BookShop\src\java\com\sharanamvaishali\controllers

AuthenticationController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.service.\*;

import java.util.Map;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

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

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

@Controller

public class AuthenticationController {

@Autowired

private SearchService searchService;

@Autowired

private BookService bookService;

@Autowired

private AuthorService authorService;

@Autowired

private PublisherService publisherService;

@Autowired

private CategoryService categoryService;

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

public String showAdminLogin(ModelMap model) {

return "adminLogin";

}

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

public String showAdminLoginError(ModelMap model) {

model.addAttribute("error", "true");

return "adminLogin";

}

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

public String showAdminLoginAfterLogout(ModelMap model) {

return "adminLogin";

}

@RequestMapping(value="/adminAccessDenied")

public String showAccessDeniedPage(ModelMap model) {

return "adminAccessDenied";

}

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

public String showLogin(ModelMap model) {

return "home";

}

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

public String showLoginError(ModelMap model, Map<String, Object> map) {

map.put("newReleasesList", bookService.listNewReleases());

map.put("ourAuthorsList", authorService.listOurAuthors());

map.put("ourPublishersList", publisherService.listOurPublishers());

map.put("allCategoriesList", categoryService.listCategory());

map.put("allBooksList", bookService.listBook());

map.put("popularSearchList", searchService.listPopularSearches());

map.put("updatedBooksList", bookService.listUpdatedBooks());

map.put("topTitlesList", bookService.listTopTitles());

map.put("search", new Search());

model.addAttribute("error", "true");

return "home";

}

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

public String showHomeAfterLogout(ModelMap model, Map<String, Object> map) {

map.put("newReleasesList", bookService.listNewReleases());

map.put("ourAuthorsList", authorService.listOurAuthors());

map.put("ourPublishersList", publisherService.listOurPublishers());

map.put("allCategoriesList", categoryService.listCategory());

map.put("allBooksList", bookService.listBook());

map.put("popularSearchList", searchService.listPopularSearches());

map.put("updatedBooksList", bookService.listUpdatedBooks());

map.put("topTitlesList", bookService.listTopTitles());

map.put("search", new Search());

return "home";

}

}

AuthorController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Author;

import com.sharanamvaishali.service.AuthorService;

import com.sharanamvaishali.service.CommonService;

import com.sharanamvaishali.service.CountryService;

import com.sharanamvaishali.service.StateService;

import java.io.IOException;

import java.io.OutputStream;

import java.sql.SQLException;

import java.util.Map;

import javax.servlet.http.HttpServletResponse;

import javax.validation.Valid;

import org.apache.commons.io.IOUtils;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.dao.DataIntegrityViolationException;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

import org.springframework.web.multipart.MultipartFile;

@Controller

@RequestMapping("/admin")

public class AuthorController {

@Autowired

private CommonService commonService;

@Autowired

private AuthorService authorService;

@Autowired

private CountryService countryService;

@Autowired

private StateService stateService;

@RequestMapping("/manageAuthors")

public String showManageAuthors(Map<String, Object> map) {

map.put("author", new Author());

map.put("authorList", authorService.listAuthor());

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageAuthors";

}

@RequestMapping("/downloadAuthorPhotograph/{authorNo}")

public String downloadAuthorPhotograph(@PathVariable("authorNo") Integer authorNo, HttpServletResponse response) {

Author author = authorService.getAuthorById(authorNo);

try {

if (author.getPhotograph()!=null) {

response.setHeader("Content-Disposition", "inline;filename=\"" + author.getFirstName() + "\"");

OutputStream out = response.getOutputStream();

response.setContentType("image/gif");

IOUtils.copy(author.getPhotograph().getBinaryStream(), out);

out.flush();

out.close();

}

} catch (IOException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

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

public String saveAuthor(Map<String, Object> map, @ModelAttribute("author") Author author, @Valid Author authorValid, BindingResult result, @RequestParam("file") MultipartFile file) throws IOException {

if (result.hasErrors()) {

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageAuthors";

} else {

if(!file.isEmpty()) {

author.setPhotograph(commonService.getBlob(file.getBytes()));

}

else {

if (author.getAuthorNo() != null) {

author.setPhotograph(authorService.getAuthorById(author.getAuthorNo()).getPhotograph());

}

}

try {

authorService.saveAuthor(author);

return "redirect:/admin/manageAuthors";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageAuthors";

}

}

}

@RequestMapping("/deleteAuthor/{authorNo}")

public String deleteAuthor(Map<String, Object> map, @PathVariable("authorNo") Integer authorNo) {

try {

authorService.removeAuthor(authorNo);

return "redirect:/admin/manageAuthors";

} catch (DataIntegrityViolationException exp) {

map.put("dbError", "Cannot delete a parent row.");

map.put("author", new Author());

return "manageAuthors";

}

}

@RequestMapping("/editAuthor/{authorNo}")

public String editAuthor(Map<String, Object> map, @PathVariable("authorNo") Integer authorNo) {

map.put("author", authorService.getAuthorById(authorNo));

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageAuthors";

}

}

BookController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Book;

import com.sharanamvaishali.service.\*;

import java.io.IOException;

import java.io.OutputStream;

import java.sql.SQLException;

import java.util.Map;

import javax.servlet.http.HttpServletResponse;

import javax.validation.Valid;

import org.apache.commons.io.IOUtils;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

import org.springframework.web.multipart.MultipartFile;

@Controller

@RequestMapping("/admin")

public class BookController {

@Autowired

private CommonService commonService;

@Autowired

private BookService bookService;

@Autowired

private PublisherService publisherService;

@Autowired

private CategoryService categoryService;

@Autowired

private AuthorService authorService;

@RequestMapping("/manageBooks")

public String showManageBooks(Map<String, Object> map) {

map.put("book", new Book());

map.put("bookList", bookService.listBook());

map.put("publisherList", publisherService.listPublisher());

map.put("categoryList", categoryService.listCategory());

map.put("authorList", authorService.listAuthor());

return "manageBooks";

}

@RequestMapping("/downloadBookPhotograph/{bookNo}")

public String downloadBookPhotograph(@PathVariable("bookNo") Integer bookNo, HttpServletResponse response) {

Book book = bookService.getBookById(bookNo);

try {

if (book.getCoverPage()!=null) {

response.setHeader("Content-Disposition", "inline;filename=\"" +book.getBookName()+ "\"");

OutputStream out = response.getOutputStream();

response.setContentType("image/gif");

IOUtils.copy(book.getCoverPage().getBinaryStream(), out);

out.flush();

out.close();

}

} catch (IOException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

@RequestMapping("/downloadBookTOC/{bookNo}")

public String downloadBookTOC(@PathVariable("bookNo") Integer bookNo, HttpServletResponse response) {

Book book = bookService.getBookById(bookNo);

try {

if (book.getToc()!=null) {

response.setHeader("Content-Disposition", "inline;filename=\"" +book.getBookName()+ "\"");

OutputStream out = response.getOutputStream();

response.setContentType("application/pdf");

IOUtils.copy(book.getToc().getBinaryStream(), out);

out.flush();

out.close();

}

} catch (IOException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

@RequestMapping("/downloadBookSampleChapter/{bookNo}")

public String downloadBookSampleChapter(@PathVariable("bookNo") Integer bookNo, HttpServletResponse response) {

Book book = bookService.getBookById(bookNo);

try {

if (book.getSampleChapter()!=null) {

response.setHeader("Content-Disposition", "inline;filename=\"" +book.getBookName()+ "\"");

OutputStream out = response.getOutputStream();

response.setContentType("application/pdf");

IOUtils.copy(book.getSampleChapter().getBinaryStream(), out);

out.flush();

out.close();

}

} catch (IOException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

return null;

}

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

public String saveBook(Map<String, Object> map, @ModelAttribute("book") Book book, @Valid Book bookValid, BindingResult result, @RequestParam("coverPageFile") MultipartFile coverPageFile, @RequestParam("tocFile") MultipartFile tocFile, @RequestParam("sampleChapterFile") MultipartFile sampleChapterFile) throws IOException {

if (result.hasErrors()) {

map.put("bookList", bookService.listBook());

map.put("publisherList", publisherService.listPublisher());

map.put("categoryList", categoryService.listCategory());

map.put("authorList", authorService.listAuthor());

return "manageBooks";

} else {

if(!coverPageFile.isEmpty()) {

book.setCoverPage(commonService.getBlob(coverPageFile.getBytes()));

}

else {

if (book.getBookNo() != null) {

book.setCoverPage(bookService.getBookById(book.getBookNo()).getCoverPage());

}

}

if(!tocFile.isEmpty()) {

book.setToc(commonService.getBlob(tocFile.getBytes()));

}

else {

if (book.getBookNo() != null) {

book.setToc(bookService.getBookById(book.getBookNo()).getToc());

}

}

if(!sampleChapterFile.isEmpty()) {

book.setSampleChapter(commonService.getBlob(sampleChapterFile.getBytes()));

}

else {

if (book.getBookNo() != null) {

book.setSampleChapter(bookService.getBookById(book.getBookNo()).getSampleChapter());

}

}

try {

bookService.saveBook(book);

bookService.notifyCustomersByMail(book);

return "redirect:/admin/manageBooks";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageBooks";

}

}

}

@RequestMapping("/deleteBook/{bookNo}")

public String deleteBook(@PathVariable("bookNo") Integer bookNo) {

bookService.removeBook(bookNo);

return "redirect:/admin/manageBooks";

}

@RequestMapping("/editBook/{bookNo}")

public String editBook(Map<String, Object> map, @PathVariable("bookNo") Integer bookNo) {

map.put("book", bookService.getBookById(bookNo));

map.put("publisherList", publisherService.listPublisher());

map.put("categoryList", categoryService.listCategory());

map.put("authorList", authorService.listAuthor());

return "manageBooks";

}

}

CartController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.CartItem;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.service.\*;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpSession;

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

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

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class CartController {

@Value("${googleMerchantID}")

String googleMerchantID;

@Autowired

private SearchService searchService;

@Autowired

private BookService bookService;

@Autowired

private AuthorService authorService;

@Autowired

private PublisherService publisherService;

@Autowired

private CategoryService categoryService;

@Autowired

private CartService cartService;

@Autowired

private UserService userService;

@RequestMapping("/showCart/{userName}")

public String showCart(Map<String, Object> map, @PathVariable("userName") String userName, HttpSession session) {

map.put("cartList", (List<CartItem>) session.getAttribute(userName));

map.put("search", new Search());

map.put("googleMerchantID", googleMerchantID);

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "showCart";

}

@RequestMapping("/addToCart/{userName}/{bookNo}")

public String addToCart(Map<String, Object> map, @PathVariable("bookNo") Integer bookNo, @PathVariable("userName") String userName, HttpSession session) {

cartService.addBookToCart(bookNo, userName, session);

map.put("newReleasesList", bookService.listNewReleases());

map.put("ourAuthorsList", authorService.listOurAuthors());

map.put("ourPublishersList", publisherService.listOurPublishers());

map.put("allCategoriesList", categoryService.listCategory());

map.put("allBooksList", bookService.listBook());

map.put("popularSearchList", searchService.listPopularSearches());

map.put("updatedBooksList", bookService.listUpdatedBooks());

map.put("topTitlesList", bookService.listTopTitles());

map.put("search", new Search());

return "redirect:/home";

}

@RequestMapping("/removeFromCart/{userName}/{bookNo}")

public String removeFromCart(Map<String, Object> map, @PathVariable("bookNo") Integer bookNo, @PathVariable("userName") String userName, HttpSession session) {

cartService.removeBookFromCart(bookNo, userName, session);

map.put("search", new Search());

return "redirect:/showCart/{userName}";

}

@RequestMapping("/saveCart/{userName}")

public String saveCart(Map<String, Object> map, @PathVariable("userName") String userName, HttpSession session) {

cartService.saveCart(userName, session);

return "redirect:/checkOut/{userName}";

}

@RequestMapping("/checkOut/{userName}")

public String checkOut(Map<String, Object> map, @PathVariable("userName") String userName, HttpSession session) {

map.put("cartList", (List<CartItem>) session.getAttribute(userName));

map.put("googleMerchantID", googleMerchantID);

session.removeAttribute(userName);

return "checkOut";

}

}

CategoryController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Category;

import com.sharanamvaishali.service.CategoryService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.dao.DataIntegrityViolationException;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

@Controller

@RequestMapping("/admin")

public class CategoryController {

@Autowired

private CategoryService categoryService;

@RequestMapping("/manageCategories")

public String showManageCategories(Map<String, Object> map) {

map.put("category", new Category());

map.put("categoryList", categoryService.listCategory());

return "manageCategories";

}

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

public String saveCategory(Map<String, Object> map, @ModelAttribute("category") Category category, @Valid Category categoryValid, BindingResult result) {

if (result.hasErrors()) {

return "manageCategories";

} else {

try {

categoryService.saveCategory(category);

return "redirect:/admin/manageCategories";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageCategories";

}

}

}

@RequestMapping("/deleteCategory/{categoryNo}")

public String deleteCategory(Map<String, Object> map, @PathVariable("categoryNo") Integer categoryNo) {

try {

categoryService.removeCategory(categoryNo);

return "redirect:/admin/manageCategories";

} catch (DataIntegrityViolationException exp) {

map.put("dbError", "Cannot delete a parent row.");

map.put("category", new Category());

return "manageCategories";

}

}

@RequestMapping("/editCategory/{categoryNo}")

public String editCategory(Map<String, Object> map, @PathVariable("categoryNo") Integer categoryNo) {

map.put("category", categoryService.getCategoryById(categoryNo));

return "manageCategories";

}

}

CountryController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Country;

import com.sharanamvaishali.service.CountryService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.dao.DataIntegrityViolationException;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

@Controller

@RequestMapping("/admin")

public class CountryController {

@Autowired

private CountryService countryService;

@RequestMapping("/manageCountries")

public String showManageCountries(Map<String, Object> map) {

map.put("country", new Country());

map.put("countryList", countryService.listCountry());

return "manageCountries";

}

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

public String saveCountry(Map<String, Object> map, @ModelAttribute("country") Country country, @Valid Country countryValid, BindingResult result) {

if (result.hasErrors()) {

return "manageCountries";

} else {

try {

countryService.saveCountry(country);

return "redirect:/admin/manageCountries";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageCountries";

}

}

}

@RequestMapping("/deleteCountry/{countryNo}")

public String deleteCountry(Map<String, Object> map, @PathVariable("countryNo") Integer countryNo) {

try {

countryService.removeCountry(countryNo);

return "redirect:/admin/manageCountries";

} catch (DataIntegrityViolationException exp) {

map.put("dbError", "Cannot delete a parent row.");

map.put("country", new Country());

return "manageCountries";

}

}

@RequestMapping("/editCountry/{countryNo}")

public String editCountry(Map<String, Object> map, @PathVariable("countryNo") Integer countryNo) {

map.put("country", countryService.getCountryById(countryNo));

return "manageCountries";

}

}

HomeController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.service.\*;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

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

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class HomeController {

@Autowired

private UserService userService;

@Autowired

private SearchService searchService;

@Autowired

private BookService bookService;

@Autowired

private AuthorService authorService;

@Autowired

private PublisherService publisherService;

@Autowired

private CategoryService categoryService;

@RequestMapping("/")

public String startHome() {

return "redirect:/home";

}

@RequestMapping("/home")

public String showHome(Map<String, Object> map, HttpServletRequest request) {

map.put("newReleasesList", bookService.listNewReleases());

map.put("ourAuthorsList", authorService.listOurAuthors());

map.put("ourPublishersList", publisherService.listOurPublishers());

map.put("allCategoriesList", categoryService.listCategory());

map.put("allBooksList", bookService.listBook());

map.put("popularSearchList", searchService.listPopularSearches());

map.put("updatedBooksList", bookService.listUpdatedBooks());

map.put("topTitlesList", bookService.listTopTitles());

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

map.put("search", new Search());

return "home";

}

@RequestMapping("/showBookDetails/{bookNo}")

public String showBookDetails(Map<String, Object> map, @PathVariable("bookNo") int bookNo) {

map.put("bookList", bookService.getBookById(bookNo));

bookService.updateHits(bookNo);

map.put("search", new Search());

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "showBookDetails";

}

@RequestMapping("/showAuthorDetails/{authorNo}")

public String showAuthorDetails(Map<String, Object> map, @PathVariable("authorNo") Integer authorNo) {

map.put("authorList", authorService.getAuthorById(authorNo));

map.put("bookList", authorService.getAllBooksByAuthor(authorNo));

map.put("search", new Search());

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "showAuthorDetails";

}

@RequestMapping("/showPublisherDetails/{publisherNo}")

public String showPublisherDetails(Map<String, Object> map, @PathVariable("publisherNo") Integer publisherNo) {

map.put("publisherList", publisherService.getPublisherById(publisherNo));

map.put("bookList", publisherService.getAllBooksByPublisher(publisherNo));

map.put("search", new Search());

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "showPublisherDetails";

}

@RequestMapping("/showCategoryDetails/{categoryNo}")

public String showCategoryDetails(Map<String, Object> map, @PathVariable("categoryNo") Integer categoryNo) {

map.put("categoryList", categoryService.getCategoryById(categoryNo));

map.put("bookList", categoryService.getAllBooksByCategory(categoryNo));

map.put("search", new Search());

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "showCategoryDetails";

}

}

PublisherController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Publisher;

import com.sharanamvaishali.service.CountryService;

import com.sharanamvaishali.service.PublisherService;

import com.sharanamvaishali.service.StateService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.dao.DataIntegrityViolationException;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

@Controller

@RequestMapping("/admin")

public class PublisherController {

@Autowired

private PublisherService publisherService;

@Autowired

private CountryService countryService;

@Autowired

private StateService stateService;

@RequestMapping("/managePublishers")

public String showManagePublishers(Map<String, Object> map) {

map.put("publisher", new Publisher());

map.put("publisherList", publisherService.listPublisher());

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "managePublishers";

}

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

public String savePublisher(Map<String, Object> map, @ModelAttribute("publisher") Publisher publisher, @Valid Publisher publisherValid, BindingResult result) {

if (result.hasErrors()) {

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "managePublishers";

} else {

try {

publisherService.savePublisher(publisher);

return "redirect:/admin/managePublishers";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "managePublishers";

}

}

}

@RequestMapping("/deletePublisher/{publisherNo}")

public String deletePublisher(Map<String, Object> map, @PathVariable("publisherNo") Integer publisherNo) {

try {

publisherService.removePublisher(publisherNo);

return "redirect:/admin/managePublishers";

} catch (DataIntegrityViolationException exp) {

map.put("dbError", "Cannot delete a parent row.");

map.put("publisher", new Publisher());

return "managePublishers";

}

}

@RequestMapping("/editPublisher/{publisherNo}")

public String editPublisher(Map<String, Object> map, @PathVariable("publisherNo") Integer publisherNo) {

map.put("publisher", publisherService.getPublisherById(publisherNo));

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "managePublishers";

}

}

SearchController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.PopularSearch;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.service.SearchService;

import com.sharanamvaishali.service.UserService;

import java.util.Map;

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

import org.springframework.security.core.context.SecurityContextHolder;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class SearchController {

@Autowired

private UserService userService;

@Autowired

private SearchService searchService;

@RequestMapping("/searchResults/{searchCriteria}")

public String showSearchResultsByCriteria(Map<String, Object> map, @PathVariable("searchCriteria") String searchCriteria) {

map.put("searchResultsList", searchService.searchResults(searchCriteria));

Search search = new Search(searchCriteria);

map.put("search", search);

PopularSearch ps = new PopularSearch();

ps.setSearchValue(searchCriteria);

searchService.savePopularSearch(ps);

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "searchResults";

}

@RequestMapping("/searchResults")

public String showSearchResults(Map<String, Object> map) {

map.put("searchResultsList", searchService.searchAllResults());

Search search = new Search();

map.put("search", search);

if(SecurityContextHolder.getContext().getAuthentication().isAuthenticated()) {

map.put("userDetails", userService.getUserByUsername(SecurityContextHolder.getContext().getAuthentication().getName()));

}

return "searchResults";

}

}

SignUpController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.model.User;

import com.sharanamvaishali.service.CountryService;

import com.sharanamvaishali.service.MailService;

import com.sharanamvaishali.service.StateService;

import com.sharanamvaishali.service.UserService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

@Controller

public class SignUpController {

@Value("${emailFrom}")

String emailFrom;

@Autowired

private MailService mailService;

@Autowired

private UserService userService;

@Autowired

private CountryService countryService;

@Autowired

private StateService stateService;

@RequestMapping("/signUp")

public String showSignUp(Map<String, Object> map) {

map.put("user", new User());

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

map.put("search", new Search());

return "signUp";

}

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

public String registerCustomer(Map<String, Object> map, @ModelAttribute("user") User user, @Valid User signUpValid, BindingResult result) {

if (result.hasErrors()) {

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

map.put("search", new Search());

return "signUp";

} else {

try {

user.setAuthority("CUSTOMER");

user.setEnabled(true);

userService.saveUser(user);

mailService.sendMail(emailFrom,

user.getEmailAddress(),

"Sharanam & Vaishali's Online Bookshop: Registration mail",

"<table width='100%' border='0' align='center' cellpadding='15' cellspacing='0' "

+ "style='font-family:Verdana, Arial, Helvetica, sans-serif; font-size:12pt; color:#5a5a5a;'>"

+ "<tr><td align='left'>Dear " + user.getFirstName() + ",</td></tr><tr>"

+ "<td align='left'>Your login details are:<br/><br/>"

+ "Username: " + user.getUserName() + "<br />"

+ "Password: " + user.getPassword() + "<br /><br/>"

+ "<p>Thank you for using this site.<br />"

+ "</p><br/><br/><p>Regards,<br />Sharanam & Vaishali's Online Bookshop<br />"

+ "</p><p><br /><br />THIS IS AN AUTOMATED MESSAGE; PLEASE DO NOT REPLY. </p>"

+ "</td></tr></table>");

return "redirect:/signUpThankYou";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "signUp";

}

}

}

@RequestMapping("/signUpThankYou")

public String showSignUpThankYou(Map<String, Object> map) {

map.put("search", new Search());

return "signUpThankYou";

}

}

StateController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.State;

import com.sharanamvaishali.service.StateService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

import org.springframework.dao.DataIntegrityViolationException;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

@Controller

@RequestMapping("/admin")

public class StateController {

@Autowired

private StateService stateService;

@RequestMapping("/manageStates")

public String showManageStates(Map<String, Object> map) {

map.put("state", new State());

map.put("stateList", stateService.listState());

return "manageStates";

}

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

public String saveState(Map<String, Object> map, @ModelAttribute("state") State state, @Valid State stateValid, BindingResult result) {

if (result.hasErrors()) {

return "manageStates";

} else {

try {

stateService.saveState(state);

return "redirect:/admin/manageStates";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageStates";

}

}

}

@RequestMapping("/deleteState/{stateNo}")

public String deleteState(Map<String, Object> map, @PathVariable("stateNo") Integer stateNo) {

try {

stateService.removeState(stateNo);

return "redirect:/admin/manageStates";

} catch (DataIntegrityViolationException exp) {

map.put("dbError", "Cannot delete a parent row.");

map.put("state", new State());

return "manageStates";

}

}

@RequestMapping("/editState/{stateNo}")

public String editState(Map<String, Object> map, @PathVariable("stateNo") Integer stateNo) {

map.put("state", stateService.getStateById(stateNo));

return "manageStates";

}

}

TransactionController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.service.TransactionService;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Map;

import java.util.logging.Level;

import java.util.logging.Logger;

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

import org.springframework.stereotype.Controller;

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

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

@Controller

@RequestMapping("/admin")

public class TransactionController {

@Autowired

private TransactionService transactionService;

@RequestMapping("/manageTransactions")

public String showManageTransactions() {

return "manageTransactions";

}

@RequestMapping("/searchTransactionsByUsername")

public String showTransactions(Map<String, Object> map) {

map.put("transactionGroupedList", transactionService.listGroupedTransactions());

map.put("transactionList", transactionService.listTransactions());

return "manageTransactions";

}

@RequestMapping("/searchTransactionsByUsername/{userName}")

public String showTransactionsByUsername(Map<String, Object> map, @PathVariable("userName") String userName) {

map.put("transactionGroupedList", transactionService.listGroupedTransactionsByUsername(userName));

map.put("transactionList", transactionService.listTransactionsByUsername(userName));

return "manageTransactions";

}

@RequestMapping("/searchTransactionsByDate/{year}/{month}/{day}")

public String showTransactionsByDate(Map<String, Object> map, @PathVariable("day") String day, @PathVariable("year") String year, @PathVariable("month") String month) {

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

try {

map.put("transactionGroupedList", transactionService.listGroupedTransactionsByDate(sdf.parse(year + "-" + month + "-" + day)));

map.put("transactionList", transactionService.listTransactionsByDate(sdf.parse(year + "-" + month + "-" + day)));

} catch (ParseException ex) {

Logger.getLogger(TransactionController.class.getName()).log(Level.SEVERE, null, ex);

}

return "manageTransactions";

}

@RequestMapping("/updateGoogleOrderNo/{transactionNo}/{orderNo}")

public String updateGoogleOrderNo(Map<String, Object> map, @PathVariable("transactionNo") Integer transactionNo, @PathVariable("orderNo") String orderNo) {

transactionService.updateGoogleOrderNo(transactionNo, orderNo);

return "manageTransactions";

}

}

UserController.txt

package com.sharanamvaishali.controllers;

import com.sharanamvaishali.model.Search;

import com.sharanamvaishali.model.User;

import com.sharanamvaishali.service.CountryService;

import com.sharanamvaishali.service.MailService;

import com.sharanamvaishali.service.StateService;

import com.sharanamvaishali.service.UserService;

import java.util.Map;

import javax.validation.Valid;

import org.hibernate.exception.ConstraintViolationException;

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

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

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

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

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

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

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

@Controller

@RequestMapping("/admin")

public class UserController {

@Value("${emailFrom}")

String emailFrom;

@Autowired

private MailService mailService;

@Autowired

private UserService userService;

@Autowired

private CountryService countryService;

@Autowired

private StateService stateService;

@RequestMapping("/manageUsers")

public String showManageUsers(Map<String, Object> map) {

map.put("user", new User());

map.put("userList", userService.listUser());

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageUsers";

}

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

public String saveUser(Map<String, Object> map, @ModelAttribute("user") User user, @Valid User userValid, BindingResult result) {

if (result.hasErrors()) {

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageUsers";

} else {

try {

userService.saveUser(user);

return "redirect:/admin/manageUsers";

} catch (ConstraintViolationException exp) {

map.put("dbError", exp.getMessage());

return "manageUsers";

}

}

}

@RequestMapping("/deleteUser/{userNo}")

public String deleteUser(Map<String, Object> map, @PathVariable("userNo") Integer userNo) {

User user = userService.getUserById(userNo);

if(!user.getUserName().equals("admin")) {

userService.removeUser(userNo);

return "redirect:/admin/manageUsers";

}

else {

map.put("dbError", "You cannot delete the admin user.");

map.put("user", new User());

map.put("userList", userService.listUser());

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageUsers";

}

}

@RequestMapping("/editUser/{userNo}")

public String editUser(Map<String, Object> map, @PathVariable("userNo") Integer userNo) {

map.put("user", userService.getUserById(userNo));

map.put("countryList", countryService.listCountry());

map.put("stateList", stateService.listState());

return "manageUsers";

}

@RequestMapping("/forgotPassword")

public String showForgotPassword(Map<String, Object> map) {

Search search = new Search();

map.put("search", search);

return "forgotPassword";

}

@RequestMapping("/retrievePassword/{userName}")

public String retrievePassword(Map<String, Object> map, @PathVariable("userName") String userName) {

User user = userService.getUserByUsername(userName);

String message;

if (user != null) {

map.put("userDetails", user);

mailService.sendMail(emailFrom,

user.getEmailAddress(),

"Sharanam & Vaishali's Online Bookshop: Forgot Password mail",

"<table width='100%' border='0' align='center' cellpadding='0' cellspacing='0' "

+ "style='font-family:Verdana, Arial, Helvetica, sans-serif; font-size:12pt; color:#5a5a5a;'>"

+ "<tr><td align='left'><p>Dear " + user.getFirstName() + ",</p></td></tr><tr>"

+ "<td align='left'><p>As requested, please find your login details below:</p><br/>"

+ "<br/><p>Username: " + user.getUserName() + "<br />Password: " + user.getPassword() +

"<br /></p><br/><p>Thank you for using this site.<br /></p><br/><br/><p>"

+ "Regards,<br />Sharanam & Vaishali's Online Bookshop<br /></p><p><br /><br />"

+ "THIS IS AN AUTOMATED MESSAGE; PLEASE DO NOT REPLY. </p></td></tr></table>");

message = "Password sent to " + user.getEmailAddress();

} else {

message = "Invalid Username. Please try again";

}

map.put("message", message);

Search search = new Search();

map.put("search", search);

return "forgotPassword";

}

}

Src\default package

Bookshop.properties

googleMerchantID=

emailFrom=kishorspathak@gmail.com

emailUser=kishorspathak@gmail.com

emailPasswd=india@2030

emailHost=smtp.gmail.com

emailPort=587

dbUsername=root

dbPassword=root

dbURL=jdbc:mysql://localhost:3306/bms

dbDriverClassName=com.mysql.jdbc.Driver

showHQL=false

sendMail=true

src\default package

Hibernate.cfg

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

<!DOCTYPE hibernate-configuration *(View Source for full doctype...)*>

[**-**](file:///D:\Backup%20data\Desktop\Spring%203%20With%20Hib%204%20For%20prof%20Src%20Code\Code\BookShop\src\java\hibernate.cfg.xml) <hibernate-configuration>

[**-**](file:///D:\Backup%20data\Desktop\Spring%203%20With%20Hib%204%20For%20prof%20Src%20Code\Code\BookShop\src\java\hibernate.cfg.xml) <session-factory>

<mapping class="**com.sharanamvaishali.model.Country**" />

<mapping class="**com.sharanamvaishali.model.Category**" />

<mapping class="**com.sharanamvaishali.model.State**" />

<mapping class="**com.sharanamvaishali.model.Publisher**" />

<mapping class="**com.sharanamvaishali.model.User**" />

<mapping class="**com.sharanamvaishali.model.Author**" />

<mapping class="**com.sharanamvaishali.model.Book**" />

<mapping class="**com.sharanamvaishali.model.Transaction**" />

<mapping class="**com.sharanamvaishali.model.PopularSearch**" />

</session-factory>

</hibernate-configuration>