# AppMain.java

* package com.cognizant.fsd.spring.config;
* import java.util.HashSet;
* import java.util.Set;
* import org.hibernate.Session;
* import org.hibernate.SessionFactory;
* import org.springframework.context.annotation.AnnotationConfigApplicationContext;
* import org.springframework.context.support.AbstractApplicationContext;
* import com.cognizant.fsd.spring.model.Book;
* import com.cognizant.fsd.spring.model.Subject;
* import com.cognizant.fsd.spring.model.UserRole;
* public class AppMain {
* public static void main(String args[]) {
* AbstractApplicationContext context = new AnnotationConfigApplicationContext(HibernateConfig.class);
* SessionFactory sessionFactory=(SessionFactory) context.getBean("sessionFactory");
* UserRole r1 = new UserRole();
* r1.setRoleId(201);
* r1.setRoleName("LIBRARIAN");
* UserRole r2 = new UserRole();
* r2.setRoleId(202);
* r2.setRoleName("PRINCIPAL");
* Session session=sessionFactory.openSession();
* session.beginTransaction();
* session.save(r1);
* session.save(r2);
* session.getTransaction().commit();
* /\* Subject subject=new Subject();
* Book b1=(Book) sessionFactory.openSession().get(Book.class, 101l);
* Book b2=(Book) sessionFactory.openSession().get(Book.class, 102l);
* System.out.println(b1 + " --- "+b2);
* Set<Book> set=new HashSet<Book>();
* set.add(b1);
* set.add(b2);
* subject.setSubjectId(203l);
* subject.setSubTitle("subTitle3");
* subject.setDurationInHours(2);
* subject.setReferences(set);
* Session session=sessionFactory.openSession();
* session.beginTransaction();
* session.save(subject);
* session.getTransaction().commit();
* System.out.println(context.getBean("sessionFactory"));\*/
* }
* }

# HibernateConfig.java

* package com.cognizant.fsd.spring.config;
* import java.util.Properties;
* import javax.sql.DataSource;
* import org.apache.tomcat.dbcp.dbcp.BasicDataSource;
* import org.hibernate.SessionFactory;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.context.annotation.Bean;
* import org.springframework.context.annotation.ComponentScan;
* import org.springframework.context.annotation.Configuration;
* import org.springframework.context.annotation.PropertySource;
* import org.springframework.core.env.Environment;
* import org.springframework.orm.hibernate4.HibernateTransactionManager;
* import org.springframework.orm.hibernate4.LocalSessionFactoryBean;
* import org.springframework.transaction.annotation.EnableTransactionManagement;
* @Configuration
* @EnableTransactionManagement
* @PropertySource({ "classpath:mysql.properties" })
* public class HibernateConfig {
* @Autowired
* private Environment env;
* @Bean
* public LocalSessionFactoryBean sessionFactory() {
* LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();
* sessionFactory.setDataSource(dataSource());
* sessionFactory.setPackagesToScan(new String[] { "com.cognizant.fsd.spring.model" });
* sessionFactory.setHibernateProperties(hibernateProperties());
* return sessionFactory;
* }
* @Bean
* public DataSource dataSource() {
* BasicDataSource dataSource = new BasicDataSource();
* dataSource.setDriverClassName(env.getProperty("jdbc.driverClassName"));
* dataSource.setUrl(env.getProperty("jdbc.url"));
* dataSource.setUsername(env.getProperty("jdbc.user"));
* dataSource.setPassword(env.getProperty("jdbc.pass"));
* return dataSource;
* }
* Properties hibernateProperties() {
* return new Properties() {
* {
* setProperty("hibernate.hbm2ddl.auto",
* env.getProperty("hibernate.hbm2ddl.auto"));
* setProperty("hibernate.dialect",
* env.getProperty("hibernate.dialect"));
* setProperty("hibernate.globally\_quoted\_identifiers",
* "true");
* setProperty("hibernate.show\_sql",
* "true");
* }
* };
* }
* @Bean
* @Autowired
* public HibernateTransactionManager transactionManager(SessionFactory s) {
* HibernateTransactionManager txManager = new HibernateTransactionManager();
* txManager.setSessionFactory(s);
* return txManager;
* }
* }

# MyWebAppInitializer.java

* **package** com.cognizant.fsd.spring.config;
* **import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
* /\*\*
* \* **@author** imssbora
* \*/
* **public** **class** MyWebAppInitializer **extends**
* AbstractAnnotationConfigDispatcherServletInitializer{
* @Override
* **protected** Class<?>[] getRootConfigClasses() {
* **return** **new** Class[]{WebConfig.**class**,RootConfig.**class**};
* }
* @Override
* **protected** Class<?>[] getServletConfigClasses() {
* **return** **null**;
* }
* @Override
* **protected** String[] getServletMappings() {
* **return** **new** String[]{"/"};
* }
* }

# RootConfig.java

* package com.cognizant.fsd.spring.config;
* import org.springframework.beans.factory.InitializingBean;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.context.annotation.Bean;
* import org.springframework.context.annotation.Configuration;
* import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
* import org.springframework.security.core.userdetails.User;
* import com.cognizant.fsd.spring.datastream.BinaryBookStreamCRUD;
* import com.cognizant.fsd.spring.datastream.BinarySubjectStreamCRUD;
* import com.cognizant.fsd.spring.service.UserService;
* @Configuration
* public class RootConfig implements InitializingBean{
* @Bean
* BinarySubjectStreamCRUD binarySubjectStreamCRUD() {
* return new BinarySubjectStreamCRUD();
* }
* @Bean
* BinaryBookStreamCRUD binaryBookStreamCRUD() {
* return new BinaryBookStreamCRUD();
* }
* @Override
* public void afterPropertiesSet() throws Exception {
* }
* }

# SecurityConfiguration.java

**package** com.cognizant.fsd.spring.config;

**import** org.springframework.beans.factory.InitializingBean;

/\*\*

\* **@author** Sudhir Kumar Thakur

\*

\*/

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

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

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

**import** org.springframework.security.config.annotation.web.builders.HttpSecurity;

**import** org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

**import** org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

**import** org.springframework.security.core.userdetails.UserDetailsService;

@Configuration

@EnableWebSecurity

**public** **class** SecurityConfiguration **extends** WebSecurityConfigurerAdapter **implements** InitializingBean{

@Autowired

@Qualifier("fsdUserDetailsService")

UserDetailsService userDetailsService;

/\* @Autowired

public void configureGlobalSecurity(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication().withUser("bill").password("abc123").roles("USER");

auth.inMemoryAuthentication().withUser("admin").password("root123").roles("ADMIN");

auth.inMemoryAuthentication().withUser("dba").password("root123").roles("ADMIN","DBA");//dba have two roles.

}\*/

@Autowired

**public** **void** configureGlobalSecurity(AuthenticationManagerBuilder auth) **throws** Exception {

auth.userDetailsService(userDetailsService);

}

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

.antMatchers("/", "/registration").permitAll()

.antMatchers("/librarian/\*\*").access("hasRole('LIBRARIAN')")

.antMatchers("/principal/\*\*").access("hasRole('PRINCIPAL')")

.and().formLogin()

.and().exceptionHandling().accessDeniedPage("/access\_denied");

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

System.***out***.println("aaaaa afterPropertiesSet...."+userDetailsService);

}

}

# SecurityWebApplicationInitializer.java

* **package** com.cognizant.fsd.spring.config;
* **import** org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer;
* **public** **class** SecurityWebApplicationInitializer
* **extends** AbstractSecurityWebApplicationInitializer {
* }

# WebConfig.java

* package com.cognizant.fsd.spring.config;
* import org.springframework.context.annotation.Bean;
* import org.springframework.context.annotation.ComponentScan;
* import org.springframework.context.annotation.Configuration;
* import org.springframework.context.annotation.Import;
* import org.springframework.web.context.request.RequestContextListener;
* import org.springframework.web.servlet.config.annotation.EnableWebMvc;
* import org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;
* import org.springframework.web.servlet.view.InternalResourceViewResolver;
* import org.springframework.web.servlet.view.JstlView;
* @EnableWebMvc
* @Configuration
* @ComponentScan(basePackages = { "com.cognizant.fsd.spring.config","com.cognizant.fsd.spring.controller","com.cognizant.fsd.spring.service","com.cognizant.fsd.spring.repository" })
* @Import(value = { SecurityConfiguration.class })
* public class WebConfig extends WebMvcConfigurerAdapter {
* @Bean
* public InternalResourceViewResolver resolver() {
* InternalResourceViewResolver resolver = new InternalResourceViewResolver();
* resolver.setViewClass(JstlView.class);
* resolver.setPrefix("/WEB-INF/views/");
* resolver.setSuffix(".jsp");
* return resolver;
* }
* @Bean
* public RequestContextListener requestContextListener() {
* return new RequestContextListener();
* }
* }

# Answer.java

* **package** com.cognizant.fsd.spring.controller;
* **public** **class** Answer {
* **private** String ans;
* **public** String getAns() {
* **return** ans;
* }
* **public** **void** setAns(String ans) {
* **this**.ans = ans;
* }
* }

# FsdCollectionEditor.java

* package com.cognizant.fsd.spring.controller;
* import java.util.Collection;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.beans.propertyeditors.CustomCollectionEditor;
* import com.cognizant.fsd.spring.model.Book;
* import com.cognizant.fsd.spring.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;
* }
* }

# GlobalExceptionHandler.java

* package com.cognizant.fsd.spring.controller;
* import javax.servlet.http.HttpServletRequest;
* import org.springframework.web.bind.annotation.ControllerAdvice;
* import org.springframework.web.bind.annotation.ExceptionHandler;
* @ControllerAdvice
* public class GlobalExceptionHandler {
* @ExceptionHandler(Exception.class)
* public String handleSQLException(HttpServletRequest request, Exception ex){
* System.err.println("global exception : "+ex);
* return "error";
* }
* }

# HomeController.java

* package com.cognizant.fsd.spring.controller;
* import java.util.List;
* import java.util.Set;
* import javax.servlet.http.HttpServletRequest;
* import javax.servlet.http.HttpServletResponse;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.security.core.Authentication;
* import org.springframework.security.core.context.SecurityContextHolder;
* import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;
* 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.spring.model.Book;
* import com.cognizant.fsd.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.Subject;
* import com.cognizant.fsd.spring.model.UserRole;
* import com.cognizant.fsd.spring.service.BookService;
* import com.cognizant.fsd.spring.service.SubjectService;
* import com.cognizant.fsd.spring.service.UserService;
* @Controller
* public class HomeController {
* @Autowired
* private SubjectService subjectService;
* @Autowired
* private BookService bookService;
* @Autowired
* private UserService userService;
* @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) {
* setUserName(model);
* return "index";
* }
* @RequestMapping(path = { "/login" }, method = RequestMethod.GET)
* public String login(Model model) {
* setUserName(model);
* return "login";
* }
* @RequestMapping(value="/logout", method = RequestMethod.GET)
* public String logoutPage (HttpServletRequest request, HttpServletResponse response) {
* Authentication auth = SecurityContextHolder.getContext().getAuthentication();
* if (auth != null){
* new SecurityContextLogoutHandler().logout(request, response, auth);
* }
* return "redirect:/login?logout";
* }
* @RequestMapping(path = { "/registration" }, method = RequestMethod.GET)
* public String registration(Model model) {
* List<UserRole> userRoles = userService.getRoleList();
* model.addAttribute("customUser", new CustomUser());
* model.addAttribute("ROLE\_LIST", userRoles);
* return "registration";
* }
* @RequestMapping(value = "/saveUser", method = RequestMethod.POST)
* public String saveUser(@ModelAttribute final CustomUser customUser, Model model) {
* System.out.println("save customUser ");
* userService.addCustomUser(customUser);
* model.addAttribute("ENTITY\_NAME", "User");
* model.addAttribute("ENTITY\_ID", customUser.getId());
* return "success";
* }
* @RequestMapping(value = "/principal/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 = "/librarian/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 = "/librarian/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 = "/principal/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");
* MyCommand myCommand=new MyCommand();
* model.addAttribute("myCommand", myCommand);
* return "exit";
* }
* @RequestMapping(value = "/saveMyCommand", method = RequestMethod.POST)
* public String saveMyCommand(@ModelAttribute MyCommand myCommand, Model model) {
* System.out.println("save saveMyCommand ");
* String s=myCommand.getSectionList().get(0).getQuestionList().get(0).getAnswerList().get(0).getAns();
* myCommand.getSectionList().get(0).getQuestionList().get(0).getAnswerList().get(0).setAns(s.concat(" from server"));
* System.out.println("save ans " +s);
* model.addAttribute("myCommand", myCommand);
* return "exit";
* }
* @RequestMapping(value = "/access\_denied", method = RequestMethod.GET)
* public String accessDenied(Model model) {
* System.out.println("access\_denied");
* return "access\_denied";
* }
* public void setUserName(Model model) {
* Authentication auth = SecurityContextHolder.getContext().getAuthentication();
* String name = auth.getName(); //get logged in username
* model.addAttribute("username", name);
* }
* }

# MyCommand.java

* package com.cognizant.fsd.spring.controller;
* import java.util.ArrayList;
* import java.util.List;
* public class MyCommand {
* private List<Section> sectionList=new ArrayList<Section>();
* public List<Section> getSectionList() {
* return sectionList;
* }
* public void setSectionList(List<Section> sectionList) {
* this.sectionList = sectionList;
* }
* }

# Question.java

* package com.cognizant.fsd.spring.controller;
* import java.util.ArrayList;
* import java.util.List;
* public class Question {
* private List<Answer> answerList=new ArrayList<Answer>();
* public List<Answer> getAnswerList() {
* return answerList;
* }
* public void setAnswerList(List<Answer> answerList) {
* this.answerList = answerList;
* }
* }

# Section.java

* package com.cognizant.fsd.spring.controller;
* import java.util.ArrayList;
* import java.util.List;
* public class Section {
* private String name;
* private List<Question> questionList=new ArrayList<Question>();
* public List<Question> getQuestionList() {
* return questionList;
* }
* public void setQuestionList(List<Question> questionList) {
* this.questionList = questionList;
* }
* public String getName() {
* return name;
* }
* public void setName(String name) {
* this.name = name;
* }
* }

# Book.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.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 Sudhir Kumar Thakur
* \*
* \*/
* @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 + "]";
* }
* }

# CustomUser.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.model;
* import java.util.HashSet;
* import java.util.List;
* import java.util.Set;
* import javax.persistence.Entity;
* import javax.persistence.FetchType;
* import javax.persistence.GeneratedValue;
* import javax.persistence.GenerationType;
* import javax.persistence.Id;
* import javax.persistence.OneToMany;
* import javax.persistence.Table;
* import javax.persistence.Transient;
* /\*\*
* \* @author Admin
* \*
* \*/
* @Entity
* @Table(name="customUser")
* public class CustomUser {
* @Id
* @GeneratedValue(strategy=GenerationType.AUTO)
* private long id;
* private String firstName;
* private String lastName;
* private String username;
* private String password;
* @Transient
* private List<Long> roleList;
* @OneToMany(fetch=FetchType.EAGER)
* private Set<UserRole> roles=new HashSet<UserRole>();
* public long getId() {
* return id;
* }
* public void setId(long id) {
* this.id = id;
* }
* public String getFirstName() {
* return firstName;
* }
* public void setFirstName(String firstName) {
* this.firstName = firstName;
* }
* public String getLastName() {
* return lastName;
* }
* public void setLastName(String lastName) {
* this.lastName = lastName;
* }
* public String getUsername() {
* return username;
* }
* public void setUsername(String username) {
* this.username = username;
* }
* public String getPassword() {
* return password;
* }
* public void setPassword(String password) {
* this.password = password;
* }
* public Set<UserRole> getRoles() {
* return roles;
* }
* public void setRoles(Set<UserRole> roles) {
* this.roles = roles;
* }
* public List<Long> getRoleList() {
* return roleList;
* }
* public void setRoleList(List<Long> roleList) {
* this.roleList = roleList;
* }
* }

# FsdUserPrincipal.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.model;
* import java.util.ArrayList;
* import java.util.Collection;
* import java.util.List;
* import org.springframework.security.core.GrantedAuthority;
* import org.springframework.security.core.authority.SimpleGrantedAuthority;
* import org.springframework.security.core.userdetails.User;
* import org.springframework.security.core.userdetails.UserDetails;
* /\*\*
* \* @author Admin
* \*
* \*/
* public class FsdUserPrincipal implements UserDetails {
* private CustomUser customUser;
* public FsdUserPrincipal(CustomUser customUser) {
* this.customUser=customUser;
* getGrantedAuthorities();
* }
* private List<GrantedAuthority> getGrantedAuthorities(){
* List<GrantedAuthority> authorities = new ArrayList<GrantedAuthority>();
* customUser.getRoles().forEach(role->{
* authorities.add(new SimpleGrantedAuthority("ROLE\_"+role.getRoleName()));
* });
* return authorities;
* }
* @Override
* public Collection<? extends GrantedAuthority> getAuthorities() {
* return getGrantedAuthorities();
* }
* @Override
* public String getPassword() {
* // TODO Auto-generated method stub
* return customUser.getPassword();
* }
* @Override
* public String getUsername() {
* // TODO Auto-generated method stub
* return customUser.getUsername();
* }
* @Override
* public boolean isAccountNonExpired() {
* // TODO Auto-generated method stub
* return true;
* }
* @Override
* public boolean isAccountNonLocked() {
* // TODO Auto-generated method stub
* return true;
* }
* @Override
* public boolean isCredentialsNonExpired() {
* // TODO Auto-generated method stub
* return true;
* }
* @Override
* public boolean isEnabled() {
* // TODO Auto-generated method stub
* return true;
* }
* public CustomUser getCustomUser() {
* return customUser;
* }
* public void setCustomUser(CustomUser customUser) {
* this.customUser = customUser;
* }
* }

# Subject.java

* package com.cognizant.fsd.spring.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 Sudhir Kumar Thakur
* \*
* \*/
* @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;
* }
* }

# UserRole.java

* package com.cognizant.fsd.spring.model;
* import javax.persistence.Entity;
* import javax.persistence.GeneratedValue;
* import javax.persistence.GenerationType;
* import javax.persistence.Id;
* import javax.persistence.Table;
* @Entity
* @Table(name="userRole")
* public class UserRole {
* @Id
* //@GeneratedValue(strategy=GenerationType.AUTO)
* private long roleId;
* private String roleName;
* public long getRoleId() {
* return roleId;
* }
* public void setRoleId(long roleId) {
* this.roleId = roleId;
* }
* public String getRoleName() {
* return roleName;
* }
* public void setRoleName(String roleName) {
* this.roleName = roleName;
* }
* }

# AbstractDao.java

* package com.cognizant.fsd.spring.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);
* }
* }

# BookRepository.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import com.cognizant.fsd.spring.model.Book;
* public interface BookRepository {
* public Book addBook(Book book);
* public boolean deleteBook(long bookId);
* public Book searchBook(long bookId);
* public List<Book> fetchAllBook();
* }

# BookRepositoryImpl.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import org.hibernate.Criteria;
* import org.hibernate.criterion.Restrictions;
* import org.springframework.stereotype.Repository;
* import com.cognizant.fsd.spring.model.Book;
* @Repository("bookRepository")
* public class BookRepositoryImpl extends AbstractDao implements BookRepository {
* @Override
* public Book addBook(Book book) {
* super.persist(book);
* return book;
* }
* @Override
* public boolean deleteBook(long bookId) {
* boolean flag = false;
* try {
* Book book = (Book) getSession().get(Book.class, bookId);
* super.delete(book);
* flag = true;
* } catch (Exception e) {
* flag = false;
* }
* return flag;
* }
* @Override
* public Book searchBook(long bookId) {
* Criteria criteria = getSession().createCriteria(Book.class, "BK");
* criteria.add(Restrictions.eq("BK.bookId", bookId));
* Object book = (Object) criteria.uniqueResult();
* return book != null ? (Book) book : null;
* }
* @Override
* public List<Book> fetchAllBook() {
* Criteria criteria = getSession().createCriteria(Book.class, "BK");
* List<Book> bookList = criteria.list();
* return bookList;
* }
* }

# SubjectRepository.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import com.cognizant.fsd.spring.model.Subject;
* public interface SubjectRepository {
* public Subject addSubject(Subject subject);
* public boolean deleteSubject(long subjectId);
* public Subject searchSubject(long subjectId);
* public List<Subject> fetchAllSubject();
* }

# SubjectRepositoryImpl.java

* package com.cognizant.fsd.spring.repository;
* import java.io.DataInputStream;
* import java.io.DataOutputStream;
* import java.io.FileInputStream;
* import java.io.FileOutputStream;
* import java.io.IOException;
* import java.util.ArrayList;
* import java.util.List;
* import org.hibernate.Criteria;
* import org.hibernate.criterion.Restrictions;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.stereotype.Repository;
* import com.cognizant.fsd.spring.datastream.BinarySubjectStreamCRUD;
* import com.cognizant.fsd.spring.model.Book;
* import com.cognizant.fsd.spring.model.Subject;
* @Repository("subjectRepository")
* public class SubjectRepositoryImpl extends AbstractDao implements SubjectRepository{
* @Override
* public Subject addSubject(Subject subject) {
* super.persist(subject);
* return subject;
* }
* @Override
* public boolean deleteSubject(long subjectId) {
* boolean flag=false;
* try {
* Subject subject=(Subject) getSession().load(Subject.class, subjectId);
* super.delete(subject);
* flag=true;
* }catch(Exception e) {
* flag=false;
* }
* return flag;
* }
* @Override
* public Subject searchSubject(long subjectId) {
* Object subject=getSession().get(Subject.class, subjectId);
* return subject!=null?(Subject)subject:null;
* }
* @Override
* public List<Subject> fetchAllSubject() {
* Criteria criteria= getSession().createCriteria(Subject.class,"SUB");
* List<Subject> subjectList=criteria.list();
* return subjectList;
* }
* }

# UserRepository.java

* package com.cognizant.fsd.spring.repository;
* import java.util.List;
* import com.cognizant.fsd.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.UserRole;
* public interface UserRepository {
* public CustomUser searchUser(long userId);
* public CustomUser addCustomUser(CustomUser customUser);
* public CustomUser loadUserByUsername(String username);
* public List<UserRole> getRoleList();
* public List<CustomUser> getCustomUser();
* }

# UserRepositoryImpl.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.repository;
* import java.util.HashSet;
* import java.util.List;
* import java.util.Set;
* import org.hibernate.Criteria;
* import org.hibernate.Session;
* import org.hibernate.criterion.Restrictions;
* import org.springframework.core.env.SystemEnvironmentPropertySource;
* import org.springframework.stereotype.Repository;
* import com.cognizant.fsd.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.UserRole;
* /\*\*
* \* @author Admin
* \*
* \*/
* @Repository("userRepository")
* public class UserRepositoryImpl extends AbstractDao implements UserRepository{
* @Override
* public CustomUser searchUser(long id) {
* CustomUser customUser=null;
* try {
* customUser = (CustomUser) getSession().get(CustomUser.class, id);
* } catch (Exception e) {}
* return customUser;
* }
* @Override
* public CustomUser addCustomUser(CustomUser customUser) {
* Session session=getSession();
* Set<UserRole> roles=new HashSet<UserRole>();
* customUser.getRoleList().forEach(roleId->{
* UserRole userRole=(UserRole) session.get(UserRole.class, roleId);
* roles.add(userRole);
* });
* customUser.setRoles(roles);
* super.persist(customUser);
* return customUser;
* }
* @Override
* public CustomUser loadUserByUsername(String username) {
* System.out.println("ppppppppp : "+username);
* CustomUser customUser=null;
* try {
* Criteria criteria=getSession().createCriteria(CustomUser.class);
* criteria.add(Restrictions.eq("username", username));
* customUser = (CustomUser) criteria.uniqueResult();
* } catch (Exception e) {
* System.err.println("errrrrr "+e);
* }
* return customUser;
* }
* @Override
* public List<UserRole> getRoleList() {
* List<UserRole> userRoles=null;
* try {
* Criteria criteria=getSession().createCriteria(UserRole.class);
* userRoles = (List<UserRole>) criteria.list();
* } catch (Exception e) {}
* return userRoles;
* }

* public List<CustomUser> getCustomUser() {
* List<CustomUser> customUsers=null;
* try {
* Criteria criteria=getSession().createCriteria(CustomUser.class);
* customUsers = (List<CustomUser>) criteria.list();
* } catch (Exception e) {}
* return customUsers;
* }
* }

# BookService.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import com.cognizant.fsd.spring.model.Book;
* public interface BookService {
* public Book addBook(Book book);
* public boolean deleteBook(long bookId);
* public Book searchBook(long bookId);
* public List<Book> fetchAllBook();
* }

# BookServiceImpl.java

* package com.cognizant.fsd.spring.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.spring.model.Book;
* import com.cognizant.fsd.spring.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.addBook(book);
* }
* @Override
* public boolean deleteBook(long bookId) {
* return bookRepository.deleteBook(bookId);
* }
* @Override
* public Book searchBook(long bookId) {
* return bookRepository.searchBook(bookId);
* }
* @Override
* public List<Book> fetchAllBook() {
* return bookRepository.fetchAllBook();
* }
* }

# FsdSpringSecurityUserServiceImpl.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.service;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.security.core.userdetails.UserDetails;
* import org.springframework.security.core.userdetails.UserDetailsService;
* import org.springframework.security.core.userdetails.UsernameNotFoundException;
* import org.springframework.stereotype.Service;
* import com.cognizant.fsd.spring.repository.UserRepository;
* /\*\*
* \* @author Admin
* \*
* \*/
* public class FsdSpringSecurityUserServiceImpl{/\*
* @Autowired
* private UserService userService;
* @Override
* public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
* userService.loadUserByUsername(username)
* return null;
* }
* \*/}

# FsdUserDetailsService.java

* package com.cognizant.fsd.spring.service;
* import org.springframework.beans.factory.InitializingBean;
* import org.springframework.beans.factory.annotation.Autowired;
* import org.springframework.beans.factory.annotation.Qualifier;
* import org.springframework.context.ApplicationContext;
* import org.springframework.security.core.userdetails.UserDetails;
* import org.springframework.security.core.userdetails.UserDetailsService;
* import org.springframework.security.core.userdetails.UsernameNotFoundException;
* import org.springframework.stereotype.Service;
* import org.springframework.transaction.annotation.Transactional;
* import com.cognizant.fsd.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.FsdUserPrincipal;
* import com.cognizant.fsd.spring.repository.UserRepository;
* @Service("fsdUserDetailsService")
* //@Transactional
* public class FsdUserDetailsService implements UserDetailsService,InitializingBean{
* @Autowired
* @Qualifier("userService")
* private UserService userService;
* @Autowired
* private ApplicationContext applicationContext;
* @Override
* public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
* FsdUserPrincipal fsdUserPrincipal=null;
* System.out.println(userService+" username:: "+username);
* CustomUser customUser =userService.loadUserByUsername(username);
* System.out.println("username 2:: "+customUser);
* if(null!=customUser) {
* fsdUserPrincipal=new FsdUserPrincipal(customUser);
* }else {
* throw new UsernameNotFoundException("User not found by name: " + username);
* }
* return fsdUserPrincipal;
* }
* @Override
* public void afterPropertiesSet() throws Exception {
* System.out.println("fsdUserDetailsService......"+userService);
* }
* }

# SubjectService.java

* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import com.cognizant.fsd.spring.model.Subject;
* public interface SubjectService {
* public Subject addSubject(Subject subject);
* public boolean deleteSubject(long subjectId);
* public Subject searchSubject(long subjectId);
* public List<Subject> fetchAllSubject();
* }

# SubjectServiceImpl.java

* package com.cognizant.fsd.spring.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.spring.model.Subject;
* import com.cognizant.fsd.spring.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.addSubject(subject);
* }
* @Override
* public boolean deleteSubject(long subjectId) {
* return subjectRepository.deleteSubject(subjectId);
* }
* @Override
* public Subject searchSubject(long subjectId) {
* return subjectRepository.searchSubject(subjectId);
* }
* @Override
* public List<Subject> fetchAllSubject() {
* return subjectRepository.fetchAllSubject();
* }
* }

# UserService.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.service;
* import java.util.List;
* import com.cognizant.fsd.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.UserRole;
* /\*\*
* \* @author Admin
* \*
* \*/
* public interface UserService {
* public CustomUser searchUser(long userId);
* public CustomUser addCustomUser(CustomUser customUser);
* public CustomUser loadUserByUsername(String username);
* public List<UserRole> getRoleList();
* public List<CustomUser> getCustomUsers();
* }

# UserServiceImpl.java

* /\*\*
* \*
* \*/
* package com.cognizant.fsd.spring.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.spring.model.CustomUser;
* import com.cognizant.fsd.spring.model.UserRole;
* import com.cognizant.fsd.spring.repository.UserRepository;
* /\*\*
* \* @author Admin
* \*
* \*/
* @Service("userService")
* @Transactional
* public class UserServiceImpl implements UserService{
* @Autowired
* private UserRepository userRepository;
* @Override
* public CustomUser searchUser(long userId) {
* return userRepository.searchUser(userId);
* }
* @Override
* public CustomUser addCustomUser(CustomUser customUser) {
* return userRepository.addCustomUser(customUser);
* }
* @Override
* public CustomUser loadUserByUsername(String username) {
* return userRepository.loadUserByUsername(username);
* }
* @Override
* public List<UserRole> getRoleList() {
* return userRepository.getRoleList();
* }
* @Override
* public List<CustomUser> getCustomUsers() {
* return userRepository.getCustomUser();
* }
* }

# access\_denied.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>
* go to <a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>
* <h1>Access Denied</h1>
* </body>
* </html>

# 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>
* <%@include file=*"header.jsp"* %>
* <a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>
* <hr/>
* <table style="width: *100%*">
* <form:form method=*"POST"* commandName=*"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>
* <%@include file=*"header.jsp"* %>
* <a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>
* <hr/>
* <table style="width:*100%*">
* <form:form method=*"POST"* commandName=*"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>
* <%@include file=*"header.jsp"* %>
* <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>
* <%@include file=*"header.jsp"* %>
* <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>
* <%@include file=*"header.jsp"* %>
* <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>
* <%@include file=*"header.jsp"* %>
* <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=*"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>
* <%@include file=*"header.jsp"* %>
* <a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>
* <hr/>
* <h1>Thanks for using this application</h1>
* <form:form method=*"POST"* commandName=*"myCommand"* action=*"*${pageContext.request.contextPath}*/saveMyCommand"*>
* <form:input path=*"sectionList[0].questionList[0].answerList[0].ans"* />
* ha ha
* <input type=*"submit"* value=*"save"*/>
* </form:form>
* </body>
* </html>

# header.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=*"sec"* uri=*"http://www.springframework.org/security/tags"* %>
* <html>
* <head>
* <meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>
* <title>Insert title here</title>
* </head>
* <body>
* <sec:authorize access=*"!isAuthenticated()"*>
* <a href=*"*<spring:url value=*"/login"*/>*"*>Login</a><br/>
* </sec:authorize>
* <sec:authorize access=*"isAuthenticated()"*>
* <h3>Hi, <sec:authentication property=*"principal.username"* /> <a href=*"*<spring:url value=*"/logout"*/>*"*>Logout</a></h3><br/>
* </sec:authorize>
* </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"* %>
* <%@ taglib uri = *"http://java.sun.com/jsp/jstl/functions"* prefix = *"fn"* %>
* <!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>
* <%@include file=*"header.jsp"* %>
* <h1>${empty username || username ne fn:toLowerCase('anonymousUser') ? '' : 'Hi '.concat(username).concat(',') } Welcome to Spring Security</h1>
* <a href=*"*<spring:url value=*"/registration"*/>*"*>Registration</a><br/>
* <a href=*"*<spring:url value=*"/principal/addSubject"*/>*"*>Add Subject</a><br/>
* <a href=*"*<spring:url value=*"/librarian/addBook"*/>*"*>Add Book</a><br/>
* <a href=*"*<spring:url value=*"/principal/deleteSubject"*/>*"*>Delete Subject</a><br/>
* <a href=*"*<spring:url value=*"/librarian/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>

# registration.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"* commandName=*"customUser"* action=*"*${pageContext.request.contextPath}*/saveUser"* >
* <tr>
* <td><form:label path=*"firstName"*>First Name:</form:label></td>
* <td><form:input path=*"firstName"*/></td>
* </tr>
* <tr>
* <td><form:label path=*"lastName"*>Last Name :</form:label></td>
* <td><form:input path=*"lastName"*/></td>
* </tr>
* <tr>
* <td><form:label path=*"username"*>User Name :</form:label></td>
* <td><form:input path=*"username"*/></td>
* </tr>
* <tr>
* <td><form:label path=*"password"*>Password :</form:label></td>
* <td><form:input path=*"password"*/></td>
* </tr>
* <tr>
* <td><form:label path=*"roleList"*>Role List :</form:label></td>
* <td>
* <form:select path=*"roleList"*>
* <form:option value=*"0"* label=*"----- Select -----"*/>
* <form:options items=*"*${ROLE\_LIST}*"* itemLabel=*"roleName"* itemValue=*"roleId"*/>
* </form:select>
* </td>
* </tr>
* <tr>
* <td colspan=*"2"*><input type=*"submit"* value=*"save"*/></td>
* </tr>
* </form:form>
* </table>
* </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>
* <%@include file=*"header.jsp"* %>
* <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>
* <%@include file=*"header.jsp"* %>
* <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>
* <%@include file=*"header.jsp"* %>
* <a href=*"*<spring:url value=*"/"*/>*"*>home</a><br/>
* <hr/>
* <h1>The ${ENTITY\_NAME} with id (${ENTITY\_ID}) has been saved successfully</h1>
* </body>
* </html>

# mysql.properties

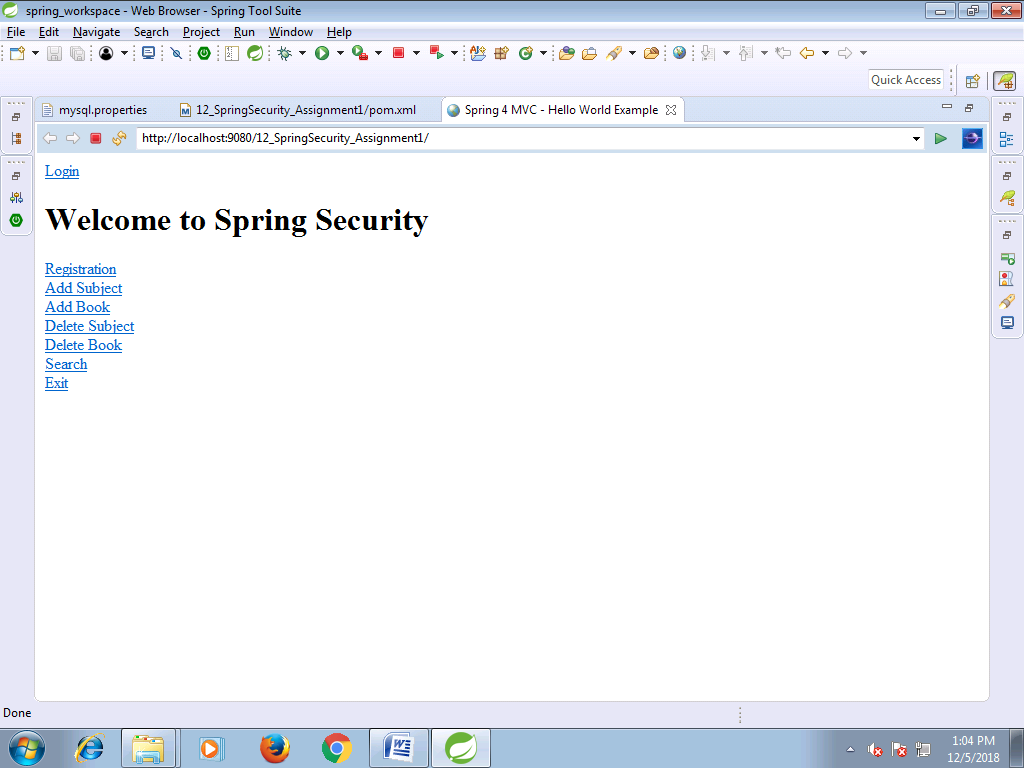
* # jdbc
* jdbc.driverClassName=com.mysql.jdbc.Driver
* jdbc.url=jdbc:mysql://localhost:3306/spring\_assignment\_12?createDatabaseIfNotExist=true
* jdbc.user=root
* jdbc.pass=pass@word1
* # hibernate
* hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
* hibernate.show\_sql=true
* hibernate.hbm2ddl.auto=updatev

# pom.xml

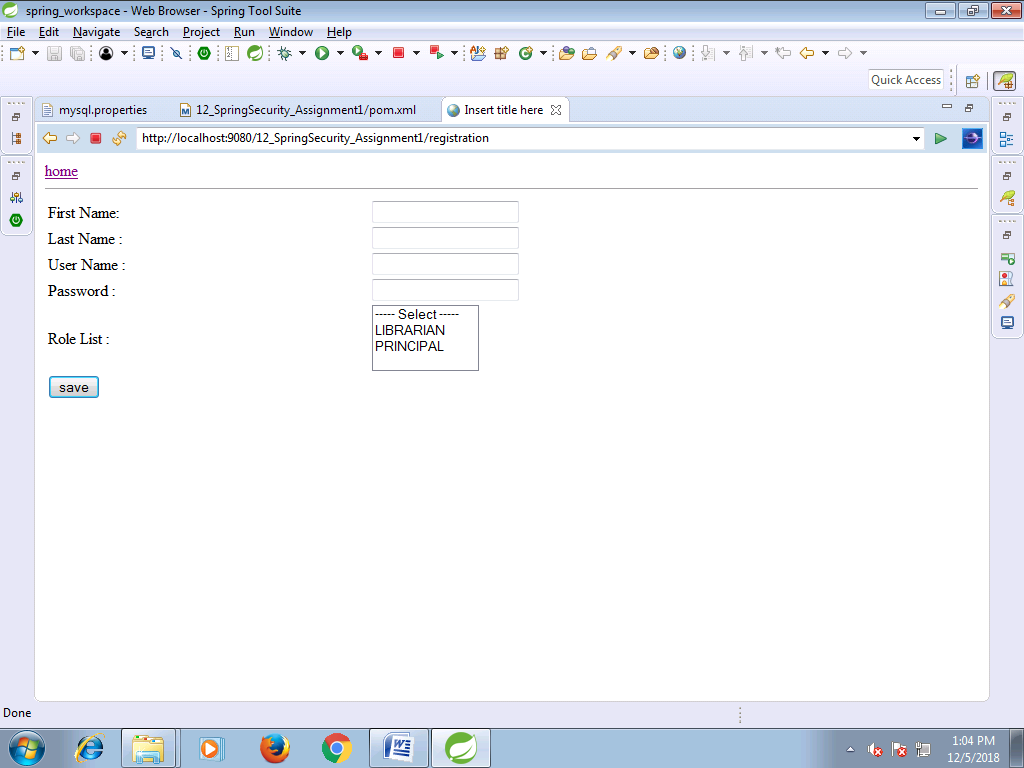
* <project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*
* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>
* <modelVersion>4.0.0</modelVersion>
* <groupId>com.cognizant.fsd.spring</groupId>
* <artifactId>12\_SpringSecurity\_Assignment1</artifactId>
* <version>0.0.1-SNAPSHOT</version>
* <packaging>war</packaging>
* <properties>
* <failOnMissingWebXml>false</failOnMissingWebXml>
* </properties>
* <dependencies>
* <!-- Spring MVC Dependency -->
* <dependency>
* <groupId>org.springframework</groupId>
* <artifactId>spring-webmvc</artifactId>
* <version>4.3.7.RELEASE</version>
* </dependency>
* <!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
* <dependency>
* <groupId>org.springframework</groupId>
* <artifactId>spring-orm</artifactId>
* <version>4.3.7.RELEASE</version>
* </dependency>
* <!-- Spring Security Core -->
* <dependency>
* <groupId>org.springframework.security</groupId>
* <artifactId>spring-security-core</artifactId>
* <version>4.0.1.RELEASE</version>
* </dependency>
* <!-- Spring Security Config -->
* <dependency>
* <groupId>org.springframework.security</groupId>
* <artifactId>spring-security-config</artifactId>
* <version>4.0.1.RELEASE</version>
* </dependency>
* <!-- Spring Security Web -->
* <dependency>
* <groupId>org.springframework.security</groupId>
* <artifactId>spring-security-web</artifactId>
* <version>4.0.1.RELEASE</version>
* </dependency>
* <!-- https://mvnrepository.com/artifact/org.springframework.security/spring-security-taglibs -->
* <dependency>
* <groupId>org.springframework.security</groupId>
* <artifactId>spring-security-taglibs</artifactId>
* <version>4.0.1.RELEASE</version>
* </dependency>
* <!-- JSTL Dependency-->
* <dependency>
* <groupId>javax.servlet.jsp.jstl</groupId>
* <artifactId>javax.servlet.jsp.jstl-api</artifactId>
* <version>1.2.1</version>
* </dependency>
* <dependency>
* <groupId>taglibs</groupId>
* <artifactId>standard</artifactId>
* <version>1.1.2</version>
* </dependency>
* <!-- Servlet Dependency -->
* <dependency>
* <groupId>javax.servlet</groupId>
* <artifactId>javax.servlet-api</artifactId>
* <version>3.1.0</version>
* <scope>provided</scope>
* </dependency>
* <!-- JSP Dependency -->
* <dependency>
* <groupId>javax.servlet.jsp</groupId>
* <artifactId>javax.servlet.jsp-api</artifactId>
* <version>2.3.1</version>
* <scope>provided</scope>
* </dependency>
* <!-- <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-core</artifactId>
* <version>5.2.2.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-entitymanager</artifactId>
* <version>5.2.2.Final</version>
* </dependency> -->
* <dependency>
* <groupId>org.apache.tomcat</groupId>
* <artifactId>tomcat-dbcp</artifactId>
* <version>7.0.55</version>
* </dependency>
* <dependency>
* <groupId>mysql</groupId>
* <artifactId>mysql-connector-java</artifactId>
* <version>5.1.32</version>
* <scope>runtime</scope>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-core</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-entitymanager</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-osgi</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-envers</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-c3p0</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-proxool</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-infinispan</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* <dependency>
* <groupId>org.hibernate</groupId>
* <artifactId>hibernate-ehcache</artifactId>
* <version>4.3.0.Final</version>
* </dependency>
* </dependencies>
* <build>
* <sourceDirectory>src/main/java</sourceDirectory>
* <plugins>
* <plugin>
* <artifactId>maven-compiler-plugin</artifactId>
* <version>3.5.1</version>
* <configuration>
* <source>1.8</source>
* <target>1.8</target>
* </configuration>
* </plugin>
* <!-- Embedded Apache Tomcat required for testing war -->
* <plugin>
* <groupId>org.apache.tomcat.maven</groupId>
* <artifactId>tomcat7-maven-plugin</artifactId>
* <version>2.2</version>
* <configuration>
* <path>/</path>
* </configuration>
* </plugin>
* </plugins>
* </build>
* </project>

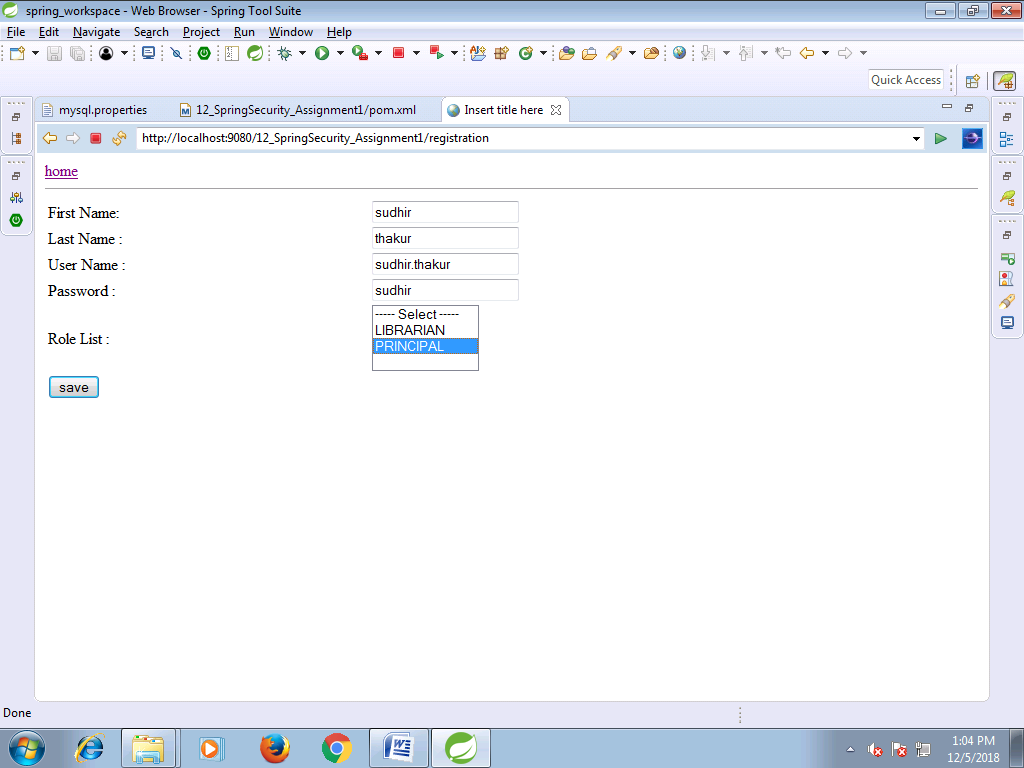
**Screenshots**

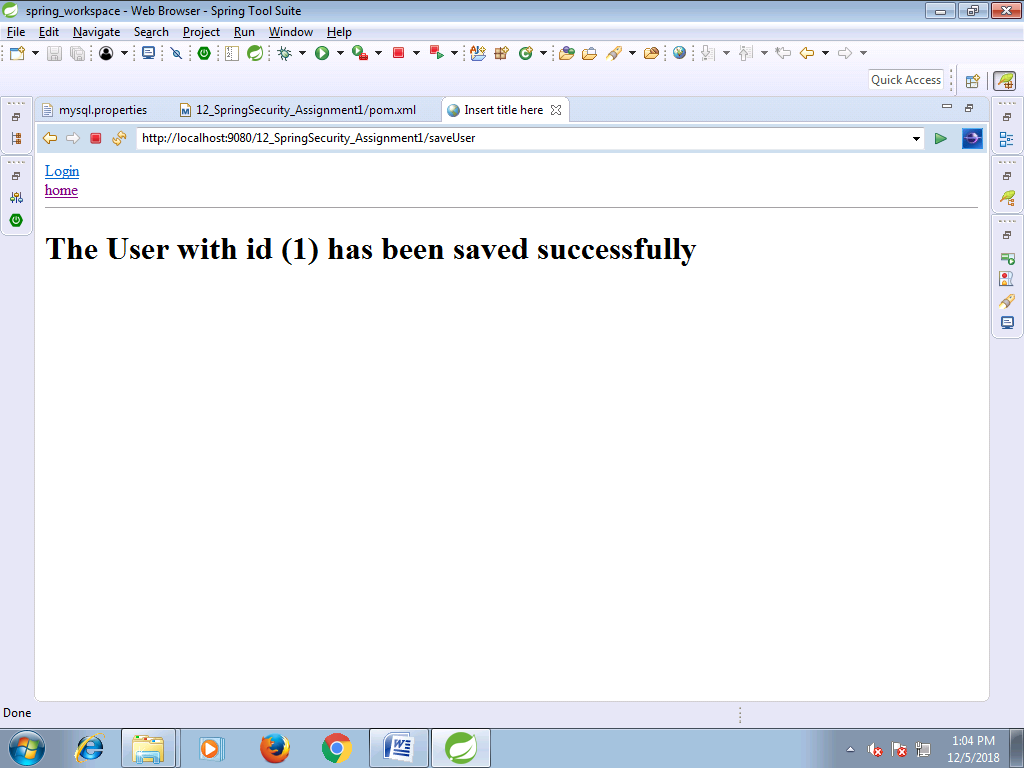
**Menus**

****

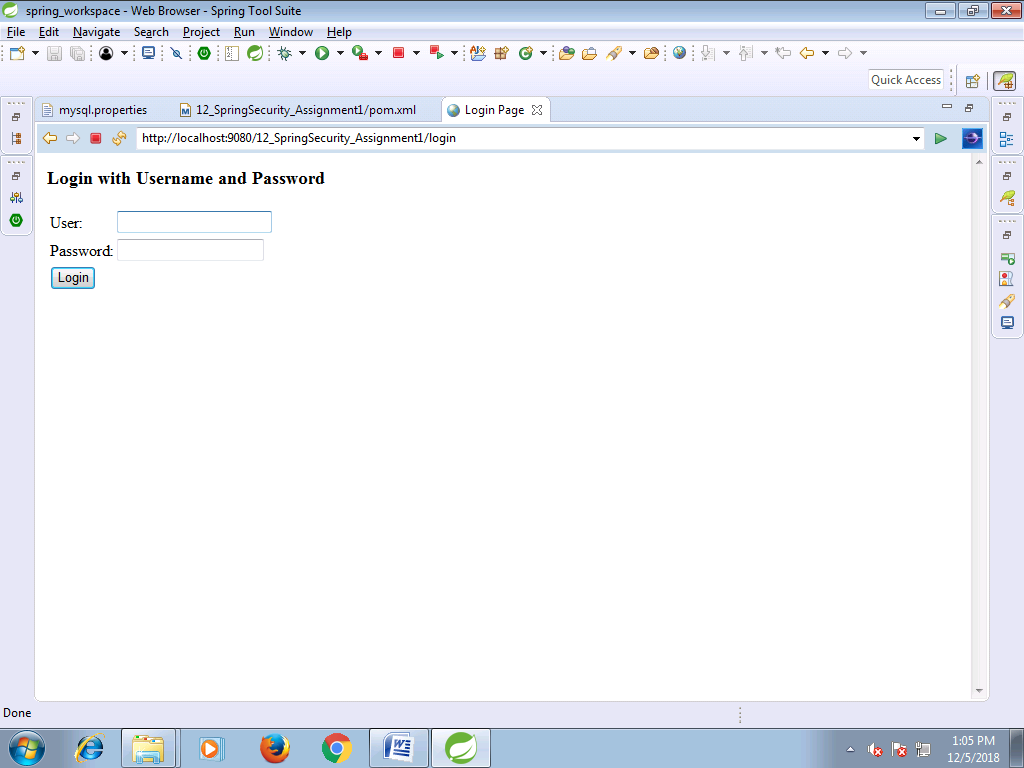
**Registration**

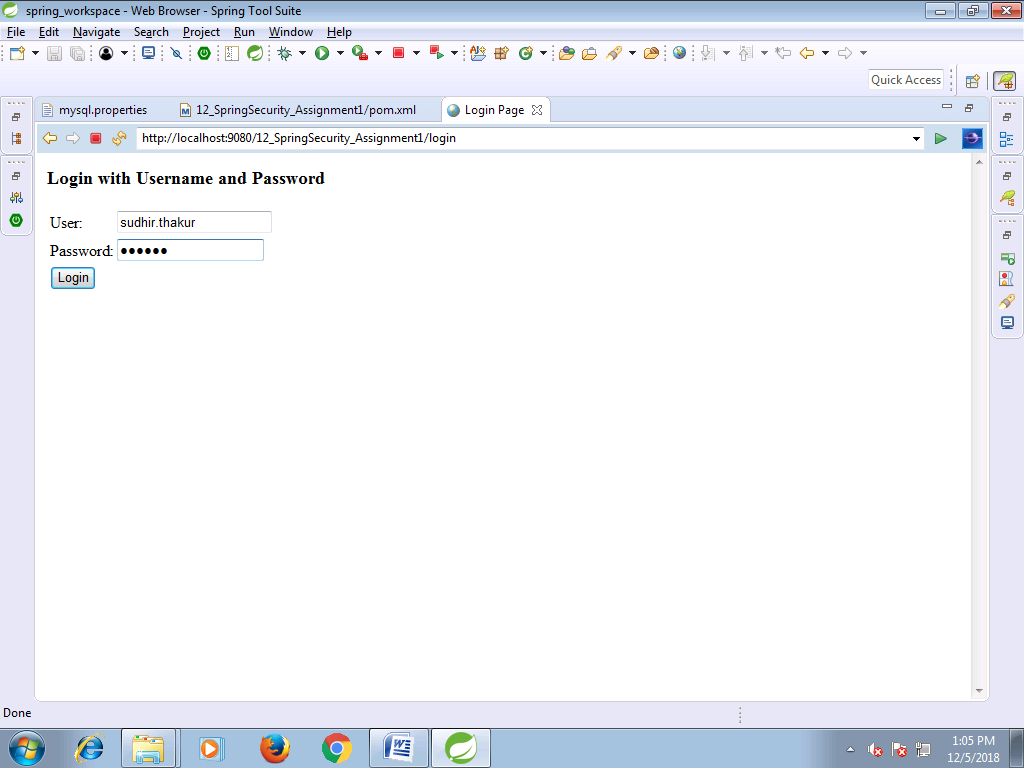
****

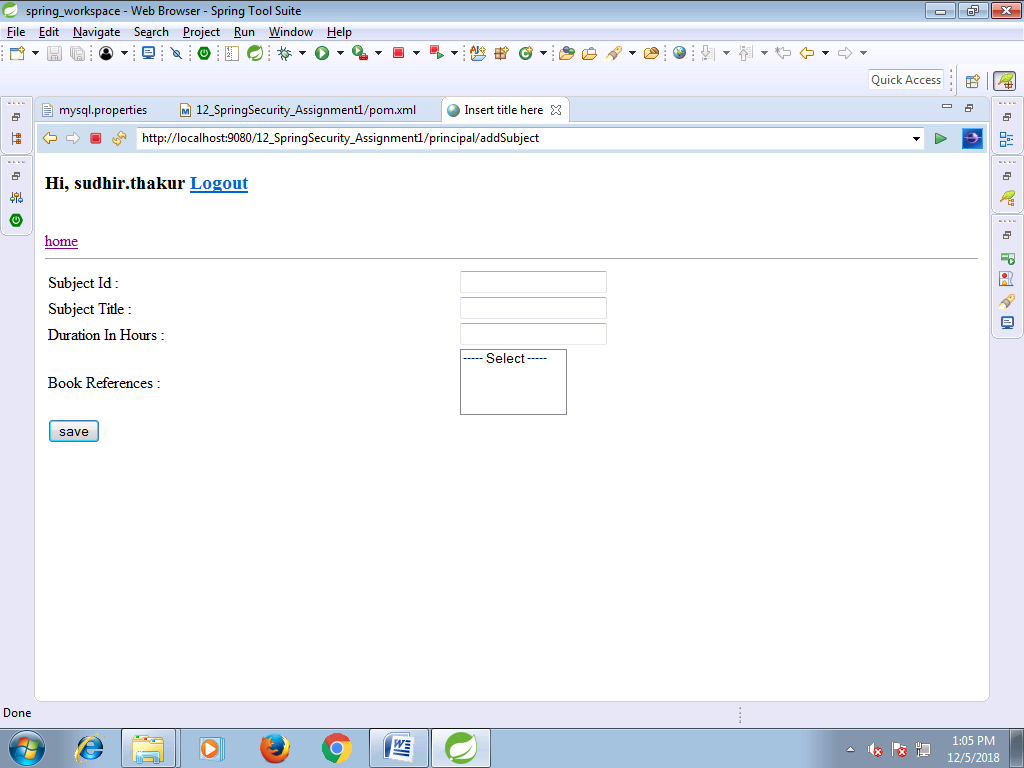
****

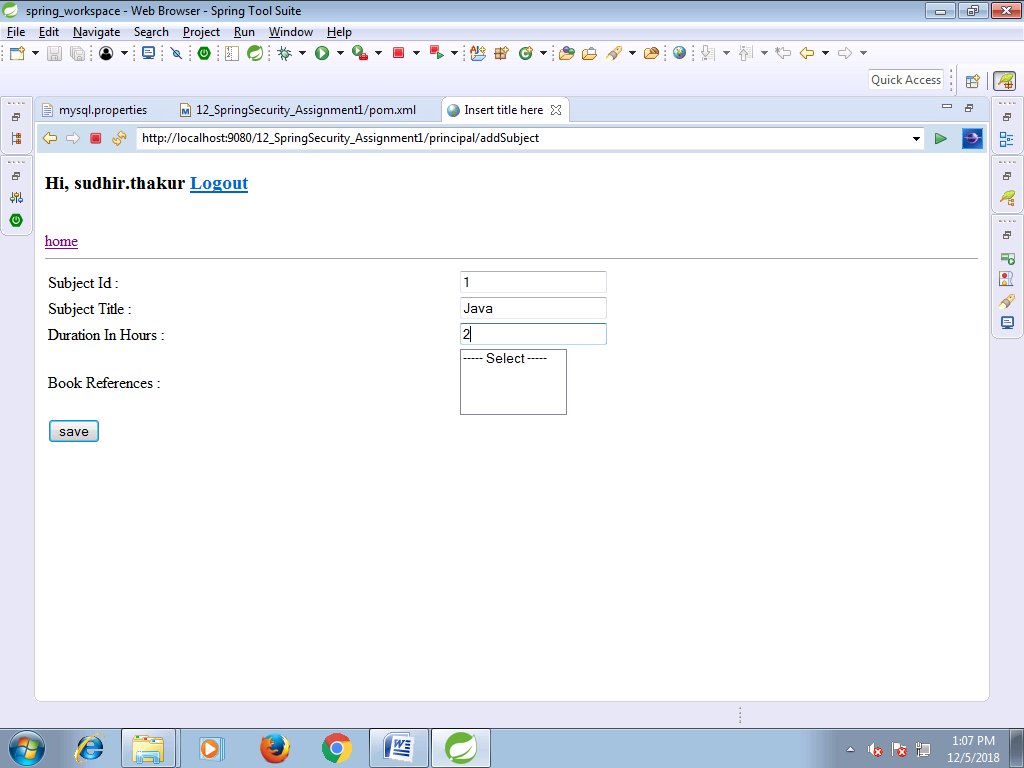
****

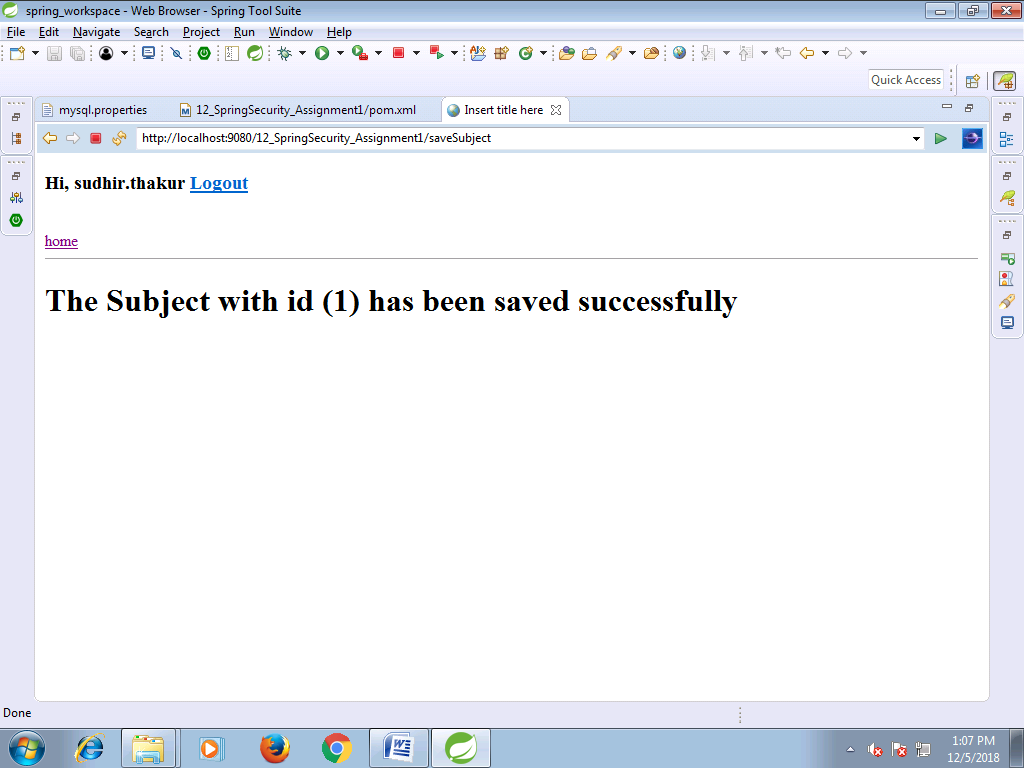
**Login & Add Subject**

****

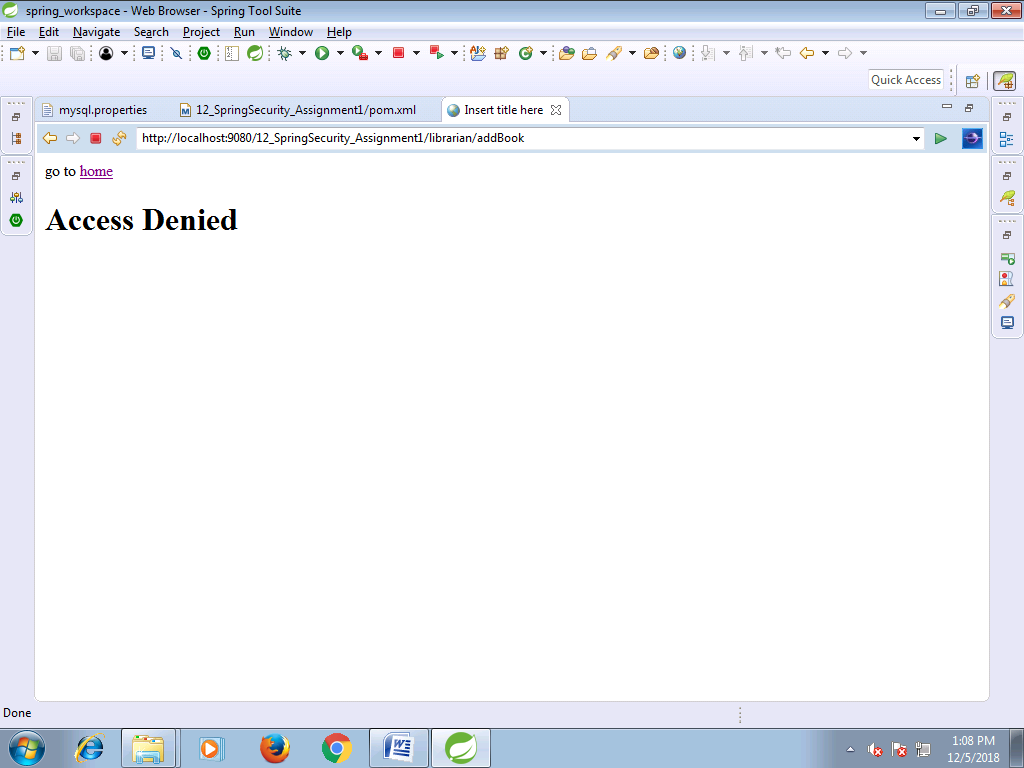
****

****

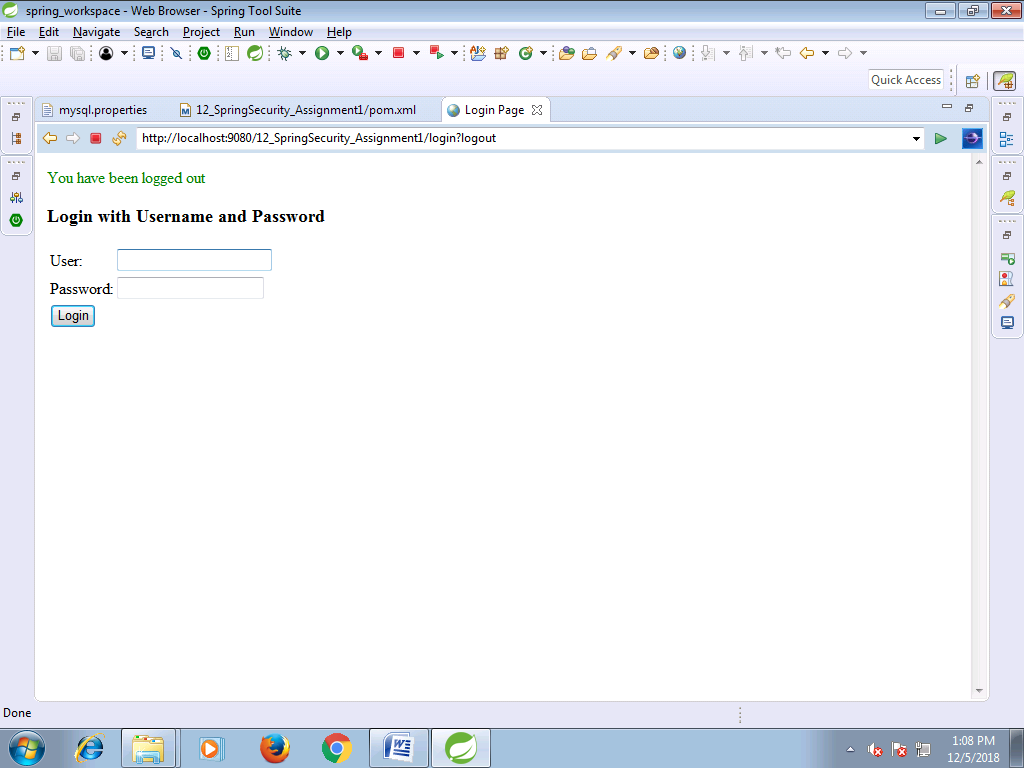
****

****

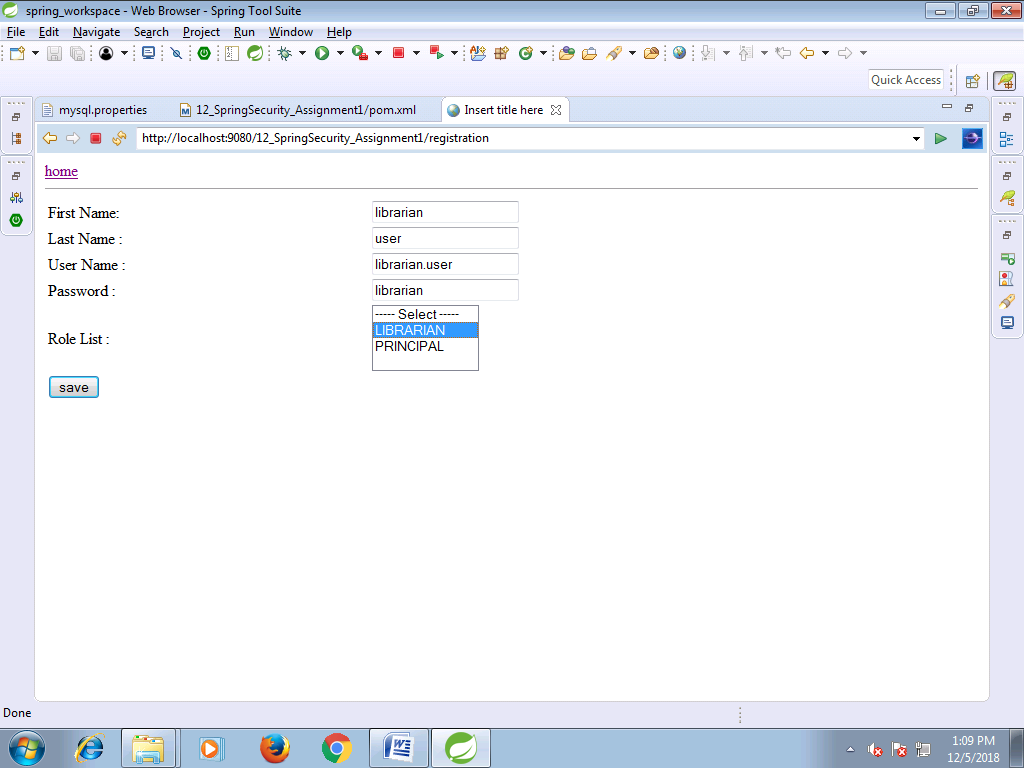
**Add a book**

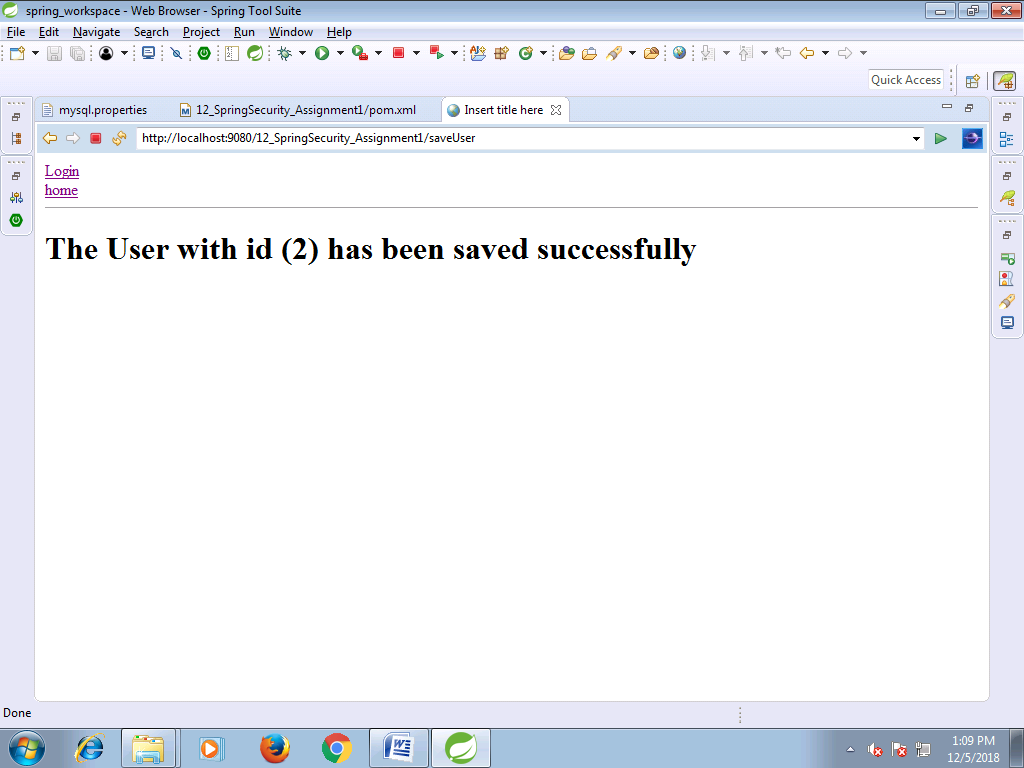
****

**Logout**

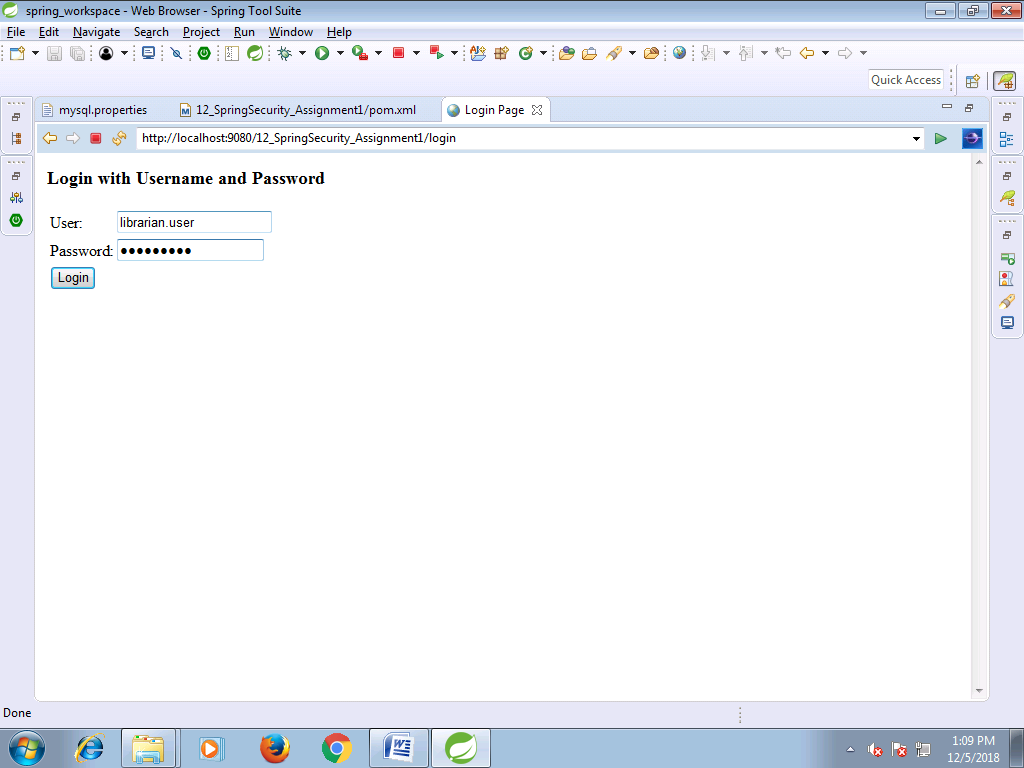
****

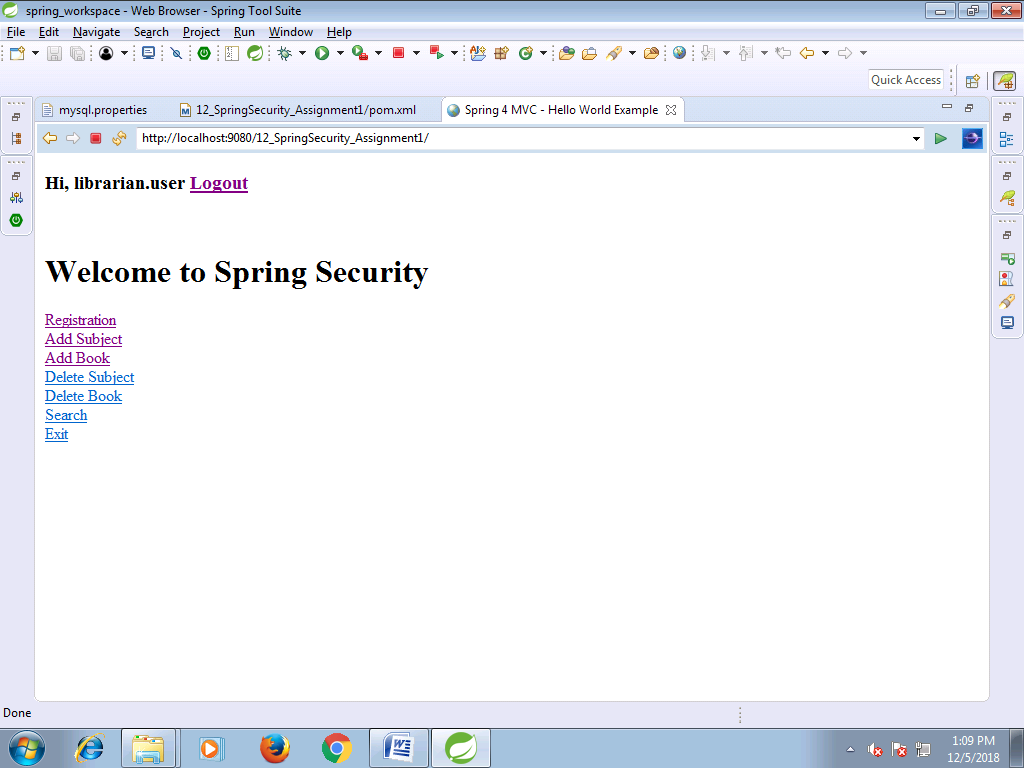
**Registration for Librarian**

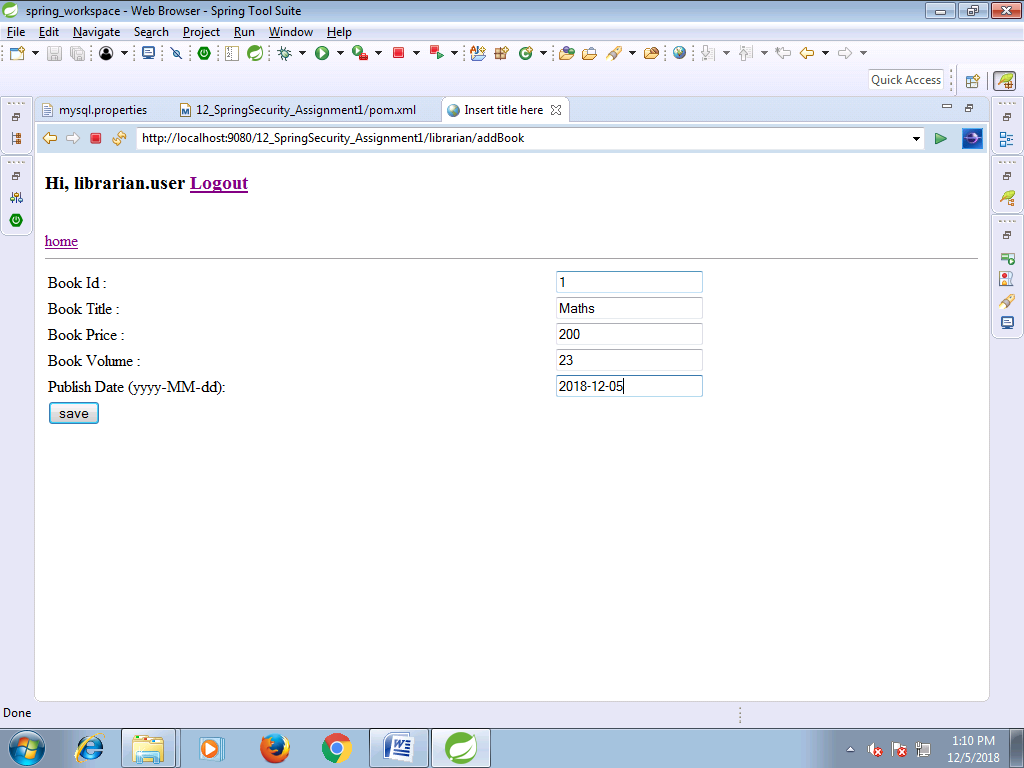
****

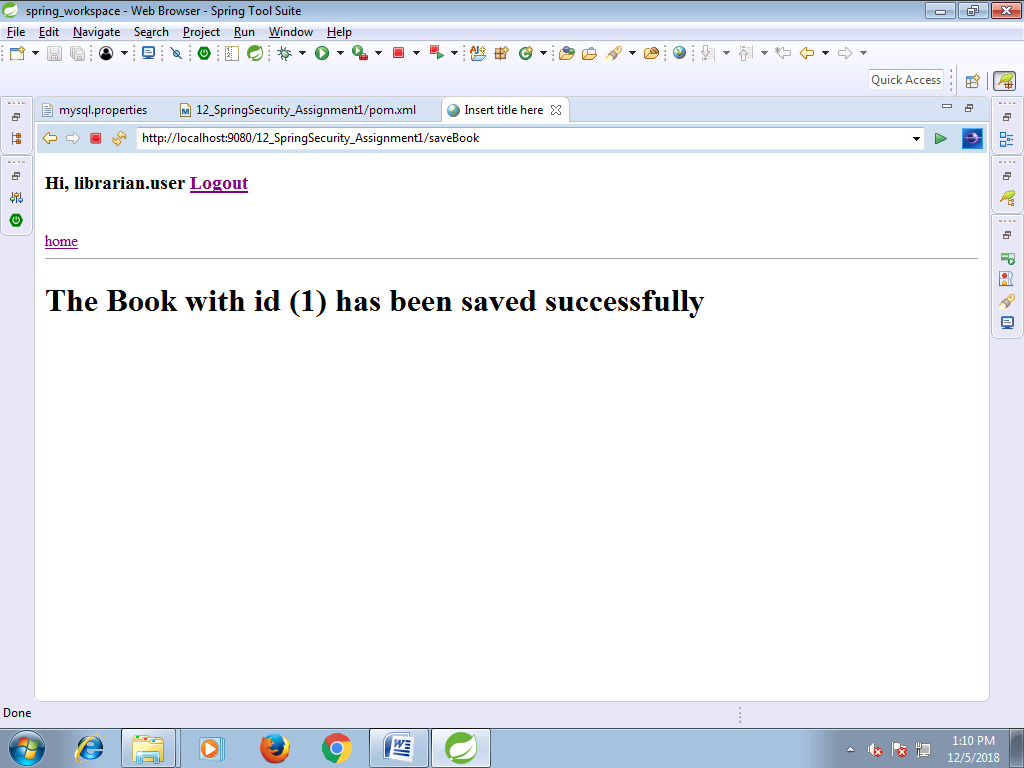
****

**Login & add a book**

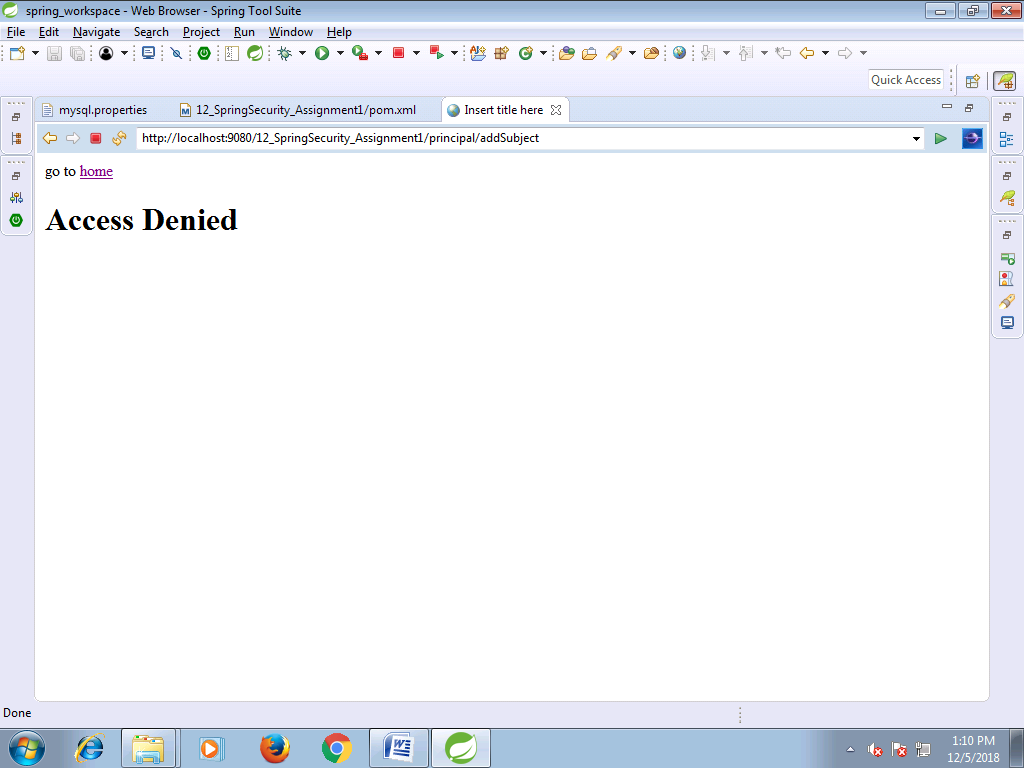
****

****

****

****

**Add Subject as librarian**

****