

```
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/XMLSchema-instance"
3    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
4    <modelVersion>4.0.0</modelVersion>
5    <parent>
6        <groupId>org.springframework.boot</groupId>
7        <artifactId>spring-boot-starter-parent</artifactId>
8        <version>2.7.1</version>
9        <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.sportyshoe</groupId>
12    <artifactId>sportyshoe</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>sportyShoe</name>
15    <description>SimplieLearn project for Spring Boot</description>
16    <properties>
17        <java.version>17</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-web</artifactId>
27        </dependency>
28        <dependency>
29            <groupId>mysql</groupId>
30            <artifactId>mysql-connector-java</artifactId>
31            <scope>runtime</scope>
32        </dependency>
33        <!-- https://mvnrepository.com/artifact/com.h2database/h2 -->
34        <dependency>
35            <groupId>com.h2database</groupId>
36            <artifactId>h2</artifactId>
```

```
37         </dependency>
38
39         <dependency>
40             <groupId>javax.servlet</groupId>
41             <artifactId>jstl</artifactId>
42         </dependency>
43         <!-- Tomcat for JSP rendering -->
44         <dependency>
45             <groupId>org.apache.tomcat.embed</groupId>
46             <artifactId>tomcat-embed-jasper</artifactId>
47             <scope>provided</scope>
48         </dependency>
49
50         <dependency>
51             <groupId>org.springframework.boot</groupId>
52             <artifactId>spring-boot-starter-test</artifactId>
53             <scope>test</scope>
54         </dependency>
55     </dependencies>
56
57     <build>
58         <plugins>
59             <plugin>
60                 <groupId>org.springframework.boot</groupId>
61                 <artifactId>spring-boot-maven-plugin</artifactId>
62             </plugin>
63         </plugins>
64     </build>
65
66 </project>
67
```

```
1 server.port=8080
2 spring.datasource.username=root
3 spring.datasource.password=root123456
4 spring.datasource.url=jdbc:mysql://localhost:3306/sportyshoe
5 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
6
7 spring.jpa.hibernate.ddl-auto=update
8 spring.jpa.show-sql=true
9
10 spring.mvc.view.prefix=/WEB-INF/jsp/
11 spring.mvc.view.suffix=.jsp
12
13 spring.servlet.multipart.max-file-size=5MB
14 spring.servlet.multipart.max-request-size=5MB
15
```

```
1 package com.spring.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6
7
8
9
10
11 public class SportyShoeApplication {
12
13     public static void main(String[] args) {
14         System.out.println("starting application");
15         SpringApplication.run(SportyShoeApplication.class, args);
16
17
18     }
19
20 }
21
```

```
1<%@page isELIgnored="false" %>
2<%@ taglib prefix="form"
   uri="http://www.springframework.org/tags/form"%>
3<!DOCTYPE html>
4<html lang="en">
5<head>
6 <title>login page</title>
7</head>
8<body>
9
10
11<h2 align="center">Admin Login Here </h2><br>
12<h2 align="center" style="color:red;">${invalid}</h2>
13
14<form action="/submitlogin" method="get">
15     <table align="center" style="font-size: 20px; color: black;">
16         <tr>
17             <th><br>UserName</th>
18             <th><br> <input type="text" name="userName" required
19                 /></th>
20         </tr>
21         <tr>
22             <th><br>Password</th>
23             <th><br> <input type="password" name="password" required
24                 /></th>
25         </tr>
26         <tr>
27             <th><br> <input type="submit" value="Submit" /></th>
28             <th><br> <input type="reset" value="Reset" /></th>
29         </tr>
30     </table>
31</form>
32
33
34
35
36</body>
```

login.jsp

Wednesday, 20 July, 2022, 7:32 pm

37</html>

```
1<%@page isELIgnored="false" %>
2<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
3<!DOCTYPE html>
4<html lang="en">
5<head>
6 <title>welcome page</title>
7</head>
8<body>
9 <h1> <center>Welcome to sporty shoe Admin</center> </h1><br>
10 <h2 >change Password </h2>
11 <h2 style="color:red;">${message}</h2>
12<form action="changePassword" method="get">
13 <table style="font-size: 20px; color: black;">
14 <tr>
15 <th><br>password</th>
16 <th><br> <input type="text" name="password" required
/></th>
17 </tr>
18 <tr>
19 <th><br>confirm Password</th>
20 <th><br> <input type="text" name="Cpassword" required
/></th>
21 </tr>
22 <tr>
23 <th><br> <input type="submit" value="Submit" /></th>
24 <th><br> <input type="reset" value="Reset" /></th>
25 </tr>
26 </table>
27</form>
28
29<br>
30
31<form action="product" method="get">
32 <tr>
33 <th><br> <input type="submit" value="manage products"
/></th>
34
35 </tr>
36</form>
```

welcome.jsp

Wednesday, 20 July, 2022, 7:34 pm

```
37
38<form action="user" method="get">
39    <tr>
40        <th><br> <input type="submit" value="user details"
    /></th>
41
42    </tr>
43
44
45</body>
46</html>
```



```
1<%@page isELIgnored="false" %>
2<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
3<!DOCTYPE html>
4<html lang="en">
5<head>
6 <title>product page</title>
7</head>
8<body>
9
10
11 <h1 style="color:blue;"> <center>Manage Products</center> </h1>
12 <a href="listOfProducts"><h2 style="color:blue;">1.view
   products</h2></a>
13<table >
14 <tr>
15 <td colspan=8 align="center">List of Products</td>
16 </tr>
17 <tr>
18 <th>ID &nbsp;  </th>
19 <th>Name &nbsp; </th>
20 <th>category &nbsp;</th>
21 <th>price &nbsp;</th>
22 </tr>
23
24 <c:forEach items="${products}" var="product">
25 <tr>
26
27 <td><c:out value="${product.getPid()}" /></td>
28 <td><c:out value="${product.getName()}" /></td>
29 <td><c:out value="${product.getCategory()}" /></td>
30 <td><c:out value="${product.getPrice()}" /></td>
31 <td><a href="listOfProducts">update</a></td>
32 </tr>
33 </c:forEach>
34</table>
35
36
37<br>
38
```

```
39<h2 style="color:blue;"> 2.Add Product </h2>
40<form action="addProduct" method="get">
41    <table style="font-size: 20px; color: black;">
42        <tr>
43            <th>product name</th>
44            <th><input type="text" name="Name" required /></th>
45        </tr>
46        <tr>
47            <th><br>product category</th>
48            <th><br> <input type="text" name="category" required
49        /></th>
50    </tr>
51    <tr>
52        <th><br>product price</th>
53        <th><br> <input type="text" name="price" required
54    /></th>
55    </tr>
56    <tr>
57        <th><br> <input type="submit" value="Submit" /></th>
58        <th><br> <input type="reset" value="Reset" /></th>
59    </tr>
60    </table>
61</form>
62<br>
63<h2 style="color:blue;"> 3.Delete Product </h2>
64<h3> Enter the product Id to delete the product </h3>
65<form action="deleteProduct" method="get">
66    <table style="font-size: 20px; color: black;">
67        <tr>
68            <th>product Id</th>
69            <th><input type="text" name="Id" required /></th>
70        </tr>
71        <tr>
72            <th><br> <input type="submit" value="Delete" /></th>
73            <th><br> <input type="reset" value="Reset" /></th>
74        </tr>
75    </table>
76</form>
```

```
76
77<br>
78
79<h2 style="color:blue;"> 4.Update Product </h2>
80 <h3> Enter the product Id to change its details </h3>
81<form action="updateProduct" method="get">
82     <table style="font-size: 20px; color: black;">
83         <tr>
84             <th><br>product Id</th>
85             <th><br> <input type="text" name="Id" required /></th>
86         </tr>
87         <tr>
88             <th><br>product name</th>
89             <th><br> <input type="text" name="Name" required
90         /></th>
91     </tr>
92     <tr>
93         <th><br>product category</th>
94         <th><br> <input type="text" name="category" required
95     /></th>
96     </tr>
97     <tr>
98         <th><br>product price</th>
99         <th><br> <input type="text" name="price" required
100     /></th>
101     </tr>
102     <tr>
103         <th><br> <input type="submit" value="Update" /></th>
104         <th><br> <input type="reset" value="Reset" /></th>
105     </tr>
106 </table>
107 </form>
108
109 </body>
110 </html>
```

```
1<%@page isELIgnored="false" %>
2<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
3<!DOCTYPE html>
4<html lang="en">
5<head>
6 <title>welcome page</title>
7</head>
8<body>
9<h1 style="color:blue;"> <center>Manage users</center> </h1>
10<br>
11<a href="getAllusers"><h2
    style="color:blue;">1.&u>Getallusers</h2></a>
12<table >
13    <tr>
14        <td colspan=8 align="center">List of users</td>
15    </tr>
16    <tr>
17        <th>ID &nbsp;  </th>
18        <th>Name &nbsp; </th>
19        <th>&u>timestamp &nbsp;</th>
20
21    </tr>
22
23    <c:forEach items="${users}" var="user">
24        <tr>
25
26            <td><c:out value="${user.getId()}" /></td>
27            <td><c:out value="${user.getUserName()}" /></td>
28            <td><c:out value="${user.getTimestamp()}" /></td>
29        </tr>
30    </c:forEach>
31</table>
32
33<br>
34
35<form action="searchUser" method="get">
36    <h2 style="color:blue;">2.Search user</h2>
37    <tr>
38        <th>UserName</th>
```

```
39         <th><input type="text" name="userName" required /></th>
40     </tr>
41
42     <tr>
43         <th><input type="submit" value="search" /></th>
44     </tr>
45
46     <br>
47 </form>
48
49 <table >
50     <tr>
51         <td colspan=8 align="center">users found</td>
52     </tr>
53     <tr>
54         <th>ID &nbsp;</th>
55         <th>Name &nbsp;</th>
56         <th>timestamp &nbsp;</th>
57     </tr>
58
59     <c:forEach items="${user}" var="user">
60         <tr>
61
62             <td><c:out value="${user.getId()}" /></td>
63             <td><c:out value="${user.getUserName()}" /></td>
64             <td><c:out value="${user.getTimestamp()}" /></td>
65         </tr>
66     </c:forEach>
67
68 </table>
69
70     <br>
71
72 <a href="purchase"><h2
73     style="color:blue;">3.GetPurchaseReport</h2></a>
74 <table >
75     <tr>
76         <td colspan=8 align="center">purchaseReport</td>
77     </tr>
```

```
77     <tr>
78         <th>userName &nbsp;  </th>
79         <th>productName &nbsp;  </th>
80         <th>category &nbsp;  </th>
81         <th>price &nbsp;  </th>
82         <th>timestamp &nbsp;  </th>
83     </tr>
84
85     <c:forEach items="${purchases}" var="purchases">
86         <tr>
87
88             <td><c:out value="${purchases.getUserName()}" /></td>
89             <td><c:out value="${purchases.getName()}" /></td>
90             <td><c:out value="${purchases.getCategory()}" /></td>
91             <td><c:out value="${purchases.getPrice()}" /></td>
92             <td><c:out value="${purchases.getTimestamp()}" /></td>
93         </tr>
94     </c:forEach>
95 </table>
96
97 <br>
98 <h2 style="color:blue;">4.Search Purchase</h2>
99 <form action="searchPurchase" method="get">
100     Select a Category:&nbsp;  
101
102     <select name="category">
103
104         <c:forEach items="${catlist}" var="cat">
105             <option>
106                 <c:out value="${cat}" />
107             </option>
108         </c:forEach>
109
110     </select>
111     <br>
112
113 Select a time:&nbsp;  
114     <select name="timestamp">
115
```

```
116         <c:forEach items="${timelist}" var="time">
117             <option>
118                 <c:out value="${time}"/>
119             </option>
120         </c:forEach>
121
122 </select>
123
124     <br> &nbsp; &nbsp; <input type="submit" value="Search" />
125 </form>
126
127
128 <table >
129     <tr>
130         <td colspan=8 align="center">Purchases found </td>
131     </tr>
132     <tr>
133         <th>userName &nbsp;</th>
134         <th>productName &nbsp;</th>
135         <th>category &nbsp;</th>
136         <th>price &nbsp;</th>
137         <th>timestamp &nbsp;</th>
138     </tr>
139
140     <c:forEach items="${purchase}" var="purchase">
141         <tr>
142
143             <td><c:out value="${purchase.getUserName()}" /></td>
144             <td><c:out value="${purchase.getName()}" /></td>
145             <td><c:out value="${purchase.getCategory()}" /></td>
146             <td><c:out value="${purchase.getPrice()}" /></td>
147             <td><c:out value="${purchase.getTimestamp()}" /></td>
148         </tr>
149     </c:forEach>
150 </table>
151
152 <br>
153 <br>
154
```

user.jsp

Wednesday, 20 July, 2022, 8:50 pm

155

156

157</body>

158</html>


```
1 package com.spring.demo.model;
2
3 import javax.persistence.Entity;
4
5 @Entity
6 @Table(name="Admin")
7 public class adminLogin {
8     @Id
9     @GeneratedValue(strategy = GenerationType.AUTO)
10    private long Aid;
11    private String userName;
12    private String password;
13
14
15    public long getAid() {
16        return Aid;
17    }
18    public void setAid(long aid) {
19        Aid = aid;
20    }
21    public String getUserName() {
22        return userName;
23    }
24    public void setUserName(String userName) {
25        this.userName = userName;
26    }
27    public String getPassword() {
28        return password;
29    }
30    public void setPassword(String password) {
31        this.password = password;
32    }
33
34    @Override
35    public String toString() {
36        return "adminLogin [Aid=" + Aid + ", userName=" + userName +
37        ", password=" + password + "]\n";
38    }
39 }
```

adminLogin.java

Wednesday, 20 July, 2022, 7:40 pm

43

44

45 }

46

```
1 package com.spring.demo.model;
2
3 import java.util.Date;
14 @Entity
15 @Table(name="user")
16 public class user {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private long uId;
20     private String userName;
21
22
23
24     public long getuId() {
25         return uId;
26     }
27     public void setuId(long uId) {
28         this.uId = uId;
29     }
30     private Date timestamp;
31     @OneToOne(cascade = CascadeType.ALL)
32     private product product;
33
34
35
36     public String getUsername() {
37         return userName;
38     }
39     public void setUsername(String userName) {
40         this.userName = userName;
41     }
42
43     public product getProduct() {
44         return product;
45     }
46     public void setProduct(product product) {
47         this.product = product;
48     }
49     public Date getTimestamp() {
```

```
50         return timestamp;
51     }
52     public void setTimestamp(Date timestamp) {
53
54         this.timestamp = timestamp;
55     }
56     @Override
57     public String toString() {
58         return "user [uId=" + uId + ", userName=" + userName + ",
59             timestamp=" + timestamp + ", product=" + product
60             + "]\n";
61     }
62
63
64
65
66 }
67
```

```
1 package com.spring.demo.model;
2
3 import javax.persistence.Entity;
4
5
6
7
8
9
10 @Entity
11 @Table(name="product")
12 public class product {
13     @Id
14     @GeneratedValue(strategy = GenerationType.IDENTITY)
15     private long Pid;
16     private String Name;
17     private String category;
18     private long price;
19     @OneToOne(mappedBy = "product")
20     private user user;
21
22     public long getPid() {
23         return Pid;
24     }
25     public void setPid(long pid) {
26         Pid = pid;
27     }
28     public String getName() {
29         return Name;
30     }
31     public void setName(String name) {
32         Name = name;
33     }
34     public String getCategory() {
35         return category;
36     }
37     public void setCategory(String category) {
38         this.category = category;
39     }
40     public long getPrice() {
41         return price;
42     }
43     public void setPrice(long price) {
44         this.price = price;
```

```
45     }
46     @Override
47     public String toString() {
48         return "product [Pid=" + Pid + ", Name=" + Name + ",
49             category=" + category + ", price=" + price + "]\n";
50     }
51
52
53 }
54
```

```
1 package com.spring.demo.model;
2
3 import java.util.Date;
4
5 public class purchase {
6     private String userName;
7     private String Name;
8     private String category;
9     private long price;
10    private Date timestamp;
11
12
13
14    public purchase(String userName, String name, String category,
15        long price, Date timestamp) {
16        super();
17        this.userName = userName;
18        Name = name;
19        this.category = category;
20        this.price = price;
21        this.timestamp = timestamp;
22    }
23    public String getUserName() {
24        return userName;
25    }
26    public void setUserName(String userName) {
27        this.userName = userName;
28    }
29
30    public Date getTimestamp() {
31        return timestamp;
32    }
33    public void setTimestamp(Date timestamp) {
34        this.timestamp = timestamp;
35    }
36    public String getName() {
37        return Name;
38    }
```

```
39     public void setName(String name) {
40         Name = name;
41     }
42     public String getCategory() {
43         return category;
44     }
45     public void setCategory(String category) {
46         this.category = category;
47     }
48     public long getPrice() {
49         return price;
50     }
51     public void setPrice(long price) {
52         this.price = price;
53     }
54
55     @Override
56     public String toString() {
57         return "purchase [userName=" + userName + ", Name=" + Name +
58             ", category=" + category + ", price=" + price
59             + ", timestamp=" + timestamp + "];"
60     }
61
62 }
63
```



```
1 package com.spring.demo.controller;
2
3 import java.util.List;
16
17 @Controller
18 public class adminController {
19     @Autowired
20     private adminService service;
21     @Autowired
22     private userService userService;
23
24
25
26     @RequestMapping("/login")
27     public String login(Model model)
28     {
29         System.out.println("login page");
30
31         return "login";
32     }
33
34     @RequestMapping("/submitlogin")
35     public String submitlogin(@ModelAttribute adminLogin admin,
36                               Model model)
37     {
38         boolean b=service.validateAdmin(admin);
39         if(b) {
40             return "welcome";
41         }else {
42             model.addAttribute("invalid", "invalid credentials
43             please try again");
44             return "login";
45         }
46     }
47
48     @RequestMapping("/changePassword")
49     public String changePassword(@RequestParam("password") String
50     p1,@RequestParam("Cpassword") String p2, Model model)
51     {
```

```
49         boolean b=false;
50         if(p1.equals(p2)) {
51             b=service.updatePassword(p1);
52             if(b) {
53                 model.addAttribute("message","password updated
successfully...!!");
54                 return "welcome";
55             }
56             else {
57                 model.addAttribute("message","updation
failed");
58                 return "welcome";
59             }
60         }
61         else
62             model.addAttribute("message","check the password
entered...");
63         return "welcome";
64     }
65
66 }
67
```

```
1 package com.spring.demo.controller;
2
3
4 import java.sql.Timestamp;
28 @Controller
29 public class UserController {
30
31     @Autowired
32     private UserService userService;
33
34
35
36
37     @RequestMapping("/getAllusers")
38     public String getAllusers(Model model)
39     {
40
41         List<user> users=userService.getAllusers();
42
43         model.addAttribute("users", users);
44         return "user";
45     }
46
47
48     @RequestMapping("/searchUser")
49     public String searchUser(@RequestParam("userName") String uN,
50     Model model)
51     {
52         ArrayList<user> u=userService.searchUser(uN);
53         model.addAttribute("user", u);
54         return "user";
55     }
56
57     @RequestMapping("/purchase")
58     public String purchase(Model model)
59     {
60         List<purchase> purchases=userService.purchase();
61         model.addAttribute("purchases", purchases);
```

```
62         return "user";
63
64     }
65
66     @RequestMapping("/searchPurchase")
67     public String searchPurchase(@RequestParam("category") String
        cat,@RequestParam("timestamp") String time, Model model)
68     {
69         Timestamp timestamp = null;
70         timestamp=timestamp.valueOf(time);
71         System.out.println("+++++date"+timestamp);
72         List<purchase> purchase=service.searchpurchase
        (cat,timestamp);
73         model.addAttribute("purchase", purchase);
74         return "user";
75
76     }
77
78
79     @RequestMapping("/user")
80     public String user(Model model) {
81         List<String> catlist = service.listcat();
82         model.addAttribute("catlist", catlist);
83         System.out.println(catlist);
84         ArrayList<Date> timelist=service.listtime();
85         model.addAttribute("timelist", timelist);
86         System.out.println(timelist);
87         return "user";
88     }
89
90
91
92 }
93
```

```
1 package com.spring.demo.controller;
2
3
4
5 import java.util.List;
19
20
21 @Controller
22 public class productController {
23
24     @Autowired
25     private productService pservice;
26
27     @RequestMapping("/listOfProducts")
28     public String getallproducts(Model model) {
29         List<product> products=pservice.getAllproducts();
30         model.addAttribute("products", products);
31         return "product";
32     }
33
34     @RequestMapping("/addProduct")
35     public void addProduct(@ModelAttribute product product, Model
model, HttpSession session)
36     {
37         session.setAttribute("product", product);
38         pservice.addproduct(product);
39     }
40
41
42     @RequestMapping("/deleteProduct")
43     public String deleteProduct(@RequestParam("Id") long id, Model
model) {
44         pservice.deleteProduct(id);
45
46         return "product";
47     }
48
49     @RequestMapping("/updateProduct")
50     public String updateProduct(@ModelAttribute("product") product
```

```
    pro,@RequestParam("Id") long id, Model model) {  
51         pservice.updateProduct(pro,id);  
52  
53         return "product";  
54     }  
55  
56     @RequestMapping("/product")  
57     public String Product() {  
58  
59  
60         return "product";  
61     }  
62  
63  
64 }  
65
```

```
1 package com.spring.demo.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
9
10 @Service
11 public class adminService {
12     @Autowired
13     private adminRepository shoeRepo;
14
15     public adminLogin insertAdmin(adminLogin admin) {
16
17         adminLogin i=shoeRepo.save(admin);
18
19         return i;
20     }
21
22     public boolean validateAdmin(adminLogin admin) {
23         adminLogin repoAdmin=shoeRepo.findById((long) 1).get();
24         if((repoAdmin.getUserName()).equalsIgnoreCase
25         (admin.getUserName())&&((repoAdmin.getPassword()).equalsIgnoreCase
26         (admin.getPassword()))){
27             return true;
28         }
29         else{
30             return false;
31         }
32     }
33
34     public boolean updatePassword(String p1) {
35         adminLogin repoAdmin=shoeRepo.findById((long) 1).get();
36         repoAdmin.setPassword(p1);
37         boolean b=shoeRepo.save(repoAdmin) != null;
38         return b;
39     }
40
41 }
42
```

```
1 package com.spring.demo.service;
2
3 import java.util.ArrayList;
16 @Service
17 public class userService {
18     @Autowired
19     private userRepository userRepo;
20
21
22     public void insertUser(user u) {
23
24         userRepo.save(u);
25     }
26
27     public List<user> getallusers() {
28         List<user> users=userRepo.findAll();
29         System.out.println(users);
30         return users;
31     }
32
33     public ArrayList<user> searchUser(String uN) {
34         ArrayList<user> u=userRepo.findByName(uN);
35         System.out.println(u);
36         return u;
37     }
38 }
39
40 public List<purchase> purchase() {
41
42     List<purchase> purchases=userRepo.purchase();
43     for(purchase p:purchases) {
44         System.out.println(p);
45     }
46
47     return purchases;
48 }
49
50 public List<String> listcat() {
51
```



```
52     List<purchase> cats=userRepo.getCategories();
53     ArrayList<String> catlist= new ArrayList<>();
54     for(purchase c:cats) {
55         String cat=c.getCategory();
56         System.out.println(cat);
57         catlist.add(cat);
58     }
59     return catlist;
60 }
61
62 public ArrayList<Date> listtime() {
63     List<purchase> times=userRepo.getTimestamps();
64     ArrayList<Date> timelist= new ArrayList<>();
65     for(purchase t:times) {
66         Date time=t.getTimestamp();
67         System.out.println(time);
68         timelist.add(time);
69     }
70     return timelist;
71 }
72
73 public List<purchase> searchpurchase(String cat, Date date) {
74
75     List<purchase> purchase=userRepo.searchPurchase(cat,date);
76     for(purchase p:purchase) {
77         System.out.println(p);
78     }
79
80     return purchase;
81 }
82
83
84
85
86
87
88 }
89
90
```

userService.java

Wednesday, 20 July, 2022, 7:45 pm

91

```
1 package com.spring.demo.service;
2
3 import java.util.List;
14
15 @Service
16 public class productService {
17
18
19     @Autowired
20     private productRepository proRepo;
21
22
23
24     public List<product> getAllproducts() {
25         List<product> products
26             = proRepo.findAll();
27         System.out.println(products);
28         return products;
29
30     }
31
32     public void addproduct(product product) {
33
34         product p=proRepo.save(product);
35         System.out.println(p);
36     }
37
38
39     public void deleteProduct(long id) {
40         proRepo.deleteById(id);
41
42     }
43
44     public void updateProduct(product pro, long id) {
45         product repoPro=proRepo.getById(id);
46         repoPro.setName(pro.getName());
47         repoPro.setCategory(pro.getCategory());
48         repoPro.setPrice(pro.getPrice());
49         product p=proRepo.save(repoPro);
```

```
50         System.out.println(p);
51     }
52
53
54 }
55
```

```
1 package com.spring.demo.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 @Repository
6 public interface adminRepository extends JpaRepository<adminLogin,
7     Long> {
8
9 }
10
11
12
```

```
1 package com.spring.demo.repository;
2
3 import java.util.ArrayList;
13
14
15 @Repository
16 public interface userRepository extends JpaRepository<user, Long> {
17
18     @Query("SELECT u FROM user u WHERE u.userName = :name")
19     ArrayList<user> findByName(@Param("name") String uN);
20
21     @Query("SELECT new com.spring.demo.model.purchase(u.userName,
        p.Name, p.category, p.price ,u.timestamp) FROM user u JOIN u.product
        p ")
22     public List<com.spring.demo.model.purchase> getCategories();
23
24     @Query("SELECT new com.spring.demo.model.purchase(u.userName,
        p.Name, p.category, p.price ,u.timestamp) FROM user u JOIN u.product
        p ")
25     public List<com.spring.demo.model.purchase> getTimestamps();
26
27     @Query("SELECT new com.spring.demo.model.purchase(u.userName,
        p.Name, p.category, p.price ,u.timestamp) FROM user u JOIN u.product
        p WHERE p.category = :cat AND u.timestamp= :time")
28     public List<com.spring.demo.model.purchase> searchPurchase
        (@Param("cat") String cat,@Param("time") Date date);
29
30     @Query("SELECT new com.spring.demo.model.purchase(u.userName,
        p.Name, p.category, p.price ,u.timestamp) FROM user u JOIN u.product
        p")
31     public List<com.spring.demo.model.purchase> purchase();
32 }
33
```

```
1 package com.spring.demo.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
10
11
12 @Repository
13 public interface productRepository extends JpaRepository<product,
    Long> {
14
15
16 }
17
```