

T.C.

ONDOKUZ MAYIS ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Eticaret Portalı

Hazırlayan

**14060292 – Tunahan Yetimoğlu**

**14060320 – Ömer Özen**

**14060304 – Cafer Ümit Salman**

Danışman

**Yrd. Doç. Dr.- İSMAİL İŞERİ**

**MAYIS/2018  
SAMSUN**

## **ÖZET**

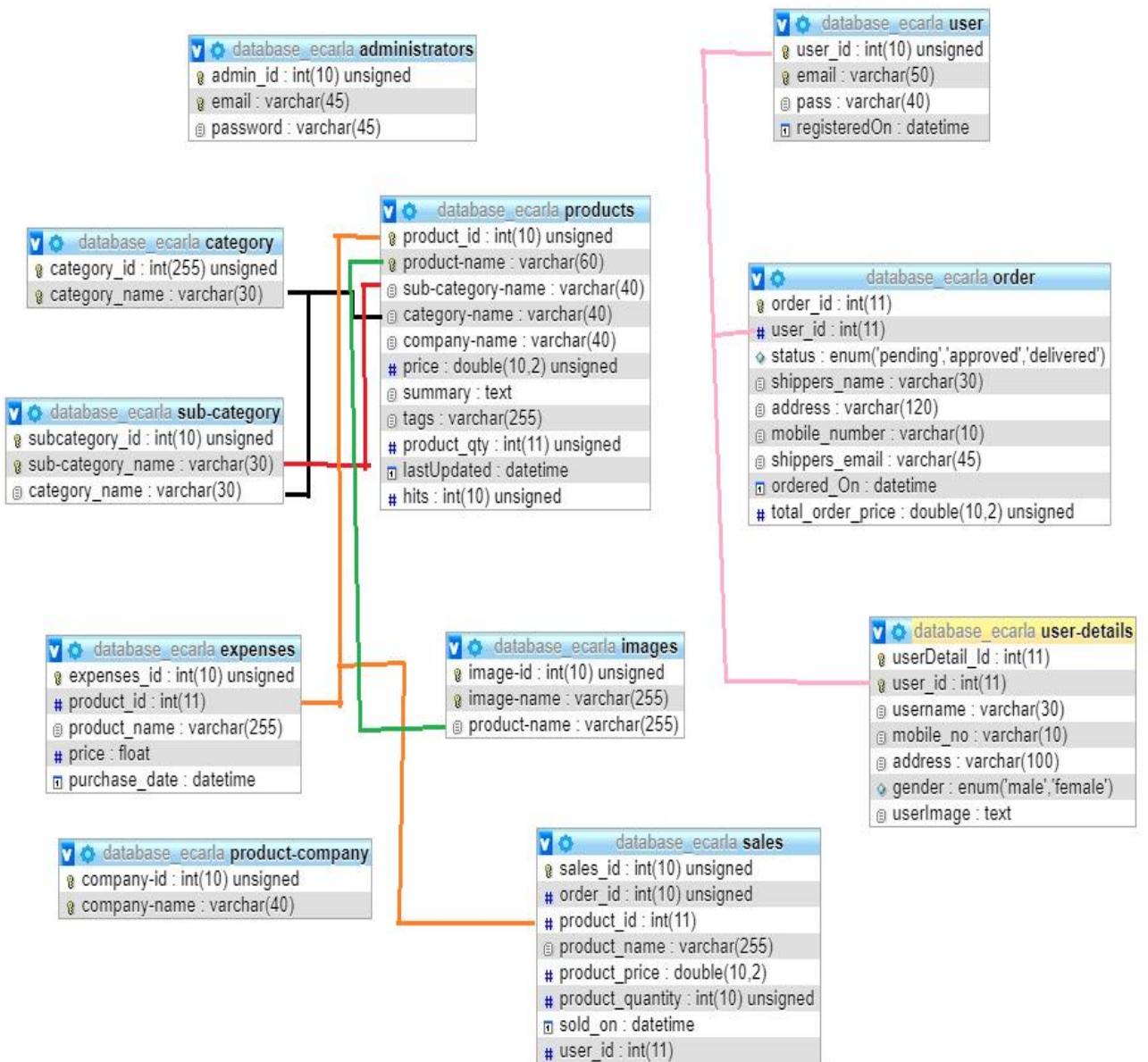
Oluşturduğumuz eCarla.com web sitesi, online olarak araç satışının yapılabilmesi amacıyla oluşturulmuştur. Sitede üyelik sistemi bulunuyor ve üye olan kullanıcılar aşağıdaki yetkilere sahip oluyor:

- Üyelik bilgilerini güncelleme
- Alışveriş sepetine ürün ekleme-çıkarma
- Sipariş verme
- Siparişi takip etme
- Fatura yazdırma

Uygulamamızda bir de üst düzey yetkilere sahip olan, özellikle site yöneticisi için oluşturulmuş yönetici kullanıcı bulunmaktadır. Yöneticinin sahip olduğu yetkiler ise aşağıdaki gibidir:

- Kullanıcıları admin yapabilme
- Kullanıcıların şifrelerini değiştirme
- Siparişleri listeleme, onaylama, takip etme
- Geçmiş siparişleri görüntüleme
- Satış bilgilerini grafiksel olarak görüntüleyebilme(çizgisel ve dairesel grafikler)
- Stoğa yeni ürün ekleme-çıkarma-güncelleme
- Stokta sayısı azalan ürünler için uyarı görüntüleme

## ER Diyagramı



## Uygulama Arayüzü

### ❖ Sitenin ana sayfası

The screenshot shows the homepage of the eCarla website. At the top, there's a dark header with the logo "eCarla" and links for "Admin Login" and "Join Now!". Below the header is a teal-colored navigation bar with a search bar containing "Quick search an item ...", and links for "HOME", "PRODUCTS", and "CONTACT". A promotional message in white text on a dark background reads: "eCarla is one of the best places to have your cars for any material We are the only one to have your thoughts at the right place... For More Information Call Us : +90-5320522830". To the right of this message is a blue button labeled "Follow me!". The main content area features a large, blurred image of a silver Audi R8 convertible driving on a road. To the right of the car is a smaller inset image of a Mercedes-Benz dashboard with the text "Mercedes.. Mercedes-Benz. The future of the Automobile." Below the main image is a sidebar with a light gray background and a teal header titled "CATEGORIES". The categories listed are "TÜPLÜ", "BENZİNLİ", "ARAZİ ARACI", "DİZEL", and "4x4". To the right of the sidebar is a section titled "En çok sevilen araçlar beybi" (Most loved cars) featuring eight small images of different cars (Nissan Juke, Audi A8, Audi A5, BMW M4, Mazda Miata, Cadillac xt5, Tesla Model 3, and Mercedez-Benz A180) each with its name, manufacturer, and price.

Quick search an item ...

HOME PRODUCTS CONTACT

eCarla is one of the best places to have your cars for any material We are the only one to have your thoughts at the right place... For More Information Call Us : +90-5320522830

Follow me!

Mercedes..

Mercedes-Benz. The future of the Automobile.

CATEGORIES

TÜPLÜ

BENZİNLİ

ARAZİ ARACI

DİZEL

4x4

En çok sevilen araçlar beybi

Nissan Juke By Nissan 4x4 £. 600.00	Audi A8 By Audi 4x4 £. 700.00	Audi A5 By Audi 4x4 £. 4500.00	BMW M4 By BMW Dizel £. 50.00
Mazda Miata By Mazda 4x4 £. 150.00	Cadillac xt5 By Cadillac Dizel £. 100.00	Tesla Model 3 By Tesla Tesla £. 460.00	Mercedez-Benz A180 By Mercedez-Benz 4x4 £. 300.00

❖ Admin için giriş paneli

The screenshot shows the eCarla website's admin login interface. At the top, there is a dark header bar with the 'eCarla' logo on the left and 'Admin Login' and 'Join Now!' buttons on the right. Below the header is a teal-colored navigation bar with a search bar containing 'Quick search an item ...', and menu items 'HOME', 'PRODUCTS', and 'CONTACT'. A promotional message in white text on the teal bar reads: 'eCarla is one of the best places to have your cars for any material We are the only one to have your thoughts at the right place... For More Information Call Us : +90-5320522830'. To the right of this message is a blue button labeled 'Follow me!!'. The main content area is titled 'Administrator Login' in bold black text. It contains two input fields: 'Email' with the value 'ehab@gmail.com' and 'Password' with several dots indicating the password. Below these fields is a large blue 'Login' button.

❖ Müşteriler için giriş paneli

The screenshot displays the eCarla website's customer section. At the top, it features the 'eCarla' logo and 'Admin Login' and 'Join Now!' buttons. The main area is divided into two sections by a vertical line. On the left, under the heading 'Login...', there are fields for 'Email' (containing 'ehab@gmail.com') and 'Password' (containing several dots). Below these is a blue 'Login!' button. On the right, under the heading 'Register Now!!', there are three fields: 'Email' (containing 'register@abc.com'), 'Password' (containing several dots), and 'Password Again' (containing several dots). Below these fields is a blue 'Register' button.

❖ Müşteri hesap ayarları

The screenshot shows a user profile interface. On the left, a sidebar menu lists "USER PROFILE", "ACCOUNT", "MY ORDERS", and "SETTINGS". The main area displays the following sections:

- Hello ehab@gmail.com**: Personal information entry field with placeholder text: "Enter in the personal information for your Account to have quick checkouts during any transaction".
- My Orders**: A table header with columns: Order No, Item, Order Summary (Price x Quantity), and Date. Below the header, there is a large black redacted area.
- User Account**: Form fields for Name (text input), Gender (dropdown menu set to "Male"), Mobile No (text input), Address (text input), and City (dropdown menu set to "İstanbul"). A note below the city field states: "We Do not accept any orders outside Istanbul". A blue "Add Details" button is at the bottom.
- Settings**: Form fields for Email (text input containing "ehab@gmail.com") and Password (text input). A blue "Change My Password" button is at the bottom.

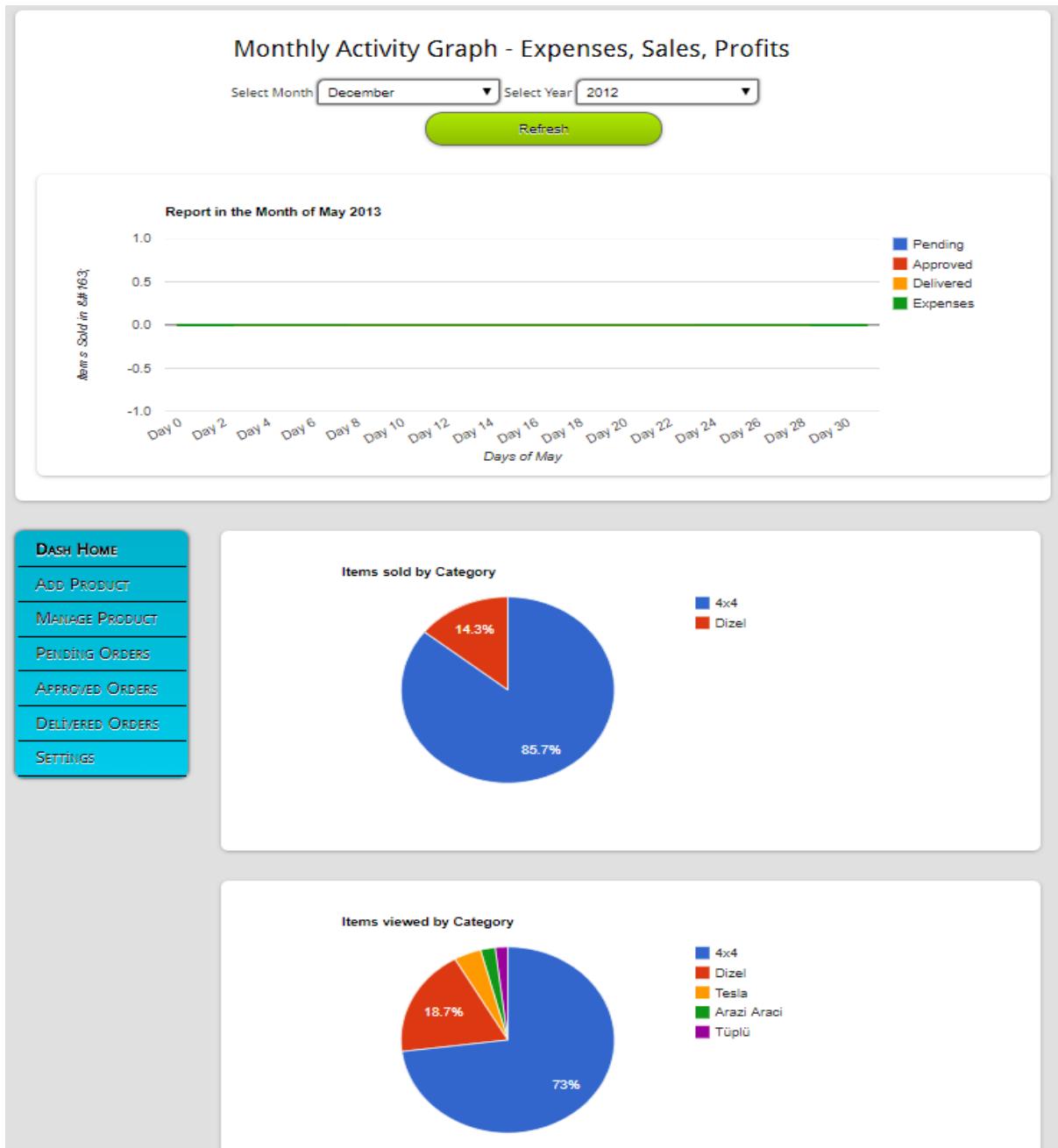
❖ Müşterinin sepeti

The screenshot shows a shopping cart summary. At the top, it says "Cars In your Cart" and "Your Cart contains following...". Below that, it lists "3 Nissan Juke added !! with price of 600.0" and "Total value price of the cart 600.0". The cart table has columns: Name Of Product, Price, and Quantity. The data is as follows:

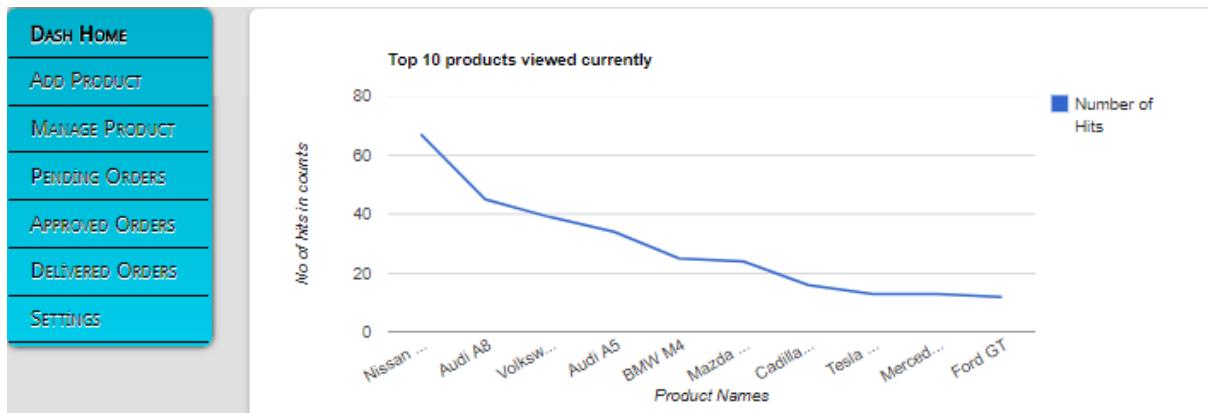
Name Of Product	Price	Quantity
Nissan Juke	£ 600.0	x1 £ 600.0
<b>Total Price of your Cart</b>		£600.0

At the bottom, there are two buttons: "Continue Shopping" and "Buy These Items".

❖ Yöneticinin görüntülediği grafik-1



## ❖ Yöneticinin görüntülediği grafik-2



## ❖ Yönetici ayarları

### Administrator Settings

**Add an Administrator**

Select Email:

Password:

Password Again:

**Add As Administrator**

**Remove Company**

**Delete Marked Company**

Mercedez-Benz
Volkswagen
BMW
Activision
Nissan
Square Enix
Saturn
Dreggonge
Cadillac
Ford
Tofas
Kartal
Plymouth
Mazda
Transend
Lincoln
Audi
Tesla
Pontiac
Ferrari
selam
adana

**Remove Category**

**Delete Marked Category**

Tüplü
Benzinli
Arazi Aracı
Dizel
4x4

## ❖ Stoğa araç ekleme (Add Product)

Add Car Step 1. Enter Car Company Step 2. Enter Category Step 3. Enter in Car Name, Tags Step 4. Enter in Pricing & Quantity and Description

DASH HOME ADD PRODUCT MANAGE PRODUCT PENDING ORDERS APPROVED ORDERS DELIVERED ORDERS SETTINGS

Step 1. Start selecting Car Company Or, Enter a one...

Select Company Select a Company ▾ Or Add A Company Company Name here Add Company Next >

## ❖ Araçları listeleme ve düzenleme (Manage Product)

En üstteki aracın stok adedi az olduğu için kırmızı uyarı çubuğu gözükmektedir

Manage Product Information

DASH HOME ADD PRODUCT MANAGE PRODUCT PENDING ORDERS APPROVED ORDERS DELIVERED ORDERS SETTINGS

Choose a Product to Edit Sort By Quantity Low ↑

Image	Product Name	Category	Quantity	Action
	Volkswagen GOLF By Volkswagen 4x4 £ 6000.00	Racing	2	Edit Product
	Cadillac xt5 By Cadillac Dizel £ 100.00	Cadillac	16	Edit Product
	Audi A8 By Audi 4x4 £ 700.00	Audi	17	Edit Product
	Audi TT By Audi 4x4 £ 50.00	Audi	17	Edit Product
	Mazda Miata By Mazda 4x4 £ 150.00	Racing	17	Edit Product
	Mercedez-Benz A180 By Mercedez-Benz 4x4 £ 300.00	A180	17	Edit Product
	BMW M4 By BMW Dizel £ 50.00	BMW	19	Edit Product
	Lincoln MKX By Lincoln Arazi Aracı £ 250.00	Land-Rover	19	Edit Product
	Tesla Model 3 By Tesla Tesla £ 460.00	Tesla	19	Edit Product
	Nissan Juke By Nissan 4x4 £ 600.00	Racing	20	Edit Product

## ❖ Bekleyen siparişleri görüntüleme (Pending orders)

Pending Orders					
I Approve these checks ▼		Change Product Status			
Order No	Item	Order Summary Price x Quantity	Date	Mobile No	Address
110535	Audi A8	£ 700.0 x1	2018-05-11	5320522830	Tunahan Yetimoğlu bütün oyuncak mah. alsancak bulvari dicle apt. 65F/2 asdsad sadasd
110536	Nissan Juke	£ 600.0 x1	2018-05-11	213125414	bütün oyuncak mah. alsancak bulvari dicle apt. 65F/2
Mercedez-Benz A180 Cadillac xt5 Audi TT		£ 300.0 x1 £ 100.0 x1 £ 50.0 x1	2018-05-11 2018-05-11 2018-05-11	213432452	safgfdg asfgdfgsad
110537	Mercedez-Benz A180	£ 300.0 x1	2018-05-11		

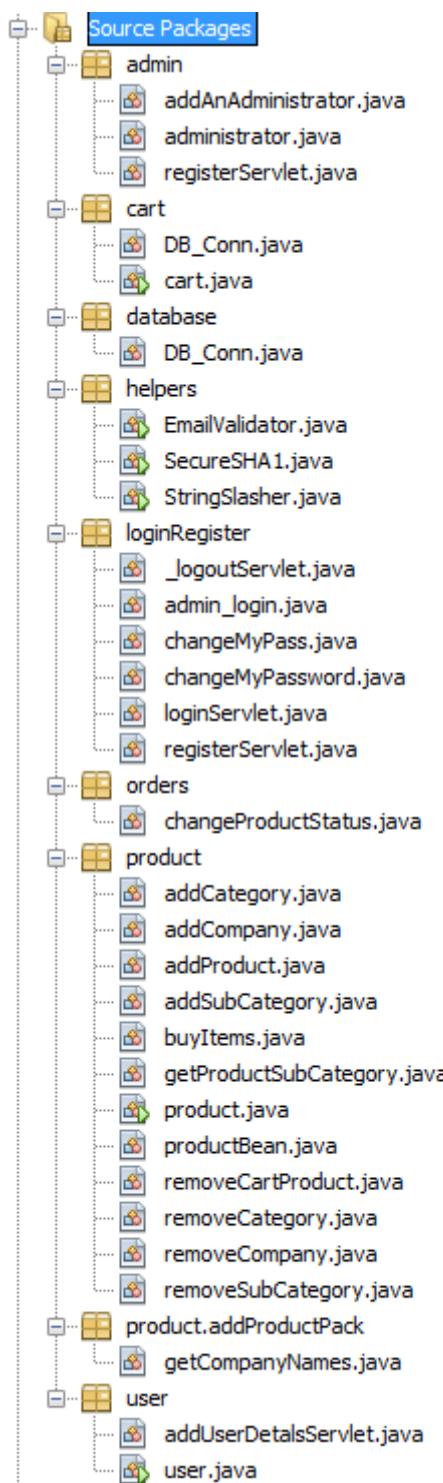
## ❖ Onaylanan siparişler (Approved Orders)

Approved Orders					
Delivered ▼		Change Product Status			
Order No	Item	Order Summary Price x Quantity	Date	Mobile No	Address
070533	Audi A8	£ 700.0 x1	2018-05-07	213213	tasida aslkjdsad
070534	Cadillac xt5 Audi A5	£ 100.0 x1 £ 4500.0 x1	2018-05-07	asdsad	sadas sadsad

## ❖ Teslim edilen siparişler (delivered orders)

Delivered Orders					
Delivered ▼		Change Product Status			
Order No	Item	Order Summary Price x Quantity	Date	Mobile No	Address
070530	Audi A8	£ 700.0 x1	2018-05-07	5320522830	Tunahan Yetimoğlu bütün oyuncak mah. alsancak bulvari dicle apt. 65F/2
070531	Nissan Juke	£ 600.0 x1	2018-05-07	5320522830	Tunahan Yetimoğlu bütün oyuncak mah. alsancak bulvari dicle apt. 65F/2
070532	Nissan Juke	£ 600.0 x1	2018-05-07	5320522830	Tunahan Yetimoğlu bütün oyuncak mah. alsancak bulvari dicle apt. 65F/2

## DAO SINIFLARI



❖ addAnAdministrator.java

The screenshot shows a Java code editor window with the file 'addAnAdministrator.java' open. The code is a servlet for adding an administrator. It includes imports for various Java packages, a class definition with annotations, and methods for processing requests and posts.

```
1 package admin;
2
3 import database.DB_Conn;
4 import helpers.EmailValidator;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.SQLIntegrityConstraintViolationException;
10 import javax.servlet.RequestDispatcher;
11 import javax.servlet.ServletException;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import javax.servlet.http.HttpSession;
16 import user.user;
17
18 /**
19 * @author Tunahan
20 */
21 public class addAnAdministrator extends HttpServlet {
22
23     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
24             throws ServletException, IOException {
25         response.setContentType("text/html;charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28
29             out.println("<html>");
30             out.println("<head>");
31             out.println("<title>Servlet addAnAdministrator</title>");
32             out.println("</head>");
33             out.println("<body>");
34             out.println("<h1>Servlet addAnAdministrator at " + request.getContextPath() + "</h1>");
35             out.println("</body>");
36             out.println("</html>");
37         } finally {
38             out.close();
39         }
40     }
41
42     @Override
43
44     protected void doPost(HttpServletRequest request, HttpServletResponse response)
45             throws ServletException, IOException {
46         //processRequest(request, response);
47     }
}
```

```
46 |     throws ServletException, IOException {
47 | //processRequest(request, response);
48 |
49 |     String message, messageDetail;
50 |     message = null;
51 |     messageDetail = null;
52 |     String email, pass, passAgain;
53 |     boolean isRegistered = false;
54 |
55 |     String messageUrl = "/message.jsp";
56 |     RequestDispatcher dispatcherMessage =
57 |         request.getServletContext().getRequestDispatcher(messageUrl);
58 |
59 |     email = request.getParameter("email");
60 |     pass = request.getParameter("pass");
61 |     passAgain = request.getParameter("passAgain");
62 |     EmailValidator validator = new EmailValidator();
63 |     boolean isEmailValid = validator.validate(email);
64 |
65 |     PrintWriter out = response.getWriter();
66 |
67 |     HttpSession userSession = request.getSession();
68 |
69 |     try {
70 |         DB_Conn con = new DB_Conn();
71 |         Connection c = con.getConnection();
72 |         if ((request.getParameter("email") != null) || (request.getParameter("email") != null)){
73 |             if (isEmailValid) {
74 |                 if (pass.length() > 7) {
75 |                     if (pass.equals(passAgain)) {
76 |                         // then the user is registered and a session is
77 |
78 |                         String sql = "INSERT INTO `database_ecarla`.`administrators` "
79 |                             + "(`admin_id`, `email`, `password` ) "
80 |                             + "VALUES (NULL , ?, SHA1( ? )); ";
81 |
82 |                         PreparedStatement psmt = c.prepareStatement(sql);
83 |
84 |                         psmt.setString(1, email);
85 |
86 |                         psmt.setString(2, pass);
87 |
88 |                         int i = psmt.executeUpdate();
89 |
90 |                         if (i == 1) {
91 |                             isRegistered = true;
92 |                             out.println("You are registered ");
93 |                             user User = new user();
94 |                             User.setUserEmail(email);
```

```

94         User.setUserEmail(email);
95         userSession.setAttribute("user", User);
96         response.sendRedirect("/eCarla/admin_settings.jsp");
97     } else {
98         isRegistered = false;
99         out.println("You are not registered");
100    }
101
102    } //Else both passwords do not match
103    else {
104        isRegistered = false;
105        message = "Passwords do not match";
106        messageDetail = "Please provide a matching passwords";
107        out.print(" nOT Success! both passwords do not match!");
108
109    }
110
111    } //or the paasword length is less than 7
112    else {
113        isRegistered = false;
114        message = "Password length is less than 7 characters";
115        messageDetail = "Please provide a passwords that has password length of minimum of seven alphanumeric characters";
116        out.print(" nOT Success!! the paasword length is less than 7");
117    }
118
119    } //or email is wrong
120    else {
121        isRegistered = false;
122        message = "No email address typed";
123        messageDetail = "Please provide a valid email address";
124        out.print(" nOT Success!! email is wrong");
125    }
126
127    else {
128        isRegistered = false;
129        message = "Please enter values";
130        messageDetail = "Please provide an email address. Your account currently is not registered";
131    }
132
133    if (isRegistered == false) {
134        request.setAttribute("message", message);
135        request.setAttribute("messageDetail", messageDetail);
136        dispatchMessage.forward(request, response);
137    }
138
139    } catch (SQLIntegrityConstraintViolationException ex) {
140        // user exists but wrong passwodt ask to CHANGE THE PASSWORD
141        messageDetail = ex.getMessage();
142        message = "You have been registered earlier please try your right password again, else change your password...";
143        out.print(" nOT Success!!" + ex);
144        request.setAttribute("message", message);
145
146    } catch (Exception ex) {
147        messageDetail = ex.getMessage();
148        message = "There was a problem in registering your account please do retry again later...";
149        out.print(" nOT Success!!" + ex);
150        request.setAttribute("message", message);
151        request.setAttribute("messageDetail", messageDetail);
152        dispatchMessage.forward(request, response);
153        response.sendError(404);
154    }
155
156
157    @Override
158    protected void doGet(HttpServletRequest request, HttpServletResponse response)
159        throws ServletException, IOException {
160        //processRequest(request, response);
161        doPost(request, response);
162    }
163
164    @Override
165    public String getServletInfo() {
166        return "Short description";
167    }
168}
169

```

❖ administrator.java

The screenshot shows a Java code editor window with the title bar "administrator.java". The menu bar includes "Source", "History", and various tool icons. The code itself is as follows:

```
1 package admin;
2
3 import database.DB_Conn;
4 import java.sql.Connection;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.ArrayList;
9
10 /**
11 * @author Tunahan
12 */
13 public class administrator {
14     Connection c;
15     String admin_email = null;
16
17     public ArrayList<String> getListsOfAdmin() throws SQLException, ClassNotFoundException {
18         c = new DB_Conn().getConn();
19         String getAdministratorsEmail = " SELECT * "
20             + " FROM `user` ";
21
22         Statement st = c.createStatement();
23         ResultSet executeQuery = st.executeQuery(getAdministratorsEmail);
24         listsOfAdmin.clear();
25         while (executeQuery.next()) {
26             listsOfAdmin.add(executeQuery.getString("email"));
27         }
28         return listsOfAdmin;
29     }
30
31     ArrayList <String> listsOfAdmin = new ArrayList<String>();
32     public String getAdmin_email() {
33         return admin_email;
34     }
35
36     public void setAdmin_email(String admin_email) {
37         this.admin_email = admin_email;
38     }
39
40 }
41 }
```

## ❖ registerServlet.java

The screenshot shows a Java code editor window with the file "registerServlet.java" open. The code implements a servlet for registering users. It includes imports for various Java packages, a constructor, and methods for processing requests via GET and POST. The code uses PrintWriter to generate HTML output.

```
1 package admin;
2
3 import loginRegister.*;
4 import database.DB_Conn;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import helpers.*;
12 import java.sql.*;
13 import javax.servlet.RequestDispatcher;
14 import javax.servlet.http.HttpSession;
15 import user.user;
16
17 public class registerServlet extends HttpServlet {
18
19
20     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
21             throws ServletException, IOException {
22         response.setContentType("text/html;charset=UTF-8");
23         PrintWriter out = response.getWriter();
24         try {
25             out.println("<html>");
26             out.println("<head>");
27             out.println("<title>Servlet registerServlet</title>");
28             out.println("</head>");
29             out.println("<body>");
30             out.println("<h1>Servlet registerServlet at " + request.getContextPath() + "</h1>");
31             out.println("</body>");
32             out.println("</html>");
33         } finally {
34             out.close();
35         }
36     }
37
38     @Override
39     protected void doGet(HttpServletRequest request, HttpServletResponse response)
40             throws ServletException, IOException {
41         processRequest(request, response);
42     }
43
44     @Override
45     protected void doPost(HttpServletRequest request, HttpServletResponse response)
46             throws ServletException, IOException {
47             // ...
48     }
49 }
```

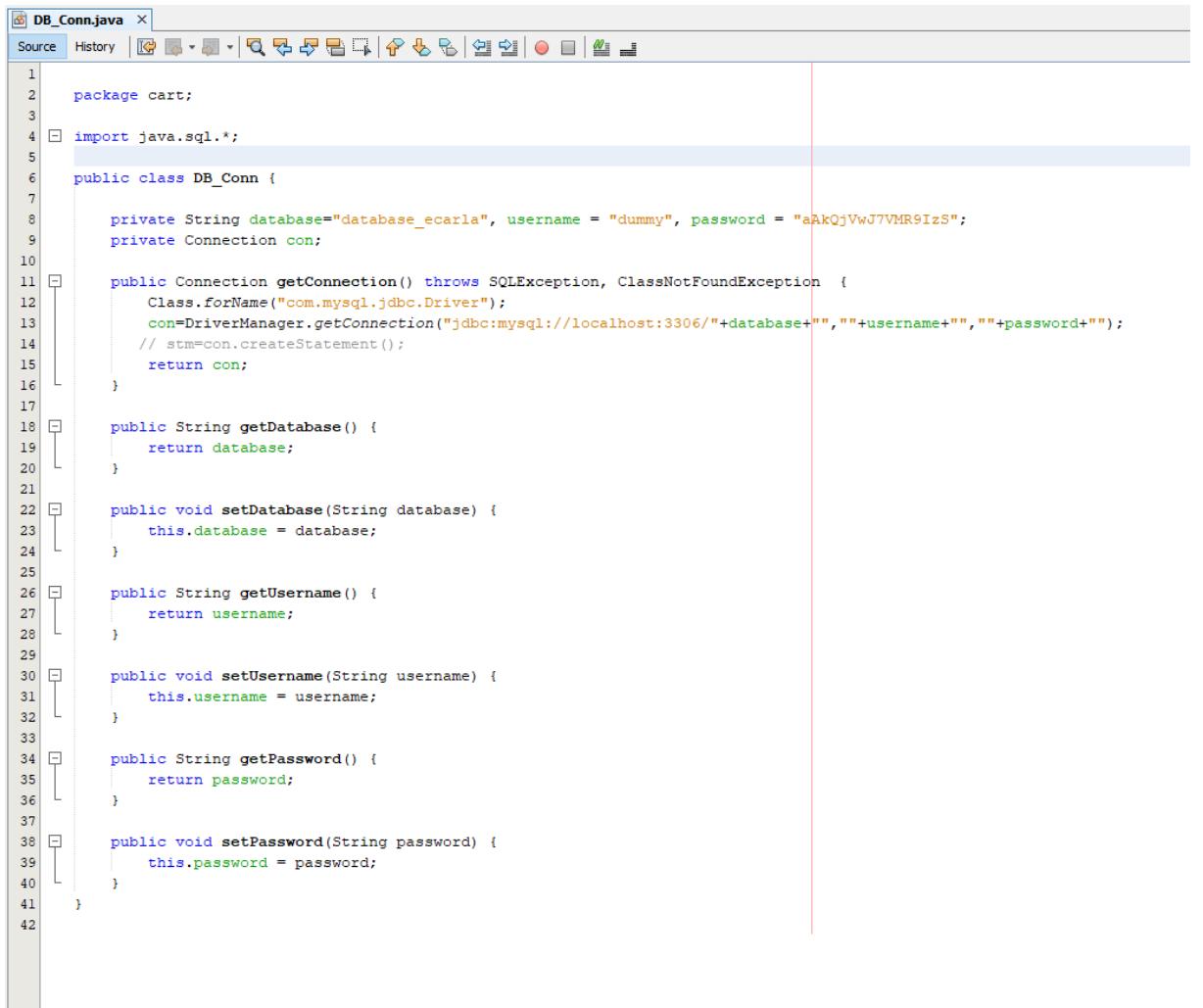
```
49 //processRequest(request, response);
50
51     String message, messageDetail;
52     message = null;
53     messageDetail = null;
54     String email, pass, passAgain;
55     boolean isRegistered = false;
56
57     String messageUrl = "/message.jsp";
58     RequestDispatcher dispatchMessage =
59         request.getServletContext().getRequestDispatcher(messageUrl);
60
61     email = request.getParameter("email");
62     pass = request.getParameter("pass");
63     passAgain = request.getParameter("passAgain");
64     EmailValidator validator = new EmailValidator();
65     boolean isEmailValid = validator.validate(email);
66
67     String ipAdd = request.getRemoteAddr();
68
69     PrintWriter out = response.getWriter();
70
71     HttpSession userSession = request.getSession();
72
73     try {
74         DB_Conn con = new DB_Conn();
75         Connection c = con.getConnection();
76         if ((request.getParameter("email") != null) || (request.getParameter("email") != null)){
77             if (isEmailValid) {
78                 if (pass.length() > 7) {
79                     if (pass.equals(passAgain)) {
80                         // then the user is registered and a session is
81
82                         String sql = "INSERT INTO `database_ecarla`.`user` "
83                             + `(`user_id` ,`email` ,`pass` ,`registeredOn` ) "
84                             + "VALUES (NULL , ? , SHA1( ? ) , NOW( )); ";
85
86                         PreparedStatement psmt = c.prepareStatement(sql);
87
88                         psmt.setString(1, email);
89
90                         psmt.setString(2, pass);
91
92                         int i = psmt.executeUpdate();
93
94                         if (i == 1) {
95                             isRegistered = true;
96                             out.println("You are registered ");
97                             user User = new user();
```

```

97         user User = new user();
98         User.setUserEmail(email);
99         userSession.setAttribute("user", User);
100        response.sendRedirect("/eCarla/index.jsp");
101    } else {
102        isRegistered = false;
103        out.println("You are not registered");
104    }
105
106    } //Else both passwords do not match
107    else {
108        isRegistered = false;
109        message = "Passwords do not match";
110        messageDetail = "Please provide a matching passwords";
111        out.print(" nOT Success! both passwords do not match!");
112
113    }
114    } //or the paasword length is less than 7
115    else {
116        isRegistered = false;
117        message = "Password length is less than 7 characters";
118        messageDetail = "Please provide a passwords that has password length of minimum of seven alphanumeric";
119        out.print(" nOT Success!! the paasword length is less than 7");
120    }
121    } //or email is wrong
122    else {
123        isRegistered = false;
124        message = "No email address typed";
125        messageDetail = "Please provide a valid email address";
126        out.print(" nOT Success!! email is wrong");
127    }
128
129    else {
130        isRegistered = false;
131        message = "Please enter values";
132        messageDetail = "Please provide an email address. Your account currently is not registered";
133    }
134
135    if (isRegistered == false) {
136        request.setAttribute("message", message);
137        request.setAttribute("messageDetail", messageDetail);
138        dispatchMessage.forward(request, response);
139    }
140
141    //try ends here
142    } catch (SQLIntegrityConstraintViolationException ex) {
143        // user exists but wrong passwotd ask to CHANGE THE PASSWORD
144        messageDetail = ex.getMessage();
145        message = "You have been registered earlier please try your right password again, else change your password...<a
```

```
134
135     if (isRegistered == false) {
136         request.setAttribute("message", message);
137         request.setAttribute("messageDetail", messageDetail);
138         dispatchMessage.forward(request, response);
139     }
140
141     //try ends here
142 } catch (SQLIntegrityConstraintViolationException ex) {
143     // user exists but wrong password ask to CHANGE THE PASSWORD
144     messageDetail = ex.getMessage();
145     message = "You have been registered earlier please try your right password again, else";
146     out.print(" NOT Success!!" + ex);
147     request.setAttribute("message", message);
148     request.setAttribute("messageDetail", messageDetail);
149     dispatchMessage.forward(request, response);
150 } catch (Exception ex) {
151     messageDetail = ex.getMessage();
152     message = "There was a problem in registering your account please do retry again later.";
153     out.print(" NOT Success!!" + ex);
154     request.setAttribute("message", message);
155     request.setAttribute("messageDetail", messageDetail);
156     dispatchMessage.forward(request, response);
157 }
158
159
160 @Override
161 public String getServletInfo() {
162     return "Short description";
163 }
164
165 }
```

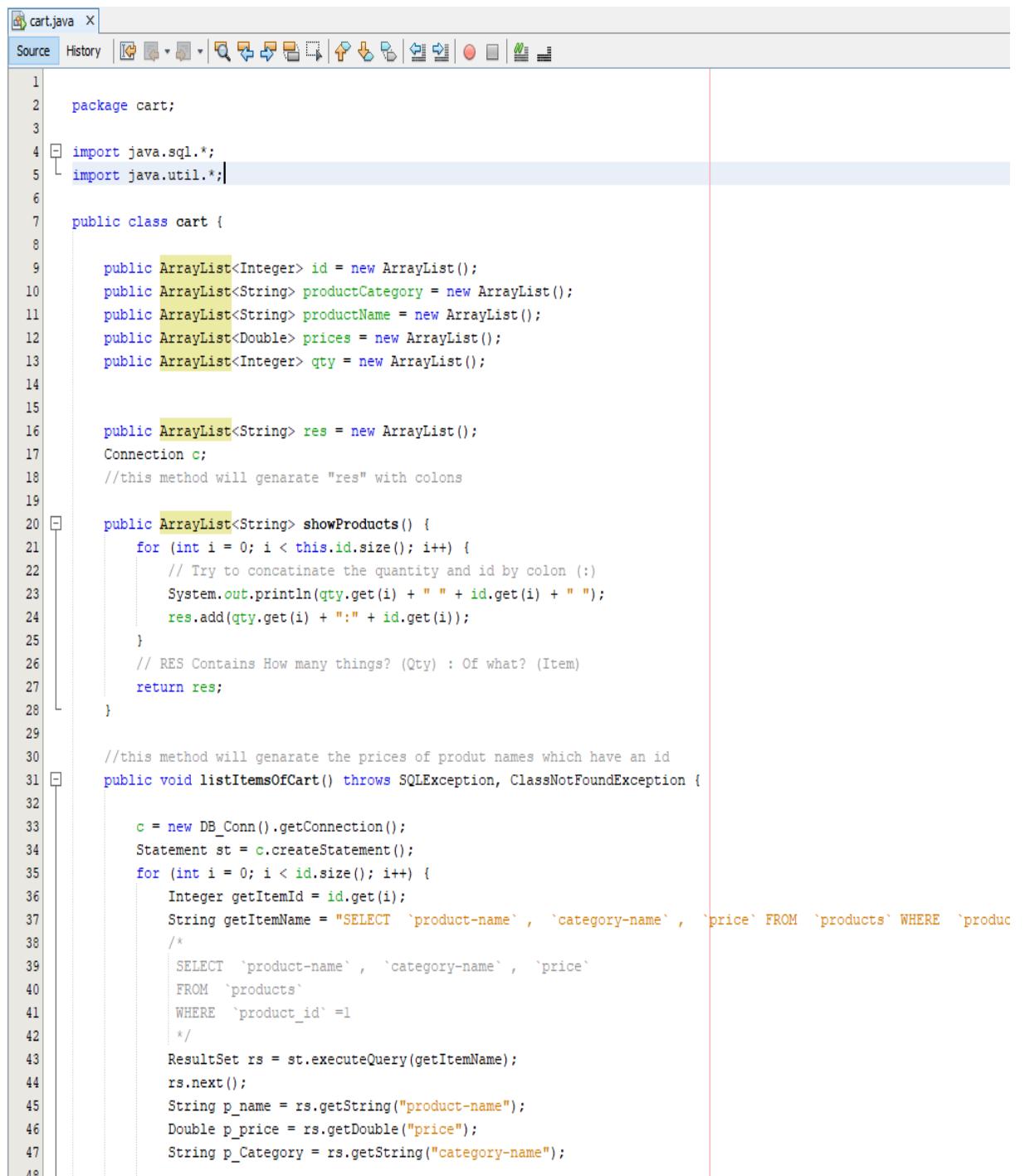
## ❖ DB\_Conn.java



The screenshot shows a Java code editor window titled "DB\_Conn.java". The code is a class named DB\_Conn with methods for connecting to a MySQL database and getting or setting database, user, and password information. The code uses JDBC to establish a connection.

```
1 package cart;
2
3 import java.sql.*;
4
5 public class DB_Conn {
6
7     private String database="database_ecarla", username = "dummy", password = "aAkQjVwJ7vMR9IzS";
8     private Connection con;
9
10    public Connection getConnection() throws SQLException, ClassNotFoundException {
11        Class.forName("com.mysql.jdbc.Driver");
12        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/"+database+"','"+username+"','"+password+"");
13        // stm=con.createStatement();
14        return con;
15    }
16
17    public String getDatabase() {
18        return database;
19    }
20
21    public void setDatabase(String database) {
22        this.database = database;
23    }
24
25    public String getUsername() {
26        return username;
27    }
28
29    public void setUsername(String username) {
30        this.username = username;
31    }
32
33    public String getPassword() {
34        return password;
35    }
36
37    public void setPassword(String password) {
38        this.password = password;
39    }
40
41 }
42
```

## ❖ cart.java



The screenshot shows a Java code editor window titled "cart.java". The code is a class named "cart" that interacts with a database to manage a shopping cart. It uses ArrayLists to store product IDs, categories, names, prices, and quantities. The "showProducts" method concatenates quantity and ID with colons. The "listItemsOfCart" method performs a SQL query to get product names, categories, and prices based on the stored IDs.

```
1 package cart;
2
3 import java.sql.*;
4 import java.util.*;
5
6 public class cart {
7
8     public ArrayList<Integer> id = new ArrayList();
9     public ArrayList<String> productCategory = new ArrayList();
10    public ArrayList<String> productName = new ArrayList();
11    public ArrayList<Double> prices = new ArrayList();
12    public ArrayList<Integer> qty = new ArrayList();
13
14
15    public ArrayList<String> res = new ArrayList();
16    Connection c;
17    //this method will generate "res" with colons
18
19    public ArrayList<String> showProducts() {
20        for (int i = 0; i < this.id.size(); i++) {
21            // Try to concatenate the quantity and id by colon (:)
22            System.out.println(qty.get(i) + " " + id.get(i) + " ");
23            res.add(qty.get(i) + ":" + id.get(i));
24        }
25        // RES Contains How many things? (Qty) : Of what? (Item)
26        return res;
27    }
28
29
30    //this method will generate the prices of product names which have an id
31    public void listItemsOfCart() throws SQLException, ClassNotFoundException {
32
33        c = new DB_Conn().getConn();
34        Statement st = c.createStatement();
35        for (int i = 0; i < id.size(); i++) {
36            Integer getItemId = id.get(i);
37            String getItemName = "SELECT `product-name` , `category-name` , `price` FROM `products` WHERE `product-id` = " + getItemId;
38            /*
39             * SELECT `product-name` , `category-name` , `price`
40             * FROM `products`
41             * WHERE `product_id` = l
42             */
43            ResultSet rs = st.executeQuery(getItemName);
44            rs.next();
45            String p_name = rs.getString("product-name");
46            Double p_price = rs.getDouble("price");
47            String p_Category = rs.getString("category-name");
48        }
49    }
50}
```

```
49     }
50     }
51     }
52   }
53 }
54
55 public ArrayList<String> getProductCategorys() {
56   return productCategory;
57 }
58
59 //getters for product names, prices
60 public ArrayList<String> getProductName() throws SQLException, ClassNotFoundException {
61   /*for (int i=0;i<productName.size();i++){
62     System.out.println(""+productName.get(i));
63   }*/
64   return productName;
65 }
66 //this will return a list
67
68 public ArrayList<Double> getPrices() throws SQLException, ClassNotFoundException {
69   /*for (int i=0;i<prices.size();i++){
70     System.out.println(""+prices.get(i));
71   }*/
72   listItemsOfCart();
73   return prices;
74 }
75
76 public String getProductName(int id) throws SQLException, ClassNotFoundException {
77   //String p_name = productName.get(id);
78   //return p_name;
79   c = new DB_Conn().getConnection();
80   String getProductName = "SELECT `product-name` FROM `products` WHERE `product_id` =" + id + "";
81   Statement st = c.createStatement();
82
83   ResultSet rs = st.executeQuery(getProductName);
84   rs.next();
85   String name = rs.getString("product-name");
86   return name;
87 }
88
89 public String getCategoryName(int id) throws SQLException, ClassNotFoundException {
90   //String p_name = productName.get(id);
91   //return p_name;
92   c = new DB_Conn().getConnection();
93   String getCategoryName = "SELECT `category-name` FROM `products` WHERE `product_id` =" + id + "";
94   Statement st = c.createStatement();
95   /*SELECT `category-name`
96   FROM `products`
```

```
97     | WHERE `product_id` =1*/
98     | ResultSet rs = st.executeQuery(getCategoryName);
99     | rs.next();
100    | String name = rs.getString("category-name");
101    | return name;
102  }
103
104  public double getProductPrice(int id) throws SQLException, ClassNotFoundException {
105      c = new DB_Conn().getConnnection();
106      String getProductName = "SELECT `price` FROM `products` WHERE `product_id` =" + id + "";
107      Statement st = c.createStatement();
108      ResultSet rs = st.executeQuery(getProductName);
109      rs.next();
110      double price = rs.getDouble("price");
111      return price;
112  }
113
114  public double getTotalPriceOfCart() throws SQLException, ClassNotFoundException {
115      ArrayList<Double> t_prices = new ArrayList(); //this will hold the prices
116
117      c = new DB_Conn().getConnnection();
118      Statement st = c.createStatement();
119      for (int i = 0; i < id.size(); i++) {
120          String sqlGetPrice = "SELECT `price` FROM `products` WHERE `product_id` =" + id.get(i).toString() + ";";
121          ResultSet rs = st.executeQuery(sqlGetPrice);
122          rs.next();
123          //theres only one value in resultset so go one step further and get the value
124          double f = rs.getFloat("price");
125          // Multiply by the quantity
126          f *= qty.get(i);
127          // Add that multiplied value to prices list
128          t_prices.add(f);
129      }
130
131      double sum = 0;
132      for (int i = 0; i < t_prices.size(); i++) {
133          double p = t_prices.get(i);
134          sum += p;
135      }
136      //System.out.println("Sum of your cart is "+ sum);
137      return sum;
138  }
139
140  public ArrayList<Integer> getQty() {
141      return qty;
142  }
143
144  public ArrayList<Integer> getId() {
```

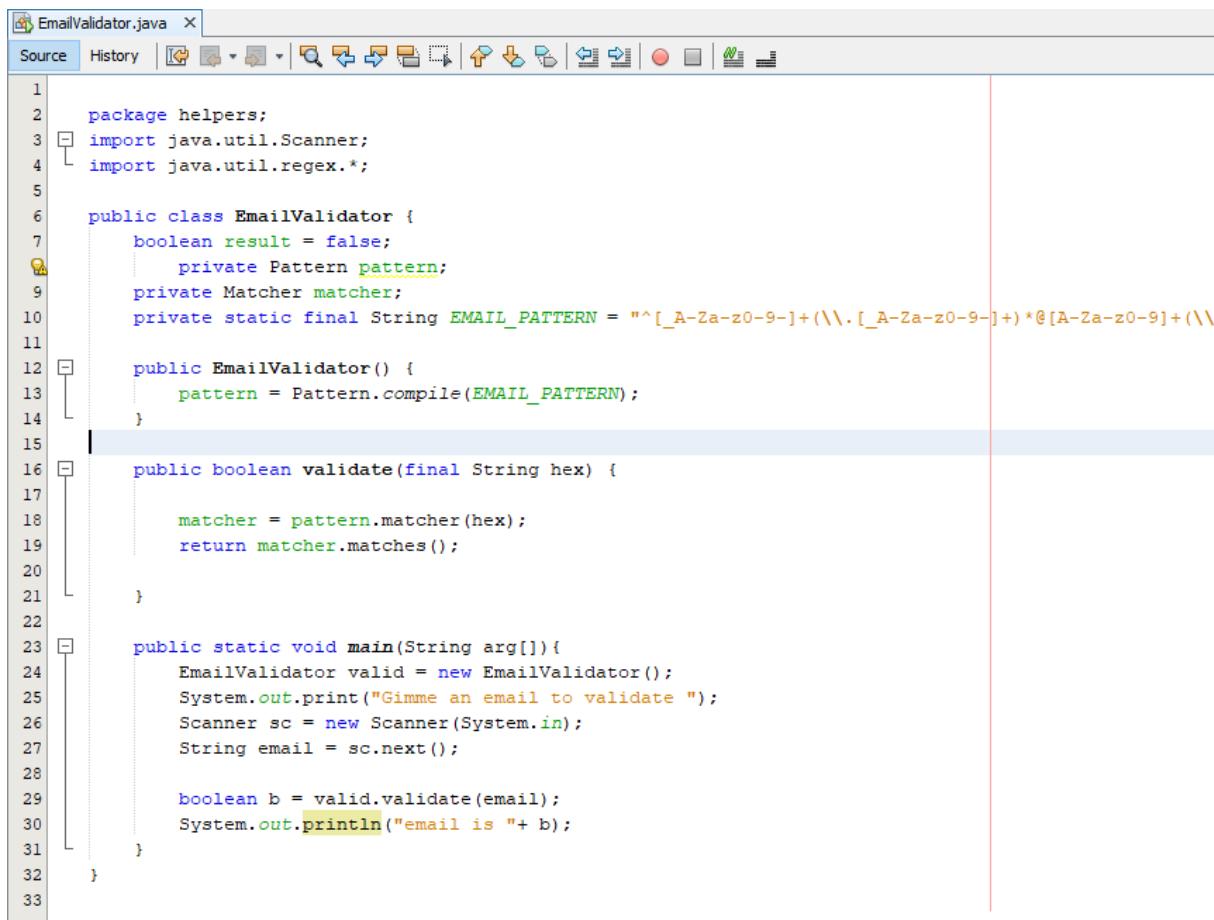
```
145     }
146 }
147
148 public boolean addProduct(int id) throws SQLException, ClassNotFoundException {
149     boolean added, matches;
150
151     c = new DB_Conn().getConnection();
152     Statement st = c.createStatement();
153
154     String sqlGetValidIds = "SELECT `product_id` FROM `products`;";
155     ResultSet rs = st.executeQuery(sqlGetValidIds);
156
157     while (rs.next()) {
158         int id_db = rs.getInt("product_id");
159         if (id_db == id) {
160             // id matches in the database
161             matches = true;
162             break;
163         }
164     }
165
166     if (matches = true) {
167         //inserti nto arraylist
168         if (this.id.contains(id)) {
169             //increment the quantity at where the product id resides
170             int index = this.id.indexOf(id);
171             this.qty.set(index, this.qty.get(index) + 1);
172         } else {
173             //Add one qty
174             this.qty.add(1);
175             //Add productID
176             this.id.add(id);
177         }
178         added = true;
179     } else {
180         //get out the function
181         //Because no product in in the db with that id
182         System.out.println("No product with that id exist in database");
183         added = false;
184     }
185     return added;
186 }
187
188 public boolean removeProduct(int productId) {
189     boolean removed;
190
191     if (this.id.contains(productId)) {
192         //where is the id's position
```

```

193 //this.id.indexOf(productId);
194 int index = this.id.indexOf(productId);
195 int quantity = this.qty.get(index);
196 //remove the entire stack from all the ArrayLists
197 this.productCategory.remove(index);
198 this.productName.remove(index);
199 this.prices.remove(index);
200 this.qty.remove(index);
201 this.id.remove(index);
202 removed = true;
203 }else {
204     removed = false;
205 }
206
207
208
209 return removed;
210 }
211
212 public static void main(String args[]) throws SQLException, ClassNotFoundException {
213     try {
214         cart c = new cart();
215         Scanner sc = new Scanner(System.in);
216
217         String choice = "1";
218         while (choice.equals("1")) {
219
220             System.out.println("What do u want to buy? Give me ID");
221
222             int id = sc.nextInt();
223
224             c.addProduct(id); //Good
225
226             System.out.println("Do u wish to continue?");
227
228             choice = sc.next();
229         }
230
231         c.showProducts(); //Good
232
233         //c.listItemsOfCart();
234         ArrayList<String> productNames = c.getProductNames();
235         ArrayList<String> p_Category = c.getProductCategorys();
236         ArrayList<Double> pricesl = c.getPrices();
237         ArrayList<Integer> qtyl = c.getQty();
238         for (int i = 0; i < productNames.size(); i++) {
239             System.out.println(p_Category.get(i) + " " + productNames.get(i) + " " + pricesl.get(i));
240         }
241         System.out.println("Total Price of the cart is :" + c.getTotalPriceOfCart());
242
243         choice = "1";
244         while ("1".equals(choice)) {
245
246             System.out.println("Give Product id to remove");
247             int id = sc.nextInt();
248             boolean removeProduct = c.removeProduct(id);
249             if (removeProduct) {
250                 for (int i = 0; i < productNames.size(); i++) {
251                     System.out.println(p_Category.get(i) + " " + productNames.get(i) + " " + pricesl.get(i));
252
253                     System.out.println("Total Price of the cart is :" + c.getTotalPriceOfCart());
254
255                 }
256             }
257             else {
258                 System.out.println("No Product to remove!!!");
259             }
260         }
261
262
263
264         //c.getTotalPriceOfCart();
265     } catch (SQLException e) {
266         System.out.println(" " + e);
267     } catch (ClassNotFoundException e) {
268         System.out.println(" " + e);
269     }
270 }
271
272 }
273

```

❖ EmailValidator.java



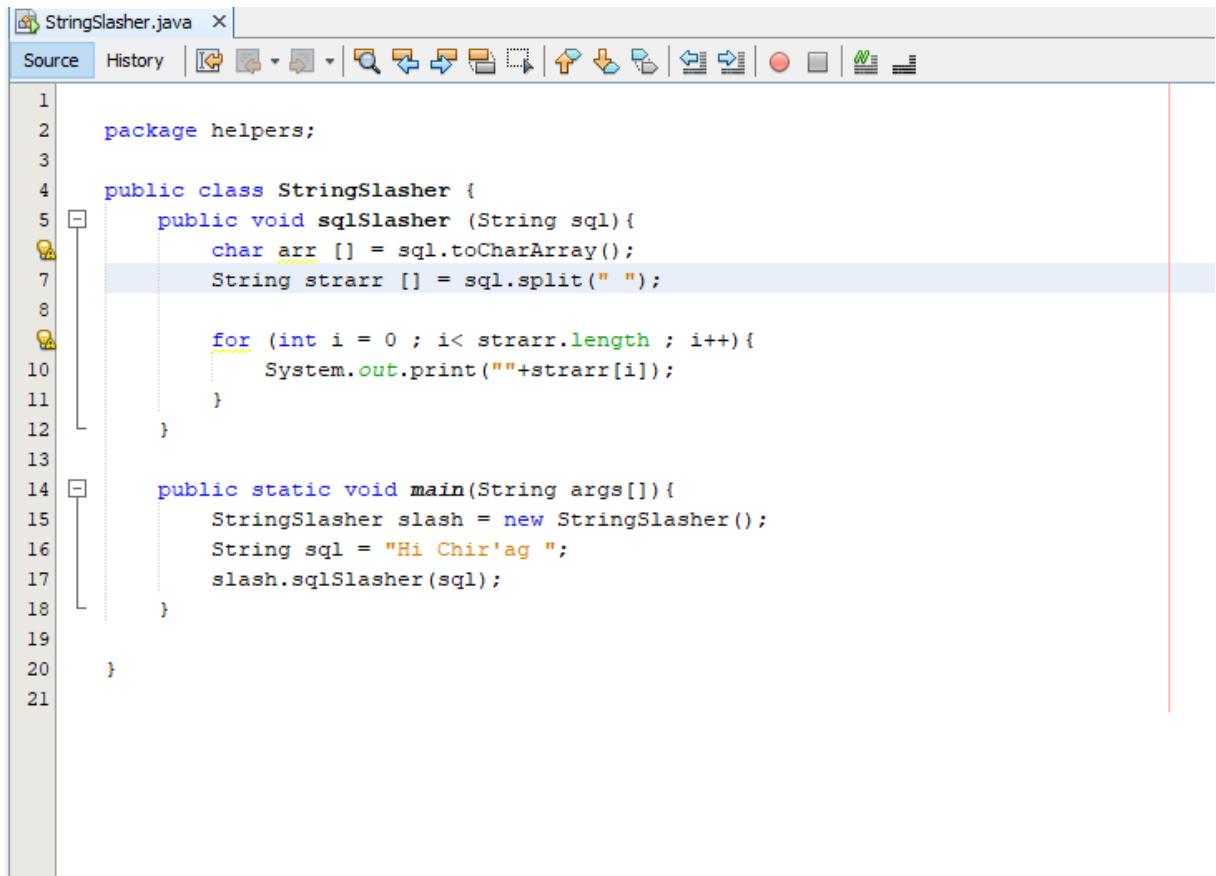
```
1 package helpers;
2 import java.util.Scanner;
3 import java.util.regex.*;
4
5 public class EmailValidator {
6     boolean result = false;
7     private Pattern pattern;
8     private Matcher matcher;
9     private static final String EMAIL_PATTERN = "^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9]+(\\."
10
11     public EmailValidator() {
12         pattern = Pattern.compile(EMAIL_PATTERN);
13     }
14
15     public boolean validate(final String hex) {
16
17         matcher = pattern.matcher(hex);
18         return matcher.matches();
19     }
20
21 }
22
23 public static void main(String arg[]){
24     EmailValidator valid = new EmailValidator();
25     System.out.print("Gimme an email to validate ");
26     Scanner sc = new Scanner(System.in);
27     String email = sc.next();
28
29     boolean b = valid.validate(email);
30     System.out.println("email is "+ b);
31 }
32
33 }
```

## ❖ SecureSHA1.java

The screenshot shows a Java code editor window with the file "SecureSHA1.java" open. The code implements a SHA-1 hash function. It includes imports for java.io, java.security, and java.util.logging. The class SecureSHA1 contains methods for converting byte arrays to hex strings and for generating SHA-1 hashes of text. A main method demonstrates the usage of the class.

```
1 package helpers;
2
3 import java.io.UnsupportedEncodingException;
4 import java.security.MessageDigest;
5 import java.security.NoSuchAlgorithmException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8
9
10 public class SecureSHA1 {
11     private static String convertToHex(byte[] data) {
12         StringBuffer buf = new StringBuffer();
13         for (int i = 0; i < data.length; i++) {
14             int halfbyte = (data[i] >>> 4) & 0x0F;
15             int two_halfs = 0;
16             do {
17                 if ((0 <= halfbyte) && (halfbyte <= 9))
18                     buf.append((char) ('0' + halfbyte));
19                 else
20                     buf.append((char) ('a' + (halfbyte - 10)));
21                 halfbyte = data[i] & 0x0F;
22             } while(two_halfs++ < 1);
23         }
24         return buf.toString();
25     }
26
27
28     public static String getSHA1(String text)
29             throws NoSuchAlgorithmException, UnsupportedEncodingException {
30         MessageDigest md;
31         md = MessageDigest.getInstance("SHA-1");
32         byte[] shalhash = new byte[40];
33         md.update(text.getBytes("iso-8859-1"), 0, text.length());
34         shalhash = md.digest();
35         return convertToHex(shalhash);
36     }
37
38     public static void main(String args[]){
39         try {
40             String s = "123";
41             String shalS = getSHA1(s);
42             System.out.print("SHA1 hash for "+ s + " is " + shalS);
43         } catch (Exception e) {
44             System.out.println(e);
45         }
46     }
47 }
48 }
```

❖ **StringSlasher.java**



```
1 package helpers;
2
3 public class StringSlasher {
4     public void sqlSlasher (String sql){
5         char arr [] = sql.toCharArray();
6         String strarr [] = sql.split(" ");
7
8         for (int i = 0 ; i< strarr.length ; i++){
9             System.out.print(""+strarr[i]);
10        }
11    }
12
13    public static void main(String args[]){
14        StringSlasher slash = new StringSlasher();
15        String sql = "Hi Chir'ag ";
16        slash.sqlSlasher(sql);
17    }
18
19}
20
21
```

## ❖ adminLogin.java

The screenshot shows a Java code editor window with the file "admin\_login.java" open. The code is a servlet named "admin\_login" that extends the "HttpServlet" class. It overrides the "doGet" and "doPost" methods to handle HTTP requests. The code includes imports for various Java packages, including "loginRegister", "admin.administrator", "database.DB\_Conn", "helpers.SecureSHA1", "java.io.IOException", "java.io.PrintWriter", "java.sql.Connection", "java.sql.PreparedStatement", "java.sql.ResultSet", "java.sql.SQLException", "javax.servlet.RequestDispatcher", "javax.servlet.ServletException", "javax.servlet.http.HttpServlet", "javax.servlet.http.HttpServletRequest", "javax.servlet.http.HttpServletResponse", "javax.servlet.http.HttpSession", and "user.user". The "processRequest" method is implemented to output an HTML page with a title and a message indicating the servlet's name and context path.

```
1 package loginRegister;
2
3 import admin.administrator;
4 import database.DB_Conn;
5 import helpers.SecureSHA1;
6 import java.io.IOException;
7 import java.io.PrintWriter;
8 import java.sql.Connection;
9 import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import javax.servlet.RequestDispatcher;
13 import javax.servlet.ServletException;
14 import javax.servlet.http.HttpServlet;
15 import javax.servlet.http.HttpServletRequest;
16 import javax.servlet.http.HttpServletResponse;
17 import javax.servlet.http.HttpSession;
18 import user.user;
19
20
21 public class admin_login extends HttpServlet {
22
23     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
24             throws ServletException, IOException {
25         response.setContentType("text/html;charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28             /* TODO output your page here. You may use following sample code. */
29             out.println("<html>");
30             out.println("<head>");
31             out.println("<title>Servlet admin_login</title>");
32             out.println("</head>");
33             out.println("<body>");
34             out.println("<h1>Servlet admin_login at " + request.getContextPath() + "</h1>");
35             out.println("</body>");
36             out.println("</html>");
37         } finally {
38             out.close();
39         }
40     }
41
42     @Override
43     protected void doGet(HttpServletRequest request, HttpServletResponse response)
44             throws ServletException, IOException {
45         processRequest(request, response);
46     }
47
48     @Override
49     protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
50     @
51     protected void doPost(HttpServletRequest request, HttpServletResponse response)
52             throws ServletException, IOException {
53         //processRequest(request, response); //processRequest(request, response);
54         String email, pass;
55         String db_email, db_pass;
56         boolean isLoggedIn = false;
57         HttpSession userSession = request.getSession();
58         PrintWriter out = response.getWriter();
59         email = request.getParameter("email");
60         pass = request.getParameter("pass");
61         String message, messageDetail;
62         message = "";
63         messageDetail = "";
64
65         String messageUrl = "/message.jsp";
66         RequestDispatcher dispatchMessage =
67             request.getServletContext().getRequestDispatcher(messageUrl);
68
69         try {
70             pass = SecureSHA1.getSHA1(pass);
71             //out.println("email " + email + " pass " + pass);
72             DB_Conn con = new DB_Conn();
73             Connection c = con.getConnection();
74             String sqlGetUsers = "SELECT * FROM `administrators` ;";
75
76             PreparedStatement st = c.prepareStatement(sqlGetUsers);
77
78             ResultSet rs = st.executeQuery();
79
80             while (rs.next()) {
81                 db_email = rs.getString("email");
82                 db_pass = rs.getString("password");
83
84                 if (email.equals(db_email)) {
85                     message = "Your email-id exists with us!";
86                     //you exist with us
87                     if (pass.equals(db_pass)) {
88                         isLoggedIn = true;
89                         //user exists and password is matching
90                         //out.print("You are logged in");
91                         User User = new user();
92                         administrator Administrator = new administrator();
93                         Administrator.setAdmin_email(db_email);
94                         User.setUserEmail(email);
95                         userSession.setAttribute("user", User);
96                         userSession.setAttribute("admin", Administrator);
97                         response.sendRedirect(request.getContextPath());
98                     }
99                 } else {
```

```
97     }
98
99
100    else {
101        isLoggedIn = false;
102        // user exists but wrong password ask to CHANGE THE PASSWORD
103        message = "Wrong Password...!";
104        messageDetail = "Password does not match with the password";
105        //out.println("wrong password Change the password now <a href='");
106        break;
107    }
108
109    else {
110        //or there no such email YOU do not exist with us Create an account
111        //out.println(" no such email Register an account now!");
112        message = "Not an Administrator";
113        messageDetail = "You are currently not an Administrator!";
114        isLoggedIn = false;
115    }
116
117    if (isLoggedIn == false){
118        request.setAttribute("message", message);
119        request.setAttribute("messageDetail", messageDetail);
120        dispatchMessage.forward(request, response);
121    }
122    catch (SQLException e) {
123        message = "Error in the Login process";
124        messageDetail = "There was an error in the process of login Please try again later";
125
126        request.setAttribute("message", message);
127        request.setAttribute("messageDetail", messageDetail);
128        //dispatchMessage.forward(request, response);
129    } catch (Exception e) {
130        message = "Error in the Login process";
131        messageDetail = "There was an error in the process of login Please try again later";
132
133        request.setAttribute("message", message);
134        request.setAttribute("messageDetail", messageDetail);
135        //dispatchMessage.forward(request, response);
136    }
137    @Override
138    public String getServletInfo() {
139        return "Short description";
140    } // </editor-fold>
141}
142
```

❖ loginServlet.java

The screenshot shows a Java code editor window with the file "loginServlet.java" open. The code is a servlet for handling user login. It includes imports for various Java packages, a main class definition, and overridden methods for doGet and doPost. The code uses PrintWriter to output HTML content.

```
1 package loginRegister;
2
3 import database.DB_Conn;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10 import helpers.*;
11 import java.sql.*;
12 import javax.servlet.RequestDispatcher;
13 import javax.servlet.http.HttpSession;
14 import user.user;
15
16 public class loginServlet extends HttpServlet {
17
18
19     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
20         throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22         PrintWriter out = response.getWriter();
23         try {
24             /* TODO output your page here. You may use following sample code. */
25             out.println("<html>");
26             out.println("<head>");
27             out.println("<title>Servlet loginServlet</title>");
28             out.println("</head>");
29             out.println("<body>");
30             out.println("<h1>Servlet loginServlet at " + request.getContextPath() + "</h1>");
31             out.println("</body>");
32             out.println("</html>");
33         } finally {
34             out.close();
35         }
36     }
37
38     @Override
39     protected void doGet(HttpServletRequest request, HttpServletResponse response)
40         throws ServletException, IOException {
41         //processRequest(request, response);
42         doPost(request, response);
43     }
44
45     @Override
46     protected void doPost(HttpServletRequest request, HttpServletResponse response)
47         throws ServletException, IOException {
48         //processRequest(request, response);
49         String email, pass;
```

```
50     String db_email, db_pass;
51     boolean isLoggedIn = false;
52     HttpSession userSession = request.getSession();
53     PrintWriter out = response.getWriter();
54     email = request.getParameter("email");
55     pass = request.getParameter("pass");
56     String message, messageDetail;
57     message = "";
58     messageDetail = "";
59
60     String messageUrl = "/message.jsp";
61     RequestDispatcher dispatchMessage =
62         request.getServletContext().getRequestDispatcher(messageUrl);
63
64     try {
65
66         pass = SecureSHA1.getSHA1(pass);
67         out.println("email " + email + " pass " + pass);
68         DB_Conn con = new DB_Conn();
69         Connection c = con.getConnection();
70         String sqlGetUsers = "SELECT `email` , "
71             + "`pass` FROM `user`; ";
72
73         PreparedStatement st = c.prepareStatement(sqlGetUsers);
74
75         ResultSet rs = st.executeQuery();
76
77         while (rs.next()) {
78             db_email = rs.getString("email");
79             db_pass = rs.getString("pass");
80
81             if (email.equals(db_email)) {
82                 message = "Your email-id exists with us!";
83                 //you exist with us
84                 if (pass.equals(db_pass)) {
85                     isLoggedIn = true;
86                     //user exists and password is matching
87                     out.print("You are logged in");
88                     User User = new user();
89                     User.setUserEmail(email);
90                     userSession.setAttribute("user", User);
91                     response.sendRedirect(request.getContextPath() + "/index.jsp");
92                 }
93                 else {
94                     isLoggedIn = false;
95                     // user exsts but wrong password ask to CHANGE THE PASSWORD
96                     message = "Wrong Password...!";
97                     messageDetail = "Password does not match with the password during login";
98                 }
99             }
100        }
101    }
102
103    if (!isLoggedIn) {
104        message = "Please Enter Your Email-ID and Password";
105        messageDetail = "Email-ID or Password is incorrect";
106    }
107
108    out.println(message);
109    out.println(messageDetail);
110
111 }
```

```
97         messageDetail = "Password does not match with the password d  
98         out.println("wrong password Change the password now <a href  
99             break;  
100        }  
101    }  
102    else {  
103        //or there no such email YOU do not exist with us Create an acco  
104        out.println(" no such email Register an account now!");  
105        message = "Email does not exists";  
106        messageDetail = "Please register with us right now to buy items  
107        isLoggedIn = false;  
108    }  
109}  
110  
111 if (isLoggedIn == false){  
112     request.setAttribute("message", message);  
113     request.setAttribute("messageDetail", messageDetail);  
114     dispatchMessage.forward(request, response);  
115 }  
116 } catch (SQLException e) {  
117     out.println(" Problem in the process execution...");  
118     //response.sendError(404);  
119     message = "An Error occoured during the process of login";  
120     messageDetail = "We are extremely sorry to have this but we had an error  
121  
122     request.setAttribute("message", message);  
123     request.setAttribute("messageDetail", messageDetail);  
124     dispatchMessage.forward(request, response);  
125 }  
126 } catch (Exception e) {  
127     out.println(" Problem in the process execution...");  
128     //response.sendError(404);  
129 }  
130 }  
131  
132 @Override  
133 public String getServletInfo() {  
134     return "Short description";  
135 }  
136 }  
137 }
```

❖ changeMyPass.java

The screenshot shows a Java code editor window with the title bar "changeMyPass.java". The menu bar includes "Source" and "History". Below the menu is a toolbar with various icons for file operations like save, open, and search. The code itself is a Java servlet named "changeMyPass". It starts with package declarations for "loginRegister" and several Java.sql and javax.servlet classes. The class definition begins with "public class changeMyPass extends HttpServlet {". Inside the class, there is a protected method "processRequest" that takes an "HttpServletRequest" and an "HttpServletResponse" as parameters. This method handles the response content type and prints some basic HTML structure to the output stream. A note at the bottom of the code area says "HttpServlet methods. Click on the + sign on the left to edit the code." The code is numbered from 1 to 123 on the left.

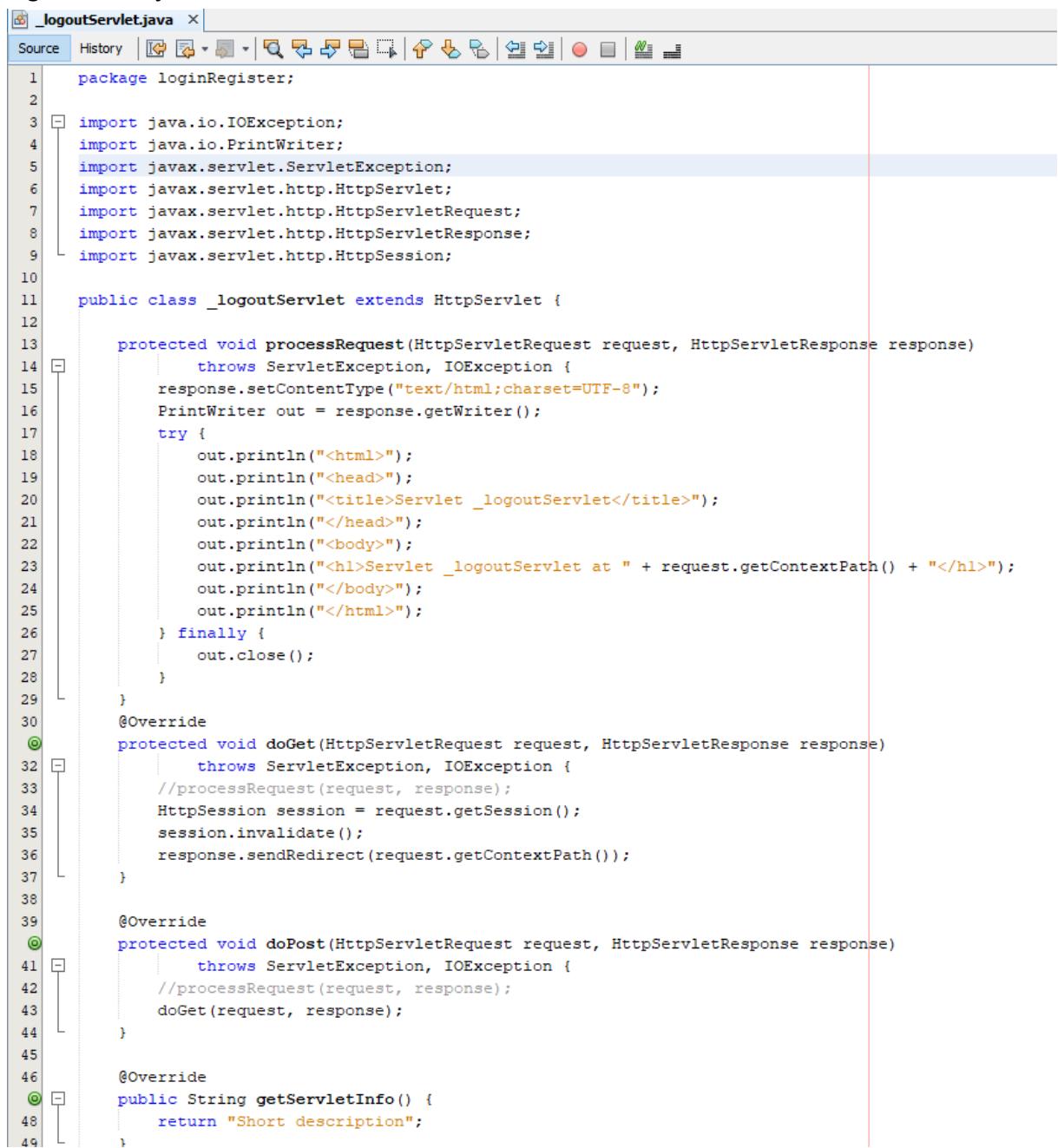
```
1 package loginRegister;
2
3
4 import database.DB_Conn;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.SQLException;
10 import java.sql.Statement;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13 import javax.servlet.ServletException;
14 import javax.servlet.http.HttpServlet;
15 import javax.servlet.http.HttpServletRequest;
16 import javax.servlet.http.HttpServletResponse;
17 import javax.servlet.http.HttpSession;
18 import user.user;
19
20 public class changeMyPass extends HttpServlet {
21
22
23     protected void processRequest(HttpServletRequest request, HttpServletResponse
24             throws ServletException, IOException {
25         response.setContentType("text/html;charset=UTF-8");
26         PrintWriter out = response.getWriter();
27         try {
28             /* TODO output your page here. You may use following sample code. */
29             out.println("<html>");
30             out.println("<head>");
31             out.println("<title>Servlet changeMyPass</title>");
32             out.println("</head>");
33             out.println("<body>");
34             out.println("<h1>Servlet changeMyPass at " + request.getContextPath());
35             out.println("</body>");
36             out.println("</html>");
37         } finally {
38             out.close();
39         }
40     }
41
42     // HttpServlet methods. Click on the + sign on the left to edit the code.
43 }
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
```

❖ registerServlet.java

The screenshot shows a Java code editor window with the file "registerServlet.java" open. The code is a servlet that extends HttpServlet and overrides the processRequest method to print an HTML response. The code includes imports for various Java packages and classes, and a try-catch block for handling exceptions. A tooltip at the bottom right of the code area provides information about HttpServlet methods.

```
1 package loginRegister;
2
3 import database.DB_Conn;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10 import helpers.*;
11 import java.sql.*;
12 import javax.servlet.RequestDispatcher;
13 import javax.servlet.http.HttpSession;
14 import user.user;
15
16 public class registerServlet extends HttpServlet {
17
18
19
20     protected void processRequest(HttpServletRequest request, HttpServletResponse
21             throws ServletException, IOException {
22         response.setContentType("text/html;charset=UTF-8");
23         PrintWriter out = response.getWriter();
24         try {
25             /*
26             * TODO output your page here. You may use following sample code.
27             */
28             out.println("<html>");
29             out.println("<head>");
30             out.println("<title>Servlet registerServlet</title>");
31             out.println("</head>");
32             out.println("<body>");
33             out.println("<h1>Servlet registerServlet at " + request.getContextPath());
34             out.println("</body>");
35             out.println("</html>");
36         } finally {
37             out.close();
38         }
39     }
40
41     // HttpServlet methods. Click on the + sign on the left to edit the code.
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
```

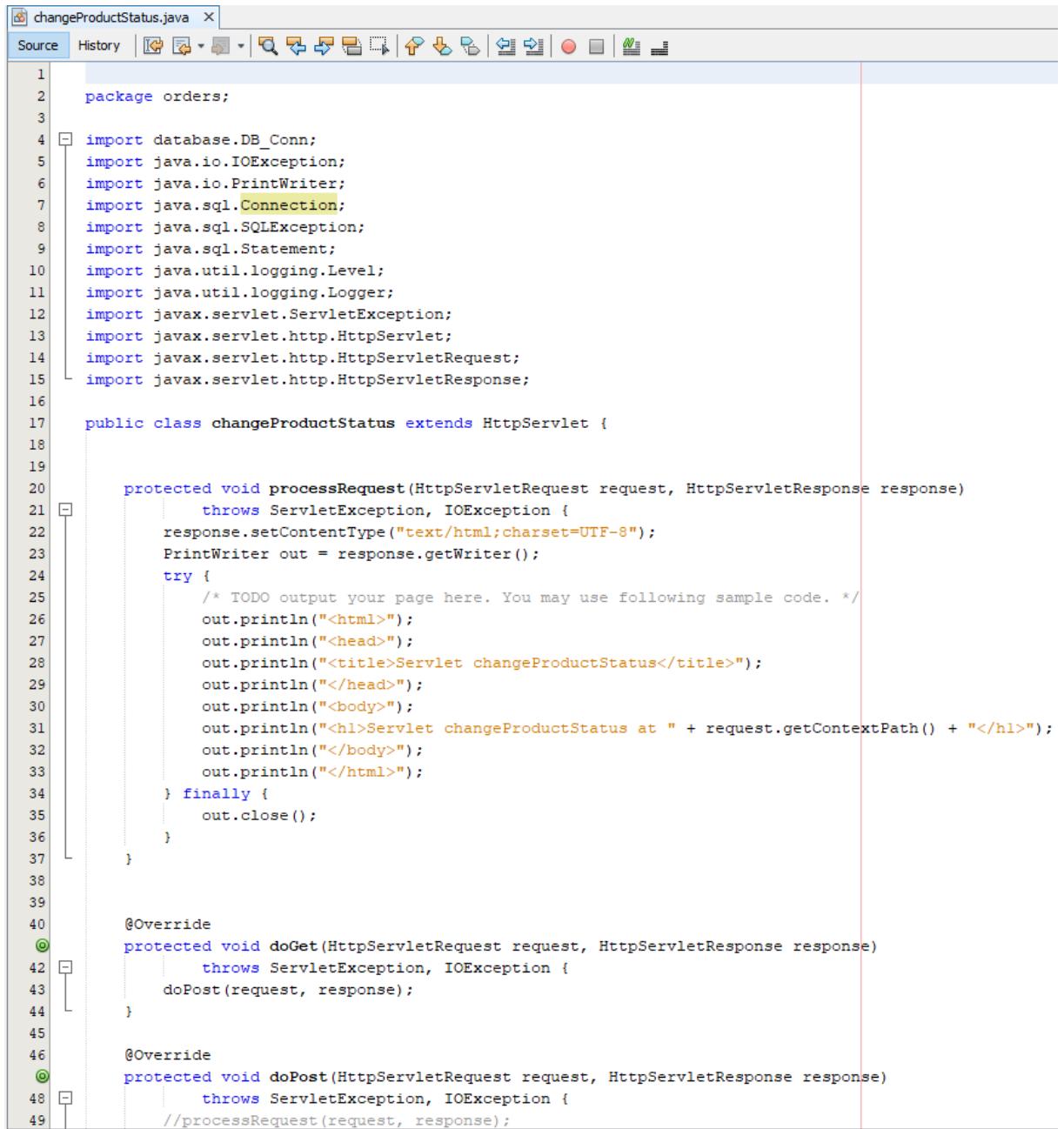
## ❖ logoutServlet.java



The screenshot shows a Java code editor with the file `_logoutServlet.java` open. The code implements a servlet for logging out users. It imports various Java and javax.servlet packages. The `processRequest` method handles both GET and POST requests by printing an HTML response that includes the servlet's name and context path. The `doGet` and `doPost` methods both call `processRequest`. The `getServletInfo` method returns a short description.

```
1 package loginRegister;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 public class _logoutServlet extends HttpServlet {
12
13     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
14             throws ServletException, IOException {
15         response.setContentType("text/html;charset=UTF-8");
16         PrintWriter out = response.getWriter();
17         try {
18             out.println("<html>");
19             out.println("<head>");
20             out.println("<title>Servlet _logoutServlet</title>");
21             out.println("</head>");
22             out.println("<body>");
23             out.println("<h1>Servlet _logoutServlet at " + request.getContextPath() + "</h1>");
24             out.println("</body>");
25             out.println("</html>");
26         } finally {
27             out.close();
28         }
29     }
30     @Override
31     protected void doGet(HttpServletRequest request, HttpServletResponse response)
32             throws ServletException, IOException {
33         //processRequest(request, response);
34         HttpSession session = request.getSession();
35         session.invalidate();
36         response.sendRedirect(request.getContextPath());
37     }
38
39     @Override
40     protected void doPost(HttpServletRequest request, HttpServletResponse response)
41             throws ServletException, IOException {
42         //processRequest(request, response);
43         doGet(request, response);
44     }
45
46     @Override
47     public String getServletInfo() {
48         return "Short description";
49     }
}
```

## ❖ changeProductsStatus.java



The screenshot shows a Java code editor window with the file "changeProductStatus.java" open. The code implements a HttpServlet for changing product status. It includes imports for various Java packages, a main class definition, a protected method for processing requests, and overridden doGet and doPost methods.

```
1 package orders;
2
3 import database.DB_Conn;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import java.sql.Connection;
7 import java.sql.SQLException;
8 import java.sql.Statement;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.servlet.ServletException;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15
16 public class changeProductStatus extends HttpServlet {
17
18
19     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
20             throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22         PrintWriter out = response.getWriter();
23         try {
24             /* TODO output your page here. You may use following sample code. */
25             out.println("<html>");
26             out.println("  <head>");
27             out.println("    <title>Servlet changeProductStatus</title>");
28             out.println("  </head>");
29             out.println("  <body>");
30             out.println("    <h1>Servlet changeProductStatus at " + request.getContextPath() + "</h1>");
31             out.println("  </body>");
32             out.println("</html>");
33         } finally {
34             out.close();
35         }
36     }
37
38
39     @Override
40     protected void doGet(HttpServletRequest request, HttpServletResponse response)
41             throws ServletException, IOException {
42         doPost(request, response);
43     }
44
45     @Override
46     protected void doPost(HttpServletRequest request, HttpServletResponse response)
47             throws ServletException, IOException {
48         //processRequest(request, response);
49     }
}
```

```

49 //processRequest(request, response);
50 String ChangeStatus = request.getParameter ("ChangeStatus");
51 String order [] = request.getParameterValues("order");
52 response.setContentType("text/html;charset=UTF-8");
53 PrintWriter out = response.getWriter();
54 out.println ("So you wanna "+ChangeStatus+ " For the Item nos of, ");
55
56 if (order != null){
57     for (int i=0; i<order.length; i++){
58         out.println ("<br/>"+order[i]);
59     }
60
61 if (ChangeStatus.equals("approved")){
62     try {
63
64         StringBuffer sqlBuffer = new StringBuffer();
65
66
67         Connection c = new DB_Conn().getConnection();
68         Statement st = c.createStatement();
69         for (int i=0; i<order.length; i++){
70             String sqlUpdatePending = "UPDATE `order` SET " +
71             " `status` = 'approved' WHERE `order`.`order_id` ='" +order[i]+ "' ; ";
72             //sqlBuffer.append(sqlUpdatePending);
73             st.addBatch(sqlUpdatePending);
74         }
75
76         out.print(" "+sqlBuffer.toString());
77         int[] executeBatch = st.executeBatch();
78         out.println (executeBatch.length +" Products Approved");
79
80
81     } catch (SQLException ex) {
82         out.print(" " +ex);
83         Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
84     } catch (ClassNotFoundException ex) {
85         out.print(" " +ex);
86         Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
87     }
88
89
90     response.sendRedirect(request.getContextPath() +"/admin_pendingOrders.jsp");
91
92 }else if (ChangeStatus.equals("delivered")){
93     //cancel the pending orders
94     try {
95
96         StringBuffer sqlBuffer = new StringBuffer();
97

```

```
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
    Connection c = new DB_Conn().getConn();
    Statement st = c.createStatement();
    for (int i=0; i<order.length; i++){
        String sqlUpdatePending = "UPDATE `order` SET " +
        " `status` = 'delivered' WHERE `order`.`order_id` ='" +order[i]+ "' ; ";
        //sqlBuffer.append(sqlUpdatePending);
        st.addBatch(sqlUpdatePending);
    }

    out.print(" "+sqlBuffer.toString());
    int[] executeBatch = st.executeBatch();
    out.println (executeBatch.length +" Products Approved");

}

} catch (SQLException ex) {
    out.print(" " +ex);
    Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
} catch (ClassNotFoundException ex) {
    out.print(" " +ex);
    Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
}

response.sendRedirect("/eCarla/admin_approvedOrders.jsp");
}
else if (ChangeStatus.equals("cancel")){
//cancel the pending orders

try {

    StringBuffer sqlBuffer = new StringBuffer();

    Connection c = new DB_Conn().getConn();
    Statement st = c.createStatement();
    for (int i=0; i<order.length; i++){

        String sqlDeleteOrder, sqlDeleteSales;
        sqlDeleteOrder = " DELETE FROM `order` WHERE `order_id` = " +order[i]+ ";";
        sqlDeleteSales = " DELETE FROM `sales` WHERE `order_id` = " +order[i]+ ";";

        st.addBatch(sqlDeleteOrder);
        st.addBatch(sqlDeleteSales);
    }

    out.print(" "+sqlBuffer.toString());
    int[] executeBatch = st.executeBatch();
    out.println (executeBatch.length +" Products Deleted");
}
```

```
135 |         sqlDeleteOrder = " DELETE FROM `order` WHERE `order_id` = "+order[i]+";";
136 |         sqlDeleteSales = " DELETE FROM `sales` WHERE `order_id` = "+order[i]+";";
137 |
138 |         st.addBatch(sqlDeleteOrder);
139 |         st.addBatch(sqlDeleteSales);
140 |     }
141 |
142 |     out.print(" "+sqlBuffer.toString());
143 |     int[] executeBatch = st.executeBatch();
144 |     out.println (executeBatch.length +" Products Deleted");
145 |
146 | } catch (SQLException ex) {
147 |     out.print(" " +ex);
148 |
149 |     Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
150 | } catch (ClassNotFoundException ex) {
151 |     out.print(" " +ex);
152 |     Logger.getLogger(changeProductStatus.class.getName()).log(Level.SEVERE, null, ex);
153 | }
154 |
155 | response.sendRedirect("/eCarla/admin_Performance.jsp");
156 |
157 | }
158 |
159 | }
160 |
161 |
162 | @Override
163 | public String getServletInfo() {
164 |     return "Short description";
165 | }
166 |
167 | }
```

## ❖ addCategory.java

The screenshot shows a Java code editor window with the file 'addCategory.java' open. The code implements a servlet for adding categories. It includes imports for various Java packages, a constructor, and methods for doGet and doPost requests. The doPost method retrieves a category name from the request parameters and prints a message to the response.

```
1 package product;
2
3
4 import database.DB_Conn;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import java.sql.*;
8 import javax.servlet.ServletException;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 public class addCategory extends HttpServlet {
14
15     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
16             throws ServletException, IOException {
17         response.setContentType("text/html;charset=UTF-8");
18         PrintWriter out = response.getWriter();
19         try {
20
21             out.println("<html>");
22             out.println("<head>");
23             out.println("<title>Servlet addCategory</title>");
24             out.println("</head>");
25             out.println("<body>");
26             out.println("<h1>Servlet addCategory at " + request.getContextPath() + "</h1>");
27             out.println("</body>");
28             out.println("</html>");
29         } finally {
30             out.close();
31         }
32     }
33
34     @Override
35     protected void doGet(HttpServletRequest request, HttpServletResponse response)
36             throws ServletException, IOException {
37         processRequest(request, response);
38     }
39
40     @Override
41     protected void doPost(HttpServletRequest request, HttpServletResponse response)
42             throws ServletException, IOException {
43         //processRequest(request, response);
44
45         String categoryName = request.getParameter("categoryName");
46         PrintWriter out = response.getWriter();
47         String message = addCategory(categoryName);
48         out.print(message);
49     }
}
```

```
51 |     private String addCategory (String categoryName){  
52 |         String message;  
53 |  
54 |         if (categoryName.equals("")){  
55 |             //out.println("Please enter in All the Values");  
56 |             message ="Please enter in a Category Name";  
57 |         }  
58 |         else {  
59 |             try {  
60 |                 //out.println("Done!!");  
61 |                 Connection con;  
62 |                 DB_Conn conn = new DB_Conn();  
63 |                 con = conn.getConnection();  
64 |                 String sqlInsertCompany ="INSERT INTO `database_ecarla`.`category` (`category_id` ,`cat  
65 |                 // "INSERT INTO `database_ecarla`.'product-company` (`company-id` ,`company-name`)VALUES  
66 |                 //String sqlInsertProduct = "INSERT INTO `database_ecarla`.'category` (`category_id` ,`  
67 |                 Statement st = con.createStatement();  
68 |                 int rows = st.executeUpdate(sqlInsertCompany);  
69 |  
70 |                 if (rows != 1){  
71 |                     // out.println("Company not inserted");  
72 |                     message = "category insertion failed.";  
73 |                 }  
74 |                 else {  
75 |                     //out.println(companyName+" Company Inserted");  
76 |                     message = categoryName+" category inserted";  
77 |                 }  
78 |                 st.close();  
79 |                 con.close();  
80 |             }  
81 |             catch (SQLException ex){  
82 |                 message = ("Please provide names without quotes");  
83 |             }  
84 |             catch (SQLIntegrityConstraintViolationException ex){  
85 |                 //out.println("A comany name with the same name exists in your database... Try being spe  
86 |                 message = "A category name with the same name exists in your database... Try being speci  
87 |             }  
88 |             catch (SQLException ex) {  
89 |                 //out.println("There was a problem in Connectiong DB <br/> Exception has occoured "+ex);  
90 |                 message = "There was a problem in Connectiong DB <br/> Exception has occoured "+ex;  
91 |             }  
92 |             catch (ClassNotFoundException ex) {  
93 |                 //out.println("Application Cannot find the Class <br/> Exception has occoured "+ex);  
94 |                 message = "Application Cannot find the Class <br/> Exception has occoured "+ex;  
95 |             }  
96 |         }  
97 |         return message;  
    }
```

❖ addCompany.java

The screenshot shows a Java code editor window with the file "addCompany.java" open. The code is a servlet for adding a company. It includes imports for various Java packages, a constructor, and two overridden methods: `processRequest` and `doPost`. The `processRequest` method handles the HTTP request and prints a simple HTML response. The `doPost` method checks if the company name is provided and prints a message if it's empty.

```
1 package product;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 import java.sql.*;
11 import database.*;
12
13
14 public class addCompany extends HttpServlet {
15
16
17     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
18             throws ServletException, IOException {
19         response.setContentType("text/html;charset=UTF-8");
20         PrintWriter out = response.getWriter();
21         try {
22             /*
23             * TODO output your page here. You may use following sample code.
24             */
25             out.println("<html>");
26             out.println("<head>");
27             out.println("<title>Servlet addCompany</title>");
28             out.println("</head>");
29             out.println("<body>");
30             out.println("<h1>Servlet addCompany at " + request.getContextPath() + "</h1>");
31             out.println("</body>");
32             out.println("</html>");
33         } finally {
34             out.close();
35         }
36     }
37
38     @Override
39     protected void doPost(HttpServletRequest request, HttpServletResponse response)
40             throws ServletException, IOException {
41         //processRequest(request, response);
42         String companyName = request.getParameter("companyName");
43         //String productName = request.getParameter("productName");
44         PrintWriter out = response.getWriter();
45         //out.print(companyName + " " + productName);
46         if (companyName.equals("")){
47             out.println("Please enter in All the Values");
48         }
49         else {
```

```
49 |
50 |     else {
51 |         try {
52 |             //out.println("Done!!!");
53 |             Connection con;
54 |             DB_Conn conn = new DB_Conn();
55 |             con = conn.getConnection();
56 |             String sqlInsertCompany = "INSERT INTO `database_ecarla`.`product-company` (`company-id` , `category_id` , `name` , `description` , `image` , `status` ) VALUES ('"+companyName+"', '"+categoryID+"', '"+companyName+"', '"+companyDescription+"', 'http://'+companyImage+'', '1') ";
57 |             Statement st = con.createStatement();
58 |             int rows = st.executeUpdate(sqlInsertCompany);
59 |
60 |             if (rows != 1){
61 |                 out.println("Company not inserted");
62 |             }else {
63 |                 out.println(companyName+" Product Company Inserted");
64 |             }
65 |             st.close();
66 |             con.close();
67 |
68 |         }
69 |
70 |         catch (SQLException ex){
71 |             out.println("Please provide names without quotes");
72 |         }
73 |         catch (SQLIntegrityConstraintViolationException ex){
74 |             out.println("Company With the same name exists...");
75 |         }
76 |
77 |         catch (SQLException ex) {
78 |             out.println("There was a problem in Connectiong DB <br/> Exception has occoured "+ex);
79 |         } catch (ClassNotFoundException ex) {
80 |             out.println("Application Cannot find the Class <br/> Exception has occoured "+ex);
81 |         }
82 |     }
83 |
84 | }
85 |
86 | @Override
87 | public String getServletInfo() {
88 |     return "Short description";
89 | }
90 |
91 }
```

❖ addProduct.java

The screenshot shows a Java code editor window with the file 'addProduct.java' open. The code is a servlet for adding products. It includes imports for various Java packages, a constructor, and methods for processing requests and handling exceptions. The code uses PrintWriter to output HTML content.

```
1 package product;
2 import java.sql.*;
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import database.*;
10 import java.util.logging.Level;
11 import java.util.logging.Logger;
12 import javax.servlet.RequestDispatcher;
13 import javax.servlet.http.HttpSession;
14
15
16
17 public class addProduct extends HttpServlet {
18
19     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
20             throws ServletException, IOException {
21         response.setContentType("text/html;charset=UTF-8");
22         PrintWriter out = response.getWriter();
23         try {
24             /*
25             * TODO output your page here. You may use following sample code.
26             */
27             out.println("<html>");
28             out.println("<head>");
29             out.println("<title>Servlet addProduct</title>");
30             out.println("</head>");
31             out.println("<body>");
32             out.println("<h1>Servlet addProduct at " + request.getContextPath() + "</h1>");
33             out.println("</body>");
34             out.println("</html>");
35         } finally {
36             out.close();
37         }
38     }
39
40     @Override
41     protected void doGet(HttpServletRequest request, HttpServletResponse response)
42             throws ServletException, IOException {
43         //processRequest(request, response);
44
45         //response.setContentType("text/html;charset=UTF-8");
46
47         String message, messageDetails, url;
48         RequestDispatcher dispatcher;
```

```
50 |         String productName, productQty, company, category, subCategory, price, dis-
51 |         company = request.getParameter("company");
52 |         productQty = request.getParameter("productQty");
53 |         productName = request.getParameter("productName");
54 |         category = request.getParameter("category");
55 |         subCategory = request.getParameter("subCategory");
56 |         price = request.getParameter("price");
57 |         tags = request.getParameter("tags");
58 |         //discountedPrice = request.getParameter("discountedPrice");
59 |         summary = request.getParameter("summary");
60 |         PrintWriter out = response.getWriter();
61 |         out.println("productName " + productName
62 |                     + " <br>company " + company
63 |                     + " <br>category " + category
64 |                     + " <br>subCategory " + subCategory
65 |                     + " <br>price " + price
66 |                     // + " <br>discountedPrice " + discountedPrice
67 |                     + " <br>summary <br>" + summary
68 |                     +" <br>summary <br>" + tags);
69 |         if (!productName.equals("") || !price.equals("")) {
70 |             try {
71 |                 String insertQuery = ""
72 |                             + "INSERT INTO "
73 |                             + "`database_ecarla`.`products` "
74 |                             + "(`product_id` ,"
75 |                             + "`product-name` ,"
76 |                             + "`sub-category-name` ,"
77 |                             + "`category-name` ,"
78 |                             + "`company-name` ,"
79 |                             + "`price` ,"
80 |                             + "`summary` ,"
81 |                             + "`tags` ,"
82 |                             + "`product_qty` ) "
83 |                             + "VALUES "
84 |                             + ""
85 |                             + "(NULL , "
86 |                             + """+productName+"", "
87 |                             + """+subCategory+"", "
88 |                             + """+category+"", "
89 |                             + """+company+"", "
90 |                             + """+price+"", "
91 |                             + """+summary+"", "
92 |                             + """+tags+"", "
93 |                             + """+productQty+"");
94 |
95 |             DB_Conn conn = new DB_Conn();
96 |             Connection con = conn.getConnection();
97 |             Statement st = con.createStatement();
```

```
97     Statement st = con.createStatement();
98
99     int rows = st.executeUpdate(insertQuery);
100    if (rows == 1){
101
102        ResultSet executeQuery = st.executeQuery("SELECT `product_id` "
103                                         + " FROM `products` "
104                                         + " WHERE `product-name` = '" +pr
105        String ProductId = "";
106        while (executeQuery.next()){
107            ProductId = executeQuery.getString("product_id");
108
109        }
110        HttpSession session = request.getSession();
111        session.setAttribute("productName", productName);
112        session.setAttribute("productId", ProductId);
113        out.print("Product inserted");
114        message = "Product Inserted...";
115        messageDetails = company +" "+ productName+ ""
116                                         + " Product Inserted Successfully... ";
117
118        url ="/productInsertImages.jsp";
119        request.setAttribute("productName", productName);
120        request.setAttribute("message", message);
121        request.setAttribute("messageDetails", messageDetails);
122        dispacher =
123        getServletContext().getRequestDispatcher(url);
124        dispacher.forward(request, response);
125    }
126    st.close();
127    con.close();
128}
129catch (SQLException ex) {
130    //out.println("SQL Exception "+ex);
131    message = "Product Not Inserted...";
132    messageDetails = company +" "+ productName+ ""
133                                         + " Product not Inserted because of a Database error";
134    out.print("Product not inserted "+ ex);
135    url ="/productInsertImages.jsp";
136    request.setAttribute("message", message);
137    request.setAttribute("messageDetails", messageDetails);
138    dispacher =
139    getServletContext().getRequestDispatcher(url);
140    //dispacher.forward(request, response);
141
142    //Logger.getLogger(addProduct.class.getName()).log(Level.SEVERE, null, ex);
143} catch (ClassNotFoundException ex) {
144    //out.println("Class Not Found Exception "+ex);
```

```
145             message = "Product Not Inserted...";  
146             messageDetails = company +" "+ productName+ ""  
147                     + " Product not Inserted because of an error";  
148  
149             url ="/productInsertImages.jsp";  
150             request.setAttribute("message", message);  
151             request.setAttribute("messageDetails", messageDetails);  
152             dispacher =  
153                 getServletContext().getRequestDispatcher(url);  
154             dispacher.forward(request, response);  
155         }  
156     }  
157 }  
158 else {  
159     //out.println("Please Enter Product-name and Price");  
160     message = "Product Not Inserted...";  
161     messageDetails = "Please Enter Product-name and Price";  
162     out.print("Product not inserted " + " "+message);  
163     url ="/productInsertImages.jsp";  
164     request.setAttribute("message", message);  
165     request.setAttribute("messageDetails", messageDetails);  
166     dispacher =  
167         getServletContext().getRequestDispatcher(url);  
168     dispacher.forward(request, response);  
169 }  
170 }  
171  
172  
173     @Override  
174     protected void doPost(HttpServletRequest request, HttpServletResponse response)  
175             throws ServletException, IOException {  
176         //processRequest(request, response);  
177         doGet(request, response);  
178     }  
179  
180     @Override  
181     public String getServletInfo() {  
182         return "Short description";  
183     }  
184 }  
185 }
```

❖ addSubCategory.java

The screenshot shows a Java code editor window with the file 'addSubCategory.java' open. The code implements a servlet for adding a subcategory. It imports various Java packages including database.DB\_Conn, java.io.IOException, java.io.PrintWriter, java.sql.\*, javax.servlet.ServletException, javax.servlet.http.HttpServlet, javax.servlet.http.HttpServletRequest, and javax.servlet.http.HttpServletResponse. The class 'addSubCategory' extends HttpServlet. It overrides the processRequest and doPost methods. The processRequest method handles the HTTP request by printing an HTML response. The doPost method retrieves parameters from the request, calls a helper method addSubCategory to handle the addition, and prints the result. The code includes TODO comments for outputting the page.

```
1 package product;
2
3 import database.DB_Conn;
4 import java.io.IOException;
5 import java.io.PrintWriter;
6 import java.sql.*;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 public class addSubCategory extends HttpServlet {
13
14     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
15             throws ServletException, IOException {
16         response.setContentType("text/html;charset=UTF-8");
17         PrintWriter out = response.getWriter();
18         try {
19             /*
20             * TODO output your page here. You may use following sample code.
21             */
22             out.println("<html>");
23             out.println("<head>");
24             out.println("<title>Servlet addSubCategory</title>");
25             out.println("</head>");
26             out.println("<body>");
27             out.println("<h1>Servlet addSubCategory at " + request.getContextPath() + "</h1>");
28             out.println("</body>");
29             out.println("</html>");
30         } finally {
31             out.close();
32         }
33     }
34
35     @Override
36     protected void doPost(HttpServletRequest request, HttpServletResponse response)
37             throws ServletException, IOException {
38         //processRequest(request, response);
39
40         String SubCategoryName = request.getParameter("SubCategoryName");
41
42         String categoryName = request.getParameter("categoryName");
43
44         PrintWriter out = response.getWriter();
45         String message = addSubCategory(SubCategoryName, categoryName);
46         out.print(message);
47     }
48 }
```

```

49
50     private String addSubCategory (String SubCategoryName, String categoryName){
51         String message;
52
53         if (SubCategoryName.equals("")){
54             //out.println("Please enter in All the Values");
55             message ="Please enter in a SubCategory name for "+categoryName ;
56         }
57         else {
58             try {
59                 //out.println("Done!!!");
60                 Connection con;
61                 DB_Conn conn = new DB_Conn();
62                 con = conn.getConnection();
63
64                 String sqlInsertCompany ="INSERT INTO `database_ecarla`.`sub-category` (`subcategory_id` ,`sub-cat
65                 // "INSERT INTO `database_ecarla`.`product-company` (`company-id` ,`company-name`)VALUES (NULL , '
66                 //String sqlInsertProduct = "INSERT INTO `database_ecarla`.`category` (`category_id` ,`category_na
67                 Statement st = con.createStatement();
68                 int rows = st.executeUpdate(sqlInsertCompany);
69
70                 if (rows != 1){
71                     // out.println("Company not inserted");
72                     message = "Sub Category insertion cancelled.";
73                 }else {
74                     //out.println(companyName+" Company Inserted");
75                     message = SubCategoryName+" Sub-Category inserted" ;
76                 }
77                 st.close();
78                 con.close();
79             }
80             catch (SQLException ex){
81                 //out.println("A comany name with the same name exists in your database... Try being specific.");
82                 message = "A SubCategory Name with the same name exists in your database... Try being specific.";
83             }
84             catch (SQLSyntaxErrorException ex){
85                 //out.println("A comany name with the same name exists in your database... Try being specific.");
86                 message = "Please provide names without quotes";
87             }
88             catch (SQLException ex) {
89                 //out.println("There was a problem in Connectiong DB <br/> Exception has occoured "+ex);
90                 message = "There was a problem in Connectiong DB <br/> Exception has occoured "+ex;
91             }
92             } catch (ClassNotFoundException ex) {
93                 //out.println("Application Cannot find the Class <br/> Exception has occoured "+ex);
94                 message = "Application Cannot find the Class <br/> Exception has occoured "+ex;
95             }
96         }
97         //out.println(companyName+" Company Inserted");
98         message = SubCategoryName+" Sub-Category inserted" ;
99
100        }
101        catch (SQLException ex){
102            //out.println("A comany name with the same name exists in your database... Try being specific.");
103            message = "A SubCategory Name with the same name exists in your database... Try being specific.";
104        }
105        catch (SQLSyntaxErrorException ex){
106            //out.println("A comany name with the same name exists in your database... Try being specific.");
107            message = "Please provide names without quotes";
108        }
109        catch (SQLException ex) {
110            //out.println("There was a problem in Connectiong DB <br/> Exception has occoured "+ex);
111            message = "There was a problem in Connectiong DB <br/> Exception has occoured "+ex;
112        }
113        } catch (ClassNotFoundException ex) {
114            //out.println("Application Cannot find the Class <br/> Exception has occoured "+ex);
115            message = "Application Cannot find the Class <br/> Exception has occoured "+ex;
116        }
117    }
118
119    @Override
120    public String getServletInfo() {
121        return "Short description";
122    }
123
124}

```

❖ **buyItems.java**

```
1 package product;
2
3 import cart.cart;
4 import database.DB_Conn;
5 import java.io.IOException;
6 import java.io.PrintWriter;
7 import java.sql.Connection;
8 import java.sql.PreparedStatement;
9 import java.sql.ResultSet;
10 import java.sql.SQLException;
11 import java.sql.Statement;
12 import java.util.ArrayList;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15 import javax.servlet.RequestDispatcher;
16 import javax.servlet.ServletException;
17 import javax.servlet.annotation.WebServlet;
18 import javax.servlet.http.HttpServlet;
19 import javax.servlet.http.HttpServletRequest;
20 import javax.servlet.http.HttpServletResponse;
21 import javax.servlet.http.HttpSession;
22 import user.user;
23
24
25 @WebServlet(name = "buyItems", urlPatterns = {"/*buyItems"})
26 public class buyItems extends HttpServlet {
27
28     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30         response.setContentType("text/html;charset=UTF-8");
31         PrintWriter out = response.getWriter();
32         try {
33             /* TODO output your page here. You may use following sample code. */
34             out.println("<html>");
35             out.println("<head>");
36             out.println("<title>Servlet buyItems</title>");
37             out.println("</head>");
38             out.println("<body>");
39             out.println("<h1>Servlet buyItems at " + request.getContextPath() + "</h1>");
40             out.println("</body>");
41             out.println("</html>");
42         } finally {
43             out.close();
44         }
45     }
46
47     @Override
48     protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
49     ... throws ServletException, IOException {
50     ... //processRequest(request, response);
51     ... doPost(request, response);
52 }
53
54 @Override
55 protected void doPost(HttpServletRequest request, HttpServletResponse response)
56     ... throws ServletException, IOException {
57
58     PrintWriter out = response.getWriter();
59
60
61
62
63     // GETTING ALL THE VALUES FROM THE USER
64     String name, age, address, mobile;
65     int order_id;
66     Connection c = null;
67     HttpSession session = request.getSession();
68     user User;
69     User = (user) session.getAttribute("user");
70     cart Cart;
71     Cart = (cart) session.getAttribute("cart");
72
73     name = request.getParameter("name");
74     age = request.getParameter("age");
75     address = request.getParameter("address");
76     mobile = request.getParameter("mobile");
77
78
79     if (name.trim().length() > 1
80         && address.trim().length() >= 5
81         && mobile.trim().length() >= 5
82         && mobile.trim().length() <= 12) {
83
84         if (!(session.getAttribute("user") == null)
85             && !(session.getAttribute("cart") == null)) {
86
87             try {
88                 response.setContentType("text/html;charset=UTF-8");
89
90                 c = new DB_Conn().getConnection();
91
92                 //***** Starting a Transaction
93                 c.setAutoCommit(false);
94                 String insertOrder;
95                 insertOrder = "    INSERT INTO `database_ecarla`.`order` ("
96                 ... + "    `order_id` ,"
```

```

97
98
99
100
101
102
103
104
105
106
107
108
109
110     PreparedStatement preparedSQL1 =
111         c.prepareStatement(insertOrder);
112
113     preparedSQL1.setInt(1, User.getUserId()); // user id
114
115     preparedSQL1.setString(2, name); //`shippers_name`
116
117     preparedSQL1.setString(3, address); //`address`
118
119     preparedSQL1.setString(4, mobile); //`mobile`
120
121     preparedSQL1.setString(5, User.getUserEmail()); //`shippers_email`
122
123     preparedSQL1.setDouble(6, Cart.getTotalPriceOfCart()); //`total_order_price`
124
125     int executeUpdatePreparedSQL1 = preparedSQL1.executeUpdate();
126
127     if (executeUpdatePreparedSQL1 == 1) {
128         //get the latest order id of the recent update
129
130         //out.println("Ok here we are onr order updted");
131         String getLatestOrderId = "SELECT `order_id` "
132             + "FROM `order` "
133             + "WHERE `user_id` = " + User.getUserId() + " "
134             + "ORDER BY `ordered_On` DESC; "
135             + "";
136
137         preparedSQL1.close();
138
139         Statement st = c.createStatement();
140         ResultSet executeQueryGetOrderId = st.executeQuery(getLatestOrderId);
141         executeQueryGetOrderId.next();
142
143         //Here we get the latest order id for the current user
144         order_id = executeQueryGetOrderId.getInt("order_id");

```

```
145
146     executeQueryGetOrderId.close();
147
148     //out.println("you user :" + User.getUserEmail() + " has bought items worth " + C
149
150     ArrayList<String> productCategorys = Cart.getProductCategorys();
151     ArrayList<String> productNames = Cart.getProductNames();
152     ArrayList<Double> prices = Cart.getPrices();
153     ArrayList<Integer> qty = Cart.getQty();
154     ArrayList<Integer> id = Cart.getId();
155
156     String insertIntoSalesSQL3 = ""
157         + "INSERT INTO `database_ecarla`.`sales` ("
158         + "`sales_id` ,"
159         + "`order_id` ,"
160         + "`product_id` ,"
161         + "`product_name` ,"
162         + "`product_price` ,"
163         + "`product_quantity` ,"
164         + "`sold_on` ,"
165         + "`user_id`"
166         + ")"
167         + "VALUES ("
168         + "    NULL ,  ?,  ?,  ?,  ?,  ?, NOW( ) ,  ? "
169         + "    );";
170
171     PreparedStatement insertIntoSalesTable = c.prepareStatement(insertIntoSalesSQL3);
172
173     for (int i = 0; i < productNames.size(); i++) {
174
175         insertIntoSalesTable.setInt(1, order_id);
176
177         insertIntoSalesTable.setInt(2, id.get(i));
178
179         insertIntoSalesTable.setString(3, productNames.get(i));
180
181         insertIntoSalesTable.setDouble(4, prices.get(i));
182
183         insertIntoSalesTable.setInt(5, qty.get(i));
184
185         insertIntoSalesTable.setInt(6, User.getUserId());
186
187         int executeUpdateSalesTable = insertIntoSalesTable.executeUpdate();
188
189         if (executeUpdateSalesTable == 1) {
190             out.println("Success");
191
192         } else {
```

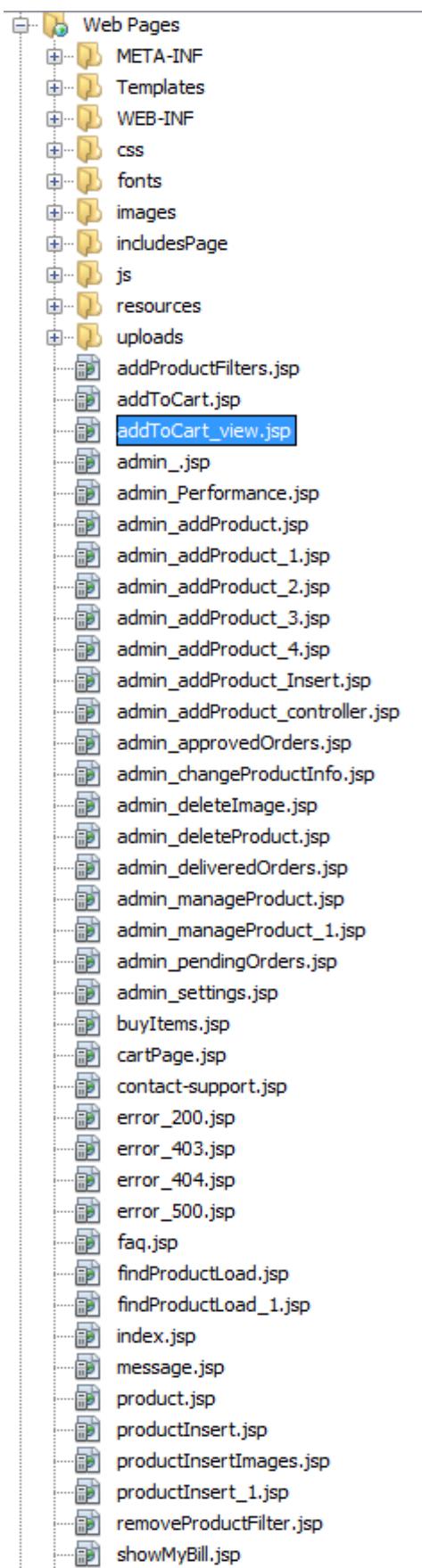
```

193         out.println("Failed for now Product name " + productNames.get(i));
194     }
195 }
196 //DECREMENT THE QUANTITY in the CART FROM products TABLE
197
198 for (int i = 0; i < productNames.size(); i++) {
199     st.addBatch(
200         " UPDATE `products` "
201         + " SET "
202         + "`product_qty` = `product_qty` - " + qty.get(i) + ""
203         + " WHERE `products`.`product_id` = " + id.get(i) + " "
204         + " AND "
205         + " `product-name` = '" + productNames.get(i) + "' ");
206     }
207 }
208 int[] executeBatch = st.executeBatch();
209 int i = 0;
210 while (i < executeBatch.length) {
211     out.print("?" + executeBatch[i]);
212     i++;
213 }
214 c.commit();
215 response.sendRedirect(request.getContextPath() + "/userinfo.jsp");
216
217 } else {
218     //not updated
219 response.sendRedirect(request.getContextPath());
220 }
221
222 } catch (SQLException ex) {
223     try {
224         Logger.getLogger(buyItems.class.getName()).log(Level.SEVERE, null, ex);
225         c.rollback();
226         String message, messageDetail;
227         String messageUrl = "/message.jsp";
228         RequestDispatcher dispatchMessage =
229             request.getServletContext().getRequestDispatcher(messageUrl);
230         message = "Oops, Less Product Stock...!";
231         messageDetail = "We see that your demand for the product was critically higher";
232         request.setAttribute("message", message);
233         request.setAttribute("messageDetail", messageDetail);
234         dispatchMessage.forward(request, response);
235         //response.sendError(500);
236
237         //response.sendRedirect("/eCarla/buyItems.jsp");
238     } catch (SQLException exl) {
239         //Logger.getLogger(buyItems.class.getName()).log(Level.SEVERE, null, exl);
240     }
}

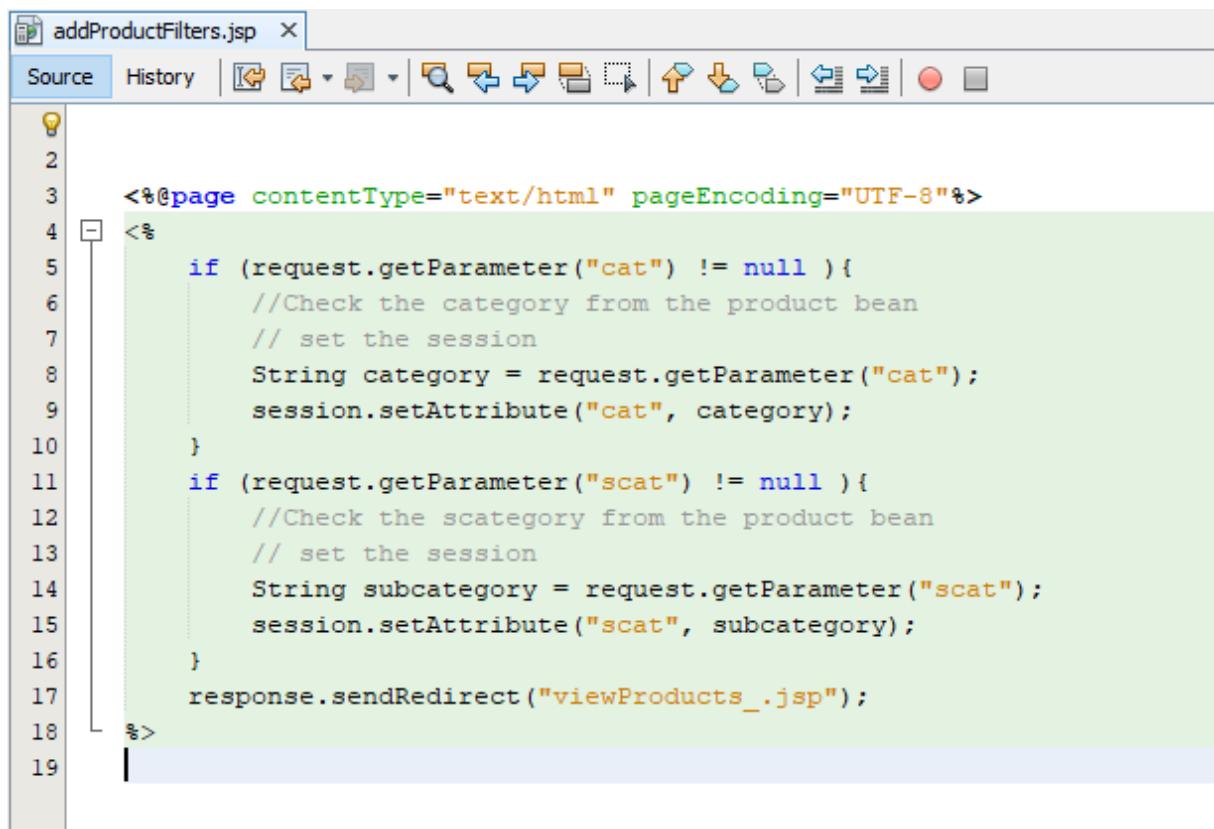
```

```
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275 }  
  
    messageDetail = "We see that your demand for the product was critical  
    request.setAttribute("message", message);  
    request.setAttribute("messageDetail", messageDetail);  
    dispatchMessage.forward(request, response);  
        //response.sendError(500);  
  
        //response.sendRedirect("/eCarla/buyItems.jsp");  
    } catch (SQLException ex) {  
        //Logger.getLogger(buyItems.class.getName()).log(Level.SEVERE, n  
  
        String message, messageDetail;  
        String messageUrl = "/message.jsp";  
        RequestDispatcher dispatchMessage =  
            request.getServletContext().getRequestDispatcher(messageUrl);  
        message = "Oops, Less Product Stock...!";  
        messageDetail = "We see that your demand for the product was critical";  
        request.setAttribute("message", message);  
        request.setAttribute("messageDetail", messageDetail);  
        dispatchMessage.forward(request, response);  
            //response.sendError(500);  
    }  
    } catch (ClassNotFoundException ex) {  
        out.println("you user " + ex);  
        //response.sendRedirect("/eCarla/buyItems.jsp");  
  
    }  
    session.removeAttribute("cart");  
} else {  
    //response.sendRedirect("/eCarla/index.jsp");  
    out.println ("No items in cart");  
}  
  
} else {  
    //response.sendRedirect("/eCarla/buyItems.jsp");  
    out.println ("NOT validated");  
}  
}  
  
}@Override  
public String getServletInfo() {  
    return "Short description";  
}
```

## View – JSP Sınıfları



❖ addProductFilters.jsp



The screenshot shows a Java IDE interface with the file "addProductFilters.jsp" open. The window title is "addProductFilters.jsp". The toolbar includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The menu bar has "Source" selected. The code editor displays the following JSP code:

```
2
3 <%@page contentType="text/html" pageEncoding="UTF-8"%>
4 <%
5     if (request.getParameter("cat") != null ){
6         //Check the category from the product bean
7         // set the session
8         String category = request.getParameter("cat");
9         session.setAttribute("cat", category);
10    }
11    if (request.getParameter("scat") != null ){
12        //Check the scategory from the product bean
13        // set the session
14        String subcategory = request.getParameter("scat");
15        session.setAttribute("scat", subcategory);
16    }
17    response.sendRedirect("viewProducts_.jsp");
18 %>
```

## ❖ addToCart.jsp

```
<%@page import="java.util.ArrayList"%>
<%@page import="user.user"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>eCarla Car - Cart</title>
        <!-- Default Stylesheets -->
        <link rel="shortcut icon" href="images/logo/ico.ico"/>
        <link rel="stylesheet" type="text/css" href="css/reset.css"/>
        <link rel="stylesheet" type="text/css" href="css/text.css"/>
        <link rel="stylesheet" type="text/css" href="css/960_16.css"/>
        <link rel="stylesheet" type="text/css" href="css/styles.css"/>
        <link rel="stylesheet" type="text/css" href="css/product.css" />
        <script type="text/javascript" src="js/jquery.js"></script>
        <script type="text/javascript" src="js/myScript.js"></script>
    <style type="text/css">
        #loginBtn {
            margin:15px auto;
            padding:5px;
            color:#000;
            text-shadow: 0 1px 0 white;
            border-radius:5px;
            background: -webkit-linear-gradient(top, #aee800, #8fbe00);
            background: -moz-linear-gradient(top, #aee800, #8fbe00);
            text-align:center;
            box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
            -webkit-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
            -moz-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
        }
        #loginBtn:hover {
            color:#000;
            border-radius:7px;
            text-shadow: 0 -1px 0 white;
            text-align:center;
            background: -webkit-linear-gradient(top, #8fbe00,#aee800);
            background: -moz-linear-gradient(top, #8fbe00,#aee800);
            box-shadow: 0 1px 4px rgba(0, 0, 0, 0.4);
            cursor:pointer;
        }
    </style>
</head>
```

```

49      </head>
50      <body>
51          <jsp:useBean id="cart" scope="session" class="cart.cart"></jsp:useBean>
52
53          <%
54              if (session.getAttribute("user") == null) { // Then new user, show join now
55          %>
56              <jsp:include page="includesPage/_joinNow.jsp"></jsp:include>
57          <%
58              } else {
59          %>
60              <jsp:include page="includesPage/_logout.jsp"></jsp:include>
61          <%
62              }
63
64          <%@include file="includesPage/_search_navigationbar.jsp" %>
65
66          <%@include file="includesPage/_facebookJoin.jsp" %>
67
68          <div class="container_16">
69              <div class="grid_16" id="whiteBox">
70                  <div class="grid_8 push_3" >
71                      <h1 class="push_2" style="padding:10px 00px">Cars In your Cart</h1>
72
73                      <%
74                          user User = new user();
75                          session.getAttribute("user");
76                          //out.println(session.getAttribute("user"));
77                          User = (user) session.getAttribute("user");
78                          if (session.getAttribute("user") == null) {
79
80                          %>
81                          <h3 class="showForm" id="loginBtn" style="padding:10px 00px">Please login</h3>
82
83                          <%
84                          } else {
85                          //out.println("login by " + User.getUserEmail());
86
87                          %>
88                          <h3 class="push_2" >Your Cart contains following...</h3>
89
90                          <%
91                          String sid = request.getParameter("id");
92                          int id;
93                          if (request.getParameter("id") == null) {
94                              response.sendRedirect("viewProduct_.jsp");
95                          } else {
96                              id = Integer.parseInt(sid);
97
98                              boolean b = cart.addProduct(id);
99
100                         %>
101                     <%
102                         if (b) {
103                             out.println("Product added successfully");
104                         } else {
105                             out.println("Product addition failed");
106                         }
107                     %>
108                 <%
109                 }
110             <%
111             }
112         <%
113         }
114     <%
115     }
116 
```

```
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114    %>
115
116
117    <div id="CartTable" style="padding:10px 00px" class="grid_11">
118        <div class="grid_5">
119            <h2>Name Of Product</h2>
120        </div>
121        <div class="grid_2">
122            <h2>Price</h2>
123        </div>
124        <div class="grid_2">
125            <h2>Quantity</h2>
126        </div>
127
128        <%
129            for (int i = 0; i < productNames.size(); i++) {
130        %>
131
132            <div class="grid_5">
133                <%=productNames.get(i)%>
134            </div>
135            <div class="grid_2">
136                &#163; <%=productPrices.get(i)%>
137            </div>
138            <div class="grid_1">
139                x<%=Qty.get(i)%>
140            </div>
141            <div class="grid_2">
142                &#163; <%= Qty.get(i) * productPrices.get(i)%>
143            </div>
144
145        <%
```

```
145      }
146      productNames.clear();
147      productPrices.clear();
148
149
150      %>
151      <br/>
152
153      <div class="grid_5">
154          <strong>Total Price of your Cart</strong>
155      </div>
156
157      <div class="grid_3 push_3">
158          &#163;<%= Math.ceil(cart.getTotalPriceOfCart()) %>
159      </div>
160      <div class="clear"></div>
161
162      <br/>
163      <br/>
164      <a href="buyItems.jsp">
165          <div class="grid_3" id="greenBtn">
166              Buy These Items
167          </div>
168      </a>
169      <a href="viewProducts_.jsp">
170          <div class="grid_3" id="greenBtn">
171              Continue Shopping
172          </div>
173      </a>
174  </div>
175
176      <br/>
177      <br/>
178
179
180      <br/>
181      <%
182          }
183      %}
184      %>
185  </div>
186
187      </div>
188  </div>
189  </body>
190
191
```

## ❖ addToCart\_view.jsp

```
1 | 
2 | <%@page import="java.util.ArrayList"%>
3 | <%@page import="user.user"%>
4 | <%@page contentType="text/html" pageEncoding="UTF-8"%>
5 | <!DOCTYPE html>
6 | <html>
7 |   <head>
8 |     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
9 |     <title>eCarla Car - Cart</title>
10 |     <!-- Default Stylesheets -->
11 |     <link rel="shortcut icon" href="images/logo/ico.ico"/>
12 |
13 |     <link rel="stylesheet" type="text/css" href="css/reset.css"/>
14 |     <link rel="stylesheet" type="text/css" href="css/text.css"/>
15 |     <link rel="stylesheet" type="text/css" href="css/960_16.css"/>
16 |     <link rel="stylesheet" type="text/css" href="css/styles.css"/>
17 |     <link rel="stylesheet" type="text/css" href="css/product.css" />
18 |
19 |     <script type="text/javascript" src="js/jquery.js"></script>
20 |
21 |     <script type="text/javascript" src="js/myScript.js"></script>
22 |
23 |     <style type="text/css">
24 |       #loginBtn {
25 |         margin:15px auto;
26 |         padding:5px;
27 |         color:#000;
28 |         text-shadow: 0 1px 0 white;
29 |         border-radius:5px;
30 |         background: -webkit-linear-gradient(top, #aee800, #8fbe00);
31 |         background: -moz-linear-gradient(top, #aee800, #8fbe00);
32 |         text-align:center;
33 |         box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
34 |         -webkit-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
35 |         -moz-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.27);
36 |
37 |       }
38 |       #loginBtn:hover {
39 |         color:#000;
40 |         border-radius:7px;
41 |         text-shadow: 0 -1px 0 white;
42 |         text-align:center;
43 |         background: -webkit-linear-gradient(top, #8fbe00, #aee800);
44 |         background: -moz-linear-gradient(top, #8fbe00, #aee800);
45 |         box-shadow: 0 1px 4px rgba(0, 0, 0, 0.4);
46 |         cursor:pointer;
47 |       }
48 |     </style>
49 |   </head>
```

```
49 </head>
50 <body>
51     <jsp:useBean id="cart" scope="session" class="cart.cart"/>
52
53     <%
54         if (session.getAttribute("user") == null) { // Then new user, show join now
55     %>
56     <jsp:include page="includesPage/_joinNow.jsp"></jsp:include>
57     <% } else {
58     %>
59     <jsp:include page="includesPage/_logout.jsp"></jsp:include>
60     <% }
61     %>
62
63     <%@include file="includesPage/_search_navigationbar.jsp" %>
64
65     <%@include file="includesPage/_facebookJoin.jsp" %>
66
67     <div class="container_16">
68         <div class="grid_16" id="whiteBox">
69             <div class="grid_8 push_3" >
70                 <h1 class="push_2" style="padding:10px 00px">Products In your Cart</h1>
71
72             <%
73                 User User = new User();
74                 session.getAttribute("user");
75                 //out.println(session.getAttribute("user"));
76                 User = (User) session.getAttribute("user");
77                 if (session.getAttribute("user") == null) {
78             %>
79                 <h3 class="showForm" id="loginBtn" style="padding:10px 00px">Please Log In</h3>
80
81             <% } else {
82                 //out.println("login by " + User.getUserEmail());
83
84             %>
85                 <h3 class="push_2" >Your Cart contains following...</h3>
86
87
88             <%
89
90                 out.println("<br/>Total value price of the cart " + cart.getTotalValue());
91                 ArrayList<String> productNames = new ArrayList();
92                 ArrayList<Double> productPrices = new ArrayList();
93                 ArrayList<Integer> Qty = new ArrayList();
94                 ArrayList<Integer> ids = new ArrayList();
95
96                 productNames = cart.getProductNames();
97                 productPrices = cart.getPrices();
```

```
97          productPrices = cart.getPrices();
98          Qty = cart.getQty();
99          ids = cart.getId();
100         %>
101
102         <div id="CartTable" style="padding:10px 00px" class="grid_11">
103             <div class="grid_5">
104                 <h2>Name Of Product</h2>
105             </div>
106             <div class="grid_2">
107                 <h2>Price</h2>
108             </div>
109             <div class="grid_2">
110                 <h2>Quantity</h2>
111             </div>
112
113
114         <%
115             for (int i = 0; i < productNames.size(); i++) {
116             %>
117
118             <div class="grid_5">
119                 <%=productNames.get(i)%>
120             </div>
121             <div class="grid_2">
122                 &#163; <%=productPrices.get(i)%>
123             </div>
124             <div class="grid_1">
125                 x<%=Qty.get(i)%>
126             </div>
127             <div class="grid_2">
128                 &#163; <%= Qty.get(i) * productPrices.get(i)%>
129             </div>
130
131         <%
132             }
133             productNames.clear();
134             productPrices.clear();
135
136         %>
137         <br/>
138
139         <div class="grid_5">
140             <strong>Total Price of your Cart</strong>
141         </div>
142
143         <div class="grid_3 push_3">
144             &#163;<%= Math.ceil(cart.getTotalPriceOfCart()) %>
145         </div>
```

```
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
```

## ❖ admin\_addProduct\_Insert.jsp

```
<%@page import="java.sql.SQLIntegrityConstraintViolationException"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="database.DB_Conn"%>
<%@page import="javazoom.upload.DBStore"%>
<%@page import="java.sql.Connection"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <jsp:useBean class="product.productBean" scope="session" id="productBean"></jsp:useBean>
        <%
            if (productBean.getCompany() == null ||
                productBean.getCompany() == ""){
                response.sendRedirect("admin_addProduct_1.jsp");
            } else if ( productBean.getCategory() == null ||
                        productBean.getCategory() == "" ||
                        productBean.getSubcategory() == null ||
                        productBean.getSubcategory() == ""){
                response.sendRedirect("admin_addProduct_2.jsp");
            } else if (productBean.getName() == null ||
                        productBean.getName() == "" ||
                        productBean.getTags() == null ||
                        productBean.getTags() == ""){
                response.sendRedirect("admin_addProduct_3.jsp");
            } else if (productBean.getPrice() == null ||
                        productBean.getPrice() == "" ||
                        productBean.getQuantity() == null ||
                        productBean.getQuantity() == "" ||
                        productBean.getSummary() == null ||
                        productBean.getSummary() == ""){
                response.sendRedirect("admin_addProduct_4.jsp");
            }
        %>
        double price = 0;
        int quantity = 0;
        try {
            price = Double.parseDouble(productBean.getPrice());
            quantity = Integer.parseInt(productBean.getQuantity());
        
```

```
-- 54      <%
55          /*
56              INSERT INTO `database_ecarla`.`products` (
57                  `product_id` ,
58                  `product-name` ,
59                  `sub-category-name` ,
60                  `category-name` ,
61                  `company-name` ,
62                  `price` ,
63                  `summary` ,
64                  `tags` ,
65                  `product_qty` ,
66                  `lastUpdated`
67              )
68              VALUES (
69                  NULL , 'Assassins Cedd' , 'Action-Game' , 'Games' , 'EA, Electronic Arts' , '3' , 'dcdcddcdcc
70              );
71
72          //Fetch the product id
73          SELECT `product_id`
74          FROM `products`
75          WHERE `product-name` = 'Assassins Creed'
76
77          //Insert into expenses
78          INSERT INTO `database_ecarla`.`expenses` (
79              `expenses_id` ,
80              `product_id` ,
81              `product_name` ,
82              `price` ,
83              `purchase_date`
84          )
85          VALUES (
86              NULL , '2' , 'Ayinga Movie ' , '123456' , NOW( )
87          );
88      */
89      String insertQuery = ""
90          +
91          "INSERT INTO "
92          +
93          "`database_ecarla`.`products` "
94          +
95          "("`product_id` ,
96          + "`product-name` ,"
97          + "`sub-category-name` ,"
98          + "`category-name` ,"
99          + "`company-name` ,"
+ "`price` ,"
+ "`summary` , "
+ "`tags` , "
```

```

100|           + "`product_qty` , "
101|           + "`lastUpdated` , "
102|           + "`hits` ) "
103|           + "VALUES "
104|           +
105|           + "(NULL , "
106|           + "'"+productBean.getName() +"', "
107|           + "'"+productBean.getSubcategory()+"', "
108|           + "'"+productBean.getCategory()+"', "
109|           + "'"+productBean.getCompany()+"', "
110|           + "'"+productBean.getPrice()+"', "
111|           + "'"+productBean.getSummary()+"', "
112|           + "'"+productBean.getTags()+"', "
113|           + "'"+productBean.getQuantity()+"', "
114|           + "NOW( ), "
115|           + " 0 );";
116|
117|       out.print(""+insertQuery);
118|
119|       Connection c = new DB_Conn().getConnection();
120|       Statement st = c.createStatement();
121|       st.execute(insertQuery);
122|       ResultSet executeQuery = st.executeQuery("SELECT `product_id` "
123|           + " FROM `products` "
124|           + " WHERE `product-name` = '"+productBean.getName()+"'");
125|       String ProductId = "";
126|       while (executeQuery.next()){
127|           ProductId = executeQuery.getString("product_id");
128|       }
129|       session.setAttribute("productName", productBean.getName());
130|       session.setAttribute("productId", ProductId);
131|
132|
133|
134|       ResultSet executeQueryFetchPid = st.executeQuery("SELECT `product_id` "
135|           + " FROM `products` "
136|           + " WHERE `product-name` = '"+productBean.getName()+"'");
137|       int pid = 0;
138|       while (executeQueryFetchPid.next()){
139|           pid = executeQueryFetchPid.getInt("product_id");
140|       }
141|
142|       st.executeUpdate(" INSERT INTO `database_ecarla`.`expenses` ("
143|           + "`expenses_id` , "
144|           + "`product_id` , "
145|           + "`product_name` , "
146|           + "`price` , "
147|           + "`purchase_date` "
148|           + " "
149|           + "VALUES ("
150|           + "NULL , ' "+pid+"', '"+productBean.getName()+"', '"+(quantity*price)+"', NOW( )"
151|           + " );");
152|       response.sendRedirect("productInsertImages.jsp");
153|   }catch(NumberFormatException e) {
154|       response.sendRedirect("admin_addProduct_4.jsp");
155|   }catch(SQLIntegrityConstraintViolationException e) {
156|       response.sendRedirect("admin_addProduct_3.jsp");
157|   }finally{
158|
159|   }
160|
161|   %>
162| </body>
</html>

```

## ❖ admin\_addProduct.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>eCarla Car</title>
        <jsp:useBean class="product.product" id="product" scope="session"></jsp:useBean>
    
```

The code block shows the source code for the JSP file admin\_addProduct.jsp. It includes JSP tags like <%@page>, <!DOCTYPE html>, <html>, <head>, and <jsp:useBean>. It also includes CSS styles defined within the head section, such as .highlight and .blueButton classes, and JavaScript imports like jquery-1.7.2.min.js, lightbox.js, and myScript.js.

```

    <%@page import="java.sql.* , database.*" %>
    <link rel="shortcut icon" href="images/logo/ico.ico"/>
    <link rel="stylesheet" type="text/css" href="css/reset.css"/>
    <link rel="stylesheet" type="text/css" href="css/text.css"/>
    <link rel="stylesheet" type="text/css" href="css/960_16.css"/>
    <link rel="stylesheet" type="text/css" href="css/product.css" />
    <link rel="stylesheet" type="text/css" href="css/lightbox.css" />
    <link rel="stylesheet" type="text/css" href="css/styles.css"/>
    <script src="js/jquery-1.7.2.min.js"></script>
    <script src="js/lightbox.js"></script>
    <script src="js/myScript.js"></script>

    <style type="text/css">
        .highlight {
            box-shadow: inset -10px 0px 10px #666;
            background: #E0E0E0;
            border-radius: 0px 27px 27px 0px;
        }

        .blueButton {
            background: -webkit-linear-gradient(top, #00AAC9, #026A84);
        }

        #buy {
            padding: 30px;
            font-size: 17px;
            -webkit-transition:.3s all ease-in-out;
            -moz-transition:.3s all ease-in-out;
        }
        #buy:hover {
            padding: 30px;
            font-size: 17px;
            box-shadow: inset 0px 2px 17px #444;
        }
        /* 
        #buy {
```

```
49 |         #buy {
50 |             padding: 30px;
51 |             font-size: 17px;
52 |             box-shadow: 0px 2px 7px #444;
53 |             -webkit-transition:.3s all ease-in-out;
54 |             -moz-transition:.3s all ease-in-out;
55 |             background: -webkit-linear-gradient(top,#00AAC9, #026A84);
56 |             background: -moz-linear-gradient(top,#00AAC9, #026A84);
57 |
58 |     #buy:active {
59 |         padding: 30px;
60 |         font-size: 17px;
61 |         box-shadow: 0px 2px 7px #444;
62 |         -webkit-box-shadow: inset 0 0 10px #000;
63 |         -moz-box-shadow: inset 0 0 5px #000;
64 |         background: -webkit-linear-gradient(top, #026A84,#00AAC9);
65 |         background: -moz-linear-gradient(top, #026A84,#00AAC9);
66 |
67 |     #buy:hover {
68 |         padding: 30px;
69 |         font-size: 17px;
70 |         box-shadow: inset 0px 2px 17px #444;
71 |         background: -webkit-linear-gradient(top, #026A84,#00AAC9);
72 |         background: -moz-linear-gradient(top, #026A84,#00AAC9);
73 |
74 |     } */
75 |     .adminMenu {
76 |         margin-top: 10px;
77 |         margin-bottom: 10px;
78 |         margin-right: 0px;
79 |         background: #FFF;
80 |         box-shadow: 0px 0px 10px #333;
81 |     }
82 |     .adminMain {
83 |         margin-top: 10px;
84 |         margin-bottom: 10px;
85 |         background: #FFF;
86 |         box-shadow: 0px -1px 10px #333;
87 |     }
88 |     #leftside {
89 |         cursor: pointer;
90 |     }
91 |     </style>
92 | </head>
93 | <body>
94 |
95 |     <%
96 |     if (session.getAttribute("user") == null ){// THen new user, show join now
97 |     %>
```

```
97      %>
98      <jsp:include page="includesPage/_joinNow.jsp"></jsp:include>
99      <%
100     }else {
101         %>
102         <jsp:include page="includesPage/_logout.jsp"></jsp:include>
103         <%
104     }
105
106     if (session.getAttribute("admin") == null) {
107         response.sendRedirect("admin_.jsp");
108     }
109     %>
110
111     <jsp:include page="includesPage/_search_navigationbar.jsp"></jsp:include>
112     <jsp:include page="includesPage/_facebookJoin.jsp"></jsp:include>
113
114     <div class="container_16">
115         <div class="grid_16" style="padding: 10px;" id="whiteBox">
116             <div class="grid_3">
117                 <br/><h1>Add Product</h1>
118             </div>
119             <div>
120
121                 <div class="grid_3">
122                     <br/><h1>Step 1. Enter Car Company</h1><hr/>
123                 </div>
124                 <div class="grid_3">
125                     <br/><h1>Step 2. Enter Category</h1><hr/>
126                 </div>
127                 <div class="grid_3">
128                     <br/><h1>Step 3. Enter in Car Name, Tags</h1><hr/>
129                 </div>
130                 <div class="grid_3">
131                     <br/><h1>Step 4. Enter in Pricing & Quantity and Description</h1><hr/>
132                 </div>
133
134             </div>
135         </div>
136         <div class="clear"></div>
137         <jsp:include page="includesPage/hesders_sidebars/admin_menuSideBar.jsp"></j:
138
139
140
141         <div class="grid_13" style="padding: 10px 0px;" id="whiteBox">
142             <div class="grid_13">
143                 <br/><h1>4 Step Process for Adding Cars...</h1><hr/>
144             </div>
145
```

## ❖ admin\_Performance.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>eCarla Car</title>
        <jsp:useBean class="product.product" id="product" scope="session"></jsp:useBean>

        <%@page import="java.sql.* , database.*" %>
        <link rel="shortcut icon" href="images/logo/ico.ico"/>
        <link rel="stylesheet" type="text/css" href="css/reset.css"/>
        <link rel="stylesheet" type="text/css" href="css/text.css"/>
        <link rel="stylesheet" type="text/css" href="css/960_16.css"/>
        <link rel="stylesheet" type="text/css" href="css/product.css" />

        <link rel="stylesheet" type="text/css" href="css/lightbox.css" />

        <link rel="stylesheet" type="text/css" href="css/styles.css"/>

        <script src="js/jquery-1.7.2.min.js"></script>
        <script src="js/lightbox.js"></script>
        <script src="js/myScript.js"></script>

        <style type="text/css">
            .adminMenu {
                margin-top: 10px;
                margin-bottom: 10px;
                margin-right: 0px;
                background: #FFF;
                box-shadow: 0px 0px 10px #333;
            }
            .adminMain {
                margin-top: 10px;
                margin-bottom: 10px;
                background: #FFF;
                box-shadow: 0px -1px 10px #333;
            }
            #leftside {
                cursor: pointer;
            }
        </style>
    </head>
    <body>

        <%
        if (session.getAttribute("user") == null ){// THen new user, show join now
        %>
```

```
49 |         %>
50 |         <jsp:include page="includesPage/_joinNow.jsp"></jsp:include>
51 |     <%
52 |     }else {
53 |         %>
54 |         <jsp:include page="includesPage/_logout.jsp"></jsp:include>
55 |     <%
56 |     }
57 |
58 |     if (session.getAttribute("admin")== null){
59 |         response.sendRedirect("admin_.jsp");
60 |     }
61 |     %>
62 |     <jsp:include page="includesPage/_search_navigationbar.jsp"></jsp:include>
63 |     <jsp:include page="includesPage/_facebookJoin.jsp"></jsp:include>
64 |
65 |     <div class="container_16">
66 |
67 |         <div class="grid_16" style="padding: 10px;" id="whiteBox">
68 |             <br/>
69 |
70 |             <%
71 |
72 |                 String sqlPending = "SELECT COUNT( `status` ) as pending"
73 | + " FROM `order` "
74 | + " WHERE `status` = 'pending'";
75 |
76 |                 String sqlApproved = "SELECT COUNT( `status` ) as approved "
77 | + " FROM `order` "
78 | + " WHERE `status` = 'approved' ";
79 |
80 |                 String sqlQuantity = "SELECT COUNT( `product-name` ) as qty"
81 | +" FROM `products` "
82 | +" WHERE `product_qty` <5 ";
83 |
84 |                 sqlQuantity = "SELECT COUNT( `product-name` ) AS qty"
85 | +" FROM `products` "
86 | +" WHERE `product_qty` <5";
87 |
88 |                 Connection c = new DB_Conn().getConnection();
89 |                 Statement st = c.createStatement();
90 |
91 |                 ResultSet rs = null;
92 |                 String qty = "0",
93 |                         pendingOrders= "0",
94 |                         approvedOrders= "0";
95 |
96 |                 rs = st.executeQuery(sqlQuantity);
97 |                 while (rs.next()) {
```

```

97 |             while (rs.next()){
98 |                 qty = rs.getString("qty");
99 |
100|             if (qty.equals("0")){
101|                 qty = "No";
102|
103|
104|             rs = st.executeQuery(sqlPending);
105|             while (rs.next()){
106|                 pendingOrders = rs.getString("pending");
107|             }
108|             if (pendingOrders.equals("0")){
109|                 pendingOrders = "No";
110|             }
111|
112|             rs = st.executeQuery(sqlApproved);
113|             while (rs.next()){
114|                 approvedOrders = rs.getString("approved");
115|             }
116|             if (approvedOrders.equals("0")){
117|                 approvedOrders = "No";
118|             }
119|         %>
120|         <h1 class="grid_15">Dashboard Home</h1><hr/>
121|         <a href="admin_manageProduct.jsp" id="buy" class="grid_3"><%= qty %> Items below Qua
122|         <a href="admin_approvedOrders.jsp" id="buy" class="grid_3"><%= approvedOrders %> It
123|         <a href="admin_pendingOrders.jsp" id="buy" class="grid_3"><%= pendingOrders %> Items
124|     </div>
125|     <div class="clear"></div>
126|     <!--Loading the AJAX API-->
127|     <script type="text/javascript" src="js/gclibrary/jsapi.js"></script>
128|     <script type="text/javascript" src="js/gclibrary/core.js"></script>
129|     <script type="text/javascript" src="js/gclibrary/corel.js"></script>
130|
131|     <jsp:include page="includesPage/gChartings/gChart_ComboLine.jsp"></jsp:include>
132|
133|     <jsp:include page="includesPage/hesders_sidebars/admin_menuSideBar.jsp"></jsp:include>
134|     <div class="grid_13">
135|         <jsp:include page="includesPage/gChartings/gCharts_pieChartItemsSoldByCategory.jsp">
136|         <jsp:include page="includesPage/gChartings/gCharts_pieChartItemsViewedByCategory.jsp"
137|         </div>
138|         <jsp:include page="includesPage/gChartings/gCharts_top10ProductsSold.jsp"></jsp:inclu
139|         <jsp:include page="includesPage/gChartings/gCharts_top10Products_viewed.jsp"></jsp:in
140|
141|     </div>
142|
143|     </body>
144| </html>
145|

```

## ❖ admin.jsp

```
admin.jsp ×  
Source History |         |     |   
1   <%@page contentType="text/html" pageEncoding="UTF-8"%>  
2   <!DOCTYPE html>  
3   <html>  
4     <head>  
5       <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
6       <title>eCarla Car</title>  
7       <jsp:useBean class="product.product" id="product" scope="session"></jsp:useBean>  
8  
9       <%@page import="java.sql.* , database.*" %>  
10      <link rel="shortcut icon" href="images/logo/ico.ico"/>  
11      <link rel="stylesheet" type="text/css" href="css/reset.css"/>  
12      <link rel="stylesheet" type="text/css" href="css/text.css"/>  
13      <link rel="stylesheet" type="text/css" href="css/960_16.css"/>  
14      <link rel="stylesheet" type="text/css" href="css/product.css" />  
15  
16      <link rel="stylesheet" type="text/css" href="css/lightbox.css" />  
17  
18      <link rel="stylesheet" type="text/css" href="css/styles.css"/>  
19  
20  
21      <script src="js/jquery-1.7.2.min.js"></script>  
22      <script src="js/lightbox.js"></script>  
23      <script src="js/myScript.js"></script>  
24    </head>  
25  <body>  
26  
27  
28  <%  
29    if (session.getAttribute("user") == null ){// THen new user, show join now  
30      %>  
31      <jsp:include page="includesPage/_joinNow.jsp"></jsp:include>  
32  <%  
33    } else {  
34      %>  
35      <jsp:include page="includesPage/_logout.jsp"></jsp:include>  
36  <%  
37    }  
38  %>  
39  
40  <jsp:include page="includesPage/_search_navigationbar.jsp"></jsp:include>  
41  <jsp:include page="includesPage/_facebookJoin.jsp"></jsp:include>  
42  
43  <div class="container_16">  
44    <div class="grid_13 push_2" id="whiteBox" style="padding:10px 0px 10px 0px;">  
45      <h1 class="push_4" style="padding: 10px;">Administrator Login</h1>  
46      <hr/> <br/><div class="grid_9 push_2" style="padding:10px;">  
47        <form method="post" action="admin_login">  
48          <div class="grid_2">  
49            Email
```

```
39
40    <jsp:include page="includesPage/_search_navigationbar.jsp"></jsp:include>
41    <jsp:include page="includesPage/_facebookJoin.jsp"></jsp:include>
42
43    <div class="container_16">
44        <div class="grid_13 push_2" id="whiteBox" style="padding:10px 0px 10px 0px;">
45            <h1 class="push_4" style="padding: 10px;">Administrator Login</h1>
46            <hr/> <br/><div class="grid_9 push_2" style="padding:10px;">
47                <form method="post" action="admin_login">
48                    <div class="grid_2">
49                        Email
50                    </div>
51                    <div class="grid_5">
52                        <input type="text" name="email" placeholder="xyz@xyz.com" />
53                    </div>
54                    <div class="clear"></div><br/>
55
56                    <div class="grid_2">
57                        Password
58                    </div>
59                    <div class="grid_5">
60                        <input type="password" name="pass" placeholder="••••"/>
61                        <input id="greenBtn" type="submit" value="Login"/>
62                    </div>
63                    <div class="clear"></div><br/>
64                </form>
65            </div>
66        </div>
67    </body>
68
69    </html>
70
```

## ❖ web.xml

The screenshot shows the 'web.xml' configuration file in a Java IDE. The interface includes a toolbar with icons for Save, Undo, Redo, Cut, Copy, Paste, Find, and Delete. Below the toolbar is a tab bar with 'Source' (selected), 'General', 'Servlets', 'Filters', 'Pages', 'References', 'Security', and 'History'. The main area displays the XML code for the web application's deployment descriptor.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
    <servlet>
        <servlet-name>loginServlet</servlet-name>
        <servlet-class>user.loginServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>registerServlet</servlet-name>
        <servlet-class>user.registerServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>logoutServlet</servlet-name>
        <servlet-class>user.logoutServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>addProduct</servlet-name>
        <servlet-class>product.addProduct</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>addProductName</servlet-name>
        <servlet-class>product.addCompanyName</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>addCompany</servlet-name>
        <servlet-class>product.addCompany</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>addCategory</servlet-name>
        <servlet-class>product.addCategory</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>addSubCategory</servlet-name>
        <servlet-class>product.addSubCategory</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>getSubCategory</servlet-name>
        <servlet-class>product.getSubCategory</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>_logoutServlet</servlet-name>
        <servlet-class>loginRegister._logoutServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>viewProductFilterServlet</servlet-name>
        <servlet-class>product.viewProductFilterServlet</servlet-class>
    </servlet>
</servlets>
```

```
47 |     <servlet>
48 |         <servlet-name>changeMyPass</servlet-name>
49 |         <servlet-class>loginRegister.changeMyPass</servlet-class>
50 |     </servlet>
51 |     <servlet>
52 |         <servlet-name>getProductSubCategory</servlet-name>
53 |         <servlet-class>product.getProductSubCategory</servlet-class>
54 |     </servlet>
55 |     <servlet>
56 |         <servlet-name>changeProductStatus</servlet-name>
57 |         <servlet-class>orders.changeProductStatus</servlet-class>
58 |     </servlet>
59 |     <servlet>
60 |         <servlet-name>admin_login</servlet-name>
61 |         <servlet-class>loginRegister.admin_login</servlet-class>
62 |     </servlet>
63 |     <servlet>
64 |         <servlet-name>addProductServlet</servlet-name>
65 |         <servlet-class>product.addProductServlet</servlet-class>
66 |     </servlet>
67 |     <servlet>
68 |         <servlet-name>addAnAdministrator</servlet-name>
69 |         <servlet-class>admin.addAnAdministrator</servlet-class>
70 |     </servlet>
71 |     <servlet>
72 |         <servlet-name>removeCompany</servlet-name>
73 |         <servlet-class>product.removeCompany</servlet-class>
74 |     </servlet>
75 |     <servlet>
76 |         <servlet-name>removeSubCategory</servlet-name>
77 |         <servlet-class>product.removeSubCategory</servlet-class>
78 |     </servlet>
79 |     <servlet>
80 |         <servlet-name>removeCategory</servlet-name>
81 |         <servlet-class>product.removeCategory</servlet-class>
82 |     </servlet>
83 |     <servlet-mapping>
84 |         <servlet-name>loginServlet</servlet-name>
85 |         <url-pattern>/login</url-pattern>
86 |     </servlet-mapping>
87 |     <servlet-mapping>
88 |         <servlet-name>registerServlet</servlet-name>
89 |         <url-pattern>/register</url-pattern>
90 |     </servlet-mapping>
91 |     <servlet-mapping>
92 |         <servlet-name>logoutServlet</servlet-name>
```

```
97 |         <url-pattern>/addProduct</url-pattern>
98 |     </servlet-mapping>
99 |     <servlet-mapping>
100 |         <servlet-name>addProductName</servlet-name>
101 |         <url-pattern>/addProductName</url-pattern>
102 |     </servlet-mapping>
103 |     <servlet-mapping>
104 |         <servlet-name>addCompany</servlet-name>
105 |         <url-pattern>/addCompany</url-pattern>
106 |     </servlet-mapping>
107 |     <servlet-mapping>
108 |         <servlet-name>addCategory</servlet-name>
109 |         <url-pattern>/addCategory</url-pattern>
110 |     </servlet-mapping>
111 |     <servlet-mapping>
112 |         <servlet-name>addSubCategory</servlet-name>
113 |         <url-pattern>/addSubCategory</url-pattern>
114 |     </servlet-mapping>
115 |     <servlet-mapping>
116 |         <servlet-name>getSubCategory</servlet-name>
117 |         <url-pattern>/getSubCategory</url-pattern>
118 |     </servlet-mapping>
119 |     <servlet>
120 |         <servlet-name>addToCart</servlet-name>
121 |         <servlet-class>controller.addToCart</servlet-class>
122 |     </servlet>
123 |     <servlet>
124 |         <servlet-name>loginServlet</servlet-name>
125 |         <servlet-class>loginRegister.loginServlet</servlet-class>
126 |     </servlet>
127 |     <servlet>
128 |         <servlet-name>registerServlet</servlet-name>
129 |         <servlet-class>loginRegister.registerServlet</servlet-class>
130 |     </servlet>
131 |     <servlet>
132 |         <servlet-name>changeMyPassword</servlet-name>
133 |         <servlet-class>loginRegister.changeMyPassword</servlet-class>
134 |     </servlet>
135 |     <servlet>
136 |         <servlet-name>removeCartProduct</servlet-name>
137 |         <servlet-class>product.removeCartProduct</servlet-class>
138 |     </servlet>
139 |     <servlet>
140 |         <servlet-name>getCompanyNames</servlet-name>
141 |         <servlet-class>product.addProductPack.getCompanyNames</servlet-class>
142 |     </servlet>
143 | </servlet-mapping>
```

web.xml

Source General Servlets Filters Pages References Security History

```
142     </servlet>
143     <servlet-mapping>
144         <servlet-name>addToCart</servlet-name>
145         <url-pattern>/addToCart</url-pattern>
146     </servlet-mapping>
147     <servlet-mapping>
148         <servlet-name>loginServlet</servlet-name>
149         <url-pattern>/loginServlet</url-pattern>
150     </servlet-mapping>
151     <servlet-mapping>
152         <servlet-name>registerServlet</servlet-name>
153         <url-pattern>/registerServlet</url-pattern>
154     </servlet-mapping>
155     <servlet-mapping>
156         <servlet-name>changeMyPassword</servlet-name>
157         <url-pattern>/changeMyPassword</url-pattern>
158     </servlet-mapping>
159     <servlet-mapping>
160         <servlet-name>removeCartProduct</servlet-name>
161         <url-pattern>/removeCartProduct</url-pattern>
162     </servlet-mapping>
163     <servlet-mapping>
164         <servlet-name>getCompanyNames</servlet-name>
165         <url-pattern>/getCompanyNames</url-pattern>
166     </servlet-mapping>
167     <servlet-mapping>
168         <servlet-name>_logoutServlet</servlet-name>
169         <url-pattern>/_logoutServlet</url-pattern>
170     </servlet-mapping>
171     <servlet-mapping>
172         <servlet-name>viewProductFilterServlet</servlet-name>
173         <url-pattern>/viewProductFilterServlet</url-pattern>
174     </servlet-mapping>
175     <servlet-mapping>
176         <servlet-name>changeMyPass</servlet-name>
177         <url-pattern>/changeMyPass</url-pattern>
178     </servlet-mapping>
179     <servlet-mapping>
180         <servlet-name>getProductSubCategory</servlet-name>
181         <url-pattern>/getProductSubCategory</url-pattern>
182     </servlet-mapping>
183     <servlet-mapping>
184         <servlet-name>changeProductStatus</servlet-name>
185         <url-pattern>/changeProductStatus</url-pattern>
186     </servlet-mapping>
187     <servlet-mapping>
188         <servlet-name>admin_login</servlet-name>
```

```
187 <servlet-mapping>
188     <servlet-name>admin_login</servlet-name>
189     <url-pattern>/admin_login</url-pattern>
190 </servlet-mapping>
191 <servlet-mapping>
192     <servlet-name>addProductServlet</servlet-name>
193     <url-pattern>/addProductServlet</url-pattern>
194 </servlet-mapping>
195 <servlet-mapping>
196     <servlet-name>addAnAdministrator</servlet-name>
197     <url-pattern>/addAnAdministrator</url-pattern>
198 </servlet-mapping>
199 <servlet-mapping>
200     <servlet-name>removeCompany</servlet-name>
201     <url-pattern>/removeCompany</url-pattern>
202 </servlet-mapping>
203 <servlet-mapping>
204     <servlet-name>removeSubCategory</servlet-name>
205     <url-pattern>/removeSubCategory</url-pattern>
206 </servlet-mapping>
207 <servlet-mapping>
208     <servlet-name>removeCategory</servlet-name>
209     <url-pattern>/removeCategory</url-pattern>
210 </servlet-mapping>
211 <session-config>
212     <session-timeout>
213         30
214     </session-timeout>
215 </session-config>
216 <error-page>
217     <error-code>404</error-code>
218     <location>/error_404.jsp</location>
219 </error-page>
220 <error-page>
221     <exception-type>java.lang.Throwable</exception-type>
222     <location>/error_500.jsp</location>
223 </error-page>
224 <error-page>
225     <error-code>403</error-code>
226     <location>/error_403.jsp</location>
227 </error-page>
228 </web-app>
229
```