package com.sportyshoes.controller;

import java.sql.Timestamp;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import java.util.TimeZone;

import javax.validation.Valid;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.validation.BindingResult;

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

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

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

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

import com.sportyshoes.DAO.SportyShoesDAO;

import com.sportyshoes.api.Admin;

import com.sportyshoes.api.OrderHistory;

import com.sportyshoes.api.Products;

import com.sportyshoes.api.User;

@Controller

public class SportyShoesController {

private static int temp;

@Autowired

SportyShoesDAO dao;

@GetMapping("/")

public String homePage() {

return "home";

}

@GetMapping("/adminLoginin")

public String adminLoginPage(@ModelAttribute("admin") Admin admin) {

return "adminLogin";

}

@GetMapping("/adminLogincheck")

public String adminCheck(@Valid @ModelAttribute("admin")Admin admin,BindingResult result) {

Admin admin1=dao.getAdmin();

if(admin.getName().equals(admin1.getName())&&admin.getPassword().equals(admin1.getPassword())){

admin.setCheck(true);

return "adminHomePage";

}

else

return "adminLogin";

}

@GetMapping("/userlogin")

public String userLogin(@ModelAttribute("user")User user) {

return "userLogin";

}

@GetMapping("/userLogincheck")

public String userCheck(@Valid @ModelAttribute("user")User user,BindingResult result,Model model) {

User user1=dao.checkUser(user);

if(user1!=null) {

List<OrderHistory> list=dao.purcReportsUser(user1.getId());

model.addAttribute("list",list);

user.setCheck(true);

return "userHomePage";

}

else {

return "userLogin";

}

}

@GetMapping("/registerUser")

public String registerUser(@ModelAttribute("user")User user) {

return "register-User";

}

@GetMapping("userRegistered")

public String userSaved(@ModelAttribute("user") User user,BindingResult result) {

dao.saveUser(user);

return "registerSuccess";

}

@GetMapping("/browseuserlist")

public String browseUserList(Model model) {

List<User> list=dao.userList();

model.addAttribute("userlist",list);

return "show-user-list";

}

@GetMapping("manageproducts")

public String products(Model model) {

model.addAttribute("product",dao.ProductList());

return "products";

}

@GetMapping("add")

public String addForm(@ModelAttribute("product")Products product) {

return "add-form";

}

@GetMapping("addedProduct")

public String add(@ModelAttribute("product")Products product) {

dao.addProduct(product);

return "added-success";

}

@GetMapping("showbuycat")

public String buyPage(Model model) {

model.addAttribute("product",dao.ProductList());

return "shop";

}

@GetMapping("orderConfirm")

public String loginuser(@ModelAttribute("productId") @RequestParam("productId") int productId,@ModelAttribute("user") User user) {

temp=productId;

return "userLoginForm";

}

@PostMapping("userLogcheck")

public String orderConfirmed(@Valid @ModelAttribute("user") User user,BindingResult result,Model model) {

User user1=dao.checkUser(user);

if(user1!=null) {

Products product=dao.getProduct(temp);

Date date=new Date();

Calendar cal=Calendar.getInstance(TimeZone.getTimeZone("Asia/Kolkata"));

cal.setTime(date);

OrderHistory history=new OrderHistory();

history.setUserId(user1.getId());

history.setUserName(user1.getName());

history.setProductId(product.getId());

history.setProductName(product.getName());

history.setProductPrice(product.getPrice());

history.setProductCategory(product.getCategory());

history.setPurchaseTime(new Timestamp(122, cal.get(Calendar.MONTH), cal.get(Calendar.DAY\_OF\_MONTH), cal.get(Calendar.HOUR), cal.get(Calendar.MINUTE), cal.get(Calendar.SECOND),0));

model.addAttribute("history",history);

dao.saveHistory(history);

user.setCheck(true);

return "order-confirm";

}

else {

return "userLoginForm";

}

}

@GetMapping("purchasereports")

public String purchaseReports(Model model) {

List<OrderHistory> list=dao.purcReports();

model.addAttribute("orderlist",list);

return "order-reports";

}

}