sportyshoes.com
Project Source Code

Version History:

Author:	Atharv Pratap SIngh
Purpose:	Project Source Code and GITHUB
Date:	27th November, 2021
Version:	1.0

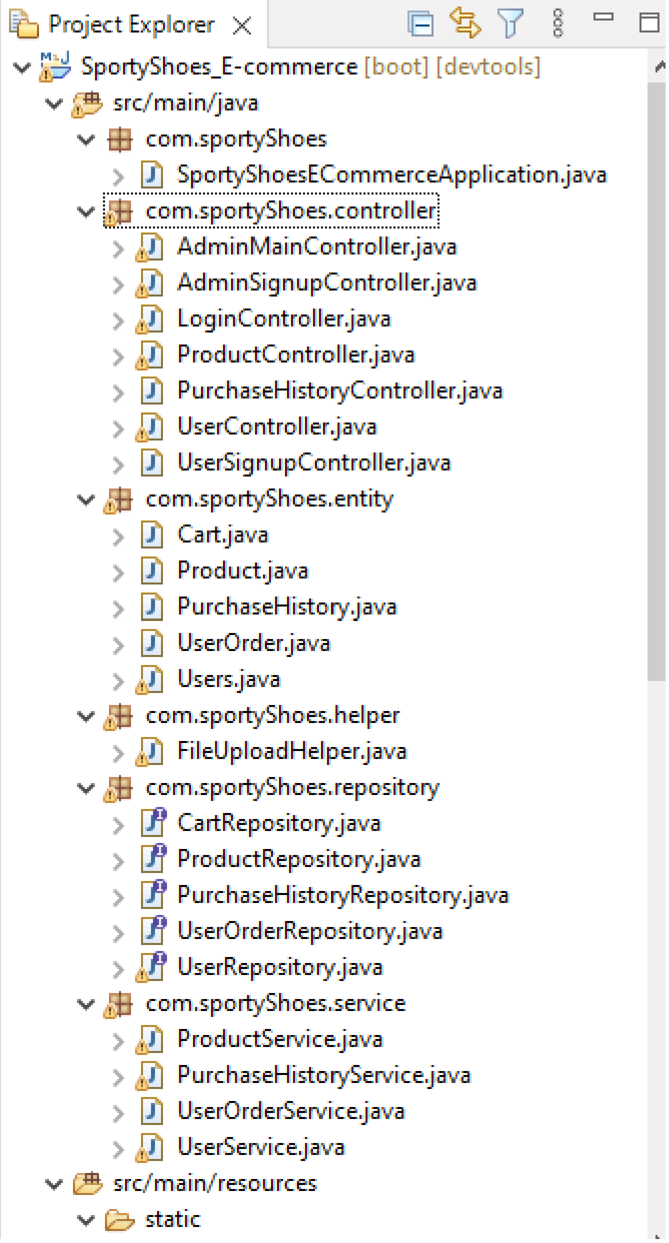
Table of Content:

Project GITHUB Link:

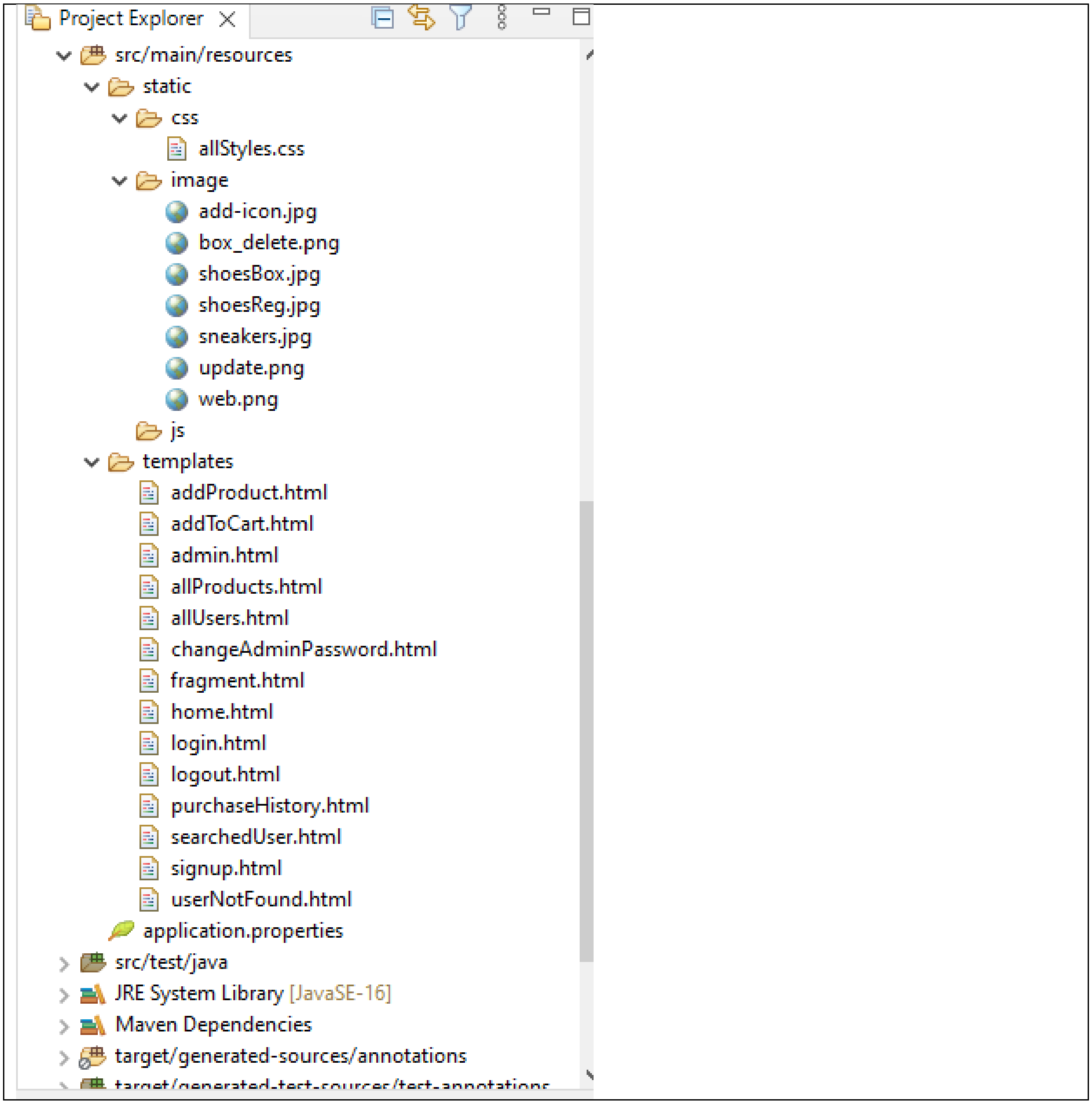
Repository Name:	SportyShoesProject
Github Link:	apstorm/Simplilearn-Phase-4: phase 4 project (github.com)

Project Code:

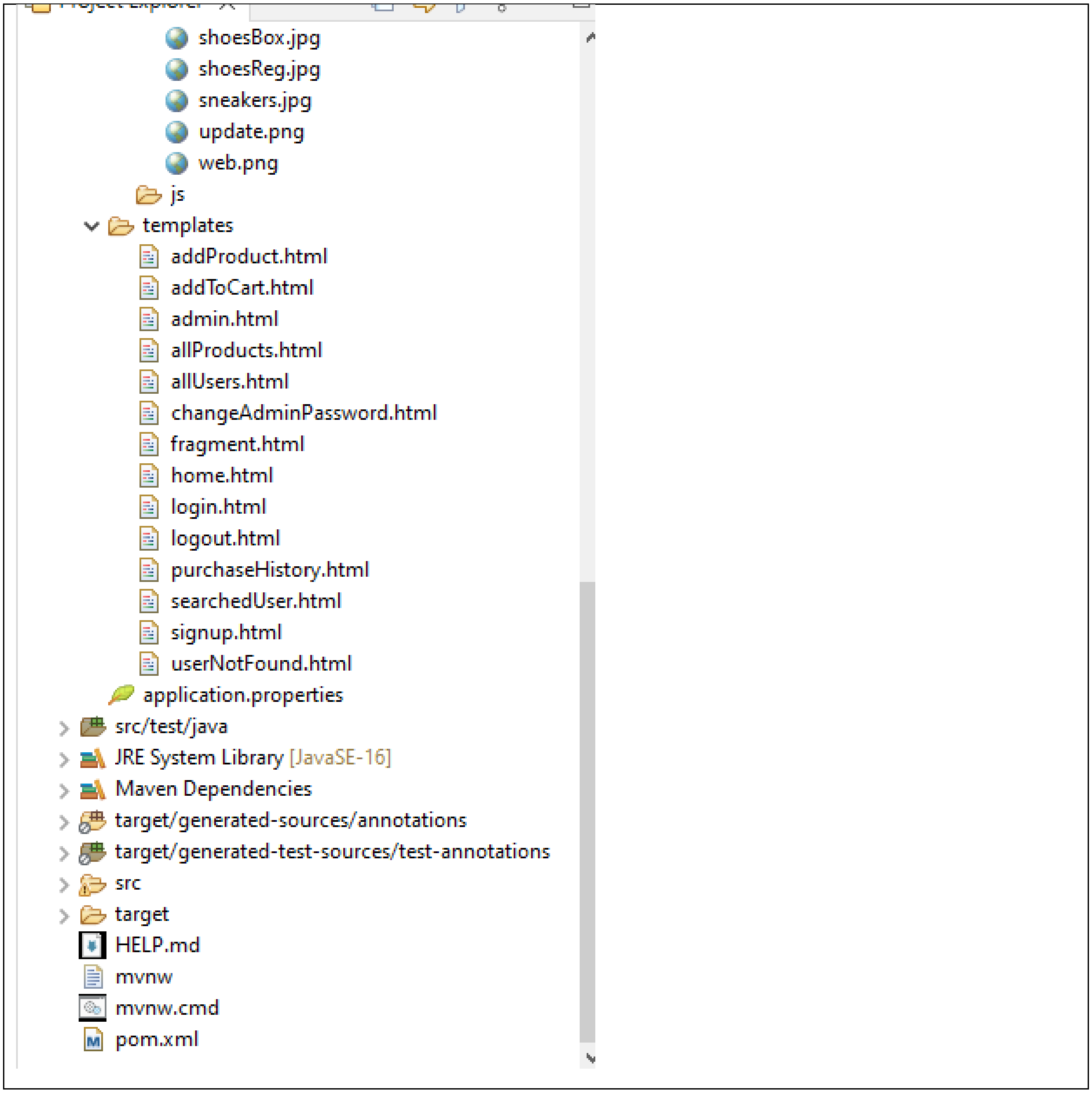
Folder Structure



```
SportyShoes_E-commerce [boot] [devtools]
  src/main/java
    com.sportyShoes
      SportyShoesECommerceApplication.java
    com.sportyShoes.controller
      AdminMainController.java
      AdminSignupController.java
      LoginController.java
      ProductController.java
      PurchaseHistoryController.java
      UserController.java
      UserSignupController.java
    com.sportyShoes.entity
      Cart.java
      Product.java
      PurchaseHistory.java
      UserOrder.java
      Users.java
    com.sportyShoes.helper
      FileUploadHelper.java
    com.sportyShoes.repository
      CartRepository.java
      ProductRepository.java
      PurchaseHistoryRepository.java
      UserOrderRepository.java
      UserRepository.java
    com.sportyShoes.service
      ProductService.java
      PurchaseHistoryService.java
      UserOrderService.java
      UserService.java
  src/main/resources
    static
```



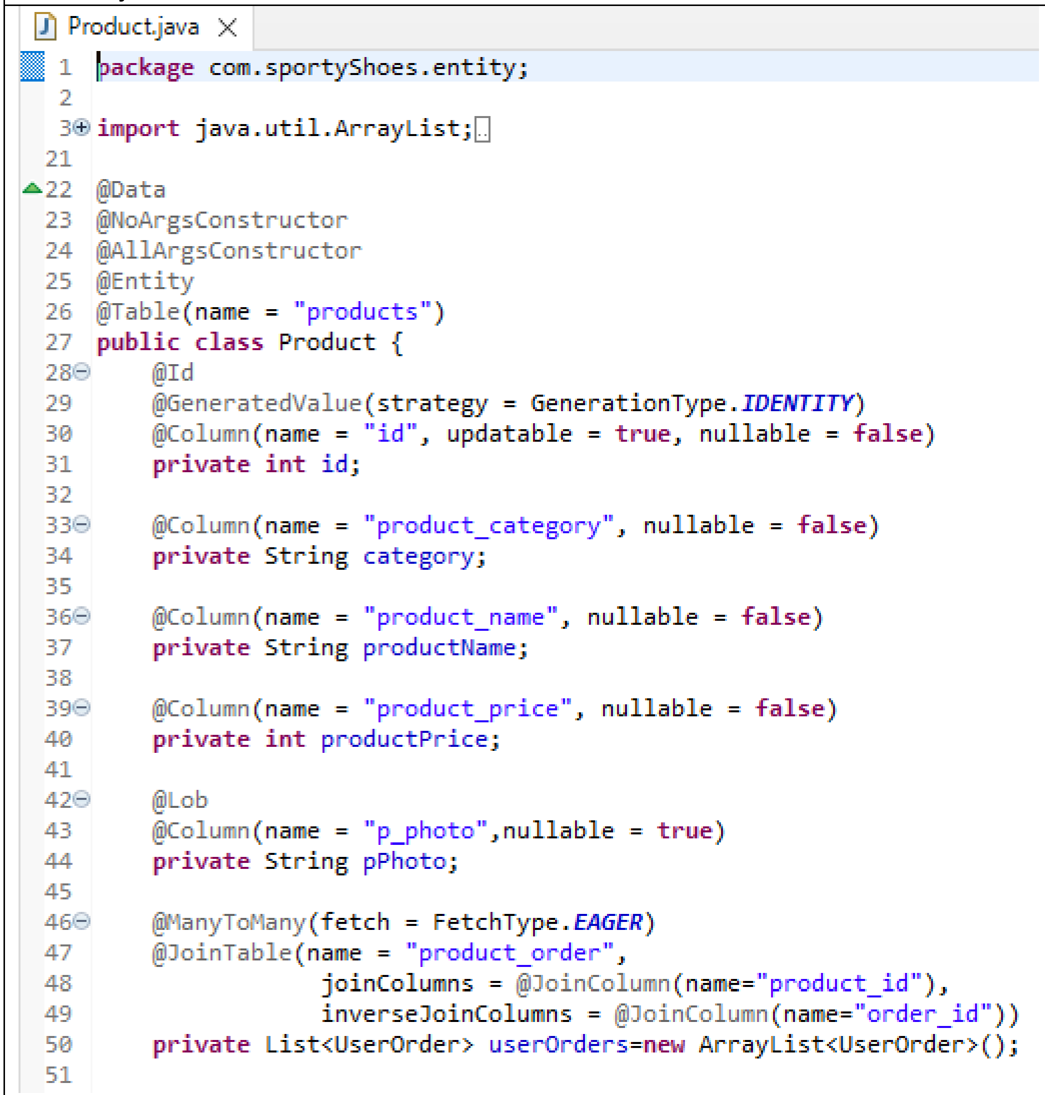
```
src/main/resources
    static
        css
            allStyles.css
        image
            add-icon.jpg
            box_delete.png
            shoesBox.jpg
            shoesReg.jpg
            sneakers.jpg
            update.png
            web.png
        js
    templates
        addProduct.html
        addToCart.html
        admin.html
        allProducts.html
        allUsers.html
        changeAdminPassword.html
        fragment.html
        home.html
        login.html
        logout.html
        purchaseHistory.html
        searchedUser.html
        signup.html
        userNotFound.html
    application.properties
src/test/java
JRE System Library [JavaSE-16]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
```



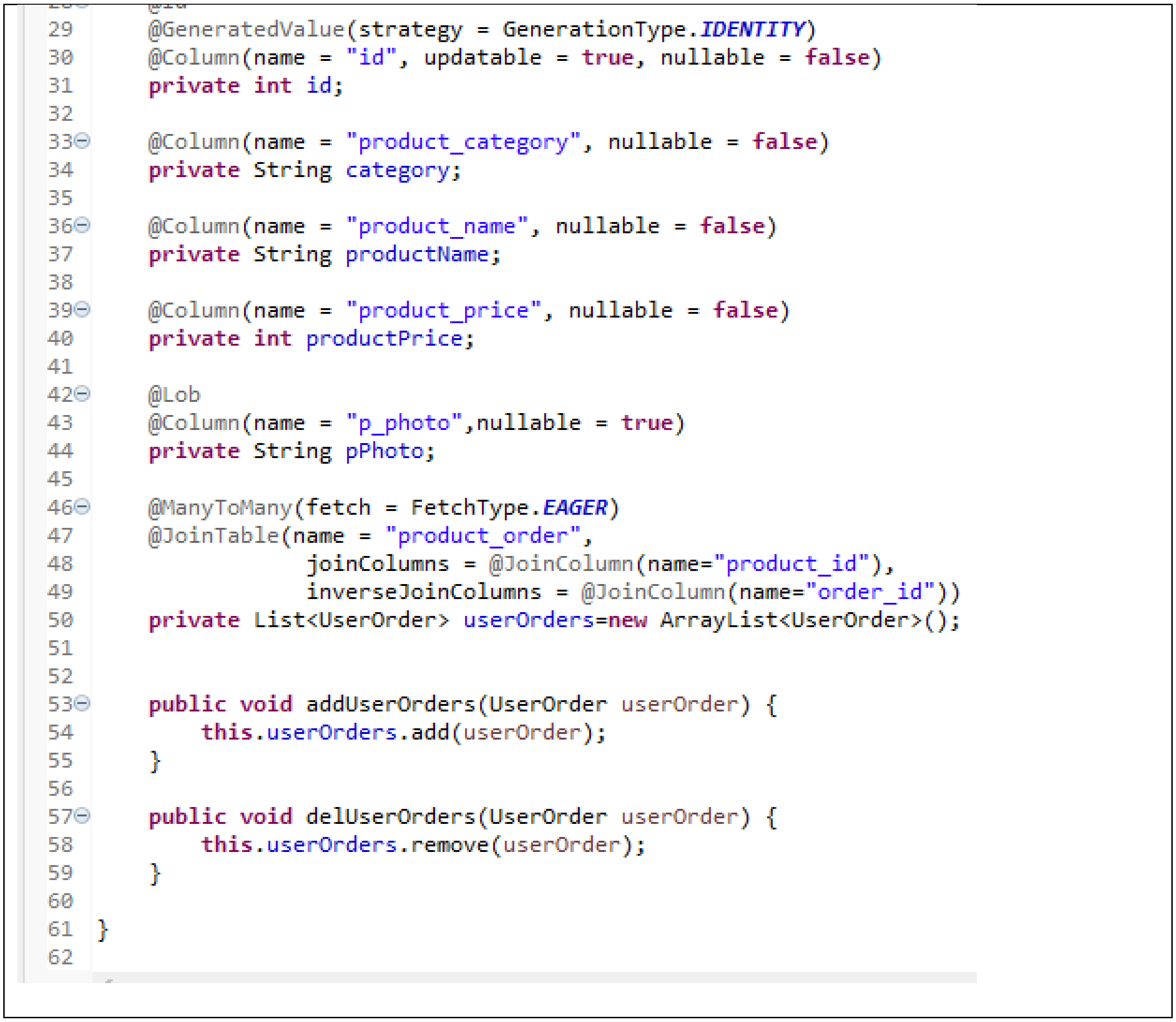
```
Project Explorer
        shoesBox.jpg
        shoesReg.jpg
        sneakers.jpg
        update.png
        web.png
      js
    templates
      addProduct.html
      addToCart.html
      admin.html
      allProducts.html
      allUsers.html
      changeAdminPassword.html
      fragment.html
      home.html
      login.html
      logout.html
      purchaseHistory.html
      searchedUser.html
      signup.html
      userNotFound.html
    application.properties
  src/test/java
  JRE System Library [JavaSE-16]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```

- ***Entity***

Product.java

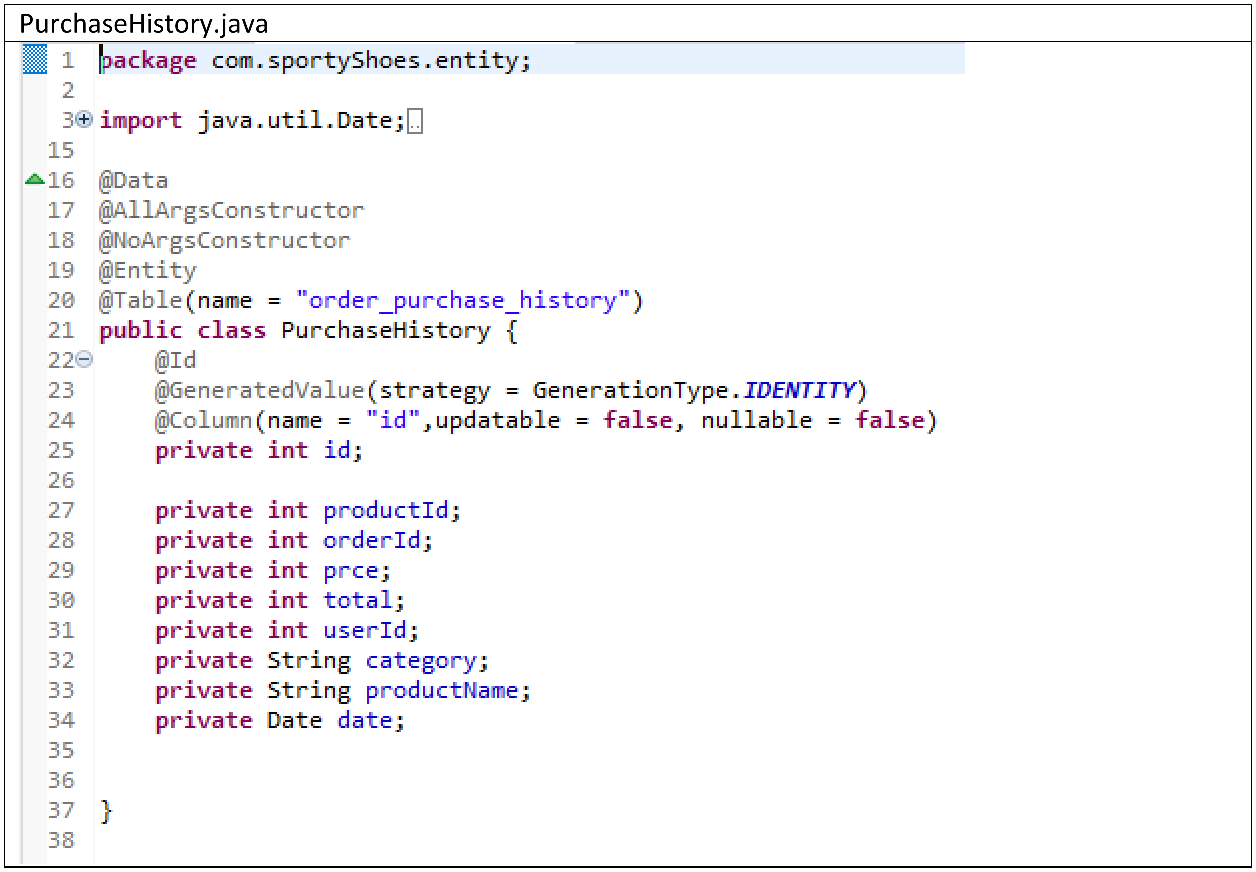


```java
package com.sportyShoes.entity;

import java.util.ArrayList;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "products")
public class Product {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", updatable = true, nullable = false)
    private int id;

    @Column(name = "product_category", nullable = false)
    private String category;

    @Column(name = "product_name", nullable = false)
    private String productName;

    @Column(name = "product_price", nullable = false)
    private int productPrice;

    @Lob
    @Column(name = "p_photo",nullable = true)
    private String pPhoto;

    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name = "product_order",
                joinColumns = @JoinColumn(name="product_id"),
                inverseJoinColumns = @JoinColumn(name="order_id"))
    private List<UserOrder> userOrders=new ArrayList<UserOrder>();

```



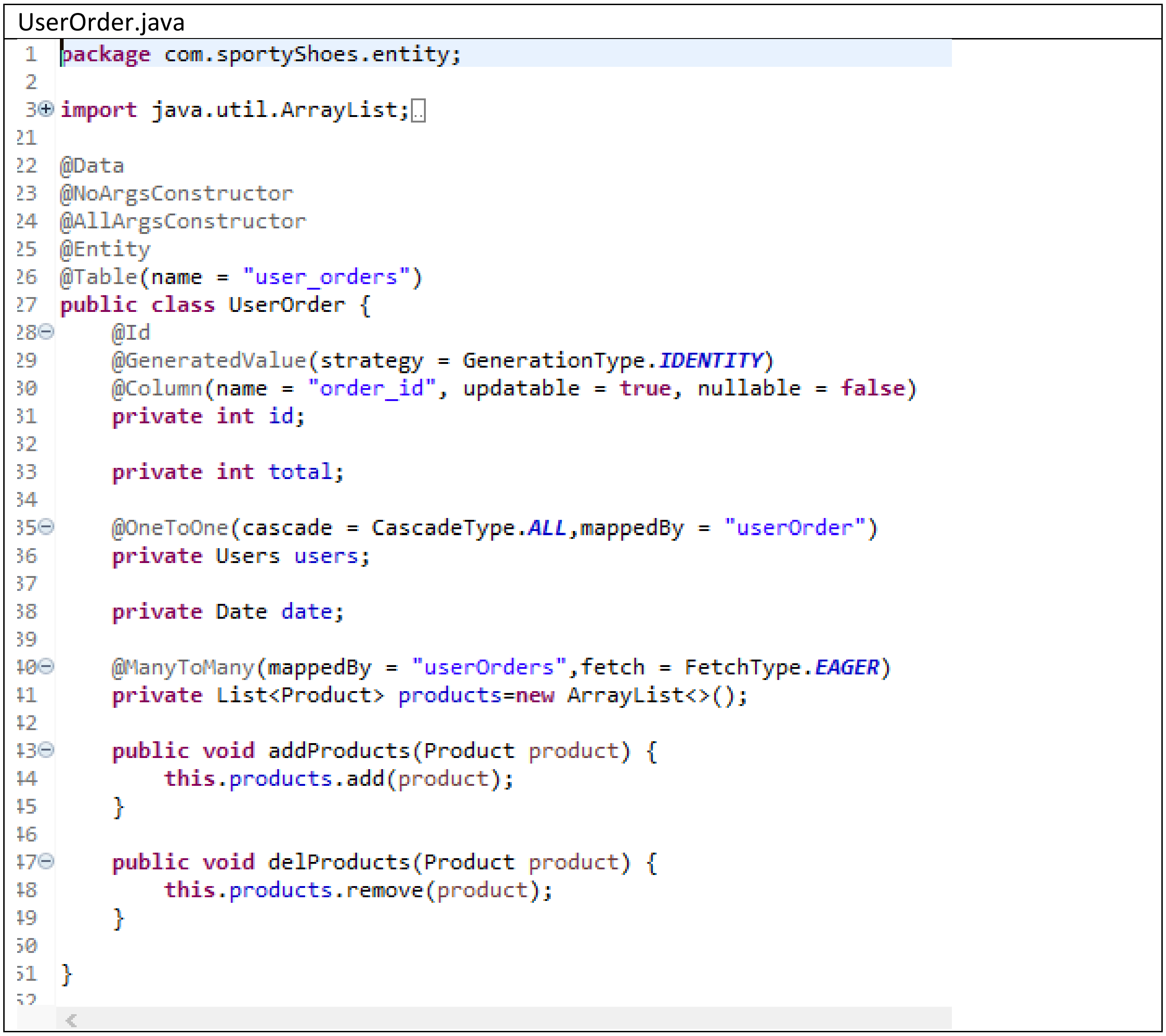
```
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", updatable = true, nullable = false)
    private int id;

    @Column(name = "product_category", nullable = false)
    private String category;

    @Column(name = "product_name", nullable = false)
    private String productName;

    @Column(name = "product_price", nullable = false)
    private int productPrice;

    @Lob
    @Column(name = "p_photo",nullable = true)
    private String pPhoto;

    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name = "product_order",
                joinColumns = @JoinColumn(name="product_id"),
                inverseJoinColumns = @JoinColumn(name="order_id"))
    private List<UserOrder> userOrders=new ArrayList<UserOrder>();


    public void addUserOrders(UserOrder userOrder) {
        this.userOrders.add(userOrder);
    }

    public void delUserOrders(UserOrder userOrder) {
        this.userOrders.remove(userOrder);
    }

}
```

PurchaseHistory.java



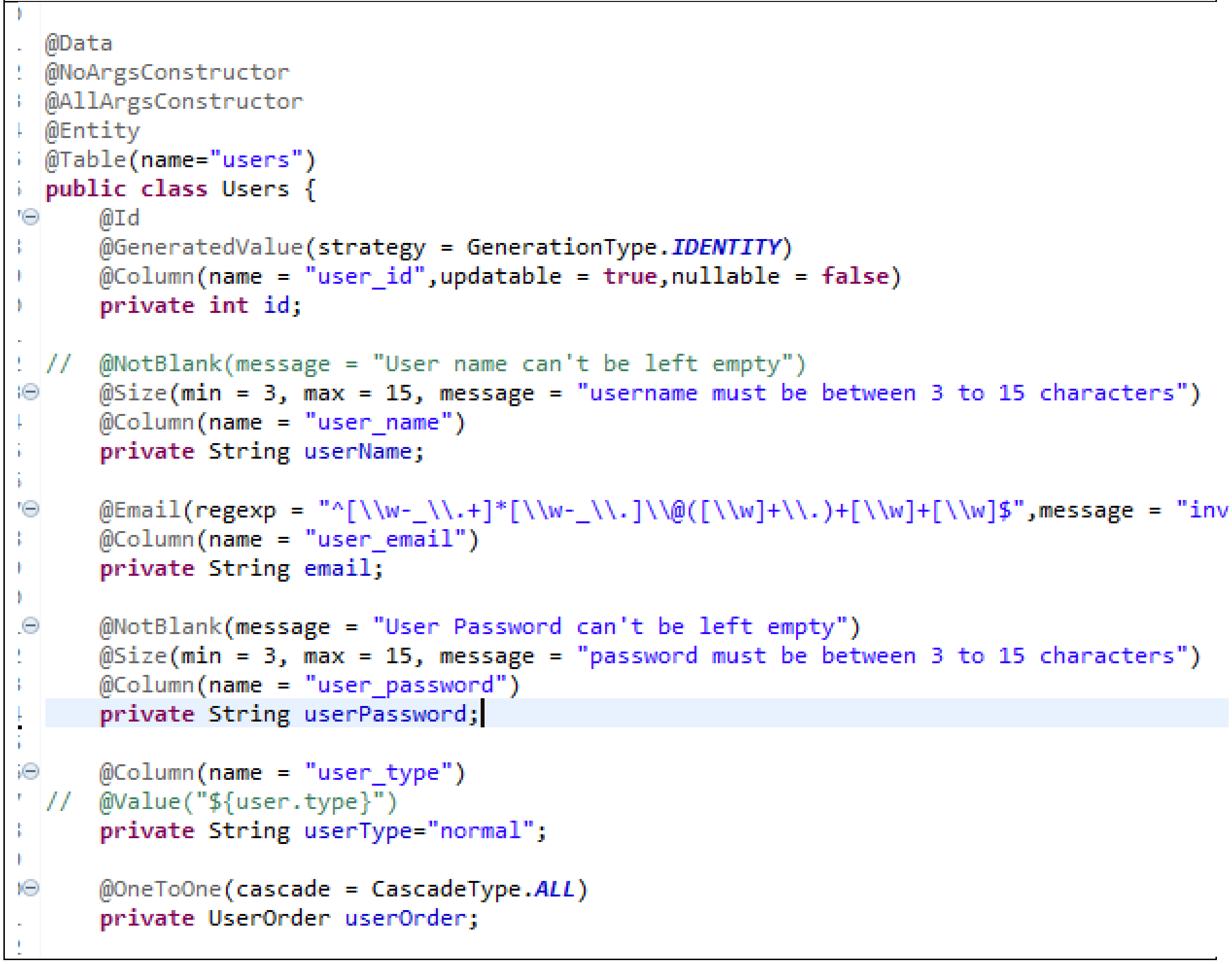
```java
package com.sportyShoes.entity;

import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Entity
@Table(name = "order_purchase_history")
public class PurchaseHistory {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id",updatable = false, nullable = false)
    private int id;

    private int productId;
    private int orderId;
    private int prce;
    private int total;
    private int userId;
    private String category;
    private String productName;
    private Date date;


}

```

UserOrder.java



```java
package com.sportyShoes.entity;

import java.util.ArrayList;

@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name = "user_orders")
public class UserOrder {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "order_id", updatable = true, nullable = false)
    private int id;

    private int total;

    @OneToOne(cascade = CascadeType.ALL,mappedBy = "userOrder")
    private Users users;

    private Date date;

    @ManyToMany(mappedBy = "userOrders",fetch = FetchType.EAGER)
    private List<Product> products=new ArrayList<>();

    public void addProducts(Product product) {
        this.products.add(product);
    }

    public void delProducts(Product product) {
        this.products.remove(product);
    }

}
```

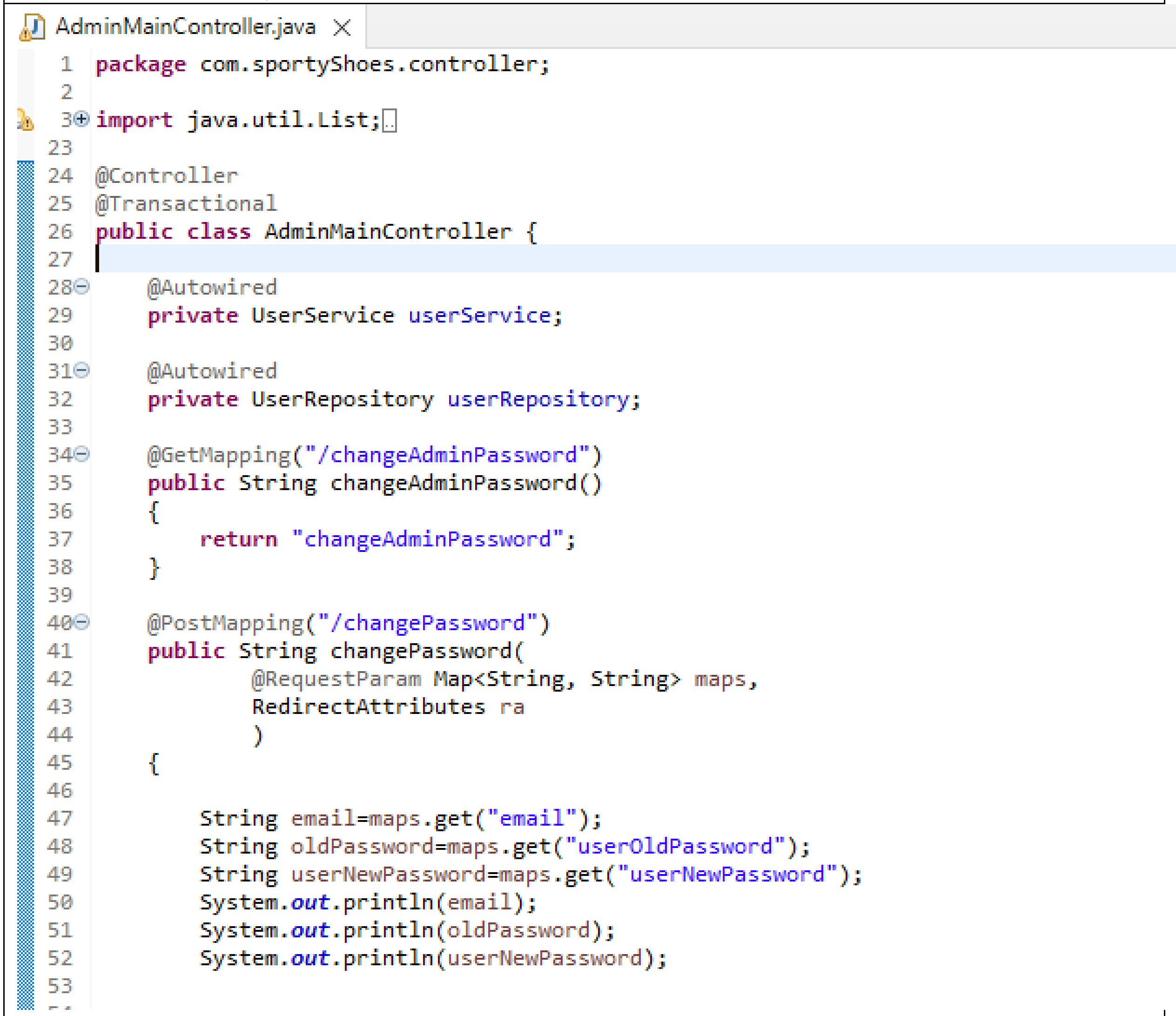
Users.java



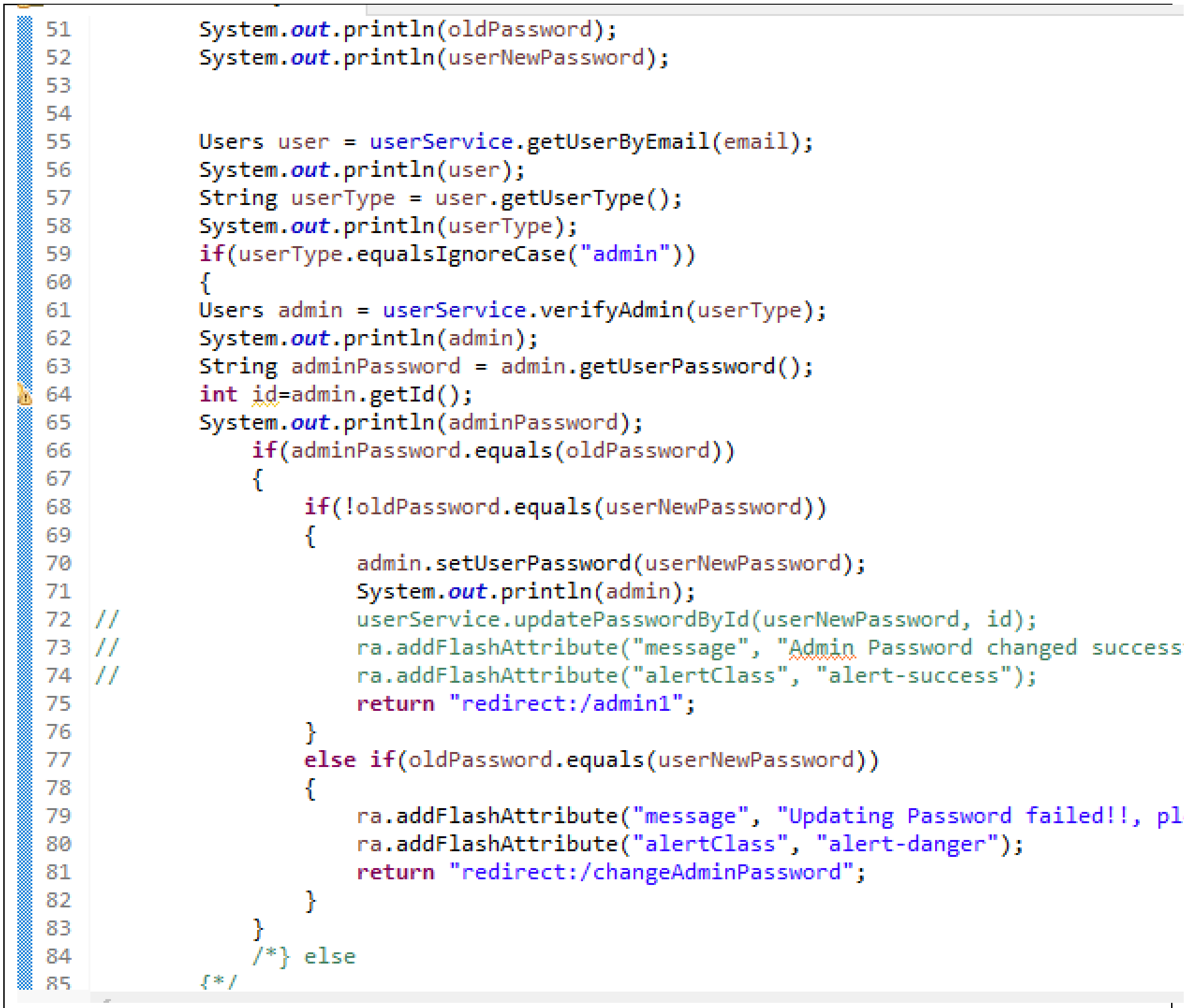
```java
@Data
@NoArgsConstructor
@AllArgsConstructor
@Entity
@Table(name="users")
public class Users {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "user_id",updatable = true,nullable = false)
    private int id;

//  @NotBlank(message = "User name can't be left empty")
    @Size(min = 3, max = 15, message = "username must be between 3 to 15 characters")
    @Column(name = "user_name")
    private String userName;

    @Email(regexp = "^[\\w-_\\.+]*[\\w-_\\.]\\@([\\w]+\\.)+[\\w]+[\\w]$",message = "inv
    @Column(name = "user_email")
    private String email;

    @NotBlank(message = "User Password can't be left empty")
    @Size(min = 3, max = 15, message = "password must be between 3 to 15 characters")
    @Column(name = "user_password")
    private String userPassword;

    @Column(name = "user_type")
//  @Value("${user.type}")
    private String userType="normal";

    @OneToOne(cascade = CascadeType.ALL)
    private UserOrder userOrder;
```

- ***CONTROLLERS***

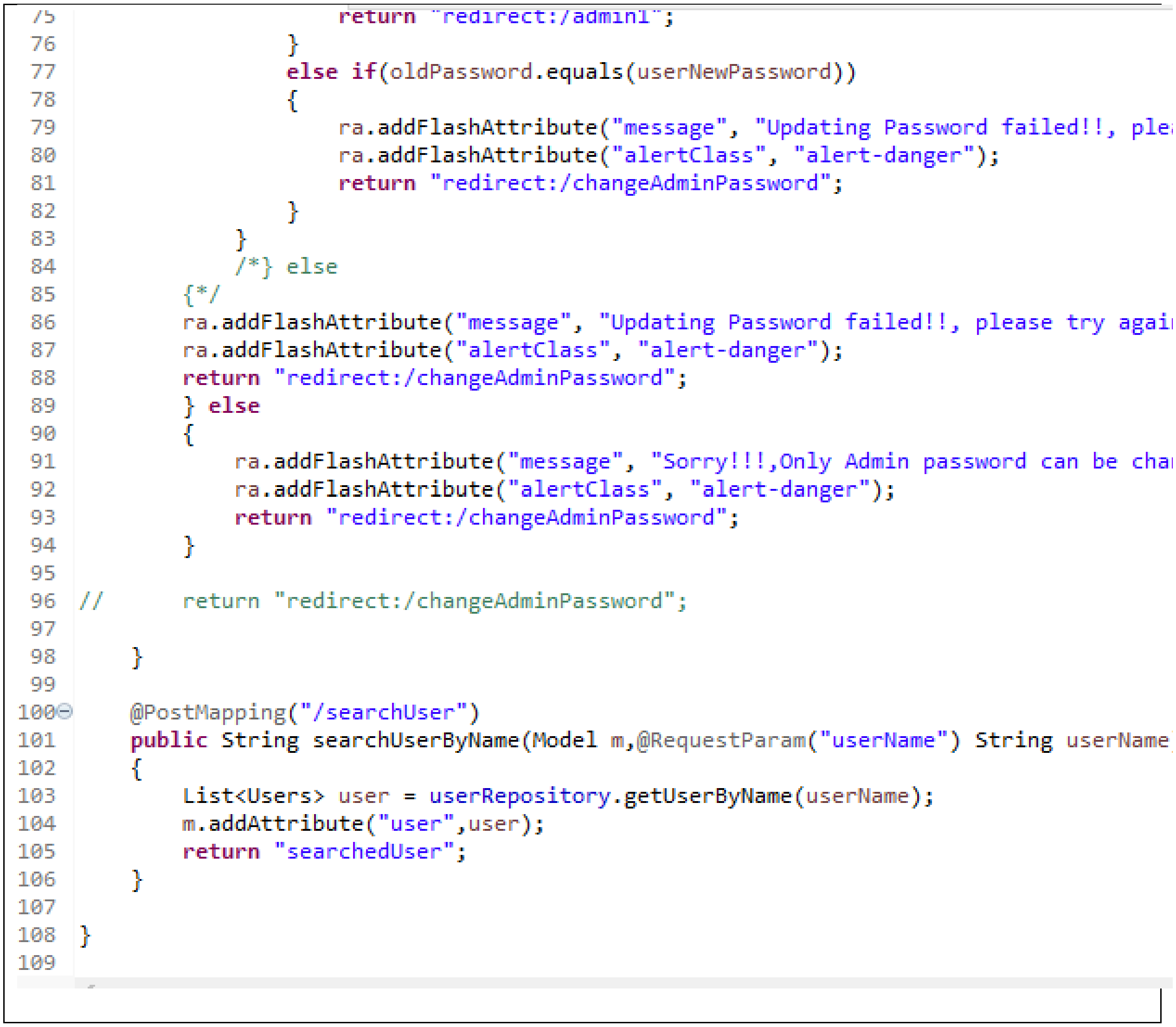
AdminMainController.java



```java
package com.sportyShoes.controller;

import java.util.List;□

@Controller
@Transactional
public class AdminMainController {

    @Autowired
    private UserService userService;

    @Autowired
    private UserRepository userRepository;

    @GetMapping("/changeAdminPassword")
    public String changeAdminPassword()
    {
        return "changeAdminPassword";
    }

    @PostMapping("/changePassword")
    public String changePassword(
            @RequestParam Map<String, String> maps,
            RedirectAttributes ra
            )
    {

        String email=maps.get("email");
        String oldPassword=maps.get("userOldPassword");
        String userNewPassword=maps.get("userNewPassword");
        System.out.println(email);
        System.out.println(oldPassword);
        System.out.println(userNewPassword);

```

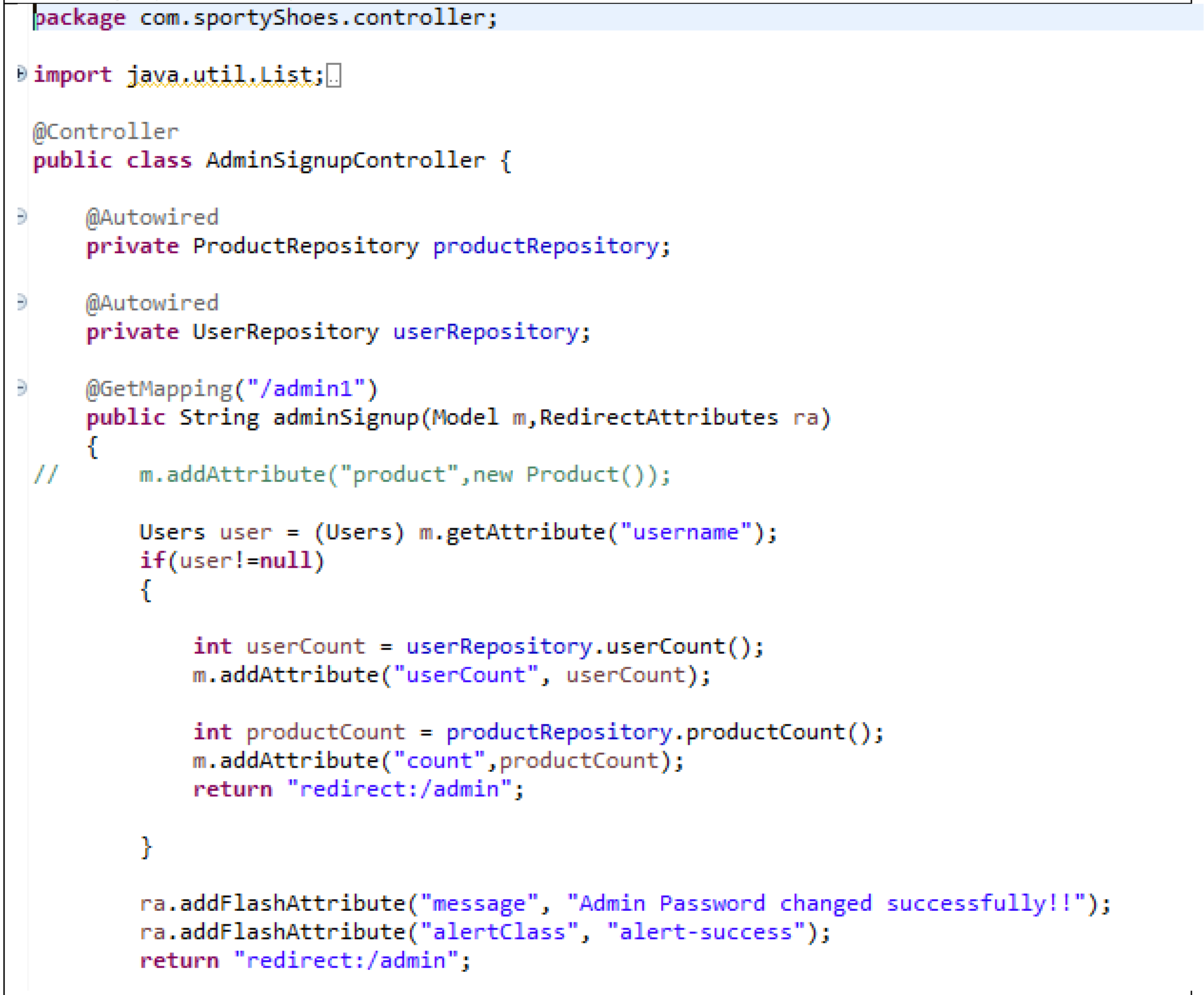


```
        System.out.println(oldPassword);
        System.out.println(userNewPassword);


        Users user = userService.getUserByEmail(email);
        System.out.println(user);
        String userType = user.getUserType();
        System.out.println(userType);
        if(userType.equalsIgnoreCase("admin"))
        {
        Users admin = userService.verifyAdmin(userType);
        System.out.println(admin);
        String adminPassword = admin.getUserPassword();
        int id=admin.getId();
        System.out.println(adminPassword);
            if(adminPassword.equals(oldPassword))
            {
                if(!oldPassword.equals(userNewPassword))
                {
                    admin.setUserPassword(userNewPassword);
                    System.out.println(admin);
//                  userService.updatePasswordById(userNewPassword, id);
//                  ra.addFlashAttribute("message", "Admin Password changed success
//                  ra.addFlashAttribute("alertClass", "alert-success");
                    return "redirect:/admin1";
                }
                else if(oldPassword.equals(userNewPassword))
                {
                    ra.addFlashAttribute("message", "Updating Password failed!!, pl
                    ra.addFlashAttribute("alertClass", "alert-danger");
                    return "redirect:/changeAdminPassword";
                }
            }
            /*} else
        {*/
```

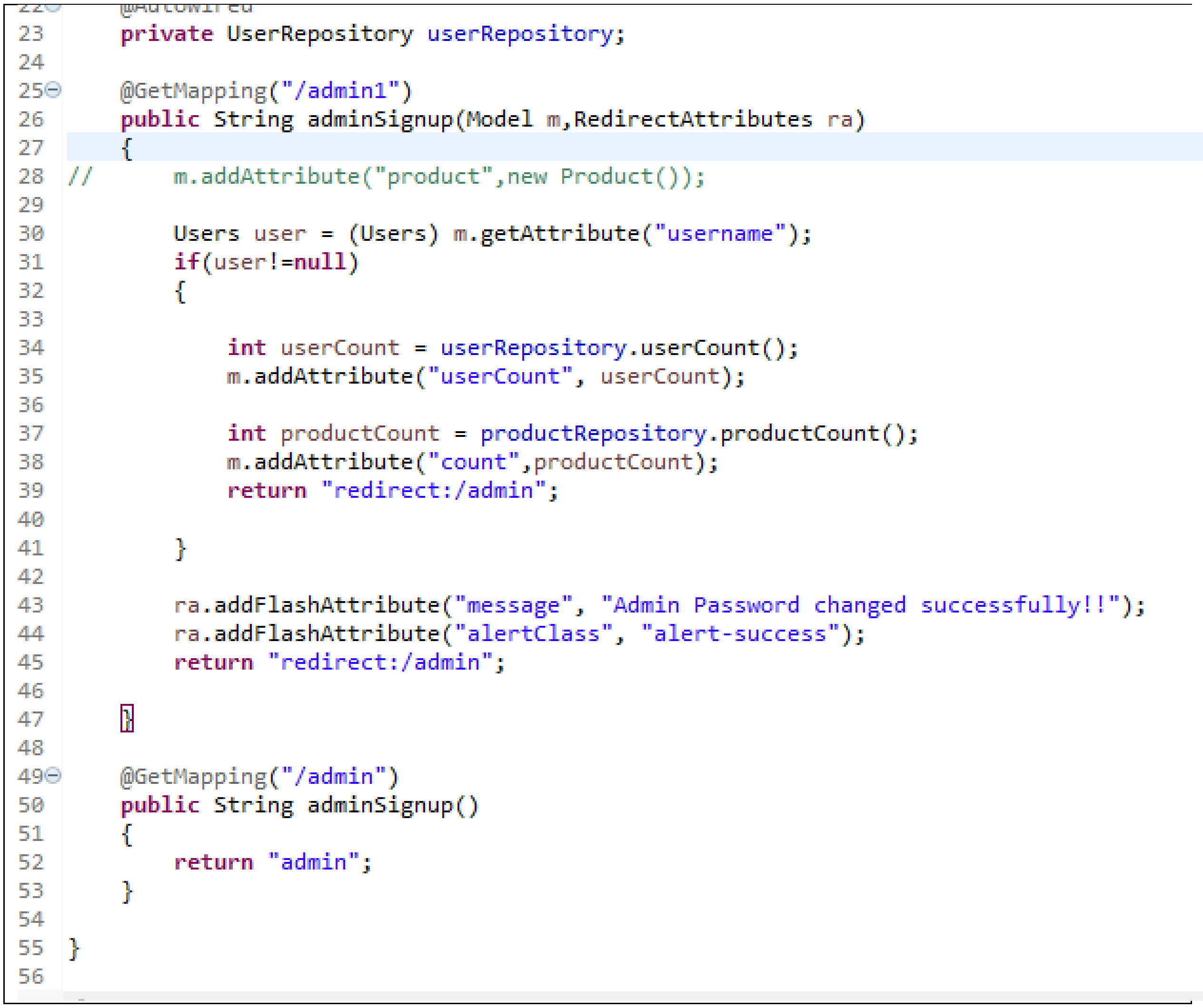


```
                    return "redirect:/admin1";
                }
                else if(oldPassword.equals(userNewPassword))
                {
                    ra.addFlashAttribute("message", "Updating Password failed!!, ple
                    ra.addFlashAttribute("alertClass", "alert-danger");
                    return "redirect:/changeAdminPassword";
                }
            }
            /*} else
        {*/
        ra.addFlashAttribute("message", "Updating Password failed!!, please try agai
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/changeAdminPassword";
        } else
        {
            ra.addFlashAttribute("message", "Sorry!!!,Only Admin password can be cha
            ra.addFlashAttribute("alertClass", "alert-danger");
            return "redirect:/changeAdminPassword";
        }

//      return "redirect:/changeAdminPassword";

    }

    @PostMapping("/searchUser")
    public String searchUserByName(Model m,@RequestParam("userName") String userName
    {
        List<Users> user = userRepository.getUserByName(userName);
        m.addAttribute("user",user);
        return "searchedUser";
    }

}
```

AdminSignupController.java

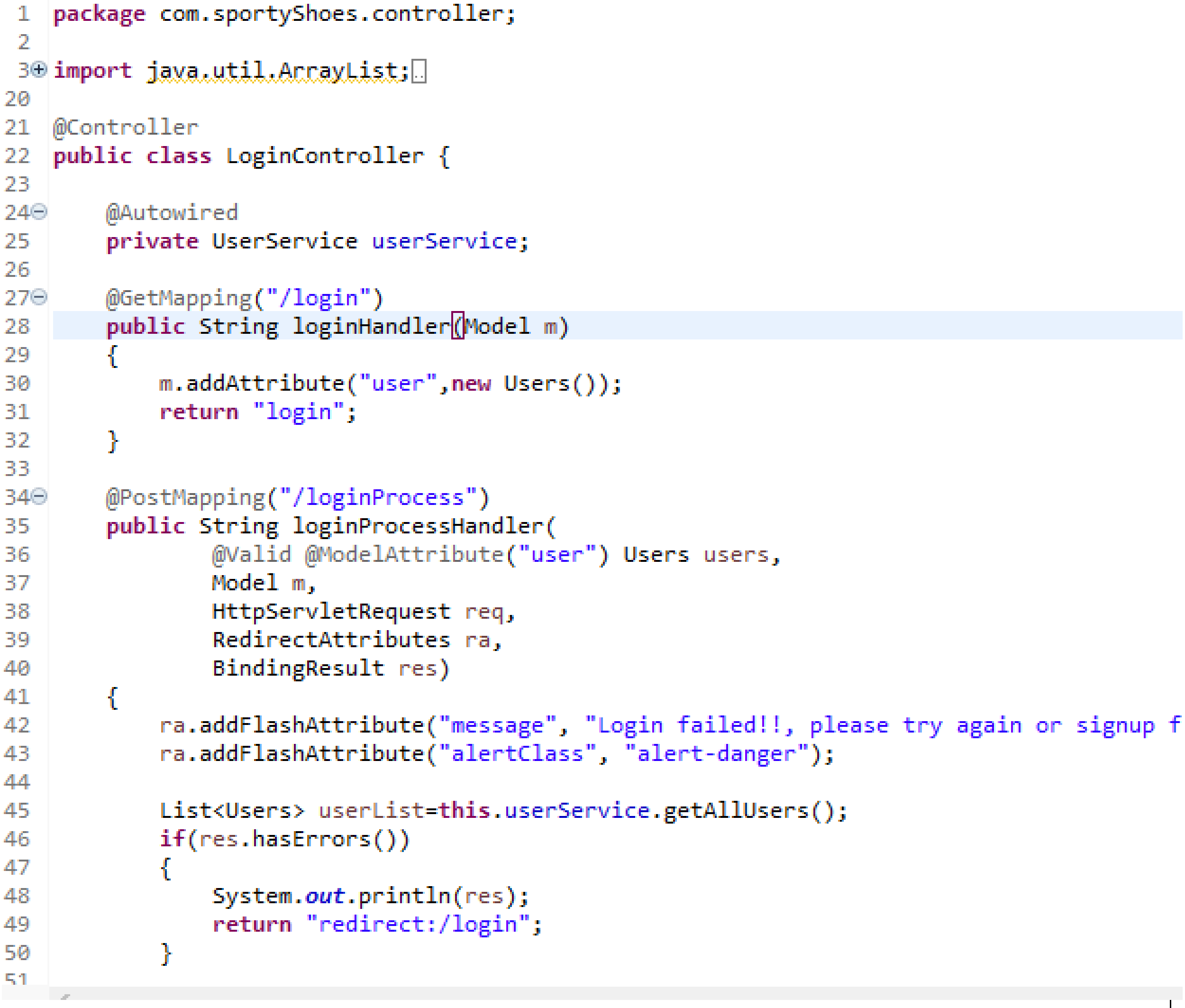


```java
package com.sportyShoes.controller;

import java.util.List;

@Controller
public class AdminSignupController {

    @Autowired
    private ProductRepository productRepository;

    @Autowired
    private UserRepository userRepository;

    @GetMapping("/admin1")
    public String adminSignup(Model m,RedirectAttributes ra)
    {
//      m.addAttribute("product",new Product());

        Users user = (Users) m.getAttribute("username");
        if(user!=null)
        {

            int userCount = userRepository.userCount();
            m.addAttribute("userCount", userCount);

            int productCount = productRepository.productCount();
            m.addAttribute("count",productCount);
            return "redirect:/admin";

        }

        ra.addFlashAttribute("message", "Admin Password changed successfully!!");
        ra.addFlashAttribute("alertClass", "alert-success");
        return "redirect:/admin";
```

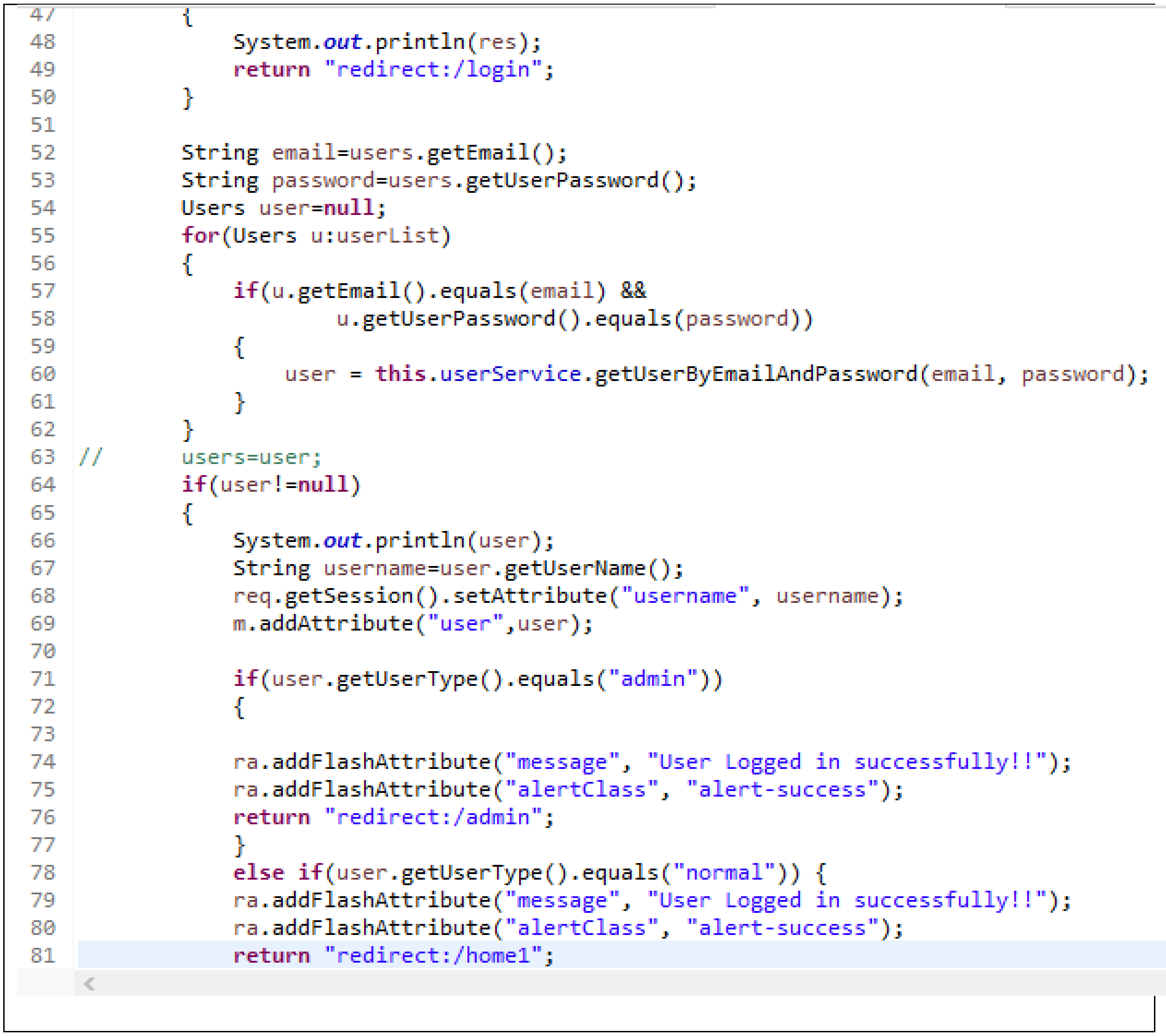


```java
    @Autowired
    private UserRepository userRepository;

    @GetMapping("/admin1")
    public String adminSignup(Model m,RedirectAttributes ra)
    {
//      m.addAttribute("product",new Product());

        Users user = (Users) m.getAttribute("username");
        if(user!=null)
        {

            int userCount = userRepository.userCount();
            m.addAttribute("userCount", userCount);

            int productCount = productRepository.productCount();
            m.addAttribute("count",productCount);
            return "redirect:/admin";

        }

        ra.addFlashAttribute("message", "Admin Password changed successfully!!");
        ra.addFlashAttribute("alertClass", "alert-success");
        return "redirect:/admin";

    }

    @GetMapping("/admin")
    public String adminSignup()
    {
        return "admin";
    }

}
```

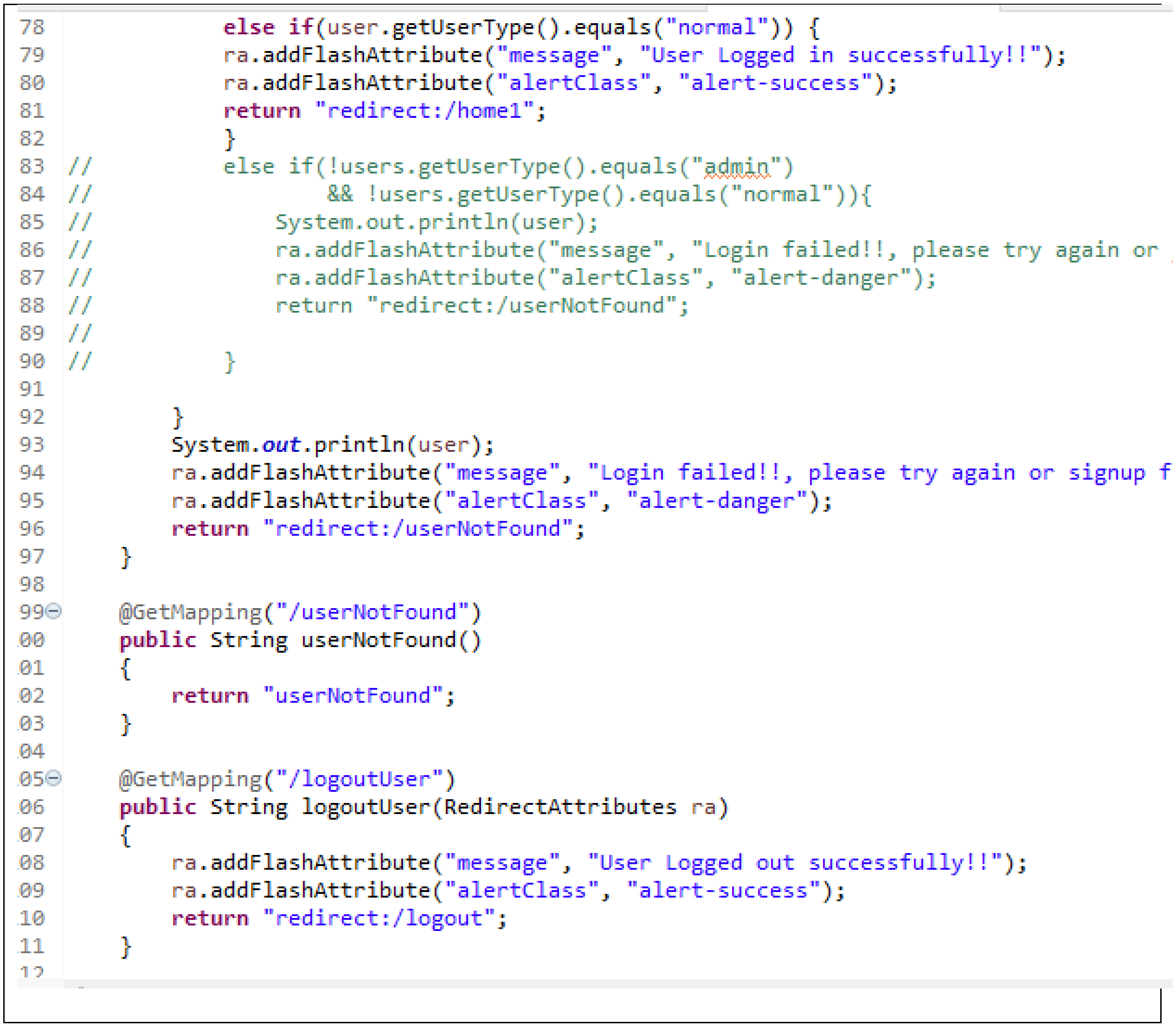
LoginController.java



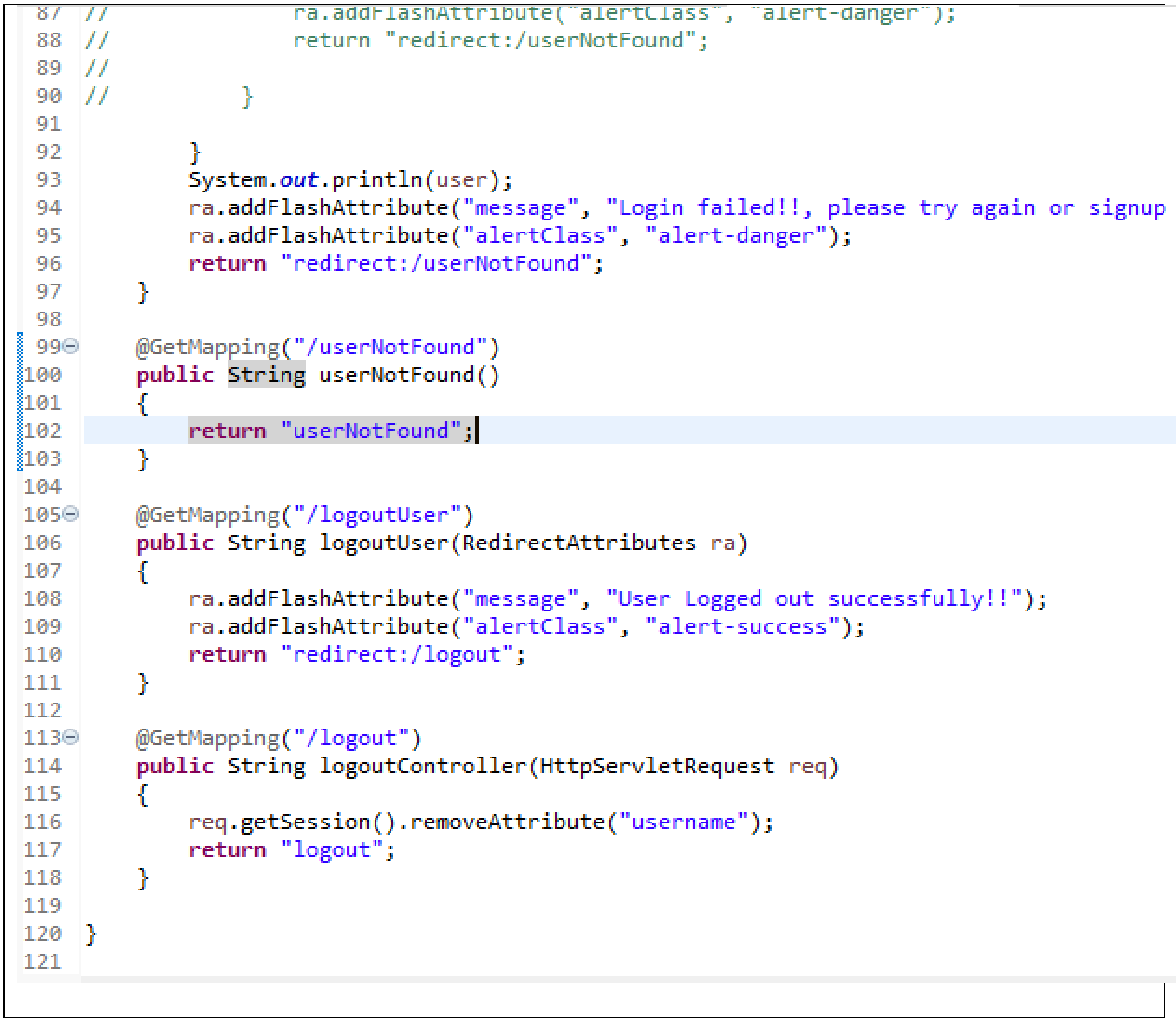
```java
package com.sportyShoes.controller;

import java.util.ArrayList;

@Controller
public class LoginController {

    @Autowired
    private UserService userService;

    @GetMapping("/login")
    public String loginHandler(Model m)
    {
        m.addAttribute("user",new Users());
        return "login";
    }

    @PostMapping("/loginProcess")
    public String loginProcessHandler(
            @Valid @ModelAttribute("user") Users users,
            Model m,
            HttpServletRequest req,
            RedirectAttributes ra,
            BindingResult res)
    {
        ra.addFlashAttribute("message", "Login failed!!, please try again or signup f
        ra.addFlashAttribute("alertClass", "alert-danger");

        List<Users> userList=this.userService.getAllUsers();
        if(res.hasErrors())
        {
            System.out.println(res);
            return "redirect:/login";
        }

```



```
        {
            System.out.println(res);
            return "redirect:/login";
        }

        String email=users.getEmail();
        String password=users.getUserPassword();
        Users user=null;
        for(Users u:userList)
        {
            if(u.getEmail().equals(email) &&
                    u.getUserPassword().equals(password))
            {
                user = this.userService.getUserByEmailAndPassword(email, password);
            }
        }
//      users=user;
        if(user!=null)
        {
            System.out.println(user);
            String username=user.getUserName();
            req.getSession().setAttribute("username", username);
            m.addAttribute("user",user);

            if(user.getUserType().equals("admin"))
            {

            ra.addFlashAttribute("message", "User Logged in successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/admin";
            }
            else if(user.getUserType().equals("normal")) {
            ra.addFlashAttribute("message", "User Logged in successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/home1";
```

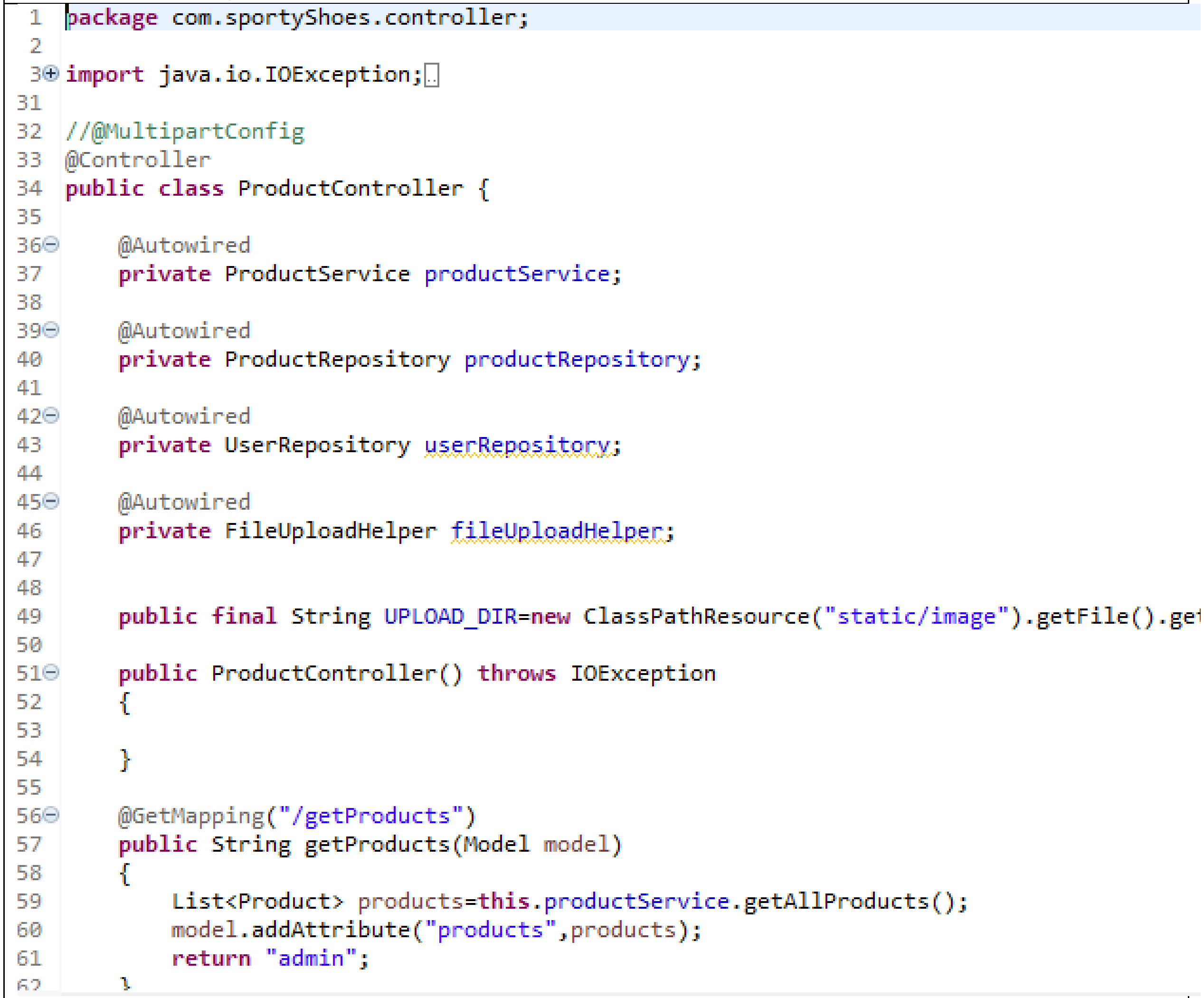


```
            else if(user.getUserType().equals("normal")) {
            ra.addFlashAttribute("message", "User Logged in successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/home1";
            }
//          else if(!users.getUserType().equals("admin")
//                  && !users.getUserType().equals("normal")){
//              System.out.println(user);
//              ra.addFlashAttribute("message", "Login failed!!, please try again or .
//              ra.addFlashAttribute("alertClass", "alert-danger");
//              return "redirect:/userNotFound";
//
//          }

        }
        System.out.println(user);
        ra.addFlashAttribute("message", "Login failed!!, please try again or signup f
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/userNotFound";
    }

    @GetMapping("/userNotFound")
    public String userNotFound()
    {
        return "userNotFound";
    }

    @GetMapping("/logoutUser")
    public String logoutUser(RedirectAttributes ra)
    {
        ra.addFlashAttribute("message", "User Logged out successfully!!");
        ra.addFlashAttribute("alertClass", "alert-success");
        return "redirect:/logout";
    }
```

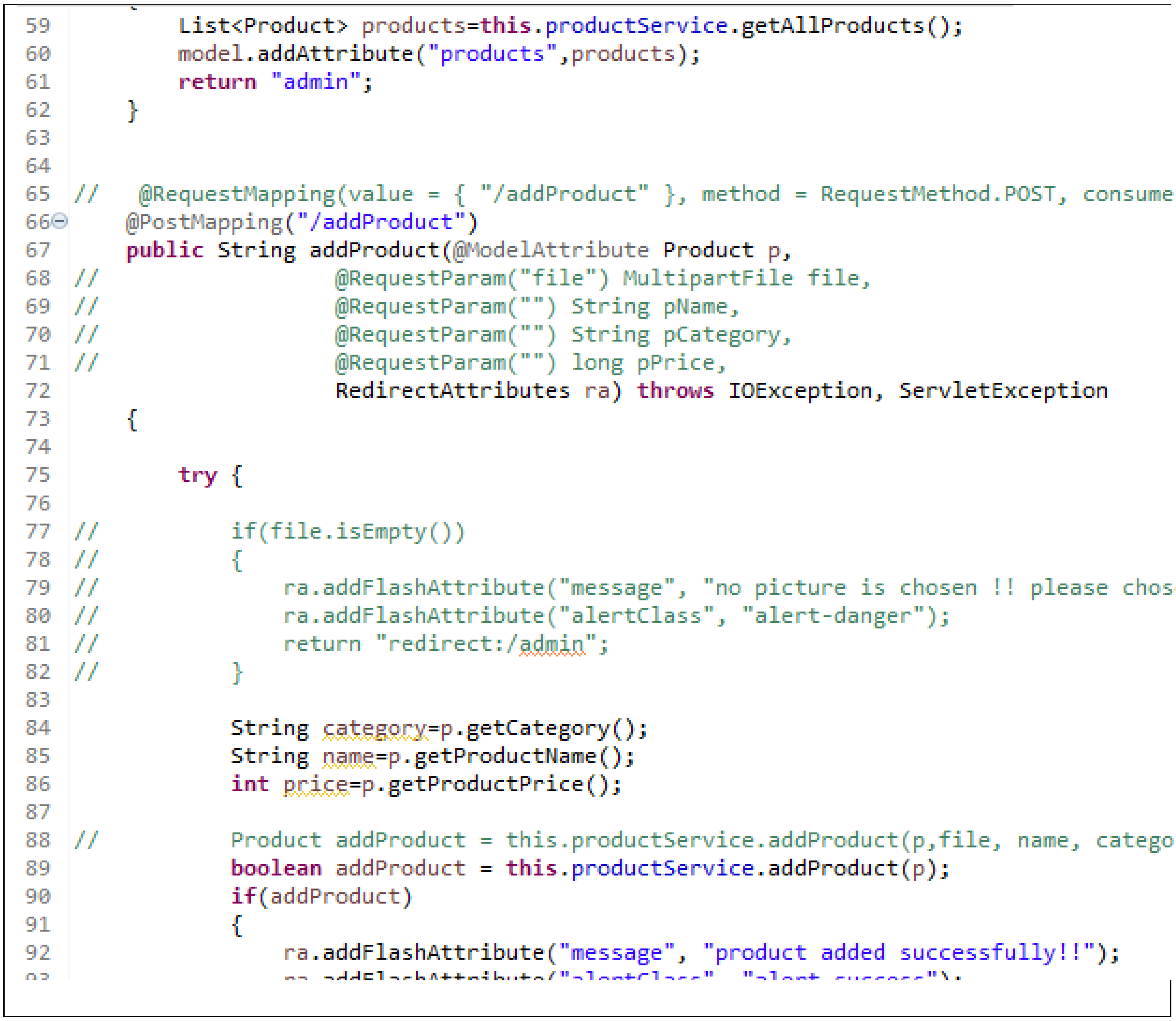


```
//              ra.addFlashAttribute("alertClass", "alert-danger");
//              return "redirect:/userNotFound";
//
//          }

        }
        System.out.println(user);
        ra.addFlashAttribute("message", "Login failed!!, please try again or signup
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/userNotFound";
    }

    @GetMapping("/userNotFound")
    public String userNotFound()
    {
        return "userNotFound";
    }

    @GetMapping("/logoutUser")
    public String logoutUser(RedirectAttributes ra)
    {
        ra.addFlashAttribute("message", "User Logged out successfully!!");
        ra.addFlashAttribute("alertClass", "alert-success");
        return "redirect:/logout";
    }

    @GetMapping("/logout")
    public String logoutController(HttpServletRequest req)
    {
        req.getSession().removeAttribute("username");
        return "logout";
    }

}
```

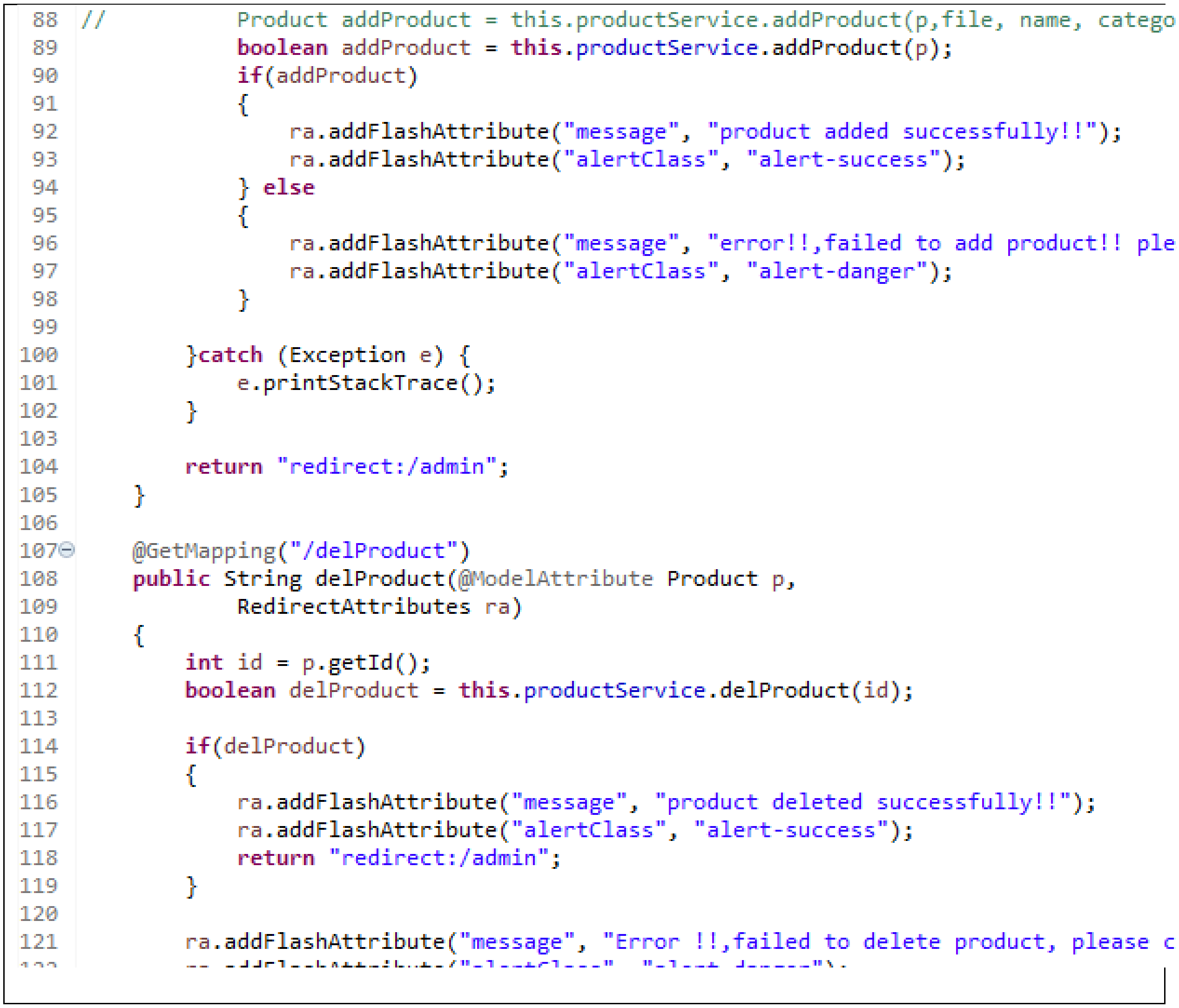
ProductController.java



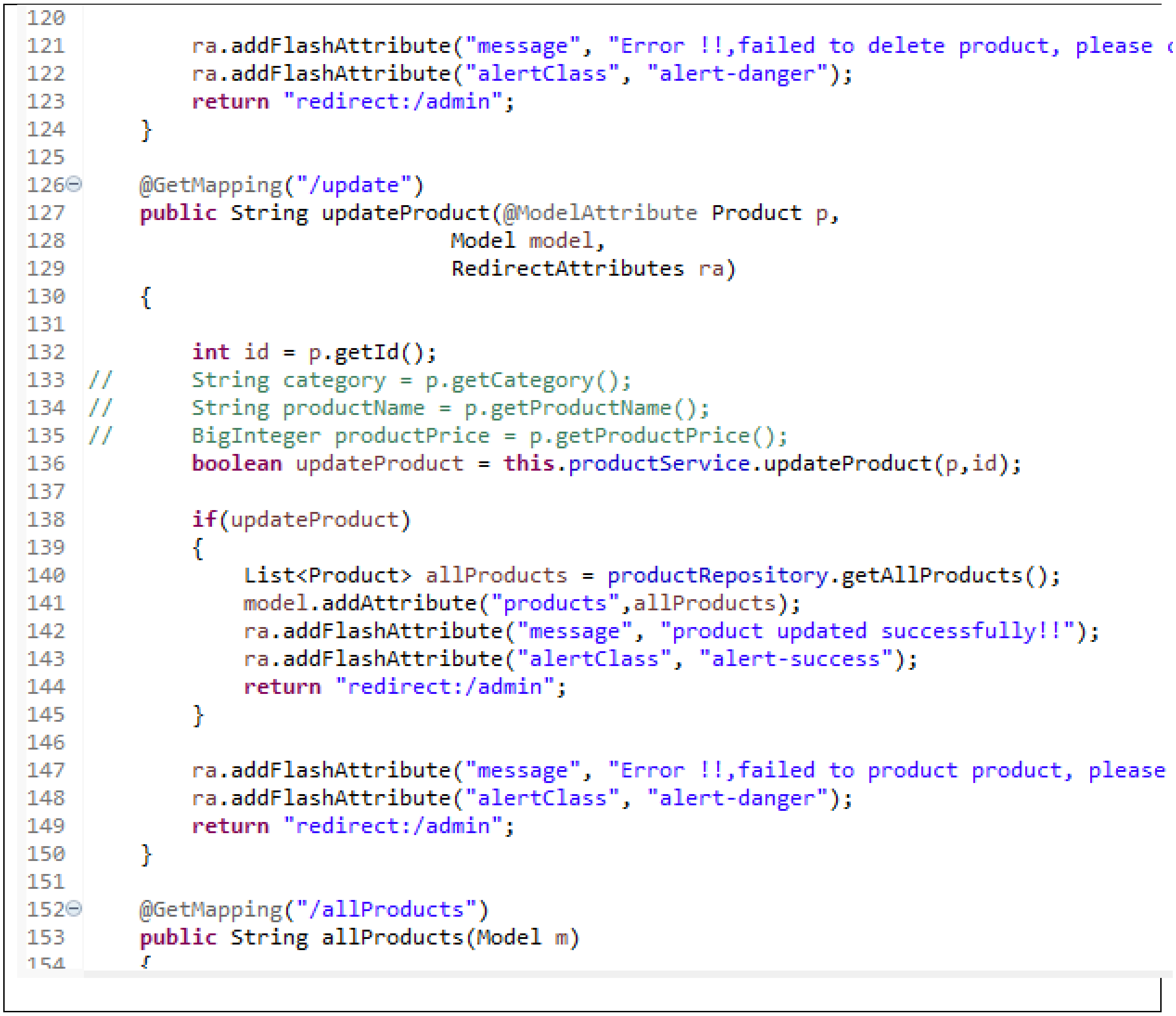
```java
package com.sportyShoes.controller;

import java.io.IOException;

//@MultipartConfig
@Controller
public class ProductController {

    @Autowired
    private ProductService productService;

    @Autowired
    private ProductRepository productRepository;

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private FileUploadHelper fileUploadHelper;


    public final String UPLOAD_DIR=new ClassPathResource("static/image").getFile().ge

    public ProductController() throws IOException
    {

    }

    @GetMapping("/getProducts")
    public String getProducts(Model model)
    {
        List<Product> products=this.productService.getAllProducts();
        model.addAttribute("products",products);
        return "admin";
    }
```



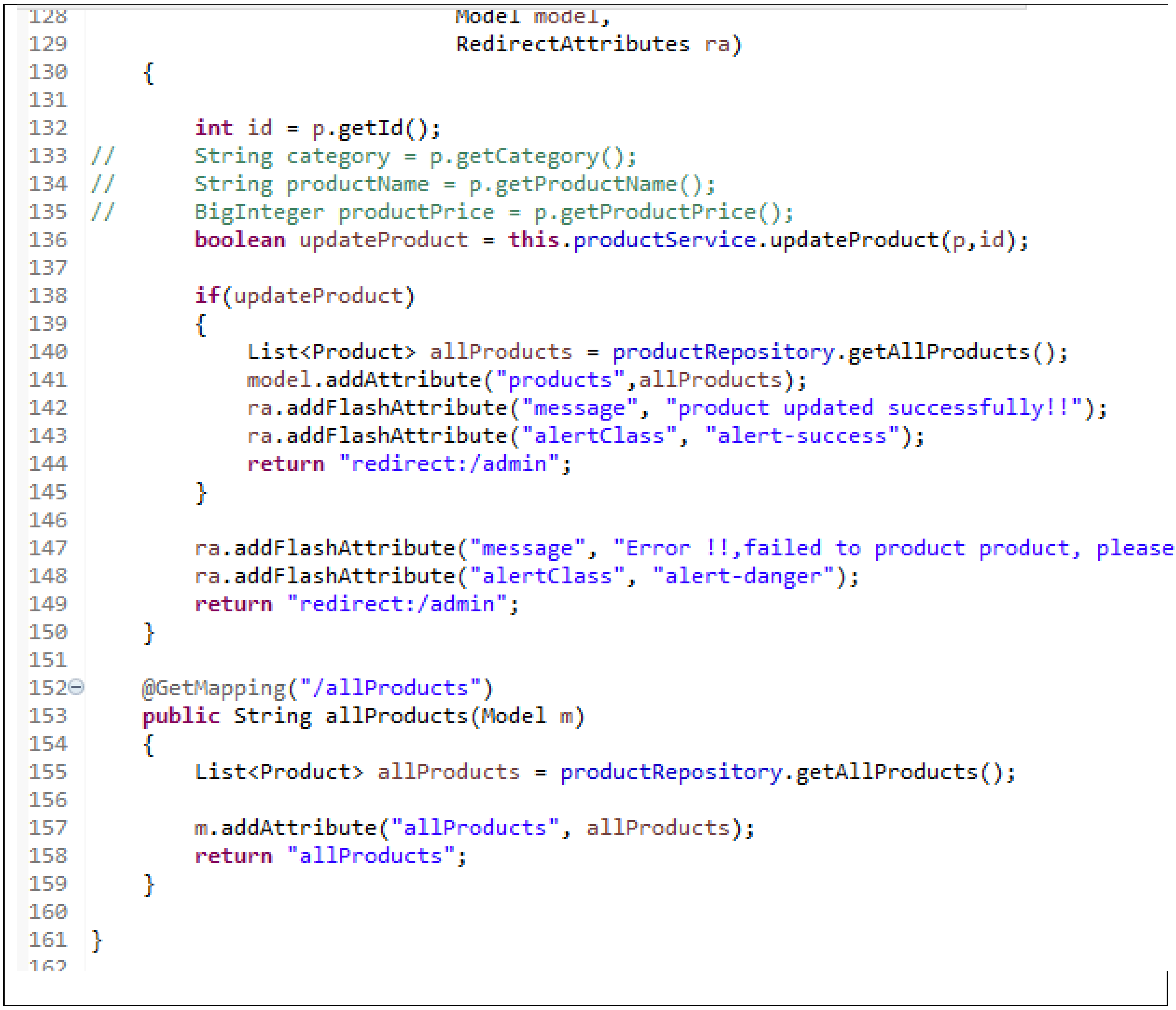
```
        List<Product> products=this.productService.getAllProducts();
        model.addAttribute("products",products);
        return "admin";
    }


//   @RequestMapping(value = { "/addProduct" }, method = RequestMethod.POST, consumes
    @PostMapping("/addProduct")
    public String addProduct(@ModelAttribute Product p,
//                  @RequestParam("file") MultipartFile file,
//                  @RequestParam("") String pName,
//                  @RequestParam("") String pCategory,
//                  @RequestParam("") long pPrice,
                    RedirectAttributes ra) throws IOException, ServletException
    {

        try {

//          if(file.isEmpty())
//          {
//              ra.addFlashAttribute("message", "no picture is chosen !! please chose
//              ra.addFlashAttribute("alertClass", "alert-danger");
//              return "redirect:/admin";
//          }

            String category=p.getCategory();
            String name=p.getProductName();
            int price=p.getProductPrice();

//          Product addProduct = this.productService.addProduct(p,file, name, categor
            boolean addProduct = this.productService.addProduct(p);
            if(addProduct)
            {
                ra.addFlashAttribute("message", "product added successfully!!");
                ra.addFlashAttribute("alertClass", "alert-success");
```



```
//          Product addProduct = this.productService.addProduct(p,file, name, catego
            boolean addProduct = this.productService.addProduct(p);
            if(addProduct)
            {
                ra.addFlashAttribute("message", "product added successfully!!");
                ra.addFlashAttribute("alertClass", "alert-success");
            } else
            {
                ra.addFlashAttribute("message", "error!!,failed to add product!! ple
                ra.addFlashAttribute("alertClass", "alert-danger");
            }

        }catch (Exception e) {
            e.printStackTrace();
        }

        return "redirect:/admin";
    }

    @GetMapping("/delProduct")
    public String delProduct(@ModelAttribute Product p,
            RedirectAttributes ra)
    {
        int id = p.getId();
        boolean delProduct = this.productService.delProduct(id);

        if(delProduct)
        {
            ra.addFlashAttribute("message", "product deleted successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/admin";
        }

        ra.addFlashAttribute("message", "Error !!,failed to delete product, please c
        ra.addFlashAttribute("alertClass", "alert-danger");
```

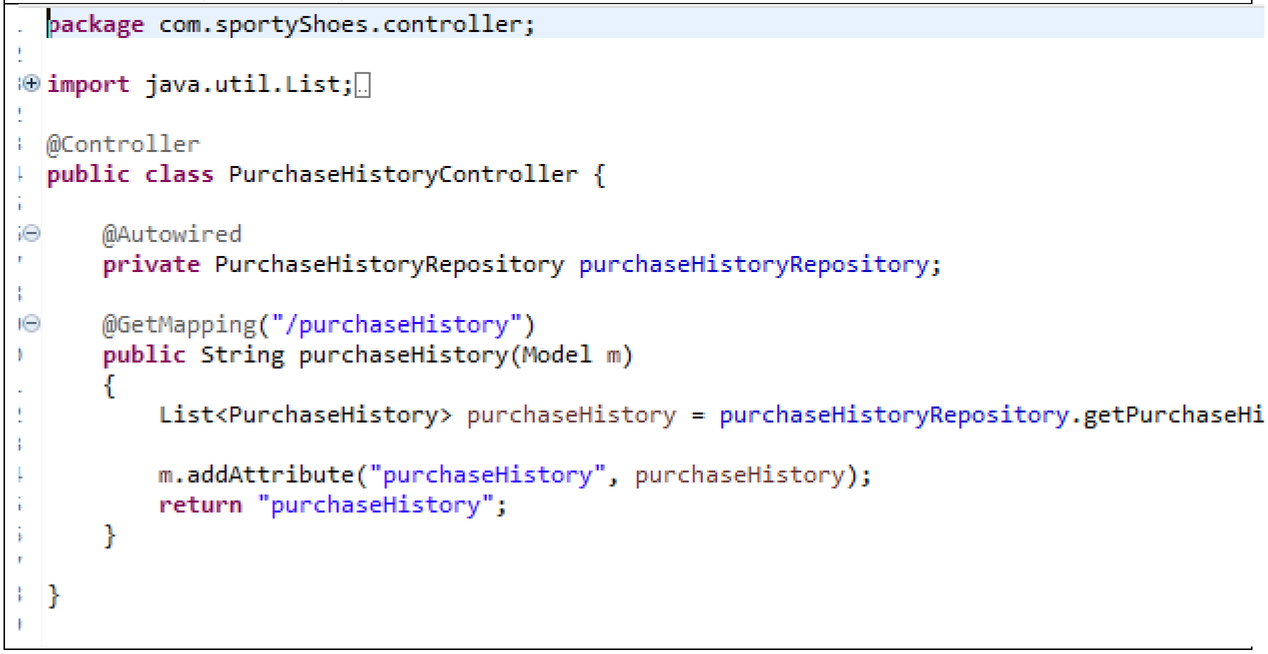


```
        ra.addFlashAttribute("message", "Error !!,failed to delete product, please 
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/admin";
    }

    @GetMapping("/update")
    public String updateProduct(@ModelAttribute Product p,
                                Model model,
                                RedirectAttributes ra)
    {

        int id = p.getId();
//      String category = p.getCategory();
//      String productName = p.getProductName();
//      BigInteger productPrice = p.getProductPrice();
        boolean updateProduct = this.productService.updateProduct(p,id);

        if(updateProduct)
        {
            List<Product> allProducts = productRepository.getAllProducts();
            model.addAttribute("products",allProducts);
            ra.addFlashAttribute("message", "product updated successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/admin";
        }

        ra.addFlashAttribute("message", "Error !!,failed to product product, please 
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/admin";
    }

    @GetMapping("/allProducts")
    public String allProducts(Model m)
    {
```



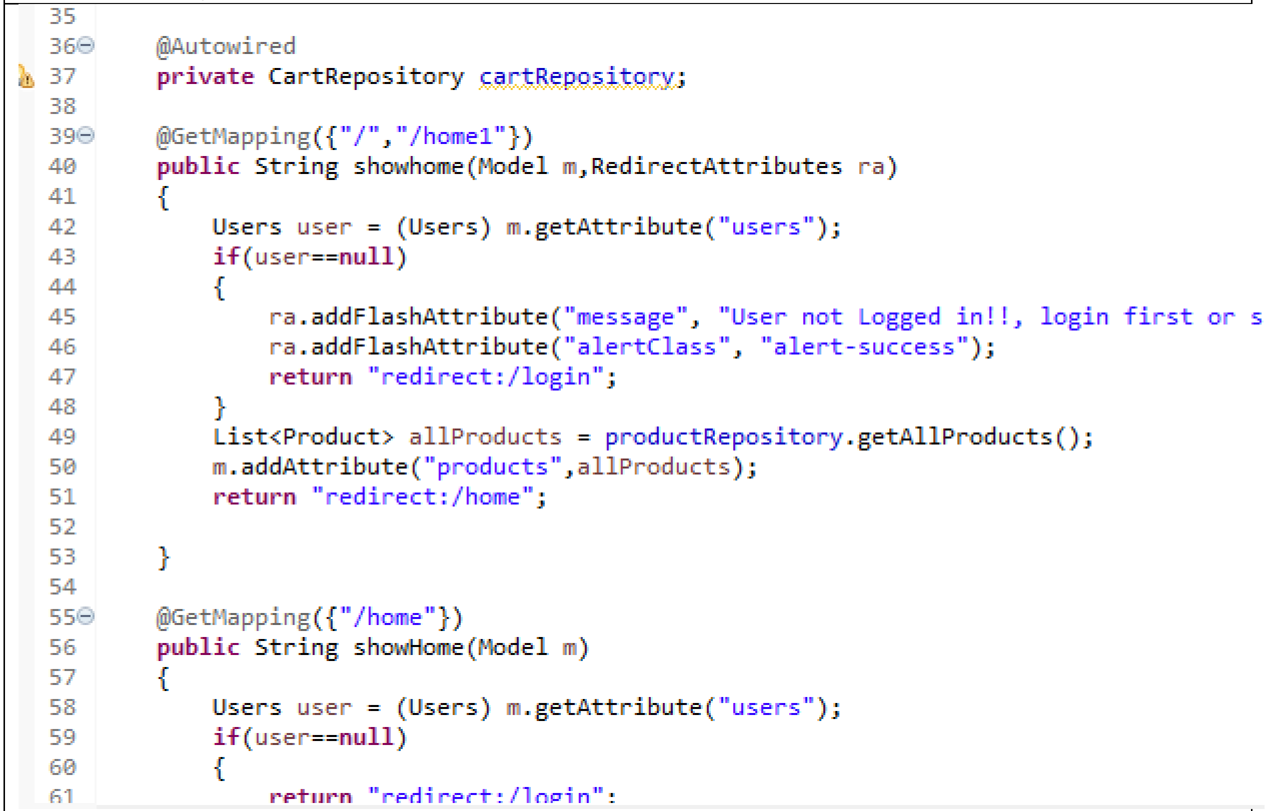
```
                            Model model,
                            RedirectAttributes ra)
    {

        int id = p.getId();
//      String category = p.getCategory();
//      String productName = p.getProductName();
//      BigInteger productPrice = p.getProductPrice();
        boolean updateProduct = this.productService.updateProduct(p,id);

        if(updateProduct)
        {
            List<Product> allProducts = productRepository.getAllProducts();
            model.addAttribute("products",allProducts);
            ra.addFlashAttribute("message", "product updated successfully!!");
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/admin";
        }

        ra.addFlashAttribute("message", "Error !!,failed to product product, please
        ra.addFlashAttribute("alertClass", "alert-danger");
        return "redirect:/admin";
    }

    @GetMapping("/allProducts")
    public String allProducts(Model m)
    {
        List<Product> allProducts = productRepository.getAllProducts();

        m.addAttribute("allProducts", allProducts);
        return "allProducts";
    }

}
```

PurchaseHistoryController.java

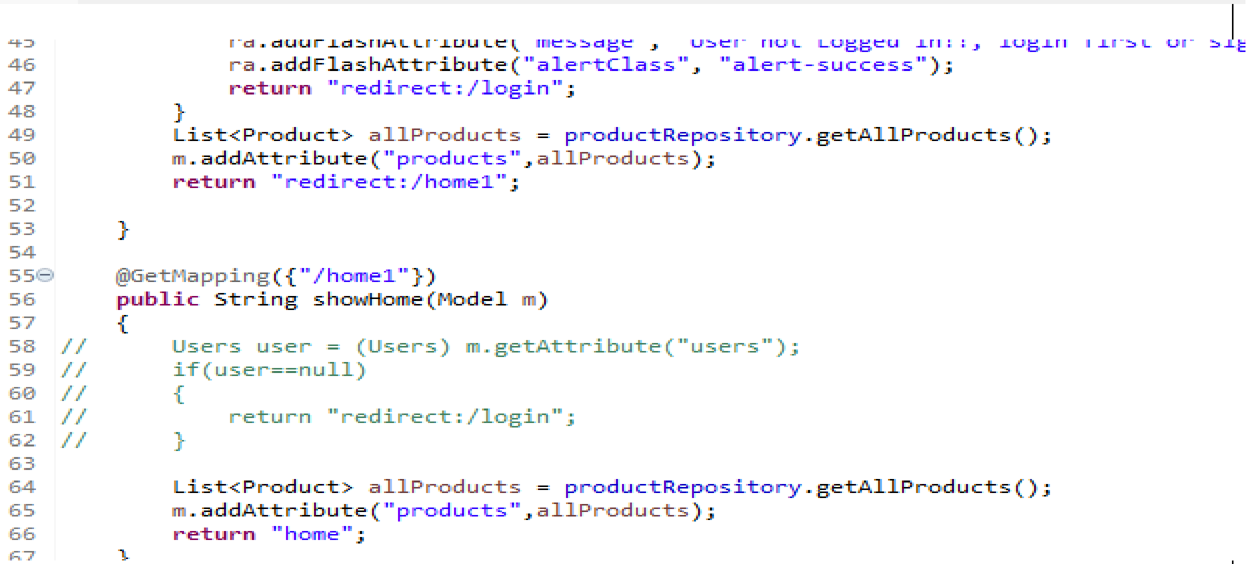


```java
package com.sportyShoes.controller;

import java.util.List;

@Controller
public class PurchaseHistoryController {

    @Autowired
    private PurchaseHistoryRepository purchaseHistoryRepository;

    @GetMapping("/purchaseHistory")
    public String purchaseHistory(Model m)
    {
        List<PurchaseHistory> purchaseHistory = purchaseHistoryRepository.getPurchaseHi

        m.addAttribute("purchaseHistory", purchaseHistory);
        return "purchaseHistory";
    }

}
```

UserController.java

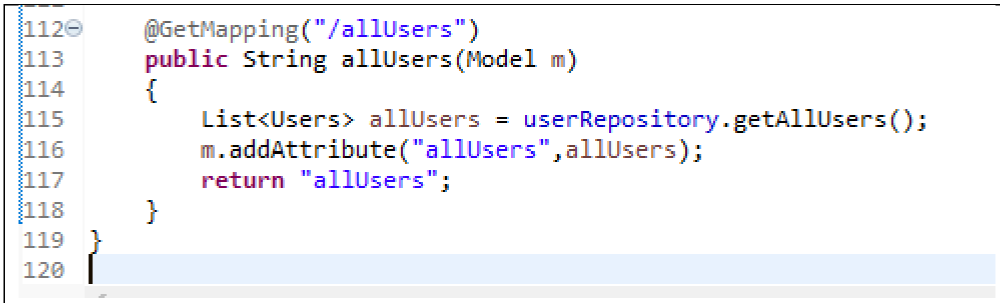


```java

    @Autowired
    private CartRepository cartRepository;

    @GetMapping({"/","/home1"})
    public String showhome(Model m,RedirectAttributes ra)
    {
        Users user = (Users) m.getAttribute("users");
        if(user==null)
        {
            ra.addFlashAttribute("message", "User not Logged in!!, login first or s
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/login";
        }
        List<Product> allProducts = productRepository.getAllProducts();
        m.addAttribute("products",allProducts);
        return "redirect:/home";

    }

    @GetMapping({"/home"})
    public String showHome(Model m)
    {
        Users user = (Users) m.getAttribute("users");
        if(user==null)
        {
            return "redirect:/login";
```



```java
            ra.addFlashAttribute("message", "User not Logged in!!, login first or sig
            ra.addFlashAttribute("alertClass", "alert-success");
            return "redirect:/login";
        }
        List<Product> allProducts = productRepository.getAllProducts();
        m.addAttribute("products",allProducts);
        return "redirect:/home1";

    }

    @GetMapping({"/home1"})
    public String showHome(Model m)
    {
//      Users user = (Users) m.getAttribute("users");
//      if(user==null)
//      {
//          return "redirect:/login";
//      }

        List<Product> allProducts = productRepository.getAllProducts();
        m.addAttribute("products",allProducts);
        return "home";
    }
```

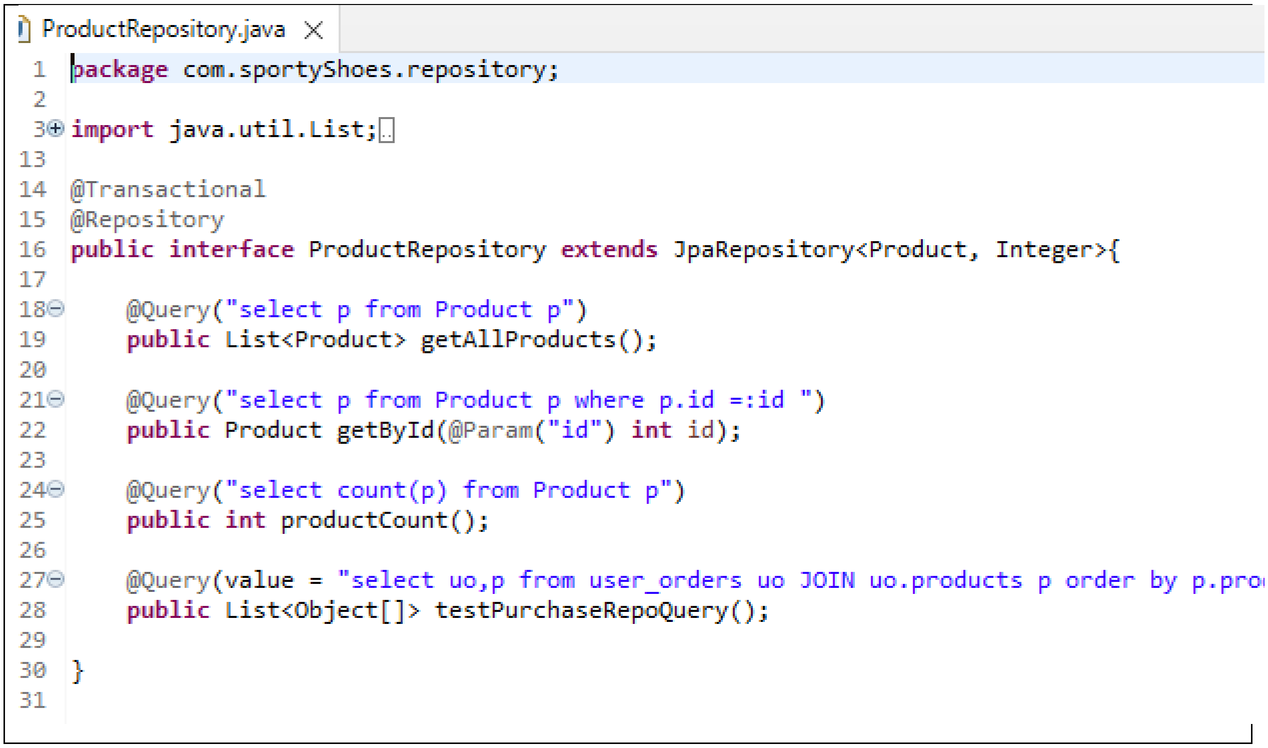
// In Between is a commented section



```
    @GetMapping("/allUsers")
    public String allUsers(Model m)
    {
        List<Users> allUsers = userRepository.getAllUsers();
        m.addAttribute("allUsers",allUsers);
        return "allUsers";
    }
}
```

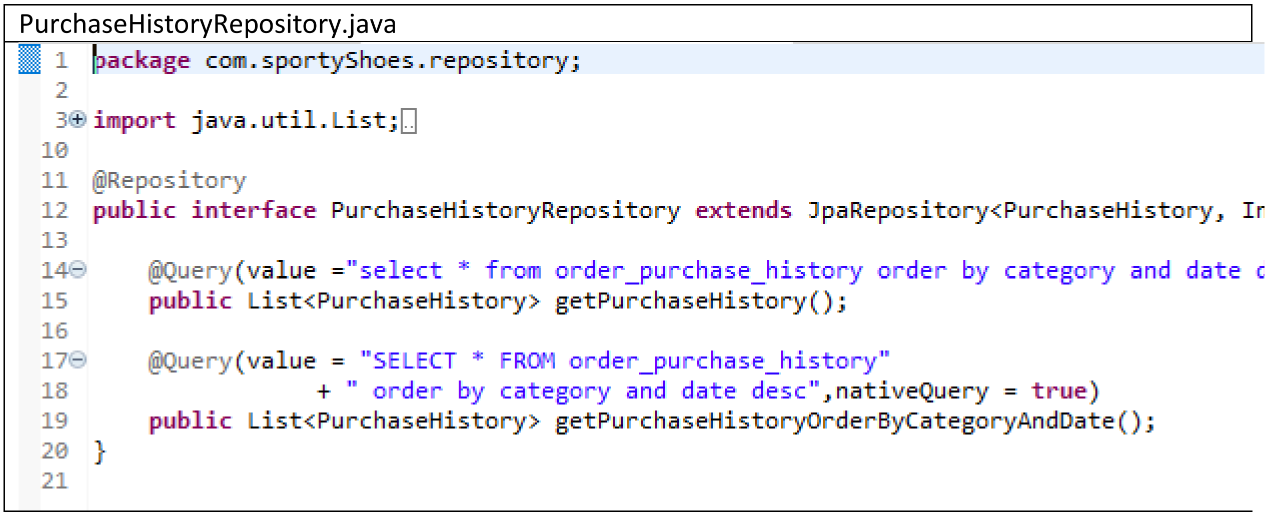
- ***REPOSITORY***

ProductRepository.java



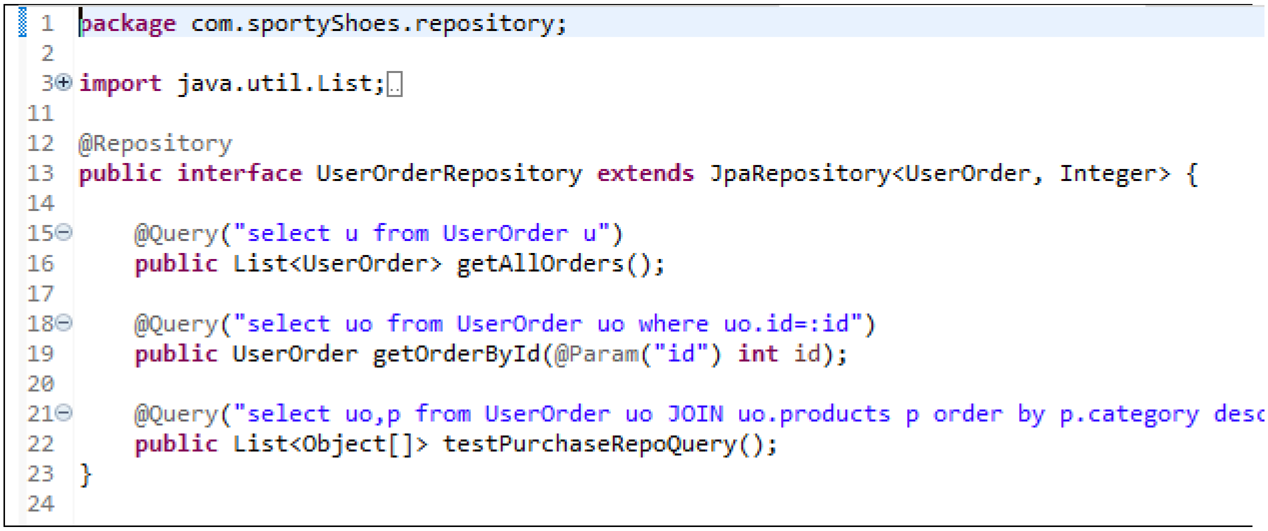
```
package com.sportyShoes.repository;

import java.util.List;

@Transactional
@Repository
public interface ProductRepository extends JpaRepository<Product, Integer>{

    @Query("select p from Product p")
    public List<Product> getAllProducts();

    @Query("select p from Product p where p.id =:id ")
    public Product getById(@Param("id") int id);

    @Query("select count(p) from Product p")
    public int productCount();

    @Query(value = "select uo,p from user_orders uo JOIN uo.products p order by p.pro
    public List<Object[]> testPurchaseRepoQuery();

}
```

PurchaseHistoryRepository.java



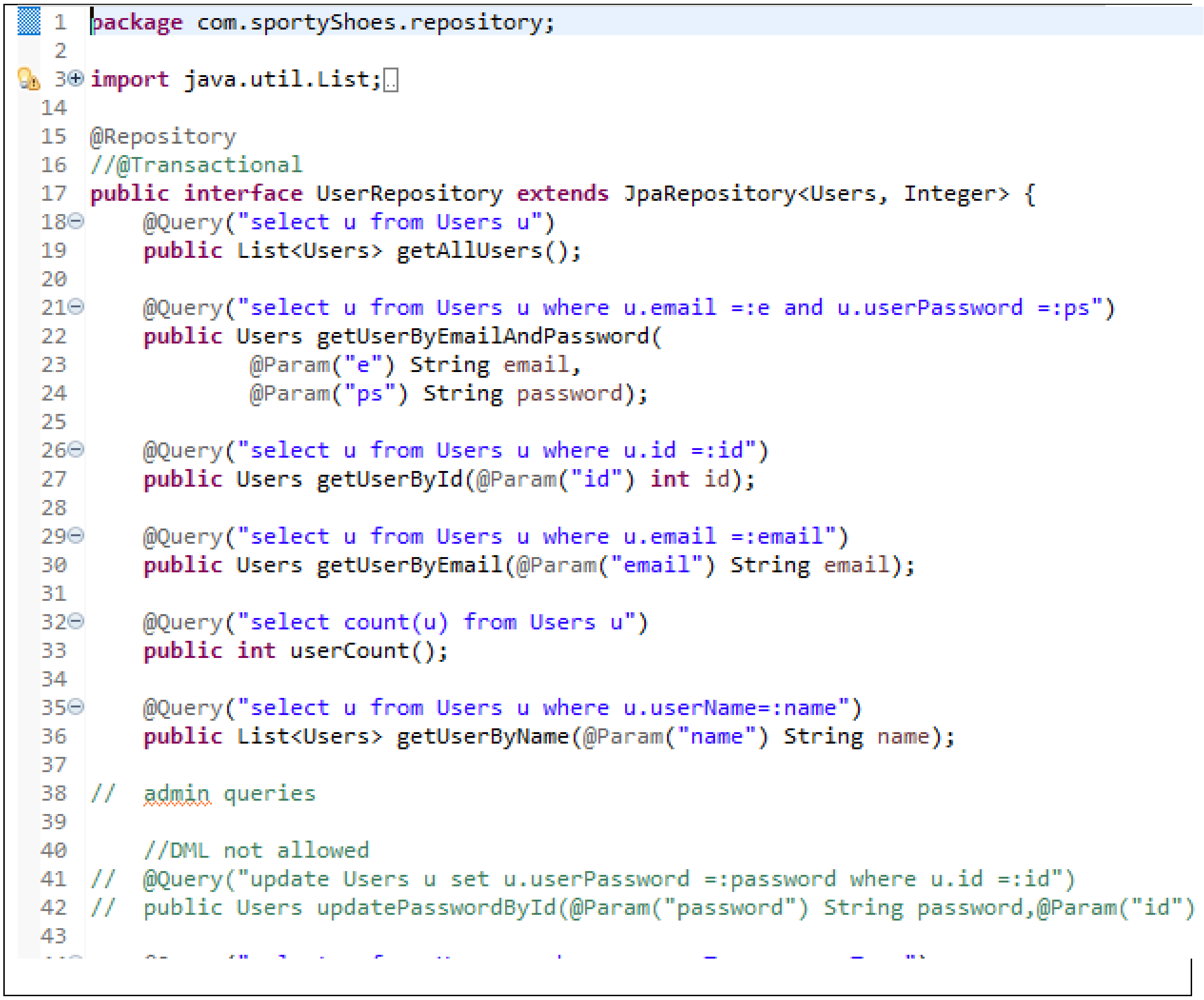
```
package com.sportyShoes.repository;

import java.util.List;

@Repository
public interface PurchaseHistoryRepository extends JpaRepository<PurchaseHistory, Ir

    @Query(value ="select * from order_purchase_history order by category and date d
    public List<PurchaseHistory> getPurchaseHistory();

    @Query(value = "SELECT * FROM order_purchase_history"
                + " order by category and date desc",nativeQuery = true)
    public List<PurchaseHistory> getPurchaseHistoryOrderByCategoryAndDate();
}
```

UserOrderRepository.java

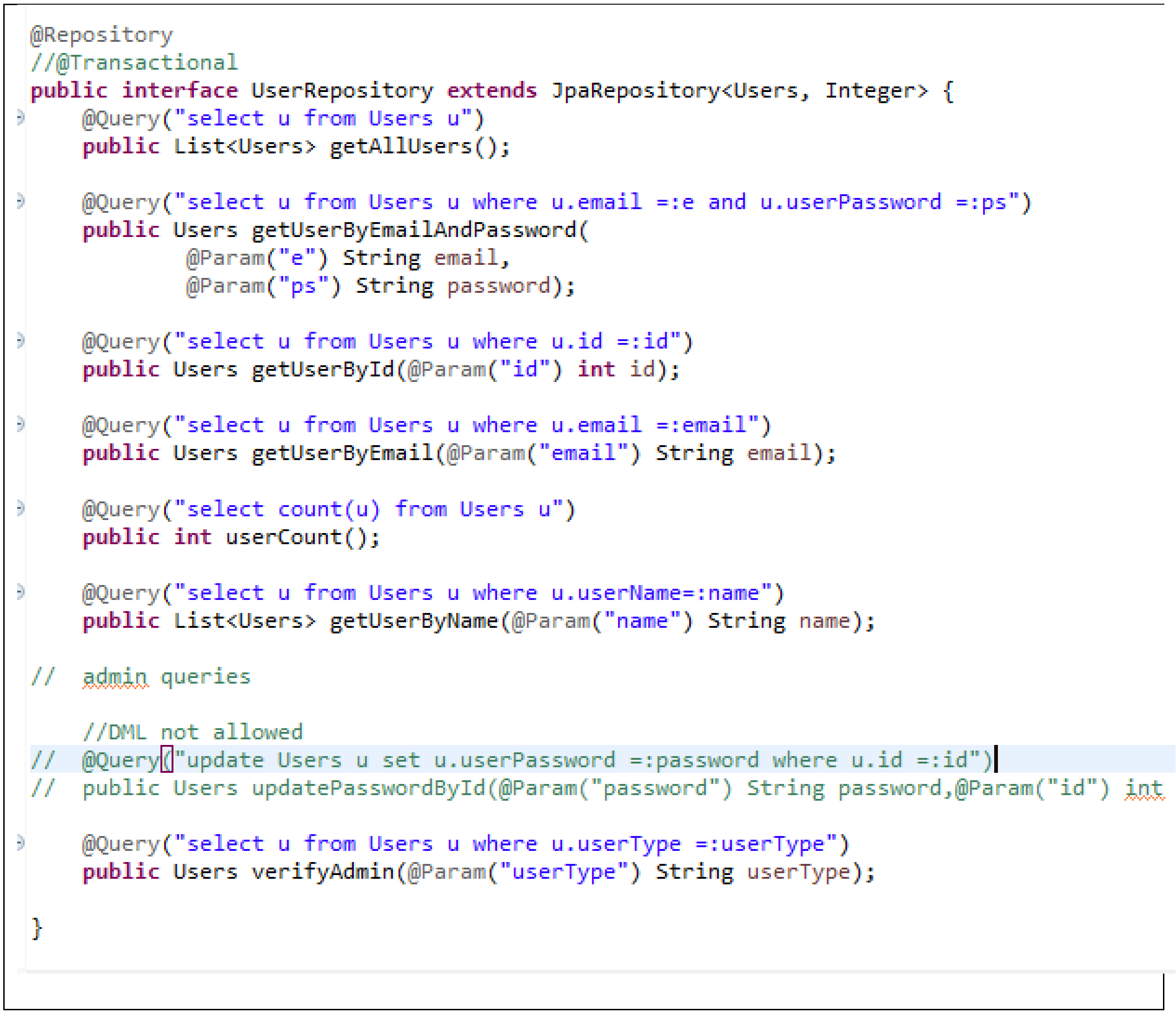


```java
package com.sportyShoes.repository;

import java.util.List;

@Repository
public interface UserOrderRepository extends JpaRepository<UserOrder, Integer> {

    @Query("select u from UserOrder u")
    public List<UserOrder> getAllOrders();

    @Query("select uo from UserOrder uo where uo.id=:id")
    public UserOrder getOrderById(@Param("id") int id);

    @Query("select uo,p from UserOrder uo JOIN uo.products p order by p.category desc
    public List<Object[]> testPurchaseRepoQuery();
}

```

UserRepository.java



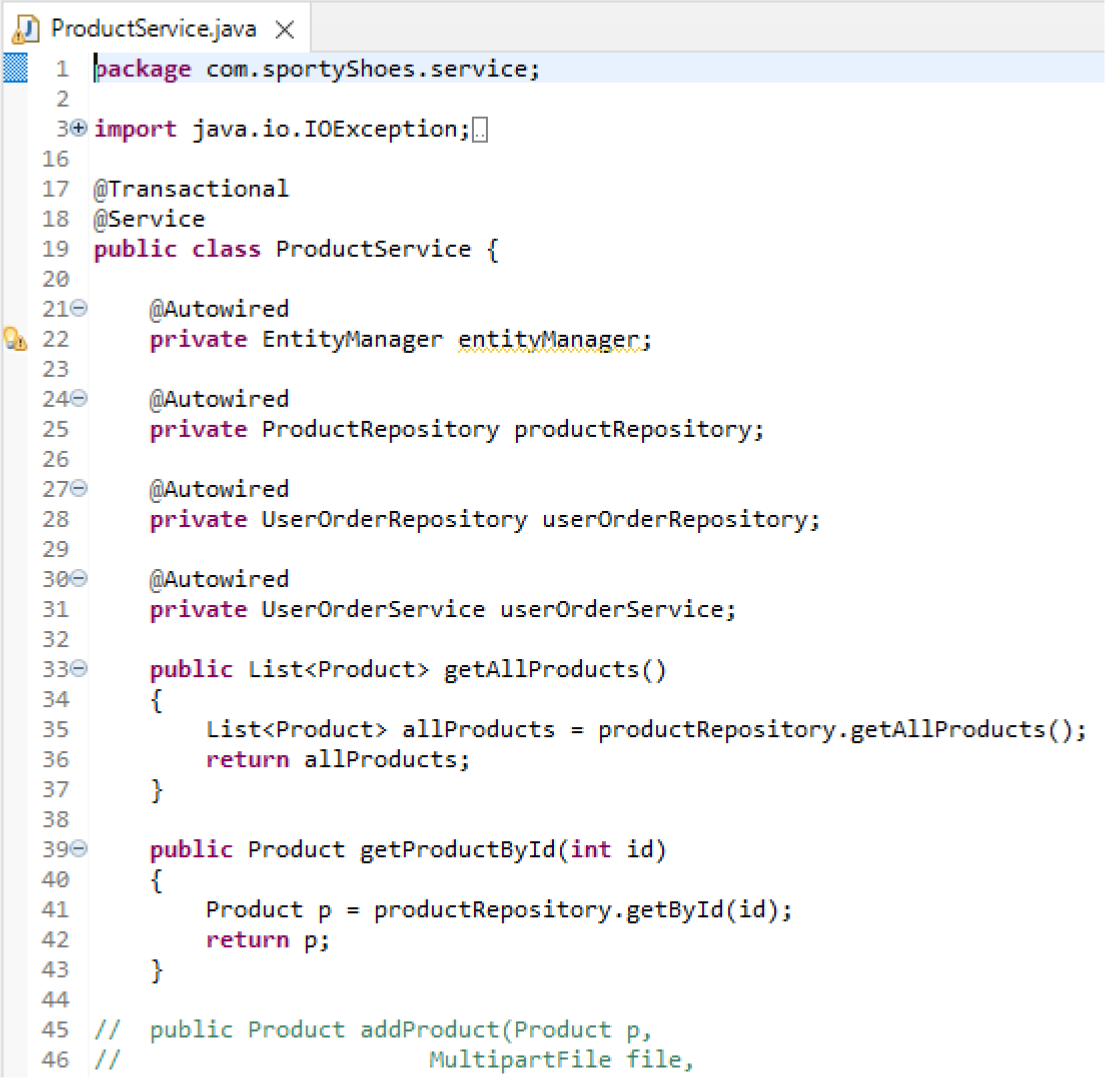
```java
package com.sportyShoes.repository;

import java.util.List;

@Repository
//@Transactional
public interface UserRepository extends JpaRepository<Users, Integer> {
    @Query("select u from Users u")
    public List<Users> getAllUsers();

    @Query("select u from Users u where u.email =:e and u.userPassword =:ps")
    public Users getUserByEmailAndPassword(
            @Param("e") String email,
            @Param("ps") String password);

    @Query("select u from Users u where u.id =:id")
    public Users getUserById(@Param("id") int id);

    @Query("select u from Users u where u.email =:email")
    public Users getUserByEmail(@Param("email") String email);

    @Query("select count(u) from Users u")
    public int userCount();

    @Query("select u from Users u where u.userName=:name")
    public List<Users> getUserByName(@Param("name") String name);

//  admin queries

    //DML not allowed
//  @Query("update Users u set u.userPassword =:password where u.id =:id")
//  public Users updatePasswordById(@Param("password") String password,@Param("id")

```



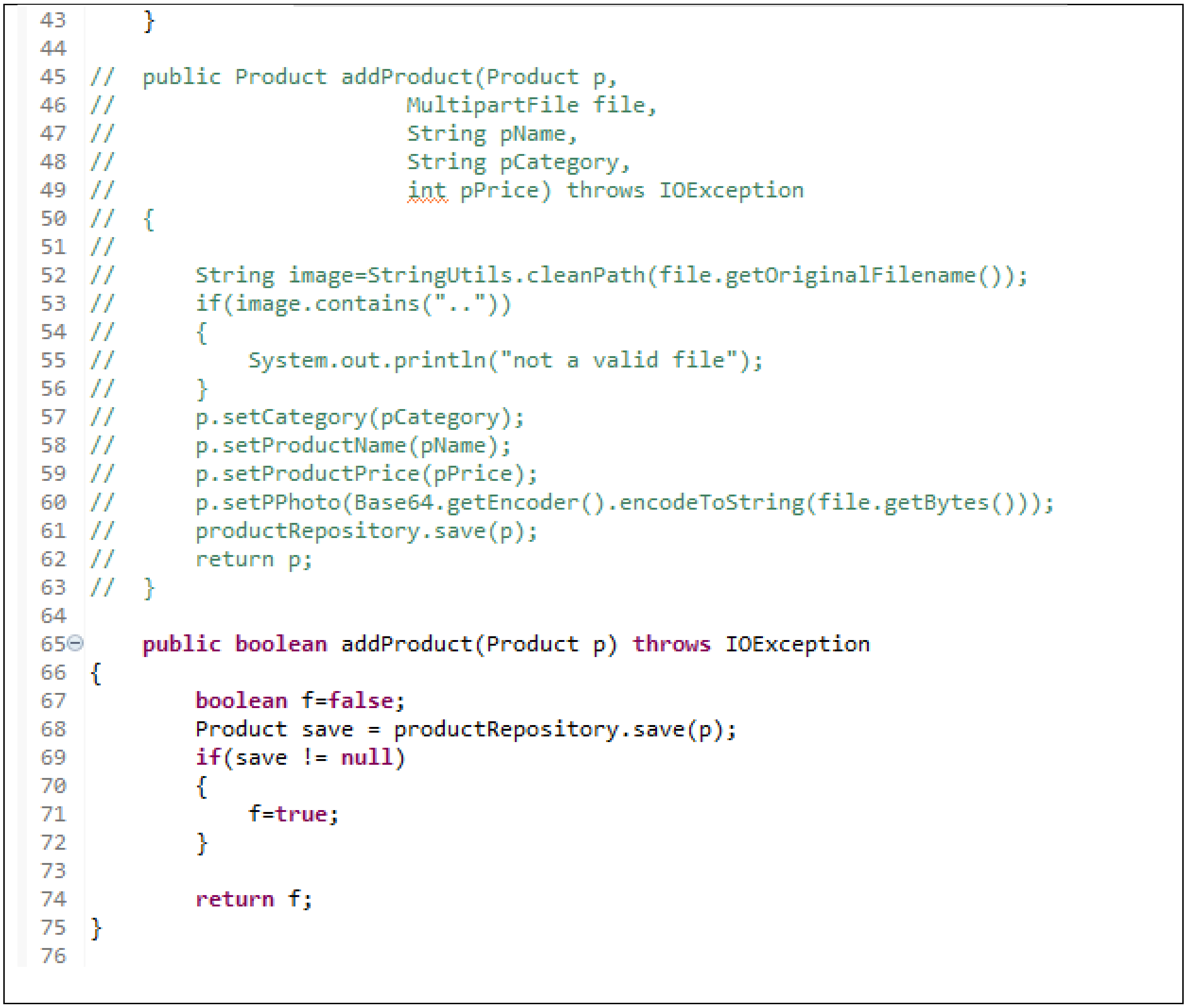
```
@Repository
//@Transactional
public interface UserRepository extends JpaRepository<Users, Integer> {
    @Query("select u from Users u")
    public List<Users> getAllUsers();

    @Query("select u from Users u where u.email =:e and u.userPassword =:ps")
    public Users getUserByEmailAndPassword(
            @Param("e") String email,
            @Param("ps") String password);

    @Query("select u from Users u where u.id =:id")
    public Users getUserById(@Param("id") int id);

    @Query("select u from Users u where u.email =:email")
    public Users getUserByEmail(@Param("email") String email);

    @Query("select count(u) from Users u")
    public int userCount();

    @Query("select u from Users u where u.userName=:name")
    public List<Users> getUserByName(@Param("name") String name);

//  admin queries

    //DML not allowed
//  @Query("update Users u set u.userPassword =:password where u.id =:id")
//  public Users updatePasswordById(@Param("password") String password,@Param("id") int

    @Query("select u from Users u where u.userType =:userType")
    public Users verifyAdmin(@Param("userType") String userType);

}
```

- ***SERVICE***

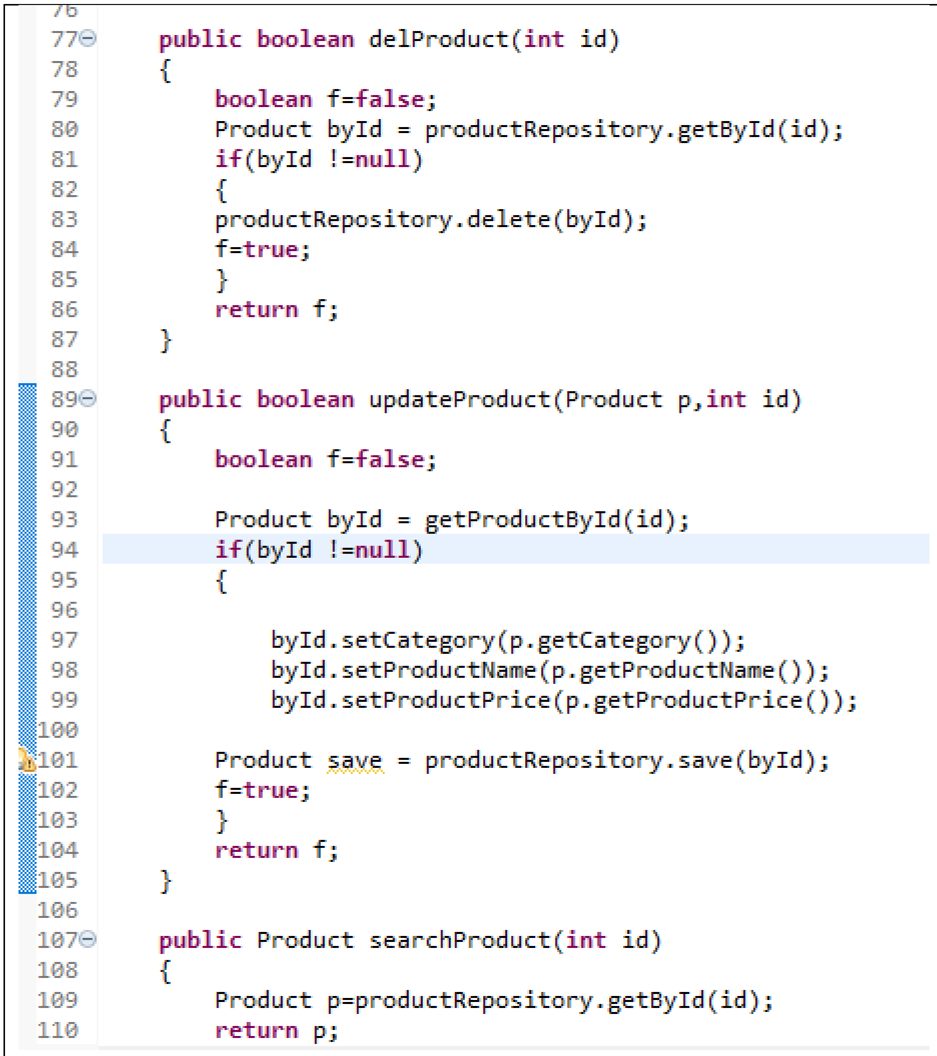
ProductService.java



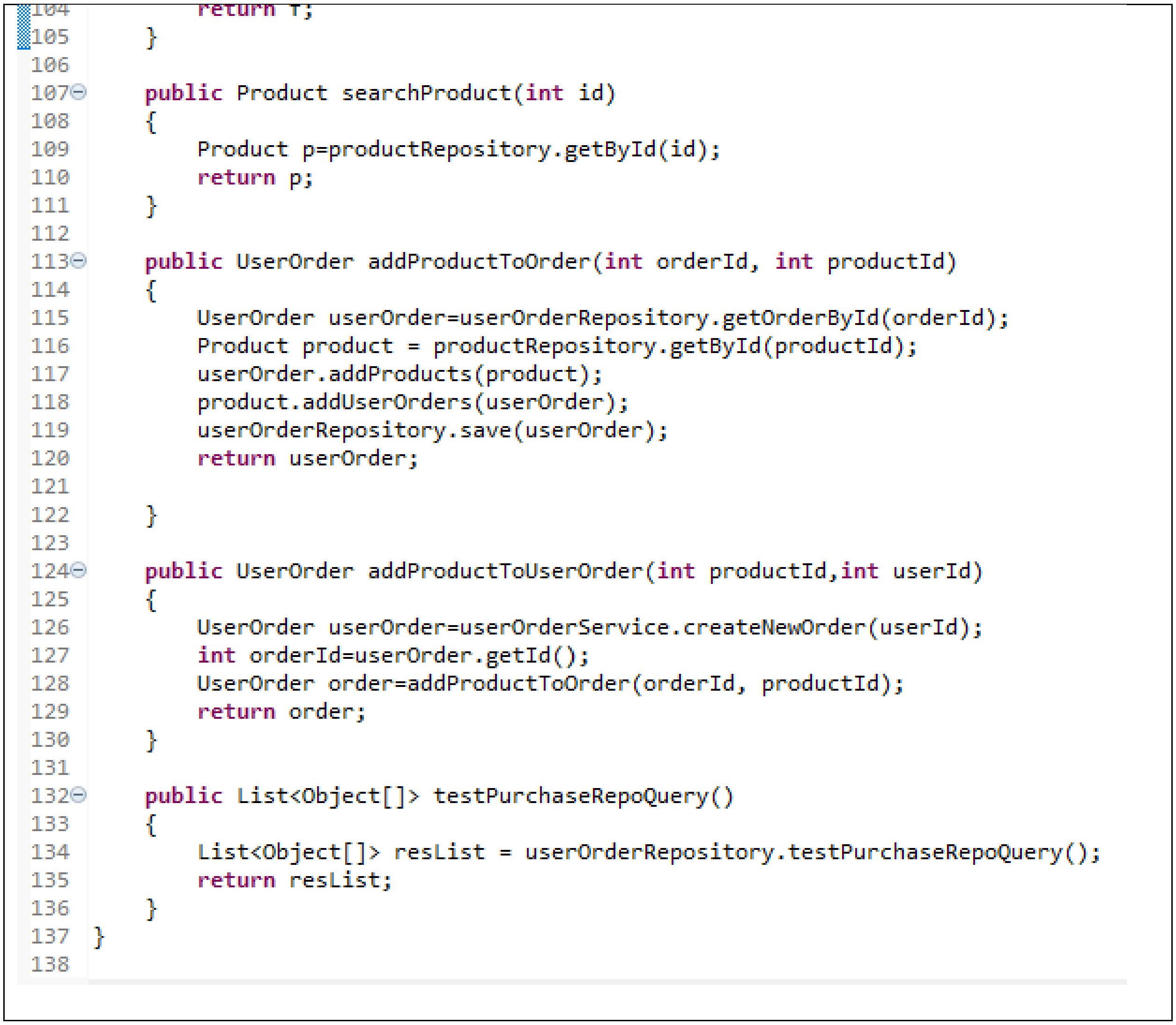
```java
package com.sportyShoes.service;

import java.io.IOException;

@Transactional
@Service
public class ProductService {

    @Autowired
    private EntityManager entityManager;

    @Autowired
    private ProductRepository productRepository;

    @Autowired
    private UserOrderRepository userOrderRepository;

    @Autowired
    private UserOrderService userOrderService;

    public List<Product> getAllProducts()
    {
        List<Product> allProducts = productRepository.getAllProducts();
        return allProducts;
    }

    public Product getProductById(int id)
    {
        Product p = productRepository.getById(id);
        return p;
    }

//  public Product addProduct(Product p,
//                        MultipartFile file,
```



```
    }

//  public Product addProduct(Product p,
//                      MultipartFile file,
//                      String pName,
//                      String pCategory,
//                      int pPrice) throws IOException
//  {
//
//      String image=StringUtils.cleanPath(file.getOriginalFilename());
//      if(image.contains(".."))
//      {
//          System.out.println("not a valid file");
//      }
//      p.setCategory(pCategory);
//      p.setProductName(pName);
//      p.setProductPrice(pPrice);
//      p.setPPhoto(Base64.getEncoder().encodeToString(file.getBytes()));
//      productRepository.save(p);
//      return p;
//  }

    public boolean addProduct(Product p) throws IOException
{
        boolean f=false;
        Product save = productRepository.save(p);
        if(save != null)
        {
            f=true;
        }

        return f;
}

```

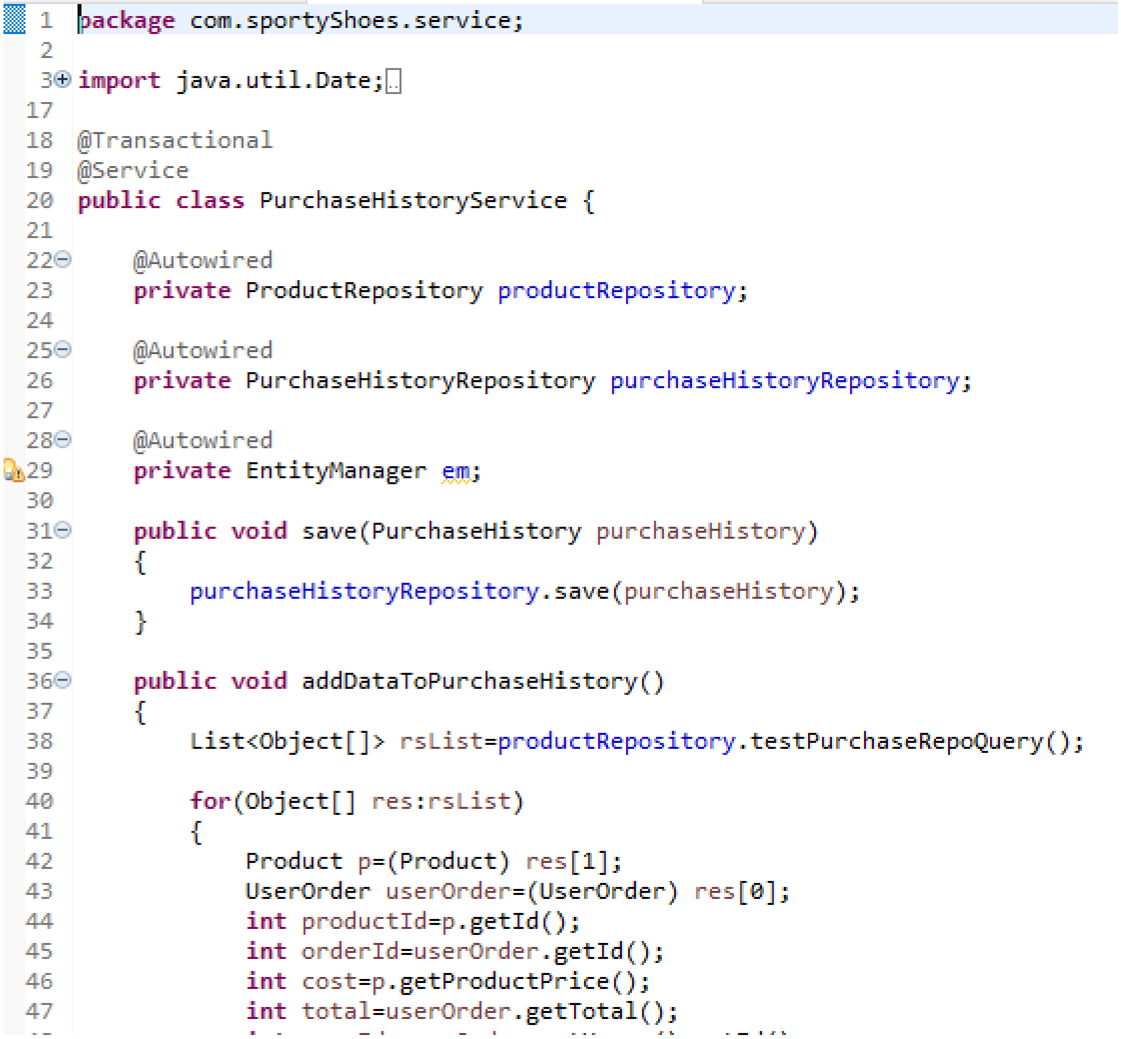


```
    public boolean delProduct(int id)
    {
        boolean f=false;
        Product byId = productRepository.getById(id);
        if(byId !=null)
        {
        productRepository.delete(byId);
        f=true;
        }
        return f;
    }

    public boolean updateProduct(Product p,int id)
    {
        boolean f=false;

        Product byId = getProductById(id);
        if(byId !=null)
        {

            byId.setCategory(p.getCategory());
            byId.setProductName(p.getProductName());
            byId.setProductPrice(p.getProductPrice());

        Product save = productRepository.save(byId);
        f=true;
        }
        return f;
    }

    public Product searchProduct(int id)
    {
        Product p=productRepository.getById(id);
        return p;
```

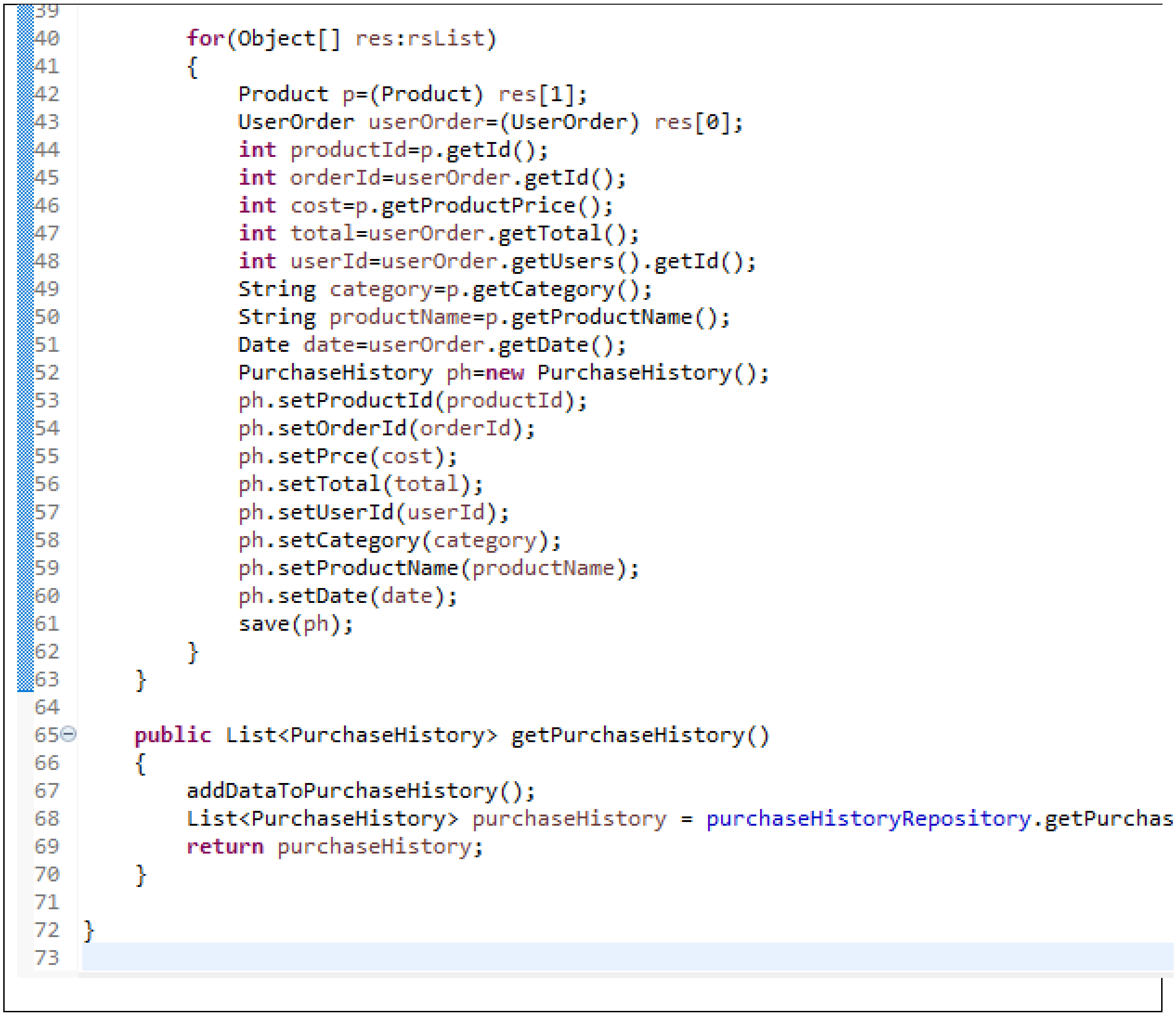


```java
        return f;
    }

    public Product searchProduct(int id)
    {
        Product p=productRepository.getById(id);
        return p;
    }

    public UserOrder addProductToOrder(int orderId, int productId)
    {
        UserOrder userOrder=userOrderRepository.getOrderById(orderId);
        Product product = productRepository.getById(productId);
        userOrder.addProducts(product);
        product.addUserOrders(userOrder);
        userOrderRepository.save(userOrder);
        return userOrder;

    }

    public UserOrder addProductToUserOrder(int productId,int userId)
    {
        UserOrder userOrder=userOrderService.createNewOrder(userId);
        int orderId=userOrder.getId();
        UserOrder order=addProductToOrder(orderId, productId);
        return order;
    }

    public List<Object[]> testPurchaseRepoQuery()
    {
        List<Object[]> resList = userOrderRepository.testPurchaseRepoQuery();
        return resList;
    }
}
```

PurchaseHistoryService.java

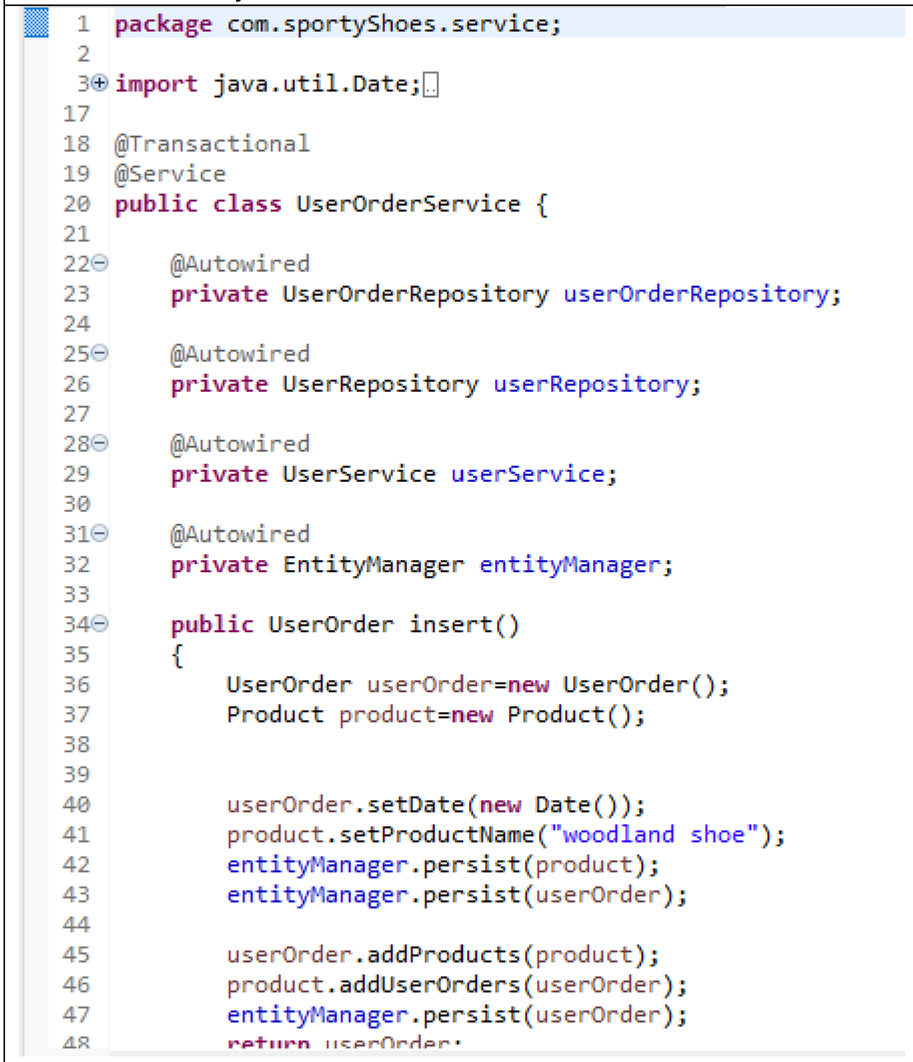


```java
package com.sportyShoes.service;

import java.util.Date;

@Transactional
@Service
public class PurchaseHistoryService {

    @Autowired
    private ProductRepository productRepository;

    @Autowired
    private PurchaseHistoryRepository purchaseHistoryRepository;

    @Autowired
    private EntityManager em;

    public void save(PurchaseHistory purchaseHistory)
    {
        purchaseHistoryRepository.save(purchaseHistory);
    }

    public void addDataToPurchaseHistory()
    {
        List<Object[]> rsList=productRepository.testPurchaseRepoQuery();

        for(Object[] res:rsList)
        {
            Product p=(Product) res[1];
            UserOrder userOrder=(UserOrder) res[0];
            int productId=p.getId();
            int orderId=userOrder.getId();
            int cost=p.getProductPrice();
            int total=userOrder.getTotal();
```

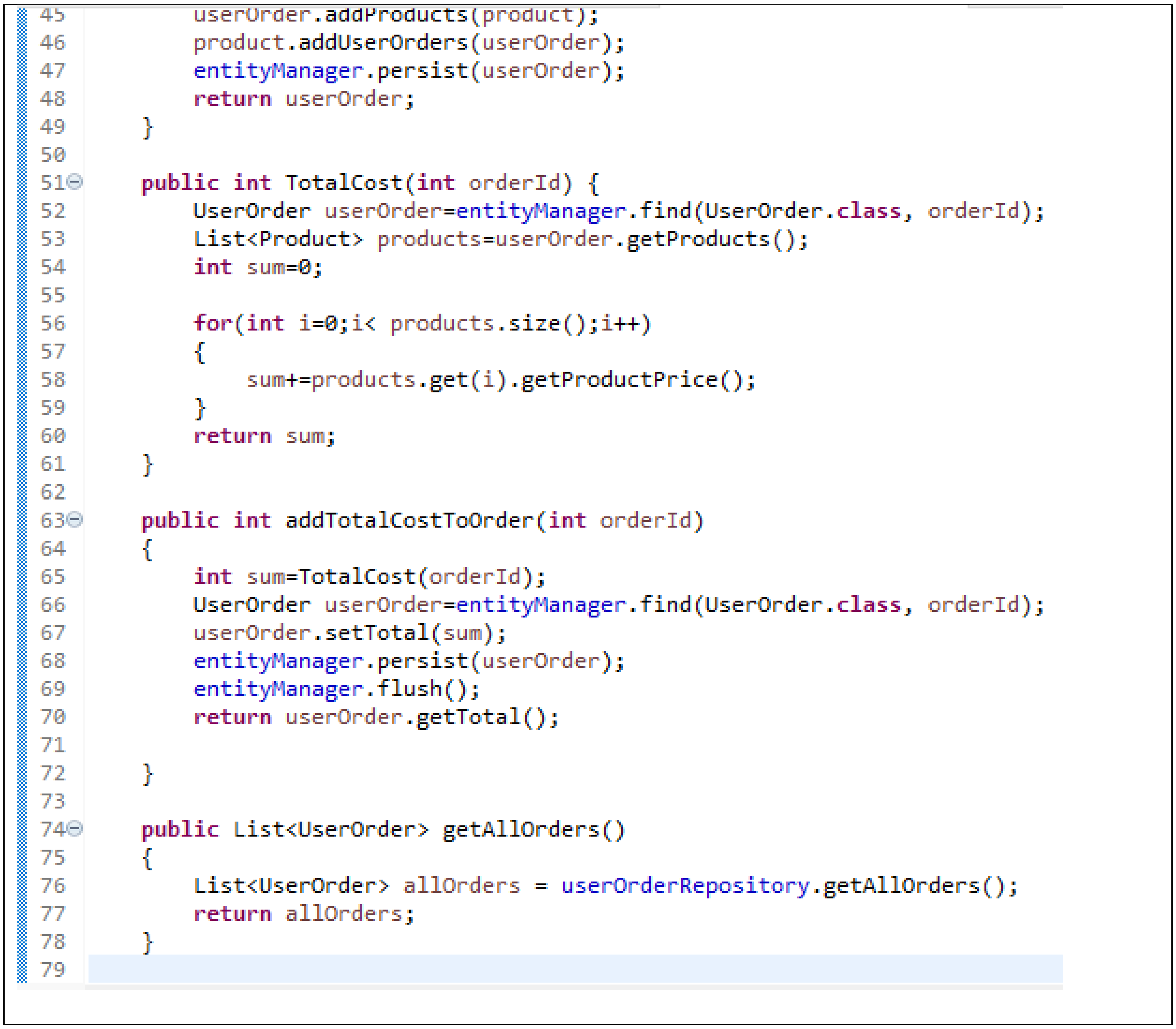


```java
        for(Object[] res:rsList)
        {
            Product p=(Product) res[1];
            UserOrder userOrder=(UserOrder) res[0];
            int productId=p.getId();
            int orderId=userOrder.getId();
            int cost=p.getProductPrice();
            int total=userOrder.getTotal();
            int userId=userOrder.getUsers().getId();
            String category=p.getCategory();
            String productName=p.getProductName();
            Date date=userOrder.getDate();
            PurchaseHistory ph=new PurchaseHistory();
            ph.setProductId(productId);
            ph.setOrderId(orderId);
            ph.setPrce(cost);
            ph.setTotal(total);
            ph.setUserId(userId);
            ph.setCategory(category);
            ph.setProductName(productName);
            ph.setDate(date);
            save(ph);
        }
    }

    public List<PurchaseHistory> getPurchaseHistory()
    {
        addDataToPurchaseHistory();
        List<PurchaseHistory> purchaseHistory = purchaseHistoryRepository.getPurchas
        return purchaseHistory;
    }

}
```

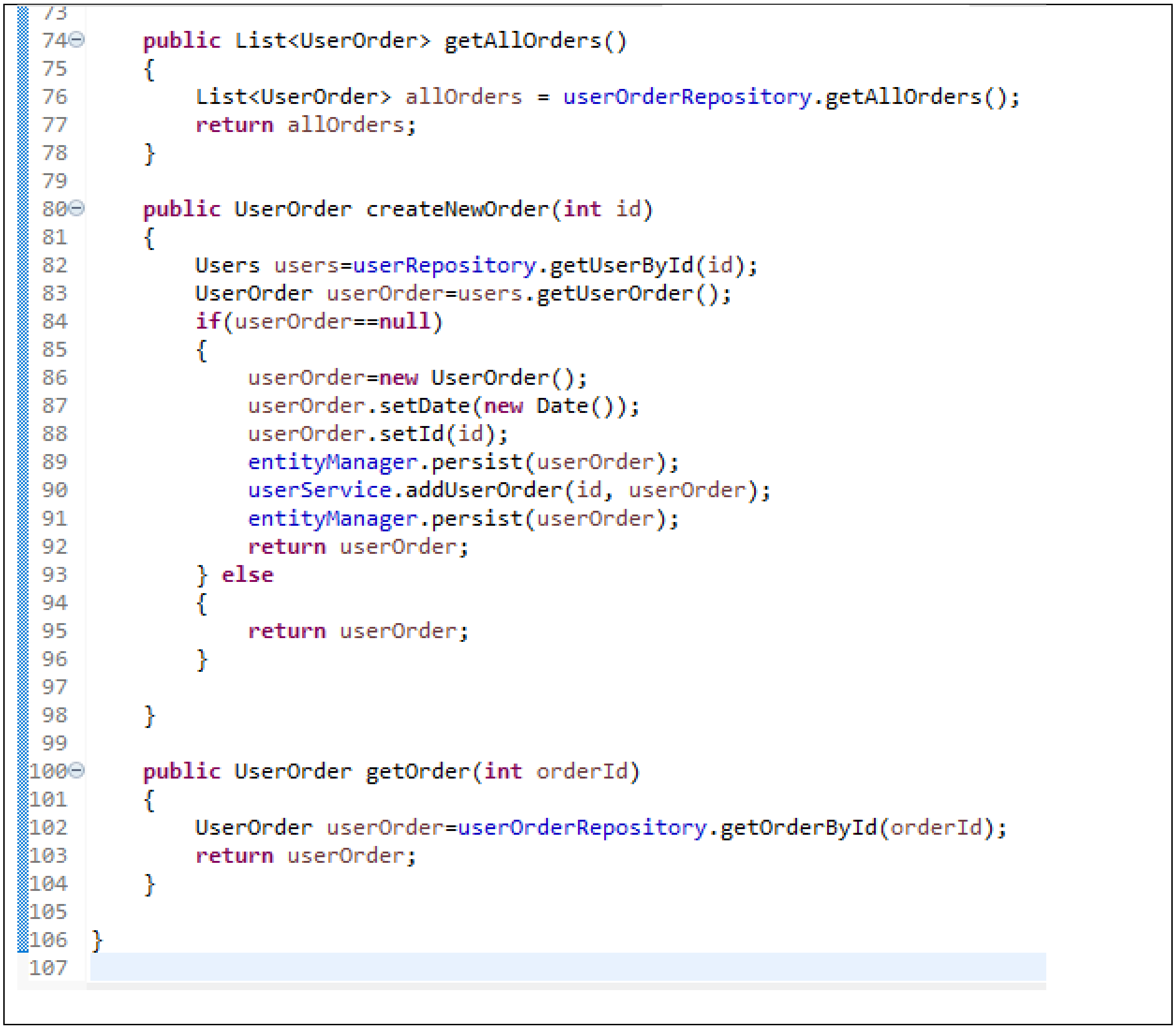
UserOrderService.java



```java
package com.sportyShoes.service;

import java.util.Date;

@Transactional
@Service
public class UserOrderService {

    @Autowired
    private UserOrderRepository userOrderRepository;

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private UserService userService;

    @Autowired
    private EntityManager entityManager;

    public UserOrder insert()
    {
        UserOrder userOrder=new UserOrder();
        Product product=new Product();


        userOrder.setDate(new Date());
        product.setProductName("woodland shoe");
        entityManager.persist(product);
        entityManager.persist(userOrder);

        userOrder.addProducts(product);
        product.addUserOrders(userOrder);
        entityManager.persist(userOrder);
        return userOrder;
```

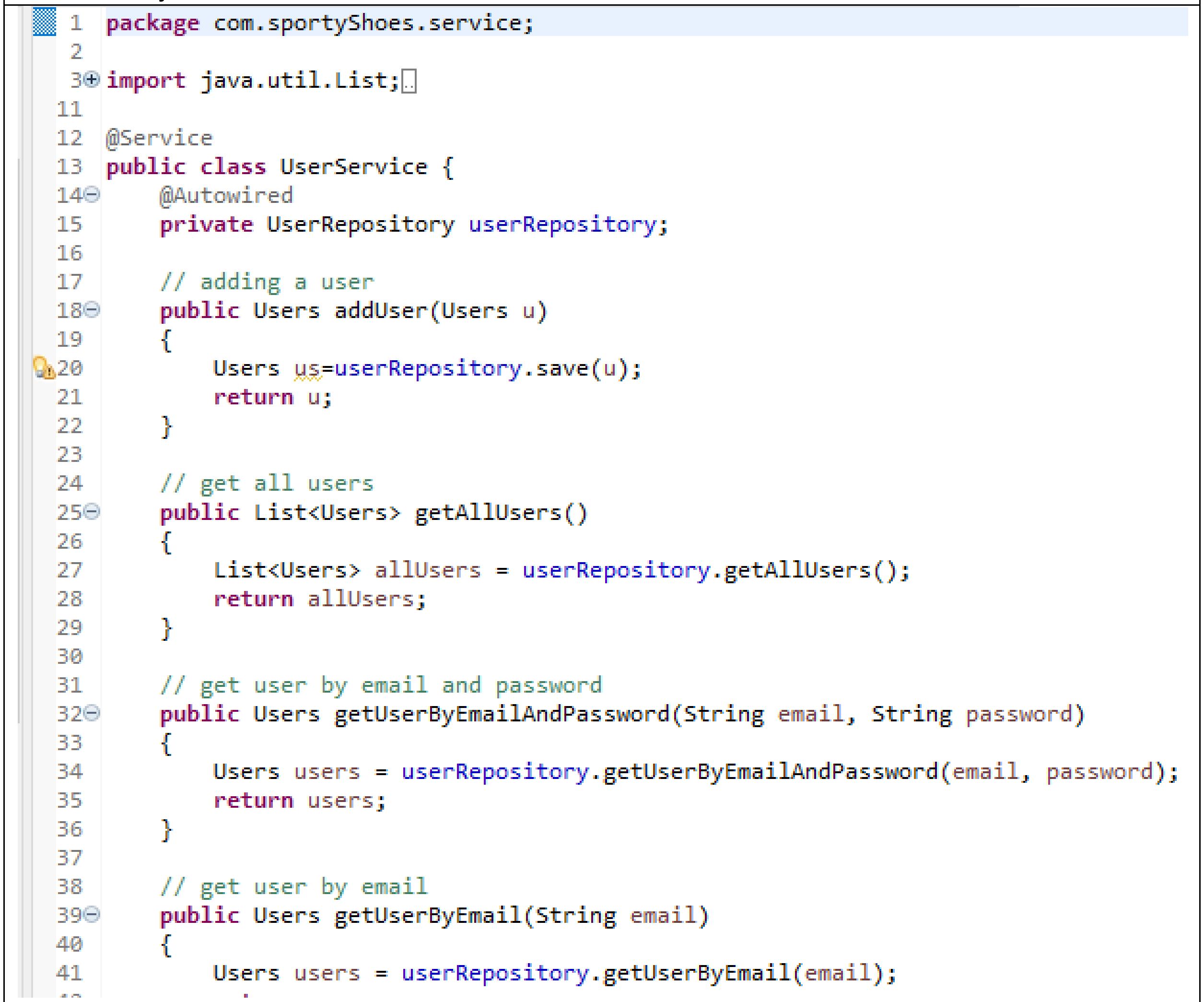


```
        userOrder.addProducts(product);
        product.addUserOrders(userOrder);
        entityManager.persist(userOrder);
        return userOrder;
    }

    public int TotalCost(int orderId) {
        UserOrder userOrder=entityManager.find(UserOrder.class, orderId);
        List<Product> products=userOrder.getProducts();
        int sum=0;

        for(int i=0;i< products.size();i++)
        {
            sum+=products.get(i).getProductPrice();
        }
        return sum;
    }

    public int addTotalCostToOrder(int orderId)
    {
        int sum=TotalCost(orderId);
        UserOrder userOrder=entityManager.find(UserOrder.class, orderId);
        userOrder.setTotal(sum);
        entityManager.persist(userOrder);
        entityManager.flush();
        return userOrder.getTotal();

    }

    public List<UserOrder> getAllOrders()
    {
        List<UserOrder> allOrders = userOrderRepository.getAllOrders();
        return allOrders;
    }

```

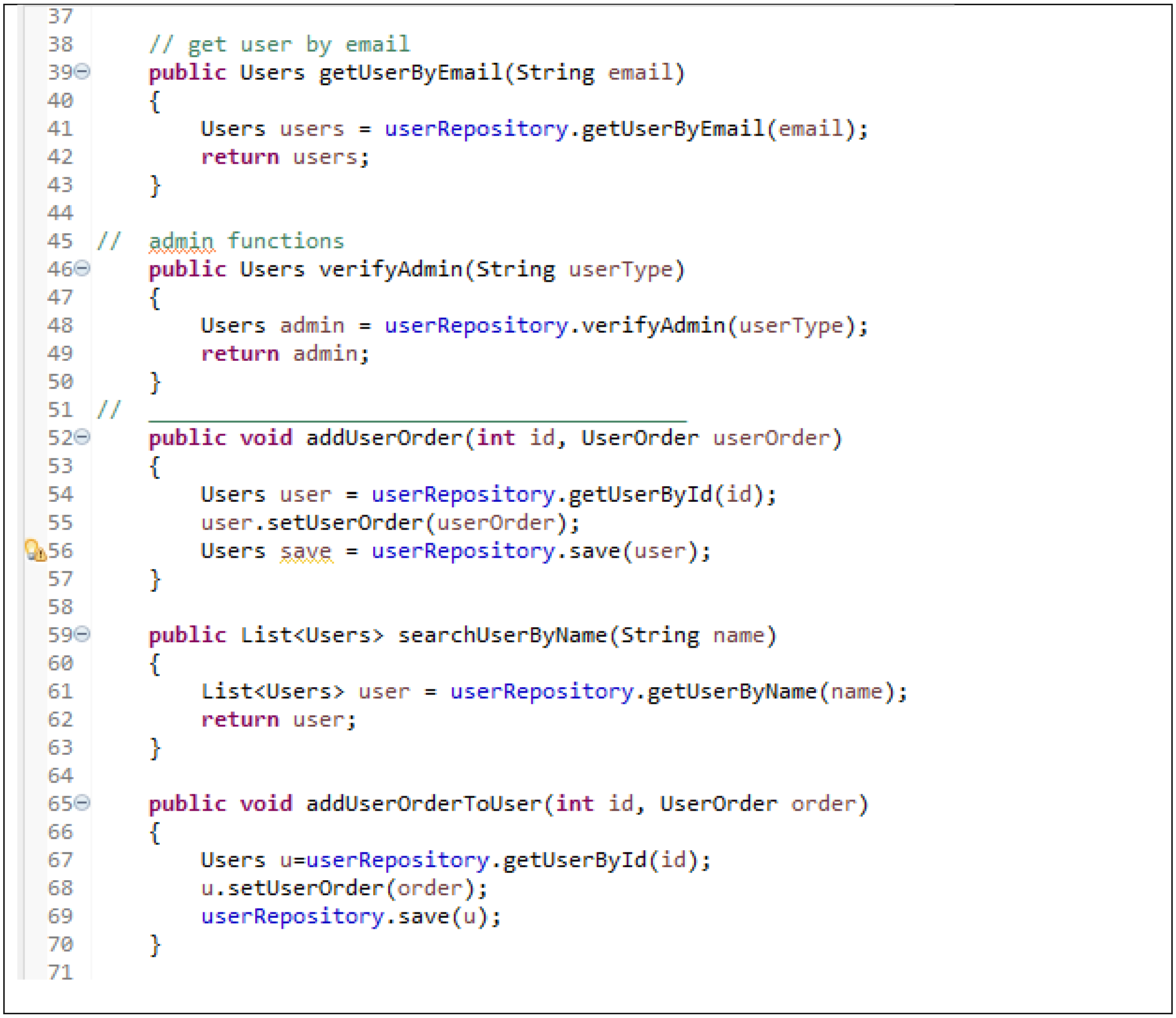


```
    public List<UserOrder> getAllOrders()
    {
        List<UserOrder> allOrders = userOrderRepository.getAllOrders();
        return allOrders;
    }

    public UserOrder createNewOrder(int id)
    {
        Users users=userRepository.getUserById(id);
        UserOrder userOrder=users.getUserOrder();
        if(userOrder==null)
        {
            userOrder=new UserOrder();
            userOrder.setDate(new Date());
            userOrder.setId(id);
            entityManager.persist(userOrder);
            userService.addUserOrder(id, userOrder);
            entityManager.persist(userOrder);
            return userOrder;
        } else
        {
            return userOrder;
        }

    }

    public UserOrder getOrder(int orderId)
    {
        UserOrder userOrder=userOrderRepository.getOrderById(orderId);
        return userOrder;
    }

}
```

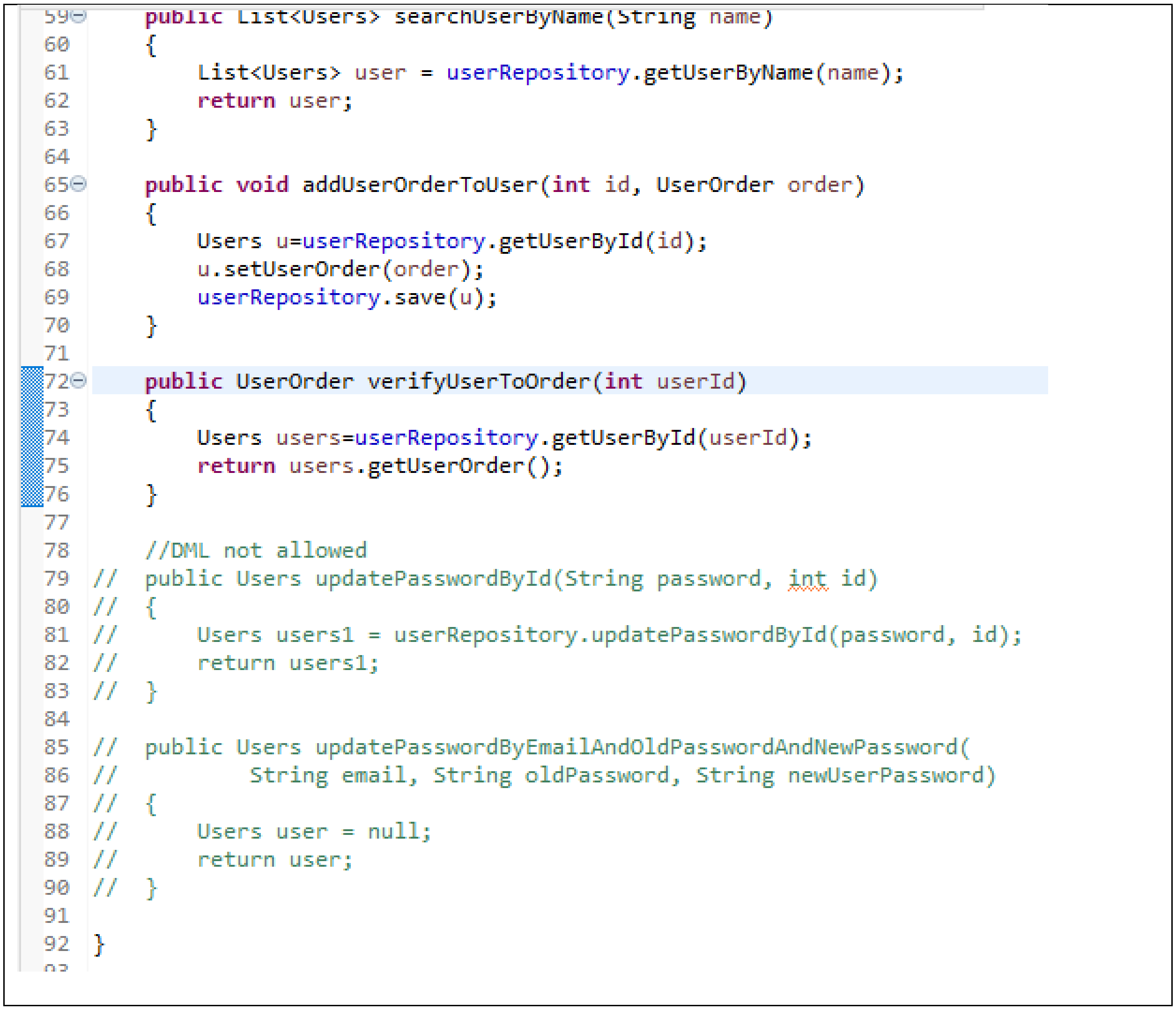
UserService.java



```
package com.sportyShoes.service;

import java.util.List;

@Service
public class UserService {
    @Autowired
    private UserRepository userRepository;

    // adding a user
    public Users addUser(Users u)
    {
        Users us=userRepository.save(u);
        return u;
    }

    // get all users
    public List<Users> getAllUsers()
    {
        List<Users> allUsers = userRepository.getAllUsers();
        return allUsers;
    }

    // get user by email and password
    public Users getUserByEmailAndPassword(String email, String password)
    {
        Users users = userRepository.getUserByEmailAndPassword(email, password);
        return users;
    }

    // get user by email
    public Users getUserByEmail(String email)
    {
        Users users = userRepository.getUserByEmail(email);
```



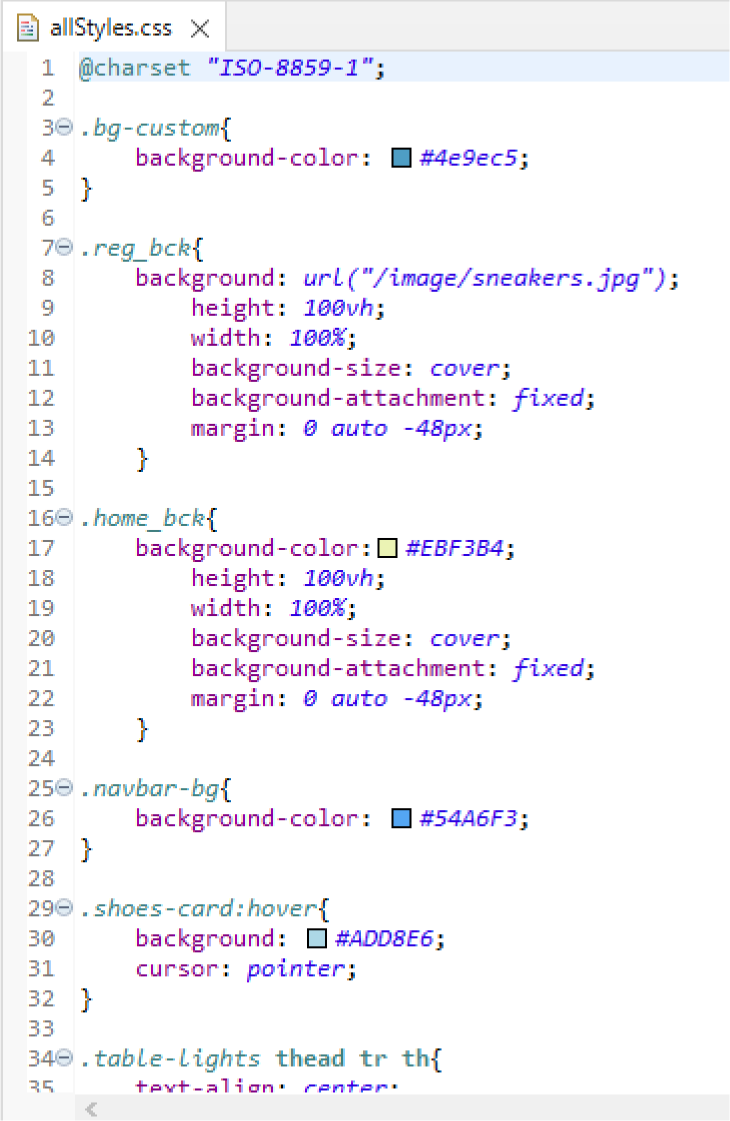
```java

    // get user by email
    public Users getUserByEmail(String email)
    {
        Users users = userRepository.getUserByEmail(email);
        return users;
    }

//  admin functions
    public Users verifyAdmin(String userType)
    {
        Users admin = userRepository.verifyAdmin(userType);
        return admin;
    }
//  _________________________________________
    public void addUserOrder(int id, UserOrder userOrder)
    {
        Users user = userRepository.getUserById(id);
        user.setUserOrder(userOrder);
        Users save = userRepository.save(user);
    }

    public List<Users> searchUserByName(String name)
    {
        List<Users> user = userRepository.getUserByName(name);
        return user;
    }

    public void addUserOrderToUser(int id, UserOrder order)
    {
        Users u=userRepository.getUserById(id);
        u.setUserOrder(order);
        userRepository.save(u);
    }

```



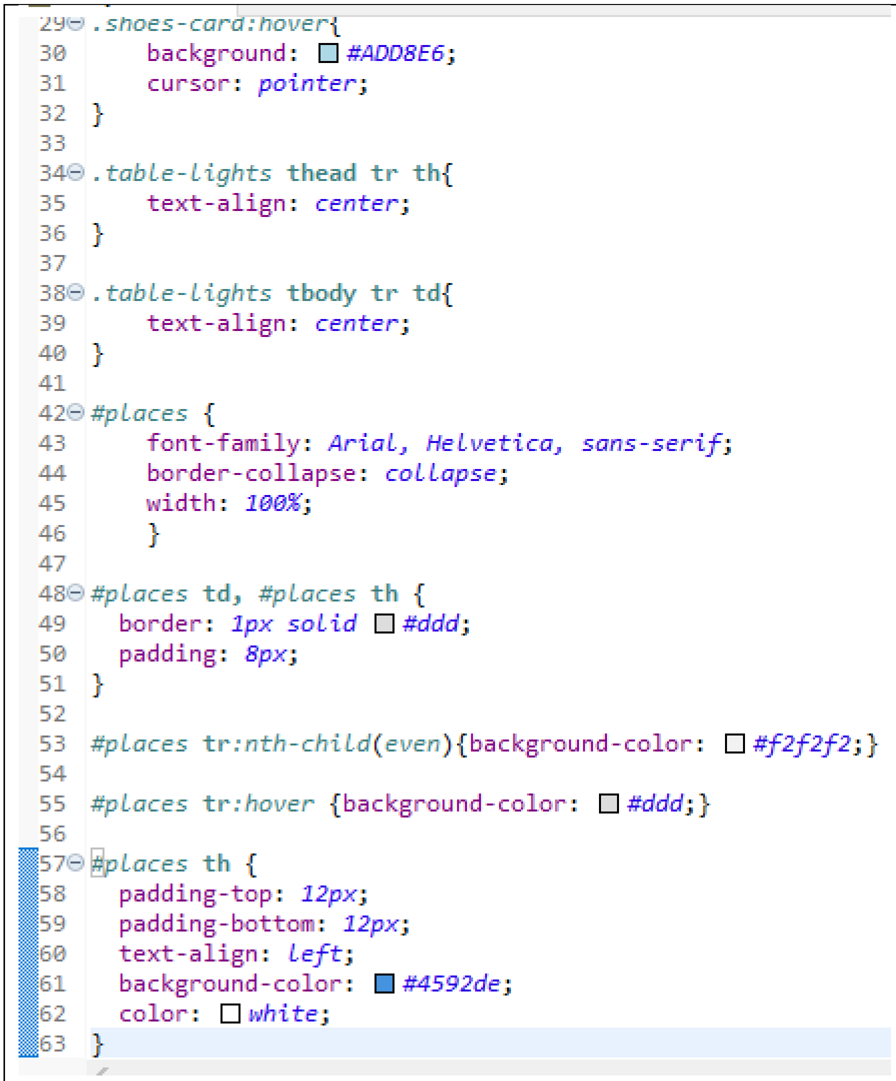
```
    public List<Users> searchUserByName(String name)
    {
        List<Users> user = userRepository.getUserByName(name);
        return user;
    }

    public void addUserOrderToUser(int id, UserOrder order)
    {
        Users u=userRepository.getUserById(id);
        u.setUserOrder(order);
        userRepository.save(u);
    }

    public UserOrder verifyUserToOrder(int userId)
    {
        Users users=userRepository.getUserById(userId);
        return users.getUserOrder();
    }

    //DML not allowed
//  public Users updatePasswordById(String password, int id)
//  {
//      Users users1 = userRepository.updatePasswordById(password, id);
//      return users1;
//  }

//  public Users updatePasswordByEmailAndOldPasswordAndNewPassword(
//          String email, String oldPassword, String newUserPassword)
//  {
//      Users user = null;
//      return user;
//  }

}
```

- ***CSS***

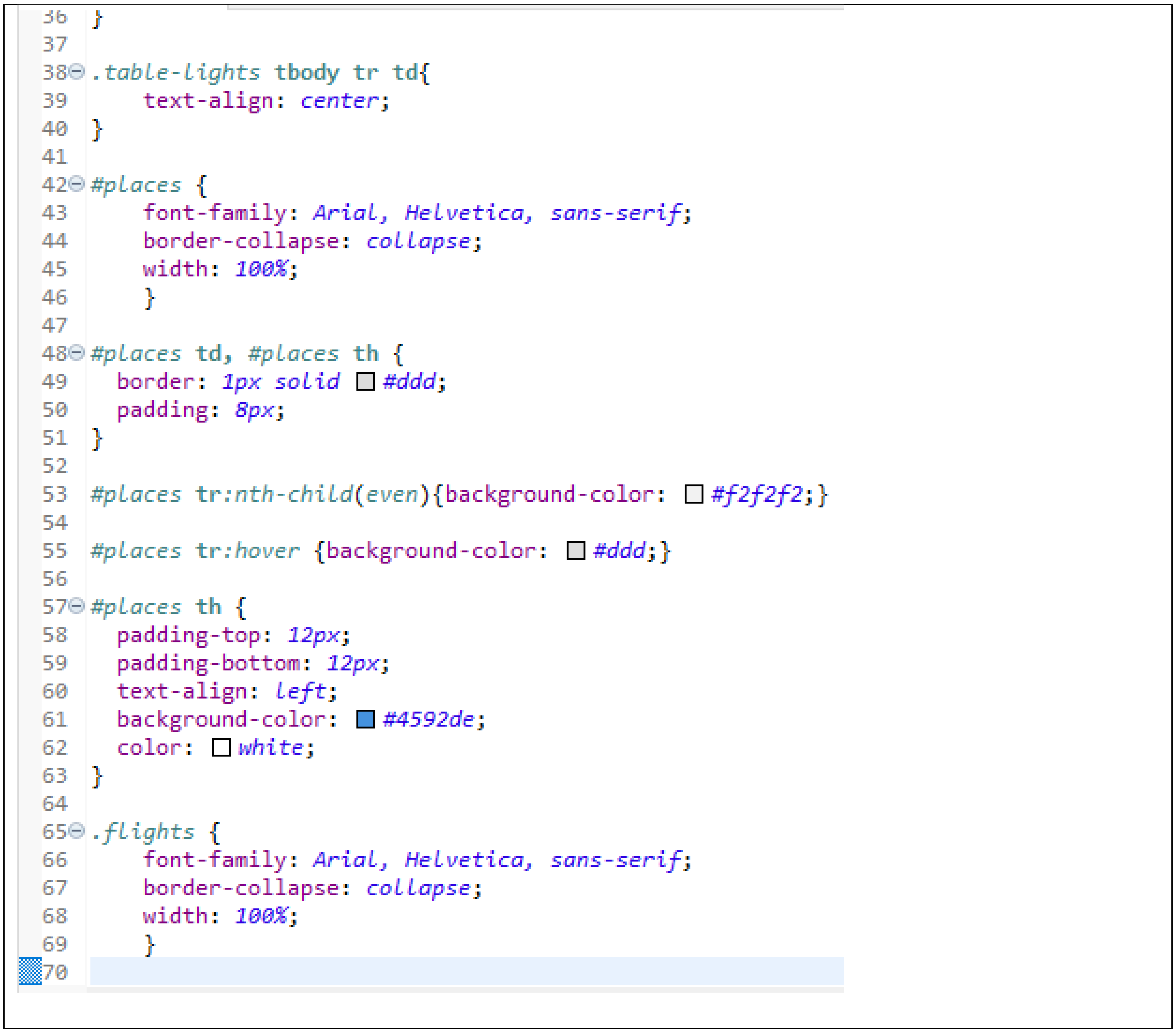
allStyles.css



```
@charset "ISO-8859-1";

.bg-custom{
    background-color: #4e9ec5;
}

.reg_bck{
    background: url("/image/sneakers.jpg");
        height: 100vh;
        width: 100%;
        background-size: cover;
        background-attachment: fixed;
        margin: 0 auto -48px;
    }

.home_bck{
    background-color: #EBF3B4;
        height: 100vh;
        width: 100%;
        background-size: cover;
        background-attachment: fixed;
        margin: 0 auto -48px;
    }

.navbar-bg{
    background-color: #54A6F3;
}

.shoes-card:hover{
    background: #ADD8E6;
    cursor: pointer;
}

.table-lights thead tr th{
    text-align: center;
```



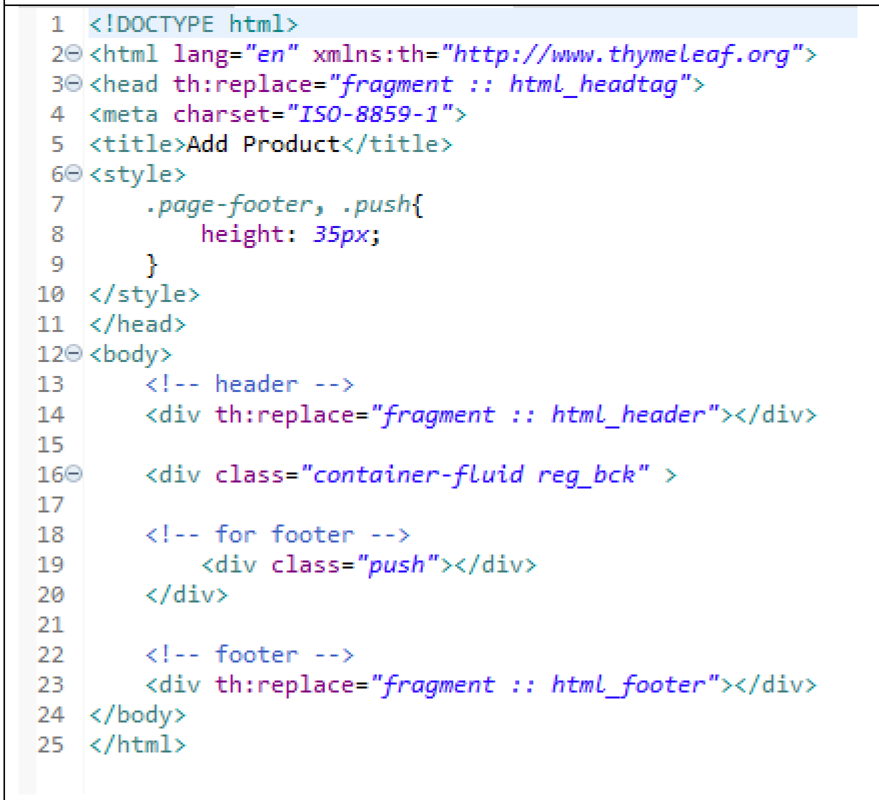
```
.shoes-card:hover{
    background: #ADD8E6;
    cursor: pointer;
}

.table-lights thead tr th{
    text-align: center;
}

.table-lights tbody tr td{
    text-align: center;
}

#places {
    font-family: Arial, Helvetica, sans-serif;
    border-collapse: collapse;
    width: 100%;
    }

#places td, #places th {
  border: 1px solid #ddd;
  padding: 8px;
}

#places tr:nth-child(even){background-color: #f2f2f2;}

#places tr:hover {background-color: #ddd;}

#places th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4592de;
  color: white;
}
```



```css
}

.table-lights tbody tr td{
    text-align: center;
}

#places {
    font-family: Arial, Helvetica, sans-serif;
    border-collapse: collapse;
    width: 100%;
    }

#places td, #places th {
  border: 1px solid #ddd;
  padding: 8px;
}

#places tr:nth-child(even){background-color: #f2f2f2;}

#places tr:hover {background-color: #ddd;}

#places th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #4592de;
  color: white;
}

.flights {
    font-family: Arial, Helvetica, sans-serif;
    border-collapse: collapse;
    width: 100%;
    }

```

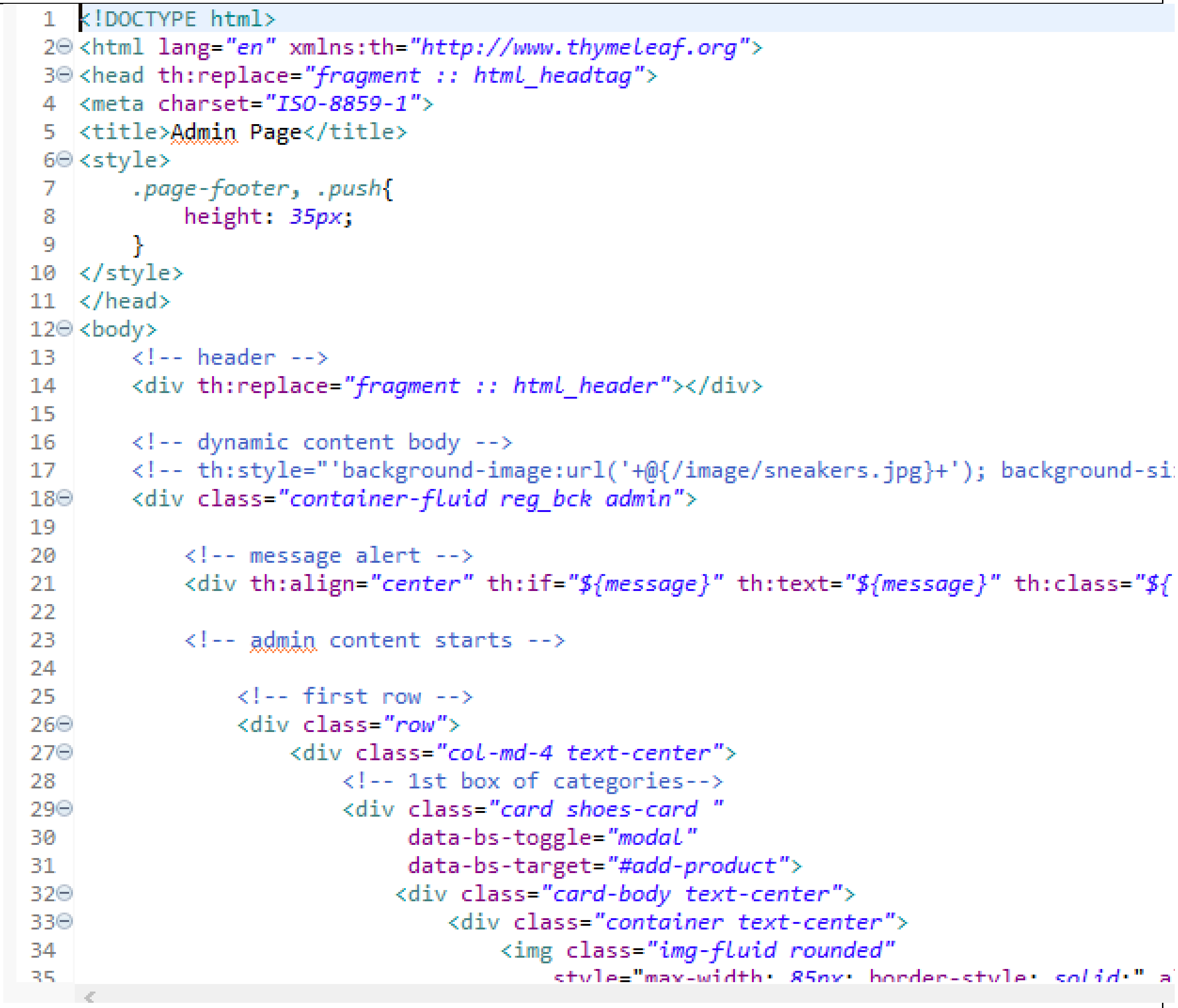
- ***TEMPLATES***

addProduct.html

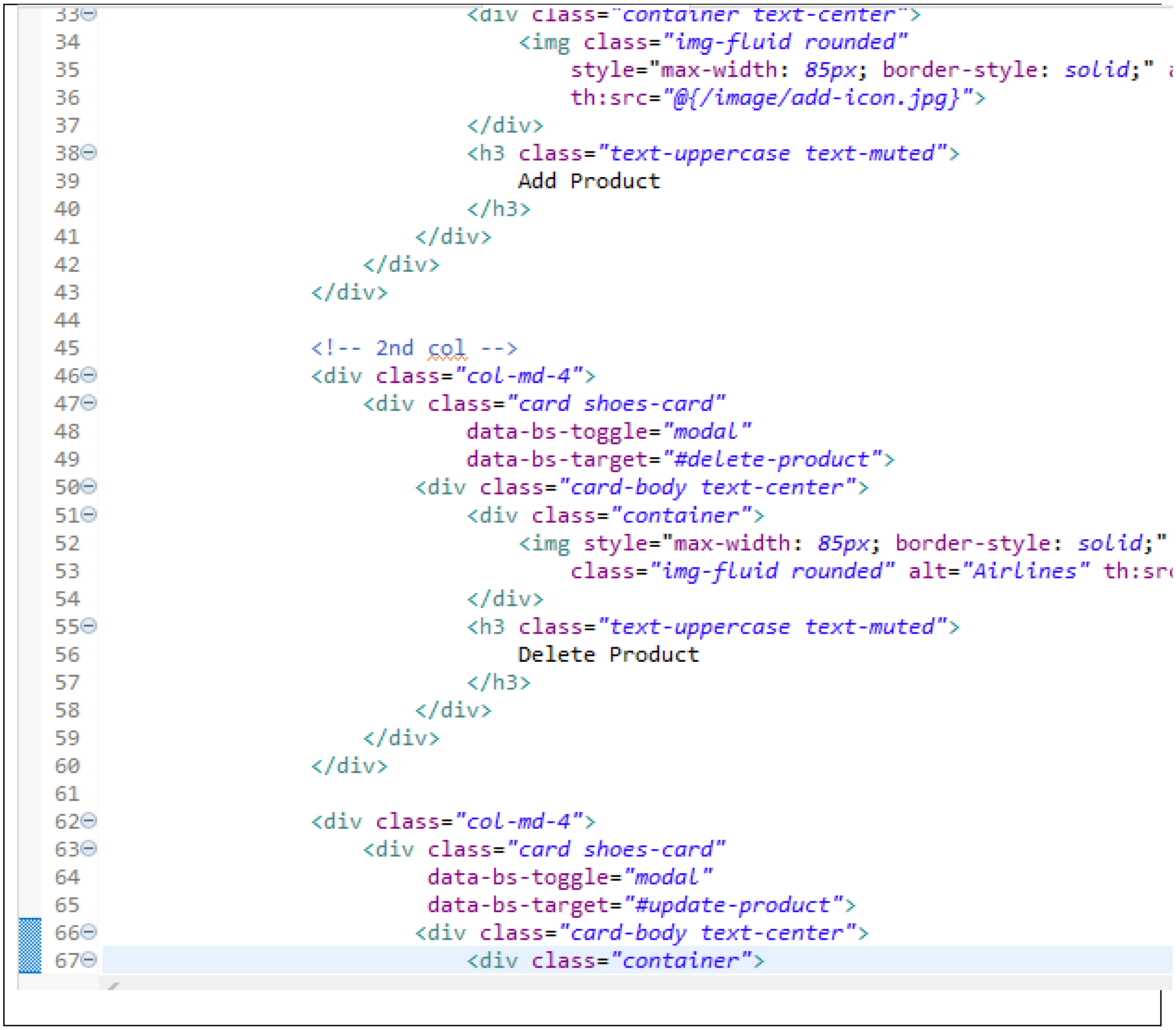


```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Add Product</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid reg_bck" >

    <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

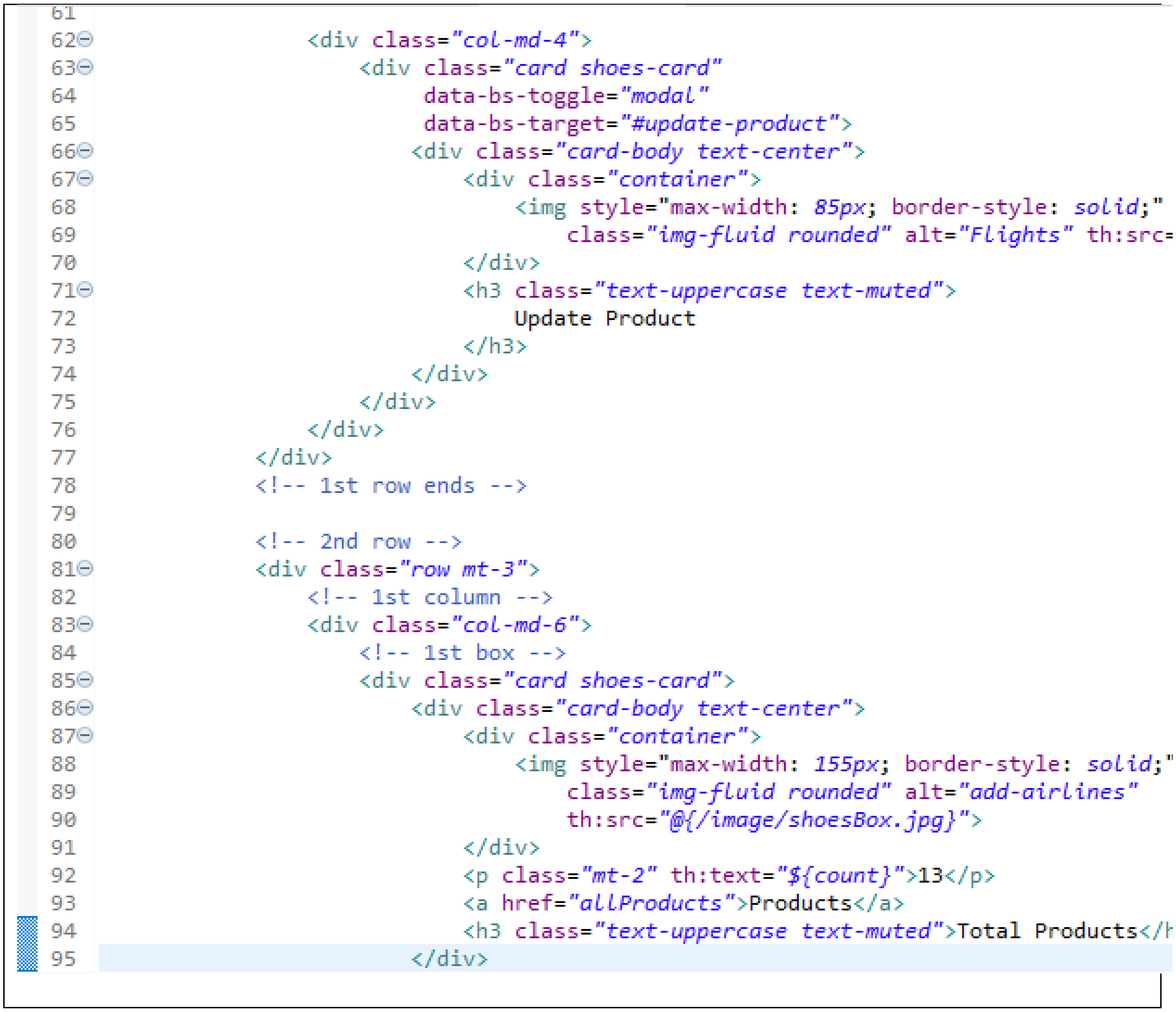
Admin.html



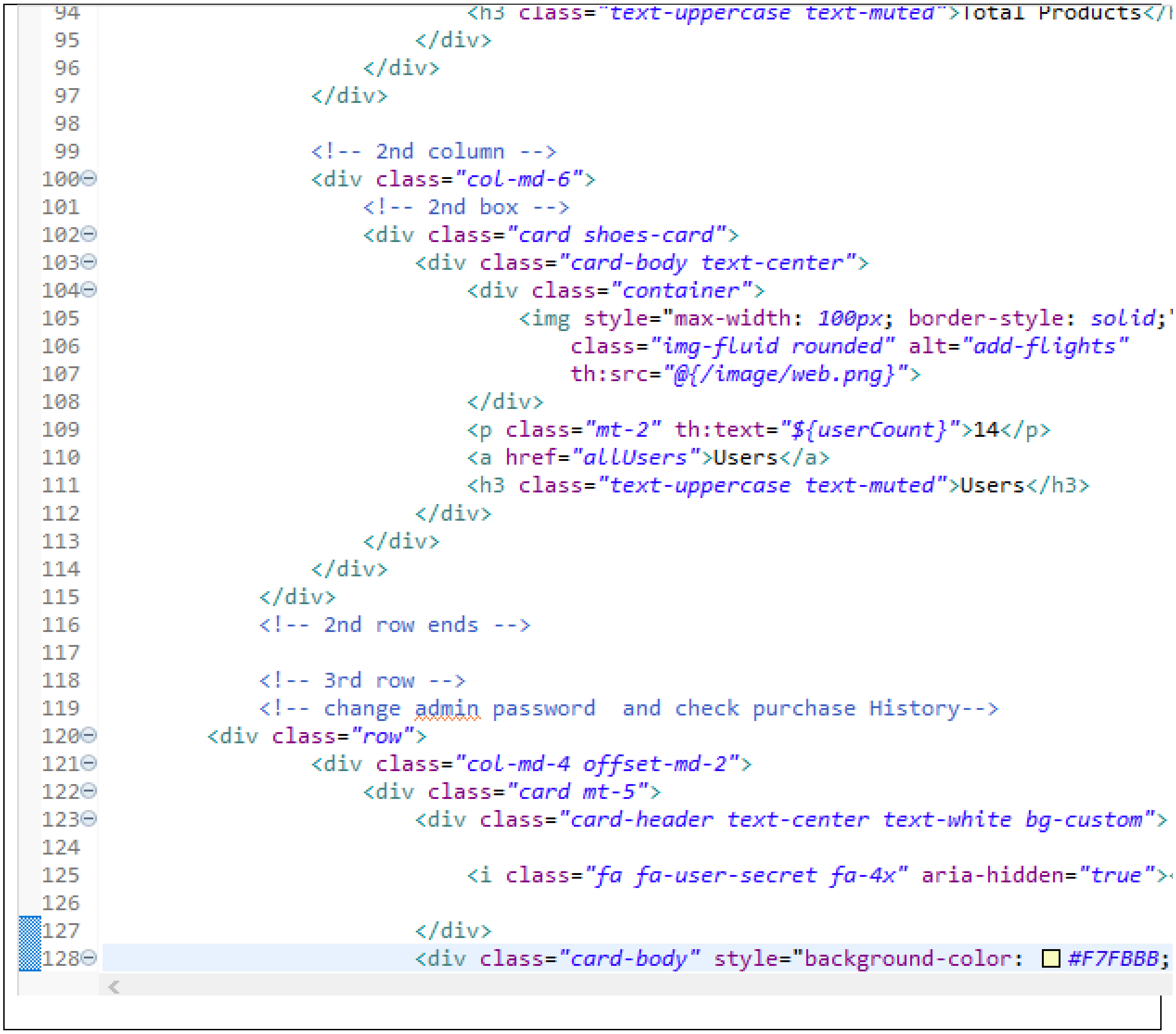
```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Admin Page</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <!-- dynamic content body -->
    <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background-si
    <div class="container-fluid reg_bck admin">

        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${

        <!-- admin content starts -->

            <!-- first row -->
            <div class="row">
                <div class="col-md-4 text-center">
                    <!-- 1st box of categories-->
                    <div class="card shoes-card "
                         data-bs-toggle="modal"
                         data-bs-target="#add-product">
                        <div class="card-body text-center">
                            <div class="container text-center">
                                <img class="img-fluid rounded"
                                    style="max-width: 85px; border-style: solid;" a
```



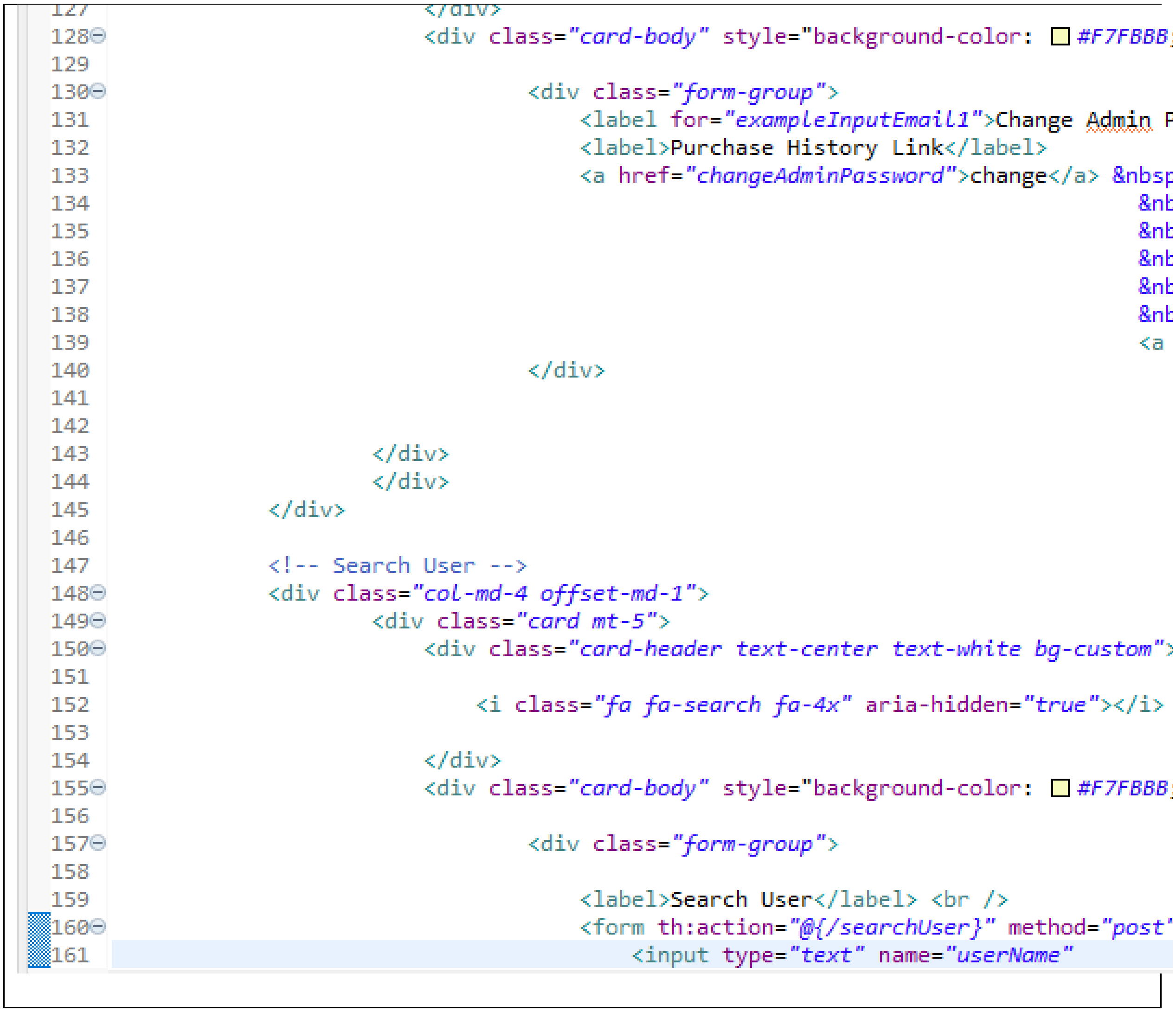
```
            <div class="container text-center">
                <img class="img-fluid rounded"
                    style="max-width: 85px; border-style: solid;" a
                    th:src="@{/image/add-icon.jpg}">
            </div>
            <h3 class="text-uppercase text-muted">
                Add Product
            </h3>
        </div>
    </div>
</div>

<!-- 2nd col -->
<div class="col-md-4">
    <div class="card shoes-card"
            data-bs-toggle="modal"
            data-bs-target="#delete-product">
        <div class="card-body text-center">
            <div class="container">
                <img style="max-width: 85px; border-style: solid;"
                    class="img-fluid rounded" alt="Airlines" th:sr
            </div>
            <h3 class="text-uppercase text-muted">
                Delete Product
            </h3>
        </div>
    </div>
</div>

<div class="col-md-4">
    <div class="card shoes-card"
         data-bs-toggle="modal"
         data-bs-target="#update-product">
        <div class="card-body text-center">
            <div class="container">
```



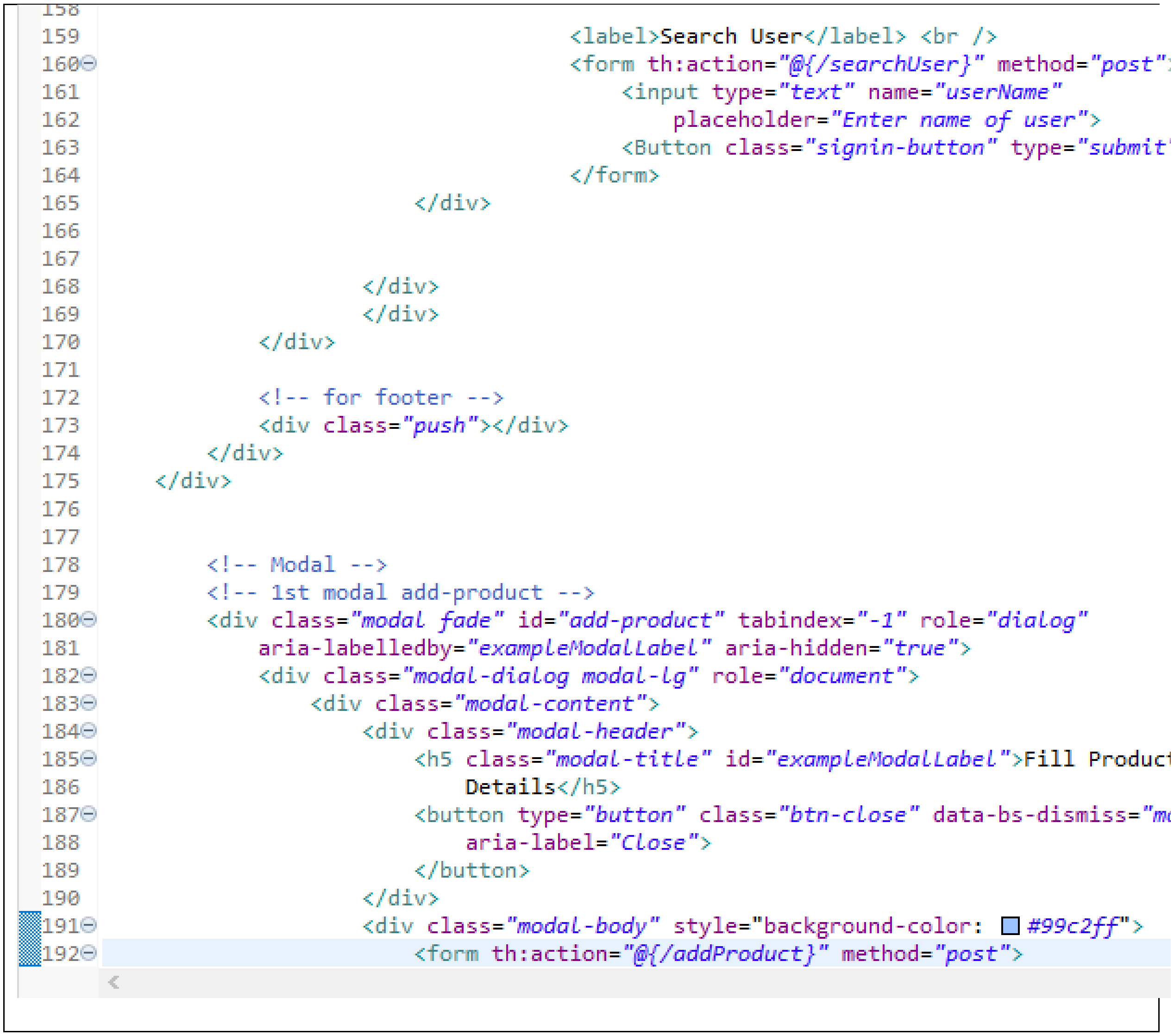
```
                <div class="col-md-4">
                    <div class="card shoes-card"
                         data-bs-toggle="modal"
                         data-bs-target="#update-product">
                        <div class="card-body text-center">
                            <div class="container">
                                <img style="max-width: 85px; border-style: solid;"
                                    class="img-fluid rounded" alt="Flights" th:src=
                            </div>
                            <h3 class="text-uppercase text-muted">
                                Update Product
                            </h3>
                        </div>
                    </div>
                </div>
            </div>
            <!-- 1st row ends -->

            <!-- 2nd row -->
            <div class="row mt-3">
                <!-- 1st column -->
                <div class="col-md-6">
                    <!-- 1st box -->
                    <div class="card shoes-card">
                        <div class="card-body text-center">
                            <div class="container">
                                <img style="max-width: 155px; border-style: solid;"
                                    class="img-fluid rounded" alt="add-airlines"
                                    th:src="@{/image/shoesBox.jpg}">
                            </div>
                            <p class="mt-2" th:text="${count}">13</p>
                            <a href="allProducts">Products</a>
                            <h3 class="text-uppercase text-muted">Total Products</h
                        </div>
```



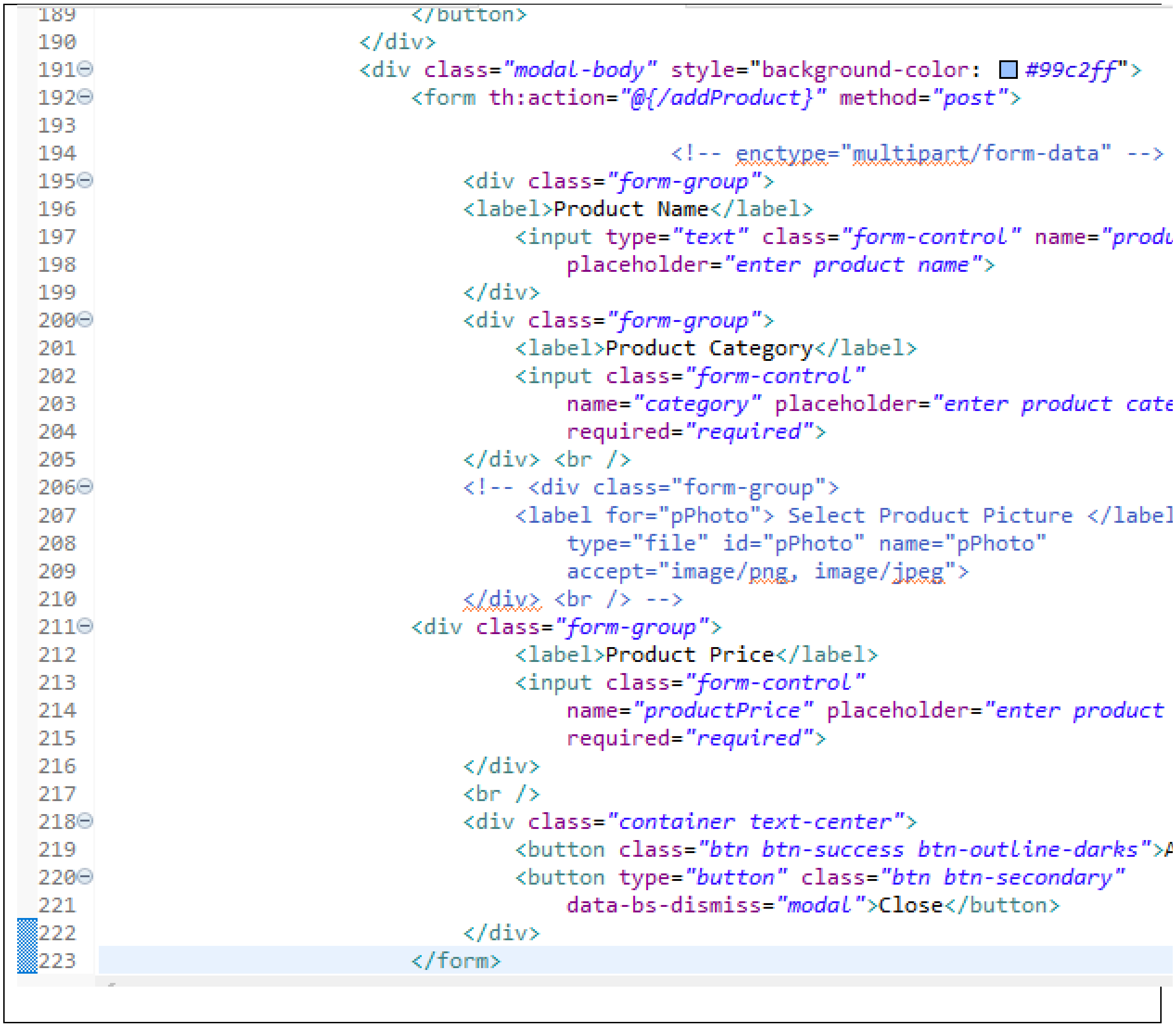
```
                            <h3 class="text-uppercase text-muted">Total Products</
                        </div>
                    </div>
                </div>

                <!-- 2nd column -->
                <div class="col-md-6">
                    <!-- 2nd box -->
                    <div class="card shoes-card">
                        <div class="card-body text-center">
                            <div class="container">
                                <img style="max-width: 100px; border-style: solid;'
                                    class="img-fluid rounded" alt="add-flights"
                                    th:src="@{/image/web.png}">
                            </div>
                            <p class="mt-2" th:text="${userCount}">14</p>
                            <a href="allUsers">Users</a>
                            <h3 class="text-uppercase text-muted">Users</h3>
                        </div>
                    </div>
                </div>
            </div>
            <!-- 2nd row ends -->

            <!-- 3rd row -->
            <!-- change admin password  and check purchase History-->
        <div class="row">
                <div class="col-md-4 offset-md-2">
                    <div class="card mt-5">
                        <div class="card-header text-center text-white bg-custom">

                            <i class="fa fa-user-secret fa-4x" aria-hidden="true">

                        </div>
                        <div class="card-body" style="background-color: #F7FBBB;
```



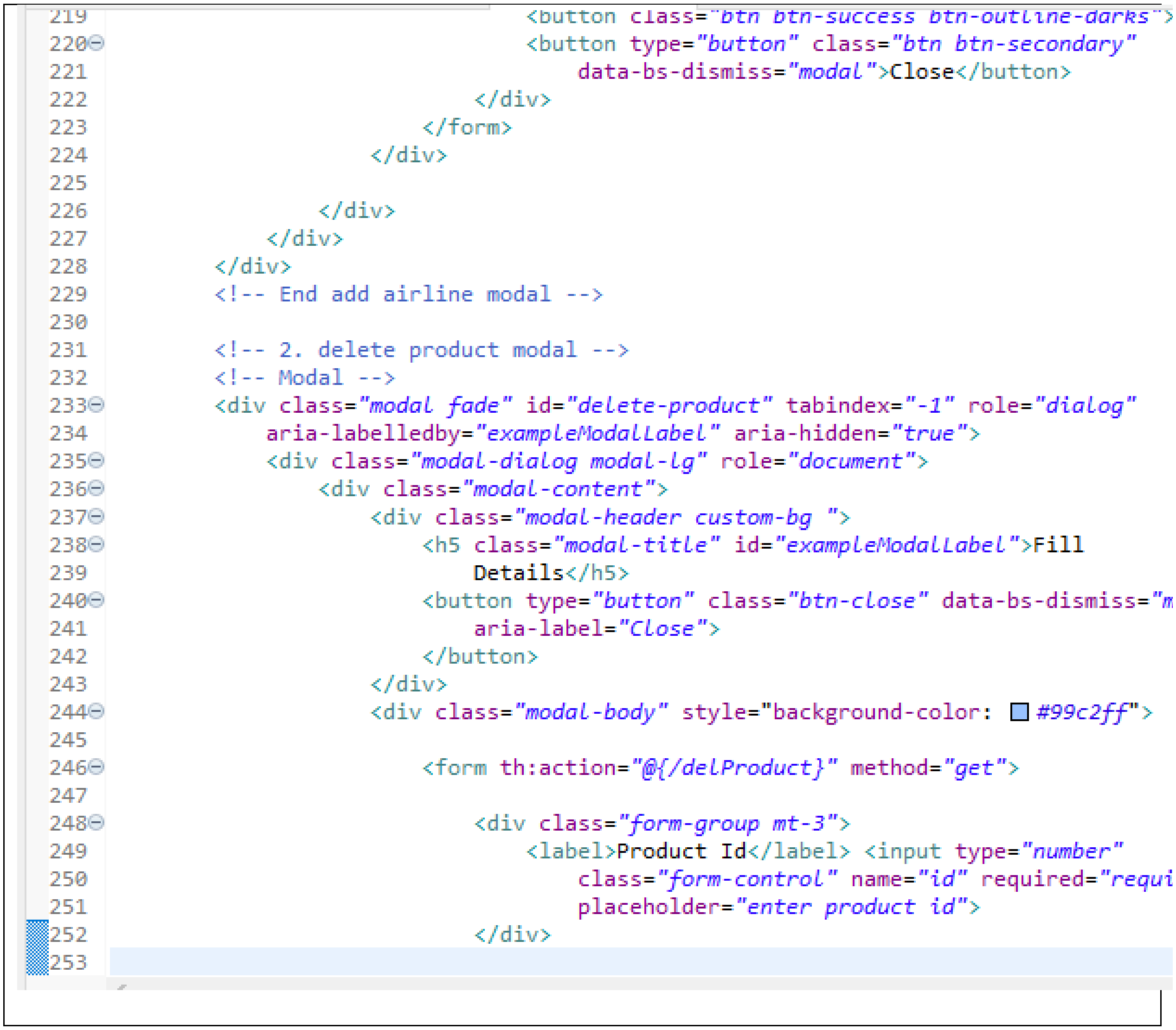
```
                        </div>
                        <div class="card-body" style="background-color: □ #F7FBBB
            
                                <div class="form-group">
                                    <label for="exampleInputEmail1">Change Admin P
                                    <label>Purchase History Link</label>
                                    <a href="changeAdminPassword">change</a> &nbsp
                                                                               &nb
                                                                               &nb
                                                                               &nb
                                                                               &nb
                                                                               &nb
                                                                               <a
                                </div>


                    </div>
                    </div>
            </div>

            <!-- Search User -->
            <div class="col-md-4 offset-md-1">
                    <div class="card mt-5">
                        <div class="card-header text-center text-white bg-custom">

                            <i class="fa fa-search fa-4x" aria-hidden="true"></i>

                        </div>
                        <div class="card-body" style="background-color: □ #F7FBBB

                                <div class="form-group">

                                    <label>Search User</label> <br />
                                    <form th:action="@{/searchUser}" method="post"
                                        <input type="text" name="userName"
```



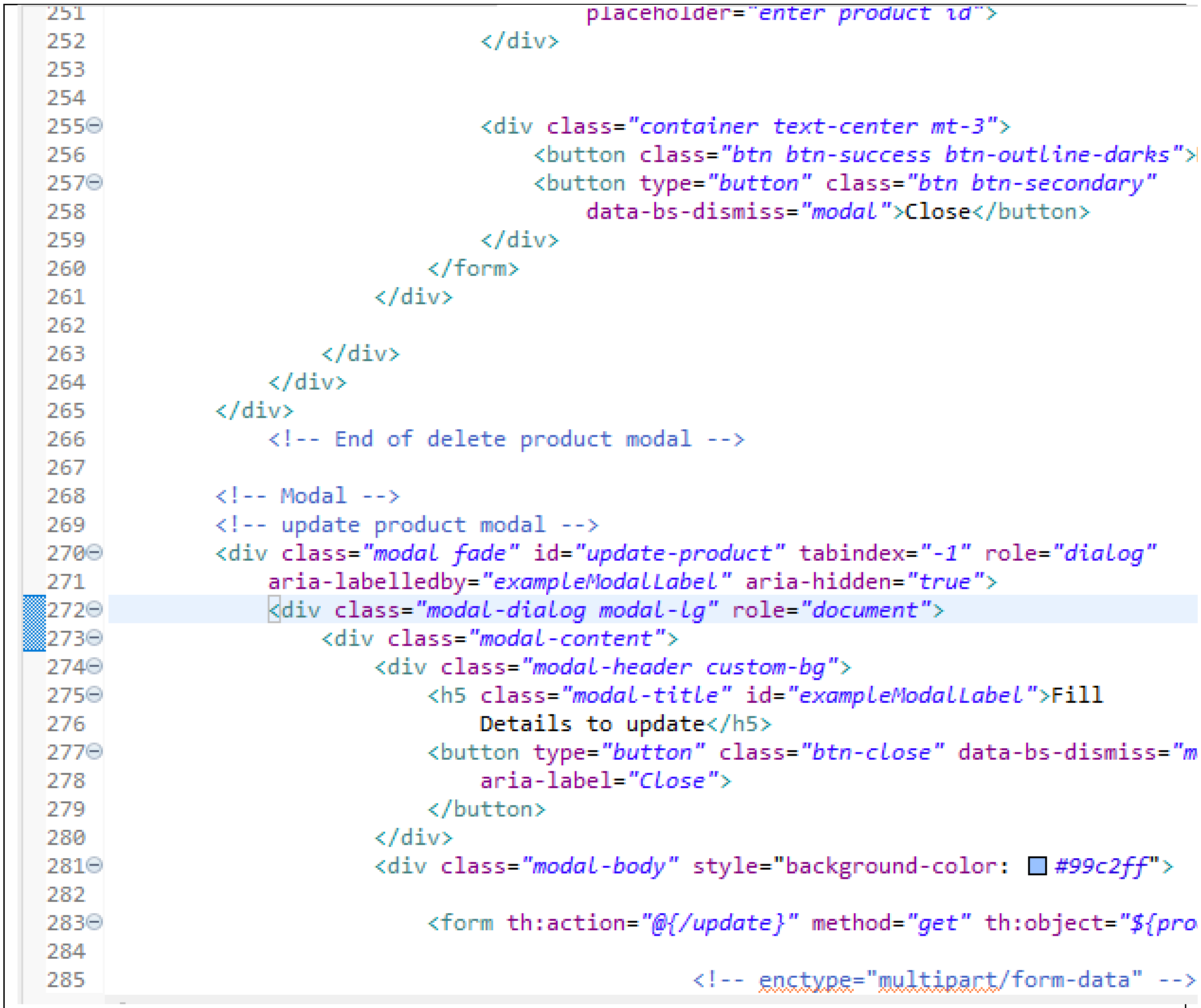
```
                                    <label>Search User</label> <br />
                                    <form th:action="@{/searchUser}" method="post"
                                        <input type="text" name="userName"
                                            placeholder="Enter name of user">
                                        <Button class="signin-button" type="submit
                                    </form>
                        </div>


                    </div>
                    </div>
            </div>

            <!-- for footer -->
            <div class="push"></div>
        </div>
    </div>


        <!-- Modal -->
        <!-- 1st modal add-product -->
        <div class="modal fade" id="add-product" tabindex="-1" role="dialog"
            aria-labelledby="exampleModalLabel" aria-hidden="true">
            <div class="modal-dialog modal-lg" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <h5 class="modal-title" id="exampleModalLabel">Fill Produc
                            Details</h5>
                        <button type="button" class="btn-close" data-bs-dismiss="m
                            aria-label="Close">
                        </button>
                    </div>
                    <div class="modal-body" style="background-color: #99c2ff">
                        <form th:action="@{/addProduct}" method="post">
```



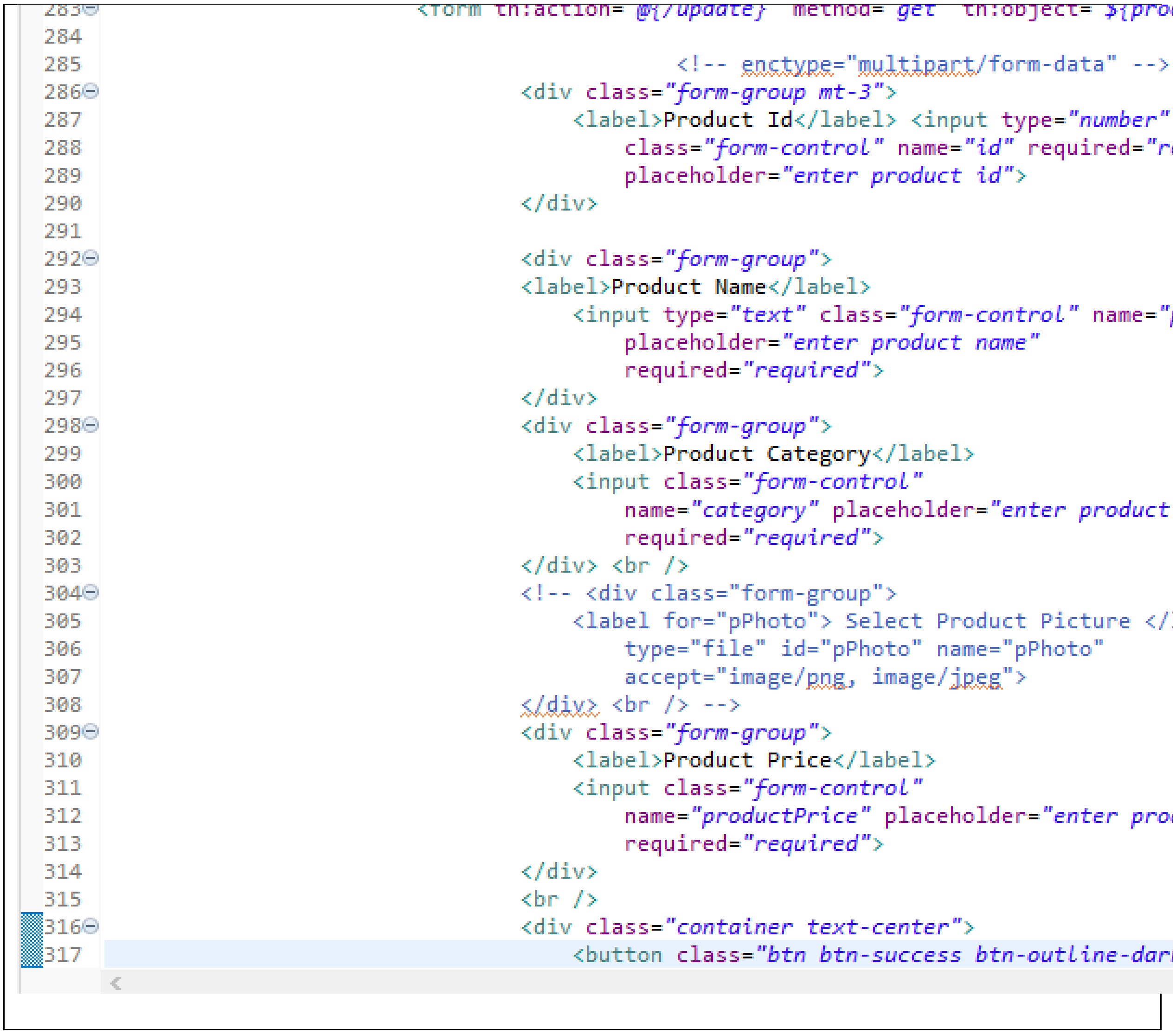
```
    </button>
</div>
<div class="modal-body" style="background-color: #99c2ff">
    <form th:action="@{/addProduct}" method="post">

                        <!-- enctype="multipart/form-data" -->
        <div class="form-group">
        <label>Product Name</label>
            <input type="text" class="form-control" name="produ
                placeholder="enter product name">
        </div>
        <div class="form-group">
            <label>Product Category</label>
            <input class="form-control"
                name="category" placeholder="enter product cate
                required="required">
        </div> <br />
        <!-- <div class="form-group">
            <label for="pPhoto"> Select Product Picture </label
                type="file" id="pPhoto" name="pPhoto"
                accept="image/png, image/jpeg">
        </div> <br /> -->
    <div class="form-group">
            <label>Product Price</label>
            <input class="form-control"
                name="productPrice" placeholder="enter product
                required="required">
        </div>
        <br />
        <div class="container text-center">
            <button class="btn btn-success btn-outline-darks">A
            <button type="button" class="btn btn-secondary"
                data-bs-dismiss="modal">Close</button>
        </div>
    </form>
```



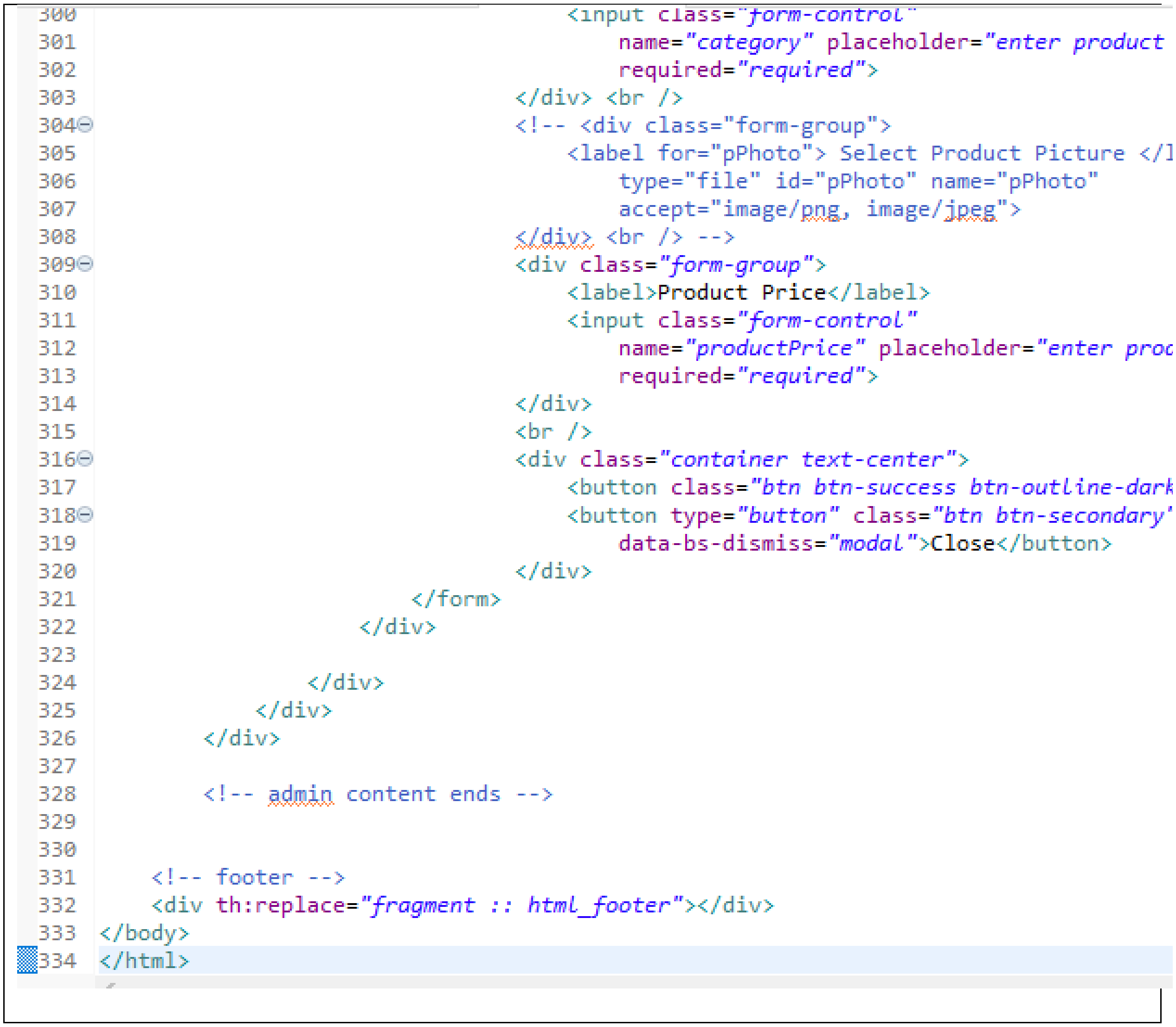
```
                                <button class="btn btn-success btn-outline-darks">
                                <button type="button" class="btn btn-secondary"
                                    data-bs-dismiss="modal">Close</button>
                            </div>
                        </form>
                    </div>

                </div>
            </div>
        </div>
        <!-- End add airline modal -->

        <!-- 2. delete product modal -->
        <!-- Modal -->
        <div class="modal fade" id="delete-product" tabindex="-1" role="dialog"
            aria-labelledby="exampleModalLabel" aria-hidden="true">
            <div class="modal-dialog modal-lg" role="document">
                <div class="modal-content">
                    <div class="modal-header custom-bg ">
                        <h5 class="modal-title" id="exampleModalLabel">Fill
                            Details</h5>
                        <button type="button" class="btn-close" data-bs-dismiss="m
                            aria-label="Close">
                        </button>
                    </div>
                    <div class="modal-body" style="background-color: ■ #99c2ff">

                        <form th:action="@{/delProduct}" method="get">

                            <div class="form-group mt-3">
                                <label>Product Id</label> <input type="number"
                                    class="form-control" name="id" required="requi
                                    placeholder="enter product id">
                            </div>

```



```
                                    placeholder="enter product id">
                            </div>


                            <div class="container text-center mt-3">
                                <button class="btn btn-success btn-outline-darks">
                                <button type="button" class="btn btn-secondary"
                                    data-bs-dismiss="modal">Close</button>
                            </div>
                        </form>
                    </div>

                </div>
            </div>
        </div>
            <!-- End of delete product modal -->

        <!-- Modal -->
        <!-- update product modal -->
        <div class="modal fade" id="update-product" tabindex="-1" role="dialog"
            aria-labelledby="exampleModalLabel" aria-hidden="true">
            <div class="modal-dialog modal-lg" role="document">
                <div class="modal-content">
                    <div class="modal-header custom-bg">
                        <h5 class="modal-title" id="exampleModalLabel">Fill
                            Details to update</h5>
                        <button type="button" class="btn-close" data-bs-dismiss="m
                            aria-label="Close">
                        </button>
                    </div>
                    <div class="modal-body" style="background-color: #99c2ff">

                        <form th:action="@{/update}" method="get" th:object="${pro

                                            <!-- enctype="multipart/form-data" -->
```

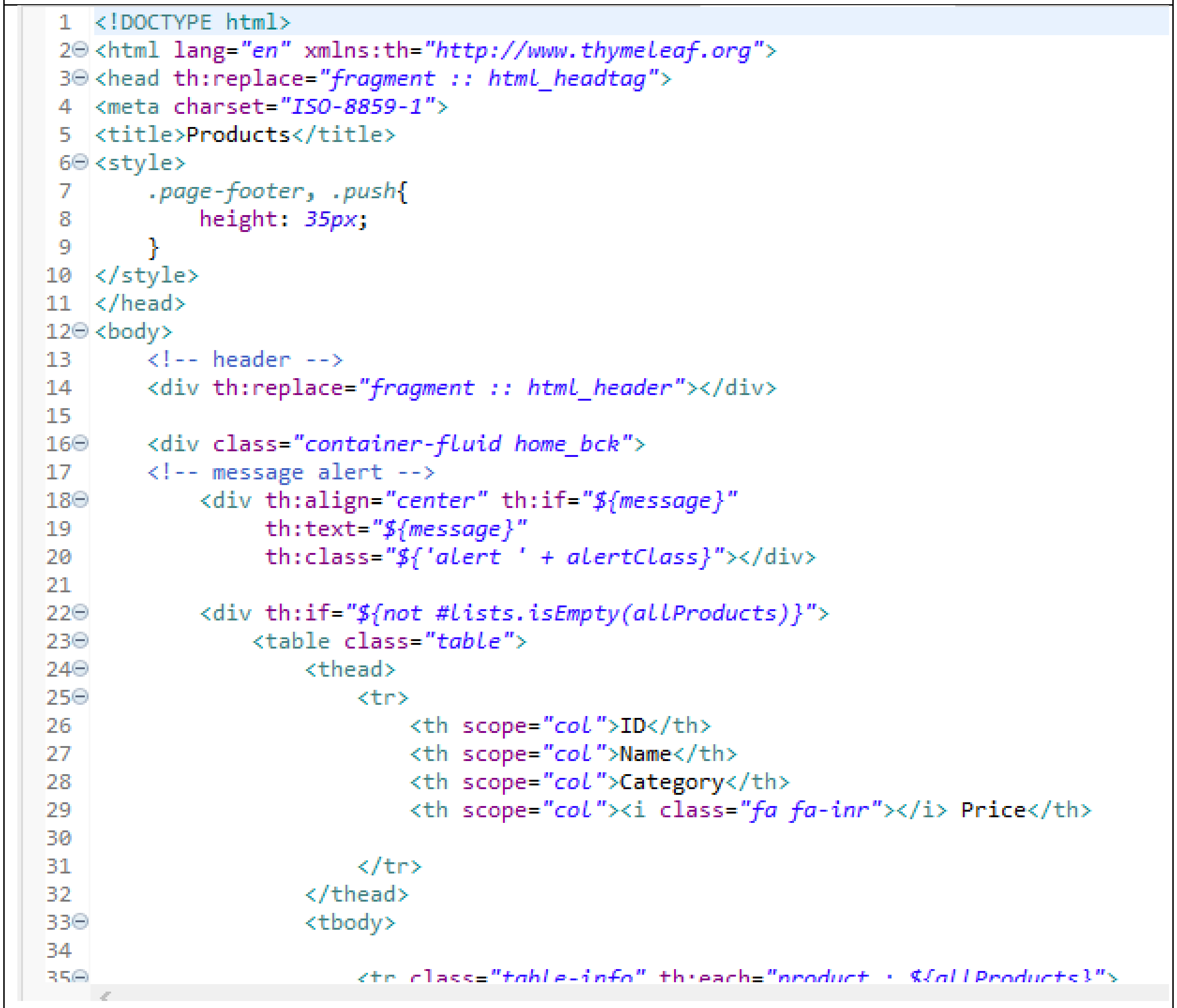


```
<form th:action="@{/update}" method="get" th:object="${pro

                    <!-- enctype="multipart/form-data" -->
        <div class="form-group mt-3">
            <label>Product Id</label> <input type="number"
                class="form-control" name="id" required="r
                placeholder="enter product id">
        </div>

        <div class="form-group">
        <label>Product Name</label>
            <input type="text" class="form-control" name="
                placeholder="enter product name"
                required="required">
        </div>
        <div class="form-group">
            <label>Product Category</label>
            <input class="form-control"
                name="category" placeholder="enter product
                required="required">
        </div> <br />
        <!-- <div class="form-group">
            <label for="pPhoto"> Select Product Picture </
                type="file" id="pPhoto" name="pPhoto"
                accept="image/png, image/jpeg">
        </div> <br /> -->
        <div class="form-group">
            <label>Product Price</label>
            <input class="form-control"
                name="productPrice" placeholder="enter pro
                required="required">
        </div>
        <br />
        <div class="container text-center">
            <button class="btn btn-success btn-outline-dar
```

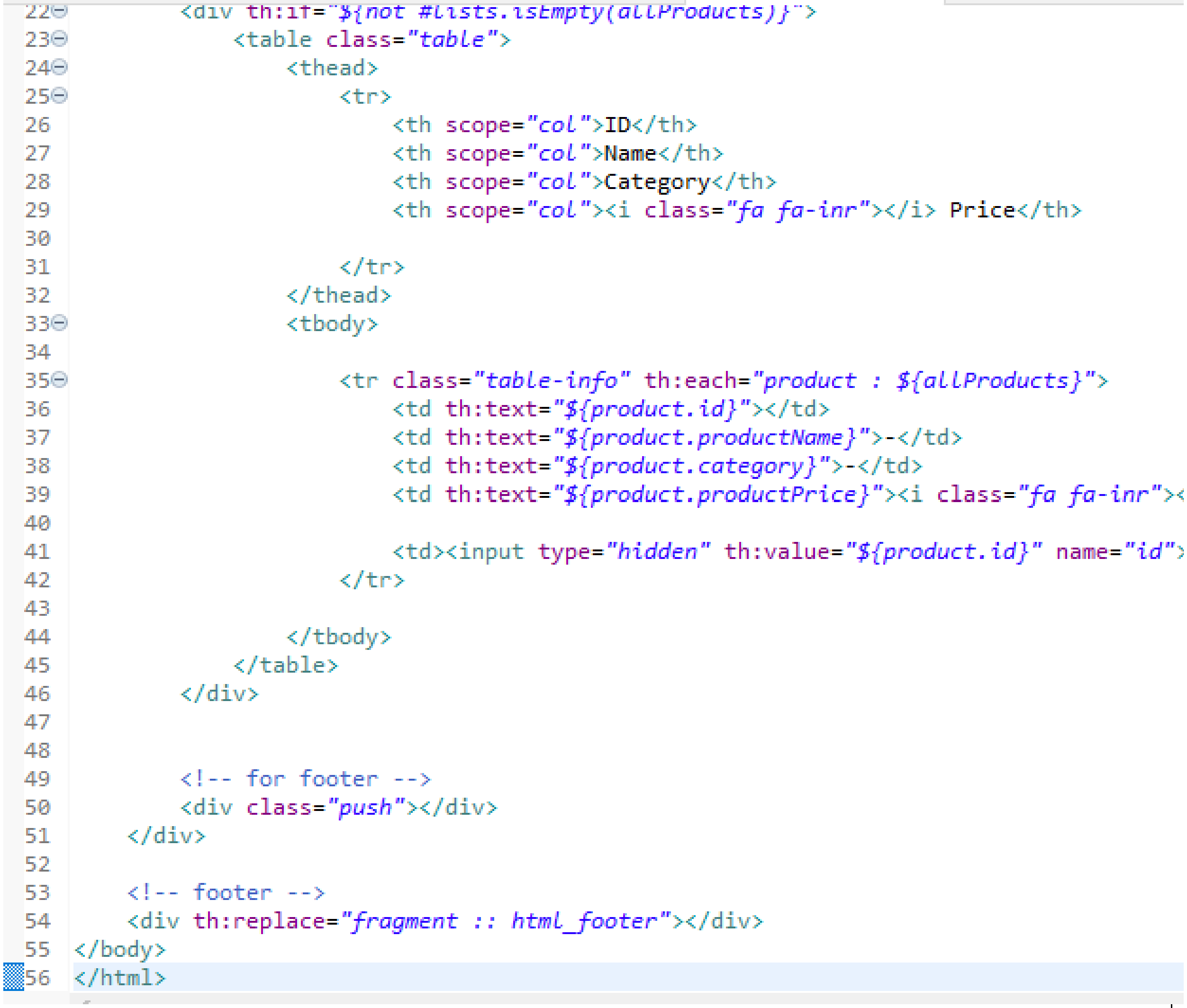


```
                                    <input class="form-control"
                                        name="category" placeholder="enter product
                                        required="required">
                                </div> <br />
                                <!-- <div class="form-group">
                                    <label for="pPhoto"> Select Product Picture </l
                                        type="file" id="pPhoto" name="pPhoto"
                                        accept="image/png, image/jpeg">
                                </div> <br /> -->
                                <div class="form-group">
                                    <label>Product Price</label>
                                    <input class="form-control"
                                        name="productPrice" placeholder="enter proc
                                        required="required">
                                </div>
                                <br />
                                <div class="container text-center">
                                    <button class="btn btn-success btn-outline-dark
                                    <button type="button" class="btn btn-secondary'
                                        data-bs-dismiss="modal">Close</button>
                                </div>
                        </form>
                    </div>

                </div>
            </div>
        </div>

        <!-- admin content ends -->


    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

AllProducts.html

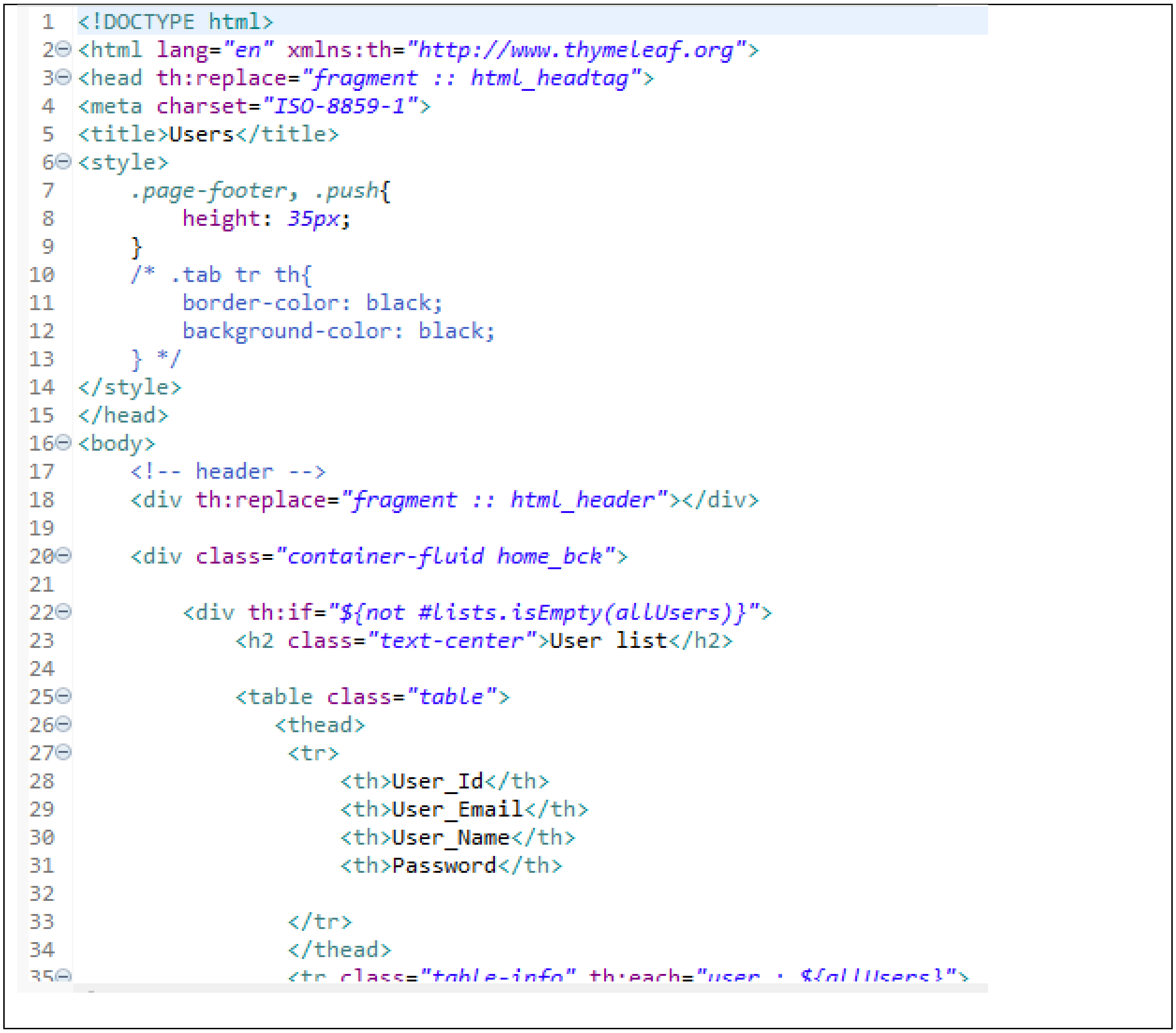


```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Products</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid home_bck">
    <!-- message alert -->
        <div th:align="center" th:if="${message}"
             th:text="${message}"
             th:class="${'alert ' + alertClass}"></div>

        <div th:if="${not #lists.isEmpty(allProducts)}">
            <table class="table">
                <thead>
                    <tr>
                        <th scope="col">ID</th>
                        <th scope="col">Name</th>
                        <th scope="col">Category</th>
                        <th scope="col"><i class="fa fa-inr"></i> Price</th>

                    </tr>
                </thead>
                <tbody>

                    <tr class="table-info" th:each="product : ${allProducts}">
```

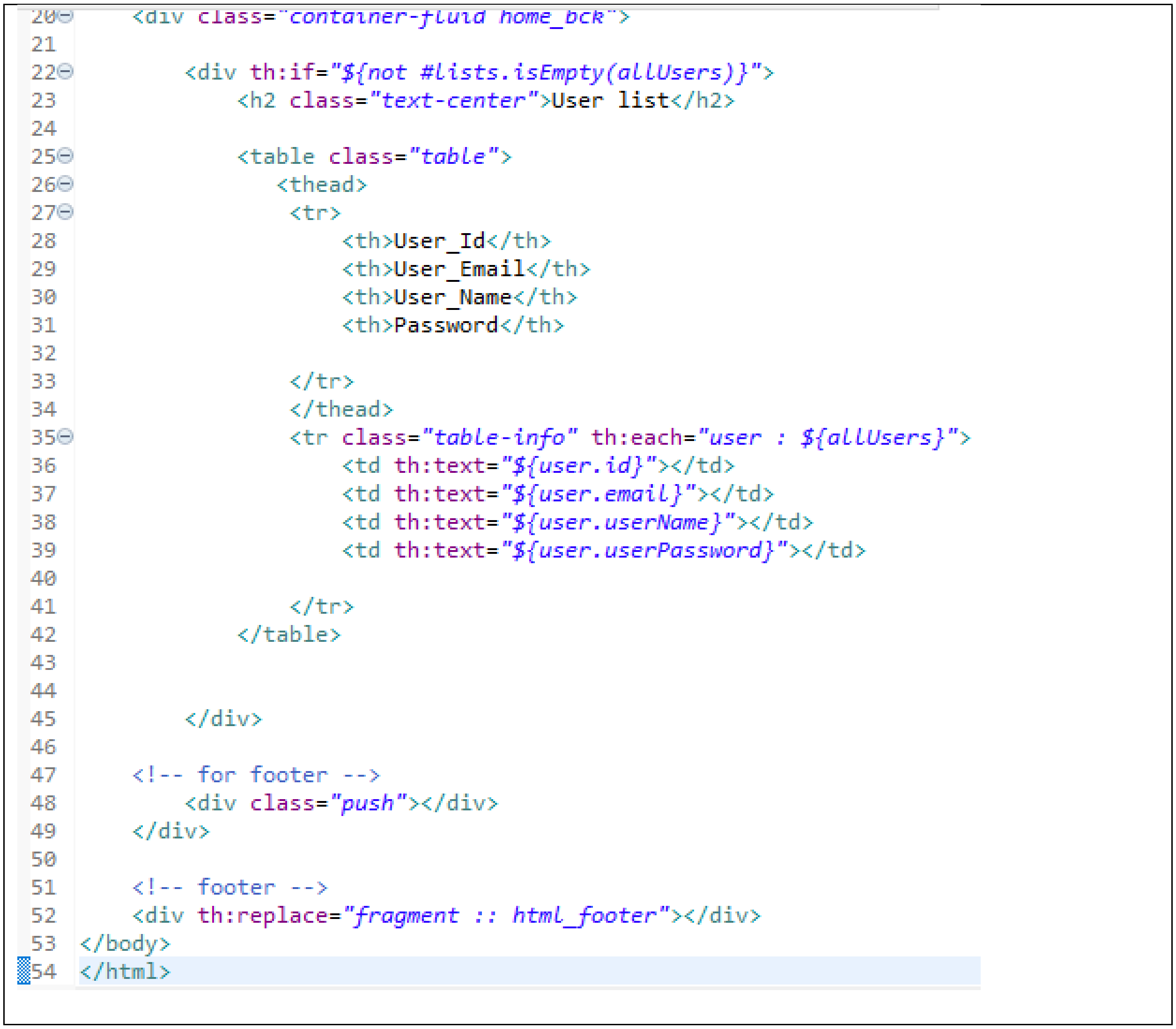


```
        <div th:if="${not #lists.isEmpty(allProducts)}">
            <table class="table">
                <thead>
                    <tr>
                        <th scope="col">ID</th>
                        <th scope="col">Name</th>
                        <th scope="col">Category</th>
                        <th scope="col"><i class="fa fa-inr"></i> Price</th>

                    </tr>
                </thead>
                <tbody>

                    <tr class="table-info" th:each="product : ${allProducts}">
                        <td th:text="${product.id}"></td>
                        <td th:text="${product.productName}">-</td>
                        <td th:text="${product.category}">-</td>
                        <td th:text="${product.productPrice}"><i class="fa fa-inr"><

                        <td><input type="hidden" th:value="${product.id}" name="id">
                    </tr>

                </tbody>
            </table>
        </div>


        <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

allUsers.html

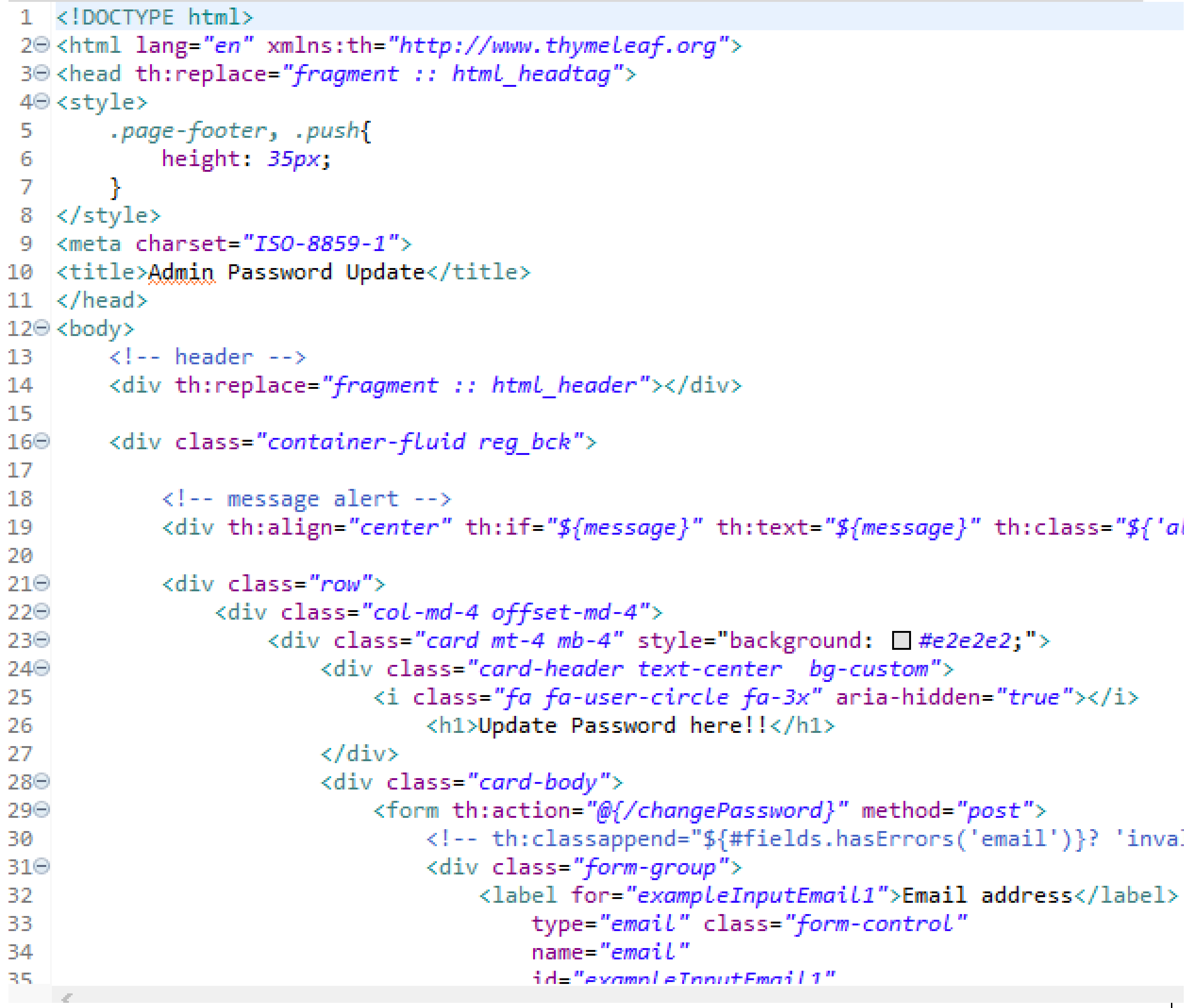


```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Users</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
    /* .tab tr th{
        border-color: black;
        background-color: black;
    } */
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid home_bck">

        <div th:if="${not #lists.isEmpty(allUsers)}">
            <h2 class="text-center">User list</h2>

            <table class="table">
               <thead>
                <tr>
                    <th>User_Id</th>
                    <th>User_Email</th>
                    <th>User_Name</th>
                    <th>Password</th>

                </tr>
                </thead>
                <tr class="table-info" th:each="user : ${allUsers}">
```

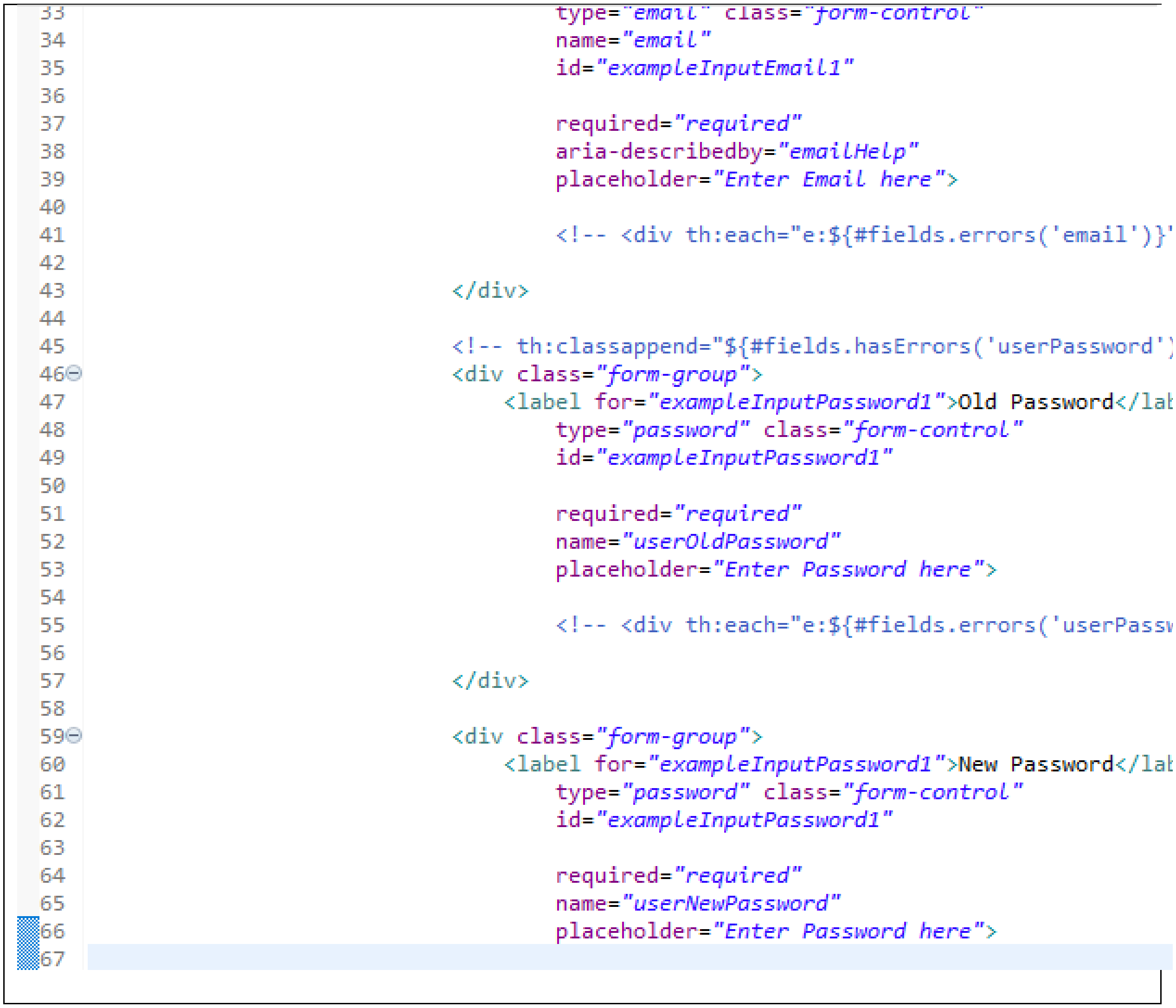


```
    <div class="container-fluid home_bck">

        <div th:if="${not #lists.isEmpty(allUsers)}">
            <h2 class="text-center">User list</h2>

            <table class="table">
               <thead>
                <tr>
                    <th>User_Id</th>
                    <th>User_Email</th>
                    <th>User_Name</th>
                    <th>Password</th>

                </tr>
                </thead>
                <tr class="table-info" th:each="user : ${allUsers}">
                    <td th:text="${user.id}"></td>
                    <td th:text="${user.email}"></td>
                    <td th:text="${user.userName}"></td>
                    <td th:text="${user.userPassword}"></td>

                </tr>
            </table>


        </div>

    <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

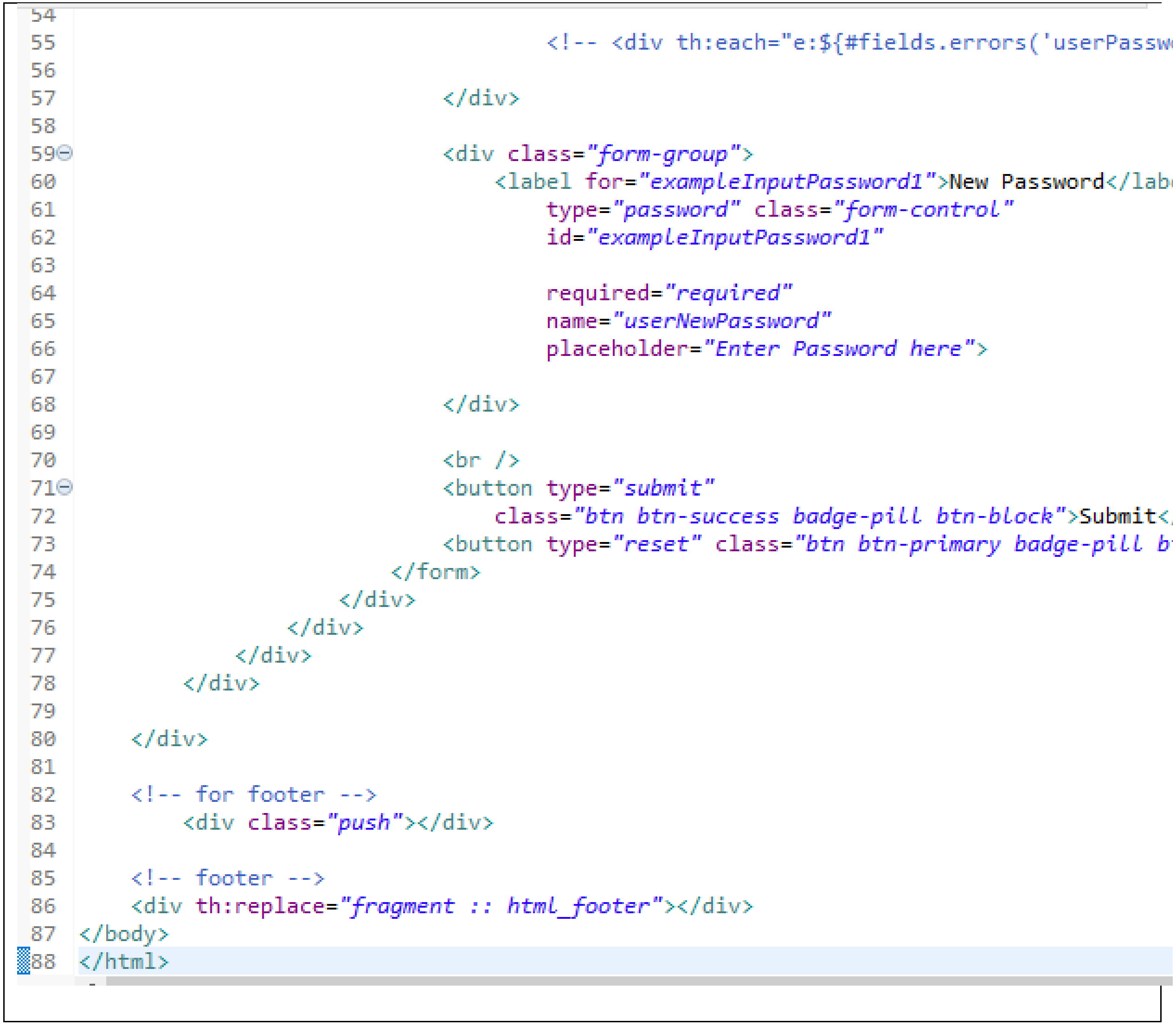
changeAdminPassword.html



```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
<meta charset="ISO-8859-1">
<title>Admin Password Update</title>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid reg_bck">

        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'a

        <div class="row">
            <div class="col-md-4 offset-md-4">
                <div class="card mt-4 mb-4" style="background: □ #e2e2e2;">
                    <div class="card-header text-center  bg-custom">
                        <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
                            <h1>Update Password here!!</h1>
                    </div>
                    <div class="card-body">
                        <form th:action="@{/changePassword}" method="post">
                            <!-- th:classappend="${#fields.hasErrors('email')}? 'inva
                            <div class="form-group">
                                <label for="exampleInputEmail1">Email address</label>
                                    type="email" class="form-control"
                                    name="email"
                                    id="exampleInputEmail1"
```

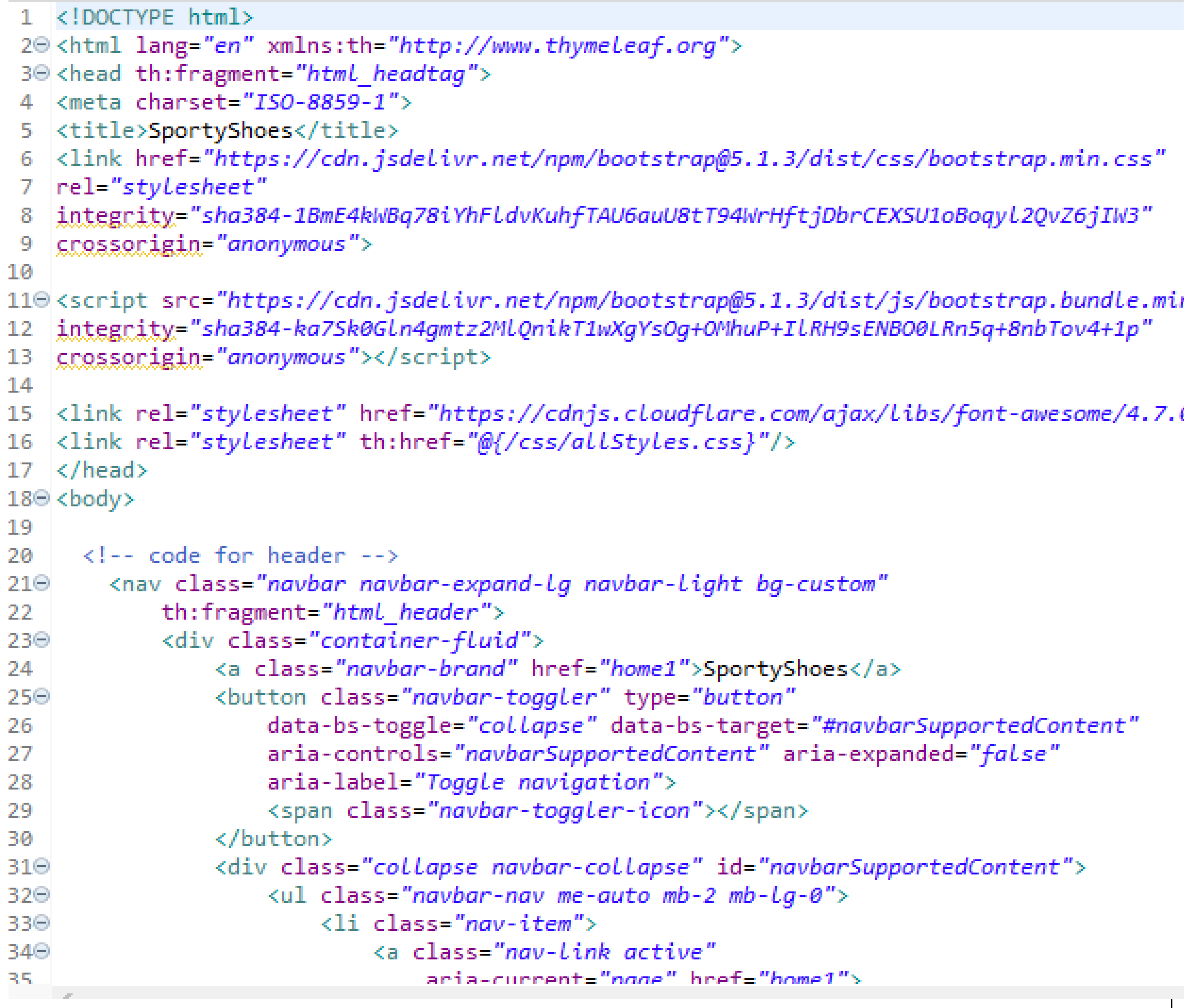


```
                                    type="email" class="form-control"
                                    name="email"
                                    id="exampleInputEmail1"

                                    required="required"
                                    aria-describedby="emailHelp"
                                    placeholder="Enter Email here">

                                    <!-- <div th:each="e:${#fields.errors('email')}'

                            </div>

                            <!-- th:classappend="${#fields.hasErrors('userPassword')
                            <div class="form-group">
                                <label for="exampleInputPassword1">Old Password</lab
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    required="required"
                                    name="userOldPassword"
                                    placeholder="Enter Password here">

                                    <!-- <div th:each="e:${#fields.errors('userPassw

                            </div>

                            <div class="form-group">
                                <label for="exampleInputPassword1">New Password</lab
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    required="required"
                                    name="userNewPassword"
                                    placeholder="Enter Password here">

```

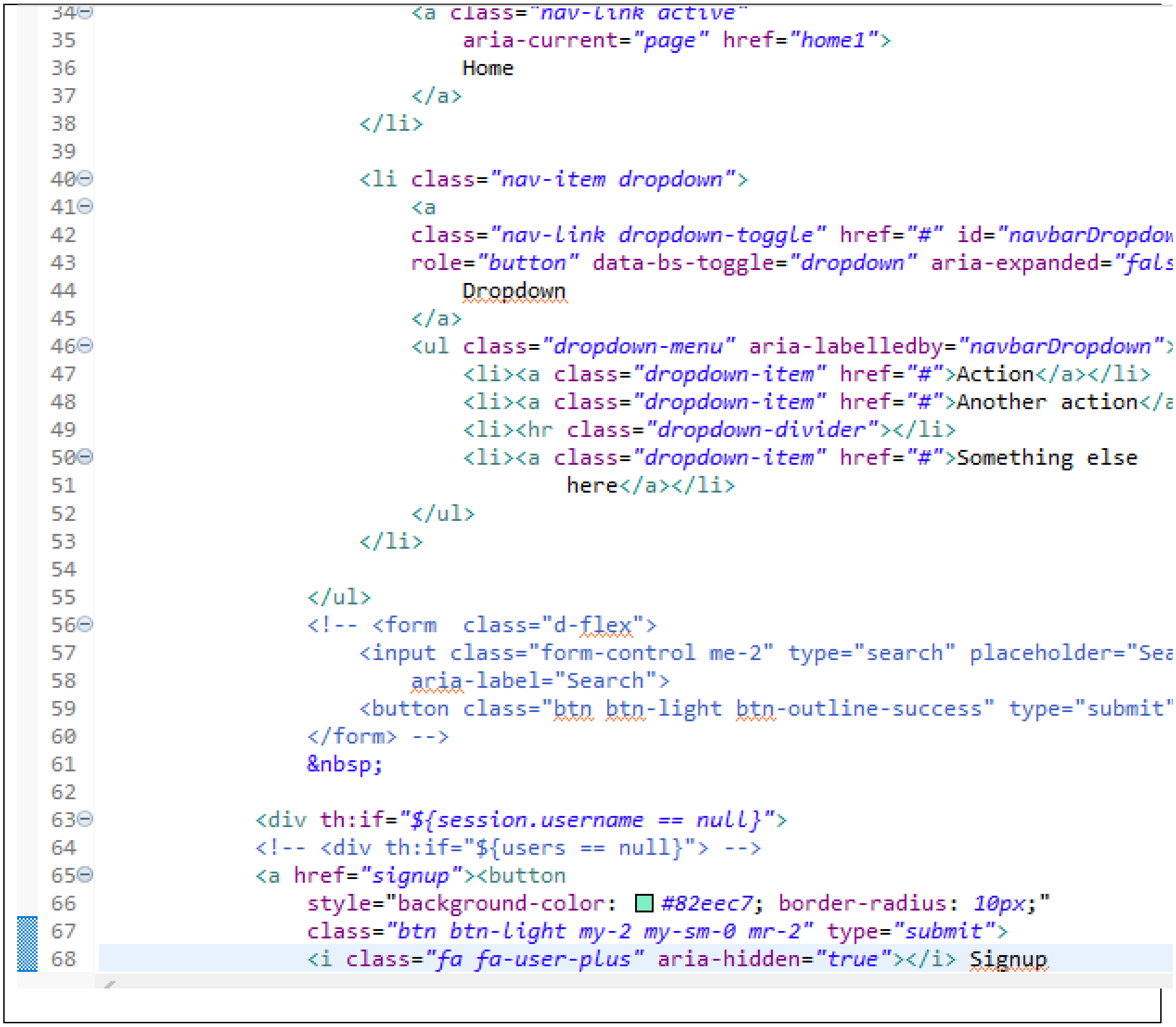


```
                                    <!-- <div th:each="e:${#fields.errors('userPassw

                            </div>

                            <div class="form-group">
                                <label for="exampleInputPassword1">New Password</lab
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    required="required"
                                    name="userNewPassword"
                                    placeholder="Enter Password here">

                            </div>

                            <br />
                            <button type="submit"
                                class="btn btn-success badge-pill btn-block">Submit<
                            <button type="reset" class="btn btn-primary badge-pill b
                        </form>
                    </div>
                </div>
            </div>
        </div>

    </div>

    <!-- for footer -->
        <div class="push"></div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

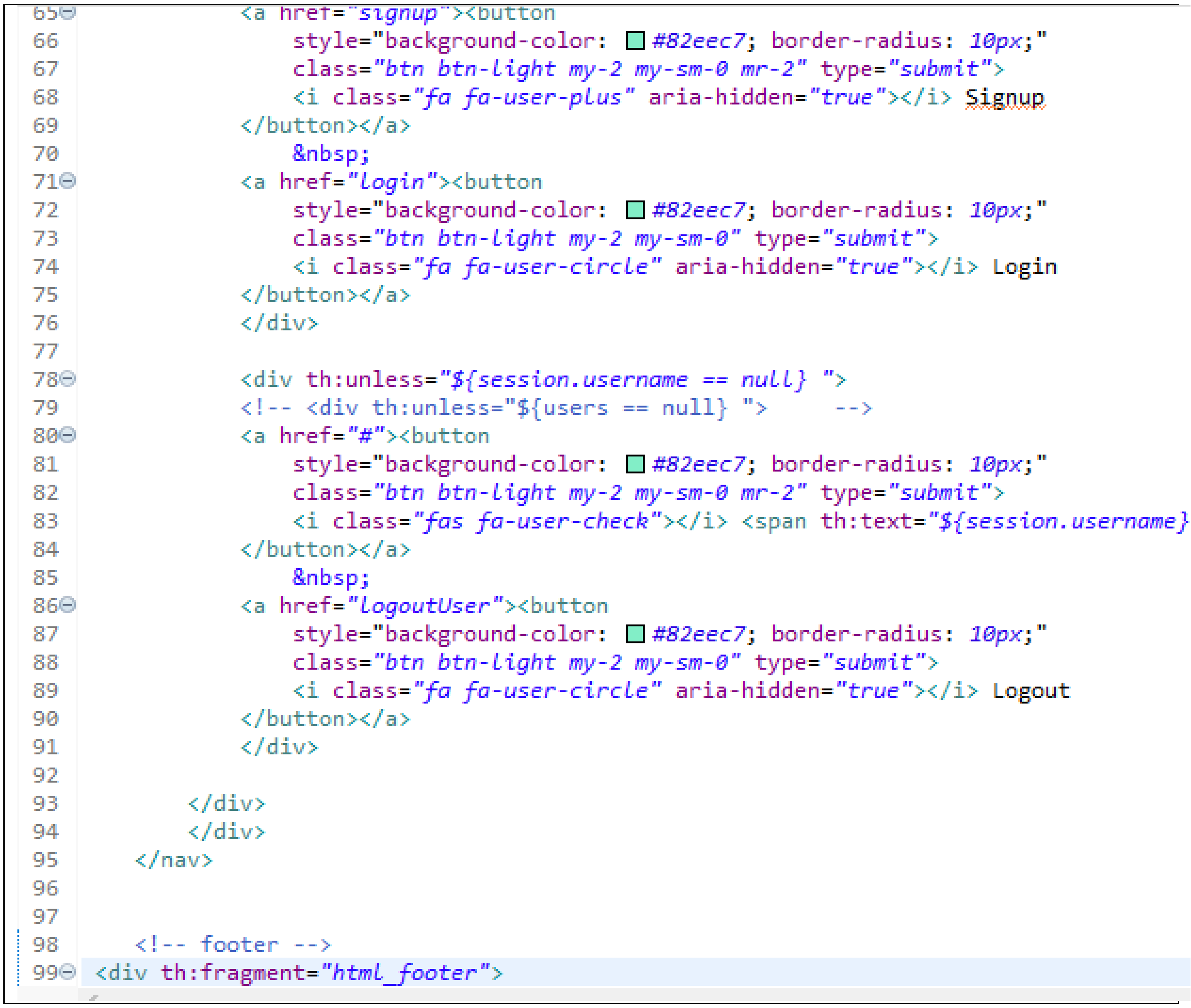
Fragment.html



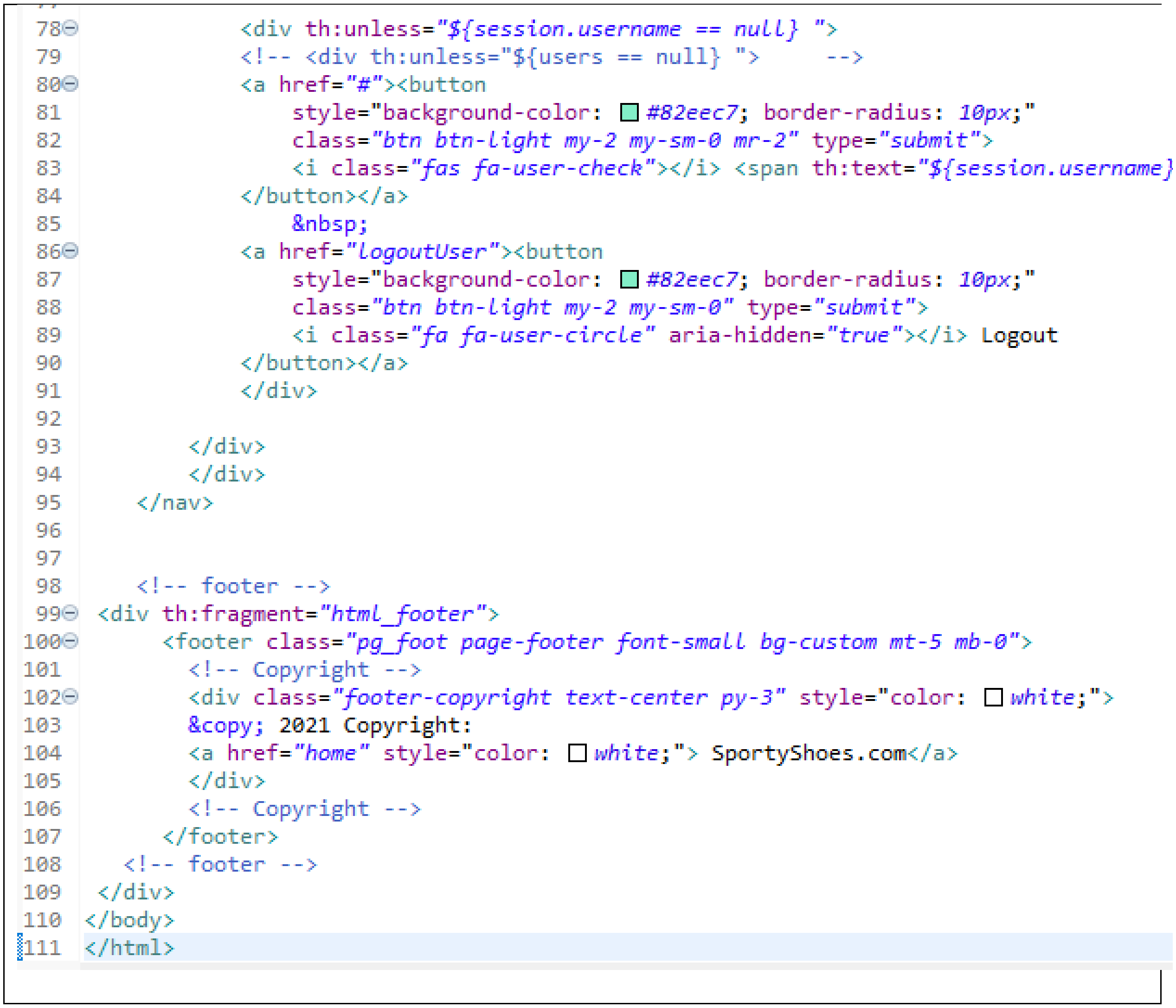
```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:fragment="html_headtag">
<meta charset="ISO-8859-1">
<title>SportyShoes</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.mi
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.
<link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
</head>
<body>

  <!-- code for header -->
    <nav class="navbar navbar-expand-lg navbar-light bg-custom"
        th:fragment="html_header">
        <div class="container-fluid">
            <a class="navbar-brand" href="home1">SportyShoes</a>
            <button class="navbar-toggler" type="button"
                data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
                aria-controls="navbarSupportedContent" aria-expanded="false"
                aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active"
                            aria-current="page" href="home1">
```



```
                        <a class="nav-link active"
                            aria-current="page" href="home1">
                            Home
                        </a>
                    </li>

                    <li class="nav-item dropdown">
                        <a
                        class="nav-link dropdown-toggle" href="#" id="navbarDropdow
                        role="button" data-bs-toggle="dropdown" aria-expanded="fals
                            Dropdown
                        </a>
                        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
                            <li><a class="dropdown-item" href="#">Action</a></li>
                            <li><a class="dropdown-item" href="#">Another action</a
                            <li><hr class="dropdown-divider"></li>
                            <li><a class="dropdown-item" href="#">Something else
                                    here</a></li>
                        </ul>
                    </li>

                </ul>
                <!-- <form  class="d-flex">
                    <input class="form-control me-2" type="search" placeholder="Sea
                        aria-label="Search">
                    <button class="btn btn-light btn-outline-success" type="submit"
                </form> -->
                &nbsp;

            <div th:if="${session.username == null}">
            <!-- <div th:if="${users == null}"> -->
            <a href="signup"><button
                style="background-color: #82eec7; border-radius: 10px;"
                class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
                <i class="fa fa-user-plus" aria-hidden="true"></i> Signup
```

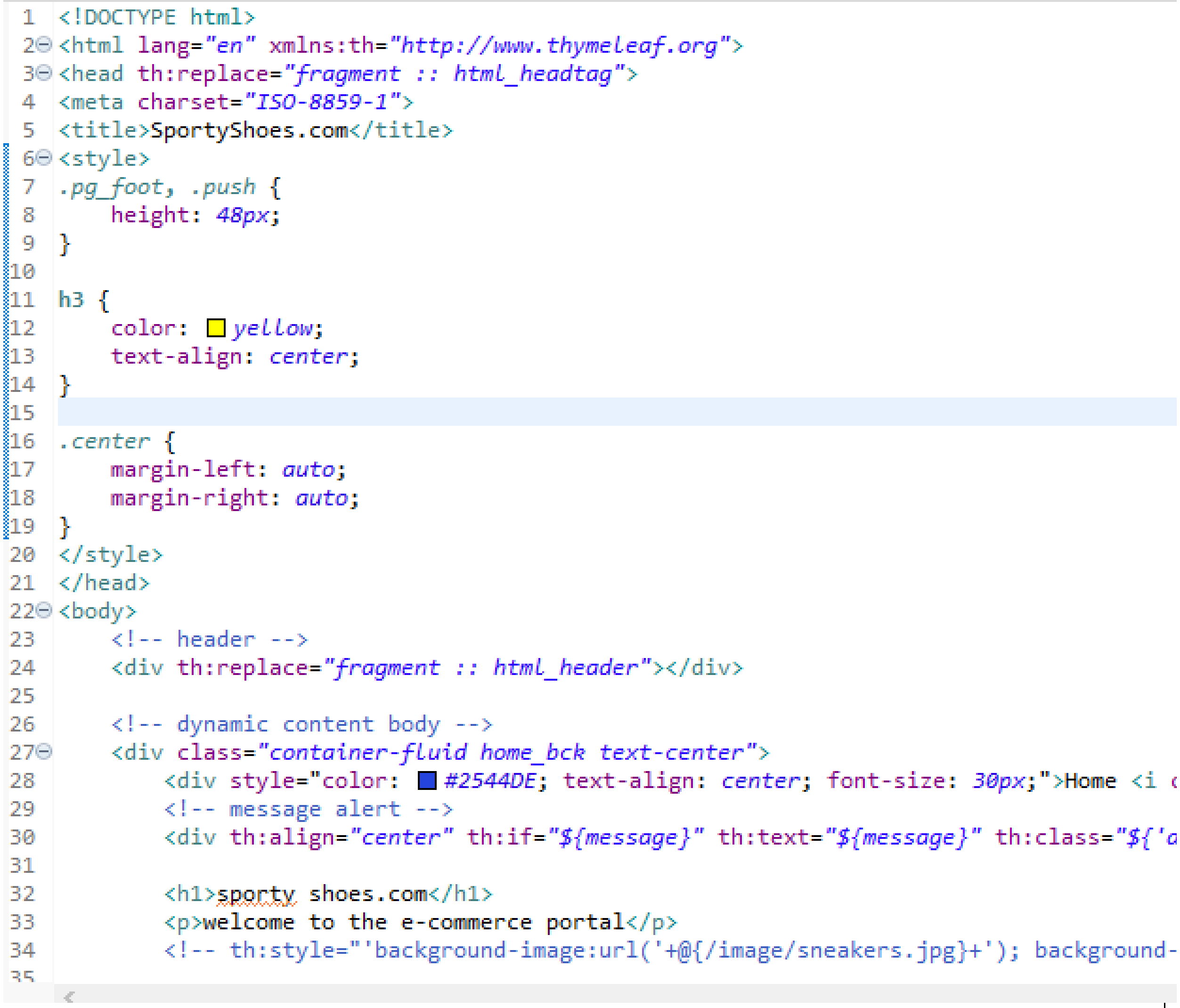


```
           <a href="signup"><button
               style="background-color: #82eec7; border-radius: 10px;"
               class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
               <i class="fa fa-user-plus" aria-hidden="true"></i> Signup
           </button></a>
               &nbsp;
           <a href="login"><button
               style="background-color: #82eec7; border-radius: 10px;"
               class="btn btn-light my-2 my-sm-0" type="submit">
               <i class="fa fa-user-circle" aria-hidden="true"></i> Login
           </button></a>
           </div>

           <div th:unless="${session.username == null} ">
           <!-- <div th:unless="${users == null} ">     -->
           <a href="#"><button
               style="background-color: #82eec7; border-radius: 10px;"
               class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
               <i class="fas fa-user-check"></i> <span th:text="${session.username}
           </button></a>
               &nbsp;
           <a href="logoutUser"><button
               style="background-color: #82eec7; border-radius: 10px;"
               class="btn btn-light my-2 my-sm-0" type="submit">
               <i class="fa fa-user-circle" aria-hidden="true"></i> Logout
           </button></a>
           </div>

       </div>
       </div>
   </nav>


   <!-- footer -->
<div th:fragment="html_footer">
```

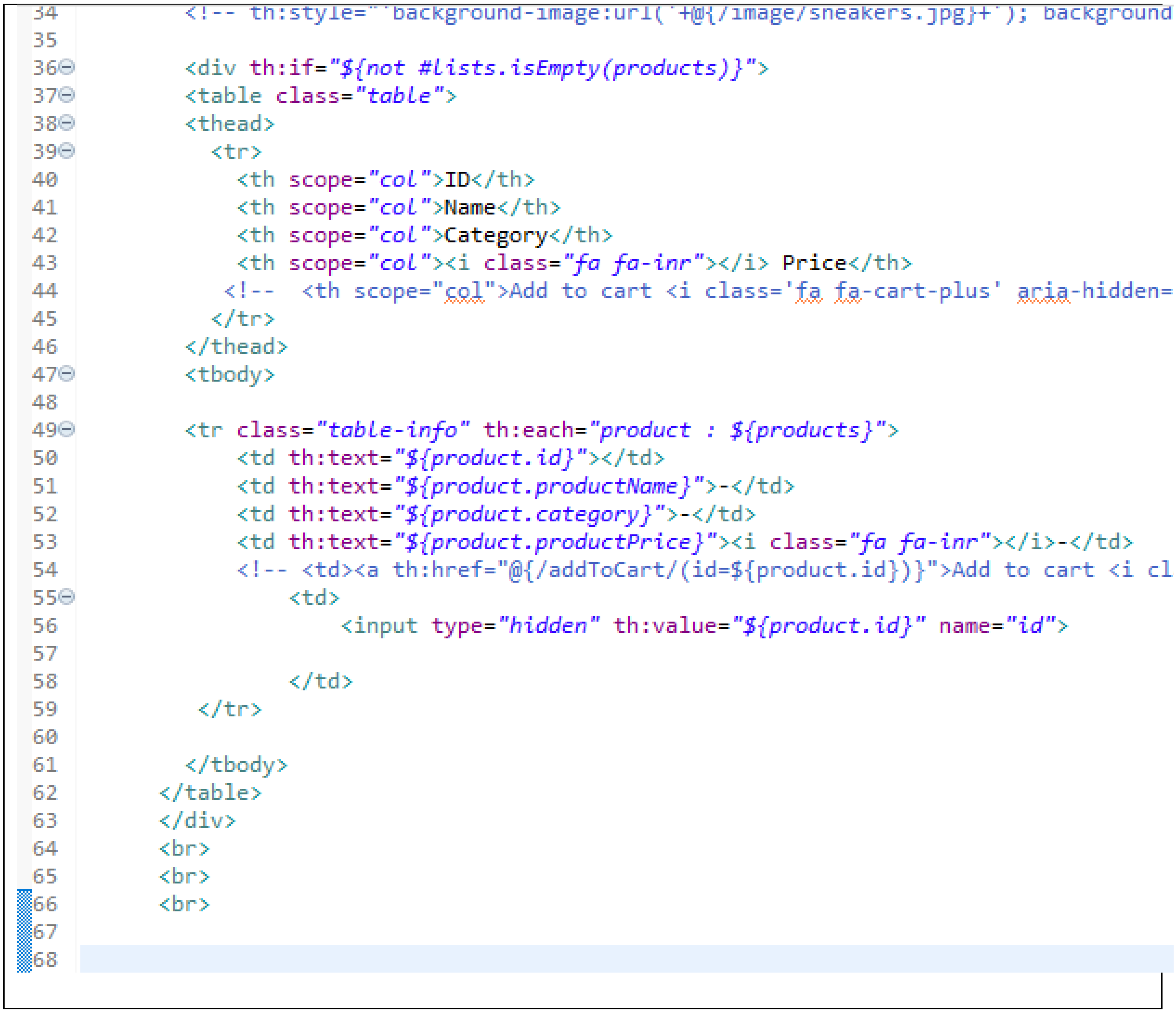


```
            <div th:unless="${session.username == null} ">
            <!-- <div th:unless="${users == null} ">     -->
            <a href="#"><button
                style="background-color: 🟩 #82eec7; border-radius: 10px;"
                class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
                <i class="fas fa-user-check"></i> <span th:text="${session.username}
            </button></a>
                &nbsp;
            <a href="logoutUser"><button
                style="background-color: 🟩 #82eec7; border-radius: 10px;"
                class="btn btn-light my-2 my-sm-0" type="submit">
                <i class="fa fa-user-circle" aria-hidden="true"></i> Logout
            </button></a>
            </div>

        </div>
        </div>
    </nav>


    <!-- footer -->
 <div th:fragment="html_footer">
      <footer class="pg_foot page-footer font-small bg-custom mt-5 mb-0">
        <!-- Copyright -->
        <div class="footer-copyright text-center py-3" style="color: □ white;">
        &copy; 2021 Copyright:
        <a href="home" style="color: □ white;"> SportyShoes.com</a>
        </div>
        <!-- Copyright -->
      </footer>
   <!-- footer -->
 </div>
</body>
</html>
```

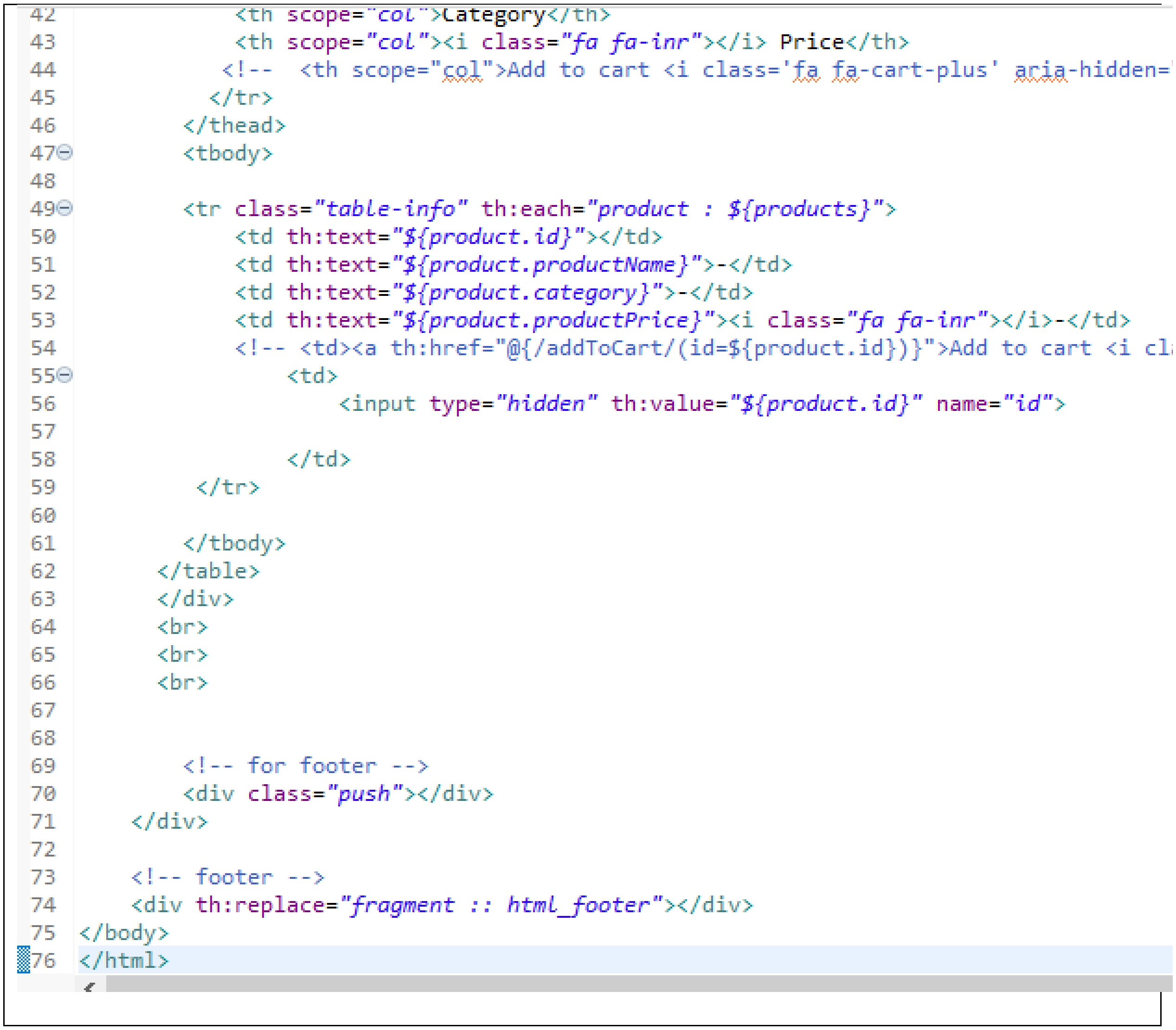
Home.html



```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>SportyShoes.com</title>
<style>
.pg_foot, .push {
    height: 48px;
}

h3 {
    color: yellow;
    text-align: center;
}

.center {
    margin-left: auto;
    margin-right: auto;
}
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <!-- dynamic content body -->
    <div class="container-fluid home_bck text-center">
        <div style="color: #2544DE; text-align: center; font-size: 30px;">Home <i c
        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'a

        <h1>sporty shoes.com</h1>
        <p>welcome to the e-commerce portal</p>
        <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background-
```

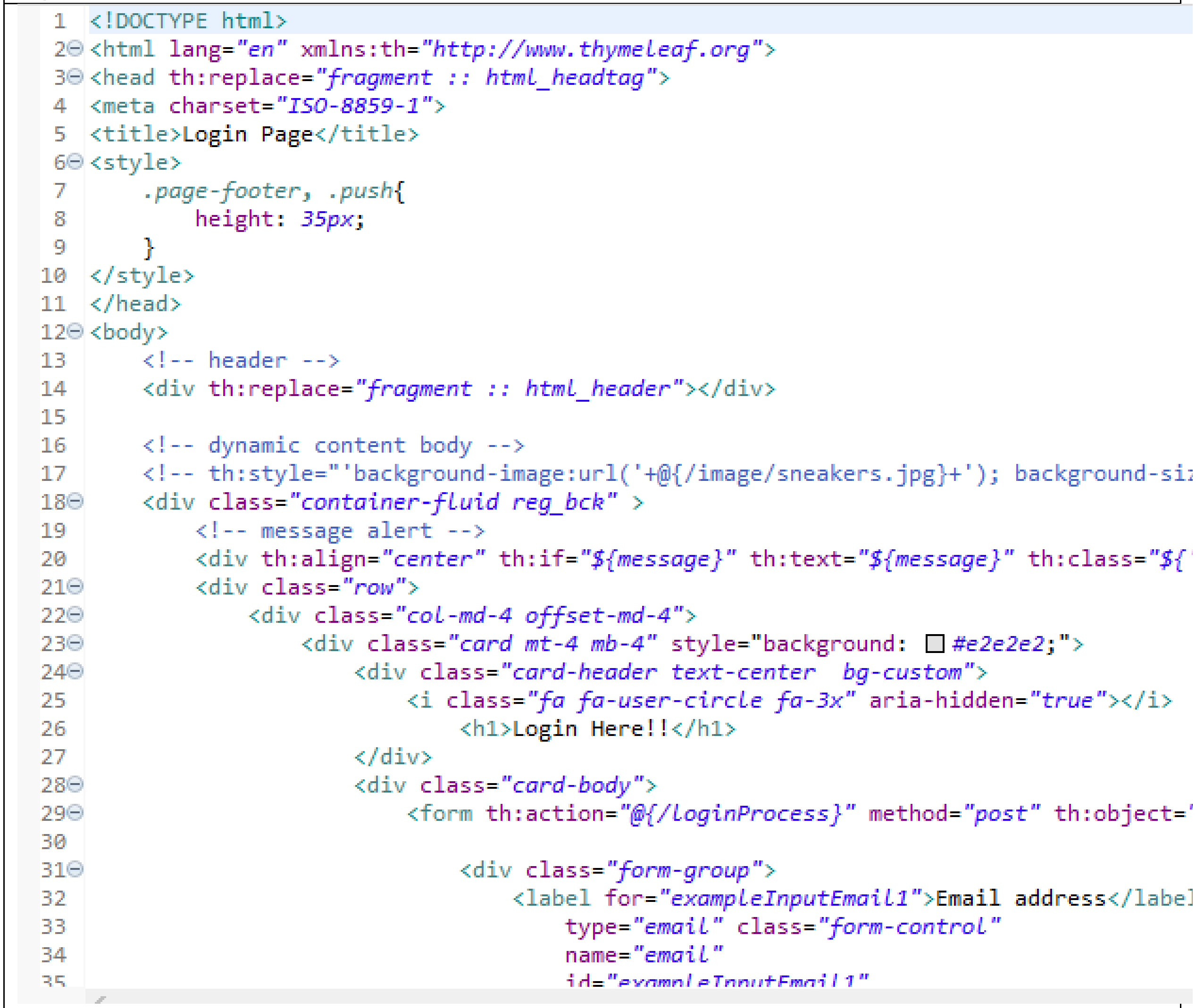


```
          <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background

          <div th:if="${not #lists.isEmpty(products)}">
          <table class="table">
          <thead>
            <tr>
              <th scope="col">ID</th>
              <th scope="col">Name</th>
              <th scope="col">Category</th>
              <th scope="col"><i class="fa fa-inr"></i> Price</th>
             <!--  <th scope="col">Add to cart <i class='fa fa-cart-plus' aria-hidden=
            </tr>
          </thead>
          <tbody>

          <tr class="table-info" th:each="product : ${products}">
              <td th:text="${product.id}"></td>
              <td th:text="${product.productName}">-</td>
              <td th:text="${product.category}">-</td>
              <td th:text="${product.productPrice}"><i class="fa fa-inr"></i>-</td>
              <!-- <td><a th:href="@{/addToCart/(id=${product.id})}">Add to cart <i cl
                  <td>
                      <input type="hidden" th:value="${product.id}" name="id">

                  </td>
           </tr>

          </tbody>
        </table>
        </div>
        <br>
        <br>
        <br>


```

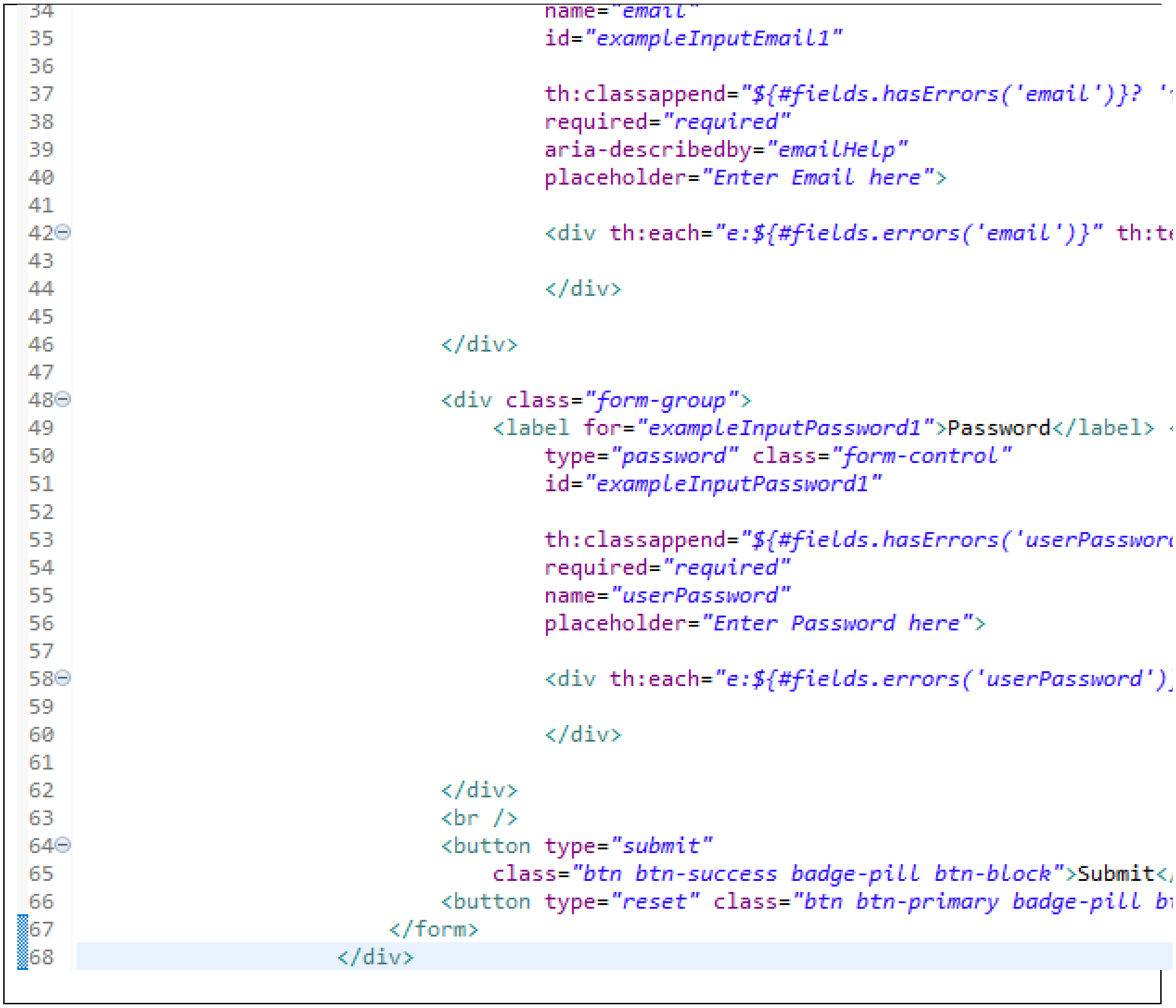


```
            <th scope="col">Category</th>
            <th scope="col"><i class="fa fa-inr"></i> Price</th>
           <!--  <th scope="col">Add to cart <i class='fa fa-cart-plus' aria-hidden='
          </tr>
        </thead>
        <tbody>

        <tr class="table-info" th:each="product : ${products}">
            <td th:text="${product.id}"></td>
            <td th:text="${product.productName}">-</td>
            <td th:text="${product.category}">-</td>
            <td th:text="${product.productPrice}"><i class="fa fa-inr"></i>-</td>
            <!-- <td><a th:href="@{/addToCart/(id=${product.id})}">Add to cart <i cl
                <td>
                    <input type="hidden" th:value="${product.id}" name="id">

                </td>
         </tr>

        </tbody>
      </table>
      </div>
      <br>
      <br>
      <br>


        <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

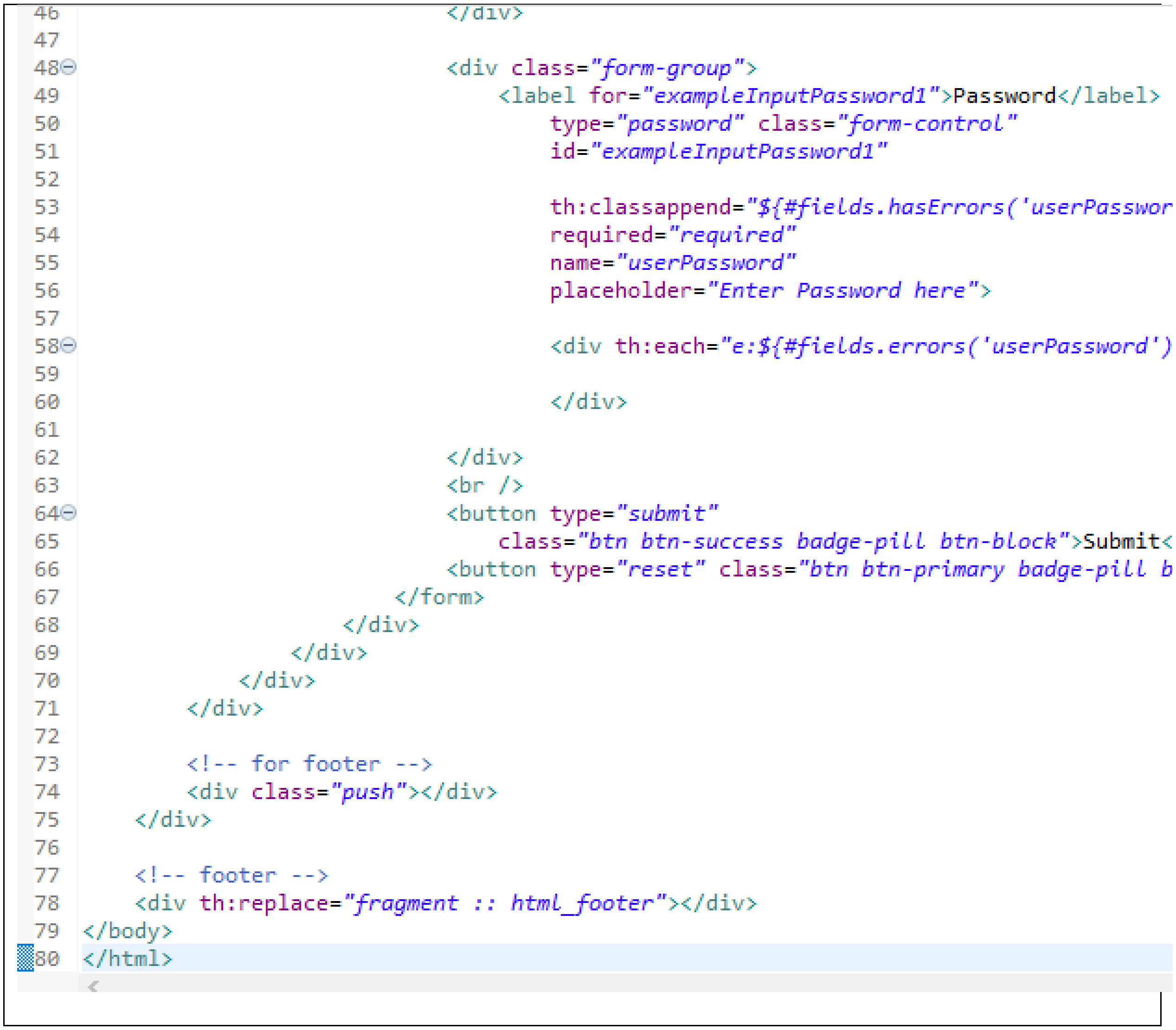
login.html



```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Login Page</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <!-- dynamic content body -->
    <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background-si
    <div class="container-fluid reg_bck" >
        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'
        <div class="row">
            <div class="col-md-4 offset-md-4">
                <div class="card mt-4 mb-4" style="background: □ #e2e2e2;">
                    <div class="card-header text-center  bg-custom">
                        <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
                            <h1>Login Here!!</h1>
                    </div>
                    <div class="card-body">
                        <form th:action="@{/loginProcess}" method="post" th:object='

                            <div class="form-group">
                                <label for="exampleInputEmail1">Email address</label
                                    type="email" class="form-control"
                                    name="email"
                                    id="exampleInputEmail1"
```

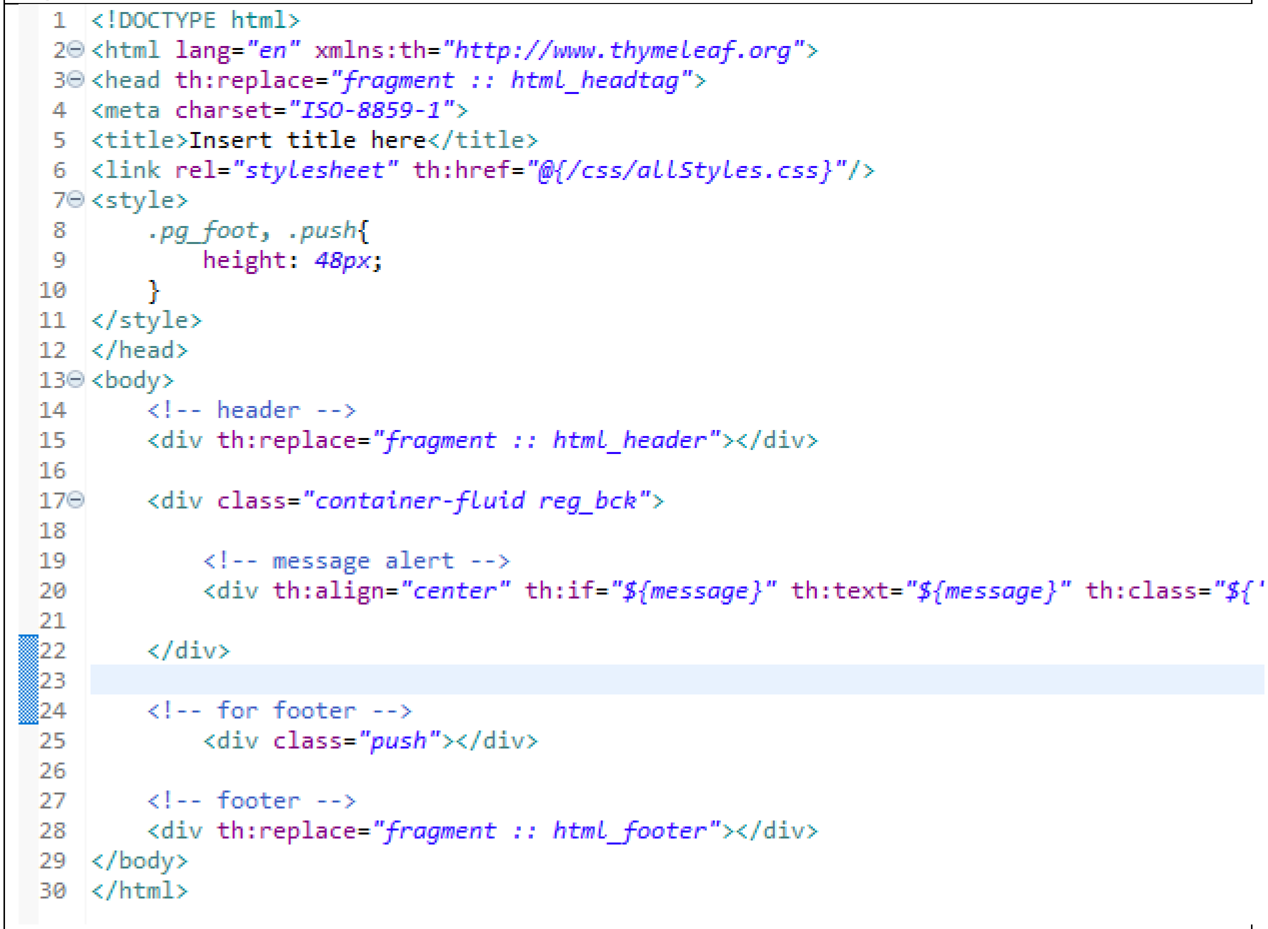


```
                                    name="email"
                                    id="exampleInputEmail1"

                                    th:classappend="${#fields.hasErrors('email')}? '
                                    required="required"
                                    aria-describedby="emailHelp"
                                    placeholder="Enter Email here">

                                    <div th:each="e:${#fields.errors('email')}" th:te

                                    </div>

                            </div>

                            <div class="form-group">
                                <label for="exampleInputPassword1">Password</label> <
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    th:classappend="${#fields.hasErrors('userPasswor
                                    required="required"
                                    name="userPassword"
                                    placeholder="Enter Password here">

                                    <div th:each="e:${#fields.errors('userPassword')

                                    </div>

                            </div>
                            <br />
                            <button type="submit"
                                class="btn btn-success badge-pill btn-block">Submit<
                            <button type="reset" class="btn btn-primary badge-pill b
                        </form>
                    </div>
```



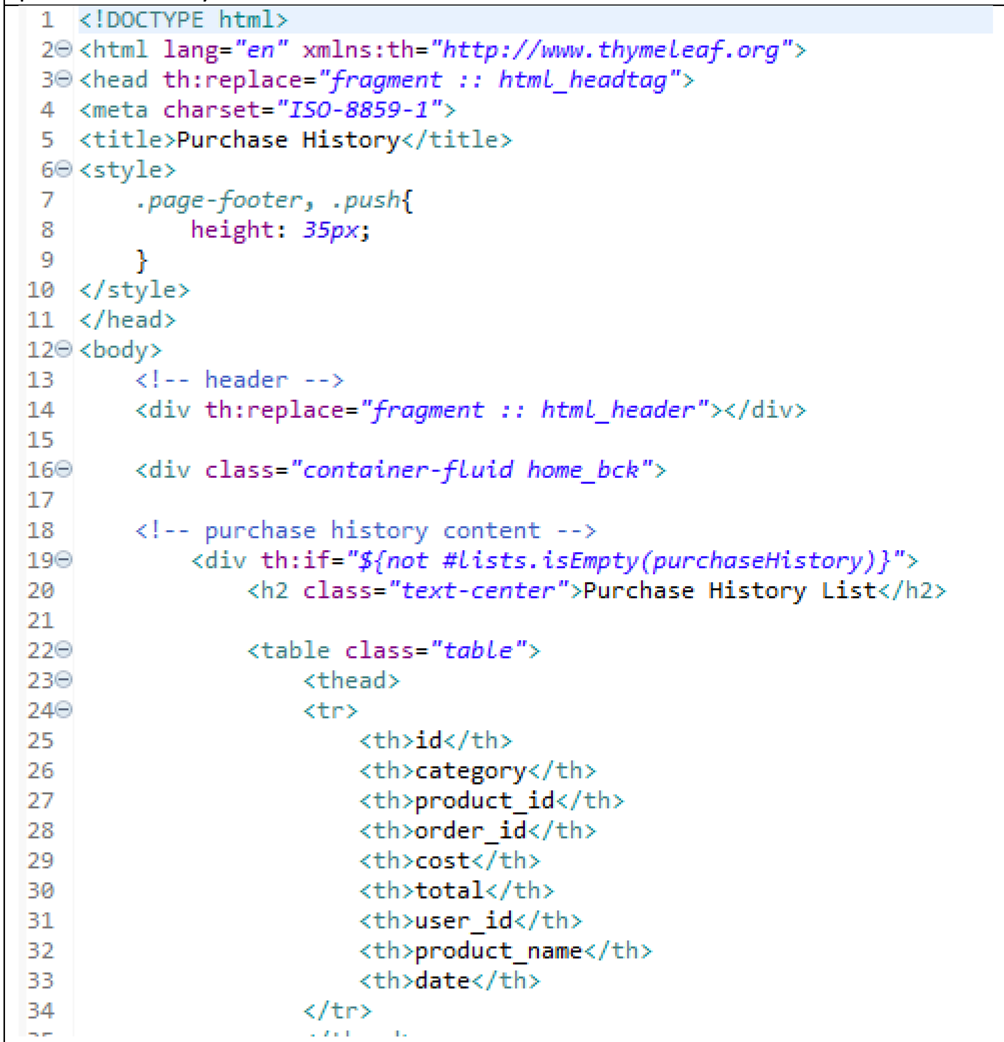
```
                            </div>

                            <div class="form-group">
                                <label for="exampleInputPassword1">Password</label>
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    th:classappend="${#fields.hasErrors('userPasswor
                                    required="required"
                                    name="userPassword"
                                    placeholder="Enter Password here">

                                    <div th:each="e:${#fields.errors('userPassword')

                                    </div>

                            </div>
                            <br />
                            <button type="submit"
                                class="btn btn-success badge-pill btn-block">Submit<
                            <button type="reset" class="btn btn-primary badge-pill b
                        </form>
                    </div>
                </div>
            </div>
        </div>

        <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

logout.html

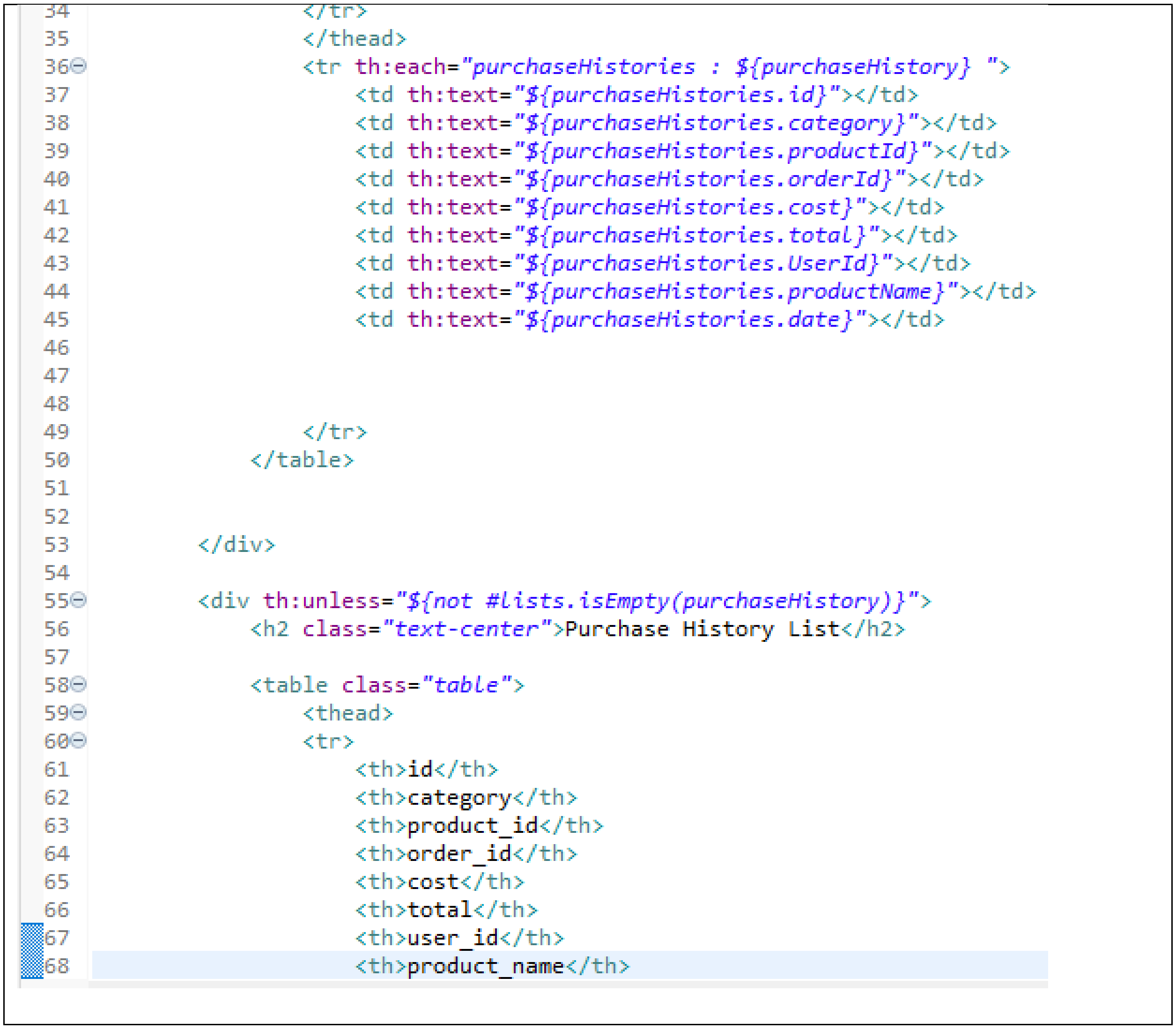


```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
<style>
    .pg_foot, .push{
        height: 48px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid reg_bck">

        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'

    </div>

    <!-- for footer -->
        <div class="push"></div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

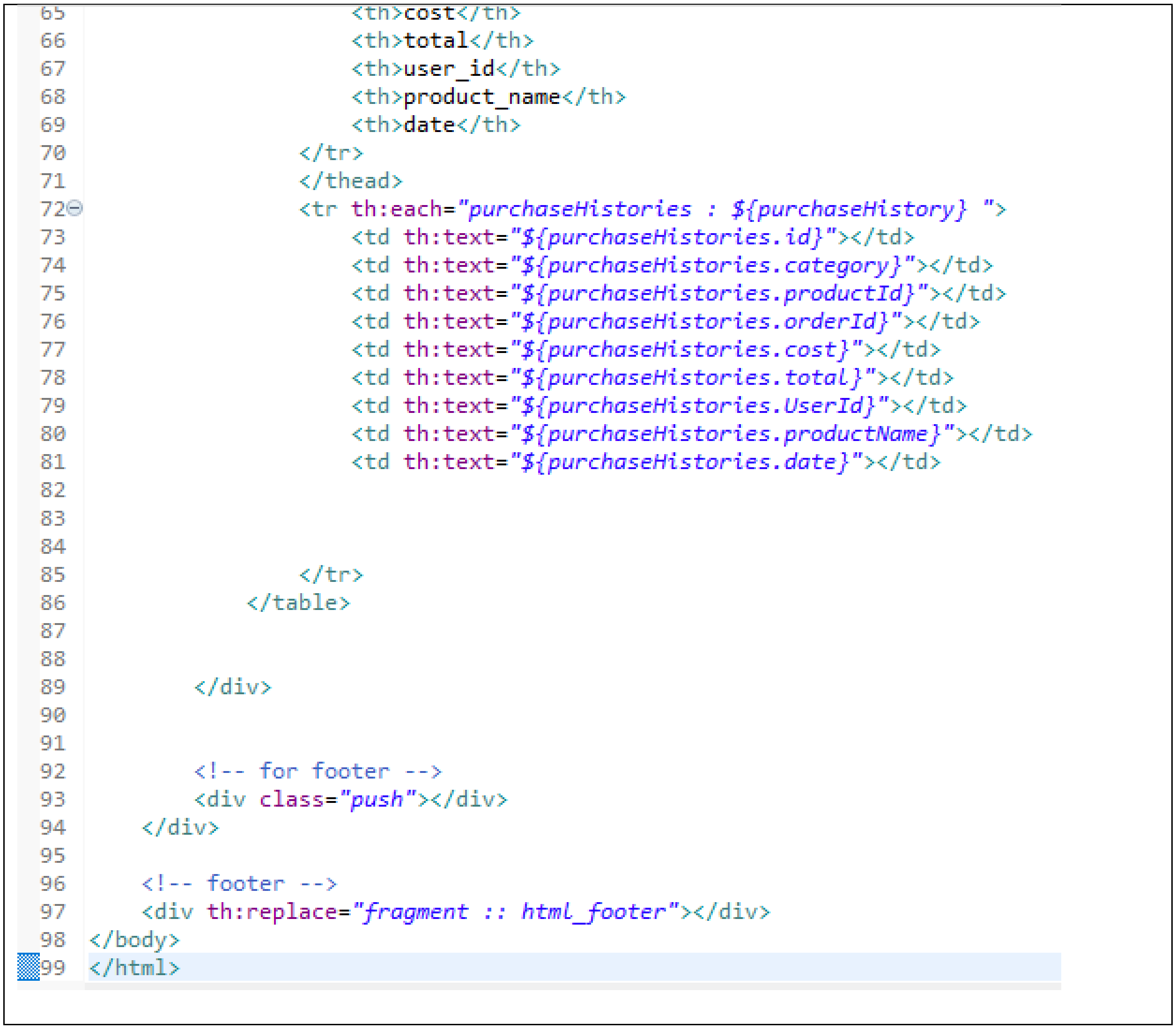
purchaseHistory.html



```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Purchase History</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid home_bck">

    <!-- purchase history content -->
        <div th:if="${not #lists.isEmpty(purchaseHistory)}">
            <h2 class="text-center">Purchase History List</h2>

            <table class="table">
                <thead>
                <tr>
                    <th>id</th>
                    <th>category</th>
                    <th>product_id</th>
                    <th>order_id</th>
                    <th>cost</th>
                    <th>total</th>
                    <th>user_id</th>
                    <th>product_name</th>
                    <th>date</th>
                </tr>
```

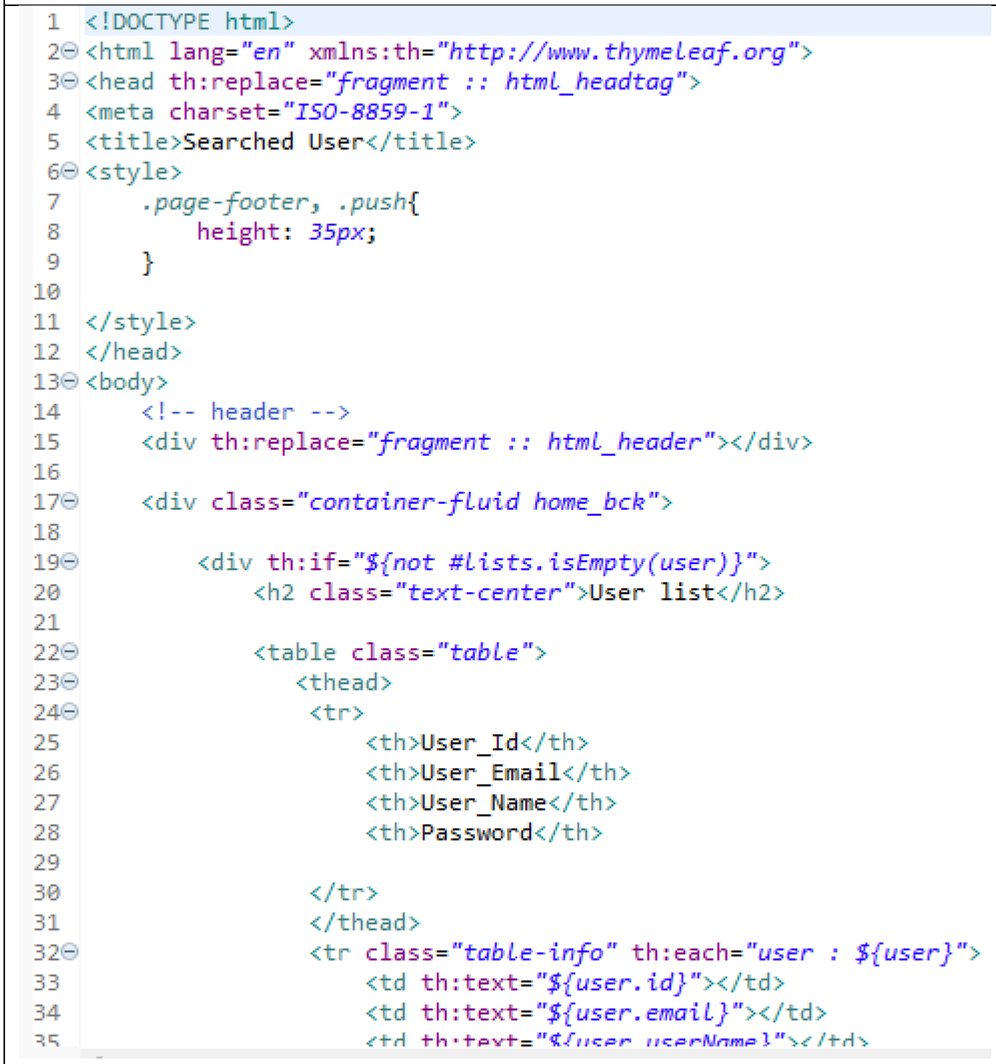


```
                </tr>
                </thead>
                <tr th:each="purchaseHistories : ${purchaseHistory} ">
                    <td th:text="${purchaseHistories.id}"></td>
                    <td th:text="${purchaseHistories.category}"></td>
                    <td th:text="${purchaseHistories.productId}"></td>
                    <td th:text="${purchaseHistories.orderId}"></td>
                    <td th:text="${purchaseHistories.cost}"></td>
                    <td th:text="${purchaseHistories.total}"></td>
                    <td th:text="${purchaseHistories.UserId}"></td>
                    <td th:text="${purchaseHistories.productName}"></td>
                    <td th:text="${purchaseHistories.date}"></td>



                </tr>
            </table>


        </div>

        <div th:unless="${not #lists.isEmpty(purchaseHistory)}">
            <h2 class="text-center">Purchase History List</h2>

            <table class="table">
                <thead>
                <tr>
                    <th>id</th>
                    <th>category</th>
                    <th>product_id</th>
                    <th>order_id</th>
                    <th>cost</th>
                    <th>total</th>
                    <th>user_id</th>
                    <th>product_name</th>
```

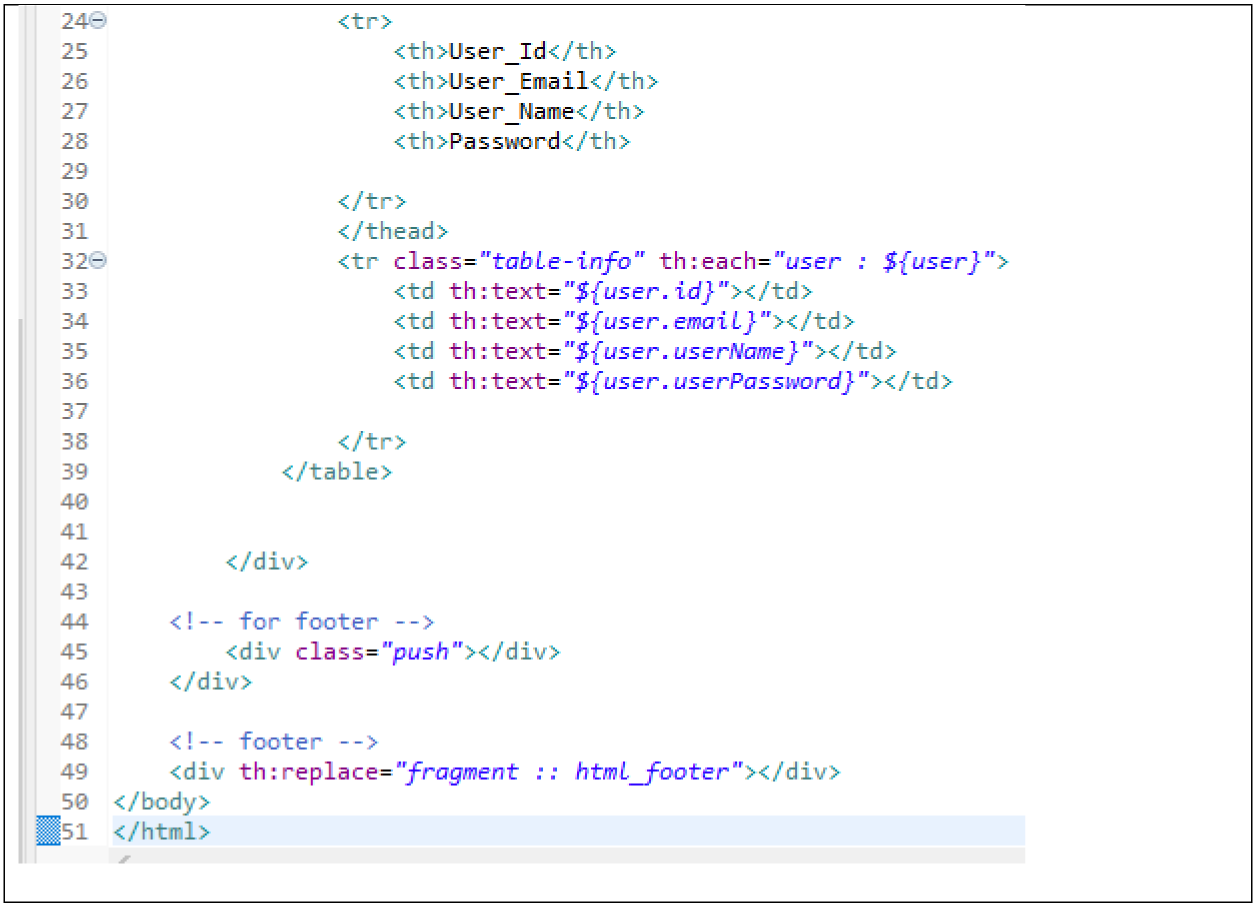


```html
                    <th>cost</th>
                    <th>total</th>
                    <th>user_id</th>
                    <th>product_name</th>
                    <th>date</th>
                </tr>
                </thead>
                <tr th:each="purchaseHistories : ${purchaseHistory} ">
                    <td th:text="${purchaseHistories.id}"></td>
                    <td th:text="${purchaseHistories.category}"></td>
                    <td th:text="${purchaseHistories.productId}"></td>
                    <td th:text="${purchaseHistories.orderId}"></td>
                    <td th:text="${purchaseHistories.cost}"></td>
                    <td th:text="${purchaseHistories.total}"></td>
                    <td th:text="${purchaseHistories.UserId}"></td>
                    <td th:text="${purchaseHistories.productName}"></td>
                    <td th:text="${purchaseHistories.date}"></td>



                </tr>
            </table>


        </div>


        <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

searchedUser.html

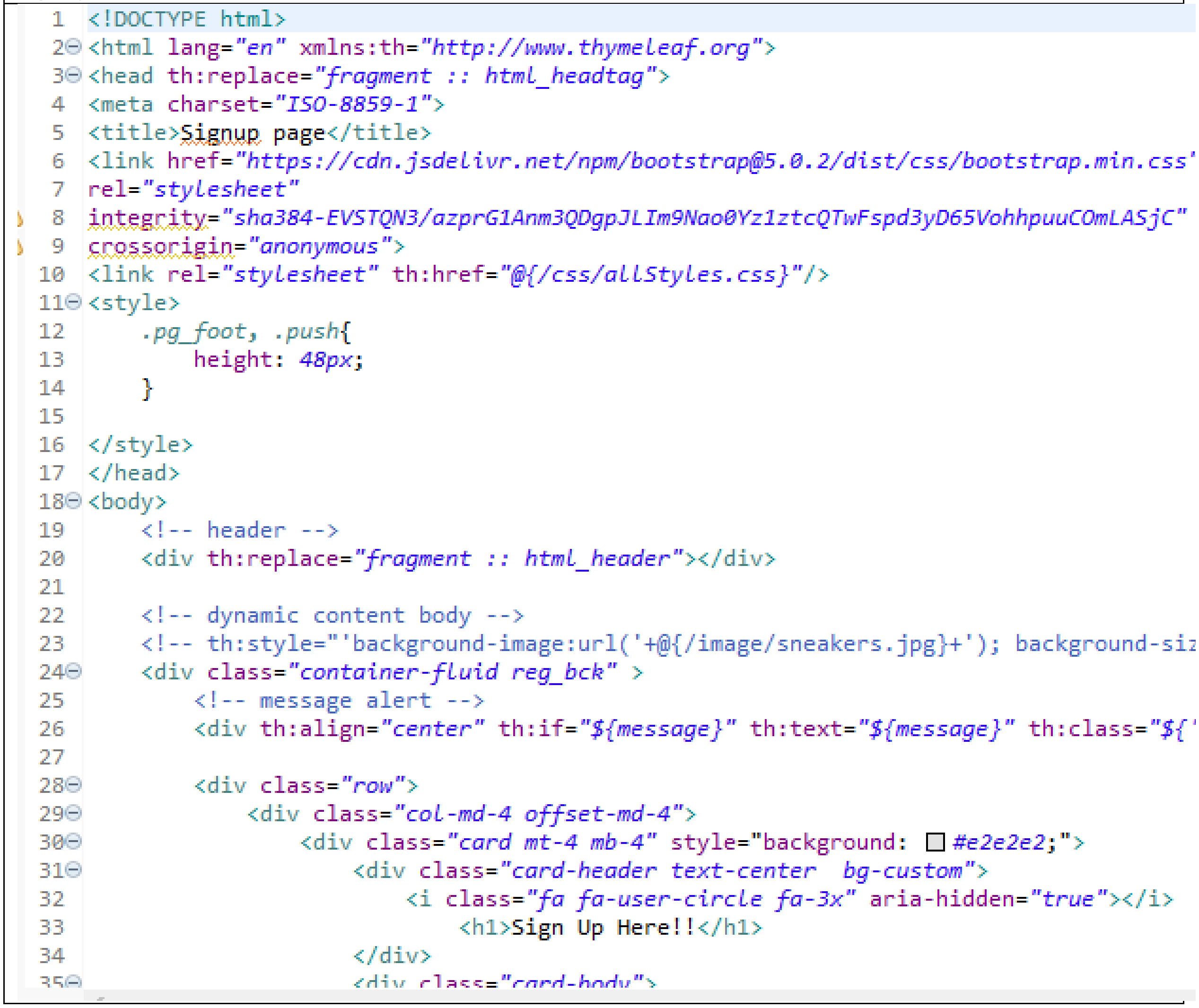


```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Searched User</title>
<style>
    .page-footer, .push{
        height: 35px;
    }

</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid home_bck">

        <div th:if="${not #lists.isEmpty(user)}">
            <h2 class="text-center">User list</h2>

            <table class="table">
               <thead>
                <tr>
                    <th>User_Id</th>
                    <th>User_Email</th>
                    <th>User_Name</th>
                    <th>Password</th>

                </tr>
                </thead>
                <tr class="table-info" th:each="user : ${user}">
                    <td th:text="${user.id}"></td>
                    <td th:text="${user.email}"></td>
                    <td th:text="${user.userName}"></td>
```

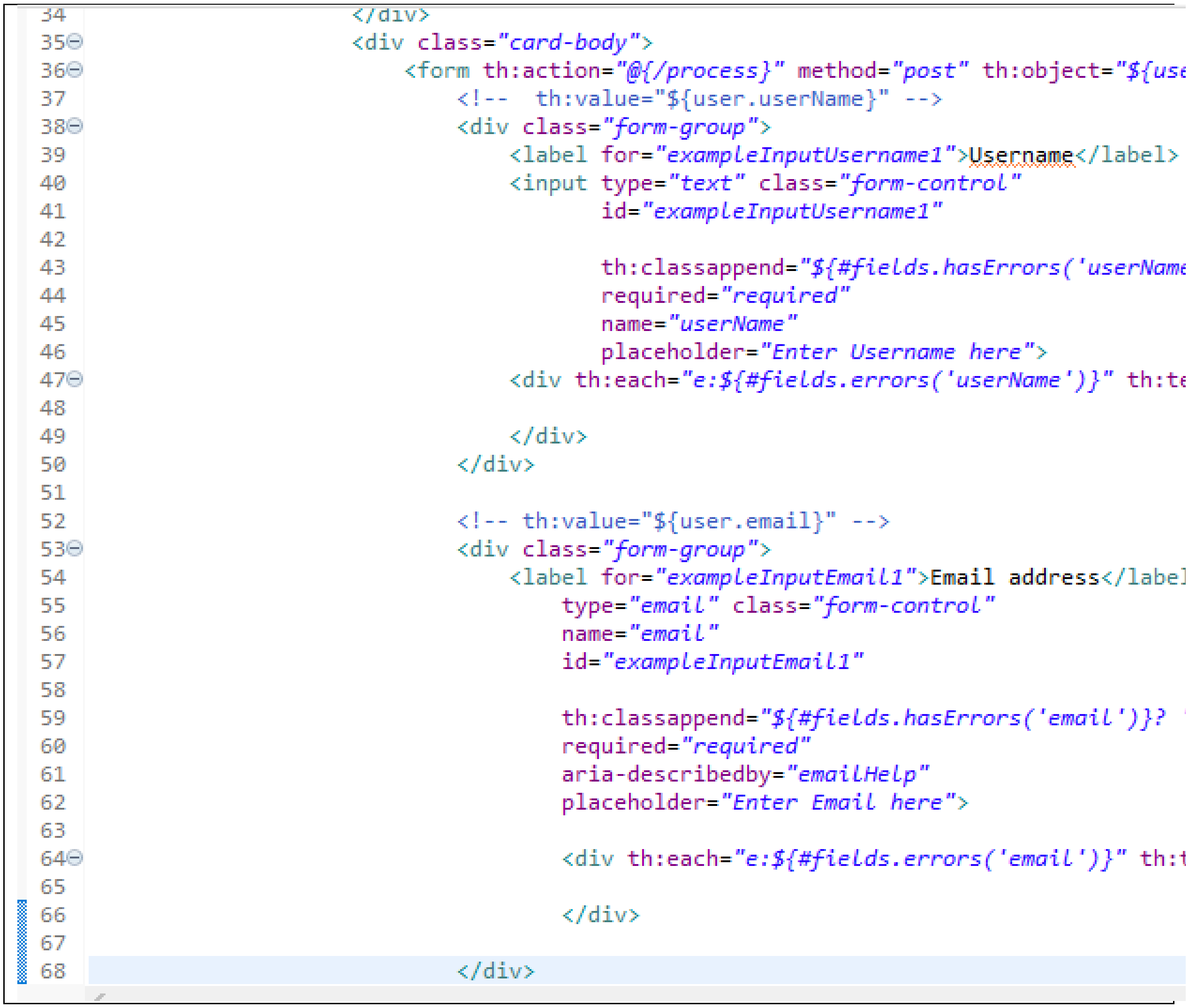


```
                <tr>
                    <th>User_Id</th>
                    <th>User_Email</th>
                    <th>User_Name</th>
                    <th>Password</th>

                </tr>
                </thead>
                <tr class="table-info" th:each="user : ${user}">
                    <td th:text="${user.id}"></td>
                    <td th:text="${user.email}"></td>
                    <td th:text="${user.userName}"></td>
                    <td th:text="${user.userPassword}"></td>

                </tr>
            </table>


        </div>

    <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

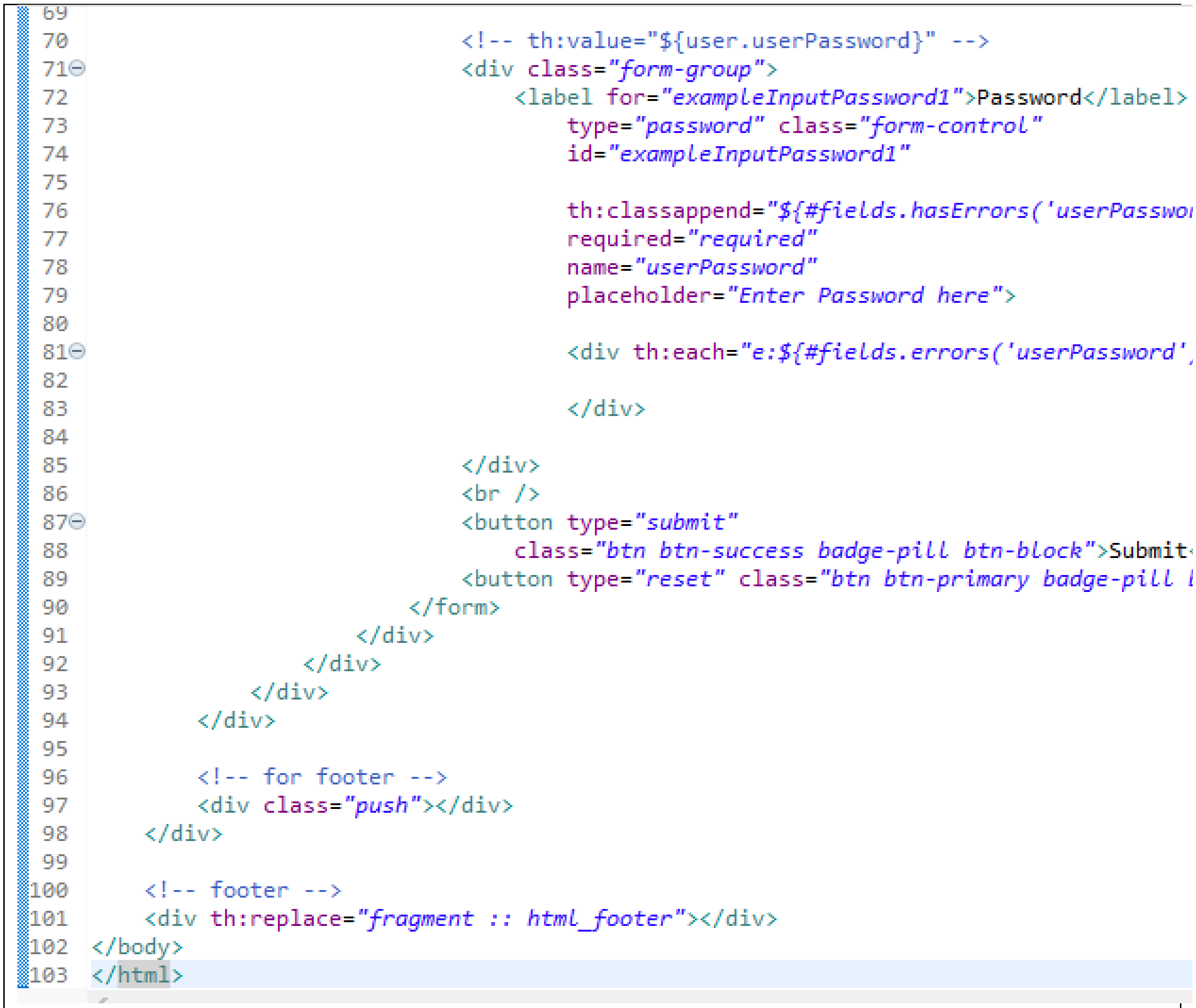
signup.html



```html
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Signup page</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css'
rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
<link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
<style>
    .pg_foot, .push{
        height: 48px;
    }

</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <!-- dynamic content body -->
    <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background-siz
    <div class="container-fluid reg_bck" >
        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'

        <div class="row">
            <div class="col-md-4 offset-md-4">
                <div class="card mt-4 mb-4" style="background: □ #e2e2e2;">
                    <div class="card-header text-center  bg-custom">
                        <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
                            <h1>Sign Up Here!!</h1>
                    </div>
                    <div class="card-body">
```

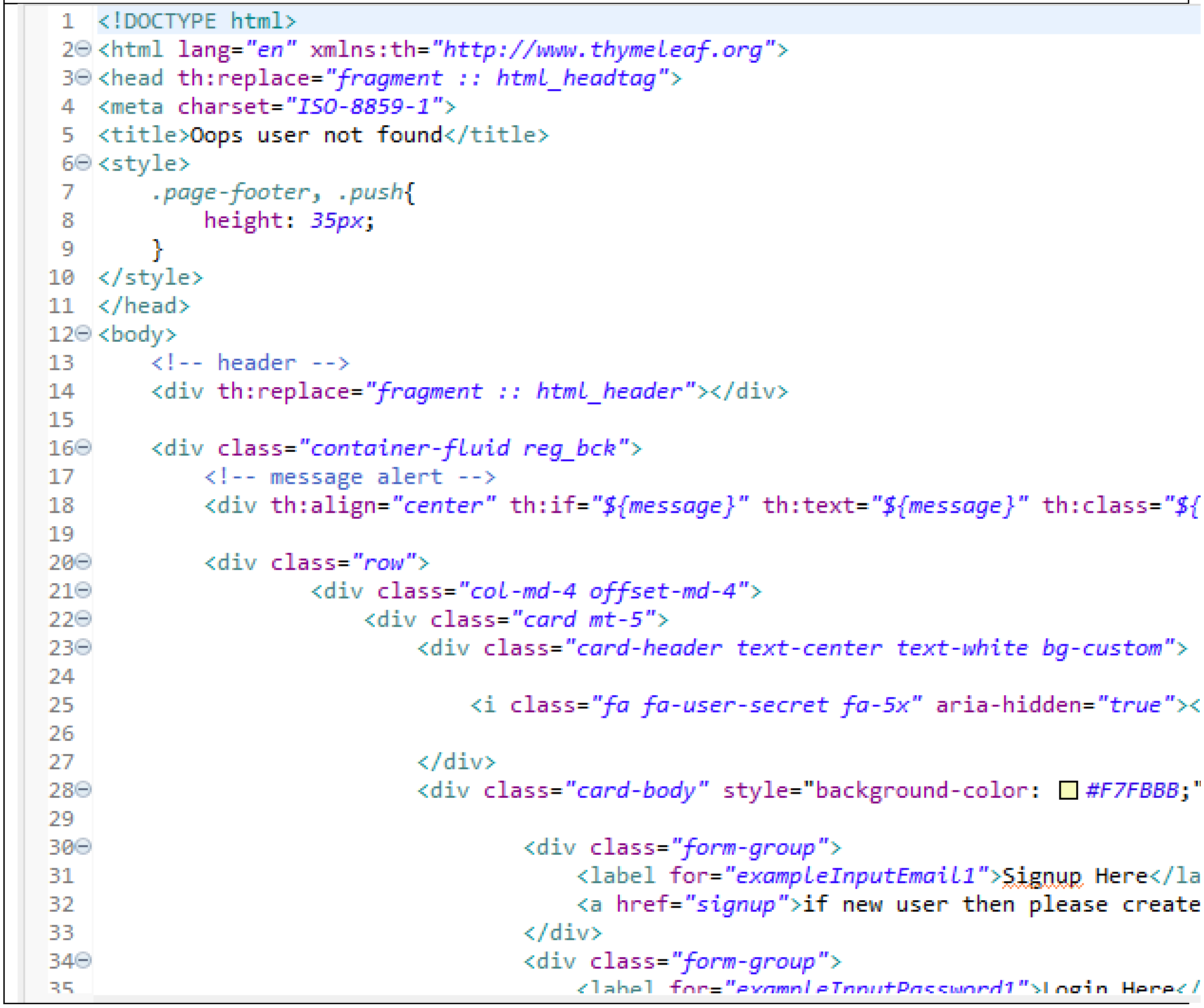


```
                    </div>
                    <div class="card-body">
                        <form th:action="@{/process}" method="post" th:object="${use
                            <!--  th:value="${user.userName}" -->
                            <div class="form-group">
                                <label for="exampleInputUsername1">Username</label>
                                <input type="text" class="form-control"
                                       id="exampleInputUsername1"

                                       th:classappend="${#fields.hasErrors('userName
                                       required="required"
                                       name="userName"
                                       placeholder="Enter Username here">
                                <div th:each="e:${#fields.errors('userName')}" th:te

                                </div>
                            </div>

                            <!-- th:value="${user.email}" -->
                            <div class="form-group">
                                <label for="exampleInputEmail1">Email address</label
                                    type="email" class="form-control"
                                    name="email"
                                    id="exampleInputEmail1"

                                    th:classappend="${#fields.hasErrors('email')}? '
                                    required="required"
                                    aria-describedby="emailHelp"
                                    placeholder="Enter Email here">

                                    <div th:each="e:${#fields.errors('email')}" th:t

                                    </div>

                            </div>
```

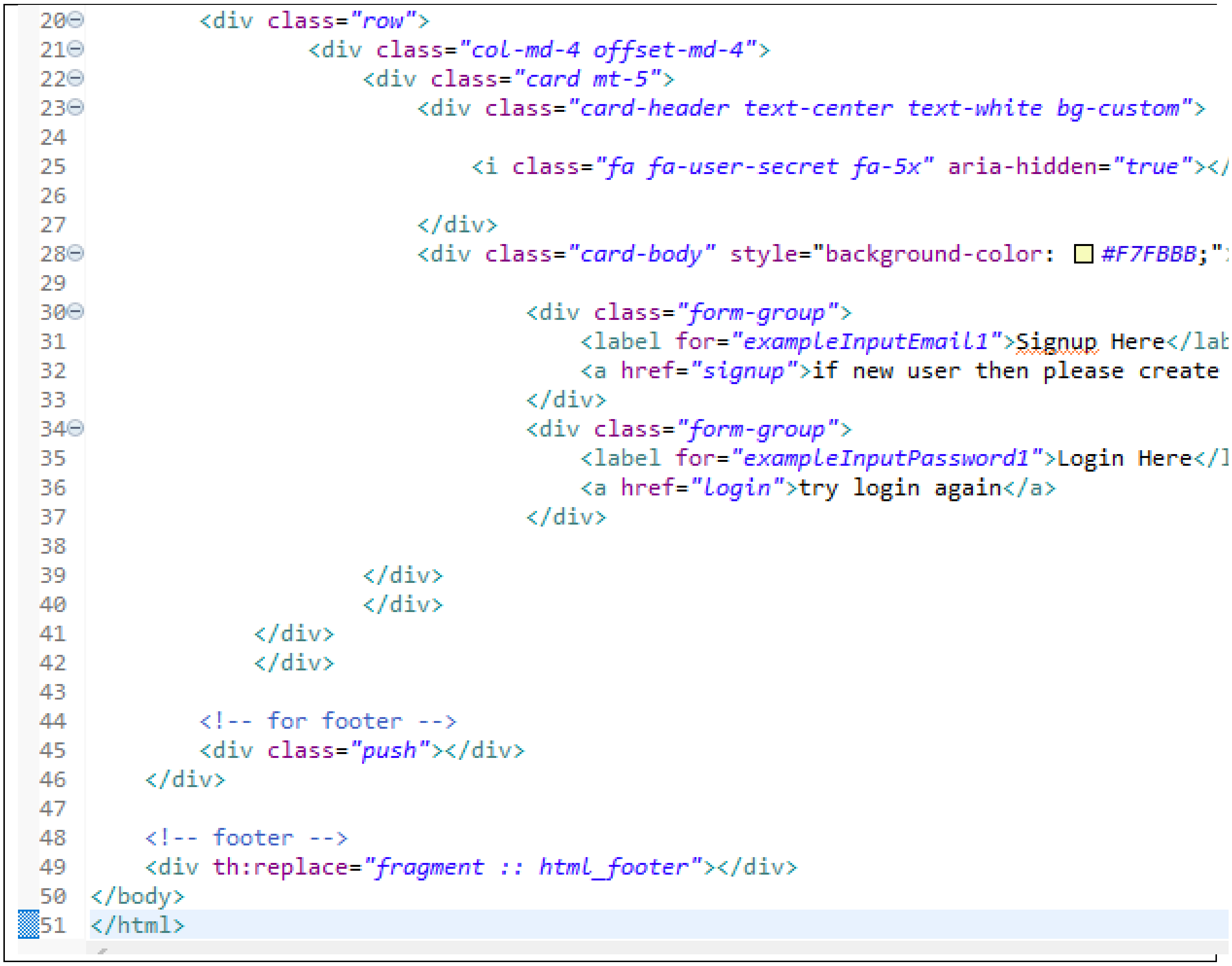


```
                            <!-- th:value="${user.userPassword}" -->
                            <div class="form-group">
                                <label for="exampleInputPassword1">Password</label>
                                    type="password" class="form-control"
                                    id="exampleInputPassword1"

                                    th:classappend="${#fields.hasErrors('userPasswo
                                    required="required"
                                    name="userPassword"
                                    placeholder="Enter Password here">

                                    <div th:each="e:${#fields.errors('userPassword',

                                    </div>

                            </div>
                            <br />
                            <button type="submit"
                                class="btn btn-success badge-pill btn-block">Submit
                            <button type="reset" class="btn btn-primary badge-pill
                        </form>
                    </div>
                </div>
            </div>
        </div>

        <!-- for footer -->
        <div class="push"></div>
    </div>

    <!-- footer -->
    <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

userNotFound.html



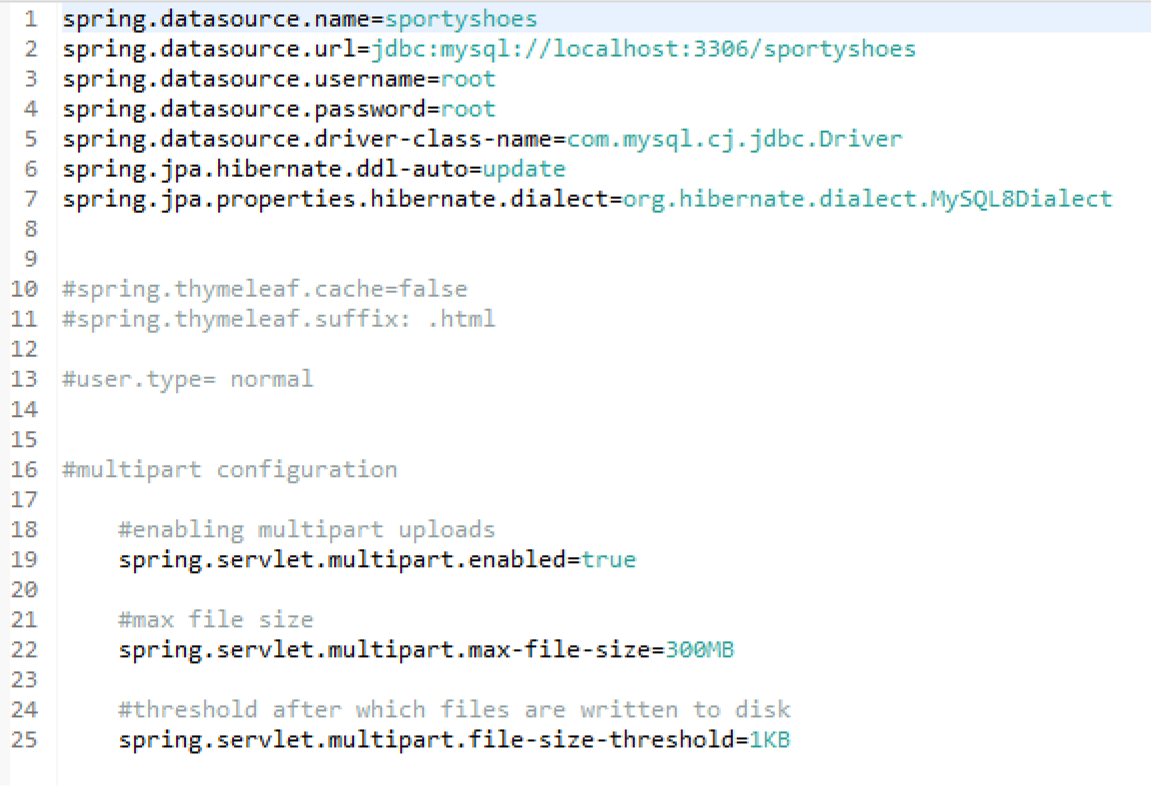
```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:replace="fragment :: html_headtag">
<meta charset="ISO-8859-1">
<title>Oops user not found</title>
<style>
    .page-footer, .push{
        height: 35px;
    }
</style>
</head>
<body>
    <!-- header -->
    <div th:replace="fragment :: html_header"></div>

    <div class="container-fluid reg_bck">
        <!-- message alert -->
        <div th:align="center" th:if="${message}" th:text="${message}" th:class="${

        <div class="row">
                <div class="col-md-4 offset-md-4">
                    <div class="card mt-5">
                        <div class="card-header text-center text-white bg-custom">

                            <i class="fa fa-user-secret fa-5x" aria-hidden="true"><

                        </div>
                        <div class="card-body" style="background-color: □ #F7FBBB;"

                                <div class="form-group">
                                    <label for="exampleInputEmail1">Signup Here</la
                                    <a href="signup">if new user then please create
                                </div>
                                <div class="form-group">
                                    <label for="exampleInputPassword1">Login Here</
```



```
    <div class="row">
        <div class="col-md-4 offset-md-4">
          <div class="card mt-5">
            <div class="card-header text-center text-white bg-custom">

              <i class="fa fa-user-secret fa-5x" aria-hidden="true"></

            </div>
            <div class="card-body" style="background-color: □ #F7FBBB;"

                <div class="form-group">
                  <label for="exampleInputEmail1">Signup Here</lab
                  <a href="signup">if new user then please create
                </div>
                <div class="form-group">
                  <label for="exampleInputPassword1">Login Here</l
                  <a href="login">try login again</a>
                </div>

          </div>
          </div>
      </div>
      </div>

    <!-- for footer -->
    <div class="push"></div>
  </div>

  <!-- footer -->
  <div th:replace="fragment :: html_footer"></div>
</body>
</html>
```

- ***APPLICATION.PROPERTIES***

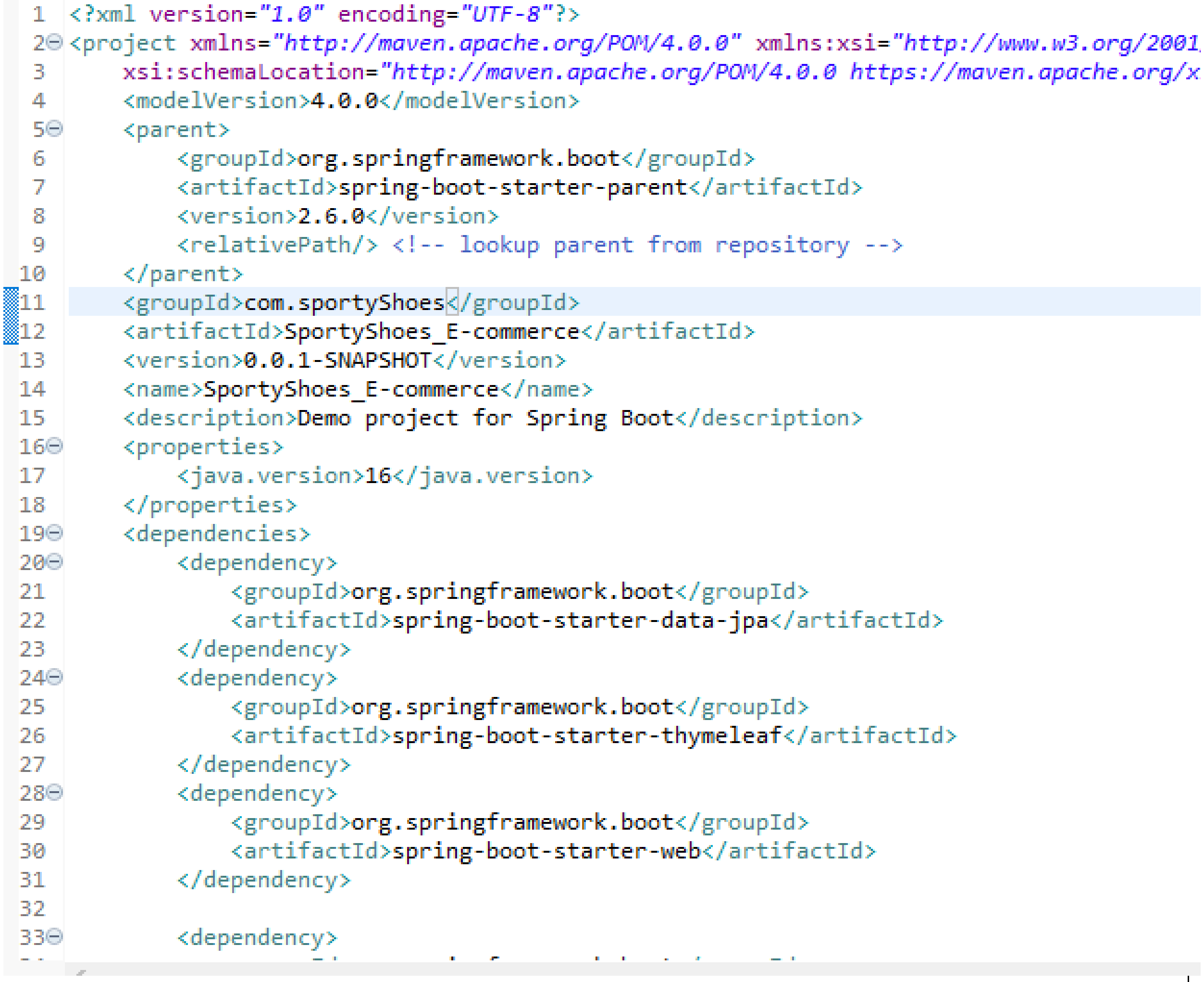
application.properties



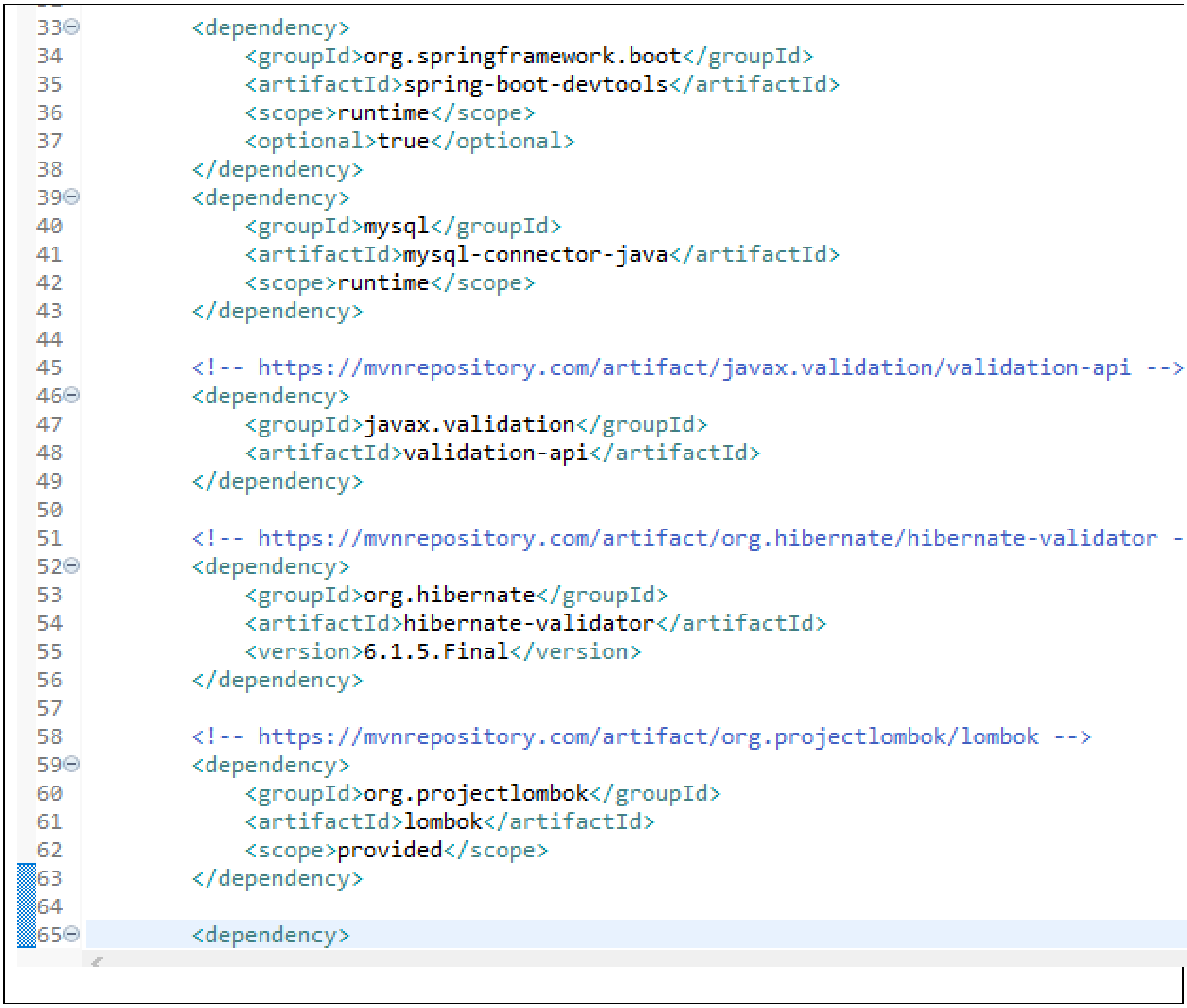
```
spring.datasource.name=sportyshoes
spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoes
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect


#spring.thymeleaf.cache=false
#spring.thymeleaf.suffix: .html

#user.type= normal


#multipart configuration

    #enabling multipart uploads
    spring.servlet.multipart.enabled=true

    #max file size
    spring.servlet.multipart.max-file-size=300MB

    #threshold after which files are written to disk
    spring.servlet.multipart.file-size-threshold=1KB
```

- ***POM.XML(MAVEN DEPENDENCY FILE)***

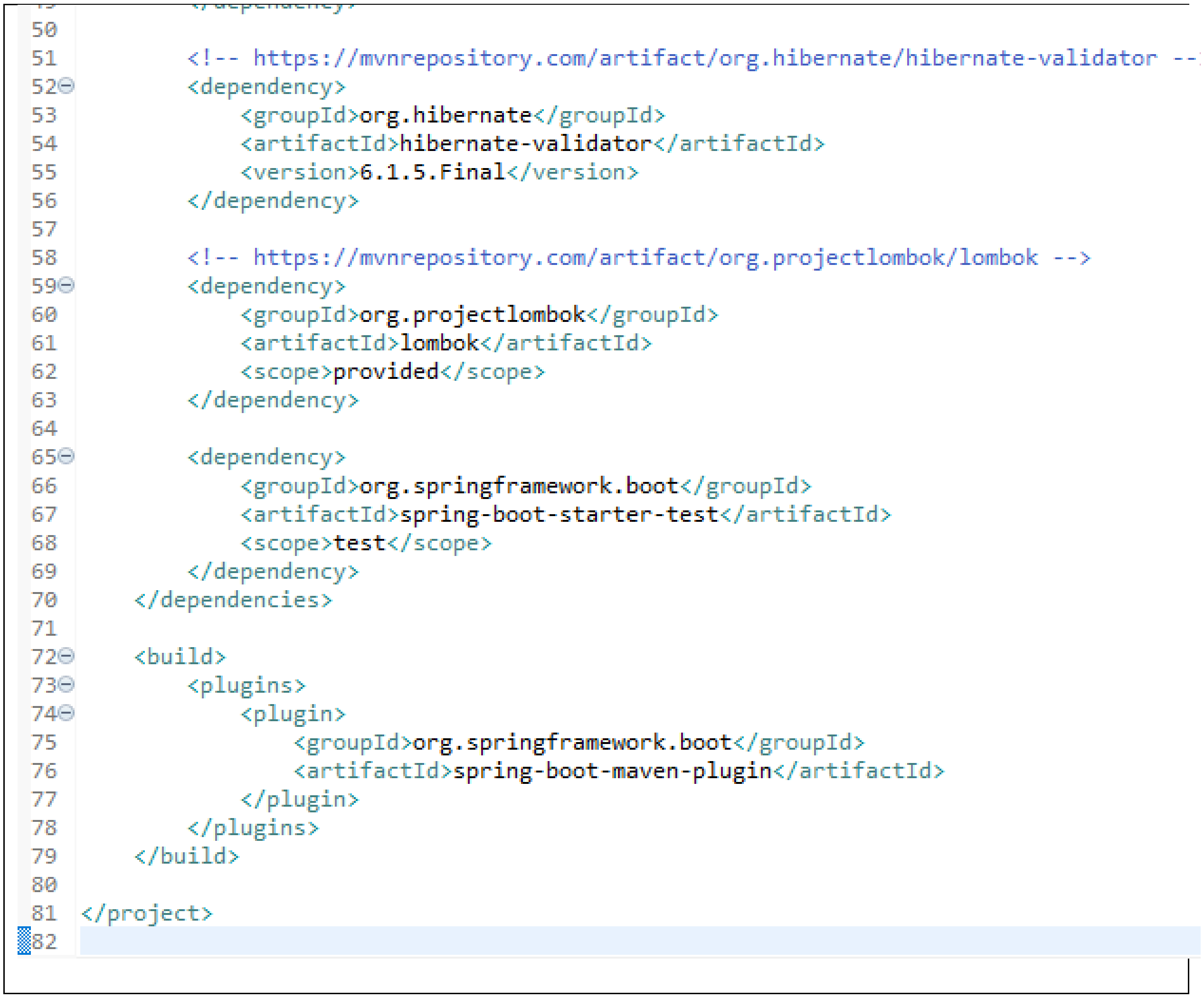
pom.xml



```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/x
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.6.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.sportyShoes</groupId>
    <artifactId>SportyShoes_E-commerce</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>SportyShoes_E-commerce</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>16</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-thymeleaf</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
```



```xml
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>

        <!-- https://mvnrepository.com/artifact/javax.validation/validation-api -->
        <dependency>
            <groupId>javax.validation</groupId>
            <artifactId>validation-api</artifactId>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator --
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-validator</artifactId>
            <version>6.1.5.Final</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <scope>provided</scope>
        </dependency>

        <dependency>
```



```
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator --
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-validator</artifactId>
            <version>6.1.5.Final</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <scope>provided</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```