**One MarketPlace**

**Summary:**

Using Spring and Hibernate developed an application replicating to marketplace. This application can be used to rent a house or sell any used/unused products. This application consists of 2 roles admin and normal user. User can post advertisement of houses and products and wait for the approval of the post from the admin. Users can view the approved advertisement.

**Functionality:**

This application can help users to connect with each other for selling their used/unused products or rent their house. Users can add their advertisement by clicking on post. User can select between housing and product tabs. While posting housing advertisement user can enter basic details of their house including the expected rent. Similarly in products user can enters its details as well as select the category. User can search housing and products separately as well as filter their searches.

Admin can view all the unapproved advertisements. Likewise admin has the authority to approve and reject the post. Admin can also add categories for products.

**Technologies Used:**

Java, Spring MVC, Hibernate, Ajax, Javascript, JSP, JQuery, MySQL

**Roles:**

1. Admin

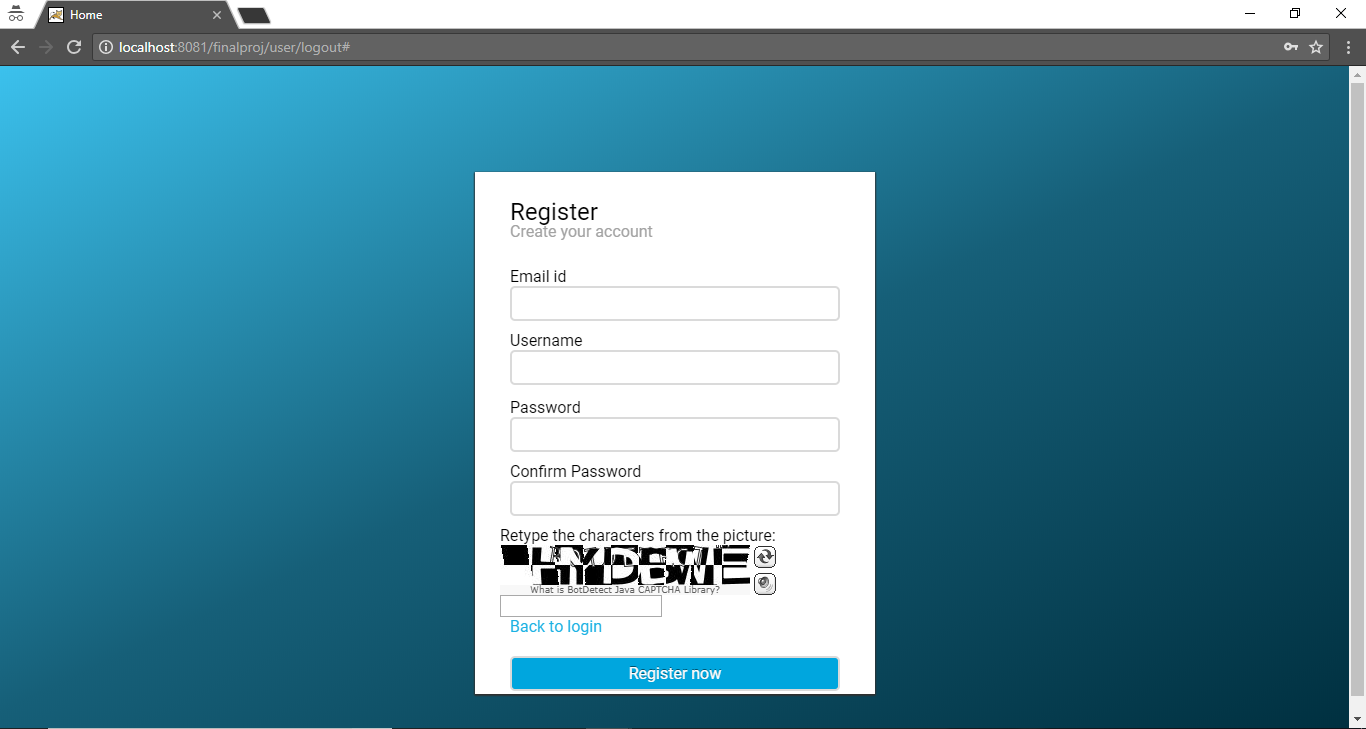
* Approve user’s housing post
* Approve user’s product post
* Add new categories
* Add new post

1. User

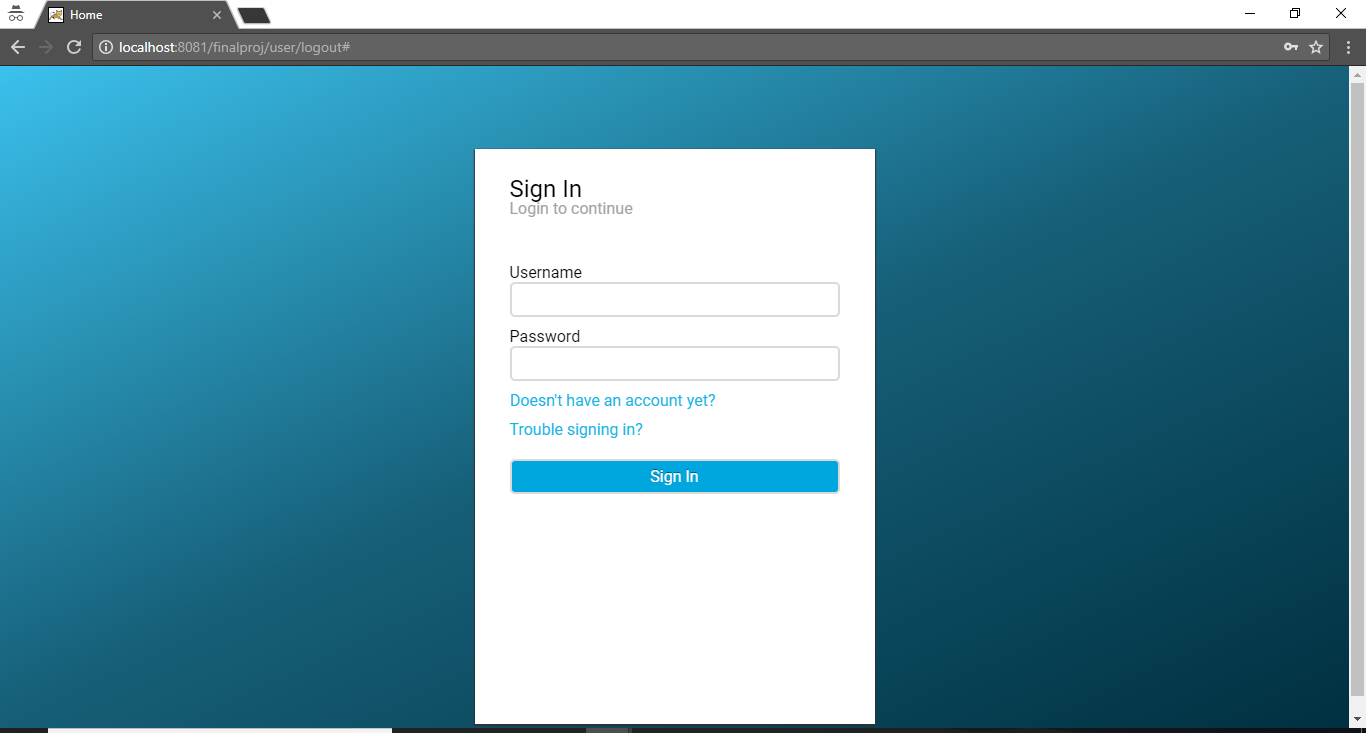
* Add new housing post
* Add new product post

**Screenshots:**

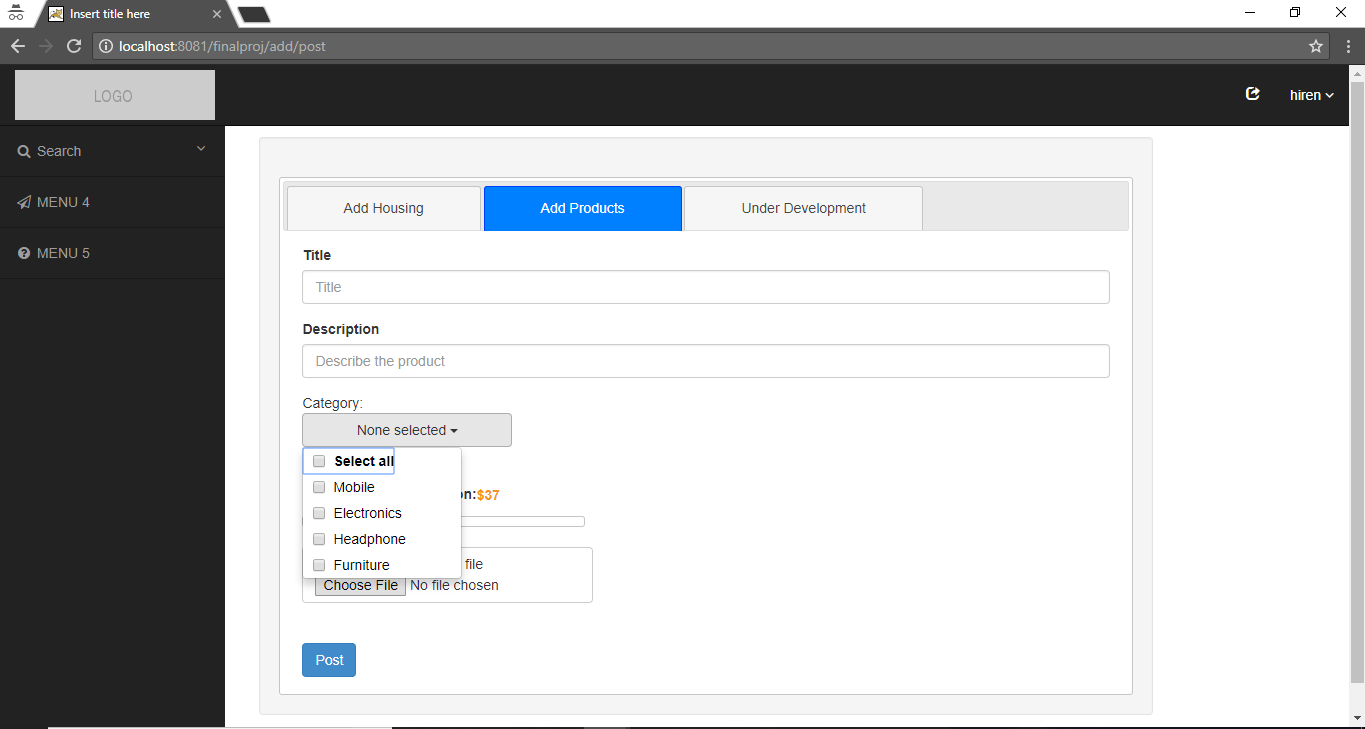
**Register Page**



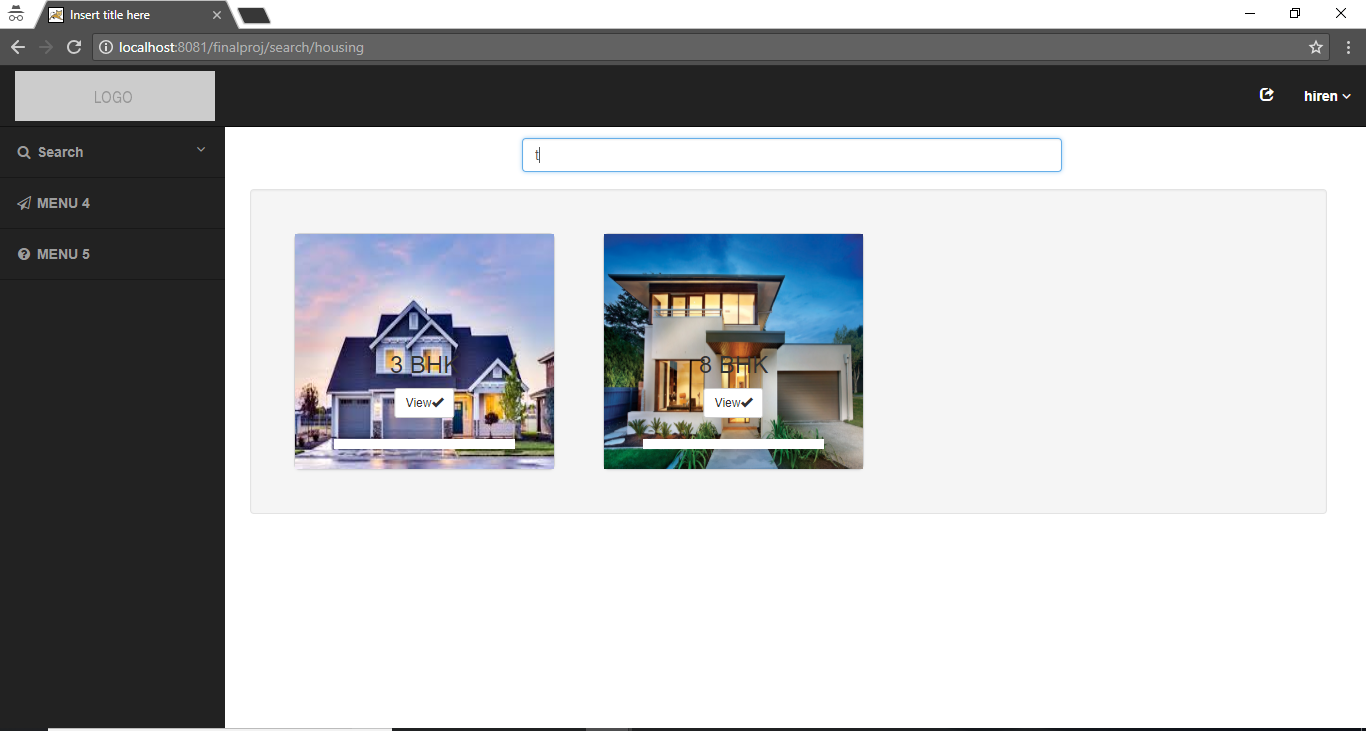
**Login page**



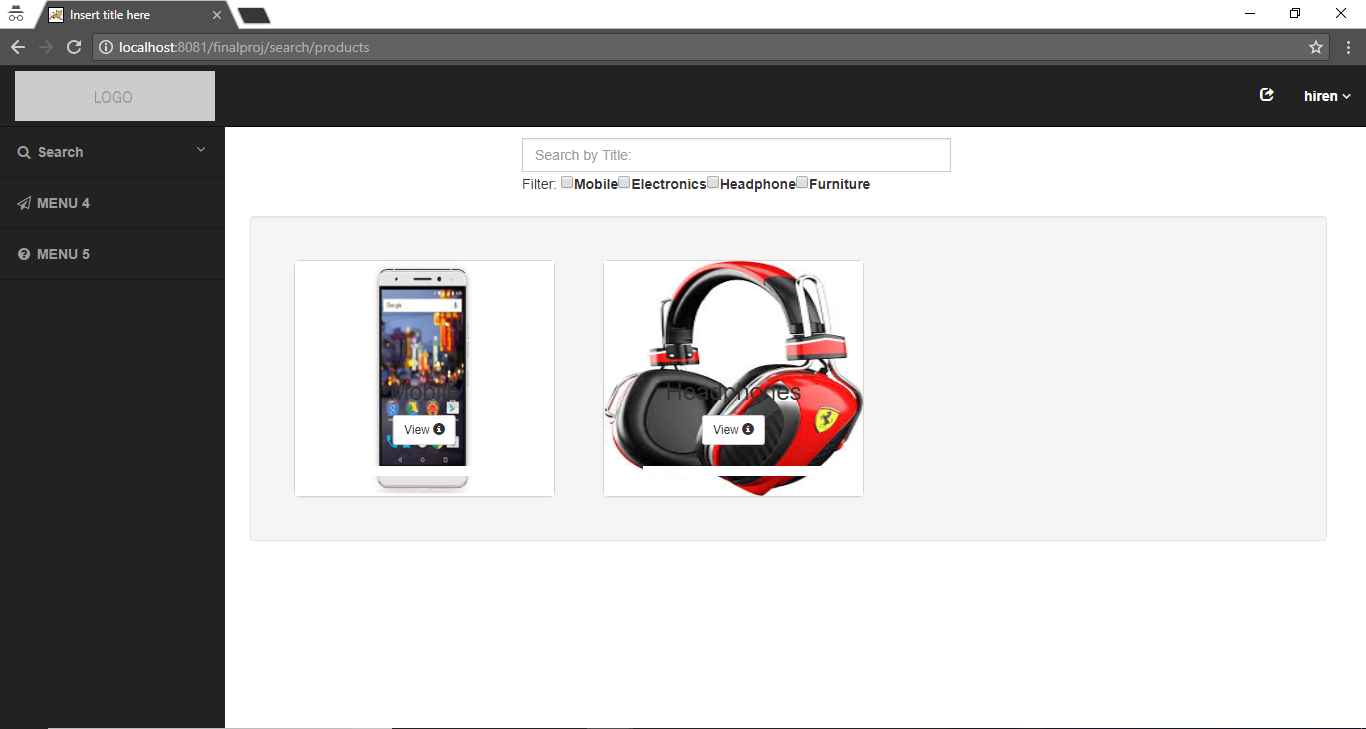
**Add Post**

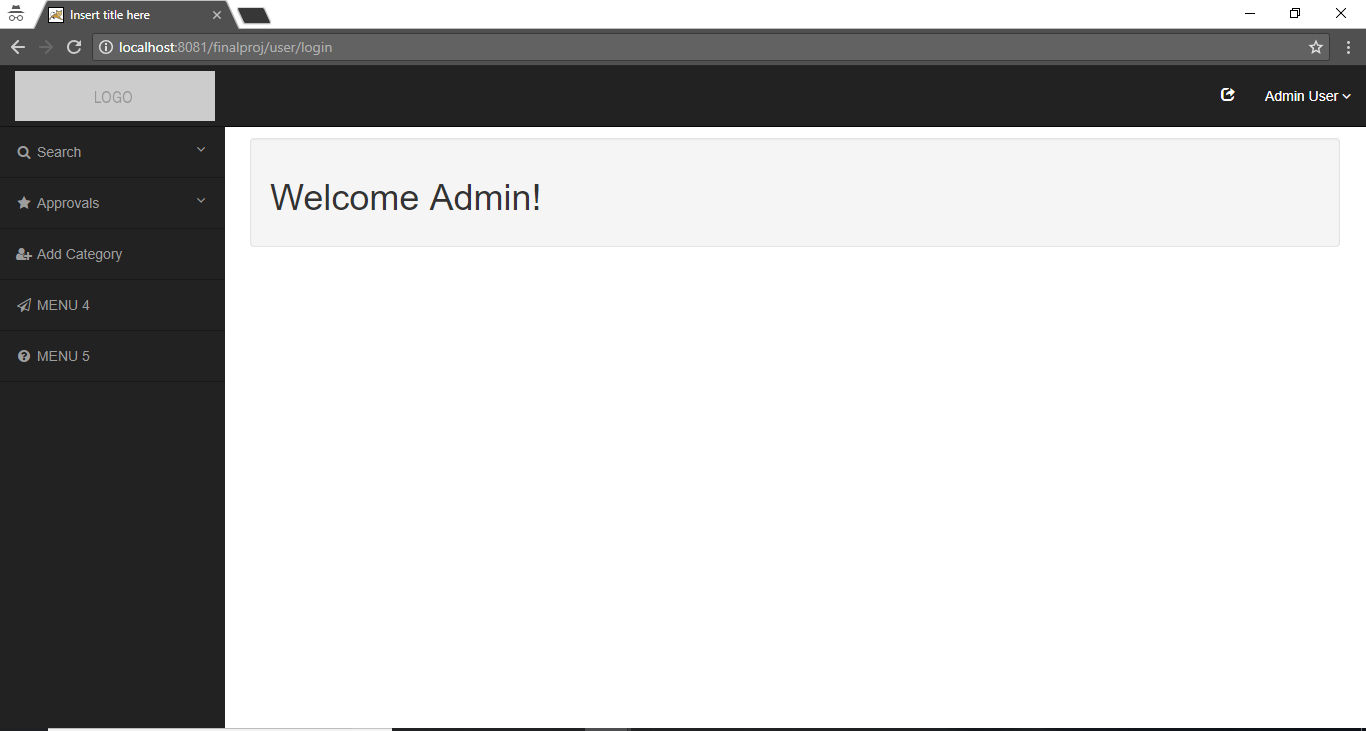


**Search page**

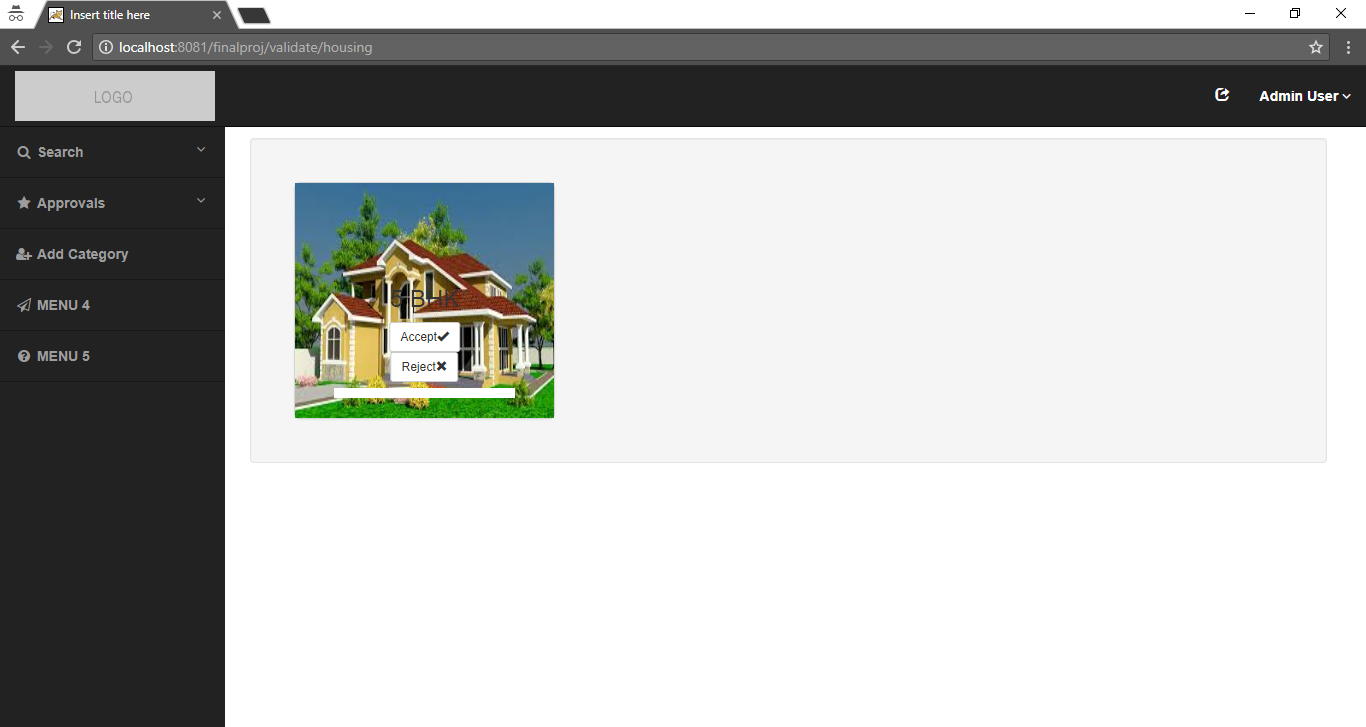


**Search Product**

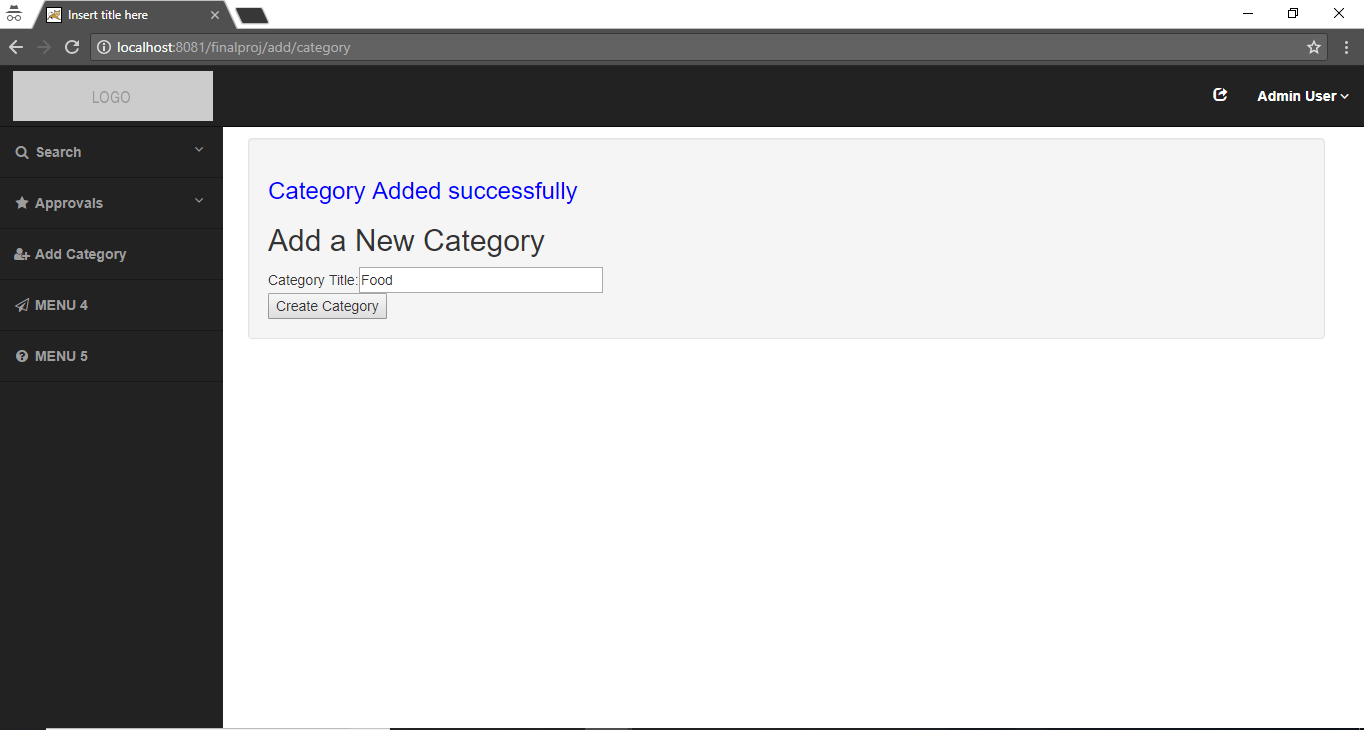




**Approve Reject Page**



**Add new Category page**



**Appendix:**

* **Controller Code**

**UserController.java**

**package** com.me.finalproj.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

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

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

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

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

**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.ResponseBody;

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

**import** com.captcha.botdetect.web.servlet.Captcha;

**import** com.me.finalproj.dao.UserDao;

**import** com.me.finalproj.exception.UserException;

**import** com.me.finalproj.pojo.User;

**import** com.me.finalproj.validator.UserValidator;

@Controller

@RequestMapping("/user/\*")

**public** **class** UserController {

@Autowired

@Qualifier("userDao")

UserDao userDao;

@Autowired

@Qualifier("userValidator")

UserValidator validator;

@InitBinder

**private** **void** initBinder(WebDataBinder binder) {

binder.setValidator(validator);

}

@RequestMapping(value = "/user/\*", method = RequestMethod.***GET***)

**protected** String loginUser(HttpServletRequest request) {

**if**(*validateSession*(request)) {

**return** "home";

}**else**

**return** **null**;

}

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

**protected** ModelAndView loginUser(HttpServletRequest request, @ModelAttribute("user") User user, BindingResult result) **throws** Exception {

HttpSession session = (HttpSession) request.getSession();

**try** {

System.***out***.println("loginUser");

User u = userDao.get(user.getUserName(),user.getPassword());

**if** (u != **null** && u.getStatus() == 1) {

session.setAttribute("user", u);

**if**(u.getRole().equalsIgnoreCase("admin") && *validateSession*(request))

**return** **new** ModelAndView("admin-home");

**else**

**return** **new** ModelAndView("user-home");

} **else** **if** (u != **null** && u.getStatus() == 0) {

**return** **new** ModelAndView("home", "errorMessage", "Your account hasn't been activated yet!");

} **else** {

**return** **new** ModelAndView("home", "errorMessage", "Invalid username/password!");

}

} **catch** (UserException e) {

System.***out***.println("Exception: " + e.getMessage());

request.setAttribute("errorMessage", "error while login");

**return** **new** ModelAndView("error", "errorMessage", "error while login");

}

}

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

**protected** ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("user") User user, BindingResult result) **throws** Exception {

validator.validate(user, result);

**if** (result.hasErrors()) {

**return** **new** ModelAndView("home", "user", user);

}

String captchacode=request.getParameter("captchaCode");

Captcha captcha=Captcha.*load*(request, "CaptchaObject");

HttpSession session=request.getSession();

**if**(captcha.validate(captchacode)) {

**try** {

System.***out***.print("registerNewUser");

User u = userDao.register(user);

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

session.setAttribute("user", u);

request.setAttribute("success", "Registration successfully done!");

**return** **new** ModelAndView("home", "user", u);

}

} **catch** (UserException e) {

System.***out***.println("Exception: " + e.getMessage());

**return** **new** ModelAndView("error", "errorMessage", "error while registration");

}

}**else** {

**return** **new** ModelAndView("home", "errorMessage", "Invalid Captcha. Re-enter again");

}

**return** **null**;

}

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

**protected** String logoutUser(HttpServletRequest request) {

HttpSession session=request.getSession();

session.invalidate();

**return** "home";

}

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

@ResponseBody

**public** String approveRejectRequest(HttpServletRequest request)

{

String result = **null**;

String username=request.getParameter("username");

// System.out.println("here"+request.getParameter("id"));

**try** {

**boolean** flag=userDao.getUserName(username);

**if**(flag) {

result="true";

}

**else** {

result="false";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println("here"+result);

**return** result;

}

**public** **static** **boolean** validateSession(HttpServletRequest request) {

HttpSession session = (HttpSession) request.getSession();

**if**(!(session.getAttribute("user")==**null**)) {

**return** **true**;

}

**else**

**return** **false**;

}

}

**AddPostController.java**

**package** com.me.finalproj.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

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

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

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

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

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

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

**import** com.me.finalproj.dao.CategoryDao;

**import** com.me.finalproj.exception.CategoryException;

**import** com.me.finalproj.pojo.Product;

@Controller

@RequestMapping("/add/\*")

**public** **class** AddPostController {

@Autowired

@Qualifier("categoryDao")

CategoryDao categoryDAO;

@Autowired

@Qualifier("product")

Product product;

@RequestMapping(value = "/add/post", method = RequestMethod.***GET***)

**protected** ModelAndView addPost(HttpServletRequest request) **throws** CategoryException {

**if**(*validateSession*(request)) {

request.setAttribute("product", product);

**return** **new** ModelAndView("addpost","categories", categoryDAO.list());

}**else** {

**return** **new** ModelAndView("home","errorMessage","Your Session Expired, kindly try to login again");

}

}

**public** **static** **boolean** validateSession(HttpServletRequest request) {

HttpSession session = (HttpSession) request.getSession();

**if**(!(session.getAttribute("user")==**null**)) {

**return** **true**;

}

**else**

**return** **false**;

}

}

**AddCategory.java**

**package** com.me.finalproj.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

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

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

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

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

**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.servlet.ModelAndView;

**import** com.me.finalproj.dao.CategoryDao;

**import** com.me.finalproj.exception.CategoryException;

**import** com.me.finalproj.pojo.Category;

**import** com.me.finalproj.validator.CategoryValidator;

@Controller

@RequestMapping("/add/category")

**public** **class** AddCategoryController {

@Autowired

@Qualifier("categoryValidator")

CategoryValidator categoryValidator;

@Autowired

@Qualifier("categoryDao")

CategoryDao categoryDAO;

@InitBinder

**private** **void** initBinder(WebDataBinder binder) {

binder.setValidator(categoryValidator);

}

@RequestMapping(value="/add/category", method = RequestMethod.***GET***)

**public** ModelAndView initializeForm() **throws** Exception {

**return** **new** ModelAndView("addcategory", "category", **new** Category());

}

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

**public** ModelAndView addCategory(HttpServletRequest request,@ModelAttribute("category") Category category, BindingResult result) **throws** Exception {

**if**(*validateSession*(request)) {

categoryValidator.validate(category, result);

**if** (result.hasErrors()) {

**return** **new** ModelAndView("addcategory", "category", category);

}

**try** {

category = categoryDAO.create(category.getName());

} **catch** (CategoryException e) {

System.***out***.println(e.getMessage());

**return** **new** ModelAndView("error", "errorMessage", "error while adding category");

}

**return** **new** ModelAndView("addcategory", "success", "Category Added successfully");

}

**else** {

**return** **new** ModelAndView("home","errorMessage","Your Session Expired, kindly try to login again");

}

}

**public** **static** **boolean** validateSession(HttpServletRequest request) {

HttpSession session = (HttpSession) request.getSession();

**if**(!(session.getAttribute("user")==**null**)) {

**return** **true**;

}

**else**

**return** **false**;

}

}

**HousingController.java**

**package** com.me.finalproj.controller;

**import** java.io.File;

**import** java.util.List;

**import** javax.servlet.ServletContext;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

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

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

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

**import** org.springframework.util.FileCopyUtils;

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

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

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

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

**import** org.springframework.web.multipart.MultipartFile;

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

**import** com.google.gson.Gson;

**import** com.me.finalproj.dao.HousingDao;

**import** com.me.finalproj.pojo.Housing;

**import** com.me.finalproj.pojo.Product;

**import** com.me.finalproj.pojo.User;

@Controller

@RequestMapping("\*\*/housing/\*\*")

**public** **class** HousingController {

@Autowired

@Qualifier("housingDao")

HousingDao housingDao;

@Autowired

ServletContext context;

@Autowired

@Qualifier("housing")

Housing housing;

@Autowired

@Qualifier("product")

Product product;

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

**protected** ModelAndView addHousing(HttpServletRequest request,@RequestParam("imagepath") MultipartFile file) **throws** Exception {

**if**(*validateSession*(request)) {

HttpSession session = (HttpSession) request.getSession();

// String uploadPath="F:\\MS Work\\SEM2\\Web Development\\Java\_Workspace\_Eclipse\\FinalProj\\data\\";

String uploadPath=context.getRealPath(File.***separator***)+"\\resources\\data\\";

// System.out.println(context.getContextPath());

System.***out***.println(uploadPath);

System.***out***.println(System.*getProperty*("user.dir"));

String title=request.getParameter("title");

String aptNo=request.getParameter("aptNo");

String street=request.getParameter("street");

**int** zipcode=Integer.*parseInt*(request.getParameter("zipcode"));

**int** quotation=Integer.*parseInt*(request.getParameter("quotation").substring(1));

housing.setTitle(title);

housing.setAptNo(aptNo);

housing.setStreet(street);

housing.setZipcode(zipcode);

housing.setQuotation(quotation);

housing.setImagepath("\\resources\\data\\"+file.getOriginalFilename());

housing.setUser((User)session.getAttribute("user"));

FileCopyUtils.*copy*(file.getBytes(), **new** File(uploadPath+file.getOriginalFilename()));

request.setAttribute("product", product);

housingDao.addHousing(housing);

**return** **new** ModelAndView("addpost","success","Housing Successfully Posted!!");

}**else** {

**return** **new** ModelAndView("home","errorMessage","Your Session Expired, kindly try to login again");

}

}

@RequestMapping(value = "/validate/housing", method = RequestMethod.***GET***)

**protected** String validateHousing(HttpServletRequest request) **throws** Exception {

**if**(*validateSession*(request)) {

**return** "housingAdmin";

}**else**

**return** **null**;

}

@RequestMapping(value="/validate/housing/ajaxload", method=RequestMethod.***GET***)

@ResponseBody

**public** String ajaxService(HttpServletRequest request)

{

String housingJson=**null**;

**if**(*validateSession*(request)) {

**try** {

List<Housing> housingList=housingDao.getPending();

housingJson = **new** Gson().toJson(housingList);

System.***out***.println(housingJson);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** housingJson;

}

@RequestMapping(value="/validate/housing/ajaxload", method=RequestMethod.***POST***)

@ResponseBody

**public** String approveRejectRequest(HttpServletRequest request)

{

String result = **null**;

**if**(*validateSession*(request)) {

**long** id=Long.*parseLong*(request.getParameter("id"));

String decision=request.getParameter("decision");

System.***out***.println("here"+request.getParameter("id"));

**try** {

**boolean** flag=housingDao.updateRequest(id,decision);

**if**(flag) {

result="done";

}

**else** {

result="error";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** result;

}

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

**protected** String searchHousing(HttpServletRequest request) **throws** Exception {

**return** "searchHousing";

}

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

@ResponseBody

**public** String searchAjax(HttpServletRequest request)

{

String housingJson=**null**;

**if**(*validateSession*(request)) {

**try** {

List<Housing> housingList=housingDao.getApproved();

housingJson = **new** Gson().toJson(housingList);

System.***out***.println(housingJson);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** housingJson;

}

@RequestMapping(value="/search/housing/searchByStreet", method=RequestMethod.***POST***)

@ResponseBody

**public** String searchStreet(HttpServletRequest request)

{

String housingJson=**null**;

**if**(*validateSession*(request)) {

String search=request.getParameter("search");

**try** {

List<Housing> housingList=housingDao.getSearched(search);

housingJson = **new** Gson().toJson(housingList);

System.***out***.println(housingJson);

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** housingJson;

}

**public** **static** **boolean** validateSession(HttpServletRequest request) {

HttpSession session = (HttpSession) request.getSession();

**if**(!(session.getAttribute("user")==**null**)) {

**return** **true**;

}

**else**

**return** **false**;

}

}

**ProductController.java**

**package** com.me.finalproj.controller;

**import** java.io.File;

**import** java.util.Arrays;

**import** java.util.HashSet;

**import** java.util.List;

**import** java.util.Set;

**import** javax.servlet.ServletContext;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpSession;

**import** org.codehaus.jettison.json.JSONArray;

**import** org.codehaus.jettison.json.JSONException;

**import** org.codehaus.jettison.json.JSONObject;

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

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

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

**import** org.springframework.util.FileCopyUtils;

**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** org.springframework.web.bind.annotation.ResponseBody;

**import** org.springframework.web.multipart.MultipartFile;

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

**import** com.me.finalproj.dao.CategoryDao;

**import** com.me.finalproj.dao.ProductDao;

**import** com.me.finalproj.pojo.Category;

**import** com.me.finalproj.pojo.Product;

**import** com.me.finalproj.pojo.User;

@Controller

@RequestMapping("\*\*/products/\*\*")

**public** **class** ProductController {

@Autowired

@Qualifier("productDao")

ProductDao productDao;

@Autowired

ServletContext context;

@Autowired

@Qualifier("product")

Product product;

@Autowired

@Qualifier("categoryDao")

CategoryDao categoryDAO;

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

**protected** ModelAndView addHousing(HttpServletRequest request,@RequestParam("imagepaths") MultipartFile file,@ModelAttribute("product") Product product) **throws** Exception {

**if**(*validateSession*(request)) {

HttpSession session = (HttpSession) request.getSession();

Set<Category> categories = **new** HashSet<Category>();

String uploadPath=context.getRealPath(File.***separator***)+"\\resources\\data\\";

String title=request.getParameter("title");

String description=request.getParameter("description");

String[] categoriesArray=product.getCategoryArray();

**int** quotation=Integer.*parseInt*(request.getParameter("quotations").substring(1));

**for**(String c:categoriesArray) {

Category category;

category=categoryDAO.get(c);

categories.add(category);

}

product.setTitle(title);

product.setDescription(description);

product.setQuotation(quotation);

product.setImagepath("\\resources\\data\\"+file.getOriginalFilename());

product.setUser((User)session.getAttribute("user"));

product.setCategories(categories);

FileCopyUtils.*copy*(file.getBytes(), **new** File(uploadPath+file.getOriginalFilename()));

productDao.addProduct(product);

**for**(Category c: product.getCategories()){

c = categoryDAO.get(c.getName());

c.getProducts().add(product);

categoryDAO.update(c);

}

**return** **new** ModelAndView("addpost","success","Product Successfully Posted!!");

}**else** {

**return** **new** ModelAndView("home","errorMessage","Your Session Expired, kindly try to login again");

}

}

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

**protected** ModelAndView searchHousing(HttpServletRequest request) **throws** Exception {

**if**(*validateSession*(request)) {

request.setAttribute("product", product);

**return** **new** ModelAndView("searchProduct","categories", categoryDAO.list());

}**else** {

**return** **new** ModelAndView("home","errorMessage","Your Session Expired, kindly try to login again");

}

}

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

@ResponseBody

**public** String searchAjax(HttpServletRequest request)

{

String productJson=**null**;

**if**(*validateSession*(request)) {

**try** {

List<Product> productList=productDao.getApproved();

JSONArray jArray = **null**;

// housingJson = new Gson().toJson(productList);

JSONObject jObject = **new** JSONObject();

**try**

{

jArray = **new** JSONArray();

**for** (Product product : productList)

{

JSONObject studentJSON = **new** JSONObject();

studentJSON.put("id", product.getId());

studentJSON.put("title", product.getTitle());

studentJSON.put("imagepath", product.getImagepath());

studentJSON.put("quotation", product.getQuotation());

jArray.put(studentJSON);

}

jObject.put("StudentList", jArray);

} **catch** (JSONException jse) {

}

System.***out***.println(jArray);

productJson=jArray.toString();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** productJson;

}

@RequestMapping(value="/search/products/searchByTitle", method=RequestMethod.***POST***)

@ResponseBody

**public** String searchByTitle(HttpServletRequest request)

{

String productJson=**null**;

**if**(*validateSession*(request)) {

String search=request.getParameter("search");

**try** {

List<Product> productList=productDao.getSearched(search);

JSONArray jArray = **null**;

// housingJson = new Gson().toJson(productList);

JSONObject jObject = **new** JSONObject();

**try**

{

jArray = **new** JSONArray();

**for** (Product product : productList)

{

JSONObject studentJSON = **new** JSONObject();

studentJSON.put("id", product.getId());

studentJSON.put("title", product.getTitle());

studentJSON.put("imagepath", product.getImagepath());

studentJSON.put("quotation", product.getQuotation());

jArray.put(studentJSON);

}

jObject.put("StudentList", jArray);

} **catch** (JSONException jse) {

}

System.***out***.println(jArray);

productJson=jArray.toString();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** productJson;

}

@RequestMapping(value = "/validate/products", method = RequestMethod.***GET***)

**protected** String validateHousing(HttpServletRequest request) **throws** Exception {

**if**(*validateSession*(request)) {

**return** "productAdmin";

}**else**

**return** **null**;

}

@RequestMapping(value="/validate/products/ajaxload", method=RequestMethod.***GET***)

@ResponseBody

**public** String ajaxService(HttpServletRequest request)

{

String housingJson=**null**;

**if**(*validateSession*(request)) {

**try** {

List<Product> productList=productDao.getPending();

JSONArray jArray = **null**;

// housingJson = new Gson().toJson(productList);

JSONObject jObject = **new** JSONObject();

**try**

{

jArray = **new** JSONArray();

**for** (Product product : productList)

{

JSONObject studentJSON = **new** JSONObject();

studentJSON.put("id", product.getId());

studentJSON.put("title", product.getTitle());

studentJSON.put("imagepath", product.getImagepath());

studentJSON.put("quotation", product.getQuotation());

jArray.put(studentJSON);

}

jObject.put("StudentList", jArray);

} **catch** (JSONException jse) {

}

System.***out***.println(jArray);

housingJson=jArray.toString();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** housingJson;

}

@RequestMapping(value="/validate/products/ajaxload", method=RequestMethod.***POST***)

@ResponseBody

**public** String approveRejectRequest(HttpServletRequest request)

{

String result = **null**;

**if**(*validateSession*(request)) {

**long** id=Long.*parseLong*(request.getParameter("id"));

String decision=request.getParameter("decision");

// System.out.println("here"+request.getParameter("id"));

**try** {

**boolean** flag=productDao.updateRequest(id,decision);

**if**(flag) {

result="done";

}

**else** {

result="error";

}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** result;

}

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

@ResponseBody

**public** String addFilter(HttpServletRequest request)

{

String categoryJson=**null**;

**if**(*validateSession*(request)) {

**try** {

List<Category> catList=categoryDAO.list();

JSONArray jArray = **null**;

// housingJson = new Gson().toJson(productList);

JSONObject jObject = **new** JSONObject();

**try**

{

jArray = **new** JSONArray();

**for** (Category category : catList)

{

JSONObject studentJSON = **new** JSONObject();

studentJSON.put("categoryId", category.getCategoryId());

studentJSON.put("name", category.getName());

studentJSON.put("products", category.getProducts());

jArray.put(studentJSON);

}

jObject.put("StudentList", jArray);

} **catch** (JSONException jse) {

}

System.***out***.println(jArray);

categoryJson=jArray.toString();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** categoryJson;

}

@RequestMapping(value="/search/products/filter", method=RequestMethod.***POST***)

@ResponseBody

**public** String filterContents(HttpServletRequest request,@RequestParam(value="clickedItem[]") String[] clickedItem)

{

String productJson=**null**;

**if**(*validateSession*(request)) {

Long[] clicked=**new** Long[clickedItem.length];

**for** (**int** i = 0; i < clicked.length; i++){

clicked[i] = Long.*parseLong*(clickedItem[i]);

System.***out***.println(clicked[i]);

}

List<Long> list = Arrays.*asList*(clicked);

**try** {

List<Product> productList;

**if**(clicked.length==0) {

productList=productDao.getApproved();

}**else** {

productList=productDao.getFiltered(list);

}

JSONArray jArray = **null**;

// housingJson = new Gson().toJson(productList);

JSONObject jObject = **new** JSONObject();

**try**

{

jArray = **new** JSONArray();

**for** (Product product : productList)

{

JSONObject studentJSON = **new** JSONObject();

studentJSON.put("id", product.getId());

studentJSON.put("title", product.getTitle());

studentJSON.put("imagepath", product.getImagepath());

studentJSON.put("quotation", product.getQuotation());

jArray.put(studentJSON);

}

jObject.put("StudentList", jArray);

} **catch** (JSONException jse) {

}

System.***out***.println(jArray);

productJson=jArray.toString();

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**return** productJson;

}

**public** **static** **boolean** validateSession(HttpServletRequest request) {

HttpSession session = (HttpSession) request.getSession();

**if**(!(session.getAttribute("user")==**null**)) {

**return** **true**;

}

**else**

**return** **false**;

}

}

* **POJO Classes**

**User.java**

package com.me.finalproj.pojo;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.PrimaryKeyJoinColumn;

import javax.persistence.Table;

@Entity

@Table(name = "user\_table")

@PrimaryKeyJoinColumn(name = "id")

public class User {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

@Column(name="id",unique=true,nullable=false)

private long id;

public User(String emailid,String username,String password) {

this.emailId=emailid;

this.userName=username;

this.password=password;

}

public User() {

}

@Column(name = "emailid")

private String emailId;

@Column(name = "username")

private String userName;

@Column(name = "password")

private String password;

@Column(name="status")

private int status;

@Column(name = "role")

private String role;

public String getEmailId() {

return emailId;

}

public void setEmailId(String emailId) {

this.emailId = emailId;

}

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 int getStatus() {

return status;

}

public void setStatus(int status) {

this.status = status;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

}

Product.java

**package** com.me.finalproj.pojo;

**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.ManyToOne;

**import** javax.persistence.Table;

**import** javax.persistence.Transient;

@Entity

@Table(name="product")

**public** **class** Product {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name="prod\_id", unique = **true**, nullable = **false**)

**private** **long** id;

@Column(name="title")

**private** String title;

@Column(name="status")

**private** String status;

@Column(name="quotation")

**private** **int** quotation;

@Column(name="description")

**private** String description;

@Column(name="imagepath")

**private** String imagepath;

@ManyToOne

**private** User user;

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

**private** Set<Category> categories = **new** HashSet<Category>();

@Transient

**private** String[] categoryArray;

**public** String[] getCategoryArray() {

**return** categoryArray;

}

**public** **void** setCategoryArray(String[] categoryArray) {

**this**.categoryArray = categoryArray;

}

**public** **long** getId() {

**return** id;

}

**public** **void** setId(**long** id) {

**this**.id = id;

}

**public** String getTitle() {

**return** title;

}

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

**this**.title = title;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** **int** getQuotation() {

**return** quotation;

}

**public** **void** setQuotation(**int** quotation) {

**this**.quotation = quotation;

}

**public** String getDescription() {

**return** description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** Set<Category> getCategories() {

**return** categories;

}

**public** **void** setCategories(Set<Category> categories) {

**this**.categories = categories;

}

**public** String getImagepath() {

**return** imagepath;

}

**public** **void** setImagepath(String imagepath) {

**this**.imagepath = imagepath;

}

}

Category.java

**package** com.me.finalproj.pojo;

**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.JoinColumn;

**import** javax.persistence.JoinTable;

**import** javax.persistence.ManyToMany;

**import** javax.persistence.Table;

**import** javax.transaction.Transactional;

@Entity

@Table(name="category")

**public** **class** Category {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name="categoryID", unique = **true**, nullable = **false**)

**private** **long** categoryId;

@Column(name="name", unique=**true**, nullable = **false**)

**private** String name;

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

@JoinTable (

name="category\_product\_table",

joinColumns = {@JoinColumn(name="categoryID", nullable = **false**, updatable = **false**)},

inverseJoinColumns = {@JoinColumn(name="prod\_id" )}

)

**private** Set<Product> products = **new** HashSet<Product>();

**public** **long** getCategoryId() {

**return** categoryId;

}

**public** **void** setCategoryId(**long** categoryId) {

**this**.categoryId = categoryId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Set<Product> getProducts() {

**return** products;

}

**public** **void** setProducts(Set<Product> products) {

**this**.products = products;

}

}

**Housing.java**

**package** com.me.finalproj.pojo;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.Table;

@Entity

@Table(name="housing")

**public** **class** Housing {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name="housingID", unique = **true**, nullable = **false**)

**private** **long** id;

@Column(name="title")

**private** String title;

@Column(name="aptno")

**private** String aptNo;

@Column(name="street")

**private** String street;

@Column(name="zipcode")

**private** **int** zipcode;

@Column(name="imagepath")

**private** String imagepath;

@Column(name="quotation")

**private** **int** quotation;

@Column(name="contactinfo")

**private** String contactinfo;

@ManyToOne

**private** User user;

@Column(name="status")

**private** String status;

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** **long** getId() {

**return** id;

}

**public** **void** setId(**long** id) {

**this**.id = id;

}

**public** String getTitle() {

**return** title;

}

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

**this**.title = title;

}

**public** String getAptNo() {

**return** aptNo;

}

**public** **void** setAptNo(String aptNo) {

**this**.aptNo = aptNo;

}

**public** String getStreet() {

**return** street;

}

**public** **void** setStreet(String street) {

**this**.street = street;

}

**public** **int** getZipcode() {

**return** zipcode;

}

**public** **void** setZipcode(**int** zipcode) {

**this**.zipcode = zipcode;

}

**public** String getImagepath() {

**return** imagepath;

}

**public** **void** setImagepath(String imagepath) {

**this**.imagepath = imagepath;

}

**public** String getContactinfo() {

**return** contactinfo;

}

**public** **void** setContactinfo(String contactinfo) {

**this**.contactinfo = contactinfo;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** **int** getQuotation() {

**return** quotation;

}

**public** **void** setQuotation(**int** quotation) {

**this**.quotation = quotation;

}

}