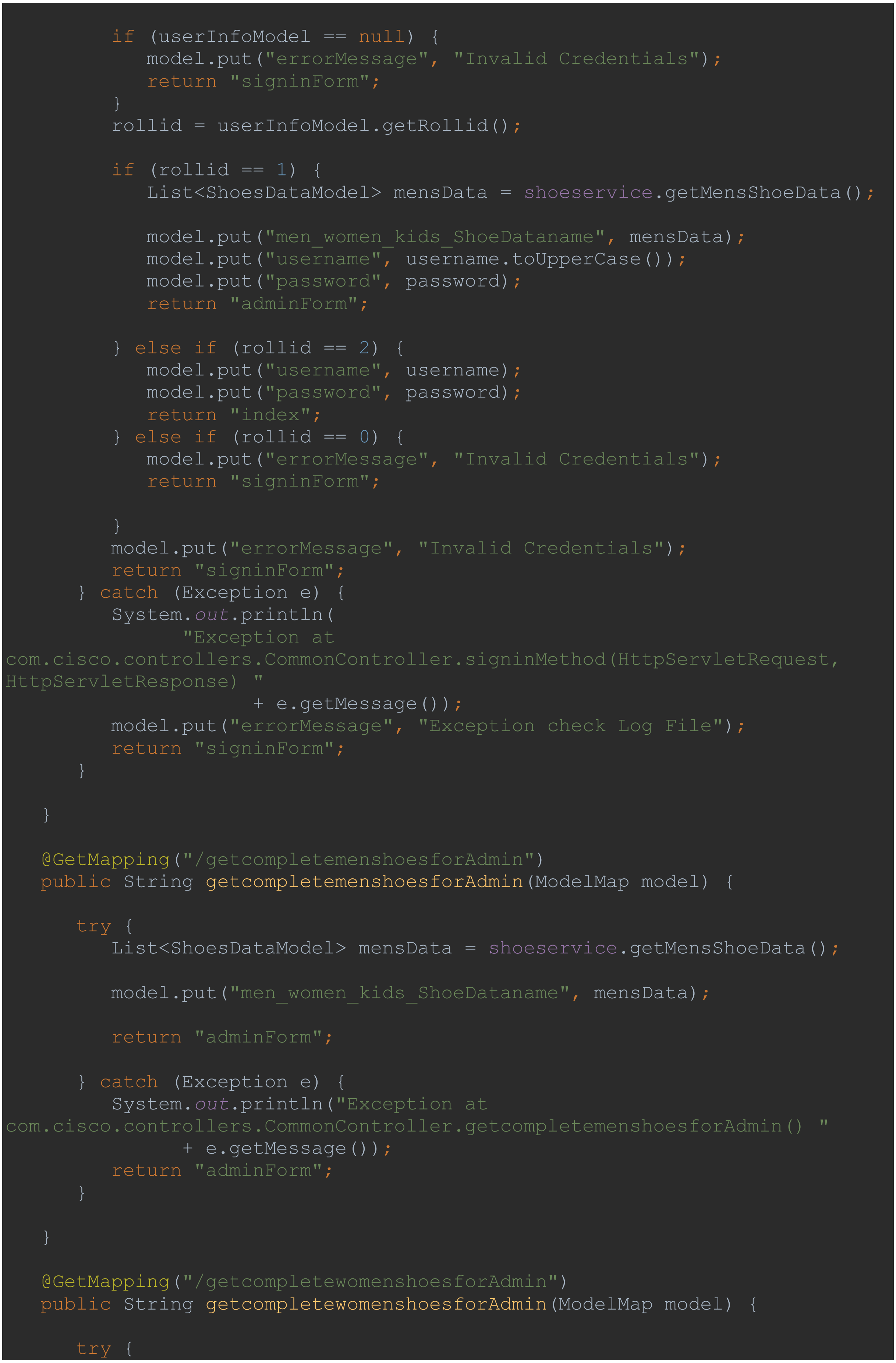
Source Code:

- ### Controller

CommonController.java

```java
package com.cisco.controller;

import java.sql.Date;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
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.SessionAttributes;
import org.springframework.web.servlet.ModelAndView;

import com.cisco.model.OrderedShoeModel;
import com.cisco.model.ShoesDataModel;
import com.cisco.model.UserInfoModel;
import com.cisco.service.OrdersService;
import com.cisco.service.ShoesService;
import com.cisco.service.UsersService;

@Controller

@RequestMapping("/")
public class CommonController {

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

   public String homeForm() {
      return "index";
   }

   @Autowired
   private ShoesService shoeservice;
   @Autowired
   private OrdersService orderservice;
   @Autowired
   private UsersService userservice;

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

   public String menShoesForm(ModelMap model) {
      List<ShoesDataModel> mensData = shoeservice.getMensShoeData();
      model.put("menshoeData", mensData);

      return "menShoesForm";
   }

   @RequestMapping(value = "/womenshoes", method = RequestMethod.GET)
   public String womenMethod(ModelMap model) {
        List<ShoesDataModel> womensData = shoeservice.getWomensShoeData();
        model.put("womenshoeData", womensData);

        return "womenShoesForm";
    }

    @RequestMapping("/kidshoes")
    public String kidMethod(ModelMap model) {

        List<ShoesDataModel> kidsData = shoeservice.getKidsShoeData();
        model.put("kidshoeData", kidsData);

        return "kidsShoesForm";
    }

    @RequestMapping(value = "/signin", method = RequestMethod.GET)
    public String signinMethod() {
        return "signinForm";
    }

    @RequestMapping(value = "/signup", method = RequestMethod.GET)
    public String sigUpMethod() {
        return "signUpForm";
    }

    @RequestMapping("/selectedShoe")
    public String selectedShoeInfotMethod(ModelMap model, @RequestParam int
id) {

        try {

            ShoesDataModel selectedShoesData =
shoeservice.getshoesDataByIdService(id);

            model.put("selectedShoesData", selectedShoesData);
            return "selectedShoeInfoForm";

        } catch (Exception e) {
            System.out.println(
                    "Exception at
com.cisco.controllers.CommonController.selectedShoeInfotMethod(HttpServletR
equest, HttpServletResponse) "
                            + e.getMessage());
            return "selectedShoeInfoForm";
        }

    }

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

    public String orderInforMethod(ModelMap model, @RequestParam int id,
@RequestParam String name,
            @RequestParam int categeory, @RequestParam double price,
@RequestParam String imglink,
            @RequestParam int quantity, @RequestParam int size) {

        try {

            OrderedShoeModel orderedshoemodel = new OrderedShoeModel();

            double totalPrice = price * quantity;
```

```
         orderedshoemodel.setShoeid(id);
         orderedshoemodel.setName(name);
         orderedshoemodel.setCategeory(categeory);
         orderedshoemodel.setPrice(price);
         orderedshoemodel.setImagelink(imglink);
         orderedshoemodel.setQuantity(quantity);
         orderedshoemodel.setTotalprice(totalPrice);
         orderedshoemodel.setSize(size);

         model.put("orderedshoemodel", orderedshoemodel);

         return "bookShoeForm";

      } catch (Exception e) {
         System.out.println(
               "Exception at 
com.cisco.controllers.CommonController.orderInforMethod(HttpServletRequest, 
HttpServletResponse) "
                     + e.getMessage());
         return "bookShoeForm";
      }

   }

   @GetMapping("/bookandpaymentreturntoHome")
   public String bookAndPaymentreturntoHomeMethod(ModelMap model, 
@RequestParam(name = "id") int shoeid,
         @RequestParam String name, @RequestParam int categeory, 
@RequestParam double price,
         @RequestParam String imglink, @RequestParam int quantity, 
@RequestParam double totalprice, @RequestParam int size) {

      try {

         long millis = System.currentTimeMillis();
         Date date = new java.sql.Date(millis);

         orderservice.insertBookingInfoService(shoeid, name, categeory, 
price, imglink, quantity, totalprice, date, size);

         return "orderplaceForm";
      } catch (Exception e) {
         System.out.println(
               "Exception at 
com.cisco.controllers.CommonController.bookAndPaymentreturntoHomeMethod(Htt
pServletRequest, HttpServletResponse) "
                     + e.getMessage());
         return "orderplaceForm";
      }
   }

   @GetMapping("/adminsignin")
   public String adminSigninMethod(ModelMap model, @RequestParam String 
username, @RequestParam String password) {

      int rollid = 0;
      try {

         UserInfoModel userInfoModel = 
userservice.isValidAdminUserService(username, password);
```

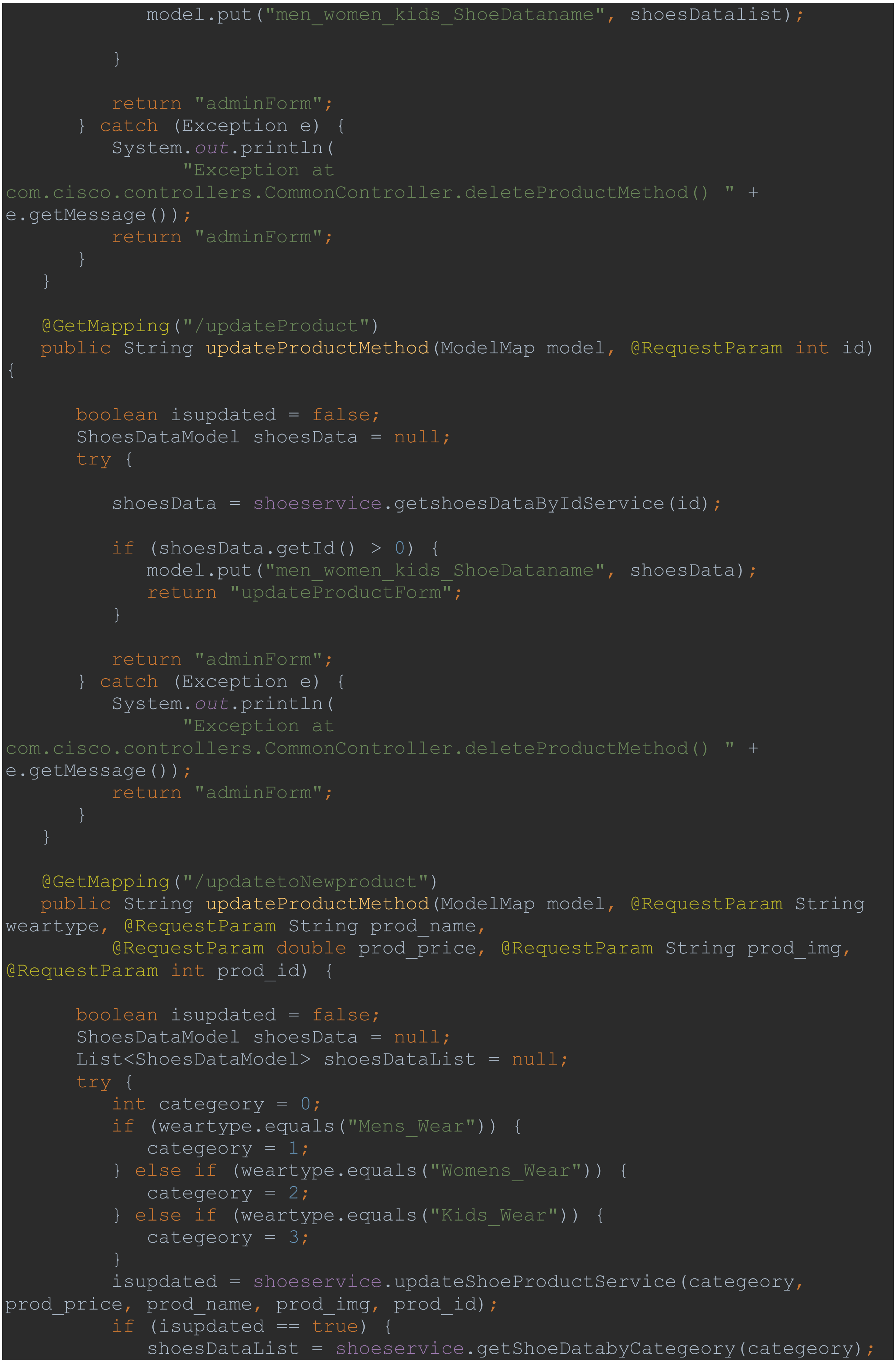


```
            if (userInfoModel == null) {
                model.put("errorMessage", "Invalid Credentials");
                return "signinForm";
            }
            rollid = userInfoModel.getRollid();

            if (rollid == 1) {
                List<ShoesDataModel> mensData = shoeservice.getMensShoeData();

                model.put("men_women_kids_ShoeDataname", mensData);
                model.put("username", username.toUpperCase());
                model.put("password", password);
                return "adminForm";

            } else if (rollid == 2) {
                model.put("username", username);
                model.put("password", password);
                return "index";
            } else if (rollid == 0) {
                model.put("errorMessage", "Invalid Credentials");
                return "signinForm";

            }
            model.put("errorMessage", "Invalid Credentials");
            return "signinForm";
        } catch (Exception e) {
            System.out.println(
                    "Exception at
com.cisco.controllers.CommonController.signinMethod(HttpServletRequest,
HttpServletResponse) "
                            + e.getMessage());
            model.put("errorMessage", "Exception check Log File");
            return "signinForm";
        }

    }

    @GetMapping("/getcompletemenshoesforAdmin")
    public String getcompletemenshoesforAdmin(ModelMap model) {

        try {
            List<ShoesDataModel> mensData = shoeservice.getMensShoeData();

            model.put("men_women_kids_ShoeDataname", mensData);

            return "adminForm";

        } catch (Exception e) {
            System.out.println("Exception at
com.cisco.controllers.CommonController.getcompletemenshoesforAdmin() "
                    + e.getMessage());
            return "adminForm";
        }

    }

    @GetMapping("/getcompletewomenshoesforAdmin")
    public String getcompletewomenshoesforAdmin(ModelMap model) {

        try {
         List<ShoesDataModel> womensData = shoeservice.getWomensShoeData();

         model.put("men_women_kids_ShoeDataname", womensData);

         return "adminForm";

      } catch (Exception e) {
         System.out.println("Exception at 
com.cisco.controllers.CommonController.getcompletewomenshoesforAdmin() "
               + e.getMessage());
         return "adminForm";
      }

   }

   @GetMapping("/getcompletekidshoesforAdmin")
   public String getcompletekidshoesforAdmin(ModelMap model) {

      try {
         List<ShoesDataModel> kidsData = shoeservice.getKidsShoeData();

         model.put("men_women_kids_ShoeDataname", kidsData);

         return "adminForm";

      } catch (Exception e) {
         System.out.println("Exception at 
com.cisco.controllers.CommonController.getcompletekidshoesforAdmin() "
               + e.getMessage());
         return "adminForm";
      }
   }

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

   public String changeAdminPasswordForm() {
      return "changeAdminPasswordForm";
   }

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

   public String addNewProductForm() {
      return "addNewProductForm";
   }

   @GetMapping("/customersList")
   public String customersListMethod(ModelMap model) {
      List<UserInfoModel> usersList = null;

      try {

         usersList = userservice.getUserDataService();

         if (!usersList.isEmpty()) {

            model.put("usersList", usersList);
         } else {
            System.out.println("No Users Found...");
         }
```

```java
         return "customersListForm";

      } catch (Exception e) {
         System.out.println(
               "Exception at
com.cisco.controllers.CommonController.customersListMethod(HttpServletReque
st, HttpServletResponse) "
                     + e.getMessage());
         return "customersListForm";
      }

   }

   @GetMapping("/reports")
   public String reportsMethod(ModelMap model) {

      List<OrderedShoeModel> orderedShoeList = null;
      try {

         orderedShoeList =
orderservice.getCompleteTransactionsDataService();

         if (orderedShoeList != null) {

            double totalSales = 0;
            for (OrderedShoeModel osl : orderedShoeList) {
               totalSales = totalSales + osl.getTotalprice();
            }

            model.put("orderedShoeList", orderedShoeList);
            model.put("totalSales", totalSales);

         }
         return "reportsForm";
      } catch (Exception e) {
         System.out.println("Exception at
com.cisco.controllers.CommonController.reportsMethod() " + e.getMessage());
         return "reportsForm";
      }
   }

   @GetMapping("/changeadminPassword")
   public String changeadminPasswordMethod(ModelMap model, @RequestParam
String currentpassword,
         @RequestParam String newpassword, @RequestParam String
confirmpassword) {

      UserInfoModel userInfoModel = null;
      boolean isCoreectPassword = false;
      try {

         userInfoModel =
userservice.isCorrectPassword_or_NotService(currentpassword);
         if (userInfoModel.getRollid() != 0) {
            boolean isUpdatedAdminPassword =
userservice.updateAdminPasswordService(userInfoModel, currentpassword,
                  newpassword, confirmpassword);
            if (isUpdatedAdminPassword == true) {
               System.out.println("Password Updated");
               return "signinForm";
            } else {
               System.out.println("Password Not Updated");
            }
         }

         else {
            System.out.println("No Data Found");
         }

         return "changeAdminPasswordForm";

      } catch (Exception e) {
         System.out.println(
               "Exception at 
com.cisco.controllers.CommonController.changeadminPasswordMethod(HttpServle
tRequest, HttpServletResponse) "
                     + e.getMessage());
         return "changeAdminPasswordForm";
      }

   }

   @GetMapping("/requiredreports")
   public String requiredReportsMethod(ModelMap model, @RequestParam String 
categeory, @RequestParam String date) {

      List<OrderedShoeModel> orderedShoeList = null;
      try {

         int categeoryId = 0;
         String categeoryName = categeory;
         if (categeoryName.equals("mens"))
            categeoryId = 1;
         else if (categeoryName.equals("womens"))
            categeoryId = 2;
         else if (categeoryName.equals("kids"))
            categeoryId = 3;
         String jspdate = date;

         Date sqldate = Date.valueOf(jspdate);

         if (categeoryId != 0) {
            orderedShoeList = 
orderservice.getRequiredCompleteTransactionsDataService(categeoryId, 
sqldate);
            if (orderedShoeList != null) {
               double totalSales = 0;
               for (OrderedShoeModel osl : orderedShoeList) {
                  totalSales = totalSales + osl.getTotalprice();
               }

               model.put("orderedShoeList", orderedShoeList);
               model.put("totalSales", totalSales);
               return "reportsForm";
            }
         }
         return "reportsForm";
      } catch (Exception e) {
         System.out.println(
               "Exception at 
com.cisco.controllers.CommonController.requiredReportsMethod() " + 
e.getMessage());
```

```
         return "reportsForm";
      }
   }

   @RequestMapping("/addnewproduct")
   public String addNewProductMethod(ModelMap model, @RequestParam String
weartype, @RequestParam String prod_name,
         @RequestParam double prod_price, @RequestParam String prod_img) {

      ModelAndView mv = null;
      try {

         int categeroy = 0;
         if (weartype.equals("Mens_Wear")) {
            categeroy = 1;
         } else if (weartype.equals("Womens_Wear")) {
            categeroy = 2;
         } else if (weartype.equals("Kids_Wear")) {
            categeroy = 3;
         }

         boolean isInsertedNewProduct = false;
         isInsertedNewProduct =
shoeservice.insertNewProductService(categeroy, prod_price, prod_name,
prod_img);
         if (isInsertedNewProduct == true) {
            if (categeroy == 1) {
               List<ShoesDataModel> mensData =
shoeservice.getMensShoeData();

               model.put("men_women_kids_ShoeDataname", mensData);
               return "adminForm";
            }

            if (categeroy == 2) {
               List<ShoesDataModel> womensData =
shoeservice.getWomensShoeData();

               model.put("men_women_kids_ShoeDataname", womensData);
               return "adminForm";
            }
            if (categeroy == 3) {
               List<ShoesDataModel> kidsData =
shoeservice.getKidsShoeData();

               model.put("men_women_kids_ShoeDataname", kidsData);
               return "adminForm";
            }

         } else {
            System.out.println("Product Not Added");
         }
         return "adminForm";

      } catch (Exception e) {
         System.out.println(
               "Exception at
com.cisco.controllers.CommonController.addNewProducttMethod(HttpServletRequ
est, HttpServletResponse) "
                     + e.getMessage());
         return "adminForm";
```

```
        }

    }

    @GetMapping("/signupnewuser")
    public String sigUpNewUserMethod(ModelMap model, @RequestParam String
firstname, @RequestParam String lastname,
            @RequestParam String password, @RequestParam String
confirmpassword, @RequestParam String roll,
            @RequestParam Long mobileno, @RequestParam String email) {
        ModelAndView mv = null;
        boolean isInserted = false;
        try {
            int rollid = 0;

            if (roll.equals("admin")) {
                rollid = 1;
            } else {
                rollid = 2;
            }

            isInserted = true;
            userservice.insertUserDataService(firstname, lastname, password,
confirmpassword, rollid, mobileno, email);
            if (isInserted == true) {

                return "signinForm";
            }

            if (isInserted == false) {

                return "testErrorForm";
            }
            return "testErrorForm";

        } catch (Exception e) {
            System.out.println(
                    "Exception at
com.cisco.controllers.CommonController.sigUpNewUserMethod(HttpServletReques
t, HttpServletResponse) "
                            + e.getMessage());
            return "testErrorForm";
        }

    }

    @GetMapping("/deleteProduct")
    public String deleteProductMethod(ModelMap model, @RequestParam int id)
{

        boolean isdeleted = false;
        List<ShoesDataModel> shoesDatalist = null;
        ShoesDataModel findwhichCategeory = null;
        try {

            findwhichCategeory = shoeservice.getshoesDataByIdService(id);
            int categeory = findwhichCategeory.getCategeory();
            isdeleted = shoeservice.deleteProductwithIdService(id);
            if (isdeleted == true) {
                shoesDatalist = shoeservice.getShoeDatabyCategeory(categeory);
```



```
            model.put("men_women_kids_ShoeDataname", shoesDatalist);

         }

         return "adminForm";
      } catch (Exception e) {
         System.out.println(
               "Exception at
com.cisco.controllers.CommonController.deleteProductMethod() " +
e.getMessage());
         return "adminForm";
      }
   }

   @GetMapping("/updateProduct")
   public String updateProductMethod(ModelMap model, @RequestParam int id)
{

      boolean isupdated = false;
      ShoesDataModel shoesData = null;
      try {

         shoesData = shoeservice.getshoesDataByIdService(id);

         if (shoesData.getId() > 0) {
            model.put("men_women_kids_ShoeDataname", shoesData);
            return "updateProductForm";
         }

         return "adminForm";
      } catch (Exception e) {
         System.out.println(
               "Exception at
com.cisco.controllers.CommonController.deleteProductMethod() " +
e.getMessage());
         return "adminForm";
      }
   }

   @GetMapping("/updatetoNewproduct")
   public String updateProductMethod(ModelMap model, @RequestParam String
weartype, @RequestParam String prod_name,
         @RequestParam double prod_price, @RequestParam String prod_img,
@RequestParam int prod_id) {

      boolean isupdated = false;
      ShoesDataModel shoesData = null;
      List<ShoesDataModel> shoesDataList = null;
      try {
         int categeory = 0;
         if (weartype.equals("Mens_Wear")) {
            categeory = 1;
         } else if (weartype.equals("Womens_Wear")) {
            categeory = 2;
         } else if (weartype.equals("Kids_Wear")) {
            categeory = 3;
         }
         isupdated = shoeservice.updateShoeProductService(categeory,
prod_price, prod_name, prod_img, prod_id);
         if (isupdated == true) {
            shoesDataList = shoeservice.getShoeDatabyCategeory(categeory);
            model.put("men_women_kids_ShoeDataname", shoesDataList);

         }

         return "adminForm";

      } catch (Exception e) {
         System.out.println(
               "Exception at
com.cisco.controllers.CommonController.deleteProductMethod()  " +
e.getMessage());
         return "adminForm";
      }
   }
}
```

- **Dao**

OrdersDao.java

```
package com.cisco.dao;

import java.sql.Date;
import java.util.List;

import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.cisco.model.OrderedShoeModel;

@Repository
public interface OrdersDao  extends CrudRepository<OrderedShoeModel,
Integer>{






    @Query("from OrderedShoeModel where categeory=:categeoryId and
date=:date")
    public List<OrderedShoeModel> getRequiredCompleteTransactionsData(int
categeoryId,Date date);

}
```

ShoesDao.java

```
package com.cisco.dao;

import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import com.cisco.model.ShoesDataModel;
import antlr.collections.List;

//@Repository
//public interface ShoesDao extends CrudRepository<ShoesDataModel, Integer>
{



   @Repository
   public interface ShoesDao extends CrudRepository<ShoesDataModel,
Integer> {

        @Query("from ShoesDataModel where categeory=:categeory")
          public Iterable<ShoesDataModel> findByCategory(int categeory);


        @Query("from ShoesDataModel where id=:seletedShoeId")
          public  ShoesDataModel  getshoesDataById(int seletedShoeId);


        @Modifying
          @Transactional
          @Query("delete from ShoesDataModel where id=:id")
          public int deleteProductwithId(@Param("id") int id);



        @Modifying
          @Transactional
          @Query("update ShoesDataModel   set categeory=:categeory,
price=:price, name=:name, imagelink=:imagelink  where id=:id")
          public int updateShoeProduct(@Param("categeory") int categeory,
@Param("price") double price,
                @Param("name") String name, @Param("imagelink") String
imagelink,
                @Param("id") int id);


}
```

UsersDao.java

```
package com.cisco.dao;

import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;
import org.springframework.transaction.annotation.Transactional;

import com.cisco.model.UserInfoModel;

@Repository
public interface UsersDao extends CrudRepository<UserInfoModel, Integer> {



   @Query("from UserInfoModel where firstname=:username and
password=:password")
    public UserInfoModel  isValidAdminUser(String username,String password
);



    @Query("from UserInfoModel where rollid=:rollid")
       public Iterable<UserInfoModel> getUserDatabyRollid(int rollid);



    @Query("from UserInfoModel where password=:password")
       public  UserInfoModel  isCorrectPassword_or_Not(String password);

    @Modifying
    @Transactional
    @Query("update UserInfoModel   set  password = :password where rollid=1
and firstname=:username")
    public int updateAdminPassword(@Param("password") String password,
@Param("username") String username);




}
```

❖ Exception

GlobalExceptionHandeler.java

```
package com.cisco.exception;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;

@ControllerAdvice
public class GlobalExceptionHandeler {
   @ExceptionHandler(value=Exception.class)
   public String handleAnyException()

   {
      return "Error";
   }
}
```

❖ Model

OrderedShoeModel.java

```
package com.cisco.model;

import java.sql.Date;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "tbl_orderinfo")
```

```
public class OrderedShoeModel {
   @Id
   @GeneratedValue
   private int orderid;
   private int shoeid ;
   private int categeory;
   private double price;
   private int quantity ;
   private Date date;
   private int size;
   private String name;
   private String imagelink;
   private double totalprice ;

   public int getQuantity() {
      return quantity;
   }
   public void setQuantity(int quantity) {
      this.quantity = quantity;
   }
   public int getSize() {
       return size;
   }

   public void setSize(int size) {
       this.size = size;
   }
   public int getOrderid() {
      return orderid;
   }
   public void setOrderid(int orderid) {
      this.orderid = orderid;
   }
   public int getShoeid() {
      return shoeid;
   }
   public void setShoeid(int shoeid) {
      this.shoeid = shoeid;
   }
   public int getCategeory() {
      return categeory;
   }
   public void setCategeory(int categeory) {
      this.categeory = categeory;
   }
   public double getPrice() {
      return price;
   }
   public void setPrice(double price) {
      this.price = price;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public String getImagelink() {
      return imagelink;
   }
```

```
    public void setImagelink(String imagelink) {
        this.imagelink = imagelink;
    }
    public double getTotalprice() {
        return totalprice;
    }
    public void setTotalprice(double totalprice) {
        this.totalprice = totalprice;
    }

    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }
    @Override
    public String toString() {
        return "OrderedShoeModel [orderid=" + orderid + ", shoeid=" + shoeid
+ ", categeory=" + categeory + ", price="
                + price + ", quantity=" + quantity + ", date=" + date + ",
name=" + name + ", imagelink=" + imagelink
                + ", totalprice=" + totalprice + ", size=" + size + "]";
    }
    public OrderedShoeModel() {
        super();
        // TODO Auto-generated constructor stub
    }
    public OrderedShoeModel(int shoeid, int categeory, double price, int
quantity, Date date, String name,
            String imagelink, double totalprice, int size) {
        super();
        this.shoeid = shoeid;
        this.categeory = categeory;
        this.price = price;
        this.quantity = quantity;
        this.date = date;
        this.name = name;
        this.imagelink = imagelink;
        this.totalprice = totalprice;
        this.size = size;
    }


}
```

ShoesDataModel.java

```
package com.cisco.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "tbl_shoesinfo")
public class ShoesDataModel {
    @Id
    @GeneratedValue
   private int id;
   private int categeory;
   private double price;
   private String name;
   private String imagelink;
   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }
   public int getCategeory() {
      return categeory;
   }
   public void setCategeory(int categeory) {
      this.categeory = categeory;
   }
   public double getPrice() {
      return price;
   }
   public void setPrice(double price) {
      this.price = price;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
   public String getImagelink() {
      return imagelink;
   }

   public void setImagelink(String imagelink) {
      this.imagelink = imagelink;
   }

   @Override
   public String toString() {
      return "ShoesDataModel [categeory=" + categeory + ", price=" + price
+ ", name=" + name + ", imagelink="
            + imagelink + "]";
   }
   public ShoesDataModel(int categeory, double price, String name, String
imagelink) {
      super();
      this.categeory = categeory;
      this.price = price;
      this.name = name;
      this.imagelink = imagelink;
   }
   public ShoesDataModel() {
      super();
      // TODO Auto-generated constructor stub
   }



}
```

UserInfoModel.java

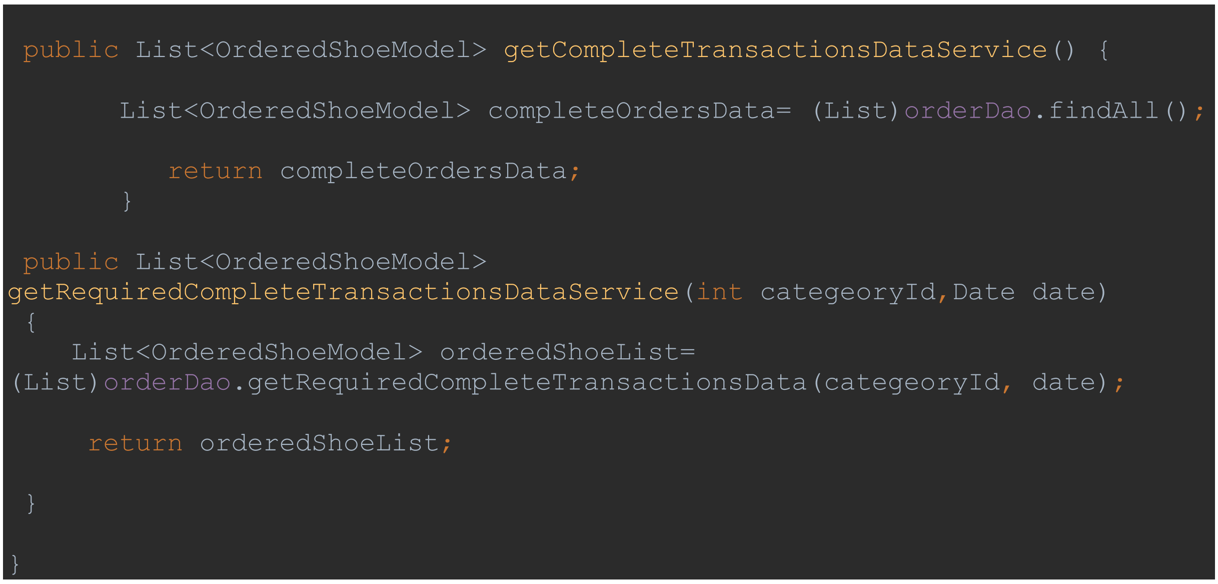
```java
package com.cisco.model;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="tbl_userinfo")
public class UserInfoModel {
   @Id
   @GeneratedValue
   private int id;
   private String firstname;
   private String lastname;
   private String password;
   private int rollid;
   private long phno;
   private String email;

   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }
   public String getFirstname() {
      return firstname;
   }
   public void setFirstname(String firstname) {
      this.firstname = firstname;
   }
   public String getLastname() {
      return lastname;
   }
   public void setLastname(String lastname) {
      this.lastname = lastname;
   }
   public String getPassword() {
      return password;
   }
   public void setPassword(String password) {
      this.password = password;
   }
   public int getRollid() {
      return rollid;
   }
   public void setRollid(int rollid) {
      this.rollid = rollid;
   }
   public long getPhno() {
      return phno;
   }
   public void setPhno(long phno) {
      this.phno = phno;
   }
   public String getEmail() {
      return email;
   }
```

```
    public void setEmail(String email) {
        this.email = email;
    }
    @Override
    public String toString() {
        return "UserInoModel [firstname=" + firstname + ", lastname=" +
lastname + ", password=" + password
                + ", rollid=" + rollid + ", phno=" + phno + ", email=" + email
+ "]";
    }
    public UserInfoModel() {
        super();
        // TODO Auto-generated constructor stub
    }
    public UserInfoModel(String firstname, String lastname, String password,
int rollid, long phno, String email) {
        super();
        this.firstname = firstname;
        this.lastname = lastname;
        this.password = password;
        this.rollid = rollid;
        this.phno = phno;
        this.email = email;
    }


}
```

❖ Service

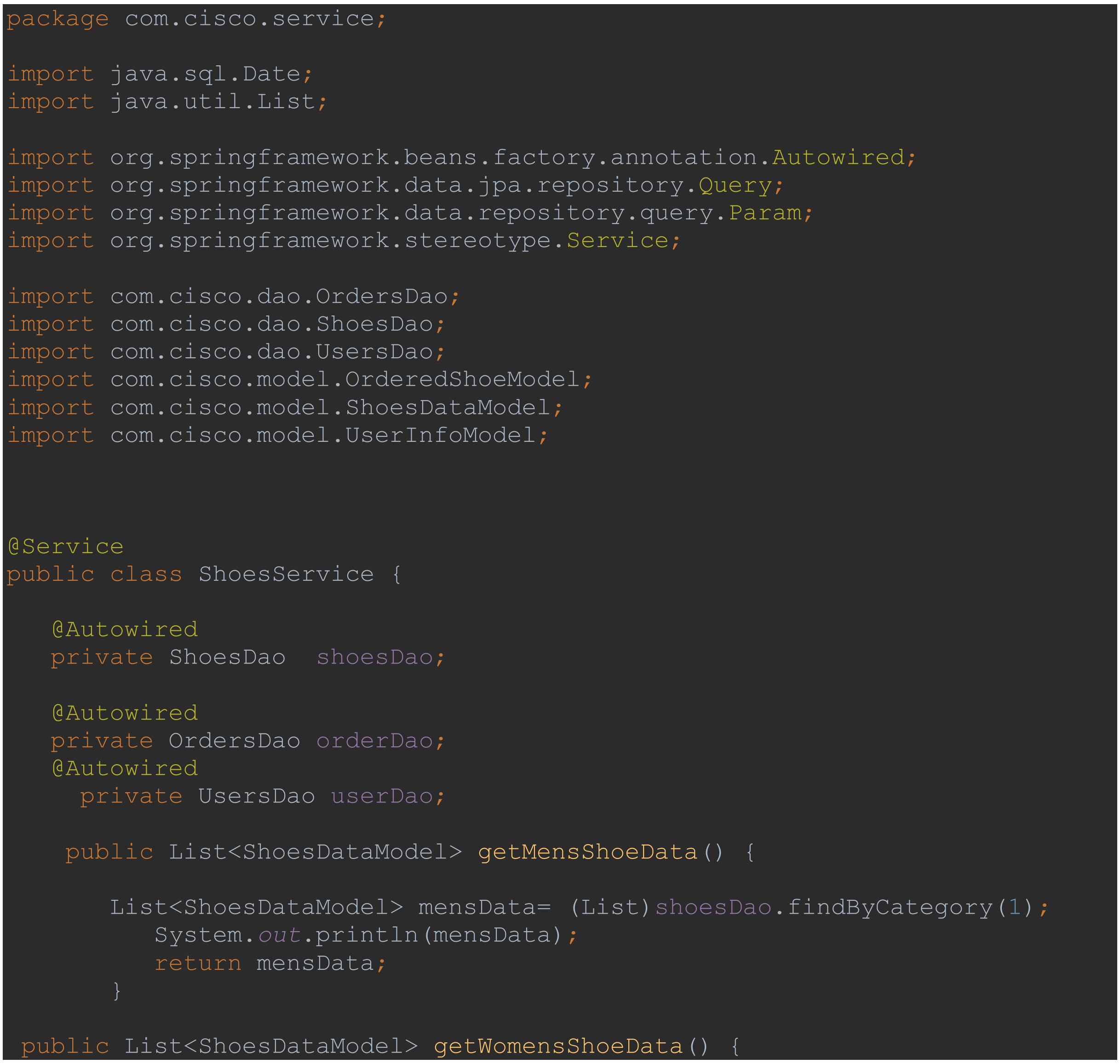
OrdersService.java

```
package com.cisco.service;

import java.sql.Date;
import java.util.List;

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

import com.cisco.dao.OrdersDao;
import com.cisco.model.OrderedShoeModel;
import com.cisco.model.ShoesDataModel;

@Service
public class OrdersService {
    @Autowired
    private OrdersDao orderDao;

     public void insertBookingInfoService(int shoeid, String name, int
categeory,
          double price, String imglink, int quantity, double
totalprice,Date date, int size )
     {
       OrderedShoeModel  orderedShoeModel =new  OrderedShoeModel(shoeid,
categeory, price, quantity, date, name, imglink, totalprice, size);
       orderDao.save(orderedShoeModel);
     }

```

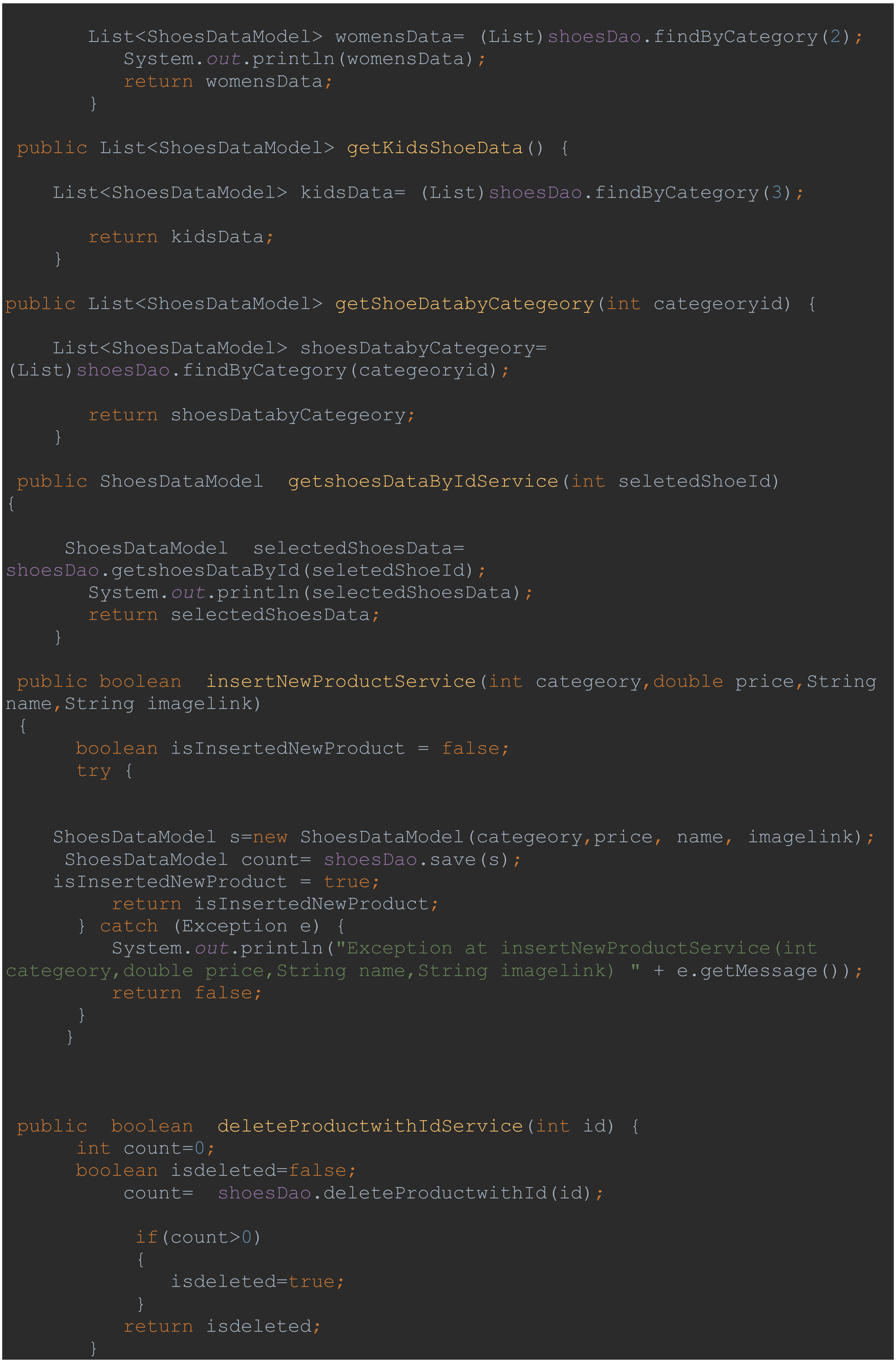


```
 public List<OrderedShoeModel> getCompleteTransactionsDataService() {

       List<OrderedShoeModel> completeOrdersData= (List)orderDao.findAll();

          return completeOrdersData;
       }

 public List<OrderedShoeModel>
getRequiredCompleteTransactionsDataService(int categeoryId,Date date)
 {
    List<OrderedShoeModel> orderedShoeList=
(List)orderDao.getRequiredCompleteTransactionsData(categeoryId, date);

     return orderedShoeList;

 }

}
```

ShoesService.java



```
package com.cisco.service;

import java.sql.Date;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Service;

import com.cisco.dao.OrdersDao;
import com.cisco.dao.ShoesDao;
import com.cisco.dao.UsersDao;
import com.cisco.model.OrderedShoeModel;
import com.cisco.model.ShoesDataModel;
import com.cisco.model.UserInfoModel;



@Service
public class ShoesService {

   @Autowired
   private ShoesDao  shoesDao;

   @Autowired
   private OrdersDao orderDao;
   @Autowired
     private UsersDao userDao;

    public List<ShoesDataModel> getMensShoeData() {

       List<ShoesDataModel> mensData= (List)shoesDao.findByCategory(1);
          System.out.println(mensData);
          return mensData;
       }

 public List<ShoesDataModel> getWomensShoeData() {
```



```
       List<ShoesDataModel> womensData= (List)shoesDao.findByCategory(2);
          System.out.println(womensData);
          return womensData;
       }

 public List<ShoesDataModel> getKidsShoeData() {

    List<ShoesDataModel> kidsData= (List)shoesDao.findByCategory(3);

       return kidsData;
    }

public List<ShoesDataModel> getShoeDatabyCategeory(int categeoryid) {

    List<ShoesDataModel> shoesDatabyCategeory=
(List)shoesDao.findByCategory(categeoryid);

       return shoesDatabyCategeory;
    }

 public ShoesDataModel  getshoesDataByIdService(int seletedShoeId)
{

     ShoesDataModel  selectedShoesData=
shoesDao.getshoesDataById(seletedShoeId);
       System.out.println(selectedShoesData);
       return selectedShoesData;
    }

 public boolean  insertNewProductService(int categeory,double price,String
name,String imagelink)
 {
      boolean isInsertedNewProduct = false;
      try {


    ShoesDataModel s=new ShoesDataModel(categeory,price, name, imagelink);
     ShoesDataModel count= shoesDao.save(s);
    isInsertedNewProduct = true;
         return isInsertedNewProduct;
      } catch (Exception e) {
         System.out.println("Exception at insertNewProductService(int
categeory,double price,String name,String imagelink) " + e.getMessage());
         return false;
      }
     }



 public  boolean  deleteProductwithIdService(int id) {
      int count=0;
      boolean isdeleted=false;
          count=  shoesDao.deleteProductwithId(id);

           if(count>0)
           {
              isdeleted=true;
           }
          return isdeleted;
       }
 public  boolean  updateShoeProductService( int categeory,   double price,
         String name,  String imagelink,
         int id) {
      int count=0;
      boolean isUpdated=false;
          count=  shoesDao.updateShoeProduct(categeory, price, name,
imagelink, id);

           if(count>0)
           {
              isUpdated=true;
           }
          return isUpdated;
       }
}
```

UsersService.java

```
package com.cisco.service;

import java.util.List;

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

import com.cisco.dao.OrdersDao;
import com.cisco.dao.UsersDao;
import com.cisco.model.OrderedShoeModel;
import com.cisco.model.ShoesDataModel;
import com.cisco.model.UserInfoModel;

@Service
public class UsersService {

   @Autowired
   private UsersDao userDao;

   public UserInfoModel isValidAdminUserService(String username, String
password) {

      UserInfoModel userInfoModel = userDao.isValidAdminUser(username,
password);

      return userInfoModel;

   }

   public List<UserInfoModel> getUserDataService() {

       List<UserInfoModel> usersList= (List)userDao.getUserDatabyRollid(2);

          return usersList;
       }



   public  UserInfoModel  isCorrectPassword_or_NotService(String password)
{
        UserInfoModel  usersList=
userDao.isCorrectPassword_or_Not(password);

          return usersList;
       }


   public  boolean  updateAdminPasswordService(UserInfoModel userInfoModel,
String currentpassword,
         String newpassword, String confirmpassword) {
      int count=0;
      boolean isUpdated=false;
          count=
userDao.updateAdminPassword(confirmpassword,userInfoModel.getFirstname());

           if(count>0)
           {
              isUpdated=true;
           }
          return isUpdated;
       }

   public void insertUserDataService(String firstname, String lastname,
String password, String confirmpassword, int rollid, Long mobileno, String
email)
   {
      UserInfoModel   u =new UserInfoModel(firstname, lastname,
confirmpassword, rollid, mobileno, email);
      userDao.save(u);
   }
}
```

- **SportyShoes**

SportyShoesApplication.java

```
package com.cisco.SportyShoes;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.autoconfigure.domain.EntityScan;
import org.springframework.context.annotation.ComponentScan;
import
org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import
org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@SpringBootApplication
@ComponentScan("com.cisco")
@EntityScan("com.cisco.model")
@EnableJpaRepositories("com.cisco.dao")
public class SportyShoesApplication {

   public static void main(String[] args) {
      SpringApplication.run(SportyShoesApplication.class, args);
   }

}
```

Style.css file

```
@charset "UTF-8";

body {
   font-family: Tahoma, sans-serif;
   background:url("https://img.freepik.com/free-photo/green-product-
background_53876-
90135.jpg?w=2000&t=st=1674899877~exp=1674900477~hmac=e052c0a60b6005a5480c4c
e5caf59447d2426190bc4458f4884ca1c8a371421e");
   background-size:cover;
   background-attachment: fixed;
}
img{
    animation: move 2s;
}
@keyframes move{
    from{transform:translateX(-100%)}
    to{transform:translateX(0)}
}

@keyframes welcome-animation {
  0% {
    transform: translateX(-100%);
    opacity: 0;
  }
  100% {
    transform: translateX(0);
    opacity: 1;
  }
}
.textboxstyle {
width: 200px;
padding: 5px;
margin: 5px;
border: 3px solid black;
}

h1 {
  animation: welcome-animation 2s ease-in-out;
  font-size: 3em;
  text-align: center;
}

.welcome-animation {
    animation: welcome-animation 2s ease-in-out;
}
.animated-message {
  text-align: center;
  font-family: Arial, sans-serif;
  position: relative;
}
```

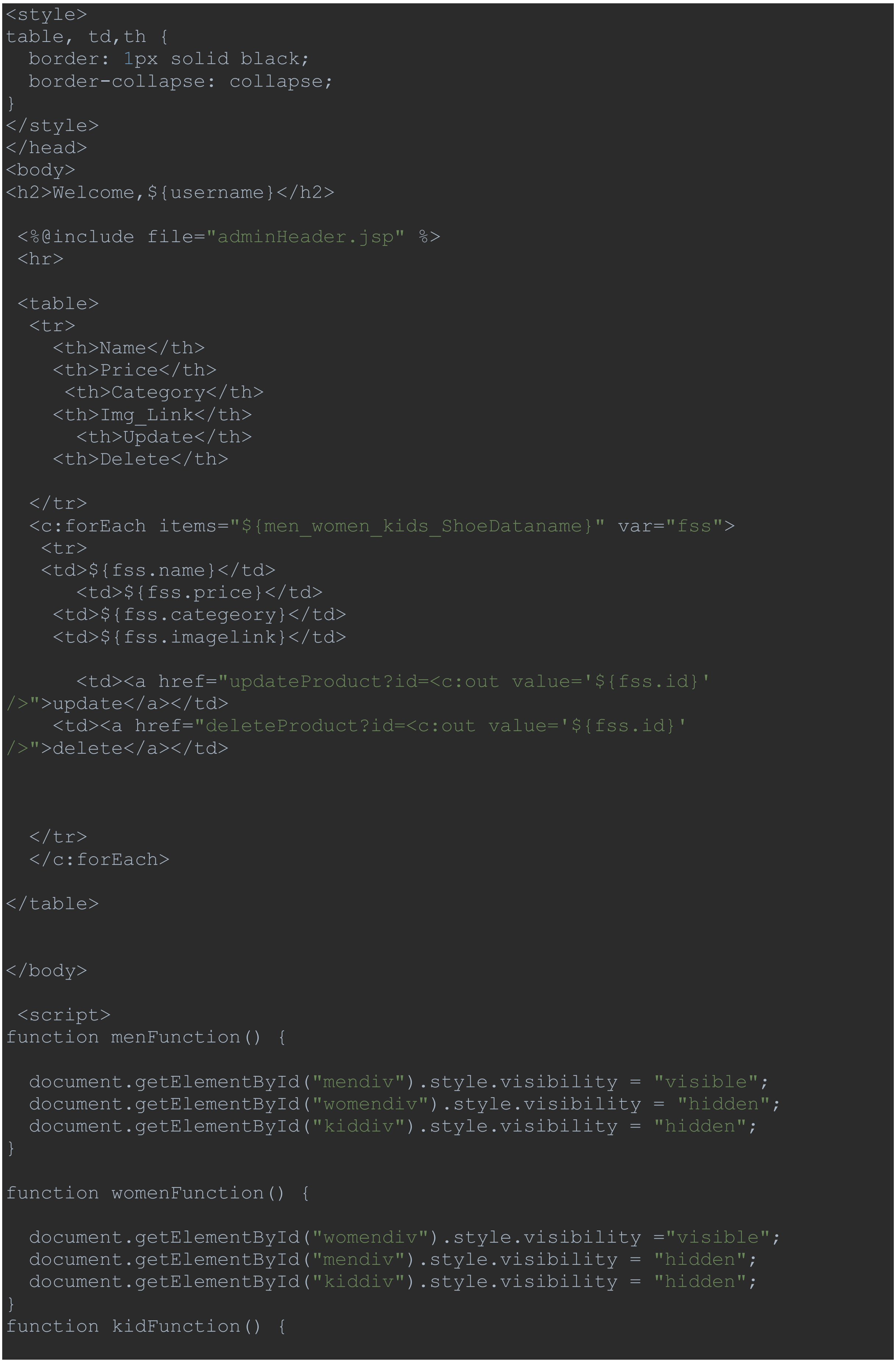
```
.animated-message h1 {
  font-size: 36px;
  color: green;
  text-transform: uppercase;
  animation: message-animation 2s ease-in-out;
}

.animated-message h1:nth-of-type(2) {
  color: #333;
  font-size: 25px;
}

@keyframes message-animation {
  from {
    transform: translateY(-50px);
    opacity: 0;
  }
  to {
    transform: translateY(0);
    opacity: 1;
  }
}
h1 {
  font-family: 'Montserrat', sans-serif; /* Use a unique font style */
  font-weight: bold; /* Make the text bold */
  color: #000000; /* Use a different color */
  text-shadow: 2px 2px #aaa; /* Add a text shadow */
  text-transform: uppercase; /* Capitalize all letters */
  letter-spacing: 2px; /* Increase the spacing between letters */
}
h1:hover {
  transform: rotate(360deg); /* Rotate the text on hover */
  transition: transform 0.5s ease; /* Add a transition effect */
}

img {
  background-color: transparent;
}
```

addNewProductForm.jsp file

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Add New Product</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
   <%@include file="adminHeader.jsp"%>
   <div class="container" align="center">
      <form name="addProductform" class="addflightformstyle"
         action="addnewproduct" onsubmit="return addFlightvalidateForm()">


<b>Add New Product  </b><br><br>
         Product Category :
            <input type="radio" id="men" name="weartype" value="Mens_Wear">
            <label for="men">Mens Wear</label><br> <input type="radio"
               id="women" name="weartype" value="Womens_Wear"> <label
               for="women">Womens Wear</label> <br> <input type="radio"
               id="kid" name="weartype" value="Kids_Wear"> <label
               for="kid">Kids Wear</label>

         <br> Product Name : <input class="textboxstyle" type="text"
name="prod_name"><br>
         <br> Product Price : <input class="textboxstyle" type="text"
name="prod_price"><br>
         <br> Img Link : <input class="textboxstyle" type="text"
name="prod_img"><br>
         <br> <input type="submit" value="Add Product">

      </form>
        </div>

</body>

<script type="text/javascript">
   function addFlightvalidateForm() {
      var weartype = document.forms["addProductform"]["weartype"].value;
      var prod_name = document.forms["addProductform"]["prod_name"].value;
      var prod_price =
document.forms["addProductform"]["prod_price"].value;
      var prod_img = document.forms["addProductform"]["prod_img"].value;

      if (weartype == "") {
         alert("weartype must be filled out");
         return false;
      }
      if (prod_name == "") {
         alert("prod_name must be filled out");
         return false;
      }
      if (prod_price == "") {
         alert("prod_price must be filled out");
         return false;
      }
      if (prod_img == "") {
         alert("prod_img must be filled out");
         return false;
      }

   }
</script>

</html>
```

admiForm.jsp file

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp"%>
<%@ taglib  uri ="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Admin Form</title>
<style><%@include file="/WEB-INF/style.css"%></style>
```



```
<style>
table, td,th {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
<h2>Welcome,${username}</h2>

 <%@include file="adminHeader.jsp" %>
 <hr>

 <table>
  <tr>
    <th>Name</th>
    <th>Price</th>
     <th>Category</th>
    <th>Img_Link</th>
      <th>Update</th>
    <th>Delete</th>

  </tr>
  <c:forEach items="${men_women_kids_ShoeDataname}" var="fss">
   <tr>
   <td>${fss.name}</td>
      <td>${fss.price}</td>
    <td>${fss.categeory}</td>
    <td>${fss.imagelink}</td>

      <td><a href="updateProduct?id=<c:out value='${fss.id}'
/>">update</a></td>
    <td><a href="deleteProduct?id=<c:out value='${fss.id}'
/>">delete</a></td>



  </tr>
  </c:forEach>

</table>


</body>

 <script>
function menFunction() {

  document.getElementById("mendiv").style.visibility = "visible";
  document.getElementById("womendiv").style.visibility = "hidden";
  document.getElementById("kiddiv").style.visibility = "hidden";
}

function womenFunction() {

  document.getElementById("womendiv").style.visibility ="visible";
  document.getElementById("mendiv").style.visibility = "hidden";
  document.getElementById("kiddiv").style.visibility = "hidden";
}
function kidFunction() {
     document.getElementById("kiddiv").style.visibility = "visible";
     document.getElementById("mendiv").style.visibility = "hidden";
     document.getElementById("womendiv").style.visibility = "hidden";
   }
</script>
</html>
```

adminHeader.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>admin page</title>
<style><%@include file="/WEB-INF/style.css"%></style>

</head>
<body>
<h3><a href="home" style="text-
decoration:none;color:black">Home</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
bsp;&nbsp;
<a href="getcompletemenshoesforAdmin" style="text-
decoration:none;color:black">Men
Shoes</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="getcompletewomenshoesforAdmin" style="text-
decoration:none;color:black">Women
Shoes</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="getcompletekidshoesforAdmin" style="text-
decoration:none;color:black">Kids Shoes</a>&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;
<a href="changeadminpassword" style="text-
decoration:none;color:black">Change
Password</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="customersList" style="text-decoration:none;color:black">Customers
List</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="addNewProductForm" style="text-decoration:none;color:black" >Add
Product</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<a href="reports" style="text-
decoration:none;color:black">Reports</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;
<a href="signin" style="text-decoration:none;color:black">Logout</a>
</h3>
<hr>
</body>
</html>
```

bookShoeForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Book Shoe</title>
```

```
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>

<%@include file="homeHeader.jsp" %>

<form action="bookandpaymentreturntoHome">
<input type="hidden" name="imglink" value="${orderedshoemodel.imagelink}"
/>
<img  src="${orderedshoemodel.imagelink}" style="float: left; width: 20%;
margin-right: 1%; margin-bottom: 0.5em;" >


<input type="hidden" name="name"  value="${orderedshoemodel.name}"  />
<label><h1>${orderedshoemodel.name}</h1></label>

<input type="hidden" name="id" value="${orderedshoemodel.shoeid}"  />
<b>Product Id :<label >${orderedshoemodel.shoeid}</label></b> <br>

<input type="hidden" name="categeory"
value="${orderedshoemodel.categeory}"  />
<b>Category :<label  >${orderedshoemodel.categeory}</label></b> <br>


<input type="hidden" name="price"  value="${orderedshoemodel.price}"  />
<h3>Price :<label  >${orderedshoemodel.price}</label> </h3><br>

<input type="hidden" name="quantity" value="${orderedshoemodel.quantity}"
/>
 <b>Quantity :<label  >${orderedshoemodel.quantity}</label> </b> <br>

 <input type="hidden" name="size" value="${orderedshoemodel.size}"  />
<b>Selected Size :<label  >${orderedshoemodel.size}</label> </b> <br>

 <input type="hidden" name="totalprice"
value="${orderedshoemodel.totalprice}"  />
 <h2>Total Price :<label  >${orderedshoemodel.totalprice}</label> </h2>
<br><br>

 <input type="submit" value="Book Order">

</form>
</body>
</html>
```

changeAdminPasswordForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Change Admin Password</title>
<style><%@include file="/WEB-INF/style.css"%>
body{
    text-align:center;
    }
form{
    margin:auto;
    width:50%;
    }
input{
    font-size:large;
    }
</style>
</head>
<body>
<%@include file="adminHeader.jsp" %>

      <form name="changeform" action="changeadminPassword" onsubmit="return
changePasswordvalidateForm()">
      <b>Change Admin password   </b><br><br>
       Current Password : <input  type="text" name="currentpassword"
><br><br>
      New Password : <input  type="text" name="newpassword" ><br><br>
      Confirm Password : <input  type="text" name="confirmpassword"
><br><br>
      <input type="submit" value="ChangePassword">
      </form>
</body>

<script>
function changePasswordvalidateForm() {
  var currentpassword =
document.forms["changeform"]["currentpassword"].value;
  var newpassword = document.forms["changeform"]["newpassword"].value;
  var confirmpassword =
document.forms["changeform"]["confirmpassword"].value;

  if (currentpassword == "") {
    alert("currentpassword must be filled out");
    return false;
  }
  if (newpassword == "") {
       alert("newpassword must be filled out");
       return false;
     }
  if (confirmpassword == "") {
       alert("confirmpassword must be filled out");
       return false;
     }
}
</script>
</html>
```

customersListForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp" %>
 <%@ taglib  uri ="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customers Information</title>
<style><%@include file="/WEB-INF/style.css"%></style>
<style>
table {
  border: 1px solid black;
  border-collapse: collapse;
  margin: 0 auto;
}
td, th {
  border: 1px solid black;
}
</style>
</head>
<body>
 <%@include file="adminHeader.jsp" %>
<h1>Customers Information</h1>

<div style="text-align: center;">
  <table>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Password</th>
      <th>Rollid</th>
      <th>PhNo</th>
      <th>Email</th>
    </tr>
    <c:forEach items="${usersList}" var="fss">
    <tr>
      <td>${fss.firstname}</td>
      <td>${fss.lastname}</td>
      <td>${fss.password}</td>
      <td>${fss.rollid}</td>
      <td>${fss.phno}</td>
      <td>${fss.email}</td>
    </tr>
    </c:forEach>
  </table>
</div>
</body>
</html>
```

Error.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isErrorPage="true"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sporty Shoes</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
      <h1>Error</h1>
      <h2><%=exception.getMessage() %><br/> </h2>

</body>
</html>
```

homeHeader.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<title>Home</title>
<style><%@include file="/WEB-INF/style.css"%></style>

</head>
<body>

 <h1 style="text-align:left;float:left;"> <a href="home" style="text-
decoration:none;color:black">Sporty Shoes </a> </h1>
 <h3 style="text-align:right;float:right;"><a href="signin" style="text-
decoration:none;color:black">SignIn</a>  |  <a href="signup" style="text-
decoration:none;color:black">SignUp</a></h3>
 <hr style="clear:both;"/>
 <b><p> <a href="home" style="text-decoration:none;color:black">Home</a> |
<a href="menshoes" style="text-decoration:none ;color:black">Men</a> | <a
href="womenshoes" style="text-decoration:none ;color:black">Women</a> | <a
href="kidshoes" style="text-decoration:none ;color:black">Kids</a></p></b>
</body>
</html>
```

Index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<html>
<head>
<title>Home</title>
<style><%@include file="/WEB-INF/style.css"%>
.footer {
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  background-color: black;
  color: white;
  text-align: center;
}
</style>
</head>
<body>
   <%@include file="homeHeader.jsp"%>
   <hr>
      <a href="menshoes" >
            <img src="https://img.freepik.com/premium-photo/new-yellow-
foxy-leather-suede-casual-sport-male-sneakers-white-background_555047-
1459.jpg?w=2000" style="float: left; width: 30%; margin-right: 1%; margin-
bottom: 0.5em;">

      </a>

      <a href="womenshoes">
            <img src="https://img.freepik.com/free-photo/basic-white-
sneakers-unisex-streetwear-fashion-shoot_53876-
102115.jpg?w=2000&t=st=1674983322~exp=1674983922~hmac=a16592d4eaec6f1bc0787
1367893c26b1d91dc687b23cfb39f09eb3b79009bfa" style="float: left; width:
30%; margin-right: 1%; margin-bottom: 0.5em;">
      </a>

      <a href="kidshoes" >
      <img src="https://img.freepik.com/free-photo/baby-shoes_1203-
6837.jpg?w=2000&t=st=1674982167~exp=1674982767~hmac=cda475f7832088e802b6ed0
b3c66e086e4214b145f7b4f0d0787770a559f8e5f" style="float: left; width: 30%;
```

```
margin-right: 1%; margin-bottom: 0.5em;">
      </a>
      <div class="footer">
      <p>Copyright □ SportyShoes. All rights reserved.</p>
      </div>

</html>
```

kidsShoesForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp"%>
<%@ taglib  uri ="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Kids Shoes</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<%@include file="homeHeader.jsp" %>
<h2>Kid's Shoes</h2>
<table>

<c:forEach items="${kidshoeData}" var="fss">
<tr>
<td><span>
              <a href="selectedShoe?id=<c:out value='${fss.id}' />">
                <img src="${fss.imagelink}" style="float: left; width: 30%;
margin-right: 1%; margin-bottom: 0.5em;">
              <br><br><label>INR ${fss.price} </label> <br>
<label>${fss.name}</label>

            </a>
            </span>
            </td>

</tr>
 </c:forEach>

</table>
</body>
</html>
```

menShoesForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"  isELIgnored="false" errorPage="Error.jsp"%>
<%@ taglib  uri ="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Mens Shoes</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<%@include file="homeHeader.jsp" %>
<h2>Men's Shoes</h2>
<table>
```

```
<c:forEach items="${menshoeData}" var="fss">
<tr>
<td><span>
              <a href="selectedShoe?id=<c:out value='${fss.id}' />">
                <img src="${fss.imagelink}" style="float: left; width: 30%;
margin-right: 1%; margin-bottom: 0.5em;">
              <br><br><label >INR ${fss.price} </label> <br>
<label>${fss.name}</label>

            </a>
            </span>
            </td>

</tr>
 </c:forEach>

</table>
</body>
</html>
```

womenShoesForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"  isELIgnored="false" errorPage="Error.jsp"%>
<%@ taglib  uri ="http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Women Shoes</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<%@include file="homeHeader.jsp" %>
<h2>Women's Shoes</h2>
<table>
<c:forEach items="${womenshoeData}" var="fss">
<tr>
<td><span>
              <a href="selectedShoe?id=<c:out value='${fss.id}' />">
                <img src="${fss.imagelink}" style="float: left; width: 30%;
margin-right: 1%; margin-bottom: 0.5em;">
              <br><br><br><br><br><label >INR ${fss.price} </label> <br>
<label>${fss.name}</label>

            </a>
            </span>
            </td>

</tr>
 </c:forEach>
</table>
</body>
</html>
```

orderplaceForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<title>OrderPlace</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<div class="animated-message">
  <h1>Payment Done. Order Placed</h1>
  <h1>Thanks for Shopping.....</h1>
</div>
<br>
<h2 align="center"> <a href="home" style="text-decoration:none">Back to
Home</a> </h2>
</body>
</html>
```

reportsForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Reports</title>
<style><%@include file="/WEB-INF/style.css"%></style>
<style>
table, td,th {
  border: 1px solid black;
  border-collapse: collapse;
}
table {
  margin: 0 auto; /* center the table */
}
</style>

</head>
<body>
   <%@include file="adminHeader.jsp"%>
   <h1>Sales Report</h1>
<form action="requiredreports" name="searchform" onsubmit="return
reportvalidateForm()" style="text-align:center;">
      <b><label for="categeory">Choose a Category:</label></b> <select
         name="categeory" id="categeory">
         <option value="mens">Mens Wear</option>
         <option value="womens">Womens Wear</option>
         <option value="kids">Kids Wear</option>
      </select>
      <b> <label for="categeory">Select Date:</label> </b>
      <input type="date" name="date">
      <input type="submit" value="Done" style="font-size: 10px; padding:
10px 10px;">
      <hr>
   </form>
   <table>
      <tr>
         <th>Orderid</th>
         <th>Shoeid</th>
         <th>Name</th>
         <th>Price</th>
         <th>Quantity</th>
         <th>Size</th>
         <th>Totalprice</th>
         <th>Categeory</th>
         <th>Date</th>
      </tr>
      <c:forEach items="${orderedShoeList}" var="fss">
         <tr>
            <td>${fss.orderid}</td>
            <td>${fss.shoeid}</td>
            <td><a>${fss.name}</a></td>
            <td>${fss.price}</td>
            <td><a>${fss.quantity}</a></td>
            <td><a>${fss.size}</a></td>
            <td><a>${fss.totalprice}</a></td>
            <td>${fss.categeory}</td>
            <td><a>${fss.date}</a></td>

         </tr>

      </c:forEach>
   </table>
   <h1 >Total Sales=  ${totalSales}</h1>
</body>
<script type="text/javascript">
   function reportvalidateForm() {
      var date = document.forms["searchform"]["date"].value;

      if (date == "") {
         alert("Date must be select");
         return false;
      }

   }
</script>

</html>
```

selectedShoeInfoForm.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Selected Shoe Info</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<%@include file="homeHeader.jsp" %>

<form action="orderinfo">

<input type="hidden" name="imglink" value="${selectedShoesData.imagelink}"
/>
<img src="${selectedShoesData.imagelink}" style="float: left; width: 20%;
margin-right: 1%; margin-bottom: 0.5em;"  >
<input type="hidden" name="name" value="${selectedShoesData.name}" />
<label><h1>${selectedShoesData.name}</h1></label>

<input type="hidden" name="id" value="${selectedShoesData.id}" />
<b>Product Id : <label>  ${selectedShoesData.id}</label></b> <br>
```

```
<input type="hidden" name="categeory"
value="${selectedShoesData.categeory}" />
<b>Categeroy : <label>${selectedShoesData.categeory}</label> <br>

<input type="hidden" name="price" value="${selectedShoesData.price}" />
<h2>Price : <label>${selectedShoesData.price}</label> </h2> </b><br>

<b> <label for="quantity">Quantity:</label></b>
  <input type="number" id="quantity" name="quantity" min="1" max="5"
value=1  ><br>
  <b> <label for="size">Size:</label></b>
<select id="size" name="size">
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
</select><br><br>

  <input type="submit" value="Place Order">
</form>
</body>
</html>
```

signinFrom.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp" isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Sign In</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
<%@include file="homeHeader.jsp" %>

            <form action="adminsignin"  >
            <hr>
          <div class="container" align="center">
           <h1>Enter your Credentials</h1>
            <b><font color="red">${errorMessage}</font></b>

         <h2> UserName <input type="text" name="username"  ></h2>
          <h2>Password  <input  type="password" name="password"  ></h2>
          <input type="submit" value="Signin">
          </div>
           </form>

</body>
</html>
```

signUpFrom.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Sign Up</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>

<%@include file="homeHeader.jsp" %>

<hr>
 <form name="adduserform" action="signupnewuser"  onsubmit="return
addFlightvalidateForm()">
       <h3>Create your Account</h3><br>
      First Name : <input type="text" name="firstname" ><br><br>
      Last Name : <input type="text" name="lastname" ><br><br>
      Password : <input type="text" name="password" ><br><br>
     Confirm Password : <input type="text" name="confirmpassword" ><br><br>
     Roll :  <div><input type="radio" id="admin" name="roll" value="admin">
  <label for="admin">Admin</label><br>
  <input type="radio" id="user" name="roll" value="user">
  <label for="user">User</label> </div><br><br>
      Mobile No : <input  type="text" name="mobileno" ><br><br>
      Email : <input  type="text" name="email" ><br><br>
      <input type="submit" class="buttonstyle"  value="SignUp">
      </form>
</body>
<script type="text/javascript">
function addFlightvalidateForm() {

     var firstname = document.forms["adduserform"]["firstname"].value;
     var lastname = document.forms["adduserform"]["lastname"].value;
     var password = document.forms["adduserform"]["password"].value;
     var confirmpassword =
document.forms["adduserform"]["confirmpassword"].value;
     var roll = document.forms["adduserform"]["roll"].value;
     var mobileno = document.forms["adduserform"]["mobileno"].value;
     var email = document.forms["adduserform"]["email"].value;


     if (firstname == "") {
          alert("firstname must be filled out");
          return false;
        }
     if (lastname == "") {
          alert("lastname must be filled out");
          return false;
        }
     if (password == "") {
          alert("password must be filled out");
          return false;
        }
        if (confirmpassword == "") {
             alert("confirmpassword must be filled out");
             return false;
           }
        if (roll == "") {
             alert("roll must be filled out");
             return false;
           }
        if (mobileno == "") {
             alert("mobileno must be filled out");
             return false;
           }
        if (email == "") {
             alert("email must be filled out");
             return false;
            }

   }
</script>
</html>
```

updateProductFrom.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Update Product</title>
<style><%@include file="/WEB-INF/style.css"%></style>
</head>
<body>
   <%@include file="adminHeader.jsp"%>
      <form name="addProductform" action="updatetoNewproduct"
onsubmit="return addFlightvalidateForm()">
         <b>Update Product Here</b><br>
         <br> Product ID : <input class="textboxstyle" type="text"
            name="prod_id" value="${men_women_kids_ShoeDataname.id}"><br>
         Product Category :

            <input type="radio" id="men" name="weartype" value="Mens_Wear">
            <label for="men">Mens Wear</label><br> <input type="radio"
               id="women" name="weartype" value="Womens_Wear"> <label
               for="women">Womens Wear</label> <br> <input type="radio"
               id="kid" name="weartype" value="Kids_Wear"> <label
               for="kid">Kids Wear</label>

         <br> Product Name : <input class="textboxstyle" type="text"
            name="prod_name"
value="${men_women_kids_ShoeDataname.name}"><br>
         <br> Product Price : <input class="textboxstyle" type="text"
            name="prod_price"
value="${men_women_kids_ShoeDataname.price}"><br>
         <br> Img Link : <input class="textboxstyle" type="text"
            name="prod_img"
value="${men_women_kids_ShoeDataname.imagelink}"><br>
         <br> <input type="submit" value="Update Product">
      </form>
</body>

<script type="text/javascript">
   function addFlightvalidateForm() {
      var weartype = document.forms["addProductform"]["weartype"].value;
      var prod_name = document.forms["addProductform"]["prod_name"].value;
      var prod_price =
document.forms["addProductform"]["prod_price"].value;
      var prod_img = document.forms["addProductform"]["prod_img"].value;

      if (weartype == "") {
         alert("weartype must be filled out");
         return false;
      }
      if (prod_name == "") {
         alert("prod_name must be filled out");
         return false;
      }
      if (prod_price == "") {
         alert("prod_price must be filled out");
         return false;
      }
      if (prod_img == "") {
         alert("prod_img must be filled out");
         return false;
      }

   }
</script>
</html>
```