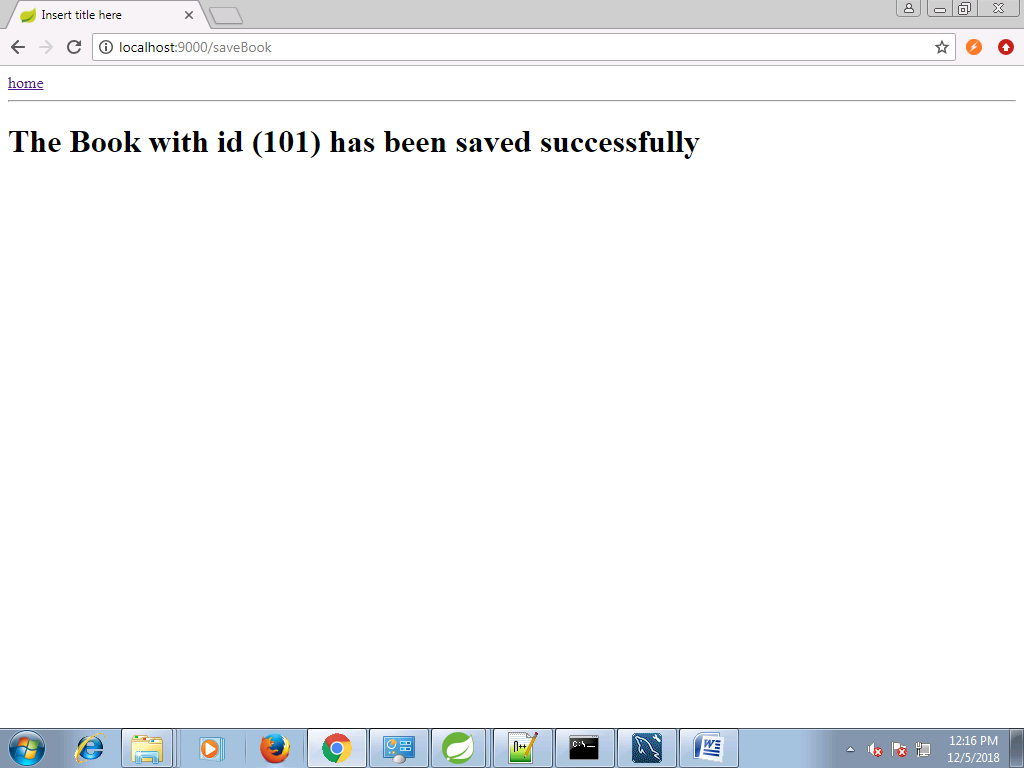


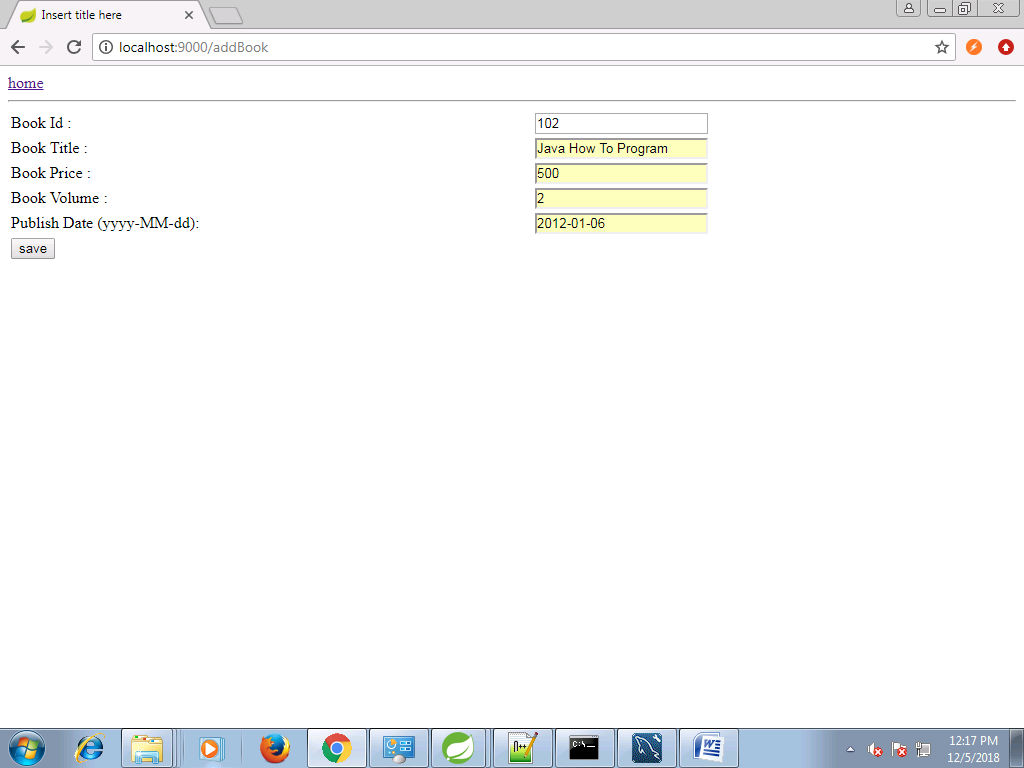


[Add Book](http://localhost:9000/addBook)



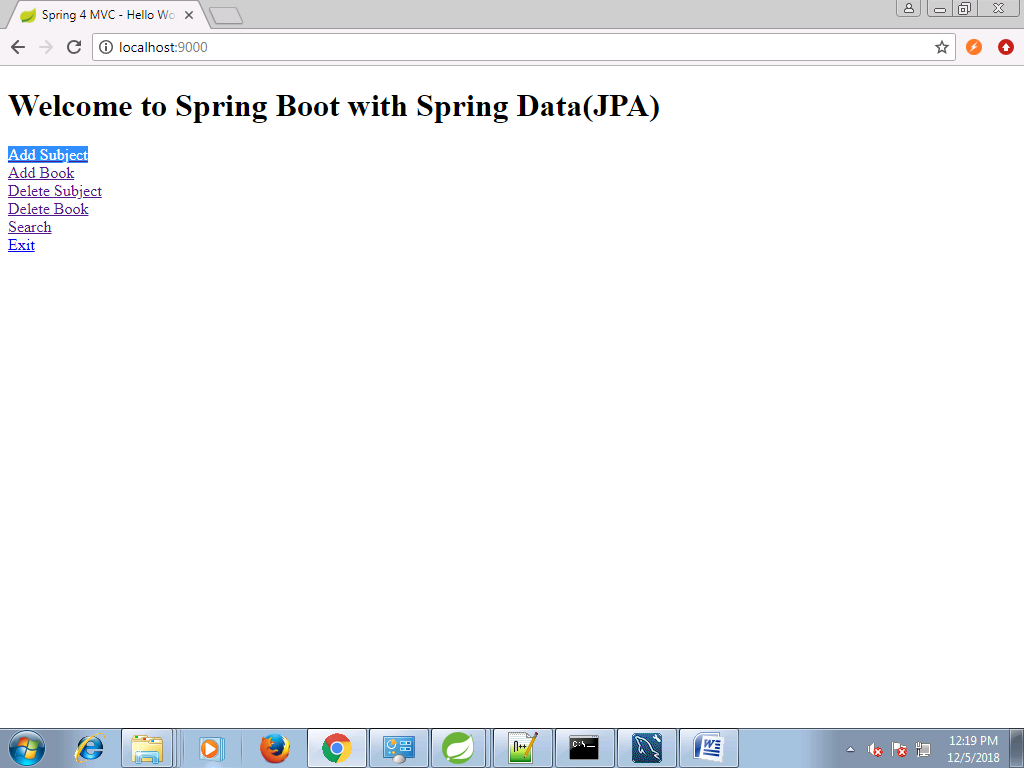


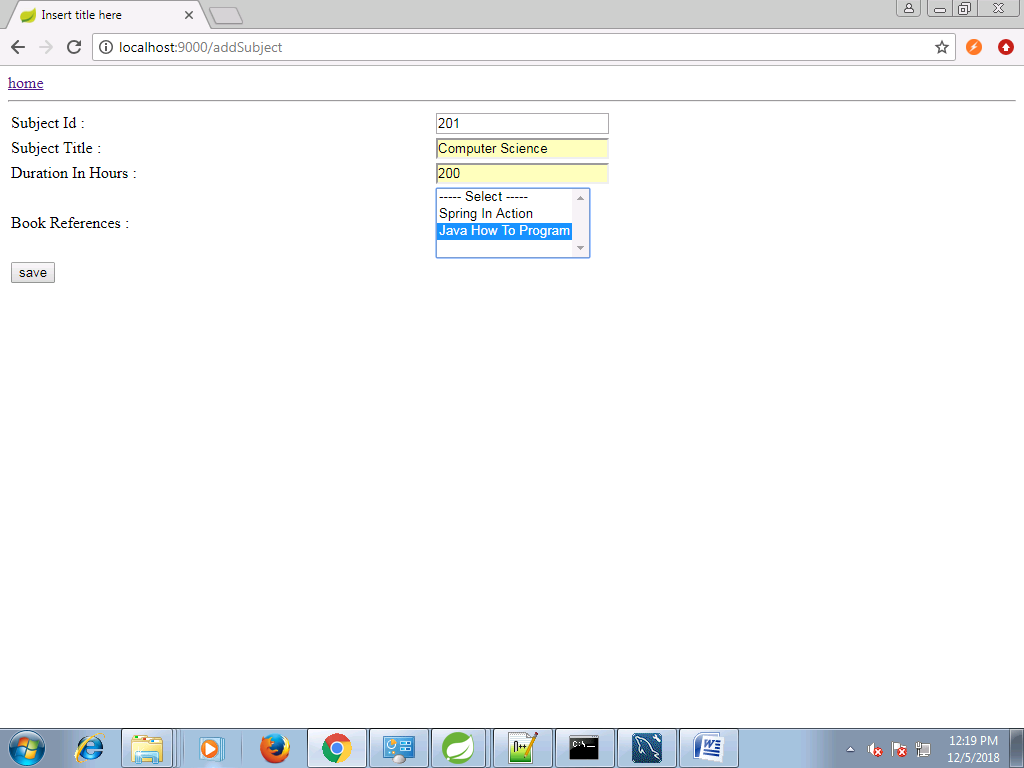


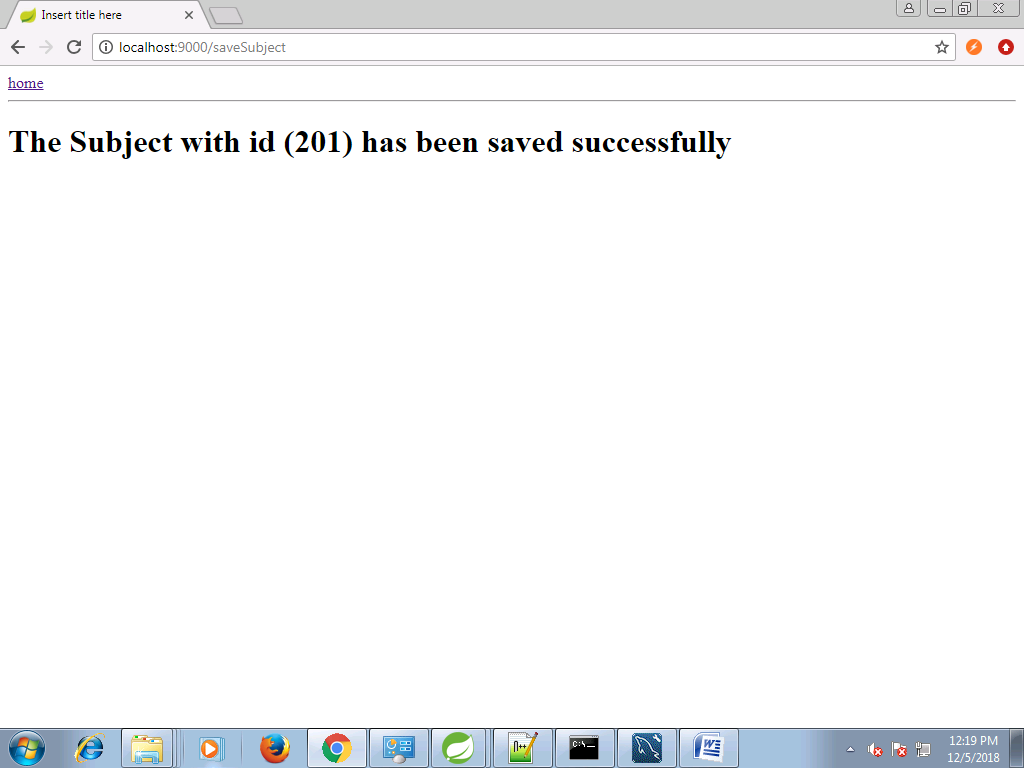


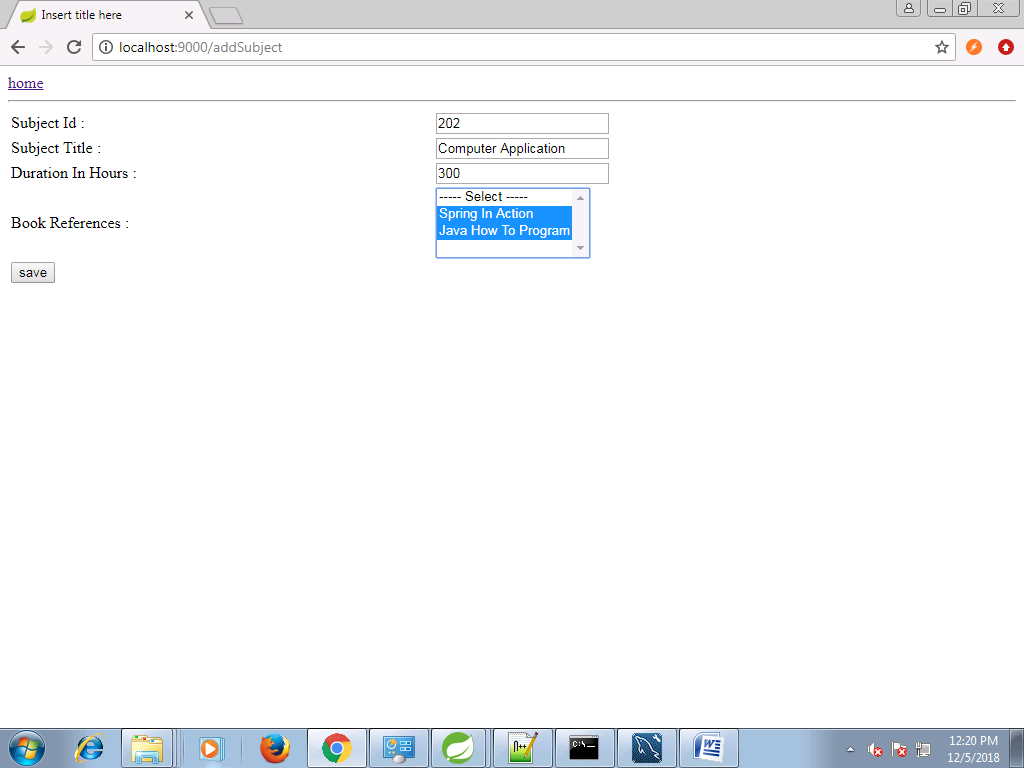


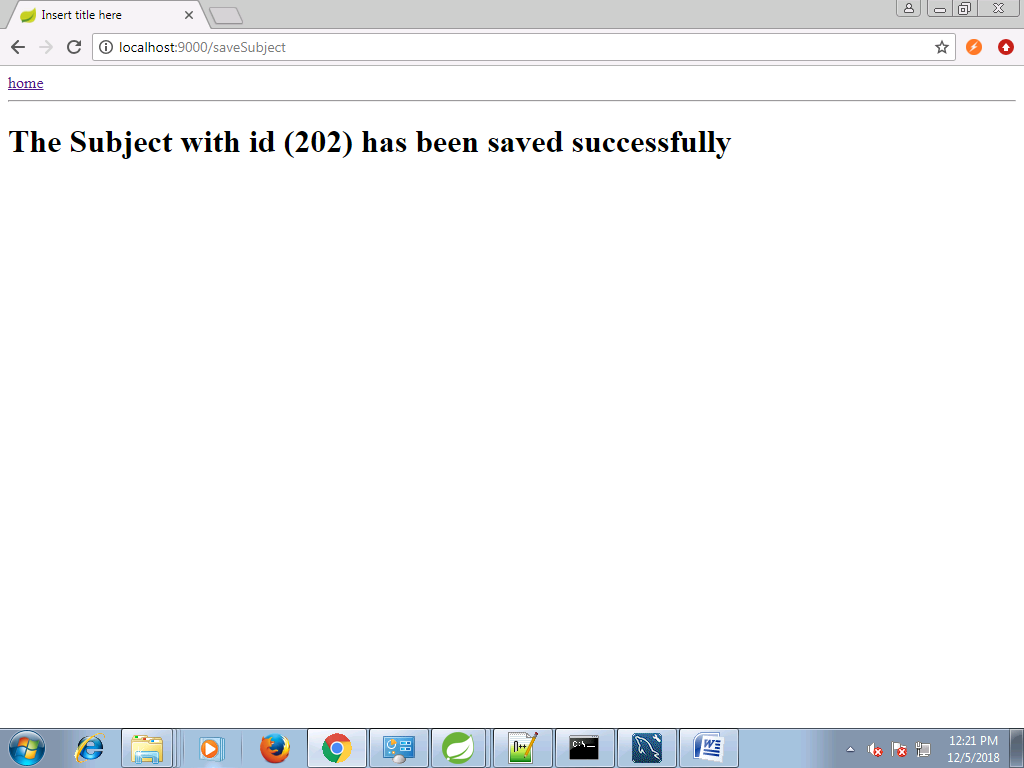
[Add Subject](http://localhost:9000/addSubject)

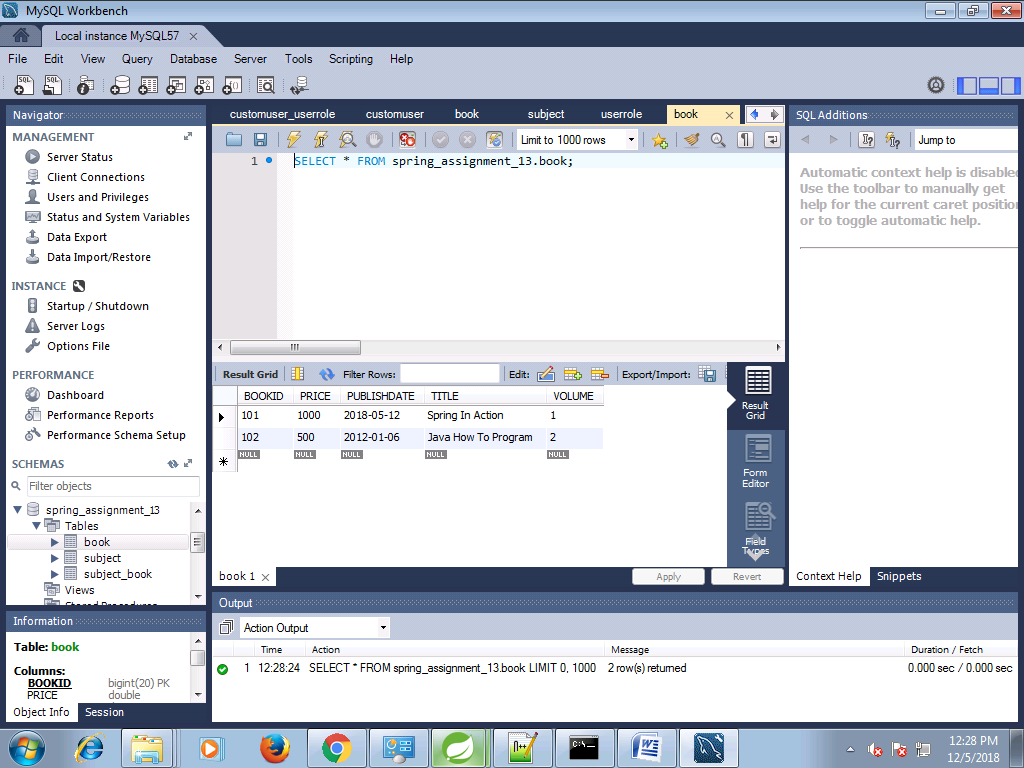


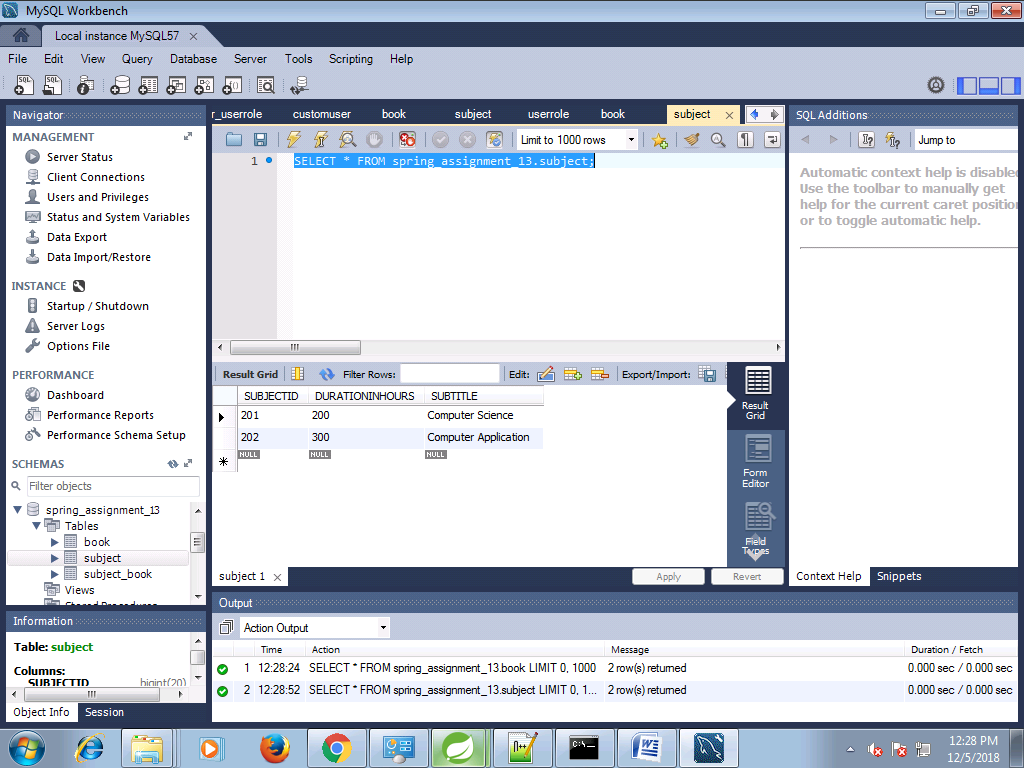


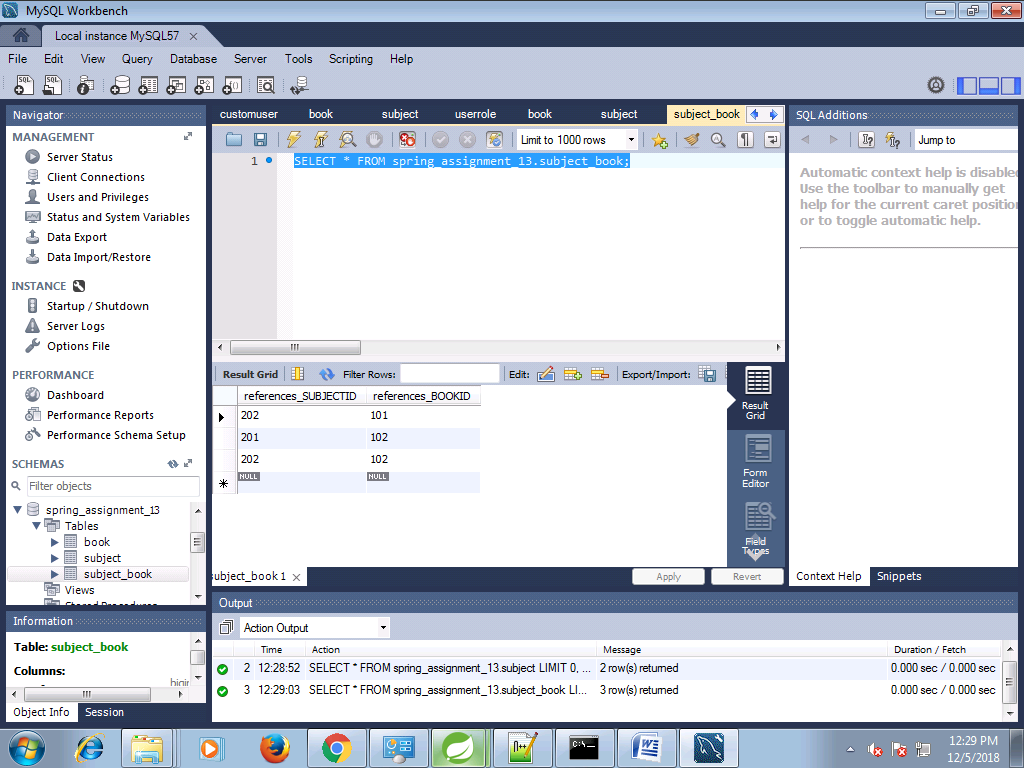


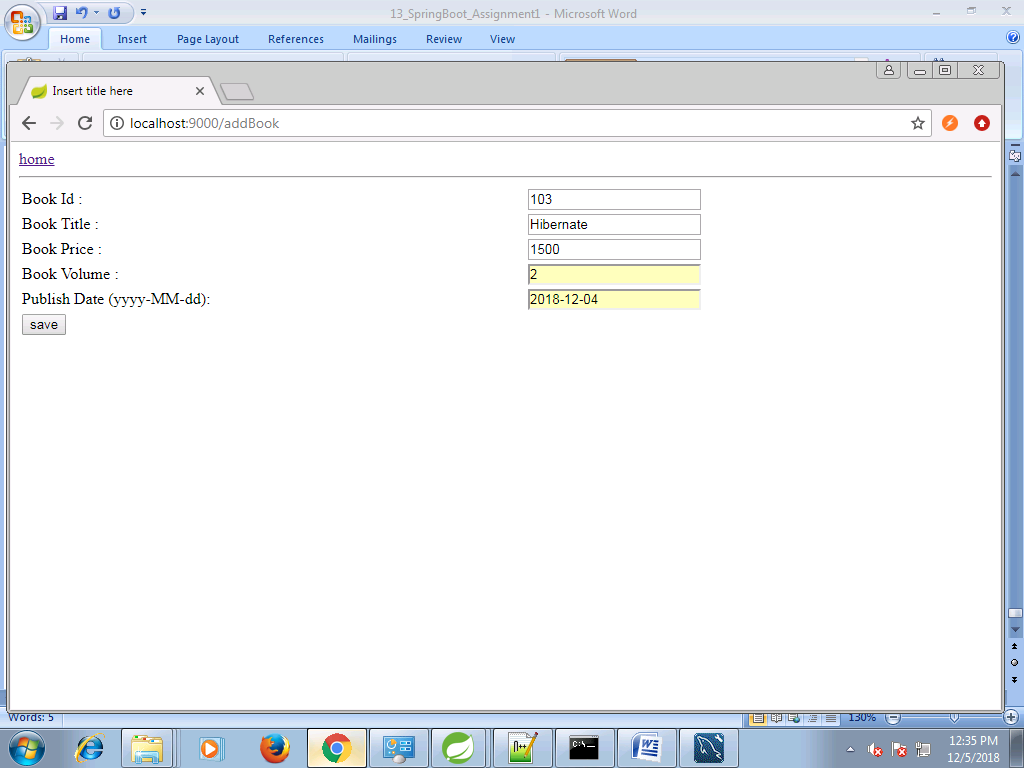


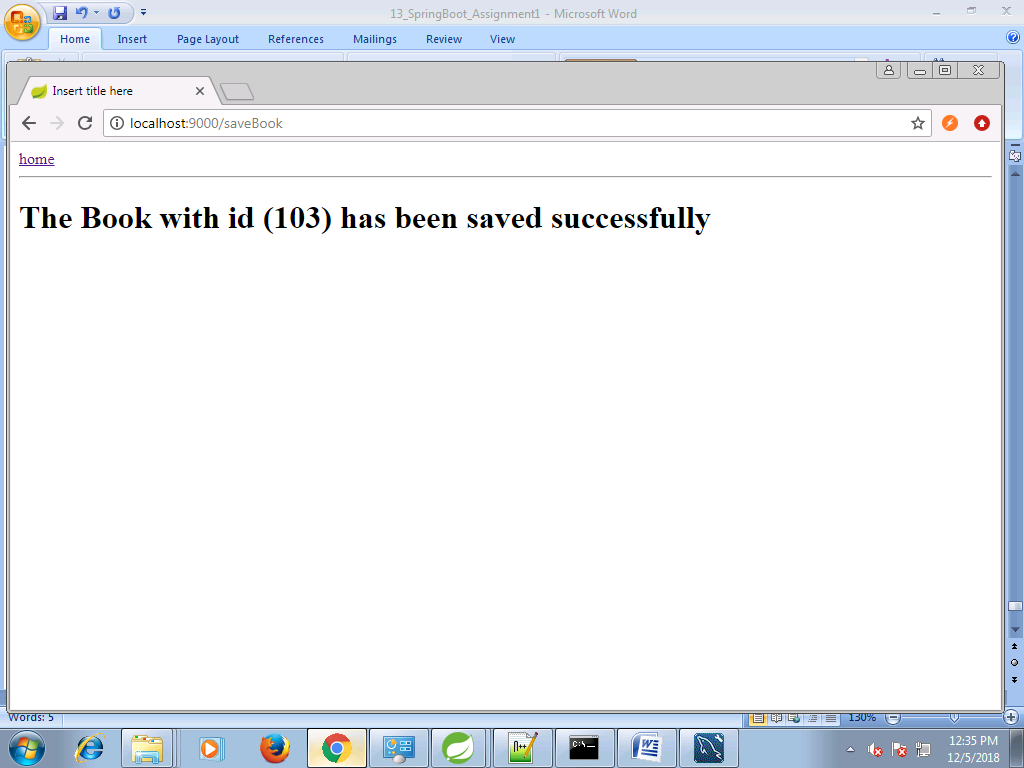


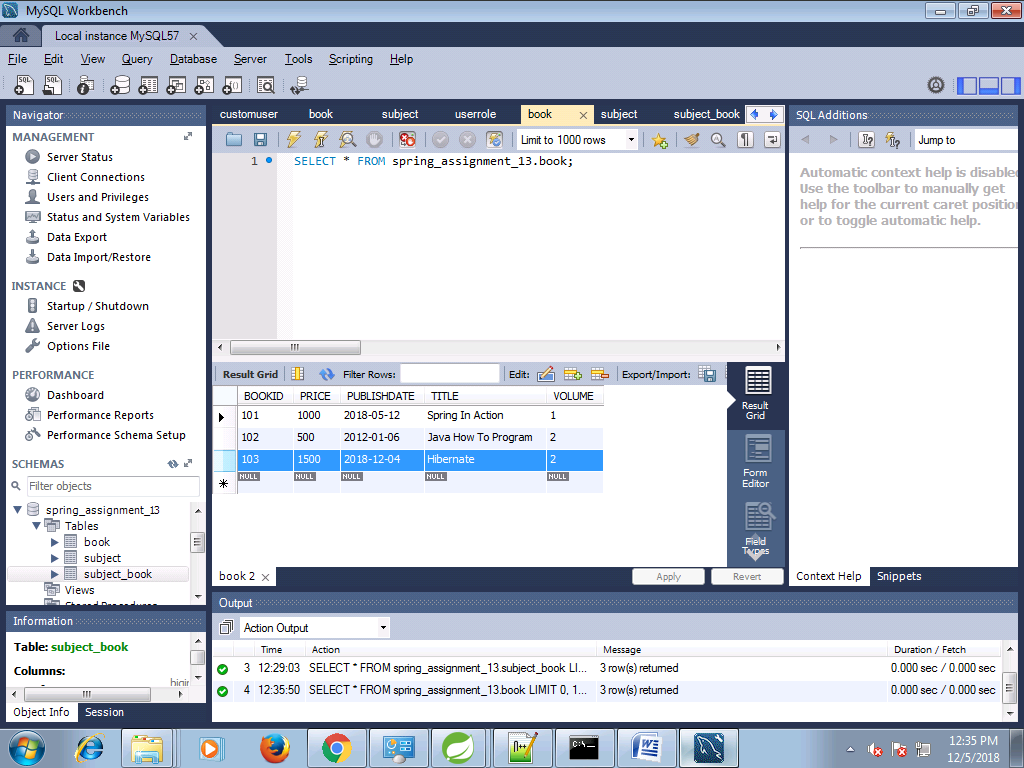




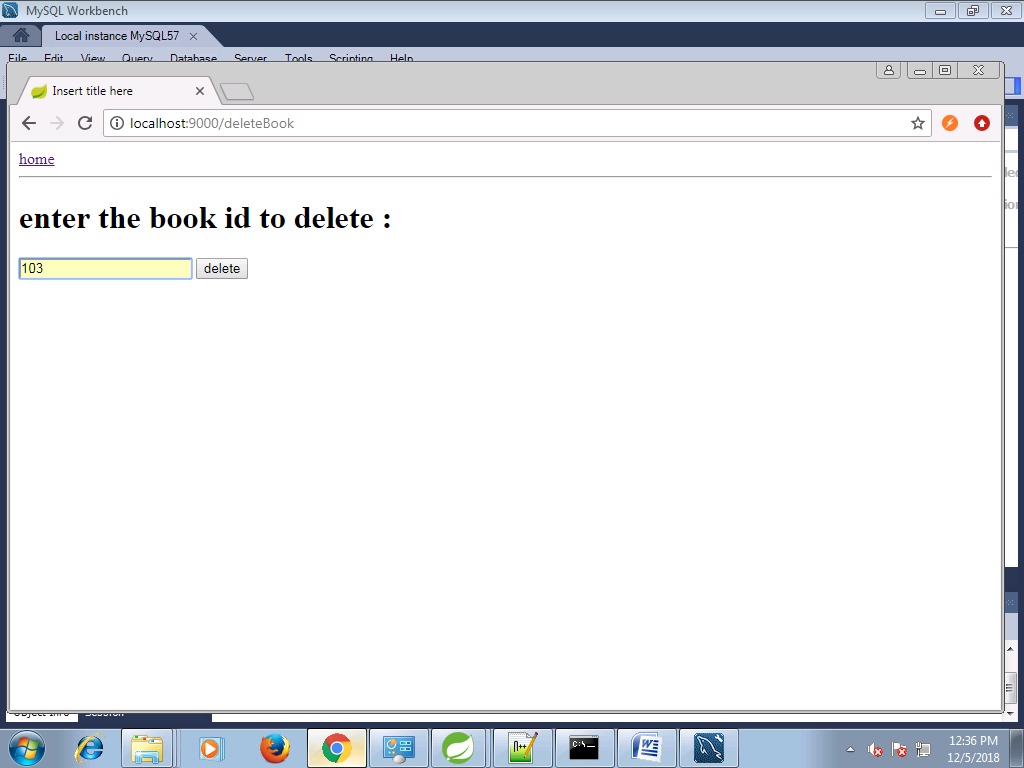


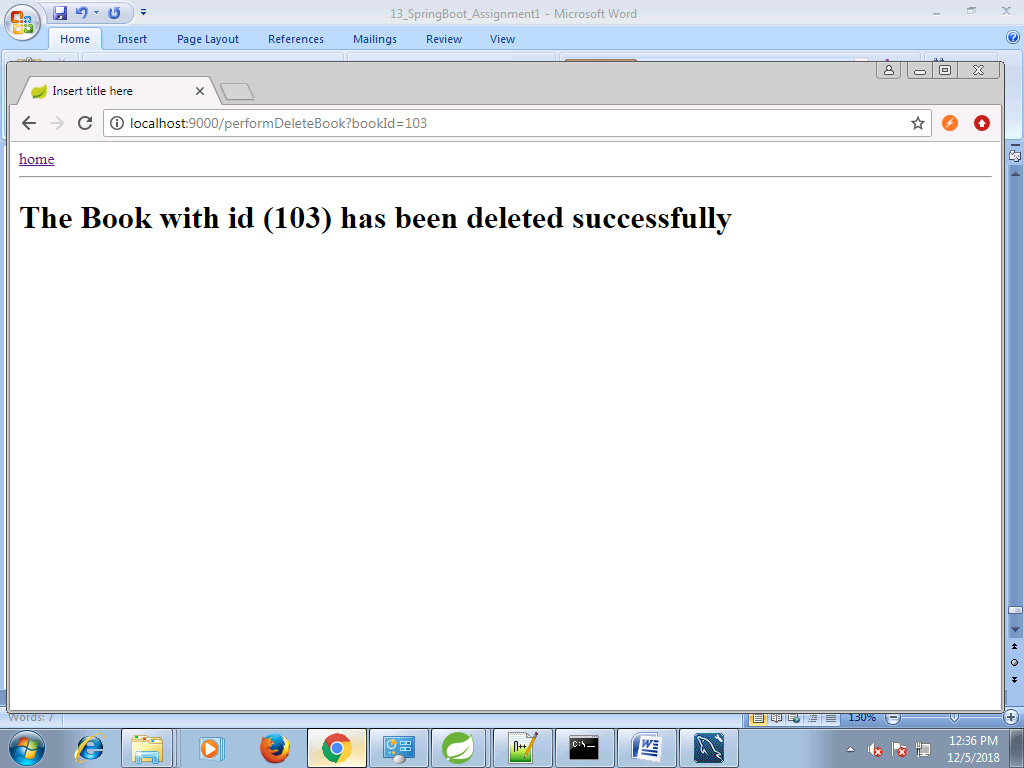


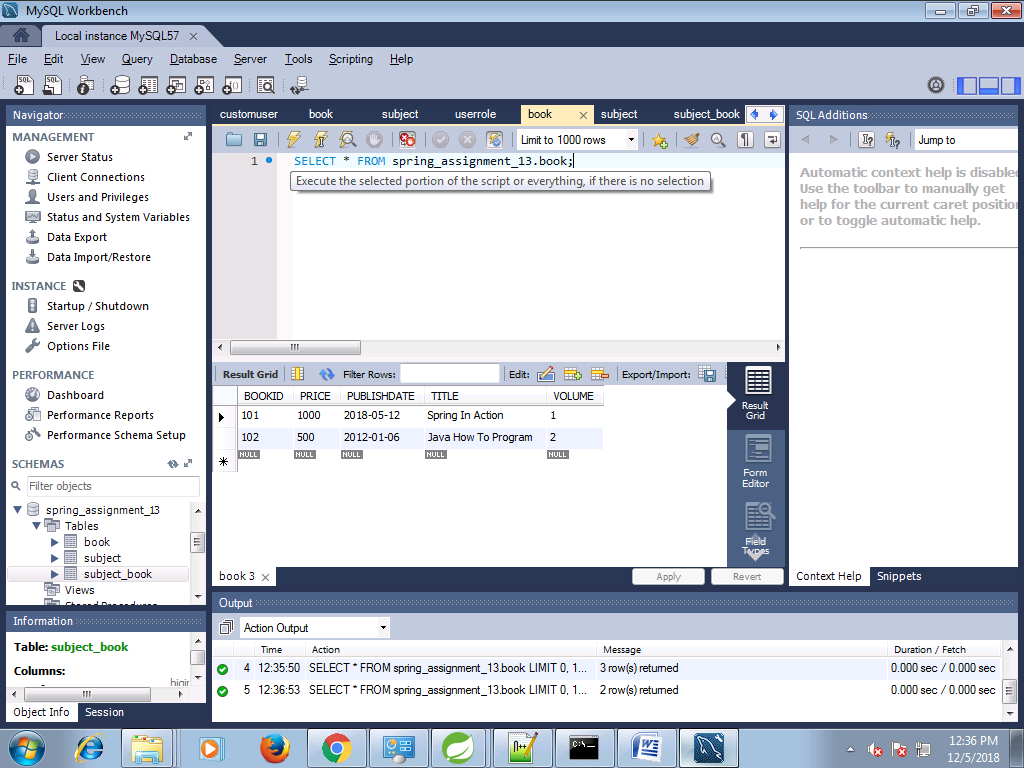


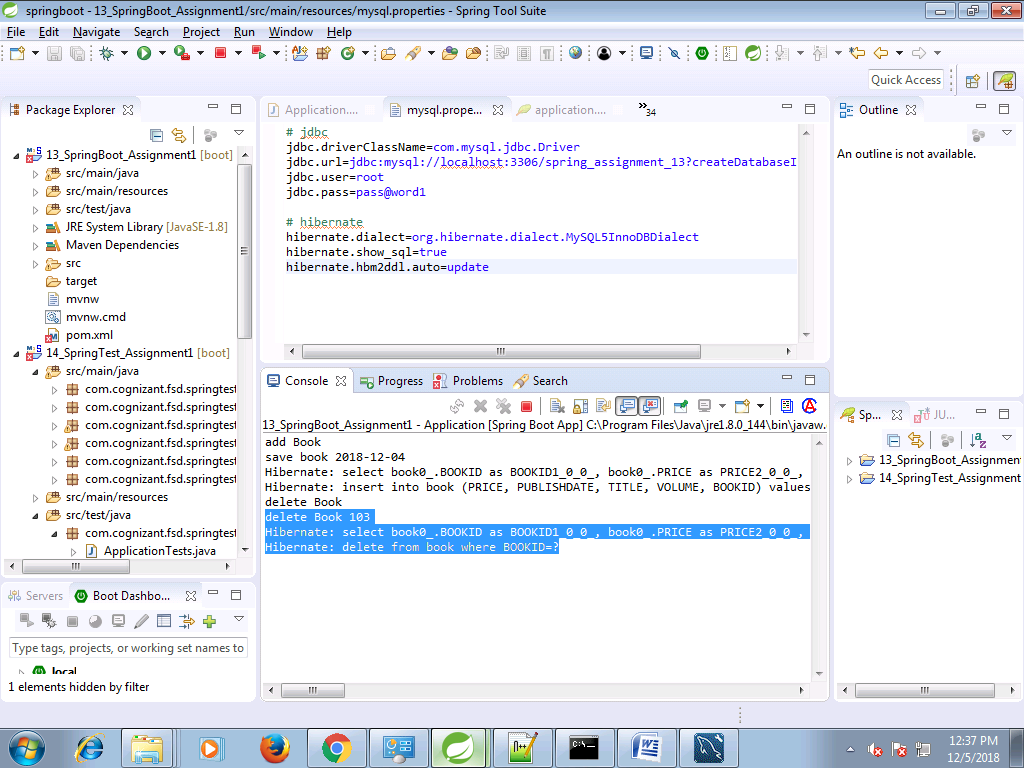


[Delete Book](http://localhost:9000/deleteBook)

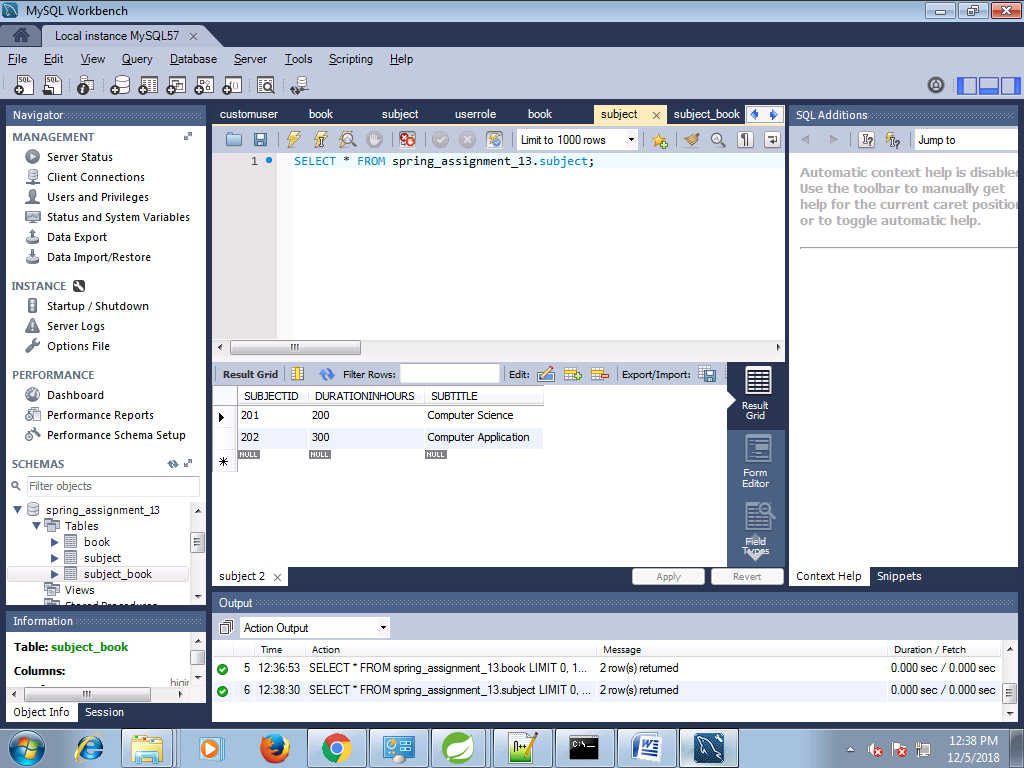


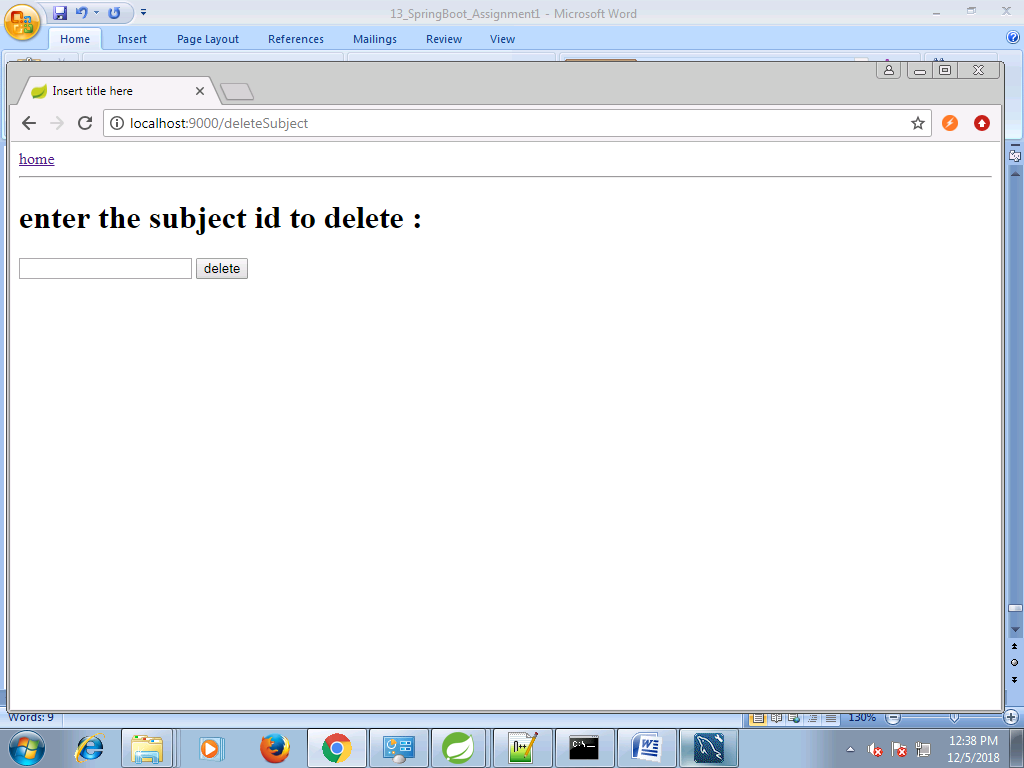


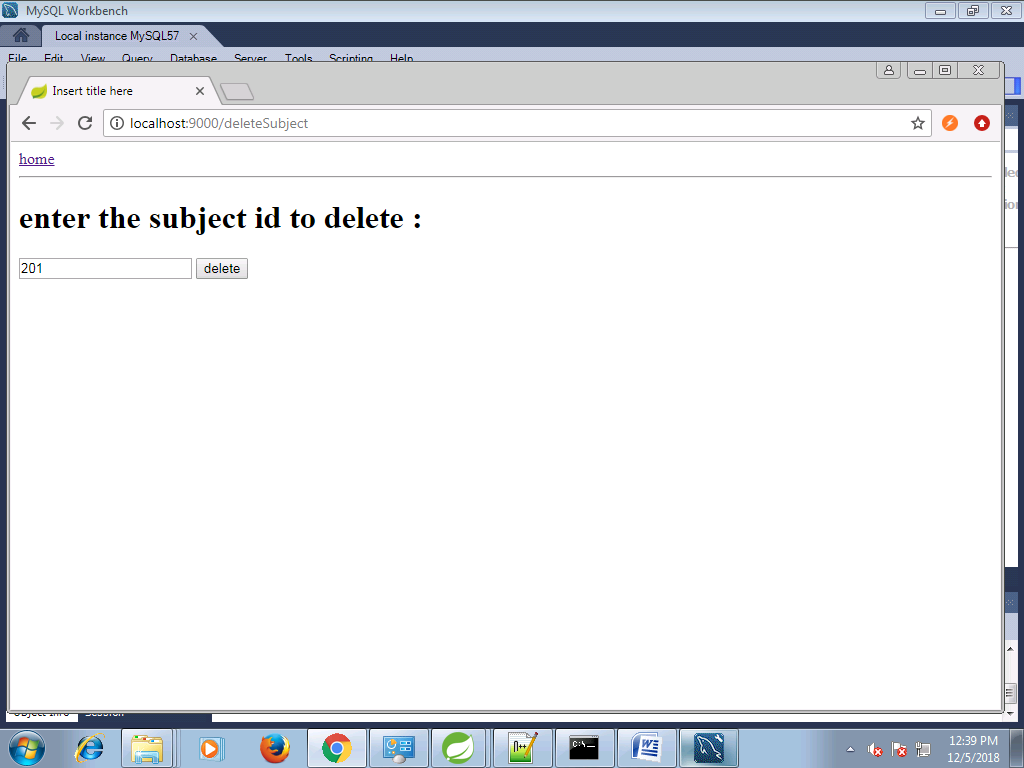


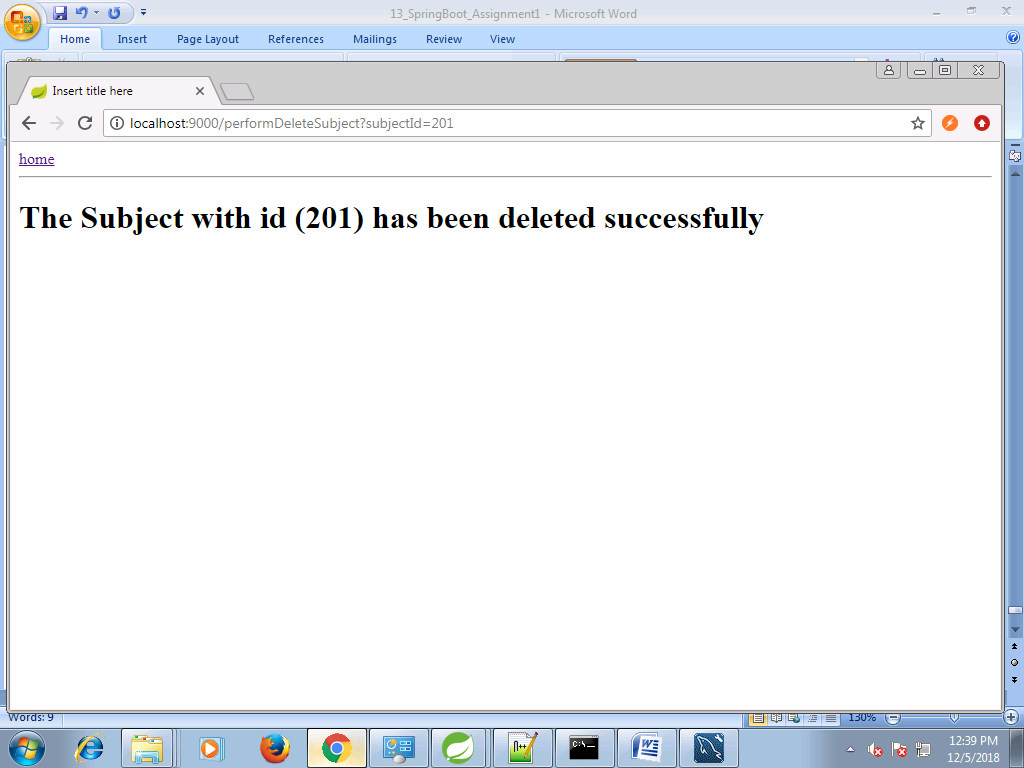


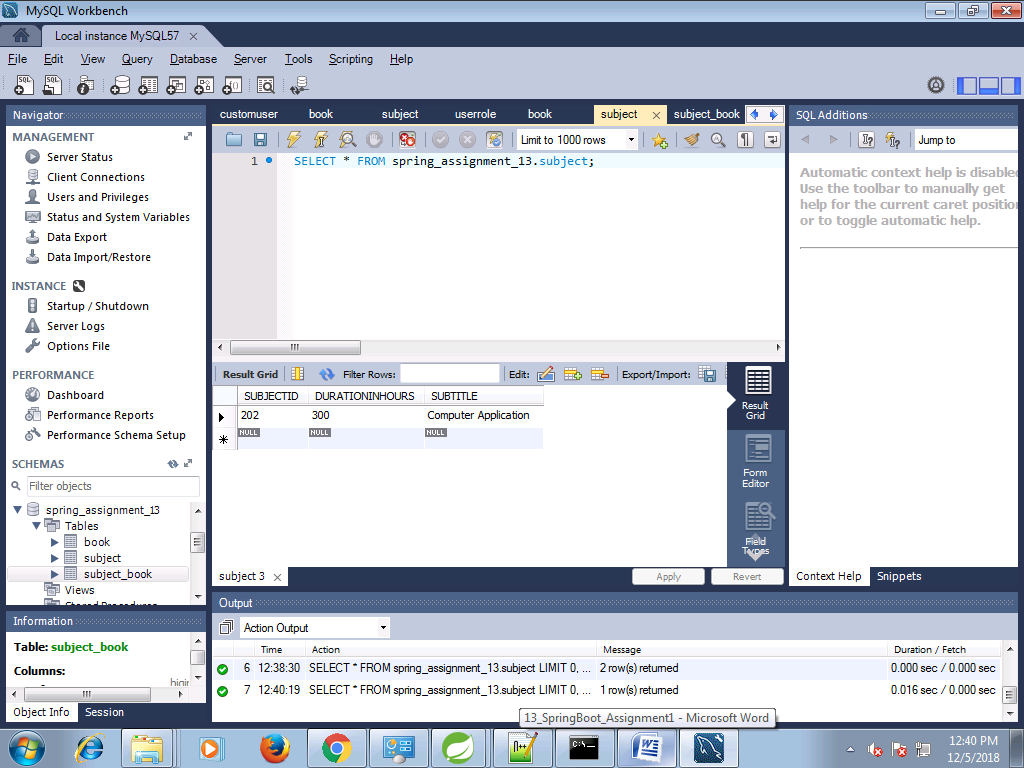
[Delete Subject](http://localhost:9000/deleteSubject)

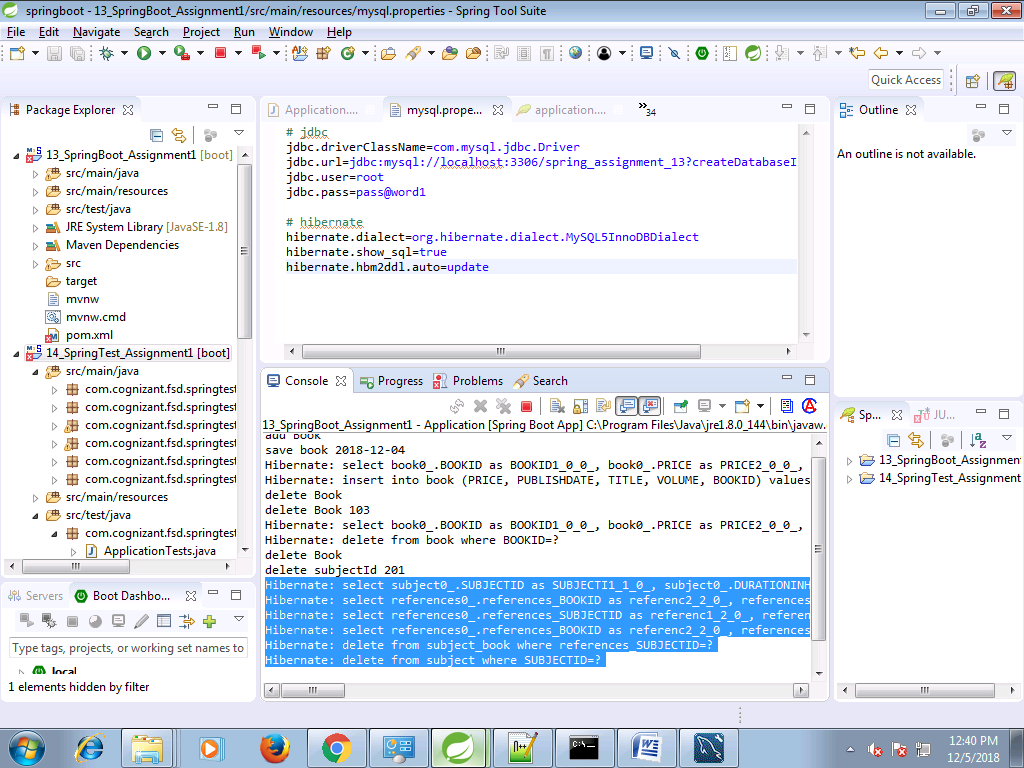




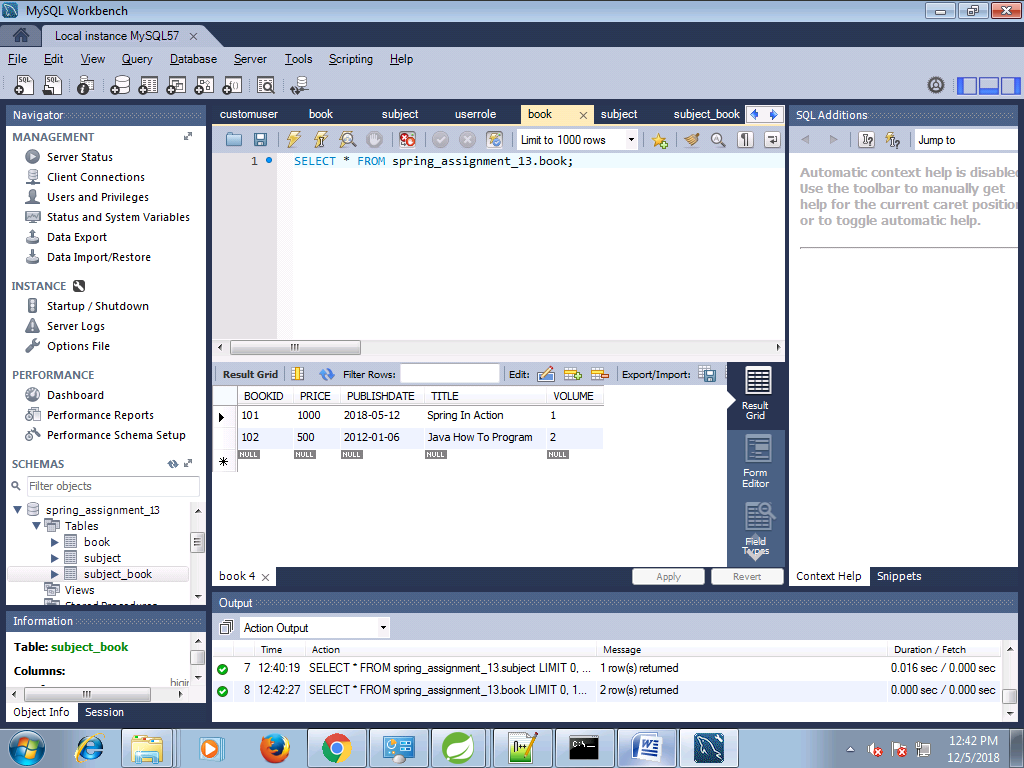


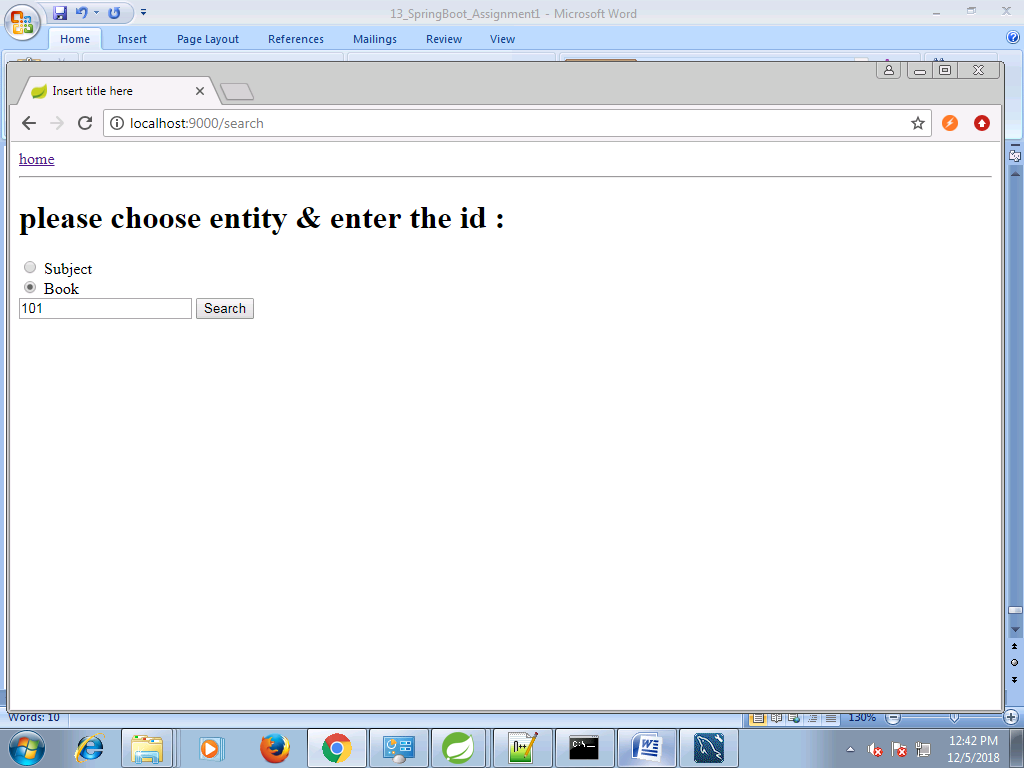


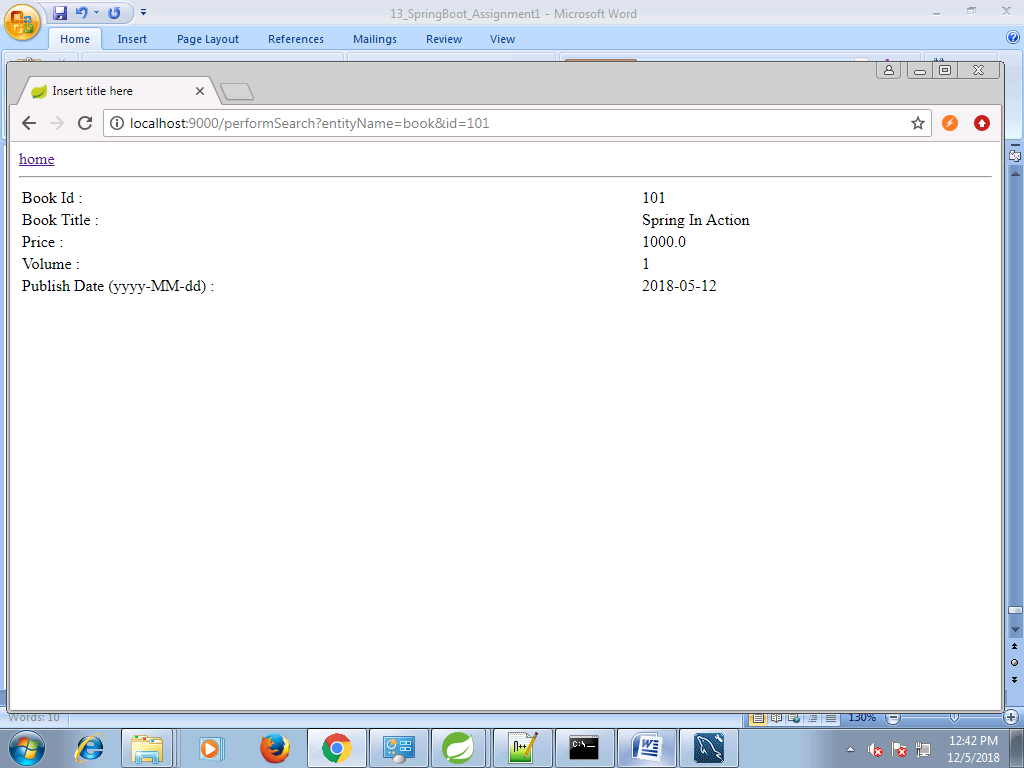




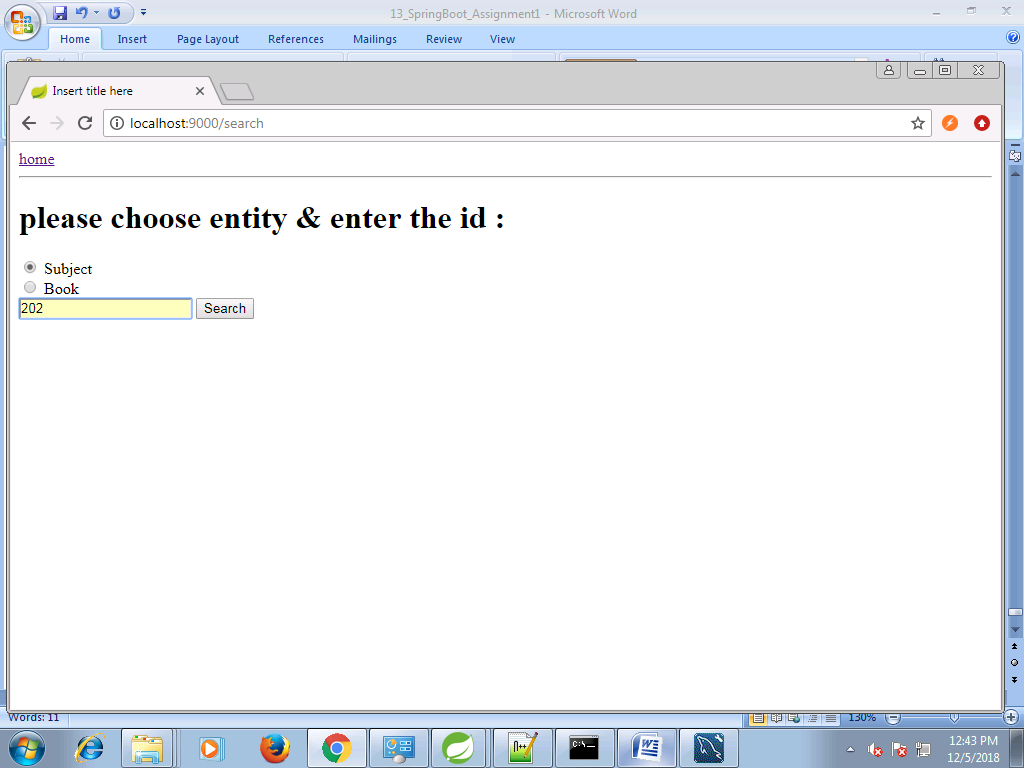
[Search](http://localhost:9000/search)

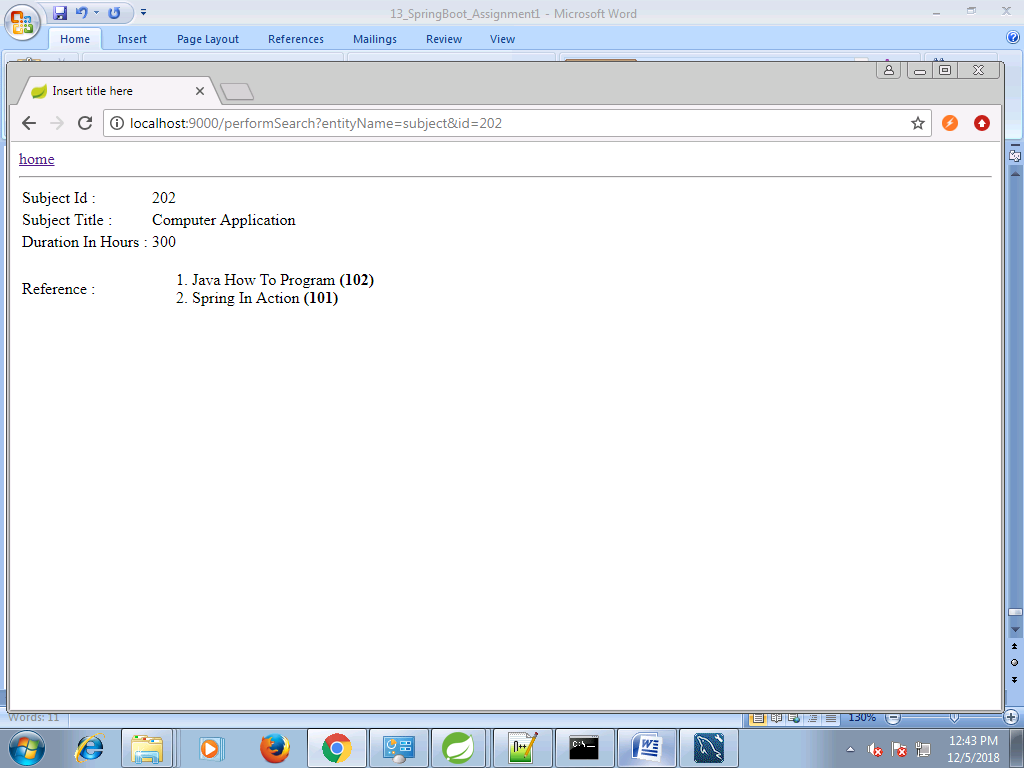






[Search](http://localhost:9000/search)





[Exit](http://localhost:9000/exit)

