User.Java

package com.anurag.sportyshoes.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "user")

public class User {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name="id")

    private int id;

    @Column(name="user\_type")

    private int userType=2;

    @Column(name="first\_name")

    private String firstName;

    @Column(name="last\_name")

    private String lastName;

    @Column(name="email")

    private String email;

    @Column(name="password")

    private String password;

    @Column(name="product\_purchased")

    private String productPurchased;

    public User() {}

    public User(String firstName, String lastName, String email, String password) {

        this.firstName = firstName;

        this.lastName = lastName;

        this.email = email;

        this.password = password;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public int getUserType() {

        return userType;

    }

    public void setUserType(int userType) {

        this.userType = userType;

    }

    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 getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getPassword() {

        return password;

    }

    public void setPassword(String password) {

        this.password = password;

    }

    public String getProductPurchased() {

        return productPurchased;

    }

    public void setProductPurchased(String productPurchased) {

        this.productPurchased = productPurchased;

    }

    @Override

    public String toString() {

        return "User [id=" + id + ", userType=" + userType + ", firstName=" + firstName + ", lastName=" + lastName

                + ", email=" + email + ", password=" + password + ", productPurchased=" + productPurchased + "]";

    }

}

Product. Java

package com.anurag.sportyshoes.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "product")

public class Product {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name="id")

    private int id;

    @Column(name="name")

    private String name;

    @Column(name = "size")

    private int size;

    @Column(name="category")

    private String category;

    @Column(name="price")

    private int price;

    @Column(name="quantity\_available")

    private String quantityAvailable;

    public Product() {}

    public Product(String name, int size, String category, int price, String quantityAvailable) {

        this.name = name;

        this.size = size;

        this.category = category;

        this.price = price;

        this.quantityAvailable = quantityAvailable;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getSize() {

        return size;

    }

    public void setSize(int size) {

        this.size = size;

    }

    public String getCategory() {

        return category;

    }

    public void setCategory(String category) {

        this.category = category;

    }

    public int getPrice() {

        return price;

    }

    public void setPrice(int price) {

        this.price = price;

    }

    public String getQuantityAvailable() {

        return quantityAvailable;

    }

    public void setQuantityAvailable(String quantityAvailable) {

        this.quantityAvailable = quantityAvailable;

    }

    @Override

    public String toString() {

        return "Product [id=" + id + ", name=" + name + ", size=" + size + ", category=" + category + ", price=" + price

                + ", quantityAvailable=" + quantityAvailable + "]";

    }

}

Purchase Detail.java

package com.anurag.sportyshoes.entity;

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class PurchaseDetail {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private int purchaseId;

    private int purchasedByUserID;

    private int purchasedProductID;

    private Date purchaseDate;

    public PurchaseDetail(int purchaseId, int purchasedByUserID, int purchasedProductID, Date purchaseDate) {

        this.purchaseId = purchaseId;

        this.purchasedByUserID = purchasedByUserID;

        this.purchasedProductID = purchasedProductID;

        this.purchaseDate = purchaseDate;

    }

    public PurchaseDetail(int purchasedByUserID, int purchasedProductID, Date purchaseDate) {

        this.purchasedByUserID = purchasedByUserID;

        this.purchasedProductID = purchasedProductID;

        this.purchaseDate = purchaseDate;

    }

    public PurchaseDetail() {

        // TODO Auto-generated constructor stub

    }

    @Override

    public String toString() {

        return "PurchaseDetail [purchaseId=" + purchaseId + ", purchasedByUserID=" + purchasedByUserID

                + ", purchasedProductID=" + purchasedProductID + ", purchaseDate=" + purchaseDate + "]";

    }

    public int getPurchaseId() {

        return purchaseId;

    }

    public void setPurchaseId(int purchaseId) {

        this.purchaseId = purchaseId;

    }

    public int getPurchasedByUserID() {

        return purchasedByUserID;

    }

    public void setPurchasedByUserID(int purchasedByUserID) {

        this.purchasedByUserID = purchasedByUserID;

    }

    public int getPurchasedProductID() {

        return purchasedProductID;

    }

    public void setPurchasedProductID(int purchasedProductID) {

        this.purchasedProductID = purchasedProductID;

    }

    public Date getPurchaseDate() {

        return purchaseDate;

    }

    public void setPurchaseDate(Date purchaseDate) {

        this.purchaseDate = purchaseDate;

    }

}

package com.anurag.sportyshoes.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

import com.anurag.sportyshoes.entity.User;

@Controller

public class LoginController {

    @GetMapping("/showMyLoginPage")

    public String showMyLoginPage() {

        return "login";

    }

    @GetMapping("/register")

    public String register(Model theModel)

    {

        System.out.println("register Handler Executed");

        User theUser = new User();

        theModel.addAttribute("user", theUser);

        return "registration-form";

    }

    @GetMapping("/logout")

    public String logout(HttpServletRequest request, HttpServletResponse response) {

        if(request.getSession(false)!=null) {

            request.getSession(false).invalidate();

            return "redirect:/showMyLoginPage";

        }

        else {

            return "redirect:/showMyLoginPage";

        }

    }

}

package com.anurag.sportyshoes.controller;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

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

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

import com.anurag.sportyshoes.entity.Product;

import com.anurag.sportyshoes.service.ProductService;

@Controller

@RequestMapping("/product")

public class ProductController {

    // inject the customer service

    @Autowired

    private ProductService productService;

    @GetMapping("/list")

    public String listProducts(Model theModel) {

        //get customers from the service

        List<Product> theProducts= productService.getProducts();

        //add the customer to the model

        theModel.addAttribute("products", theProducts);

        return "list-products";

    }

    @GetMapping("/productAddForm")

    public String productAddForm(Model theModel) {

        //create model attribute to bind the form data

        Product theProduct = new Product();

        theModel.addAttribute("product", theProduct);

        return "product-form";

    }

    @PostMapping("/saveProduct")

    public String saveProduct(@ModelAttribute("product") Product theProduct) {

        // save the productusing the service

        productService.saveProduct(theProduct);

        return "redirect:/product/list";

    }

    @GetMapping("showFormForUpdate")

    public String showFormForUpdate(@RequestParam("productID") int theID, Model theModel) {

        //get the customer from the service

        Product theProduct = productService.getProduct(theID);

        //set  customer as a model attribute to pre-populate the form

        theModel.addAttribute("product", theProduct);

        //send over to our form

        return "product-form";

    }

    @GetMapping("/delete")

    public String deleteProduct(@RequestParam("productID") int theId) {

        //delete the product

        productService.deleteProduct(theId);

        return "redirect:/product/list";

    }

    @GetMapping("/buy")

    public String buyProduct(@RequestParam("productID") int theId) {

        // buy the product by updating the time stamp against the user id

        productService.buyProduct(theId);

        return "redirect:user/store";

    }

    package com.anurag.sportyshoes.controller;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.util.List;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

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

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

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

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

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

import com.anurag.sportyshoes.dao.PurchaseDetailDAO;

import com.anurag.sportyshoes.entity.Product;

import com.anurag.sportyshoes.entity.PurchaseDetail;

import com.anurag.sportyshoes.entity.User;

import com.anurag.sportyshoes.service.ProductService;

import com.anurag.sportyshoes.service.UserService;

@Controller

@RequestMapping("/user")

public class UserController {

    // inject the customer service

    @Autowired

    private UserService userService;

    @Autowired

    private ProductService productService;

    @Autowired

    private PurchaseDetailDAO purchaseDetailDao;

    @GetMapping("/list")

    public String listCustomers(Model theModel) {

        // get customers from the service

        List<User> theUsers = userService.getUsers();

        // add the customer to the model

        theModel.addAttribute("users", theUsers);

        return "list-users";

    }

    @GetMapping("/userAddForm")

    public String userAddForm(Model theModel) {

        // Create the model attrubute to bind the data

        User theUser = new User();

        theModel.addAttribute("user", theUser);

        return "registration-form";

    }

    @PostMapping("/saveUser")

    public String saveUser(@ModelAttribute("user") User theUser) {

        userService.saveUser(theUser);

        return "store";

    }

    @GetMapping("/search")

    public String searchUsers(@RequestParam("theSearchName") String theSearchName, Model theModel) {

        // search customers from the service

        List<User> theUsers = userService.searchUsers(theSearchName);

        // add the customers to the model

        theModel.addAttribute("users", theUsers);

        return "list-users";

    }

    @GetMapping("/store")

    String store(Model model)

    {

        model.addAttribute("products", productService.getProducts());

        return "store";

    }

      @PostMapping("/authenticateTheUser") public String authenticateTheUser(Model

      theModel,HttpServletRequest request,HttpServletResponse response, @RequestParam("email") String email,@RequestParam("password") String password) {

          System.out.println("Email = "+email + " password = " + password);

          User u=userService.validateLogin(email, password);

          System.out.println("Received User :: "+u);

          if(u==null) {

              System.out.println("Invalid email or password");

              return "redirect:/showMyLoginPage";

          }else if(u.getUserType() == 1) {

              System.out.println("Admin login Authenticated");

              request.getSession(true).setAttribute("user\_admin", u);

              return "redirect:/product/list";

          }else {

              System.out.println("Customer login Authenticated");

              request.getSession(true).setAttribute("user\_customer", u);

              return "redirect:/user/store";

          }

      }

      @PostMapping("/purchaseProduct")

      public String purchaseProduct(@RequestParam ("purchaseByUserId") String userId, @RequestParam("purchaseProductId") String purchaseproductID) {

          System.out.println("user ID "+userId+" purchase Product ID "+purchaseproductID);

          Date date = new Date();

          System.out.println(date);

          PurchaseDetail pd1= new PurchaseDetail(Integer.parseInt(userId), Integer.parseInt(purchaseproductID), date);

          System.out.println(pd1);

          int x = purchaseDetailDao.save(pd1);

          System.out.println("Generated purchase Id "+x);

          return "redirect:/user/store";

      }

      @GetMapping("/changePassword")

      public String changePass() {

          return "change-password-form";

      }

      @PostMapping("/changePassword")

      public String changePassword(Model

              theModel,HttpServletRequest request,HttpServletResponse response, @RequestParam("new-password") String newPass, @RequestParam("confirm-password") String confirmPass) {

          if(newPass.equals(confirmPass)) {

                    User u =(User) request.getSession(false).getAttribute("user\_admin");

                        u.setPassword(newPass);

                        userService.saveUser(u);

                        System.out.println("Password Updated success");

          }

          return "redirect:/product/list";

      }

      @GetMapping("/listPurchasesByDate")

      public String listpurchasebydate(Model model)

      {

          System.out.println("ListPurchasessbyDate Executed");

          List<PurchaseDetail> purchaseList = purchaseDetailDao.getAllPurchases();

          List<User> userpurchaselist = new ArrayList<>();

          List<Product> productpurchaselist = new ArrayList<>();

          for(PurchaseDetail pd : purchaseList)

          {

             User u1= userService.getuserById(pd.getPurchasedByUserID());

             userpurchaselist.add(u1);

             Product p1=productService.getProduct(pd.getPurchasedProductID());

             productpurchaselist.add(p1);

          }

          purchaseList.forEach(pl-> {System.out.println(pl);});

          userpurchaselist.forEach(up->{System.out.println(up);});

          productpurchaselist.forEach(pp ->{System.out.println(pp);});

          model.addAttribute("purchaselist", purchaseList);

          model.addAttribute("prodlist", productpurchaselist);

          model.addAttribute("userplist", userpurchaselist);

          return "list-purchases";

      }

}

}