

Shiva Mahitha Maddi
NEU ID: 001061161



“SPICY FOOD”

ONLINE FOOD ORDER SYSTEM



Summary of the project:

The main purpose of this web application is to benefit both the customers and the restaurant's business. It enables customers to search the restaurant's menu and explore all the menu options. Customers can view the status of the orders they have placed and all their past orders. Customer can also book catering and can choose either in store pickup or home delivery. Restaurant Owner can receive and manage the customer's orders. It helps the restaurant owner to edit the menu of the restaurant and create accounts for delivery men. Delivery man can view the customer address and deliver the food items to the customer who has placed the order.



Technologies Used:

Front-End: JSP, Html, CSS, JS, jquery, Bootstrap, Ajax

Server-side: Spring MVC, Hibernate

Back-end: MYSQL

Server: Apache Tomcat

Concepts used:

Session Management

Asynchronous requests

Mail sending -- SMTP API

Download invoice (AbstractPdfView)

Search Autocomplete(Search Criteria)

Images upload to server(Multiple file upload to the server)

Criteria

One-Many bi-directional mapping

Interceptor

USER ROLES



Admin/Restaurant Owner



Delivery Men



Customers

FUNCTIONALITIES

Admin/Restaurant Owner

- Login
- Accept/Reject Orders
- Assign orders to delivery men
- Add/View/Edit/Delete Menu Items
- Accept Delivery Men sign up requests
- Delete Accounts of Delivery Men

Delivery Men

- Raise a registration request to the admin/owner
- Can login once registration request is approved by the owner
- Update his profile
- View all the orders assigned to him and change the status of the orders to delivered once delivered

FUNCTIONALITIES

Customers

- Signup and Login
- Search for menu items
- Add items to cart and remove from cart
- Checkout and place an order
- Cancel placed orders(only orders which are not yet assigned to delivery men by the restaurant owner)
- Track the current orders status
- View previous/delivered orders and provide rating/feedback for the food and the delivery service

FUNCTIONALITIES



FUNCTIONALITIE S





Burgers

Our juicy burgers are hand smashed and served with piping hot!

[View Details](#)

Snacks

Snacking does n't automatically mean weight gain.

[View Details](#)

Pizzas

We make our pizzas with the finest ingredients!!

[View Details](#)

Desserts

Taste these festive brownies for an easy holiday treat.

[View Details](#)

Login Below!

E-Mail Address

Password

☐ Remember Me

Login

[Forgot Your Password?](#)



Spicy Food Wanna Delivery food?

New user? Register below!

First Name

Last Name

Phone Number

Address

E-Mail Address

Password

Register

Already have an account? [login](#)

Thank you for registering to Spicy Food!! Your promo code inside!



mahithareddy42@gmail.com

to likithm87 ▾

Dear Likith Reddy Maddi,

Thank you for registering with us.

Use the promo code INS0412RL to enjoy happy first meal from us!!

We look forward to being part of your next memorable experience.

Spicy Food

↩ Reply

➡ Forward

- Admin Page

Delivery Men Accounts!!

User Id	FirstName	LastName	Phone Number	Email ID	Address	
26	Likith	Reddy	7813547028	likithm87@gmail.com	4 Kimball Court 114 Apartment number	Delete
28	Joy	James	2143658701	james@gmail.com	201 brook village road , apt 2	Delete
30	Arjun	Reddy	1345267890	arjun@gmail.com	201 brook village road , apt 2 114 Apartment number	Delete
31	Karthik	Yelihanka	1029837465	karthik@gmail.com	18/118/1-, Road No:05, Austin, US	Delete


Orders places by Customers!!

Order Id	Delivery Address	Items	Total Price	
19	Hunky street, 601 Apt Number, 30602	Butterscotch Cake Chicken Pizza Kofta rolls	40.0	Accept Reject
21	115 Great Avenue, New York	Orange juice Mocktail	60.0	Accept Reject
23	4 Kimball Court 114 Apartment number	Alu Tikka Kofta rolls Spicy chips	130.0	Accept Reject

Assign Delivery Men

Order Id	Delivery Address	Items	Total Price	
20	304,115 Apt , 30063	Chicken Pizza	36.0	Likith Reddy <input type="button" value="Assign"/>
22	201 brook village road , apt 2	Tomato Soup Green Pepper Orange lemonade Mocktail Strawberry Cake	140.0	Likith Reddy <input type="button" value="Assign"/>
27	405 brook village road , Rochester	Spicy chips	81.0	Likith Reddy <input type="button" value="Assign"/>

Admin Page


 **Spicy Food** **Orders** **Menu** **Delivery Men**

Sign Up requests

Approve Signup Requests

Delete Account

User Id	FirstName	LastName	Phone Number	Email ID
29	Steven	Kin	1098765432	kin@gmail
32	Vivek	Hunidi	1523467890	vivek@gmail
33	Vinoda	Reddy	1980234567	vinoda@gmail

 **Spicy Food** **Orders** **Menu** **Delivery Men**

Orders places

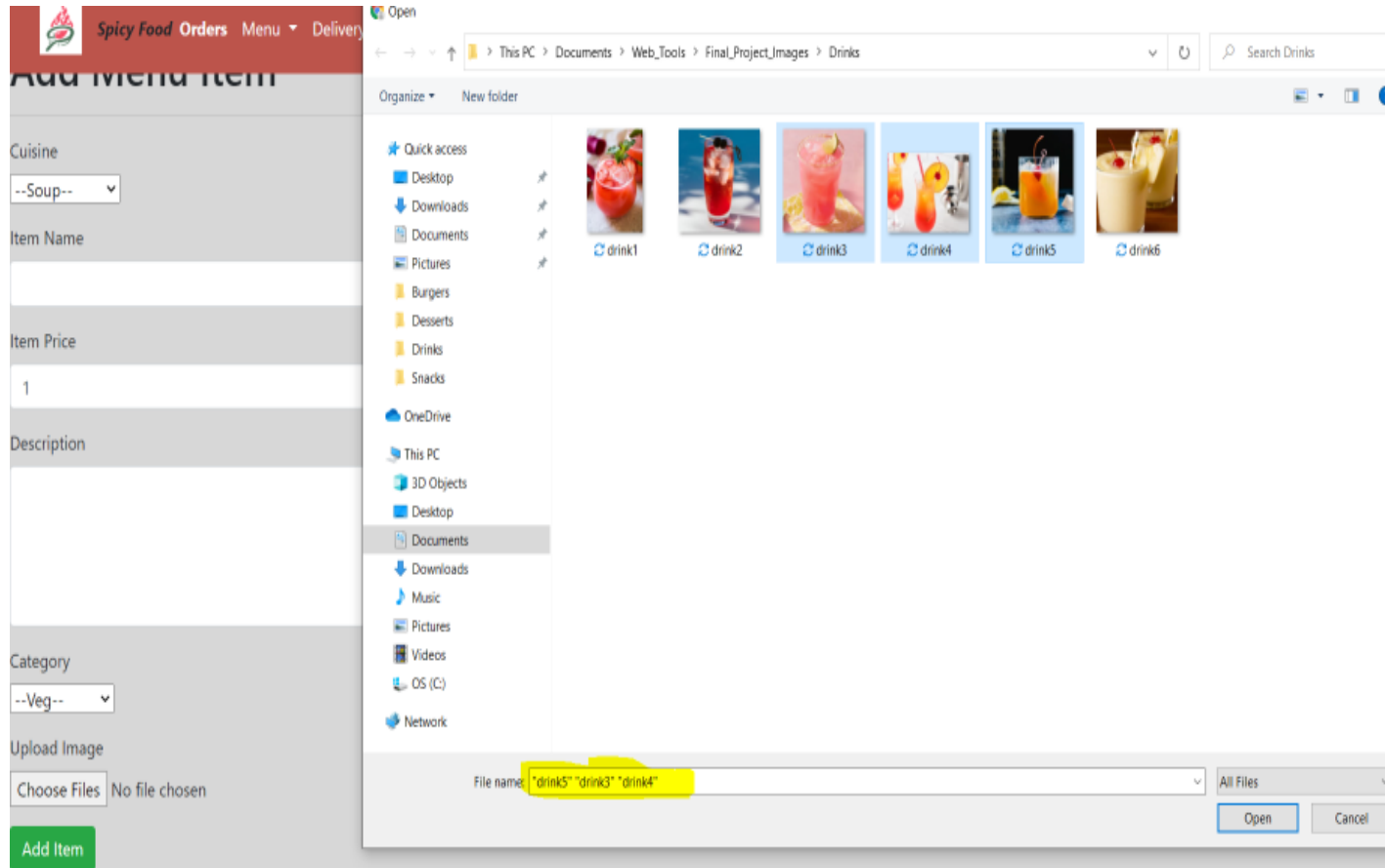
View/Edit Menu

Add Menu Item

Delete Menu Item

Order Id	Delivery Address
19	Hunky street, 601 Apt Number, 30602

Admin Page



MULTIPLE
FILE
UPLOAD TO
THE SERVER

Menu Items!!


Cuisine	Item Name	Item Price	Item Description	
soup	Tomato Soup	<input type="text" value="1.0"/>	<input type="text" value="Yummy tomato soup!!"/>	<input type="button" value="Edit"/>
soup	Potato Soup	<input type="text" value="2.0"/>	<input type="text" value="Yummy Potato Soup!!"/>	<input type="button" value="Edit"/>
soup	Carrot	<input type="text" value="1.0"/>	<input type="text" value="Carrot soup!!made with c"/>	<input type="button" value="Edit"/>
soup	Corn Soup	<input type="text" value="5.0"/>	<input type="text" value="Corn soupppl"/>	<input type="button" value="Edit"/>
soup	Green Pepper	<input type="text" value="4.0"/>	<input type="text" value="Made with green healthy"/>	<input type="button" value="Edit"/>
appetizer	Spring Rolls	<input type="text" value="4.0"/>	<input type="text" value="Crispy rolls with stuffed v"/>	<input type="button" value="Edit"/>

ADMIN PAGE

Cuisine	Item Name	Item Description	Item Price	
soup	Tomato Soup	Yummy tomato soup!!	1.0	<input type="button" value="Delete"/>
soup	Potato Soup	Yummy Potato Soup!!	2.0	<input type="button" value="Delete"/>
soup	Carrot	Carrot soup!!made with carrots!!	1.0	<input type="button" value="Delete"/>



• **Delivery Man Page**

 **Spicy Food** Wanna Delivery food?

Want to Delivery Food? Register below!

First Name

Last Name

Phone Number

Address

E-Mail Address

Password

Update profile below!!

Edit Phone Number:

2143658701

Edit Address:

201 brook village road , apt 2


Edit Password:

Update profile!!

Congratulations!! Your sign up request has been submitted!!

You will receive an mail once your request is approved!

[Go back to home page!](#)

 **Spicy Food**

Update Profile

View Orders ▾

Assigned

Delivered

Orders Assigned

OrderID	Order Items	Customer Name
---------	-------------	---------------




Orders Assigned!					
OrderID	Order Items	Customer Name	Delivery Address	Order Total Price	
16	Pista Cake Icecream rolls	mahitha maddi	201 brook village road , apt 2	10.0	Delivered

Orders delivered!						
OrderID	Order Items	Customer Name	Delivery Address	Order Total Price	Delivered Date	
16	Pista Cake Icecream rolls	mahitha maddi	201 brook village road , apt 2	10.0	2019-12-29 11:44:02.0	Delivered



- **Customer Page**

[Spicy Food](#) [Menu](#) [Previous Orders](#) [Track Orders](#) [Search Items](#) [View Cart](#) [Logout](#)

Provide Feedback

View Cancelled/Rejected Orders

Soups

Appetizers

Pizzas

Burgers

Snacks

Drinks


Desserts

ps

Tomato Soup \$1.0

Qty:


Add to Cart



Potato Soup \$2.0

Qty:

Add to Cart



- Mushroom pizza
- Chicken Pizza
- American Cheese Burger
- Chicken burger
- Spicy chips
- Banana Chips
- Butterscotch Cake
- Chocolate Cake
- Chocolate muffins
- Butterscotch Milkshake



[Go back to search page](#)

Mushroom pizza \$8.0

Qty:



Chicken Pizza \$9.0

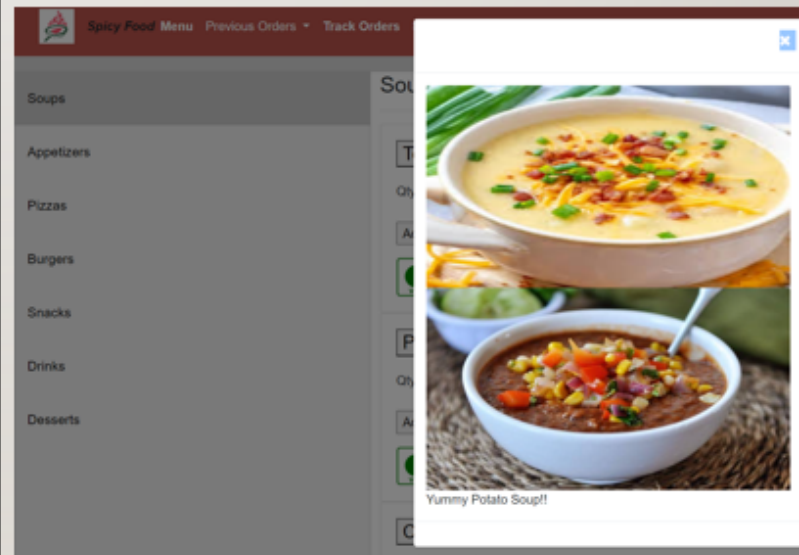
Qty:



American Cheese Burger \$5.0

Qty:

- Customer Page



Customer Page

orders

Search

localhost:8080 says
Successfully added to cart!

OK

Soups

Tomato Soup \$1.0

Qty: 1

Add to Cart

VEG

Checkout

Tomato Soup

Edit Qty: 3

Total Price: 3.0 Edit Qty

Remove from Cart

Spicy chips

Edit Qty: 7

Total Price: 63.0 Edit Qty

Remove from Cart

Checkout page!

Total Amount: 96.0

☐ Default Address

Kimball Court...

☒ Other Address

4 Kimball Court
114 Apartment

Payment Mode:

☒ Debit Card

☐ Credit Card

Card Details:

Name on Card: myCardName Card Number: myCardNumber

Expiration:

CVV:

Place Order

Spicy Food Menu Previous Orders Track Orders Search Items

Hurray!! Order is placed successfully!! Download invoice

download (3).pdf

File C:/Users/mahit/Downloads/do...

Invoice of the Customer

Item Name: Tomato Soup
Item Price: 1.0
Item Quantity: 3
Total Price: 3.0

Item Name: Spicy chips
Item Price: 9.0
Item Quantity: 7
Total Price: 63.0

Item Name: Orange lemonade
Item Price: 6.0
Item Quantity: 5
Total Price: 30.0

Total Amount Paid: 96.0

f t g+ in

© 2020 Copyright: LMNTrix

Customer Page

Order ID	Order Date	Total Price	Order Status	
31	2019-12-29 00:13:53.0	96.0	placed	Refund will be provided in 2 hours!! Cancel Order
32	2019-12-29 00:17:13.0	61.0	assigned Delivery Man Details: Name: LikithReddy Phone Number: 7813547028	Delivery Man is on the way!!
33	2019-12-29 00:17:59.0	51.0	accepted	Order has been accepted!! Delivery Man will be assigned soon!!

Order ID	Order Date	Total Price	Delivered Date	Ordered Items	Feedback
16	2019-12-29 03:53:32.0	10.0	2019-12-29 11:44:02.0	Pista Cake Icecream rolls	Not Worthy(*) Provide Feedback

Order ID	Order Date	Total Price	Order Status
15	2019-12-29 02:52:37.0	75.0	canceled
17	2019-12-29 04:20:36.0	20.0	rejected



Controllers Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.mycompany.controller;

import com.mycompany.dao.OrderDao;
import com.mycompany.dao.UserDao;
import com.mycompany.pojo.User;
import java.io.IOException;
import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.SimpleMailMessage;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;
```

```
/**
```

```
 *
```

```
 * @author mahit
```

```
 */
```

```
@Controller
```

```
public class AdminDeliveryManController {
```

```
    @Autowired
```

```
    private JavaMailSender mailSender;
```

```
    @GetMapping(value = "approveDelivRequests.htm")
```

```
    public String doGet(ModelMap model, UserDao userDao) {
```

```
        List<User> pendingRequests = userDao.getUsers("pending");
```

```
        List<User> acceptedRequests = userDao.getUsers("deliveryMan");
```

```
        if (pendingRequests == null) {
```

```
            model.addAttribute("error", "No pending requests!!");
```

```
        } else {
```

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

```
        }
```

```
model.addAttribute("pendingRequests", pendingRequests);  
model.addAttribute("acceptedRequests", acceptedRequests);  
return "approveDelivRequests";  
}
```

```
@PostMapping(value = "approveDelivRequests.htm")
```

```
public String doPost(ModelMap model, OrderDao orderDao, UserDao userDao, HttpServletRequest  
request, HttpServletResponse response, HttpSession session) throws IOException {
```

```
String optionValue = (String) request.getParameter("option");  
int userId = Integer.parseInt(request.getParameter("userId"));  
String emailId = (String) request.getParameter("emailId");  
//deliveryManName
```

```
SimpleMailMessage email = new SimpleMailMessage();  
if (optionValue.equals("accept")) {  
    userDao.update(userId, "deliveryMan");  
    // creates a simple e-mail object
```

```
email.setTo(emailId);  
String text = "Hi" + ",\n" + "\n"  
    + "Congratulations!!\n"
```

```
+ "\n"
+ "Your signup request for delivery man post has been approved!!\n"
+ "\n"
+ "We look forward to being part of your next memorable experience.\n"
+ "\n"
+ "Spicy Food\n" + "Management team";
email.setSubject("Get ready to deliver!! Thank you for registering to Spicy Food!! ");
email.setText(text);
```

```
// sends the e-mail
mailSender.send(email);
```

```
} else if (optionValue.equals("reject")) {
```

```
    userDao.update(userId, "undefined");
    // creates a simple e-mail object
```

```
    email.setTo(emailId);
    String text = "Hi" + ",\n" + "\n"
        + "We are sorry to inform you.!!\n"
        + "\n"
        + "Your signup request for delivery man post has been rejected.\n"
```



```
+ "\n"
+ "We look forward to connecting with you in the future.\n"
+ "\n"
+ "Spicy Food\n" + "Management team";
email.setSubject("Thank you for registering to Spicy Food!! ");
email.setText(text);

// sends the e-mail
mailSender.send(email);
}

List<User> pendingRequests = userDao getUsers("pending");
List<User> acceptedRequests = userDao getUsers("deliveryMan");
if (pendingRequests == null) {
    model.addAttribute("error", "No pending requests!!");
} else {
    model.addAttribute("error", "");
}
model.addAttribute("pendingRequests", pendingRequests);
model.addAttribute("acceptedRequests", acceptedRequests);
return "approveDelivRequests";
```

}

@GetMapping(value = "deleteAccounts.htm")

public String doGetdeleteAccounts(ModelMap model, OrderDao orderDao, UserDao userDao) {

List<User> deliveryMen = userDao.getUsers("deliveryMan");

model.addAttribute("deliveryMen", deliveryMen);

return "deleteAccounts";

}

@PostMapping(value = "deleteAccounts.htm")

public String doPostdeleteAccounts(ModelMap model, OrderDao orderDao, UserDao userDao,

HttpServletRequest request,

HttpServletResponse response, HttpSession session) throws IOException {

int userId = Integer.parseInt(request.getParameter("userId"));

int resultI = userDao.delete(userId);

List<User> deliveryMen = userDao.getUsers("deliveryMan");

model.addAttribute("deliveryMen", deliveryMen);

return "deleteAccounts";

}

}

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package com.mycompany.controller;
```

```
import com.mycompany.dao.MenuItemDao;  
import com.mycompany.dao.MenuItemImageDao;  
import com.mycompany.pojo.MenuItem;  
import com.mycompany.pojo.MenuItemImage;  
import com.mycompany.validator.AddMenuItemValidator;  
import com.mycompany.validator.EditMenuItemValidator;  
import java.io.File;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
import javax.servlet.ServletContext;  
import javax.servlet.http.HttpServletRequest;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;
```

```
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.support.SessionStatus;
import org.springframework.web.multipart.MultipartFile;
```

```
/**
 *
 * @author mahit
 */
@Controller
public class AdminMenuController {
```

```
    @Autowired
    ServletContext context;
```

```
    @Autowired
    AddMenuItemValidator addMenuItemValidator;
    //editMenuItemValidator
    @Autowired
```

EditMenuItemValidator editMenuItemValidator;

```
@GetMapping(value = "addMenuItem.htm")  
public String doGetAdd(ModelMap model, MenuItem menuItem) {  
    //form-backing object  
    menuItem.setCuisine("Soup");  
    menuItem.setCategory("Veg");  
    menuItem.setDescription("");  
    menuItem.setItemName("");  
    menuItem.setItemPrice(1);  
  
    model.addAttribute("menuItem", menuItem);  
    return "addMenuItem";  
}
```

```
@GetMapping(value = "deleteMenuItem.htm")  
public String doGetDelete(ModelMap model, MenuItem menuItem) {  
    MenuItemDao menuItemDao = new MenuItemDao();  
    List<MenuItem> menuItemsList = menuItemDao.getMenuItems();  
    model.addAttribute("menuItemsList", menuItemsList);  
    return "deleteMenuItem";  
}
```

```
@GetMapping(value = "viewAdminMenu.htm")  
public String doGetView(ModelMap model, MenuItem menuItem) {  
    MenuItemDao menuItemDao = new MenuItemDao();  
    List<MenuItem> menuItemList = menuItemDao.getMenuItems();  
    model.addAttribute("menuItemsList", menuItemList);  
    return "viewAdminMenu";  
}
```

```
@PostMapping(value = "deleteMenuItem.htm")  
public String doPostDelete(MenuItemDao menuItemDao, HttpServletRequest request,  
MenuItemImageDao menuItemImageDao) {  
    // MenuItemDao menuItemDao = new MenuItemDao();  
    Long itemId = Long.parseLong(request.getParameter("itemId"));  
    MenuItem menuItem = menuItemDao.getItem(itemId);  
    for (MenuItemImage menuItemImage : menuItem.getItemImages()) {  
        int res = menuItemImageDao.deleteMenuItemImage(menuItemImage.getItemImagesId());  
    }  
    int resultI = menuItemDao.deleteMenuItem(itemId);  
    List<MenuItem> menuItemList = menuItemDao.getMenuItems();  
    request.setAttribute("menuItemsList", menuItemList);  
    return "deleteMenuItem";
```



```
}
```

```
@PostMapping(value = "viewAdminMenu.htm")
```

```
public String doPostView(@ModelAttribute("menuItem") MenuItem menuItem, BindingResult result,
```

```
SessionStatus status,
```

```
HttpServletRequest request) {
```

```
editMenuItemValidator.validate(menuItem, result);
```

```
if (result.hasErrors()) {
```

```
    return "viewAdminMenu";
```

```
} else {
```

```
    MenuItemDao menuItemDao = new MenuItemDao();
```

```
    Long itemId = Long.parseLong(request.getParameter("itemId"));
```

```
    menuItemDao.updateMenuItemPrice(itemId, menuItem.getItemPrice());
```

```
    menuItemDao.updateMenuItemDescription(itemId, menuItem.getDescription());
```

```
    status.setComplete();
```

```
    List<MenuItem> menuItemList = menuItemDao.getItemList();
```

```
    request.setAttribute("menuItemList", menuItemList);
```

```
    return "viewAdminMenu";
```

```
}
```

```
}
```

```
@PostMapping(value = "addMenuItem.htm")  
public String doPostAdd(@RequestParam("file") ArrayList<MultipartFile> files,  
@ModelAttribute("menuItem") MenuItem menuItem, BindingResult result, SessionStatus status,  
ModelMap model, HttpServletRequest request, MenuItemImageDao menuItemImageDao) throws  
IOException {  
    addMenuItemValidator.validate(menuItem, result);  
    if (result.hasErrors()) {  
        return "addMenuItem";  
    } else {  
        MenuItemDao menuItemDao = new MenuItemDao();  
  
        int result2 = menuItemDao.add(menuItem);  
        MenuItem mi = menuItemDao.getItem(menuItem.getItemName());  
        for (MultipartFile file : files) {  
  
            String fileName = file.getOriginalFilename();  
            File imageFile = new File(request.getRealPath("/"), fileName);  
            try {  
                file.transferTo(imageFile);  
                MenuItemImage menuItemImage = new MenuItemImage();  
                menuItemImage.setImageName(file.getOriginalFilename());  
            }  
        }  
    }  
}
```

```
menuItemImage.setItemId(mi);  
mi.getItemImages().add(menuItemImage);
```

```
menuItemImageDao.add(menuItemImage);  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

```
}  
status.setComplete();  
menuItem.setCuisine("Soup");  
menuItem.setCategory("Veg");  
menuItem.setDescription("");  
menuItem.setItemName("");  
menuItem.setItemPrice(1);  
model.addAttribute("menuItem", menuItem);  
return "addMenuItem";
```

```
}
```

```
}
```

```
}
```

```
package com.mycompany.controller;

import com.mycompany.dao.OrderDao;
import com.mycompany.dao.UserDao;
import com.mycompany.pojo.Order;
import com.mycompany.pojo.User;
import java.io.IOException;
import java.text.ParseException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

/**
 *
 * @author mahit
```

***/**

@Controller

public class AdminOrdersController {

@GetMapping(value = "adminOrders.htm")

public String doGet(ModelMap model, OrderDao orderDao, UserDao userDao) {

List<Order> placedOrders = orderDao.getOrders("placed");

List<Order> acceptedOrders = orderDao.getOrders("accepted");

List<User> deliveryMen = userDao.getDeliveryMen();

if (placedOrders == null) {

model.addAttribute("error", "No orders placed!!");

} else {

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

}

model.addAttribute("placedOrders", placedOrders);

model.addAttribute("acceptedOrders", acceptedOrders);

model.addAttribute("deliveryMen", deliveryMen);

return "adminOrders";

}

@PostMapping(value = "adminOrders.htm")

```
public String doPost(ModelMap model,OrderDao orderDao,UserDao userDao, HttpServletRequest  
request, HttpServletResponse response, HttpSession session) throws IOException {
```

```
String optionValue = (String) request.getParameter("option");  
int orderId = Integer.parseInt(request.getParameter("orderId"));
```

```
if (optionValue.equals("accept")) {  
    try {  
        orderDao.update(orderId,"accepted");  
    } catch (ParseException ex) {  
        Logger.getLogger(AdminOrdersController.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

```
} else  
    if (optionValue.equals("reject")){  
        try {  
            orderDao.update(orderId,"rejected");  
        } catch (ParseException ex) {  
            Logger.getLogger(AdminOrdersController.class.getName()).log(Level.SEVERE, null, ex);  
        }  
    }  
}  
else{
```



```
int deliveryManId = Integer.parseInt(request.getParameter("deliveryManName"));

orderDao.update(orderId,deliveryManId,"assigned");
}

List<Order> placedOrders = orderDao.getOrders("placed");
List<Order> acceptedOrders= orderDao.getOrders("accepted");
List<User> deliveryMen=userDao.getDeliveryMen();
if (placedOrders == null) {
    model.addAttribute("error", "No orders placed!!");
} else {
    model.addAttribute("error", "");
}
model.addAttribute("placedOrders", placedOrders);
model.addAttribute("acceptedOrders", acceptedOrders);
model.addAttribute("deliveryMen", deliveryMen);

return "adminOrders";
}

}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package com.mycompany.controller;
```

```
import com.mycompany.View.MyPdfInvoiceView;  
import com.mycompany.dao.OrderDao;  
import com.mycompany.dao.OrderItemDao;  
import com.mycompany.dao.UserDao;  
import com.mycompany.pojo.DebitCard;  
import com.mycompany.pojo.Order;  
import com.mycompany.pojo.OrderItem;  
import com.mycompany.pojo.User;  
import com.mycompany.validator.CardValidator;  
import java.io.IOException;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.support.SessionStatus;
import org.springframework.web.servlet.View;
```

```
/**
 *
 * @author mahit
 */
@Controller
public class CheckoutController {
```

```
    @Autowired
    CardValidator cardValidator;
```

```
@GetMapping(value = "checkout.htm")
public String doGetCart(HttpServletRequest request, HttpSession session, DebitCard
debitCard, UserDao userDao) {
    ArrayList<OrderItem> list = (ArrayList<OrderItem>) session.getAttribute("items");

    if (list == null || list.size()==0) {
        request.setAttribute("error", "Cart is empty!");
        return "viewCart";
    }

    else{
        User user=(User)session.getAttribute("user");
        String customerAddress = userDao.getCustomerAddress(user.getEmailId());
        if(customerAddress!=null){
            request.setAttribute("customerAddress", customerAddress);
        }
        else{
            request.setAttribute("customerAddress", "no saved address! enter addresss!!");
        }
        return "checkout";
    }
}
```

```
}
```

```
@PostMapping(value = "checkout.htm")
```

```
public String doPostCart(@ModelAttribute("debitCard") DebitCard debitCard, BindingResult result,  
SessionStatus status,
```

```
Order order, HttpSession session, HttpServletRequest request, OrderDao orderDao,  
OrderItemDao orderItemDao) throws IOException, ParseException {
```

```
    cardValidator.validate(debitCard, result);
```

```
    if (result.hasErrors()) {
```

```
        return "checkout";
```

```
    }
```

```
    User user = (User) session.getAttribute("user");
```

```
    double total=0;
```

```
    ArrayList<OrderItem> orderItemsList = (ArrayList<OrderItem>) session.getAttribute("items");
```

```
    if (orderItemsList != null) {
```

```
        for (OrderItem orderItem : orderItemsList) {
```

```
            total=total+orderItem.getTotalPrice();
```

```
        }
```

```
    }
```

```
UserDao userDao=new UserDao();
User userI = (User)userDao.get(user.getEmailId());
int customerId=userI.getUserId();
String orderAddress;
String opt=(String)request.getParameter("radioBtn");
if(opt.equals("other")){
    orderAddress=(String)request.getParameter("addressI");
}
else{
    orderAddress=(String)request.getParameter("address");
}

order.setCustomerId(customerId);
order.setOrderAddress(orderAddress);
order.setOrderStatus("placed");
order.setOrderTotalPrice(total);
String pattern = "YYYY-MM-DD hh:mm:ss";
SimpleDateFormat simpleDateFormat = new SimpleDateFormat(pattern);

String date = simpleDateFormat.format(new Date());
Date dateI=new SimpleDateFormat("YYYY-MM-DD hh:mm:ss").parse(date);
order.setOrderDate(dateI); //////////////////////////////////////////
```



```
System.out.println("this is my order date" +date);//date
System.out.println("this is my order date" +date l);//date
orderDao.add(order);
int orderId=order.getOrderId();
int y=0;
if (orderItemsList != null) {

    for (OrderItem orderItem : orderItemsList) {
        orderItem.setOrderId(orderId);
        orderItemDao.add(orderItem);

    }
    ArrayList<OrderItem> orderItemsListl =orderItemsList;

    session.setAttribute("orderItemsListl", orderItemsListl);
    session.removeAttribute("items");

    status.setComplete();
    return "checkoutSuccess";
}
//create order record with this customer
```

```
    //create items and add order id to their columns  
    return "checkout";  
}
```

```
@RequestMapping(value = "/invoice.pdf", method = RequestMethod.GET)  
public View doGet(ModelMap modelMap, HttpSession session) {
```

```
    MyPdfInvoiceView myPdfInvoiceView = new MyPdfInvoiceView();
```

```
    return myPdfInvoiceView;
```

```
}
```

```
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package com.mycompany.controller;
```

```
import com.mycompany.dao.OrderDao;  
import com.mycompany.pojo.Order;  
import com.mycompany.pojo.User;  
import java.text.ParseException;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpSession;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;
```

```
/**
```

*** @author mahit**

***/**

@Controller

public class CustomerOrderTrackingController extends HttpServlet {

@GetMapping(value = "trackOrders.htm")

public String doGet(OrderDao orderDao, HttpSession session, HttpServletRequest request) {

User user = (User) session.getAttribute("user");

List<Order> customerOrders = orderDao.getcustomerOrders(user.getUserId());

if (customerOrders == null) {

request.setAttribute("error", "You have no current orders placed!!");

} else {

request.setAttribute("customerOrders", customerOrders);

}

return "trackOrders";

}

@PostMapping(value = "trackOrders.htm")

public String doPost(OrderDao orderDao, HttpSession session, HttpServletRequest request) {

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

if (orderId != null) {

```
int orderId=Integer.parseInt(orderIdI);
try {
    orderDao.update(orderId,"canceled");
} catch (ParseException ex) {
    Logger.getLogger(CustomerOrderTrackingController.class.getName()).log(Level.SEVERE, null,
ex);
}
}
User user = (User) session.getAttribute("user");
List<Order> customerOrders = orderDao.getcustomerOrders(user.getUserId());
if (customerOrders == null) {
    request.setAttribute("error", "You have no current orders placed!!");
} else {
    request.setAttribute("customerOrders", customerOrders);
}
return "trackOrders";
}
```

```
@GetMapping(value = "rejectedOrders.htm")
public String doGetI (OrderDao orderDao, HttpSession session, HttpServletRequest request) {
    User user = (User) session.getAttribute("user");
    List<Order> customerRejectedOrders = orderDao.getcustomerRejectedOrders(user.getUserId());
```

```
if (customerRejectedOrders == null) {  
    request.setAttribute("error", "You have no rejected/canceled orders placed!!");  
} else {  
    request.setAttribute("customerRejectedOrders", customerRejectedOrders);  
}  
return "rejectedOrdersCustomers";  
}
```

```
@GetMapping(value = "provideFeedback.htm")  
public String doGet2(OrderDao orderDao, HttpSession session, HttpServletRequest request) {  
    User user = (User) session.getAttribute("user");  
    List<Order> deliveredCustomerOrders = orderDao.getdeliveredCustomerOrders(user.getUserId());  
    if (deliveredCustomerOrders == null) {  
        request.setAttribute("error", "You have no orders delivered!!");  
    } else {  
        request.setAttribute("deliveredCustomerOrders", deliveredCustomerOrders);  
    }  
    return "provideFeedbackCustomers";  
}
```

```
@PostMapping(value = "provideFeedback.htm")  
public String doPost2(OrderDao orderDao, HttpSession session, HttpServletRequest request) {
```



```
String orderId=(String)request.getParameter("orderId");
String feedback=(String)request.getParameter("feedback");
if(orderId!=null){
    int orderId=Integer.parseInt(orderId);
    orderDao.updateFeedback(orderId,feedback);

}
User user = (User) session.getAttribute("user");
List<Order> deliveredCustomerOrders = orderDao.getdeliveredCustomerOrders(user.getUserId());
if (deliveredCustomerOrders == null) {
    request.setAttribute("error", "You have no orders delivered!!");
} else {
    request.setAttribute("deliveredCustomerOrders", deliveredCustomerOrders);
}
return "provideFeedbackCustomers";
}

}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package com.mycompany.controller;
```

```
import com.mycompany.dao.OrderDao;  
import com.mycompany.dao.UserDao;  
import com.mycompany.pojo.Order;  
import com.mycompany.pojo.User;  
import java.io.IOException;  
import java.text.ParseException;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.PostMapping;
```

```
/**
```

```
 *
```

```
 * @author mahit
```

```
 */
```

```
@Controller
```

```
public class DeliveryManOrdersController {
```

```
    @GetMapping(value = "ordersassigneddev.htm")
```

```
    public String doGet(ModelMap model, HttpSession session, OrderDao orderDao, UserDao userDao) {
```

```
        User deliveryMan=(User)session.getAttribute("user");
```

```
        int deliveryManId=deliveryMan.getUserId();
```

```
        List<Order> assignedOrders = orderDao.getDeliveryManOrders(deliveryManId,"assigned");
```

```
        if (assignedOrders == null) {
```

```
            model.addAttribute("error", "No orders assigned!!");
```

```
        } else {
```

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

```
        }
```

```
        model.addAttribute("assignedOrders", assignedOrders);
```

```
        return "ordersassigneddev";
```

```
    }
```

@PostMapping(value = "ordersassigneddev.htm")

public String doPost(ModelMap model, OrderDao orderDao, UserDao userDao, HttpServletRequest request, HttpServletResponse response, HttpSession session) throws IOException {

int orderId = Integer.parseInt(request.getParameter("orderId"));

**try {
 orderDao.update(orderId, "delivered");
} catch (ParseException ex) {
 Logger.getLogger(DeliveryManOrdersController.class.getName()).log(Level.SEVERE, null, ex);
}**

User deliveryMan=(User)session.getAttribute("user");

int deliveryManId=deliveryMan.getUserId();

List<Order> assignedOrders = orderDao.getDeliveryManOrders(deliveryManId, "assigned");

**if (assignedOrders == null) {
 model.addAttribute("error", "No orders assigned!!");
} else {
 model.addAttribute("error", "");
}**

model.addAttribute("assignedOrders", assignedOrders);

```
    return "ordersassigneddev";  
}
```

```
@GetMapping(value = "ordersdelivered.htm")  
public String doGet1 (ModelMap model,HttpSession session,OrderDao orderDao,UserDao userDao) {  
    User deliveryMan=(User)session.getAttribute("user");  
    int deliveryManId=deliveryMan.getUserId();  
    List<Order> deliveredOrders = orderDao.getDeliveryManOrders(deliveryManId,"delivered");  
    if (deliveredOrders == null) {  
        model.addAttribute("error", "No orders delivered!!");  
    } else {  
        model.addAttribute("error", "");  
    }  
    model.addAttribute("deliveredOrders", deliveredOrders);  
    return "ordersdelivered";  
}
```

```
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package com.mycompany.controller;
```

```
import com.mycompany.dao.UserDao;  
import com.mycompany.pojo.User;  
import com.mycompany.validator.UpdateDelivProfileValidator;  
import javax.servlet.http.HttpSession;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.validation.BindingResult;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.support.SessionStatus;
```

```
/**  
 *
```

*** @author mahit**

***/**

@Controller

public class DeliveryManUpdateProfileController {

@Autowired

UpdateDelivProfileValidator updateDelivProfileValidator;

@GetMapping(value = "updatedeliv.htm")

public String doGet(ModelMap model, User user3, HttpSession session) {

User user1=(User)session.getAttribute("user");

user3.setAddress(user1.getAddress());

user3.setPassword(user1.getPassword());

user3.setPhoneNum(user1.getPhoneNum());

model.addAttribute("user3",user3);

return "updatedeliv";

}

@PostMapping(value = "updatedeliv.htm")

public String doPost(@ModelAttribute("user3") User user3, BindingResult result, SessionStatus status, HttpSession session) {

updateDelivProfileValidator.validate(user3, result);

if (result.hasErrors()) {

```
        return "updatedeliv";
    } else {
        UserDao userDao = new UserDao();
        User user1=(User)session.getAttribute("user");
        int result1 = userDao.update(user1.getUserId(),user3);

        if (result1 != 1) {
            return "updatedeliv";
        }
        status.setComplete();
        return "loginSuccessDelivery";
    }
}
}
```



```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */
```

```
package com.mycompany.controller;
```

```
import com.google.gson.Gson;  
import com.mycompany.dao.MenuItemDao;  
import com.mycompany.pojo.MenuItem;  
import com.mycompany.pojo.OrderItem;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.List;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;
```

```
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;
```

```
/**
```

```
 *
```

```
 * @author mahit
```

```
 */
```

```
@Controller
```

```
public class MenuController extends HttpServlet {
```

```
    @GetMapping(value = "menu.htm")
```

```
    public String doGet(ModelMap model, MenuItemDao menuDao) {
```

```
        List<MenuItem> soupItemsList = menuDao.getItems("soup");
```

```
        List<MenuItem> appetizerItemsList = menuDao.getItems("appetizer");
```

```
        List<MenuItem> dessertItemsList = menuDao.getItems("dessert");
```

```
        List<MenuItem> pizzaItemsList = menuDao.getItems("pizza");
```

```
        List<MenuItem> burgerItemsList = menuDao.getItems("burger");
```

```
        List<MenuItem> snackItemsList = menuDao.getItems("snack");
```

```
        List<MenuItem> drinkItemsList = menuDao.getItems("drink");
```

```
model.addAttribute("souplItemsList", souplItemsList);
model.addAttribute("appetizerItemsList", appetizerItemsList);
model.addAttribute("dessertItemsList", dessertItemsList);
model.addAttribute("pizzalItemsList", pizzalItemsList);
model.addAttribute("burgerItemsList", burgerItemsList);
model.addAttribute("snackItemsList", snackItemsList);
model.addAttribute("drinkItemsList", drinkItemsList);
return "menu";
}
```

```
//@PostMapping(value = "addCart.htm")
```

```
@RequestMapping(value = "addCart.htm", method = RequestMethod.POST)
```

```
@ResponseBody
```

```
public String doPost(@RequestBody String validationResponse, HttpServletRequest request, HttpSession session, MenuItemDao menuDao, HttpServletResponse response) throws IOException {
```

```
    ArrayList<OrderItem> items;
```

```
    boolean flag = false;
```

```
    /** String selectedItemName = "item I";
```

```
    String selectedItemDescription = "My appetizer I";
```

```
    int qty = (4);
```

```
    **/
```

```
String[] arrOfStr = validationResponse.split("&", 5);
String[] itemNameArray=arrOfStr[0].split("=",2);
String[] QtyArray=arrOfStr[1].split("=",2);
String[] descriptionArray=arrOfStr[2].split("=",2);
String selectedItemName=itemNameArray[1].replace("+"," ");
int qty=Integer.parseInt(QtyArray[1].replace("+"," "));
String selectedItemDescription=descriptionArray[1].replace("+"," ");
```

```
// System.out.println(validationResponse);
// System.out.println(validationResponse.getItemName());
/**String selectedItemName = request.getParameter("itemName");
String selectedItemDescription = request.getParameter("description");
int qty = Integer.parseInt(request.getParameter("itemQty"));
////////////////////////////////////
if (qty == 0) {
    response.sendRedirect("menu.htm");
    return;
}

if (session == null) {
    response.sendRedirect("login.htm");
```

```
    return;
}**/

double itemPrice = menuDao.getItemPrice(selectedItemName);
if (session.getAttribute("items") != null) {
    items = (ArrayList<OrderItem>) session.getAttribute("items");
    for (OrderItem item : items) {
        if (item.getItemName().equals(selectedItemName)) {
            flag = true;
            item.setItemName(selectedItemName);
            item.setItemPrice(itemPrice);
            int qtyI = item.getQuantity() + qty;
            item.setQuantity(qtyI);
            item.setTotalPrice(itemPrice * qtyI);

            break;
        }
    }

    if (flag == false) {
        OrderItem orderItem = new OrderItem();
        orderItem.setItemName(selectedItemName);
    }
}
```

```
    orderItem.setItemPrice(itemPrice);  
    orderItem.setQuantity(qty);  
    orderItem.setTotalPrice(itemPrice * qty);  
    orderItem.setDescription(selectedItemDescription);  
    items.add(orderItem);  
}
```

```
} else {  
    items = new ArrayList<>();  
    OrderItem orderItem = new OrderItem();  
    orderItem.setItemName(selectedItemName);  
    orderItem.setItemPrice(itemPrice);  
    orderItem.setQuantity(qty);  
    orderItem.setTotalPrice(itemPrice * qty);  
    orderItem.setDescription(selectedItemDescription);  
  
    items.add(orderItem);  
}
```

```
session.setAttribute("items", items);  
//int x = 0;  
//response.sendRedirect("menu.htm");
```

```
//return "success";  
String string = "successful"; // The String which Need To Be Converted  
response.setStatus(HttpServletResponse.SC_ACCEPTED);  
return string;  
  
}  
  
@GetMapping(value = "viewCart.htm")  
public String doGetViewCart(HttpServletRequest request, HttpSession session) {  
    ArrayList<OrderItem> itemlist = (ArrayList<OrderItem>) session.getAttribute("items");  
  
    request.setAttribute("itemsI", itemlist);  
  
    ArrayList<OrderItem> list = (ArrayList<OrderItem>) session.getAttribute("items");  
    if (list == null || list.size() == 0) {  
        request.setAttribute("error", "Cart is empty!");  
    }  
    return "viewCart";  
}  
  
@PostMapping(value = "viewCart.htm")
```

```
public String doPostViewCart(HttpServletRequest request, HttpServletResponse response) throws
IOException {
    HttpSession session = request.getSession(false);
    String optionValue = (String) request.getParameter("optionValue");
    String itemName = request.getParameter("itemName");
    ArrayList<OrderItem> itemlist = (ArrayList<OrderItem>) session.getAttribute("items");
    if (optionValue.equals("editQty")) {
        int qty = Integer.parseInt(request.getParameter("itemQty"));
        if (itemlist != null) {
            for (OrderItem orderItem : itemlist) {
                if (orderItem.getItemName().equals(itemName)) {
                    orderItem.setQuantity(qty);
                    break;
                }
            }
            session.setAttribute("items", itemlist);
        }

        return "viewCart";
    }
}
```



```
if (itemlist != null) {  
    for (OrderItem orderItem : itemlist) {  
        if (orderItem.getItemName().equals(ItemName)) {  
            itemlist.remove(orderItem);  
            break;  
        }  
    }  
    session.setAttribute("items", itemlist);  
}  
ArrayList<OrderItem> list = (ArrayList<OrderItem>) session.getAttribute("items");  
if (list == null || list.size() == 0) {  
    request.setAttribute("error", "Cart is empty!");  
}  
return "viewCart";  
}
```

```
@GetMapping(value = "searchItems.htm")  
public String doGetsearchItems(HttpServletRequest request, HttpSession session) {  
    return "searchItems";  
}
```

```
@RequestMapping(value = "/search.htm", method = RequestMethod.GET)  
public @ResponseBody  
String doGetsearchItemsResults(HttpServletRequest request, HttpSession session,MenuItemDao  
menuItemDao){  
    Gson gson=new Gson();  
    List<String> list=menuItemDao.search(request.getParameter("term"));  
    return gson.toJson(list);  
}  
  
@PostMapping(value = "search.htm")  
public String doPostsearchItemsResults(HttpServletRequest request, HttpSession session,MenuItemDao  
menuItemDao ) {  
    String itemName=(String)request.getParameter("itemName");  
    List<MenuItem> menuItemList=menuItemDao.searchI(itemName);  
    request.setAttribute("menuItemList",menuItemList);  
    return "searchResults";  
}  
}
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package com.mycompany.controller;
```

```
import com.mycompany.dao.UserDao;  
import com.mycompany.pojo.ForgotPassword;  
import com.mycompany.pojo.User;  
import com.mycompany.validator.ForgotPasswordValidator;  
import com.mycompany.validator.LoginValidator;  
import com.mycompany.validator.UserValidator;  
import javax.servlet.http.Cookie;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.mail.SimpleMailMessage;  
import org.springframework.mail.javamail.JavaMailSender;  
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.support.SessionStatus;
```

```
/**
```

```
 *
```

```
 * @author mahit
```

```
 */
```

```
@Controller
```

```
public class UserController extends HttpServlet {
```

```
    @Autowired
```

```
    ForgotPasswordValidator forgotPasswordValidator;
```

```
    @Autowired
```

```
    UserValidator validator;
```

```
@Autowired  
LoginValidator loginValidator;
```

```
@Autowired  
private JavaMailSender mailSender;
```

```
@GetMapping(value = "signup.htm")  
public String doGet(ModelMap model, User user) {  
    return "signup";  
}
```

```
@PostMapping(value = "signup.htm")  
public String doPost(@ModelAttribute("user") User user, BindingResult result, SessionStatus status) {  
    validator.validate(user, result);  
    if (result.hasErrors()) {  
        return "signup";  
    } else {  
        user.setRole("customer");  
        UserDao userDao = new UserDao();  
        int resultI = userDao.add(user);  
  
        if (resultI != 1) {
```

```
    return "signup";
}
status.setComplete();
// creates a simple e-mail object
try{
SimpleMailMessage email = new SimpleMailMessage();
email.setTo(user.getEmailId());
String text = "Dear "+user.getFirstName()+" "+user.getLastName()+",\n" + "\n"
    + "Thank you for registering with us.\n"
    + "\n"
    + "Use the promo code INS04I2RL to enjoy happy first meal from us!!\n"
    + "\n"
    + "We look forward to being part of your next memorable experience.\n"
    + "\n"
    + "Spicy Food";
email.setSubject(" Thank you for registering to Spicy Food!! Your promo code inside!");
email.setText(text);

// sends the e-mail
mailSender.send(email);
}
catch(org.springframework.mail.MailAuthenticationException e){
```

```
    }  
    return "signupSuccess";  
}  
}
```

```
@GetMapping(value = "updatepassword.htm")  
public String doGetUpdatepassword(ModelMap model, ForgotPassword forgotPassword) {  
    return "updatepassword";  
}
```

```
@PostMapping(value = "updatepassword.htm")  
public String doPostUpdatepassword(@ModelAttribute("forgotPassword")ForgotPassword  
forgotPassword, BindingResult result, SessionStatus status) {  
    forgotPasswordValidator.validate(forgotPassword, result);  
    if (result.hasErrors()) {  
        return "updatepassword";  
    } else {  
        UserDao userDao = new UserDao();  
        int resultI = userDao.updatePassword(forgotPassword.getEmailId(),forgotPassword.getPassword());
```

```
        if (result != 1) {  
            return "updatepassword";  
        }  
        status.setComplete();  
        return "updateSuccess";  
    }  
}
```

```
@GetMapping(value = "signupDelivery.htm")  
public String doGetSignupDelivery(ModelMap model, User user) {  
    return "signupDelivery";  
}
```

```
@PostMapping(value = "signupDelivery.htm")  
public String doPostSignupDelivery(@ModelAttribute("user") User user, BindingResult result,  
SessionStatus status) {  
    validator.validate(user, result);  
    if (result.hasErrors()) {  
        return "signupDelivery";  
    } else {  
        user.setRole("pending");  
        UserDao userDao = new UserDao();
```



```
int resultI = userDao.add(user);

if (resultI != 1) {
    return "signupDelivery";
}
status.setComplete();
return "signupDeliverySuccess";
}
}
```

```
@GetMapping(value = "login.htm")
public String doGetLogin(ModelMap model, User user, UserDao userDao, HttpServletRequest request,
HttpSession session) {
    Cookie[] cookies = request.getCookies();

    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookie.getName().equals("userCookie")) {
                String cookieValue = cookie.getValue();
                String[] array = cookieValue.split("%20");
                String emailId = array[0];
                String password = array[1];
            }
        }
    }
}
```

```
//populateLoggedInUserPage(name, response);
User userI = userDao.get(emailId);
session.setAttribute("user", userI);
if (userI.getRole().equals("customer")) {
    return "loginSuccess";
} else if (userI.getRole().equals("admin")) {
    return "loginSuccessAdmin";
} else if (userI.getRole().equals("deliveryMan")) {
    return "loginSuccessDelivery";
}
}
}
}

return "login";
}
```

```
@PostMapping(value = "login.htm")
public String doPostLogin(@ModelAttribute("user") User user,
    BindingResult result, SessionStatus status, HttpServletResponse response, HttpServletRequest
request, HttpSession session) {
    loginValidator.validate(user, result);
```

```
if (result.hasErrors()) {  
    return "login";  
}  
String remember = request.getParameter("remember");  
String userCookie = user.getEmailId() + "%20" + user.getPassword();  
  
if (remember != null) {  
    Cookie userCookieI = new Cookie("userCookie", userCookie);  
    userCookieI.setMaxAge(365 * 24 * 60 * 60);  
    response.addCookie(userCookieI);  
}  
UserDao userDao = new UserDao();  
User userI = userDao.get(user.getEmailId());  
session.setAttribute("user", userI);  
status.setComplete();  
if (userI.getRole().equals("customer")) {  
    return "loginSuccess";  
} else if (userI.getRole().equals("admin")) {  
    return "loginSuccessAdmin";  
} else {  
    return "loginSuccessDelivery";  
}
```

```
}
```

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

```
public String logout(HttpServletRequest request, HttpSession session) {
```

```
    if (session != null) {
```

```
        session.removeAttribute("user");
```

```
        session.invalidate();
```

```
    }
```

```
    return "redirect";
```

```
}
```

```
}
```