

sportyshoes.com

Project Source Code

Version History:

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

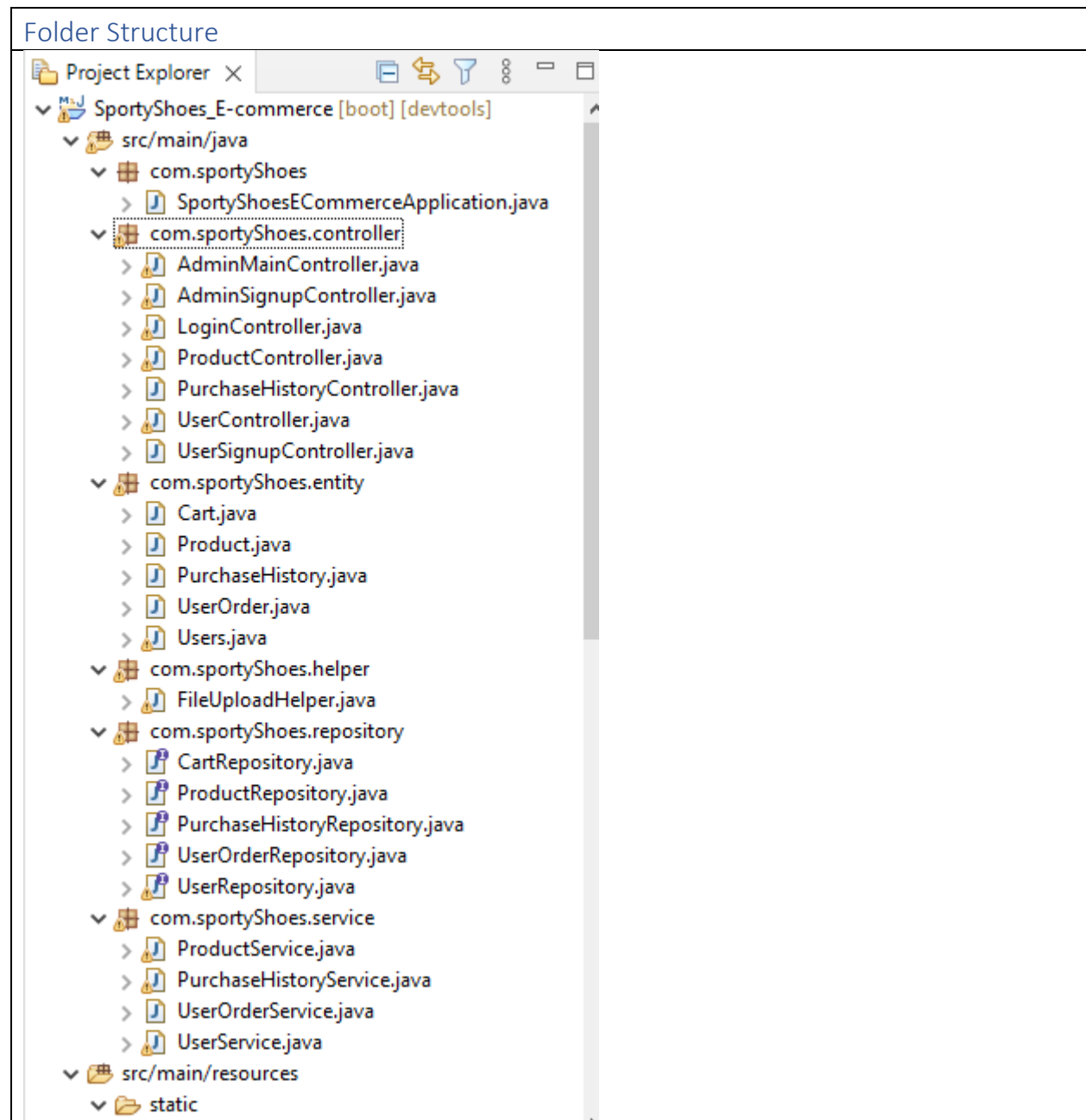
Table of Content:

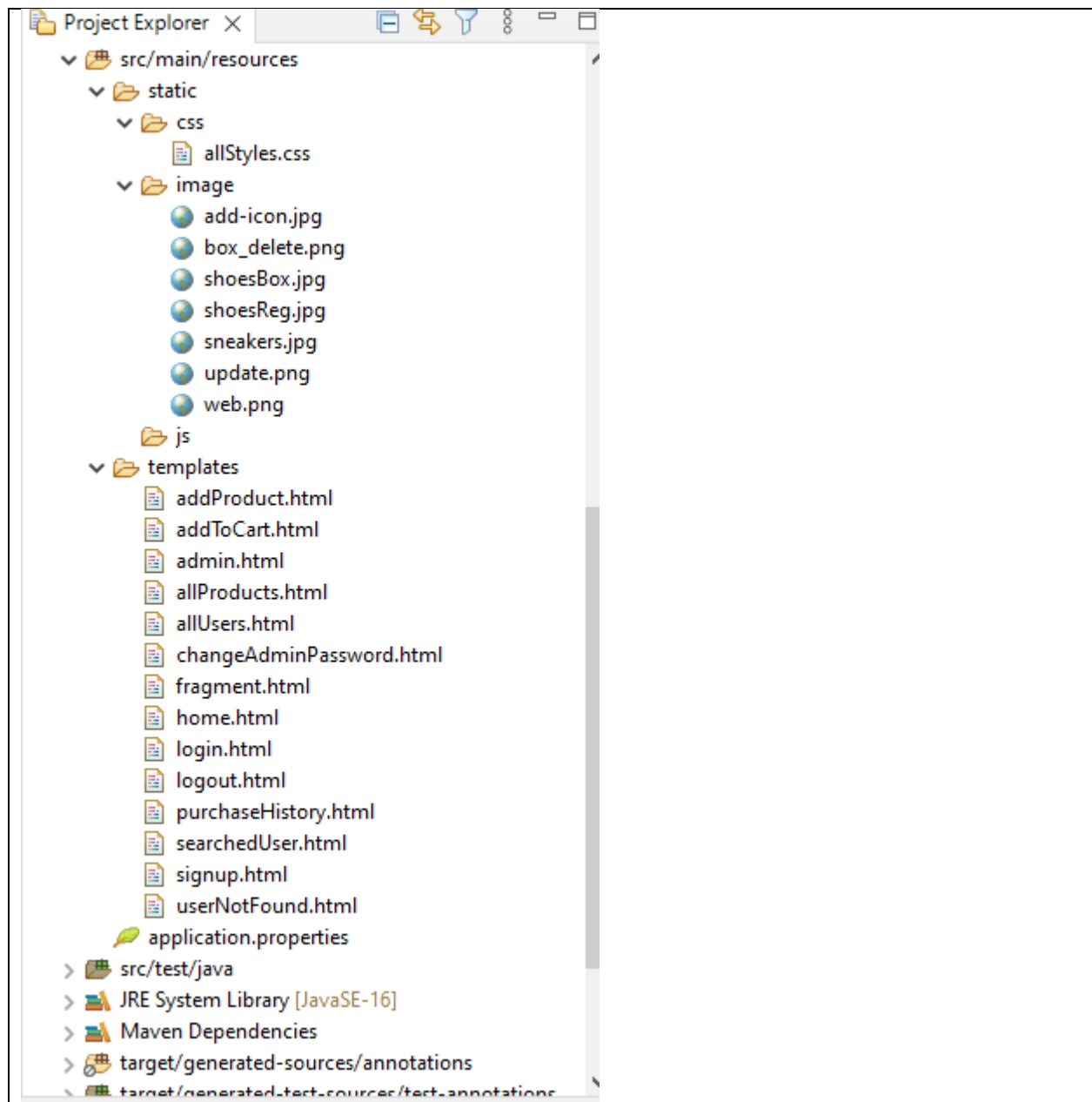
Project GITHUB Link:	3
Project Code:	4

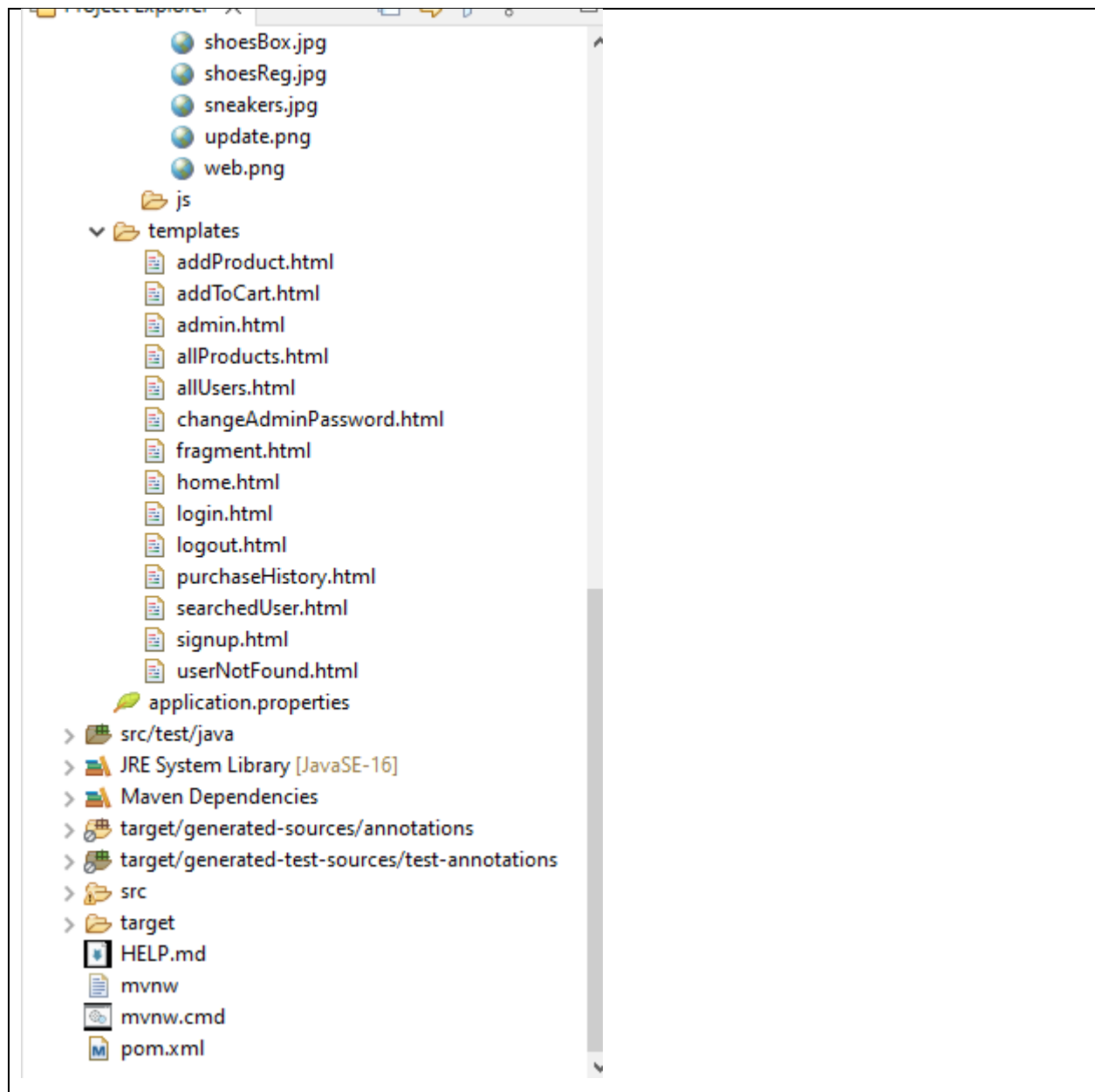
Project GITHUB Link:

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

Project Code:







- **Entity**

Product.java

```
Product.java X
1 package com.sportyShoes.entity;
2
3 import java.util.ArrayList;
21
22 @Data
23 @NoArgsConstructor
24 @AllArgsConstructor
25 @Entity
26 @Table(name = "products")
27 public class Product {
28     @Id
29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     @Column(name = "id", updatable = true, nullable = false)
31     private int id;
32
33     @Column(name = "product_category", nullable = false)
34     private String category;
35
36     @Column(name = "product_name", nullable = false)
37     private String productName;
38
39     @Column(name = "product_price", nullable = false)
40     private int productPrice;
41
42     @Lob
43     @Column(name = "p_photo", nullable = true)
44     private String pPhoto;
45
46     @ManyToMany(fetch = FetchType.EAGER)
47     @JoinTable(name = "product_order",
48         joinColumns = @JoinColumn(name="product_id"),
49         inverseJoinColumns = @JoinColumn(name="order_id"))
50     private List<UserOrder> userOrders=new ArrayList<UserOrder>();
51
```

```

29     @GeneratedValue(strategy = GenerationType.IDENTITY)
30     @Column(name = "id", updatable = true, nullable = false)
31     private int id;
32
33     @Column(name = "product_category", nullable = false)
34     private String category;
35
36     @Column(name = "product_name", nullable = false)
37     private String productName;
38
39     @Column(name = "product_price", nullable = false)
40     private int productPrice;
41
42     @Lob
43     @Column(name = "p_photo", nullable = true)
44     private String pPhoto;
45
46     @ManyToMany(fetch = FetchType.EAGER)
47     @JoinTable(name = "product_order",
48         joinColumns = @JoinColumn(name="product_id"),
49         inverseJoinColumns = @JoinColumn(name="order_id"))
50     private List<UserOrder> userOrders=new ArrayList<UserOrder>();
51
52
53     public void addUserOrders(UserOrder userOrder) {
54         this.userOrders.add(userOrder);
55     }
56
57     public void delUserOrders(UserOrder userOrder) {
58         this.userOrders.remove(userOrder);
59     }
60
61 }
62

```


PurchaseHistory.java

```
1 package com.sportyShoes.entity;
2
3 import java.util.Date;
4
5
6
7
8
9
10
11
12
13
14
15
16 @Data
17 @AllArgsConstructor
18 @NoArgsConstructor
19 @Entity
20 @Table(name = "order_purchase_history")
21 public class PurchaseHistory {
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     @Column(name = "id",updatable = false, nullable = false)
25     private int id;
26
27     private int productId;
28     private int orderId;
29     private int price;
30     private int total;
31     private int userId;
32     private String category;
33     private String productName;
34     private Date date;
35
36
37 }
38
```

UserOrder.java

```
1 package com.sportyShoes.entity;
2
3 import java.util.ArrayList;
4
5
6
7
8
9
10
11
12
13 @Data
14 @NoArgsConstructor
15 @AllArgsConstructor
16 @Entity
17 @Table(name = "user_orders")
18 public class UserOrder {
19     @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     @Column(name = "order_id", updatable = true, nullable = false)
22     private int id;
23
24     private int total;
25
26     @OneToOne(cascade = CascadeType.ALL, mappedBy = "userOrder")
27     private Users users;
28
29     private Date date;
30
31     @ManyToMany(mappedBy = "userOrders", fetch = FetchType.EAGER)
32     private List<Product> products=new ArrayList<>();
33
34     public void addProducts(Product product) {
35         this.products.add(product);
36     }
37
38     public void delProducts(Product product) {
39         this.products.remove(product);
40     }
41 }
42
```

Users.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

```
AdminMainController.java X
1 package com.sportyShoes.controller;
2
3 import java.util.List;
23
24 @Controller
25 @Transactional
26 public class AdminMainController {
27
28     @Autowired
29     private UserService userService;
30
31     @Autowired
32     private UserRepository userRepository;
33
34     @GetMapping("/changeAdminPassword")
35     public String changeAdminPassword()
36     {
37         return "changeAdminPassword";
38     }
39
40     @PostMapping("/changePassword")
41     public String changePassword(
42         @RequestParam Map<String, String> maps,
43         RedirectAttributes ra
44     )
45     {
46
47         String email=maps.get("email");
48         String oldPassword=maps.get("userOldPassword");
49         String userNewPassword=maps.get("userNewPassword");
50         System.out.println(email);
51         System.out.println(oldPassword);
52         System.out.println(userNewPassword);
53
54     }
```

```

51     System.out.println(oldPassword);
52     System.out.println(userNewPassword);
53
54
55     Users user = userService.getUserByEmail(email);
56     System.out.println(user);
57     String userType = user.getUserType();
58     System.out.println(userType);
59     if(userType.equalsIgnoreCase("admin"))
60     {
61         Users admin = userService.verifyAdmin(userType);
62         System.out.println(admin);
63         String adminPassword = admin.getUserPassword();
64         int id=admin.getId();
65         System.out.println(adminPassword);
66         if(adminPassword.equals(oldPassword))
67         {
68             if(!oldPassword.equals(userNewPassword))
69             {
70                 admin.setUserPassword(userNewPassword);
71                 System.out.println(admin);
72                 userService.updatePasswordById(userNewPassword, id);
73                 // ra.addFlashAttribute("message", "Admin Password changed success
74                 // ra.addFlashAttribute("alertClass", "alert-success");
75                 return "redirect:/admin1";
76             }
77             else if(oldPassword.equals(userNewPassword))
78             {
79                 ra.addFlashAttribute("message", "Updating Password failed!!, pl
80                 ra.addFlashAttribute("alertClass", "alert-danger");
81                 return "redirect:/changeAdminPassword";
82             }
83         }
84         /*} else
85         {*/

```

```

75         return "redirect:/admin1";
76     }
77     else if(oldPassword.equals(userNewPassword))
78     {
79         ra.addFlashAttribute("message", "Updating Password failed!!, please try again");
80         ra.addFlashAttribute("alertClass", "alert-danger");
81         return "redirect:/changeAdminPassword";
82     }
83 }
84 /*} else
85 {*/
86 ra.addFlashAttribute("message", "Updating Password failed!!, please try again");
87 ra.addFlashAttribute("alertClass", "alert-danger");
88 return "redirect:/changeAdminPassword";
89 } else
90 {
91     ra.addFlashAttribute("message", "Sorry!!!,Only Admin password can be changed");
92     ra.addFlashAttribute("alertClass", "alert-danger");
93     return "redirect:/changeAdminPassword";
94 }
95
96 // return "redirect:/changeAdminPassword";
97
98 }
99
100 @PostMapping("/searchUser")
101 public String searchUserByName(Model m,@RequestParam("userName") String userName)
102 {
103     List<Users> user = userRepository.getUserByName(userName);
104     m.addAttribute("user",user);
105     return "searchedUser";
106 }
107
108 }
109

```

AdminSignupController.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";
    }
}
```

```

22  @Autowired
23  private UserRepository userRepository;
24
25  @GetMapping("/admin1")
26  public String adminSignup(Model m, RedirectAttributes ra)
27  {
28      // m.addAttribute("product", new Product());
29
30      Users user = (Users) m.getAttribute("username");
31      if(user != null)
32      {
33
34          int userCount = userRepository.userCount();
35          m.addAttribute("userCount", userCount);
36
37          int productCount = productRepository.productCount();
38          m.addAttribute("count", productCount);
39          return "redirect:/admin";
40
41      }
42
43      ra.addFlashAttribute("message", "Admin Password changed successfully!!");
44      ra.addFlashAttribute("alertClass", "alert-success");
45      return "redirect:/admin";
46
47      
48
49  @GetMapping("/admin")
50  public String adminSignup()
51  {
52      return "admin";
53  }
54
55  }
56

```


LoginController.java

```
1 package com.sportyShoes.controller;
2
3+ import java.util.ArrayList;
20
21 @Controller
22 public class LoginController {
23
24-     @Autowired
25     private UserService userService;
26
27-     @GetMapping("/login")
28     public String loginHandler(Model m)
29     {
30         m.addAttribute("user", new Users());
31         return "login";
32     }
33
34-     @PostMapping("/loginProcess")
35     public String loginProcessHandler(
36         @Valid @ModelAttribute("user") Users users,
37         Model m,
38         HttpServletRequest req,
39         RedirectAttributes ra,
40         BindingResult res)
41     {
42         ra.addFlashAttribute("message", "Login failed!!, please try again or signup f
43         ra.addFlashAttribute("alertClass", "alert-danger");
44
45         List<Users> userList=this.userService.getAllUsers();
46         if(res.hasErrors())
47         {
48             System.out.println(res);
49             return "redirect:/login";
50         }
51
```

```
47     {
48         System.out.println(res);
49         return "redirect:/login";
50     }
51
52     String email=users.getEmail();
53     String password=users.getUserPassword();
54     Users user=null;
55     for(Users u:userList)
56     {
57         if(u.getEmail().equals(email) &&
58            u.getUserPassword().equals(password))
59         {
60             user = this.userService.getUserByEmailAndPassword(email, password);
61         }
62     }
63     // users=user;
64     if(user!=null)
65     {
66         System.out.println(user);
67         String username=user.getUserName();
68         req.getSession().setAttribute("username", username);
69         m.addAttribute("user",user);
70
71         if(user.getUserType().equals("admin"))
72         {
73
74             ra.addFlashAttribute("message", "User Logged in successfully!!");
75             ra.addFlashAttribute("alertClass", "alert-success");
76             return "redirect:/admin";
77         }
78         else if(user.getUserType().equals("normal")) {
79             ra.addFlashAttribute("message", "User Logged in successfully!!");
80             ra.addFlashAttribute("alertClass", "alert-success");
81             return "redirect:/home1";
```

<

```

78         else if(user.getUserType().equals("normal")) {
79             ra.addFlashAttribute("message", "User Logged in successfully!!");
80             ra.addFlashAttribute("alertClass", "alert-success");
81             return "redirect:/home1";
82         }
83         // else if(!users.getUserType().equals("admin")
84         //         && !users.getUserType().equals("normal")){
85         //         System.out.println(user);
86         //         ra.addFlashAttribute("message", "Login failed!!, please try again or
87         //         ra.addFlashAttribute("alertClass", "alert-danger");
88         //         return "redirect:/userNotFound";
89         //
90         //     }
91     }
92     System.out.println(user);
93     ra.addFlashAttribute("message", "Login failed!!, please try again or signup f
94     ra.addFlashAttribute("alertClass", "alert-danger");
95     return "redirect:/userNotFound";
96 }
97
98
99 @GetMapping("/userNotFound")
00 public String userNotFound()
01 {
02     return "userNotFound";
03 }
04
05 @GetMapping("/logoutUser")
06 public String logoutUser(RedirectAttributes ra)
07 {
08     ra.addFlashAttribute("message", "User Logged out successfully!!");
09     ra.addFlashAttribute("alertClass", "alert-success");
10     return "redirect:/logout";
11 }
12

```

```
87 //         ra.addFlashAttribute("alertClass", "alert-danger");
88 //         return "redirect:/userNotFound";
89 //
90 //     }
91
92     }
93     System.out.println(user);
94     ra.addFlashAttribute("message", "Login failed!!, please try again or signup
95     ra.addFlashAttribute("alertClass", "alert-danger");
96     return "redirect:/userNotFound";
97 }
98
99 @GetMapping("/userNotFound")
100 public String userNotFound()
101 {
102     return "userNotFound";
103 }
104
105 @GetMapping("/logoutUser")
106 public String logoutUser(RedirectAttributes ra)
107 {
108     ra.addFlashAttribute("message", "User Logged out successfully!!");
109     ra.addFlashAttribute("alertClass", "alert-success");
110     return "redirect:/logout";
111 }
112
113 @GetMapping("/logout")
114 public String logoutController(HttpServletRequest req)
115 {
116     req.getSession().removeAttribute("username");
117     return "logout";
118 }
119
120 }
121
```

ProductController.java

```
1 package com.sportyShoes.controller;
2
3 import java.io.IOException;
4
5 // @MultimapConfig
6 @Controller
7 public class ProductController {
8
9     @Autowired
10     private ProductService productService;
11
12     @Autowired
13     private ProductRepository productRepository;
14
15     @Autowired
16     private UserRepository userRepository;
17
18     @Autowired
19     private FileUploadHelper fileUploadHelper;
20
21     public final String UPLOAD_DIR=new ClassPathResource("static/image").getFile().getAbsolutePath();
22
23     public ProductController() throws IOException
24     {
25
26     }
27
28     @GetMapping("/getProducts")
29     public String getProducts(Model model)
30     {
31         List<Product> products=this.productService.getAllProducts();
32         model.addAttribute("products",products);
33         return "admin";
34     }
35 }
```

```

59     List<Product> products=this.productService.getAllProducts();
60     model.addAttribute("products",products);
61     return "admin";
62 }
63
64
65 // @RequestMapping(value = { "/addProduct" }, method = RequestMethod.POST, consumes
66 @PostMapping("/addProduct")
67 public String addProduct(@ModelAttribute Product p,
68 // @RequestParam("file") MultipartFile file,
69 // @RequestParam("") String pName,
70 // @RequestParam("") String pCategory,
71 // @RequestParam("") long pPrice,
72 RedirectAttributes ra) throws IOException, ServletException
73 {
74
75     try {
76
77         // if(file.isEmpty())
78         // {
79         //     ra.addFlashAttribute("message", "no picture is chosen !! please chose
80         //     ra.addFlashAttribute("alertClass", "alert-danger");
81         //     return "redirect:/admin";
82         // }
83
84         String category=p.getCategory();
85         String name=p.getProductName();
86         int price=p.getProductPrice();
87
88         // Product addProduct = this.productService.addProduct(p,file, name, categor
89         boolean addProduct = this.productService.addProduct(p);
90         if(addProduct)
91         {
92             ra.addFlashAttribute("message", "product added successfully!!");
93             ra.addFlashAttribute("alertClass", "alert-success");

```

```

88 //      Product addProduct = this.productService.addProduct(p,file, name, catego
89      boolean addProduct = this.productService.addProduct(p);
90      if(addProduct)
91      {
92          ra.addFlashAttribute("message", "product added successfully!!");
93          ra.addFlashAttribute("alertClass", "alert-success");
94      } else
95      {
96          ra.addFlashAttribute("message", "error!!,failed to add product!! ple
97          ra.addFlashAttribute("alertClass", "alert-danger");
98      }
99
100      }catch (Exception e) {
101          e.printStackTrace();
102      }
103
104      return "redirect:/admin";
105  }
106
107 @GetMapping("/delProduct")
108 public String delProduct(@ModelAttribute Product p,
109     RedirectAttributes ra)
110 {
111     int id = p.getId();
112     boolean delProduct = this.productService.delProduct(id);
113
114     if(delProduct)
115     {
116         ra.addFlashAttribute("message", "product deleted successfully!!");
117         ra.addFlashAttribute("alertClass", "alert-success");
118         return "redirect:/admin";
119     }
120
121     ra.addFlashAttribute("message", "Error !!,failed to delete product, please c
122     ra.addFlashAttribute("alertClass", "alert-danger");

```

```

120
121     ra.addFlashAttribute("message", "Error !!,failed to delete product, please c
122     ra.addFlashAttribute("alertClass", "alert-danger");
123     return "redirect:/admin";
124 }
125
126 @GetMapping("/update")
127 public String updateProduct(@ModelAttribute Product p,
128                             Model model,
129                             RedirectAttributes ra)
130 {
131
132     int id = p.getId();
133     // String category = p.getCategory();
134     // String productName = p.getProductName();
135     // BigInteger productPrice = p.getProductPrice();
136     boolean updateProduct = this.productService.updateProduct(p,id);
137
138     if(updateProduct)
139     {
140         List<Product> allProducts = productRepository.getAllProducts();
141         model.addAttribute("products",allProducts);
142         ra.addFlashAttribute("message", "product updated successfully!!");
143         ra.addFlashAttribute("alertClass", "alert-success");
144         return "redirect:/admin";
145     }
146
147     ra.addFlashAttribute("message", "Error !!,failed to product product, please
148     ra.addFlashAttribute("alertClass", "alert-danger");
149     return "redirect:/admin";
150 }
151
152 @GetMapping("/allProducts")
153 public String allProducts(Model m)
154 {

```



```

128         Model model,
129         RedirectAttributes ra)
130     {
131
132         int id = p.getId();
133         // String category = p.getCategory();
134         // String productName = p.getProductName();
135         // BigInteger productPrice = p.getProductPrice();
136         boolean updateProduct = this.productService.updateProduct(p,id);
137
138         if(updateProduct)
139         {
140             List<Product> allProducts = productRepository.getAllProducts();
141             model.addAttribute("products",allProducts);
142             ra.addFlashAttribute("message", "product updated successfully!!");
143             ra.addFlashAttribute("alertClass", "alert-success");
144             return "redirect:/admin";
145         }
146
147         ra.addFlashAttribute("message", "Error !!,failed to product product, please
148         ra.addFlashAttribute("alertClass", "alert-danger");
149         return "redirect:/admin";
150     }
151
152     @GetMapping("/allProducts")
153     public String allProducts(Model m)
154     {
155         List<Product> allProducts = productRepository.getAllProducts();
156
157         m.addAttribute("allProducts", allProducts);
158         return "allProducts";
159     }
160
161 }
162

```

PurchaseHistoryController.java

```
1 package com.sportyShoes.controller;
2
3 import java.util.List;
4
5 @Controller
6 public class PurchaseHistoryController {
7
8     @Autowired
9     private PurchaseHistoryRepository purchaseHistoryRepository;
10
11     @GetMapping("/purchaseHistory")
12     public String purchaseHistory(Model m)
13     {
14         List<PurchaseHistory> purchaseHistory = purchaseHistoryRepository.getPurchaseHi
15
16         m.addAttribute("purchaseHistory", purchaseHistory);
17         return "purchaseHistory";
18     }
19 }
```

UserController.java

```

35
36 @Autowired
37 private CartRepository cartRepository;
38
39 @GetMapping("/{"/home1"})
40 public String showhome(Model m,RedirectAttributes ra)
41 {
42     Users user = (Users) m.getAttribute("users");
43     if(user==null)
44     {
45         ra.addFlashAttribute("message", "User not Logged in!!, login first or s
46         ra.addFlashAttribute("alertClass", "alert-success");
47         return "redirect:/login";
48     }
49     List<Product> allProducts = productRepository.getAllProducts();
50     m.addAttribute("products",allProducts);
51     return "redirect:/home";
52
53 }
54
55 @GetMapping("/{"/home"})
56 public String showHome(Model m)
57 {
58     Users user = (Users) m.getAttribute("users");
59     if(user==null)
60     {
61         return "redirect:/login";
62
63         ra.addFlashAttribute( message , "User not Logged in!!, login first or sig
64         ra.addFlashAttribute("alertClass", "alert-success");
65         return "redirect:/login";
66     }
67
68     List<Product> allProducts = productRepository.getAllProducts();
69     m.addAttribute("products",allProducts);
70     return "redirect:/home1";
71
72 }
73
74
75 @GetMapping("/{"/home1"})
76 public String showHome(Model m)
77 {
78     //     Users user = (Users) m.getAttribute("users");
79     //     if(user==null)
80     //     {
81     //         return "redirect:/login";
82     //     }
83
84     List<Product> allProducts = productRepository.getAllProducts();
85     m.addAttribute("products",allProducts);
86     return "home";
87 }

```

// In Between is a commented section

```
112 @GetMapping("/allUsers")
113 public String allUsers(Model m)
114 {
115     List<Users> allUsers = userRepository.getAllUsers();
116     m.addAttribute("allUsers",allUsers);
117     return "allUsers";
118 }
119 }
120 }
```

- **REPOSITORY**

ProductRepository.java

```

ProductRepository.java X
1 package com.sportyShoes.repository;
2
3 import java.util.List;
13
14 @Transactional
15 @Repository
16 public interface ProductRepository extends JpaRepository<Product, Integer>{
17
18     @Query("select p from Product p")
19     public List<Product> getAllProducts();
20
21     @Query("select p from Product p where p.id =:id ")
22     public Product getById(@Param("id") int id);
23
24     @Query("select count(p) from Product p")
25     public int productCount();
26
27     @Query(value = "select uo,p from user_orders uo JOIN uo.products p order by p.pro
28     public List<Object[]> testPurchaseRepoQuery();
29
30 }
31

```

```

PurchaseHistoryRepository.java
1 package com.sportyShoes.repository;
2
3 import java.util.List;
10
11 @Repository
12 public interface PurchaseHistoryRepository extends JpaRepository<PurchaseHistory, Ir
13
14     @Query(value = "select * from order_purchase_history order by category and date c
15     public List<PurchaseHistory> getPurchaseHistory();
16
17     @Query(value = "SELECT * FROM order_purchase_history"
18             + " order by category and date desc",nativeQuery = true)
19     public List<PurchaseHistory> getPurchaseHistoryOrderByCategoryAndDate();
20 }
21

```

```

UserOrderRepository.java

```

```

1 package com.sportyShoes.repository;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12 @Repository
13 public interface UserOrderRepository extends JpaRepository<UserOrder, Integer> {
14
15     @Query("select u from UserOrder u")
16     public List<UserOrder> getAllOrders();
17
18     @Query("select uo from UserOrder uo where uo.id=:id")
19     public UserOrder getOrderById(@Param("id") int id);
20
21     @Query("select uo,p from UserOrder uo JOIN uo.products p order by p.category desc")
22     public List<Object[]> testPurchaseRepoQuery();
23 }
24

```

UserRepository.java

```

1 package com.sportyShoes.repository;
2
3 import java.util.List;
4
5 @Repository
6 // @Transactional
7 public interface UserRepository extends JpaRepository<Users, Integer> {
8     @Query("select u from Users u")
9     public List<Users> getAllUsers();
10
11     @Query("select u from Users u where u.email =:e and u.userPassword =:ps")
12     public Users getUserByEmailAndPassword(
13         @Param("e") String email,
14         @Param("ps") String password);
15
16     @Query("select u from Users u where u.id =:id")
17     public Users getUserById(@Param("id") int id);
18
19     @Query("select u from Users u where u.email =:email")
20     public Users getUserByEmail(@Param("email") String email);
21
22     @Query("select count(u) from Users u")
23     public int userCount();
24
25     @Query("select u from Users u where u.userName=:name")
26     public List<Users> getUserByName(@Param("name") String name);
27
28     // admin queries
29
30     // DML not allowed
31     // @Query("update Users u set u.userPassword =:password where u.id =:id")
32     // public Users updatePasswordById(@Param("password") String password, @Param("id")
33

```

```

@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

```
ProductService.java X
1 package com.sportyShoes.service;
2
3+ import java.io.IOException;
16
17 @Transactional
18 @Service
19 public class ProductService {
20
21     @Autowired
22     private EntityManager entityManager;
23
24     @Autowired
25     private ProductRepository productRepository;
26
27     @Autowired
28     private UserOrderRepository userOrderRepository;
29
30     @Autowired
31     private UserOrderService userOrderService;
32
33     public List<Product> getAllProducts()
34     {
35         List<Product> allProducts = productRepository.getAllProducts();
36         return allProducts;
37     }
38
39     public Product getProductById(int id)
40     {
41         Product p = productRepository.getById(id);
42         return p;
43     }
44
45     // public Product addProduct(Product p,
46     //                             MultipartFile file,
```

```
43     }
44
45 // public Product addProduct(Product p,
46 //                             MultipartFile file,
47 //                             String pName,
48 //                             String pCategory,
49 //                             int pPrice) throws IOException
50 // {
51 //
52 //     String image=StringUtils.cleanPath(file.getOriginalFilename());
53 //     if(image.contains(".."))
54 //     {
55 //         System.out.println("not a valid file");
56 //     }
57 //     p.setCategory(pCategory);
58 //     p.setProductName(pName);
59 //     p.setProductPrice(pPrice);
60 //     p.setPPhoto(Base64.getEncoder().encodeToString(file.getBytes()));
61 //     productRepository.save(p);
62 //     return p;
63 // }
64
65 public boolean addProduct(Product p) throws IOException
66 {
67     boolean f=false;
68     Product save = productRepository.save(p);
69     if(save != null)
70     {
71         f=true;
72     }
73
74     return f;
75 }
76
```

```

76
77 public boolean delProduct(int id)
78 {
79     boolean f=false;
80     Product byId = productRepository.getById(id);
81     if(byId !=null)
82     {
83         productRepository.delete(byId);
84         f=true;
85     }
86     return f;
87 }
88
89 public boolean updateProduct(Product p,int id)
90 {
91     boolean f=false;
92
93     Product byId = getProductById(id);
94     if(byId !=null)
95     {
96
97         byId.setCategory(p.getCategory());
98         byId.setProductName(p.getProductName());
99         byId.setProductPrice(p.getProductPrice());
100
101         Product save = productRepository.save(byId);
102         f=true;
103     }
104     return f;
105 }
106
107 public Product searchProduct(int id)
108 {
109     Product p=productRepository.getById(id);
110     return p;

```

```

104         return r;
105     }
106
107     public Product searchProduct(int id)
108     {
109         Product p=productRepository.getById(id);
110         return p;
111     }
112
113     public UserOrder addProductToOrder(int orderId, int productId)
114     {
115         UserOrder userOrder=userOrderRepository.getOrderById(orderId);
116         Product product = productRepository.getById(productId);
117         userOrder.addProducts(product);
118         product.addUserOrders(userOrder);
119         userOrderRepository.save(userOrder);
120         return userOrder;
121     }
122
123
124     public UserOrder addProductToUserOrder(int productId,int userId)
125     {
126         UserOrder userOrder=userOrderService.createNewOrder(userId);
127         int orderId=userOrder.getId();
128         UserOrder order=addProductToOrder(orderId, productId);
129         return order;
130     }
131
132     public List<Object[]> testPurchaseRepoQuery()
133     {
134         List<Object[]> resList = userOrderRepository.testPurchaseRepoQuery();
135         return resList;
136     }
137 }
138

```

PurchaseHistoryService.java

```
1 package com.sportyShoes.service;
2
3 import java.util.Date;
4
17
18 @Transactional
19 @Service
20 public class PurchaseHistoryService {
21
22     @Autowired
23     private ProductRepository productRepository;
24
25     @Autowired
26     private PurchaseHistoryRepository purchaseHistoryRepository;
27
28     @Autowired
29     private EntityManager em;
30
31     public void save(PurchaseHistory purchaseHistory)
32     {
33         purchaseHistoryRepository.save(purchaseHistory);
34     }
35
36     public void addDataToPurchaseHistory()
37     {
38         List<Object[]> rsList=productRepository.testPurchaseRepoQuery();
39
40         for(Object[] res:rsList)
41         {
42             Product p=(Product) res[1];
43             UserOrder userOrder=(UserOrder) res[0];
44             int productId=p.getId();
45             int orderId=userOrder.getId();
46             int cost=p.getProductPrice();
47             int total=userOrder.getTotal();
48             ...
49         }
50     }
51 }
```

```

39
40     for(Object[] res:rsList)
41     {
42         Product p=(Product) res[1];
43         UserOrder userOrder=(UserOrder) res[0];
44         int productId=p.getId();
45         int orderId=userOrder.getId();
46         int cost=p.getProductPrice();
47         int total=userOrder.getTotal();
48         int userId=userOrder.getUsers().getId();
49         String category=p.getCategory();
50         String productName=p.getProductName();
51         Date date=userOrder.getDate();
52         PurchaseHistory ph=new PurchaseHistory();
53         ph.setProductId(productId);
54         ph.setOrderId(orderId);
55         ph.setPrce(cost);
56         ph.setTotal(total);
57         ph.setUserId(userId);
58         ph.setCategory(category);
59         ph.setProductName(productName);
60         ph.setDate(date);
61         save(ph);
62     }
63 }
64
65 public List<PurchaseHistory> getPurchaseHistory()
66 {
67     addDataToPurchaseHistory();
68     List<PurchaseHistory> purchaseHistory = purchaseHistoryRepository.getPurchas
69     return purchaseHistory;
70 }
71
72 }
73

```

UserOrderService.java

```
1 package com.sportyShoes.service;
2
3+ import java.util.Date;
17
18 @Transactional
19 @Service
20 public class UserOrderService {
21
22-     @Autowired
23     private UserOrderRepository userOrderRepository;
24
25-     @Autowired
26     private UserRepository userRepository;
27
28-     @Autowired
29     private UserService userService;
30
31-     @Autowired
32     private EntityManager entityManager;
33
34-     public UserOrder insert()
35     {
36         UserOrder userOrder=new UserOrder();
37         Product product=new Product();
38
39
40         userOrder.setDate(new Date());
41         product.setProductName("woodland shoe");
42         entityManager.persist(product);
43         entityManager.persist(userOrder);
44
45         userOrder.addProducts(product);
46         product.addUserOrders(userOrder);
47         entityManager.persist(userOrder);
48         return userOrder;
```

```
45         userOrder.addProducts(product);
46         product.addUserOrders(userOrder);
47         entityManager.persist(userOrder);
48         return userOrder;
49     }
50
51     public int TotalCost(int orderId) {
52         UserOrder userOrder=entityManager.find(UserOrder.class, orderId);
53         List<Product> products=userOrder.getProducts();
54         int sum=0;
55
56         for(int i=0;i< products.size();i++)
57         {
58             sum+=products.get(i).getProductPrice();
59         }
60         return sum;
61     }
62
63     public int addTotalCostToOrder(int orderId)
64     {
65         int sum=TotalCost(orderId);
66         UserOrder userOrder=entityManager.find(UserOrder.class, orderId);
67         userOrder.setTotal(sum);
68         entityManager.persist(userOrder);
69         entityManager.flush();
70         return userOrder.getTotal();
71     }
72
73
74     public List<UserOrder> getAllOrders()
75     {
76         List<UserOrder> allOrders = userOrderRepository.getAllOrders();
77         return allOrders;
78     }
79
```



```

73
74 public List<UserOrder> getAllOrders()
75 {
76     List<UserOrder> allOrders = userOrderRepository.getAllOrders();
77     return allOrders;
78 }
79
80 public UserOrder createNewOrder(int id)
81 {
82     Users users=userRepository.getUserById(id);
83     UserOrder userOrder=users.getUserOrder();
84     if(userOrder==null)
85     {
86         userOrder=new UserOrder();
87         userOrder.setDate(new Date());
88         userOrder.setId(id);
89         entityManager.persist(userOrder);
90         userService.addUserOrder(id, userOrder);
91         entityManager.persist(userOrder);
92         return userOrder;
93     } else
94     {
95         return userOrder;
96     }
97 }
98
99
100 public UserOrder getOrder(int orderId)
101 {
102     UserOrder userOrder=userOrderRepository.getOrderById(orderId);
103     return userOrder;
104 }
105
106 }
107

```

UserService.java

```
1 package com.sportyShoes.service;
2
3+ import java.util.List;
11
12 @Service
13 public class UserService {
14-     @Autowired
15     private UserRepository userRepository;
16
17     // adding a user
18-     public Users addUser(Users u)
19     {
20         Users us=userRepository.save(u);
21         return u;
22     }
23
24     // get all users
25-     public List<Users> getAllUsers()
26     {
27         List<Users> allUsers = userRepository.getAllUsers();
28         return allUsers;
29     }
30
31     // get user by email and password
32-     public Users getUserByEmailAndPassword(String email, String password)
33     {
34         Users users = userRepository.getUserByEmailAndPassword(email, password);
35         return users;
36     }
37
38     // get user by email
39-     public Users getUserByEmail(String email)
40     {
41         Users users = userRepository.getUserByEmail(email);
42     }
43 }
```

```

37
38 // get user by email
39 public Users getUserByEmail(String email)
40 {
41     Users users = userRepository.getUserByEmail(email);
42     return users;
43 }
44
45 // admin functions
46 public Users verifyAdmin(String userType)
47 {
48     Users admin = userRepository.verifyAdmin(userType);
49     return admin;
50 }
51 //
52 public void addUserOrder(int id, UserOrder userOrder)
53 {
54     Users user = userRepository.getUserById(id);
55     user.setUserOrder(userOrder);
56     Users save = userRepository.save(user);
57 }
58
59 public List<Users> searchUserByName(String name)
60 {
61     List<Users> user = userRepository.getUserByName(name);
62     return user;
63 }
64
65 public void addUserOrderToUser(int id, UserOrder order)
66 {
67     Users u=userRepository.getUserById(id);
68     u.setUserOrder(order);
69     userRepository.save(u);
70 }
71

```

```

59 public List<Users> searchUserByName(String name)
60 {
61     List<Users> user = userRepository.getUserByName(name);
62     return user;
63 }
64
65 public void addUserOrderToUser(int id, UserOrder order)
66 {
67     Users u=userRepository.getUserById(id);
68     u.setUserOrder(order);
69     userRepository.save(u);
70 }
71
72 public UserOrder verifyUserToOrder(int userId)
73 {
74     Users users=userRepository.getUserById(userId);
75     return users.getUserOrder();
76 }
77
78 //DML not allowed
79 // public Users updatePasswordById(String password, int id)
80 // {
81 //     Users users1 = userRepository.updatePasswordById(password, id);
82 //     return users1;
83 // }
84
85 // public Users updatePasswordByEmailAndOldPasswordAndNewPassword(
86 //     String email, String oldPassword, String newUserPassword)
87 // {
88 //     Users user = null;
89 //     return user;
90 // }
91
92 }
93

```

- CSS

allStyles.css

```
allStyles.css X
1 @charset "ISO-8859-1";
2
3 .bg-custom{
4     background-color: #4e9ec5;
5 }
6
7 .reg_bck{
8     background: url("/image/sneakers.jpg");
9     height: 100vh;
10    width: 100%;
11    background-size: cover;
12    background-attachment: fixed;
13    margin: 0 auto -48px;
14 }
15
16 .home_bck{
17     background-color: #EBF3B4;
18     height: 100vh;
19     width: 100%;
20     background-size: cover;
21     background-attachment: fixed;
22     margin: 0 auto -48px;
23 }
24
25 .navbar-bg{
26     background-color: #54A6F3;
27 }
28
29 .shoes-card:hover{
30     background: #ADD8E6;
31     cursor: pointer;
32 }
33
34 .table-lights thead tr th{
35     text-align: center;
```

```
29⌵ .shoes-card:hover{
30     background: #ADD8E6;
31     cursor: pointer;
32 }
33
34⌵ .table-lights thead tr th{
35     text-align: center;
36 }
37
38⌵ .table-lights tbody tr td{
39     text-align: center;
40 }
41
42⌵ #places {
43     font-family: Arial, Helvetica, sans-serif;
44     border-collapse: collapse;
45     width: 100%;
46 }
47
48⌵ #places td, #places th {
49     border: 1px solid #ddd;
50     padding: 8px;
51 }
52
53 #places tr:nth-child(even){background-color: #f2f2f2;}
54
55 #places tr:hover {background-color: #ddd;}
56
57⌵ #places th {
58     padding-top: 12px;
59     padding-bottom: 12px;
60     text-align: left;
61     background-color: #4592de;
62     color: white;
63 }
```

```
36 }
37
38⊖ .table-lights tbody tr td{
39     text-align: center;
40 }
41
42⊖ #places {
43     font-family: Arial, Helvetica, sans-serif;
44     border-collapse: collapse;
45     width: 100%;
46 }
47
48⊖ #places td, #places th {
49     border: 1px solid #ddd;
50     padding: 8px;
51 }
52
53 #places tr:nth-child(even){background-color: #f2f2f2;}
54
55 #places tr:hover {background-color: #ddd;}
56
57⊖ #places th {
58     padding-top: 12px;
59     padding-bottom: 12px;
60     text-align: left;
61     background-color: #4592de;
62     color: white;
63 }
64
65⊖ .flights {
66     font-family: Arial, Helvetica, sans-serif;
67     border-collapse: collapse;
68     width: 100%;
69 }
70
```

- **TEMPLATES**

addProduct.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Add Product</title>
6 <style>
7     .page-footer, .push{
8         height: 35px;
9     }
10 </style>
11 </head>
12 <body>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16 <div class="container-fluid reg_bck" >
17
18     <!-- for footer -->
19     <div class="push"></div>
20 </div>
21
22 <!-- footer -->
23 <div th:replace="fragment :: html_footer"></div>
24 </body>
25 </html>
```


Admin.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Admin Page</title>
6 <style>
7     .page-footer, .push{
8         height: 35px;
9     }
10 </style>
11 </head>
12 <body>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16     <!-- dynamic content body -->
17     <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+'); background-size: cover;'" -->
18     <div class="container-fluid reg_bck admin">
19
20         <!-- message alert -->
21         <div th:align="center" th:if="${message}" th:text="${message}" th:class="${message != '' ? 'alert alert-success' : ''}">
22             ${message}
23         </div>
24
25         <!-- admin content starts -->
26
27         <!-- first row -->
28         <div class="row">
29             <div class="col-md-4 text-center">
30                 <!-- 1st box of categories -->
31                 <div class="card shoes-card "
32                     data-bs-toggle="modal"
33                     data-bs-target="#add-product">
34                     <div class="card-body text-center">
35                         <div class="container text-center">
36                             <img class="img-fluid rounded"
37                                 style="max-width: 85px; border-style: solid; border-width: 1px; border-color: #ccc;" alt="Sneakers image" data-bbox="400 580 450 600"/>
38                         </div>
39                     </div>
40                 </div>
41             </div>
42         </div>
43     </div>
44 
```

```

33<div class="container text-center">
34    
37    </div>
38<h3 class="text-uppercase text-muted">
39    Add Product
40</h3>
41</div>
42</div>
43</div>
44
45<!-- 2nd col -->
46<div class="col-md-4">
47    <div class="card shoes-card"
48        data-bs-toggle="modal"
49        data-bs-target="#delete-product">
50        <div class="card-body text-center">
51            <div class="container">
52                <img style="max-width: 85px; border-style: solid;"
53                    class="img-fluid rounded" alt="Airlines" th:src
54            </div>
55            <h3 class="text-uppercase text-muted">
56                Delete Product
57            </h3>
58        </div>
59    </div>
60</div>
61
62<div class="col-md-4">
63    <div class="card shoes-card"
64        data-bs-toggle="modal"
65        data-bs-target="#update-product">
66        <div class="card-body text-center">
67            <div class="container">

```

```

61
62<
63<div class="col-md-4">
64    <div class="card shoes-card"
65        data-bs-toggle="modal"
66        data-bs-target="#update-product">
67        <div class="card-body text-center">
68            <div class="container">
69                <img style="max-width: 85px; border-style: solid;"
70                    class="img-fluid rounded" alt="Flights" th:src=
71            </div>
72            <h3 class="text-uppercase text-muted">
73                Update Product
74            </h3>
75        </div>
76    </div>
77</div>
78<!-- 1st row ends -->
79
80<!-- 2nd row -->
81<div class="row mt-3">
82    <!-- 1st column -->
83    <div class="col-md-6">
84        <!-- 1st box -->
85        <div class="card shoes-card">
86            <div class="card-body text-center">
87                <div class="container">
88                    
91                </div>
92                <p class="mt-2" th:text="${count}">13</p>
93                <a href="allProducts">Products</a>
94                <h3 class="text-uppercase text-muted">Total Products</h3>
95            </div>

```

```

94         <h3 class="text-uppercase text-muted">Total Products</h3>
95     </div>
96 </div>
97 </div>
98
99 <!-- 2nd column -->
100 <div class="col-md-6">
101     <!-- 2nd box -->
102     <div class="card shoes-card">
103         <div class="card-body text-center">
104             <div class="container">
105                 
108             </div>
109             <p class="mt-2" th:text="${userCount}">14</p>
110             <a href="allUsers">Users</a>
111             <h3 class="text-uppercase text-muted">Users</h3>
112         </div>
113     </div>
114 </div>
115 </div>
116 <!-- 2nd row ends -->
117
118 <!-- 3rd row -->
119 <!-- change admin password and check purchase History-->
120 <div class="row">
121     <div class="col-md-4 offset-md-2">
122         <div class="card mt-5">
123             <div class="card-header text-center text-white bg-custom">
124
125                 <i class="fa fa-user-secret fa-4x" aria-hidden="true">
126
127             </div>
128             <div class="card-body" style="background-color: #F7FBBB;

```

```

127         </div>
128     <div class="card-body" style="background-color: #F7F8BB;
129
130         <div class="form-group">
131             <label for="exampleInputEmail1">Change Admin P
132             <label>Purchase History Link</label>
133             <a href="changeAdminPassword">change</a> &nbsp;
134                                     &nt
135                                     &nt
136                                     &nt
137                                     &nt
138                                     &nt
139                                     <a
140
141         </div>
142
143     </div>
144 </div>
145 </div>
146
147 <!-- Search User -->
148 <div class="col-md-4 offset-md-1">
149     <div class="card mt-5">
150         <div class="card-header text-center text-white bg-custom">
151
152             <i class="fa fa-search fa-4x" aria-hidden="true"></i>
153
154         </div>
155         <div class="card-body" style="background-color: #F7F8BB;
156
157             <div class="form-group">
158
159                 <label>Search User</label> <br />
160                 <form th:action="@{/searchUser}" method="post"
161                 <input type="text" name="userName"

```

```

158
159         <label>Search User</label> <br />
160         <form th:action="@{/searchUser}" method="post">
161             <input type="text" name="userName"
162                 placeholder="Enter name of user">
163             <Button class="signin-button" type="submit">
164         </form>
165     </div>
166
167
168     </div>
169 </div>
170 </div>
171
172     <!-- for footer -->
173     <div class="push"></div>
174 </div>
175 </div>
176
177
178 <!-- Modal -->
179 <!-- 1st modal add-product -->
180 <div class="modal fade" id="add-product" tabindex="-1" role="dialog"
181     aria-labelledby="exampleModalLabel" aria-hidden="true">
182     <div class="modal-dialog modal-lg" role="document">
183         <div class="modal-content">
184             <div class="modal-header">
185                 <h5 class="modal-title" id="exampleModalLabel">Fill Product
186                     Details</h5>
187                 <button type="button" class="btn-close" data-bs-dismiss="modal"
188                     aria-label="Close">
189             </button>
190         </div>
191         <div class="modal-body" style="background-color: #99c2ff">
192             <form th:action="@{/addProduct}" method="post">

```

```

189         </button>
190     </div>
191     <div class="modal-body" style="background-color: #99c2ff">
192         <form th:action="@{/addProduct}" method="post">
193
194             <!-- enctype="multipart/form-data" -->
195             <div class="form-group">
196                 <label>Product Name</label>
197                 <input type="text" class="form-control" name="product"
198                     placeholder="enter product name">
199             </div>
200             <div class="form-group">
201                 <label>Product Category</label>
202                 <input class="form-control"
203                     name="category" placeholder="enter product category"
204                     required="required">
205             </div> <br />
206             <!-- <div class="form-group">
207                 <label for="pPhoto"> Select Product Picture </label>
208                 <input type="file" id="pPhoto" name="pPhoto"
209                     accept="image/png, image/jpeg">
210             </div> <br /> -->
211             <div class="form-group">
212                 <label>Product Price</label>
213                 <input class="form-control"
214                     name="productPrice" placeholder="enter product price"
215                     required="required">
216             </div>
217             <br />
218             <div class="container text-center">
219                 <button class="btn btn-success btn-outline-dark">Add Product</button>
220                 <button type="button" class="btn btn-secondary"
221                     data-bs-dismiss="modal">Close</button>
222             </div>
223         </form>

```

```

219         <button class="btn btn-success btn-outline-dark">
220             <button type="button" class="btn btn-secondary"
221                 data-bs-dismiss="modal">Close</button>
222         </div>
223     </form>
224 </div>
225
226     </div>
227 </div>
228 </div>
229 <!-- End add airline modal -->
230
231 <!-- 2. delete product modal -->
232 <!-- Modal -->
233 <div class="modal fade" id="delete-product" tabindex="-1" role="dialog"
234     aria-labelledby="exampleModalLabel" aria-hidden="true">
235     <div class="modal-dialog modal-lg" role="document">
236         <div class="modal-content">
237             <div class="modal-header custom-bg ">
238                 <h5 class="modal-title" id="exampleModalLabel">Fill
239                     Details</h5>
240                 <button type="button" class="btn-close" data-bs-dismiss="m
241                     aria-label="Close">
242             </button>
243         </div>
244         <div class="modal-body" style="background-color: #99c2ff">
245
246             <form th:action="@{/delProduct}" method="get">
247
248                 <div class="form-group mt-3">
249                     <label>Product Id</label> <input type="number"
250                         class="form-control" name="id" required="requi
251                         placeholder="enter product id">
252                 </div>
253

```



```

251         placeholder="enter product id">
252     </div>
253
254
255     <div class="container text-center mt-3">
256         <button class="btn btn-success btn-outline-dark">
257             <button type="button" class="btn btn-secondary"
258                 data-bs-dismiss="modal">Close</button>
259         </div>
260     </form>
261 </div>
262
263 </div>
264 </div>
265 </div>
266 <!-- End of delete product modal -->
267
268 <!-- Modal -->
269 <!-- update product modal -->
270 <div class="modal fade" id="update-product" tabindex="-1" role="dialog"
271     aria-labelledby="exampleModalLabel" aria-hidden="true">
272     <div class="modal-dialog modal-lg" role="document">
273         <div class="modal-content">
274             <div class="modal-header custom-bg">
275                 <h5 class="modal-title" id="exampleModalLabel">Fill
276                     Details to update</h5>
277                 <button type="button" class="btn-close" data-bs-dismiss="m
278                     aria-label="Close">
279                 </button>
280             </div>
281             <div class="modal-body" style="background-color: #99c2ff">
282
283                 <form th:action="@{/update}" method="get" th:object="${pro
284
285                     <!-- enctype="multipart/form-data" -->

```

```

283 <form th:action="@{/update}" method="get" th:object=${product}
284
285         <!-- enctype="multipart/form-data" -->
286 <div class="form-group mt-3">
287     <label>Product Id</label> <input type="number"
288         class="form-control" name="id" required="required"
289         placeholder="enter product id">
290 </div>
291
292 <div class="form-group">
293     <label>Product Name</label>
294     <input type="text" class="form-control" name="name"
295         placeholder="enter product name"
296         required="required">
297 </div>
298 <div class="form-group">
299     <label>Product Category</label>
300     <input class="form-control"
301         name="category" placeholder="enter product
302         category" required="required">
303 </div> <br />
304 <!-- <div class="form-group">
305     <label for="pPhoto"> Select Product Picture </label>
306     <input type="file" id="pPhoto" name="pPhoto"
307         accept="image/png, image/jpeg">
308 </div> <br /> -->
309 <div class="form-group">
310     <label>Product Price</label>
311     <input class="form-control"
312         name="productPrice" placeholder="enter product
313         price" required="required">
314 </div>
315 <br />
316 <div class="container text-center">
317     <button class="btn btn-success btn-outline-dark">

```

```

300         <input class="form-control"
301             name="category" placeholder="enter product
302             required="required">
303     </div> <br />
304     <!-- <div class="form-group">
305         <label for="pPhoto"> Select Product Picture </l
306             type="file" id="pPhoto" name="pPhoto"
307             accept="image/png, image/jpeg">
308     </div> <br /> -->
309     <div class="form-group">
310         <label>Product Price</label>
311         <input class="form-control"
312             name="productPrice" placeholder="enter proc
313             required="required">
314     </div>
315     <br />
316     <div class="container text-center">
317         <button class="btn btn-success btn-outline-dark
318             <button type="button" class="btn btn-secondary'
319             data-bs-dismiss="modal">Close</button>
320     </div>
321     </form>
322 </div>
323
324     </div>
325 </div>
326 </div>
327
328 <!-- admin content ends -->
329
330
331 <!-- footer -->
332 <div th:replace="fragment :: html_footer"></div>
333 </body>
334 </html>

```

AllProducts.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Products</title>
6 <style>
7     .page-footer, .push{
8         height: 35px;
9     }
10 </style>
11 </head>
12 <body>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16 <div class="container-fluid home_bck">
17     <!-- message alert -->
18     <div th:align="center" th:if="${message}"
19         th:text="${message}"
20         th:class="${'alert ' + alertClass}"></div>
21
22     <div th:if="${not #lists.isEmpty(allProducts)}">
23         <table class="table">
24             <thead>
25                 <tr>
26                     <th scope="col">ID</th>
27                     <th scope="col">Name</th>
28                     <th scope="col">Category</th>
29                     <th scope="col"><i class="fa fa-inr"></i> Price</th>
30
31                 </tr>
32             </thead>
33             <tbody>
34
35                 <tr class="table-info" th:each="product : ${allProducts}>

```

```

22<=    <div th:if="{not #lists.isEmpty(allProducts)}">
23<=        <table class="table">
24<=            <thead>
25<=                <tr>
26<=                    <th scope="col">ID</th>
27<=                    <th scope="col">Name</th>
28<=                    <th scope="col">Category</th>
29<=                    <th scope="col"><i class="fa fa-inr"></i> Price</th>
30<=
31<=                </tr>
32<=            </thead>
33<=            <tbody>
34<=
35<=                <tr class="table-info" th:each="product : ${allProducts}">
36<=                    <td th:text="{product.id}"></td>
37<=                    <td th:text="{product.productName}"></td>
38<=                    <td th:text="{product.category}"></td>
39<=                    <td th:text="{product.productPrice}"><i class="fa fa-inr"></i>
40<=
41<=                    <td><input type="hidden" th:value="{product.id}" name="id">
42<=                </tr>
43<=
44<=            </tbody>
45<=        </table>
46<=    </div>
47<=
48<=
49<=    <!-- for footer -->
50<=    <div class="push"></div>
51<= </div>
52<=
53<=    <!-- footer -->
54<=    <div th:replace="fragment :: html_footer"></div>
55<= </body>
56<= </html>

```

allUsers.html

```

1 <!DOCTYPE html>
2<Ⓜhtml lang="en" xmlns:th="http://www.thymeleaf.org">
3<Ⓜhead th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Users</title>
6<Ⓜstyle>
7     .page-footer, .push{
8         height: 35px;
9     }
10    /* .tab tr th{
11        border-color: black;
12        background-color: black;
13    } */
14</style>
15</head>
16<Ⓜbody>
17    <!-- header -->
18    <div th:replace="fragment :: html_header"></div>
19
20<Ⓜdiv class="container-fluid home_bck">
21
22    <div th:if="${not #lists.isEmpty(allUsers)}">
23        <h2 class="text-center">User list</h2>
24
25        <table class="table">
26            <thead>
27                <tr>
28                    <th>User_Id</th>
29                    <th>User_Email</th>
30                    <th>User_Name</th>
31                    <th>Password</th>
32
33                </tr>
34            </thead>
35            <tr class="table-info" th:each="user : ${allUsers}">

```

```

20 <div class="container-fluid home_bck">
21
22 <div th:if="{not #lists.isEmpty(allUsers)}">
23 <h2 class="text-center">User list</h2>
24
25 <table class="table">
26 <thead>
27 <tr>
28 <th>User_Id</th>
29 <th>User_Email</th>
30 <th>User_Name</th>
31 <th>Password</th>
32
33 </tr>
34 </thead>
35 <tr class="table-info" th:each="user : {allUsers}">
36 <td th:text="{user.id}"></td>
37 <td th:text="{user.email}"></td>
38 <td th:text="{user.userName}"></td>
39 <td th:text="{user.userPassword}"></td>
40
41 </tr>
42 </table>
43
44
45 </div>
46
47 <!-- for footer -->
48 <div class="push"></div>
49 </div>
50
51 <!-- footer -->
52 <div th:replace="fragment :: html_footer"></div>
53 </body>
54 </html>

```

changeAdminPassword.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <style>
5     .page-footer, .push{
6         height: 35px;
7     }
8 </style>
9 <meta charset="ISO-8859-1">
10 <title>Admin Password Update</title>
11 </head>
12 <body>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16     <div class="container-fluid reg_bck">
17
18         <!-- message alert -->
19         <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'alert-' + ${message}.get(0).toLowerCase()}">
20             ${message}
21         </div>
22
23         <div class="row">
24             <div class="col-md-4 offset-md-4">
25                 <div class="card mt-4 mb-4" style="background: #e2e2e2;">
26                     <div class="card-header text-center bg-custom">
27                         <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
28                         <h1>Update Password here!!</h1>
29                     </div>
30                     <div class="card-body">
31                         <form th:action="@{/changePassword}" method="post">
32                             <!-- th:classappend="${#fields.hasErrors('email')?'invalid':''}" -->
33                             <div class="form-group">
34                                 <label for="exampleInputEmail1">Email address</label>
35                                 <input type="email" class="form-control" name="email" id="exampleInputEmail1">

```



```

33         type="email" class="form-control"
34         name="email"
35         id="exampleInputEmail1"
36
37         required="required"
38         aria-describedby="emailHelp"
39         placeholder="Enter Email here">
40
41         <!-- <div th:each="e:${#fields.errors('email')}}"
42
43     </div>
44
45     <!-- th:classappend="${#fields.hasErrors('userPassword')}"
46 <div class="form-group">
47     <label for="exampleInputPassword1">Old Password</label>
48     type="password" class="form-control"
49     id="exampleInputPassword1"
50
51     required="required"
52     name="userOldPassword"
53     placeholder="Enter Password here">
54
55     <!-- <div th:each="e:${#fields.errors('userPassv
56
57 </div>
58
59 <div class="form-group">
60     <label for="exampleInputPassword1">New Password</label>
61     type="password" class="form-control"
62     id="exampleInputPassword1"
63
64     required="required"
65     name="userNewPassword"
66     placeholder="Enter Password here">
67

```

```

54
55         <!-- <div th:each="e:${#fields.errors('userPassw
56
57     </div>
58
59     <div class="form-group">
60         <label for="exampleInputPassword1">New Password</lab
61         type="password" class="form-control"
62         id="exampleInputPassword1"
63
64         required="required"
65         name="userNewPassword"
66         placeholder="Enter Password here">
67
68     </div>
69
70     <br />
71     <button type="submit"
72         class="btn btn-success badge-pill btn-block">Submit<
73     <button type="reset" class="btn btn-primary badge-pill b
74 </form>
75 </div>
76 </div>
77 </div>
78 </div>
79
80 </div>
81
82 <!-- for footer -->
83 <div class="push"></div>
84
85 <!-- footer -->
86 <div th:replace="fragment :: html_footer"></div>
87 </body>
88 </html>

```

Fragment.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:fragment="html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>SportyShoes</title>
6 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
7 rel="stylesheet"
8 integrity="sha384-18mE4kWBq78iYhFlvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
9 crossorigin="anonymous">
10
11 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
12 integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
13 crossorigin="anonymous"></script>
14
15 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.1/css/all.min.css">
16 <link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
17 </head>
18 <body>
19
20 <!-- code for header -->
21 <nav class="navbar navbar-expand-lg navbar-light bg-custom"
22 th:fragment="html_header">
23 <div class="container-fluid">
24 <a class="navbar-brand" href="home1">SportyShoes</a>
25 <button class="navbar-toggler" type="button"
26 data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
27 aria-controls="navbarSupportedContent" aria-expanded="false"
28 aria-label="Toggle navigation">
29 <span class="navbar-toggler-icon"></span>
30 </button>
31 <div class="collapse navbar-collapse" id="navbarSupportedContent">
32 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
33 <li class="nav-item">
34 <a class="nav-link active"
35 aria-current="page" href="home1">
```

```

34< a class="nav-link active"
35    aria-current="page" href="home1">
36    Home
37</a>
38</li>
39
40<li class="nav-item dropdown">
41<a
42    class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
43    role="button" data-bs-toggle="dropdown" aria-expanded="false"
44    Dropdown
45</a>
46<ul class="dropdown-menu" aria-labelledby="navbarDropdown">
47    <li><a class="dropdown-item" href="#">Action</a></li>
48    <li><a class="dropdown-item" href="#">Another action</a>
49    <li><hr class="dropdown-divider"></li>
50    <li><a class="dropdown-item" href="#">Something else
51        here</a></li>
52</ul>
53</li>
54
55</ul>
56<!-- <form class="d-flex">
57    <input class="form-control me-2" type="search" placeholder="Search"
58        aria-label="Search">
59    <button class="btn btn-light btn-outline-success" type="submit">
60</form> -->
61    &nbsp;
62
63<div th:if="${session.username == null}">
64<!-- <div th:if="${users == null}"> -->
65<a href="/signup"><button
66    style="background-color: #82e0aa; border-radius: 10px;"
67    class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
68    <i class="fa fa-user-plus" aria-hidden="true"></i> Signup

```

```

65 <a href="signup"><button
66     style="background-color: #82eec7; border-radius: 10px;"
67     class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
68     <i class="fa fa-user-plus" aria-hidden="true"></i> Signup
69 </button></a>
70     &nbsp;
71 <a href="login"><button
72     style="background-color: #82eec7; border-radius: 10px;"
73     class="btn btn-light my-2 my-sm-0" type="submit">
74     <i class="fa fa-user-circle" aria-hidden="true"></i> Login
75 </button></a>
76 </div>
77
78 <div th:unless="{session.username == null}" >
79 <!-- <div th:unless="{users == null}" > -->
80 <a href="#"><button
81     style="background-color: #82eec7; border-radius: 10px;"
82     class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
83     <i class="fas fa-user-check"></i> <span th:text="{session.username}"
84 </button></a>
85     &nbsp;
86 <a href="logoutUser"><button
87     style="background-color: #82eec7; border-radius: 10px;"
88     class="btn btn-light my-2 my-sm-0" type="submit">
89     <i class="fa fa-user-circle" aria-hidden="true"></i> Logout
90 </button></a>
91 </div>
92
93 </div>
94 </div>
95 </nav>
96
97
98 <!-- footer -->
99 <div th:fragment="html_footer">

```

```

78     <div th:unless="{session.username == null}" >
79     <!-- <div th:unless="{users == null}" >      -->
80     <a href="#"><button
81         style="background-color: #82e0aa; border-radius: 10px;"
82         class="btn btn-light my-2 my-sm-0 mr-2" type="submit">
83         <i class="fas fa-user-check"></i> <span th:text="{session.username}"
84     </button></a>
85         &nbsp;
86     <a href="logoutUser"><button
87         style="background-color: #82e0aa; border-radius: 10px;"
88         class="btn btn-light my-2 my-sm-0" type="submit">
89         <i class="fa fa-user-circle" aria-hidden="true"></i> Logout
90     </button></a>
91     </div>
92
93     </div>
94     </div>
95 </nav>
96
97
98 <!-- footer -->
99 <div th:fragment="html_footer">
100     <footer class="pg_foot page-footer font-small bg-custom mt-5 mb-0">
101         <!-- Copyright -->
102         <div class="footer-copyright text-center py-3" style="color: white;">
103             &copy; 2021 Copyright:
104             <a href="home" style="color: white;"> SportyShoes.com</a>
105         </div>
106         <!-- Copyright -->
107     </footer>
108 <!-- footer -->
109 </div>
110 </body>
111 </html>

```

Home.html

```

1 <!DOCTYPE html>
2<⊖<html lang="en" xmlns:th="http://www.thymeleaf.org">
3<⊖<head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>SportyShoes.com</title>
6<⊖<style>
7 .pg_foot, .push {
8     height: 48px;
9 }
10
11 h3 {
12     color: yellow;
13     text-align: center;
14 }
15
16 .center {
17     margin-left: auto;
18     margin-right: auto;
19 }
20 </style>
21 </head>
22<⊖<body>
23     <!-- header -->
24     <div th:replace="fragment :: html_header"></div>
25
26     <!-- dynamic content body -->
27<⊖<div class="container-fluid home_bck text-center">
28     <div style="color: #2544DE; text-align: center; font-size: 30px;">Home <i c
29     <!-- message alert -->
30     <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'a
31
32     <h1>sporty shoes.com</h1>
33     <p>welcome to the e-commerce portal</p>
34     <!-- th:style="'background-image:url('+@{/image/sneakers.jpg}+');" background-
35

```

```

34      <!-- th:style="'background-image:url('@{/image/sneakers.jpg}+ '); background
35
36      <div th:if="${not #lists.isEmpty(products)}">
37      <table class="table">
38      <thead>
39      <tr>
40          <th scope="col">ID</th>
41          <th scope="col">Name</th>
42          <th scope="col">Category</th>
43          <th scope="col"><i class="fa fa-inr"></i> Price</th>
44          <!-- <th scope="col">Add to cart <i class='fa fa-cart-plus' aria-hidden=
45      </tr>
46      </thead>
47      <tbody>
48
49      <tr class="table-info" th:each="product : ${products}">
50          <td th:text="${product.id}"></td>
51          <td th:text="${product.productName}"></td>
52          <td th:text="${product.category}"></td>
53          <td th:text="${product.productPrice}"><i class="fa fa-inr"></i></td>
54          <!-- <td><a th:href="@{/addToCart/(id=${product.id})}">Add to cart <i cl
55      <td>
56              <input type="hidden" th:value="${product.id}" name="id">
57
58          </td>
59      </tr>
60
61      </tbody>
62      </table>
63      </div>
64      <br>
65      <br>
66      <br>
67
68

```



```

42         <th scope="col">Category</th>
43         <th scope="col"><i class="fa fa-inr"></i> Price</th>
44         <!-- <th scope="col">Add to cart <i class='fa fa-cart-plus' aria-hidden=
45         </tr>
46     </thead>
47     <tbody>
48
49     <tr class="table-info" th:each="product : ${products}">
50         <td th:text="${product.id}"></td>
51         <td th:text="${product.productName}"></td>
52         <td th:text="${product.category}"></td>
53         <td th:text="${product.productPrice}"><i class="fa fa-inr"></i></td>
54         <!-- <td><a th:href="@{/addToCart/(id=${product.id})}">Add to cart <i cl
55         <td>
56             <input type="hidden" th:value="${product.id}" name="id">
57
58         </td>
59     </tr>
60
61     </tbody>
62 </table>
63 </div>
64 <br>
65 <br>
66 <br>
67
68     <!-- for footer -->
69     <div class="push"></div>
70 </div>
71
72     <!-- footer -->
73     <div th:replace="fragment :: html_footer"></div>
74 </body>
75 </html>
76

```

login.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4   <meta charset="ISO-8859-1">
5   <title>Login Page</title>
6 </head>
7   .page-footer, .push{
8     height: 35px;
9   }
10 </body>
11 </html>
12 <!-- header -->
13 <div th:replace="fragment :: html_header"></div>
14 
15 <!-- dynamic content body -->
16 <!-- th:style="'background-image:url('@{/image/sneakers.jpg}'); background-size:">
17 <div class="container-fluid reg_bck" >
18   <!-- message alert -->
19   <div th:align="center" th:if="${message}" th:text="${message}" th:class="${message}">
20     <div class="row">
21       <div class="col-md-4 offset-md-4">
22         <div class="card mt-4 mb-4" style="background-color: #e2e2e2;">
23           <div class="card-header text-center bg-custom">
24             <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
25             <h1>Login Here!!</h1>
26           </div>
27           <div class="card-body">
28             <form th:action="@{/loginProcess}" method="post" th:object=
29               <div class="form-group">
30                 <label for="exampleInputEmail1">Email address</label>
31                 <input type="email" class="form-control"
32                   name="email"
33                   id="exampleInputEmail1"

```

```

34         name="email"
35         id="exampleInputEmail1"
36
37         th:classappend="${#fields.hasErrors('email')}? 'required'"
38         required="required"
39         aria-describedby="emailHelp"
40         placeholder="Enter Email here">
41
42         <div th:each="e:${#fields.errors('email')}}" th:title="${e.message}">
43
44         </div>
45
46     </div>
47
48     <div class="form-group">
49         <label for="exampleInputPassword1">Password</label>
50         <input type="password" class="form-control"
51             id="exampleInputPassword1"
52
53             th:classappend="${#fields.hasErrors('userPassword')}? 'required'"
54             required="required"
55             name="userPassword"
56             placeholder="Enter Password here">
57
58         <div th:each="e:${#fields.errors('userPassword')}}" th:title="${e.message}">
59
60         </div>
61
62     </div>
63     <br />
64     <button type="submit"
65         class="btn btn-success badge-pill btn-block">Submit</button>
66     <button type="reset" class="btn btn-primary badge-pill btn-block">Reset</button>
67 </form>
68 </div>

```

```

46         </div>
47
48     <div class="form-group">
49         <label for="exampleInputPassword1">Password</label>
50         type="password" class="form-control"
51         id="exampleInputPassword1"
52
53         th:classappend="${#fields.hasErrors('userPassword')
54         required="required"
55         name="userPassword"
56         placeholder="Enter Password here">
57
58         <div th:each="e:${#fields.errors('userPassword')
59
60         </div>
61
62     </div>
63     <br />
64     <button type="submit"
65         class="btn btn-success badge-pill btn-block">Submit<
66     <button type="reset" class="btn btn-primary badge-pill b
67     </form>
68 </div>
69 </div>
70 </div>
71 </div>
72
73 <!-- for footer -->
74 <div class="push"></div>
75 </div>
76
77 <!-- footer -->
78 <div th:replace="fragment :: html_footer"></div>
79 </body>
80 </html>

```

logout.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 <link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
7 <style>
8     .pg_foot, .push{
9         height: 48px;
10    }
11 </style>
12 </head>
13 <body>
14     <!-- header -->
15     <div th:replace="fragment :: html_header"></div>
16
17     <div class="container-fluid reg_bck">
18
19         <!-- message alert -->
20         <div th:align="center" th:if="${message}" th:text="${message}" th:class="${'
21
22     </div>
23
24     <!-- for footer -->
25     <div class="push"></div>
26
27     <!-- footer -->
28     <div th:replace="fragment :: html_footer"></div>
29 </body>
30 </html>
```

purchaseHistory.html

```

1 <!DOCTYPE html>
2<Ⓜhtml lang="en" xmlns:th="http://www.thymeleaf.org">
3<Ⓜhead th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Purchase History</title>
6<Ⓜstyle>
7     .page-footer, .push{
8         height: 35px;
9     }
10</style>
11</head>
12<Ⓜbody>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16<Ⓜdiv class="container-fluid home_bck">
17
18     <!-- purchase history content -->
19<Ⓜdiv th:if="${not #lists.isEmpty(purchaseHistory)}">
20         <h2 class="text-center">Purchase History List</h2>
21
22<Ⓜtable class="table">
23<Ⓜthead>
24<Ⓜtr>
25         <th>id</th>
26         <th>category</th>
27         <th>product_id</th>
28         <th>order_id</th>
29         <th>cost</th>
30         <th>total</th>
31         <th>user_id</th>
32         <th>product_name</th>
33         <th>date</th>
34     </tr>
35     <tbody>

```

```

34         </tr>
35     </thead>
36     <tr th:each="purchaseHistories : ${purchaseHistory}" >
37         <td th:text="${purchaseHistories.id}"></td>
38         <td th:text="${purchaseHistories.category}"></td>
39         <td th:text="${purchaseHistories.productId}"></td>
40         <td th:text="${purchaseHistories.orderId}"></td>
41         <td th:text="${purchaseHistories.cost}"></td>
42         <td th:text="${purchaseHistories.total}"></td>
43         <td th:text="${purchaseHistories.UserId}"></td>
44         <td th:text="${purchaseHistories.productName}"></td>
45         <td th:text="${purchaseHistories.date}"></td>
46     </tr>
47
48
49     </table>
50
51
52
53 </div>
54
55 <div th:unless="${not #lists.isEmpty(purchaseHistory)}">
56     <h2 class="text-center">Purchase History List</h2>
57
58     <table class="table">
59         <thead>
60             <tr>
61                 <th>id</th>
62                 <th>category</th>
63                 <th>product_id</th>
64                 <th>order_id</th>
65                 <th>cost</th>
66                 <th>total</th>
67                 <th>user_id</th>
68                 <th>product_name</th>

```

```

65         <th>cost</th>
66         <th>total</th>
67         <th>user_id</th>
68         <th>product_name</th>
69         <th>date</th>
70     </tr>
71 </thead>
72 <tr th:each="purchaseHistories : ${purchaseHistory}" >
73     <td th:text="${purchaseHistories.id}"></td>
74     <td th:text="${purchaseHistories.category}"></td>
75     <td th:text="${purchaseHistories.productId}"></td>
76     <td th:text="${purchaseHistories.orderId}"></td>
77     <td th:text="${purchaseHistories.cost}"></td>
78     <td th:text="${purchaseHistories.total}"></td>
79     <td th:text="${purchaseHistories.UserId}"></td>
80     <td th:text="${purchaseHistories.productName}"></td>
81     <td th:text="${purchaseHistories.date}"></td>
82
83
84 </tr>
85 </table>
86
87
88 </div>
89
90
91
92 <!-- for footer -->
93 <div class="push"></div>
94 </div>
95
96 <!-- footer -->
97 <div th:replace="fragment :: html_footer"></div>
98 </body>
99 </html>

```


searchedUser.html

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Searched User</title>
6 <style>
7     .page-footer, .push{
8         height: 35px;
9     }
10
11 </style>
12 </head>
13 <body>
14     <!-- header -->
15     <div th:replace="fragment :: html_header"></div>
16
17     <div class="container-fluid home_bck">
18
19         <div th:if="${not #lists.isEmpty(user)}">
20             <h2 class="text-center">User list</h2>
21
22             <table class="table">
23                 <thead>
24                     <tr>
25                         <th>User_Id</th>
26                         <th>User_Email</th>
27                         <th>User_Name</th>
28                         <th>Password</th>
29
30                     </tr>
31                 </thead>
32                 <tr class="table-info" th:each="user : ${user}">
33                     <td th:text="${user.id}"></td>
34                     <td th:text="${user.email}"></td>
35                     <td th:text="${user.userName}"></td>
```

```
24<tr>
25    <th>User_Id</th>
26    <th>User_Email</th>
27    <th>User_Name</th>
28    <th>Password</th>
29
30</tr>
31</thead>
32<tr class="table-info" th:each="user : ${user}">
33    <td th:text="${user.id}"></td>
34    <td th:text="${user.email}"></td>
35    <td th:text="${user.userName}"></td>
36    <td th:text="${user.userPassword}"></td>
37
38</tr>
39</table>
40
41
42</div>
43
44<!-- for footer -->
45    <div class="push"></div>
46</div>
47
48<!-- footer -->
49    <div th:replace="fragment :: html_footer"></div>
50</body>
51</html>
```

signup.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Signup page</title>
6 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
7 rel="stylesheet"
8 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
9 crossorigin="anonymous">
10 <link rel="stylesheet" th:href="@{/css/allStyles.css}"/>
11 <style>
12     .pg_foot, .push{
13         height: 48px;
14     }
15
16 </style>
17 </head>
18 <body>
19     <!-- header -->
20     <div th:replace="fragment :: html_header"></div>
21
22     <!-- dynamic content body -->
23     <!-- th:style="'background-image:url('@{/image/sneakers.jpg}+'); background-size:cover'" -->
24     <div class="container-fluid reg_bck" >
25         <!-- message alert -->
26         <div th:align="center" th:if="${message}" th:text="${message}" th:class="${message}">
27
28         <div class="row">
29             <div class="col-md-4 offset-md-4">
30                 <div class="card mt-4 mb-4" style="background: #e2e2e2;">
31                     <div class="card-header text-center bg-custom">
32                         <i class="fa fa-user-circle fa-3x" aria-hidden="true"></i>
33                         <h1>Sign Up Here!!</h1>
34                     </div>
35                     <div class="card-body">

```

```

34         </div>
35     <div class="card-body">
36         <form th:action="@{/process}" method="post" th:object="${user}" -->
37             <!-- th:value="${user.userName}" -->
38             <div class="form-group">
39                 <label for="exampleInputUsername1">Username</label>
40                 <input type="text" class="form-control"
41                     id="exampleInputUsername1"
42
43                     th:classappend="${#fields.hasErrors('userName')}?
44                         required="required"
45                         name="userName"
46                         placeholder="Enter Username here">
47                 <div th:each="e:${#fields.errors('userName')}}" th:t
48
49             </div>
50         </div>
51
52         <!-- th:value="${user.email}" -->
53         <div class="form-group">
54             <label for="exampleInputEmail1">Email address</label>
55             <input type="email" class="form-control"
56                 name="email"
57                 id="exampleInputEmail1"
58
59                 th:classappend="${#fields.hasErrors('email')}?
60                     required="required"
61                     aria-describedby="emailHelp"
62                     placeholder="Enter Email here">
63
64             <div th:each="e:${#fields.errors('email')}}" th:t
65
66         </div>
67
68     </div>

```

```

69
70 <!-- th:value="{user.userPassword}" -->
71 <div class="form-group">
72   <label for="exampleInputPassword1">Password</label>
73   type="password" class="form-control"
74   id="exampleInputPassword1"
75
76   th:classappend="{#fields.hasErrors('userPassword')
77   required="required"
78   name="userPassword"
79   placeholder="Enter Password here">
80
81   <div th:each="e:{#fields.errors('userPassword')
82
83   </div>
84
85   </div>
86   <br />
87   <button type="submit"
88     class="btn btn-success badge-pill btn-block">Submit
89   <button type="reset" class="btn btn-primary badge-pill l
90   </form>
91   </div>
92   </div>
93   </div>
94   </div>
95
96   <!-- for footer -->
97   <div class="push"></div>
98 </div>
99
100 <!-- footer -->
101 <div th:replace="fragment :: html_footer"></div>
102 </body>
103 </html>

```

userNotFound.html

```

1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head th:replace="fragment :: html_headtag">
4 <meta charset="ISO-8859-1">
5 <title>Oops user not found</title>
6 <style>
7     .page-footer, .push{
8         height: 35px;
9     }
10 </style>
11 </head>
12 <body>
13     <!-- header -->
14     <div th:replace="fragment :: html_header"></div>
15
16     <div class="container-fluid reg_bck">
17         <!-- message alert -->
18         <div th:align="center" th:if="${message}" th:text="${message}" th:class="${
19
20         <div class="row">
21             <div class="col-md-4 offset-md-4">
22                 <div class="card mt-5">
23                     <div class="card-header text-center text-white bg-custom">
24
25                         <i class="fa fa-user-secret fa-5x" aria-hidden="true"><
26
27                     </div>
28                     <div class="card-body" style="background-color: #F7F8BB;"
29
30                         <div class="form-group">
31                             <label for="exampleInputEmail1">Signup Here</la
32                             <a href="signup">if new user then please create
33                         </div>
34                     <div class="form-group">
35                         <label for="exampleInputPassword1">Login Here</

```

```

20<div class="row">
21  <div class="col-md-4 offset-md-4">
22    <div class="card mt-5">
23      <div class="card-header text-center text-white bg-custom">
24
25        <i class="fa fa-user-secret fa-5x" aria-hidden="true"></i>
26
27      </div>
28      <div class="card-body" style="background-color: #F7FBBB;">
29
30        <div class="form-group">
31          <label for="exampleInputEmail1">Signup Here</label>
32          <a href="signup">if new user then please create</a>
33        </div>
34        <div class="form-group">
35          <label for="exampleInputPassword1">Login Here</label>
36          <a href="login">try login again</a>
37        </div>
38
39      </div>
40    </div>
41  </div>
42</div>
43
44  <!-- for footer -->
45  <div class="push"></div>
46</div>
47
48  <!-- footer -->
49  <div th:replace="fragment :: html_footer"></div>
50</body>
51</html>

```

- **APPLICATION.PROPERTIES**

application.properties

```
1 spring.datasource.name=sportyshoes
2 spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoes
3 spring.datasource.username=root
4 spring.datasource.password=root
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
6 spring.jpa.hibernate.ddl-auto=update
7 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
8
9
10 #spring.thymeleaf.cache=false
11 #spring.thymeleaf.suffix: .html
12
13 #user.type= normal
14
15
16 #multipart configuration
17
18     #enabling multipart uploads
19     spring.servlet.multipart.enabled=true
20
21     #max file size
22     spring.servlet.multipart.max-file-size=300MB
23
24     #threshold after which files are written to disk
25     spring.servlet.multipart.file-size-threshold=1KB
```


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

pom.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/x
4     <modelVersion>4.0.0</modelVersion>
5 <parent>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-parent</artifactId>
8     <version>2.6.0</version>
9     <relativePath/> <!-- lookup parent from repository -->
10 </parent>
11 <groupId>com.sportyShoes</groupId>
12 <artifactId>SportyShoes_E-commerce</artifactId>
13 <version>0.0.1-SNAPSHOT</version>
14 <name>SportyShoes_E-commerce</name>
15 <description>Demo project for Spring Boot</description>
16 <properties>
17     <java.version>16</java.version>
18 </properties>
19 <dependencies>
20     <dependency>
21         <groupId>org.springframework.boot</groupId>
22         <artifactId>spring-boot-starter-data-jpa</artifactId>
23     </dependency>
24     <dependency>
25         <groupId>org.springframework.boot</groupId>
26         <artifactId>spring-boot-starter-thymeleaf</artifactId>
27     </dependency>
28     <dependency>
29         <groupId>org.springframework.boot</groupId>
30         <artifactId>spring-boot-starter-web</artifactId>
31     </dependency>
32
33     <dependency>
```

```
33- <dependency>
34     <groupId>org.springframework.boot</groupId>
35     <artifactId>spring-boot-devtools</artifactId>
36     <scope>runtime</scope>
37     <optional>true</optional>
38 </dependency>
39- <dependency>
40     <groupId>mysql</groupId>
41     <artifactId>mysql-connector-java</artifactId>
42     <scope>runtime</scope>
43 </dependency>
44
45 <!-- https://mvnrepository.com/artifact/javax.validation/validation-api -->
46- <dependency>
47     <groupId>javax.validation</groupId>
48     <artifactId>validation-api</artifactId>
49 </dependency>
50
51 <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->
52- <dependency>
53     <groupId>org.hibernate</groupId>
54     <artifactId>hibernate-validator</artifactId>
55     <version>6.1.5.Final</version>
56 </dependency>
57
58 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
59- <dependency>
60     <groupId>org.projectlombok</groupId>
61     <artifactId>lombok</artifactId>
62     <scope>provided</scope>
63 </dependency>
64
65- <dependency>
```

```
50      </dependency>
51      <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->
52      <dependency>
53          <groupId>org.hibernate</groupId>
54          <artifactId>hibernate-validator</artifactId>
55          <version>6.1.5.Final</version>
56      </dependency>
57
58      <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
59      <dependency>
60          <groupId>org.projectlombok</groupId>
61          <artifactId>lombok</artifactId>
62          <scope>provided</scope>
63      </dependency>
64
65      <dependency>
66          <groupId>org.springframework.boot</groupId>
67          <artifactId>spring-boot-starter-test</artifactId>
68          <scope>test</scope>
69      </dependency>
70  </dependencies>
71
72  <build>
73      <plugins>
74          <plugin>
75              <groupId>org.springframework.boot</groupId>
76              <artifactId>spring-boot-maven-plugin</artifactId>
77          </plugin>
78      </plugins>
79  </build>
80
81 </project>
82
```