Source Code for Sporty Shoes

1. Controller Code

CommonController.java

package com.mjava.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.mjava.model.OrderedShoeModel;

import com.mjava.model.ShoesDataModel;

import com.mjava.model.UserInfoModel;

import com.mjava.service.OrdersService;

import com.mjava.service.ShoesService;

import com.mjava.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.mjava.controllers.CommonController.selectedShoeInfotMethod(HttpServletRequest, 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) {

        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);

            model.put("orderedshoemodel", orderedshoemodel);

            return "bookShoeForm";

        } catch (Exception e) {

            System.out.println(

                    "Exception at com.mjava.controllers.CommonController.orderInforMethod(HttpServletRequest, HttpServletResponse) "

                            + e.getMessage());

            return "bookShoeForm";

        }

    }

    @GetMapping("/bookandpaymentreturntoHome")

    // @RequestMapping(value = "/bookandpaymentreturntoHome", method =

    // RequestMethod.GET)

    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) {

        try {

            long millis = System.currentTimeMillis();

            Date date = new java.sql.Date(millis);

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

            return "orderplaceForm";

        } catch (Exception e) {

            System.out.println(

                    "Exception at com.mjava.controllers.CommonController.bookAndPaymentreturntoHomeMethod(HttpServletRequest, 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.mjava.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.mjava.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.mjava.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.mjava.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...");

            }

            return "customersListForm";

        } catch (Exception e) {

            System.out.println(

                    "Exception at com.mjava.controllers.CommonController.customersListMethod(HttpServletRequest, 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.mjava.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.mjava.controllers.CommonController.changeadminPasswordMethod(HttpServletRequest, 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.mjava.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.mjava.controllers.CommonController.addNewProducttMethod(HttpServletRequest, 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.mjava.controllers.CommonController.sigUpNewUserMethod(HttpServletRequest, 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.mjava.controllers.CommonController.deleteProductMethod() " + e.getMessage());

            return "adminForm";

        }

    }

    @GetMapping("/updateProduct")

    public String updateProductMethod(ModelMap model, @RequestParam int id) {

        boolean isupdated = false;

        ShoesDataModel shoesData = null;

        // ShoesDataModel findwhichCategeory = null;

        try {

            shoesData = shoeservice.getshoesDataByIdService(id);

            if (shoesData.getId() > 0) {

                model.put("men\_women\_kids\_ShoeDataname", shoesData);

                return "updateProductForm";

            }

//          int categeory = findwhichCategeory.getCategeory();

//          isdeleted = shoeservice.deleteProductwithIdService(id);

//          if (isdeleted == true) {

//              shoesDatalist = shoeservice.getShoeDatabyCategeory(categeory);

//

//              model.put("men\_women\_kids\_ShoeDataname", shoesDatalist);

//

//          }

//

//          return "adminForm";

            return "adminForm";

        } catch (Exception e) {

            System.out.println(

                    "Exception at com.mjava.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;

        // ShoesDataModel findwhichCategeory = 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.mjava.controllers.CommonController.deleteProductMethod() " + e.getMessage());

            return "adminForm";

        }

    }

}

1. Dao codes

OrdersDao.java

package com.mjava.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.mjava.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.mjava.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.mjava.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.mjava.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.mjava.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);

}

1. Model codes

OrderedShoeModel.java

package com.mjava.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 String name;

    private String imagelink;

    private double totalprice ;

    public int getQuantity() {

        return quantity;

    }

    public void setQuantity(int quantity) {

        this.quantity = quantity;

    }

    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 + "]";

    }

    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) {

        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;

    }

}

ShoesDataModel.java

package com.mjava.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

package com.mjava.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;

    }

}

1. Service Codes

OrdersService.java

package com.mjava.service;

import java.sql.Date;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.mjava.dao.OrdersDao;

import com.mjava.model.OrderedShoeModel;

import com.mjava.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 )

     {

         OrderedShoeModel  orderedShoeModel =new  OrderedShoeModel(shoeid, categeory, price, quantity, date, name, imglink, totalprice);

         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.mjava.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.mjava.dao.OrdersDao;

import com.mjava.dao.ShoesDao;

import com.mjava.dao.UsersDao;

import com.mjava.model.OrderedShoeModel;

import com.mjava.model.ShoesDataModel;

import com.mjava.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.mjava.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.mjava.dao.OrdersDao;

import com.mjava.dao.UsersDao;

import com.mjava.model.OrderedShoeModel;

import com.mjava.model.ShoesDataModel;

import com.mjava.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);

    }

}

1. Exception Handling Code

GlobalExceptionHandler.java

package com.mjava.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";

    }

}

1. SportyShoes Spring Application Code

SportyShoesOnlineApplication.java

package com.mjava.SportyShoesOnline;

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.mjava")

@EntityScan("com.mjava.model")

@EnableJpaRepositories("com.mjava.dao")

public class SportyShoesOnlineApplication  {

    public static void main(String[] args) {

        SpringApplication.run(SportyShoesOnlineApplication.class, args);

    }

}

1. Property file code

Application.properties

logging.level.org.springframework.web: DEBUG

spring.mvc.view.prefix=/WEB-INF/view/

spring.mvc.view.suffix=.jsp

server.port=8080

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoes\_db

spring.datasource.username=root

spring.datasource.password=root

1. Mysql Database Queries

To create a database :

create schema sportyshoes\_db;

Creating tables:

use sportyshoes\_db;

create table tbl\_orderinfo (

orderid int AUTO\_INCREMENT PRIMARY KEY,

shoeid int,

categeory int,

price double,

quantity int,

date date,

name varchar(30),

imagelink varchar(300),

totalprice double

);

create table tbl\_shoesinfo (

id int AUTO\_INCREMENT PRIMARY KEY,

categeory int,

price double,

name varchar(30),

imagelink varchar(300)

);

create table tbl\_userinfo (

id int AUTO\_INCREMENT PRIMARY KEY,

firstname varchar(30),

lastname varchar(30),

password varchar(30),

rollid int,

phno long,

email varchar(30)

);

1. JSP codes

Index.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

    pageEncoding="ISO-8859-1" errorPage="Error.jsp"%>

<html>

<title>Home</title>

<body>

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

    <hr>

        <a href="menshoes" >

        <img src="https://th.bing.com/th/id/OIP.x0Xme1w50KvWwpYGOys9dwHaHa?pid=ImgDet&rs=1" style="float: left; width: 30%; margin-right: 1%; margin-bottom: 0.5em;">

        </a>

        <a href="womenshoes">

                <img src="https://ae01.alicdn.com/kf/HTB1YzeoeEcKL1JjSZFzq6AfJXXaa/2017-Women-Running-Shoes-Breathable-Women-Sneaker-Outdoor-Sports-Shoes-Ladies-White-Shoes-Comfortable-Sports-Athletic.jpg" style="float: left; width: 30%; margin-right: 1%; margin-bottom: 0.5em;">

        </a>

        <a href="kidshoes" > <img src="https://th.bing.com/th/id/OIP.RtE-mIvi0qOytqf8dkuqewHaHa?pid=ImgDet&rs=1https://th.bing.com/th/id/OIP.RtE-mIvi0qOytqf8dkuqewHaHa?pid=ImgDet&rs=1" style="float: left; width: 30%; margin-right: 1%; margin-bottom: 0.5em;">

        </a>

</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>

</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>

signUpForm.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>

</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>

signinForm.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>

</head>

<body>

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

            <form action="adminsignin"  >

            <hr>

           <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">

           </form>

</body>

</html>

adminForm.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>Admin Form</title>

<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>

</head>

<body>

<h3><a href="home" style="text-decoration:none;color:black">Home</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&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>

addNewProductForm.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>Add New Product</title>

</head>

<body>

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

        <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>

</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>

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>

table, td,th {

  border: 1px solid black;

  border-collapse: collapse;

}

</style>

</head>

<body>

 <%@include file="adminHeader.jsp" %>

<h1>Customers Information</h1>

<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>

</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>

</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>

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>

</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>

</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>

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>

</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>

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>

table, td,th {

  border: 1px solid black;

  border-collapse: collapse;

}

</style>

</head>

<body>

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

    <h1>Sales Report</h1>

    <form action="requiredreports" name="searchform"

        onsubmit="return reportvalidateForm()">

        <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">

        <hr>

    </form>

    <table>

        <tr>

            <th>Orderid</th>

            <th>Shoeid</th>

            <th>Name</th>

            <th>Price</th>

            <th>Quantity</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.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>

updateProductForm.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>

</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>

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>

</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><br><br>

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

</form>

</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>

</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>Categeroy :<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="totalprice"  value="${orderedshoemodel.totalprice}"  />

 <h2>Total Price :<label  >${orderedshoemodel.totalprice}</label> </h2> <br><br>

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

</form>

</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>

</head>

<body>

<h1>Payment Done.Order Placed </h1>

<h2>Thanks  for the Shopping..... </h2><br>

<br>

<h2 align="center"> <a href="home" style="text-decoration:none">Back to Home</a> </h2>

</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>

</head>

<body>

        <h1>Error</h1>

        <h2><%=exception.getMessage() %><br/> </h2>

</body>

</html>