

NovaNest



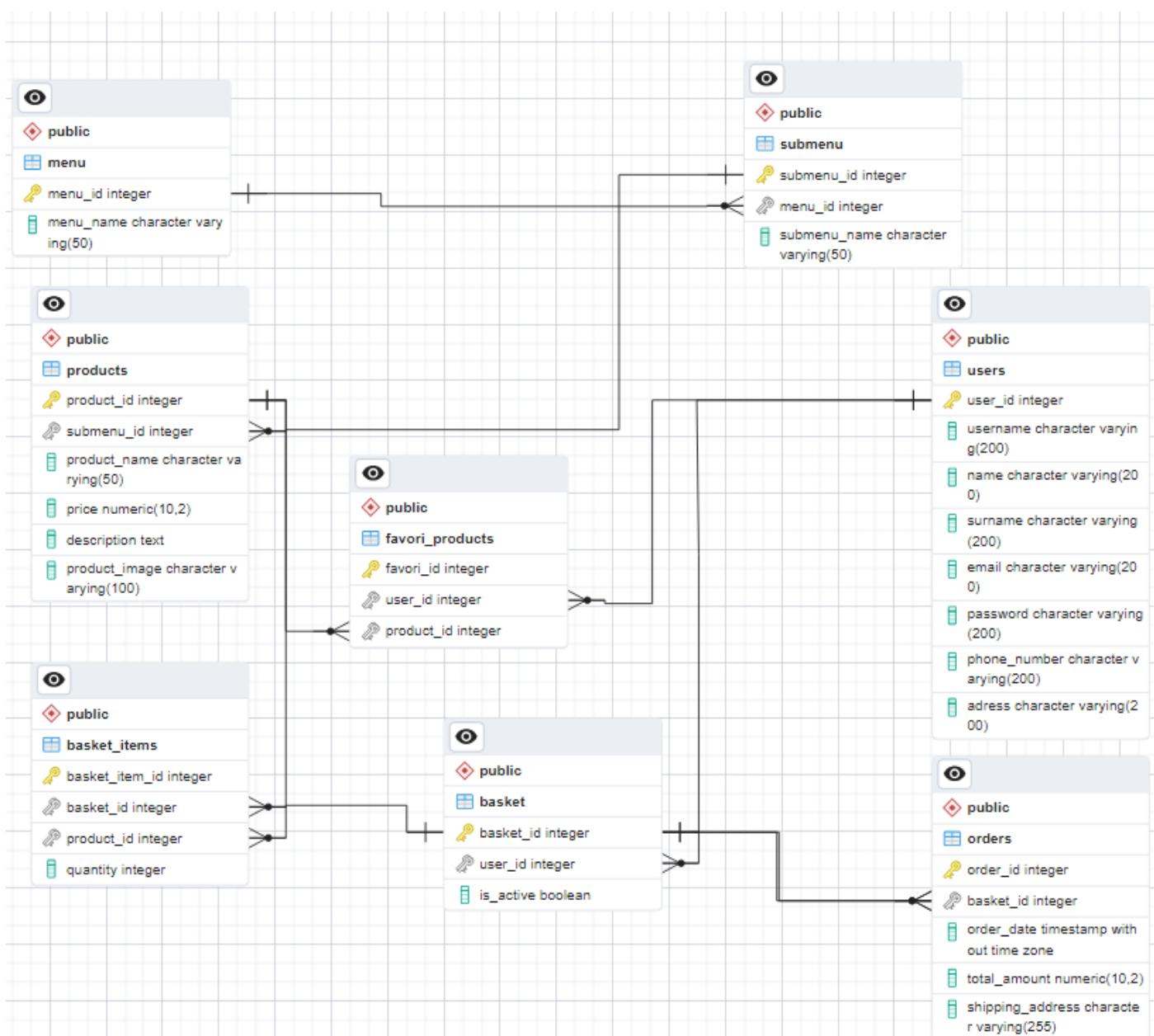
Project Name: NovaNest Online Alışveriş Website

Designed By : Bilal AYAKDAŞ

Project Description:

My application scenario is an e-commerce site. My application consists of JSP pages. In my application, I made connections to the database and database operations using Java. While writing this application, I also used **HTML, CSS, Bootstrap, JavaScript, JQuery**, and **JSP** in the web part. I used **PostgreSQL** in the database part. There are also functions I wrote in the database for some complex operations. Below are the database ERD diagram, database tables, web screenshots, and screenshots of other codes for the project.

ERD -Diagram



E-Commerce Database Tables

menu Table

	menu_id [PK] integer	menu_name character varying (50)
1	1	Elektronik
2	2	Moda
3	3	Ev
4	4	Yaşam
5	5	Kırtasiye
6	6	Ofis
7	7	Oto
8	8	Bahçe
9	9	Yapı Market
10	10	Oyuncak
11	11	Spor
12	12	Outdoor
13	13	Kozmetik
14	14	Kişisel Bakım
15	15	Süpermarket
16	16	Kitap
17	17	Müzik
18	18	Film
19	19	Hobi

submenu Table

	submenu_id [PK] integer	menu_id integer	submenu_name character varying (50)
1	1	1	CepTelefonları
2	2	1	Bilgisayarlar
3	3	1	Televizyon ve Ses Sistemleri
4	4	1	Ev Aletleri
5	5	2	Erkek Giyim
6	6	2	Kadın Giyim
7	7	2	Ayakkabılar
8	8	2	Aksesuarlar
9	9	3	Mobilya
10	10	3	Ev Dekorasyonu
11	11	3	Mutfak Gereçleri
12	12	3	Ev Tekstili
13	13	4	SağlıkveFitness
14	14	4	Seyahat ve Outdoor Malzemeler
15	15	4	Evcil Hayvan Malzemeleri
16	16	4	Parti ve Eğlence
17	17	5	Defter ve Kağıtlar
18	18	5	Kalemve Yazı Gereçleri
19	19	5	Ajanda ve Takvimler
20	20	5	Ofis Malzemeleri
21	21	6	Yapıştırıcılar ve Bantlar
22	22	6	Dosyalama ve Organizasyon

products Table

	product_id [PK] integer	submenu_id integer	product_name character varying (50)	price numeric (10,2)	description text	product_image character varying (100)
1	1	1	IPHONE 12	50000.99	Ürün 1 Açıklaması	images/Images.jpeg
2	2	1	XIOMI 11 LITE	20000.99	Ürün 2 Açıklaması	images/xiomi11lite.jpeg
3	3	1	XIOMI NOTE 12 PRO	12000.99	Ürün 3 Açıklaması	images/xiominote12.jpeg
4	4	1	XIOMI 11 ULTRA	30000.99	Ürün 1 Açıklaması	images/xiaomi-mi-11-ultra.jpg
5	5	2	LENOVA IDEAPAD 3	25000.99	Ürün 2 Açıklaması	images/ideaped.jpeg
6	6	2	MSI PULSE 64GB	90000.99	Ürün 3 Açıklaması	images/msi-pulse.jpg
7	7	2	APPLE MACBOOK PRO	120000.99	Ürün 1 Açıklaması	images/macbookpro.jpeg
8	8	5	DERIMOD DERIMONT	8000.99	Ürün 2 Açıklaması	images/derimont.jpeg
9	9	5	H&M KARGO PANTOLON	900.99	Ürün 3 Açıklaması	images/kargo_pant.jpg
10	10	5	SWEET DÜZ	700.50	ALTINYILDIZ	images/sweet_duz.jpg

users Table

user_id	username	name	surname	email	password	phone_number	address
[PK] integer	character varying (200)	character varying (200)	character varying (200)	character varying (200)	character varying (200)	character varying (200)	character varying (200)
1	isabel11	Sara	Brown	jrlley@example.org	1rzJKlI5%!	558-808-6561	42102 Ramirez Trail Apt. 849, Williamsside, FM 91312
2	gary47	Katherine	Gomez	asheppard@example.net)n_SRhBl"*2	885-560-1548	58345 Kevin Place Apt. 846, Craigton, SC 82084
3	hblackburn	Christopher	White	caitlincole@example.org	2l\$0UQ6nBj	+1-213-798-1420x705	4030 Christopher Mount, East Rodney, DE 26892
4	ayalajennifer	Corey	Davis	vhunt@example.com	(4SSQ#Vn9P	970.269.4829x25476	37084 Chloe Fields Suite 321, New Kimberlymouth, WY 503
5	spatterson	Jonathon	Clark	torresclaudia@example.net	5bq4X7dr_m	(806)710-8937x56468	76502 Thompson Port, New Denismouth, RI 84237
6	masondale	Ashlee	Garcia	abigailblanchard@example.com	ATS)4C_tW(535-660-8308x37011	138 Jacob Meadows Apt. 874, East Roy, MO 74343
7	christophernorris	Denise	Curry	michael05@example.com	!9RUuVPmnK	001-654-487-3993x3652	29132 Cody Extension, Jasonfort, ME 59176
8	palmerjoe	Daniel	Baker	xcombs@example.net	#PP5jb8h8&	453.520.9975x71172	32244 Torres Ports, Lake Seth, TX 31319
9	danafrazier	Caleb	Nicholson	kathleen14@example.com	09*W4vR#*!	+1-865-826-2541x82125	021 Coffey Mall, Douglasmouth, HI 89687
10	briantorres	Mary	Crawford	carlcarroll@example.org	gm3+HrZGh&	785.668.3648x73821	9765 Wells Road, Mitchellmouth, NH 67872
11	bayakdas	Bilal	Ayakdas	bilalayakdas@gmail.com	123456	5554443322	Yunus Emre Erkek öğrenci yurdu
12	nnnnn	dddd	ddd	dddd	1234555	45353456	ffdfgff

basket Table:

	basket_id	user_id	is_active
	[PK] integer	integer	boolean
1	18	12	true
2	19	12	true
3	20	12	true
4	21	11	false
5	22	11	false
6	26	11	false
7	27	11	true

basket_items Table:

	basket_item_id	basket_id	product_id	quantity
	[PK] integer	integer	integer	integer
1	1	21	1	1
2	3	21	2	1
3	4	21	3	1
4	13	22	1	1
5	15	22	8	1
6	16	26	1	1
7	26	27	1	1

favori_products Table:

	favori_id	user_id	product_id
	[PK] integer	integer	integer
1	14	12	2
2	17	11	5
3	18	11	6

orders Table:

	order_id	basket_id	order_date	total_amount	shipping_address
	[PK] integer	integer	timestamp without time zone	numeric (10,2)	character varying (255)
1	1	21	2024-01-04 00:00:00	82002.97	Yunus Emre Erkek öğrenci yurdu
2	2	22	2024-01-04 20:46:31.957044	58001.98	Yunus Emre Erkek öğrenci yurdu
3	3	26	2024-01-06 13:24:30.731328	50000.99	Yunus Emre Erkek öğrenci yurdu

USED LIBRARIES

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<%@ page import="java.sql.Connection" %>
<%@ page import="javax.naming.Context" %>
<%@ page import="javax.naming.InitialContext" %>
<%@ page import="javax.sql.DataSource" %>
<%@ page import="java.sql.Connection" %>
<%@ page import="java.sql.DriverManager" %>
<%@ page import="java.sql.ResultSet" %>
<%@ page import="java.sql.SQLException" %>
<%@ page import=" java.sql.Statement" %>
<%@ page import="java.util.logging.Level" %>
<%@ page import=" java.util.logging.Logger" %>


```

```

<!-- Bootstrap CSS -->
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">
<!-- jQuery -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<!-- Bootstrap JS -->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

```

LOGIN PAGE



Email address bilalayakdas@gmail.com

Password

☒ Remember me

[Sign in](#)

Not a member? [Register](#)

LoginPage.jsp:

LoginPage.jsp contains text boxes where the user will enter their information to log in, and if they are not a member, they will be directed to the registration page to become a member by clicking on the Sign Up text. If they become a member, when they fill in their login information and press the Login button, the information in the text boxes is transferred to javascript variables with jquery and sent to the ajaxLogin.jsp page. And it can log in according to the value returned from the ajaxLogin.jsp page. And if the login is successful, it redirects to MainPage.jsp.

```

<div style="display: flex; flex-direction: column; ">
<div id="image" style=" margin-left: 500px; margin-top: 100px; width:350px; height:100px;" class="box"></div>
<div style="margin-top: 0px;" >
<table style="margin-left: 500px;">
<tr>
<div class="form-outline mb-4">
<th>Email address </th>
<td><input type="email" id="email" class="form-control"/></td>
</div>
</tr>
<tr>
<div class="form-outline mb-4">
<th>Password </th>
<td><input type="password" id="password" class="form-control"/></td>
</div>
</tr>
<tr>
<td></td>
<td>
<div class="form-check">
<input class="form-check-input" type="checkbox" value="" id="form2Example31" checked />
<label class="form-check-label" for="form2Example31"> Remember me </label>
</div>
</td>
</tr>
<tr>
<td></td>
<td>
<button type="button" onclick="loginFonks()" style="width:250px;" class="btn btn-primary mb-4">Sign in</button>
</td>
</tr>
<tr>
<td></td>
<td><div class="text-center"><p>Not a member? <a href="http://localhost:8080/ECommerceProject/signup.jsp">Register</a></p></div></td>
</tr>
</table>
</div>
</div>

```

```

function loginFonks(){
var email = $('#email').val();
var password = $('#password').val();
var dogrumu = "";
$.ajax({
url: "ajaxLogin.jsp",
type: "GET",
data: {
email: email,
password: password
},
success: function(responseMessage) {
dogrumu = responseMessage.trim();
if(dogrumu == "1"){
alert("Giriş Başarılı.");
window.location.href = "http://localhost:8080/ECommerceProject/mainPage.jsp";
}else if(dogrumu == "0"){
alert("Şifre yada email bilgisi hatalı.Tekrar deneyin");
}
},
error: function(xhr, status, error) {
console.error(error);
}
});
}

```

ajaxLogin.jsp Page:

First of all, on this page, the information taken as a parameter from the Login.jsp Page is transferred to variables, and if there is a user connected to the email variable entered with jdbc, their password and userId are retrieved. User verification is performed. If the user login is successful, the userId of the user is transferred to a variable and then the userId is transferred to the session object for use when necessary. And it returns 1 to the Login page. If the user login is not successful, it returns 0.

```
//Email ve Password Parametre Olarak Alma
String email = request.getParameter("email");
String password = request.getParameter("password");
String responseMessage="";
try {

String url = "jdbc:postgresql://localhost/ecommerce";
Connection connection;
ResultSet rs;
String sqlstr = "SELECT password,user_id FROM users WHERE email='"+email+"'";
Statement st;
Class.forName("org.postgresql.Driver");
connection = DriverManager.getConnection(url,"postgres","root");
st = connection.createStatement();
rs = st.executeQuery(sqlstr);

rs.next();
String dbPasss=rs.getString(1);

if(password.equals(dbPasss)){
    String userId=rs.getString(2);
    session.setAttribute("userId",userId);
    out.print("1");
}else{
    out.print("0");
}
} catch (Exception e) {
    responseMessage = "Error: " + e.getMessage();
}
```

SIGN UP PAGE:

Name:	<input type="text"/>
Surname:	<input type="text"/>
Address:	<input type="text"/>
Email:	<input type="text"/>
Phone Number:	<input type="text"/>
User name:	<input type="text"/>
Password	<input type="password"/>
	<input type="button" value="Sign Up"/>

SignUp.jsp:

There are textboxes where the user will enter their information to become a member. The user fills in the information required to become a member. When you press the Sign Up button after filling in the information, the information in the textboxes is transferred to javascript variables with jquery and sent to the ajaxSignUp.jsp page. And it redirects to LoginPage.jsp.


```

<table style="margin-left: 500px;margin-top: 100px">
  <tr>
    <div class="form-outline mb-4">
      <th>Name:</th>
      <td><input type="name" id="name" class="form-control"/></td>
    </div>
  </tr>
  <tr>
    <div class="form-outline mb-4">
      <th>Surname:</th>
      <td><input type="text" id="surname" class="form-control"/></td>
    </div>
  </tr>
  <tr>
    <div class="form-outline mb-4">
      <th>Address:</th>
      <td><textarea id="address" style="height: 100px; width: 250px;" class="form-control"/></td>
    </div>
  </tr>
  <tr>
    <div class="form-outline mb-4">
      <th>Email: </th>
      <td><input type="email" id="email" class="form-control"/></td>
    </div>
  </tr>
  <tr>
    <div class="form-outline mb-4">
      <th>Phone Number:</th>
      <td><input type="text" id="phoneNumber" class="form-control"/></td>
    </div>
  </tr>
  <tr>
    <div class="form-outline mb-4">
      <th>User name:</th>
      <td><input type="text" id="userName" class="form-control"/></td>
    </div>
  </tr>

```

```

});

```

```

function signUpFonks() {
  var name = $('#name').val();
  var surname = $('#surname').val();
  var address = $('#address').val();
  var email = $('#email').val();
  var phoneNumber = $('#phoneNumber').val();
  var userName = $('#userName').val();
  var password = $('#password').val();

  $.ajax({
    url: "ajaxSignUp.jsp",
    type: "GET",
    data: {
      name: name,
      surname: surname,
      address: address,
      email: email,
      phoneNumber: phoneNumber,
      userName: userName,
      password: password
    },
    success: function(responseMessage) {
      alert(responseMessage);
      window.location.href = "http://localhost:8080/ECommerceProject/Login.jsp";
    },
    error: function(xhr, status, error) {
      alert("Kayıt Edilemedi Hatalı.");
      console.error(error);
    }
  });
}

```

ajaxSignup.jsp:

The user information sent on the ajaxSignup.jsp page is taken as a parameter and thrown into the relevant variables. A prepared statement is used to add a user to the users table in the database, and then the prepared statement is executed.


```
<%  
  
String hata = "";  
String name = request.getParameter("name");  
String surname = request.getParameter("surname");  
String address = request.getParameter("address");  
String email = request.getParameter("email");  
String phoneNumber = request.getParameter("phoneNumber");  
String userName = request.getParameter("userName");  
String password = request.getParameter("password");  
  
try {  
String url = "jdbc:postgresql://localhost/ecommerce";  
Connection connection;  
String s;  
Class.forName("org.postgresql.Driver");  
connection = DriverManager.getConnection(url, "postgres", "root");  
String INSERT_USERS_SQL = "INSERT INTO users" +  
    " (username, name, surname, email, password, phone_number, address ) VALUES " +  
    " (?, ?, ?, ?, ?, ?, ?)";  
  
PreparedStatement preparedStatement = connection.prepareStatement(INSERT_USERS_SQL);  
preparedStatement.setString(1, userName);  
preparedStatement.setString(2, name);  
preparedStatement.setString(3, surname);  
preparedStatement.setString(4, email);  
preparedStatement.setString(5, password);  
preparedStatement.setString(6, phoneNumber);  
preparedStatement.setString(7, address);  
out.println(preparedStatement);  
preparedStatement.executeUpdate();  
  
} catch (Exception e) {  
    hata = "Error: " + e.getMessage();  
    out.print(hata);  
}
```

MAIN PAGE:

NovaNest
Online Alışveriş


MENU
Elektronik
Moda
Ev
Yaşam
Kırtasiye
Ofis
Oto
Bahçe
Yapı Market
Oyuncak
Spor
Outdoor
Kozmetik

HesabımFavorilerimSepetimOrders




IPHONE 12
50000.99 TL

Sepete Ekle




XIOMI 11 LITE
20000.99 TL

Sepete Ekle




XIOMI NOTE 12 PRO
12000.99 TL

Sepete Ekle




XIOMI 11 ULTRA
30000.99 TL

Sepete Ekle




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle




MSI PULSE 64GB
90000.99 TL

Sepete Ekle




APPLE MACBOOK PRO
120000.99 TL

Sepete Ekle




DERIMOD DERIMONT
8000.99 TL

Sepete Ekle



H&M KARGO PANTOLON
900.99 TL

Sepete Ekle




SWEET DÜZ
700.50 TL

Sepete Ekle

NovaNest
Online Alışveriş


MENU
Elektronik
Moda
Ev
Yaşam
Kırtasiye
Ofis
Oto
Bahçe
Yapı Market
Oyuncak
Spor
Outdoor

HesabımFavorilerimSepetimOrders




XIOMI 11 LITE
20000.99 TL

Sepete Ekle



XIOMI NOTE 12 PRO
12000.99 TL

Sepete Ekle




XIOMI 11 ULTRA
30000.99 TL

Sepete Ekle


Search

Hesabım Favorilerim Sepetim Orders




IPHONE 12
50000.99 TL

Sepete Ekle




XIOMI 11 LITE
20000.99 TL

Sepete Ekle




XIOMI NOTE 12 PRO
12000.99 TL

Sepete Ekle




XIOMI 11 ULTRA
30000.99 TL

Sepete Ekle




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle




MSI PULSE 64GB
90000.99 TL

Sepete Ekle




APPLE MACBOOK PRO
120000.99 TL

Sepete Ekle




DERİMOD DERİMONT
8000.99 TL

Sepete Ekle



H&M KARGO PANTOLON
900.99 TL

Sepete Ekle




SWEET DÜZ
700.50 TL

Sepete Ekle


Search

Hesabım Favorilerim Sepetim Orders




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle



MSI PULSE 64GB
90000.99 TL

Sepete Ekle



APPLE MACBOOK PRO
120000.99 TL


Sepete Ekle

Search

Hesabım Favorilerim Sepetim Orders


localhost:8080 şunu diyor:
Ürün Sepete eklendi.

Tamam




IPHONE 12
50000.99 TL

Sepete Ekle




XIOMI 11 LITE
20000.99 TL

Sepete Ekle




XIOMI NOTE 12 PRO
12000.99 TL

Sepete Ekle




XIOMI 11 ULTRA
30000.99 TL

Sepete Ekle




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle




MSI PULSE 64GB
90000.99 TL

Sepete Ekle




APPLE MACBOOK PRO
120000.99 TL

Sepete Ekle




DERİMOD DERİMONT
8000.99 TL

Sepete Ekle



H&M KARGO PANTOLON
900.99 TL

Sepete Ekle



SWEET DÜZ
700.50 TL

Sepete Ekle

E -Commerce Main



localhost:8080/ECommerceProject/basket.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimOrders

Sepetim

	PRODUCT IMAGE	PRODUCT NAME	PRICE	AMOUNT	
1		IPHONE 12	50000.99 TL	1	

Toplam Tutar : 50000,99 TL

Ödeme Yap

E -Commerce Main

localhost:8080/ECommerceProject/mainPage.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimOrders

Search

localhost:8080 şunu diyor:
Ürün favorilere eklendi.

Tamam

Elektronik

Moda

Ev

Yaşam

Kırtasiye

Ofis

Oto

Bahçe


Yapı Market

Oyuncak

Spor


Outdoor

Kozmetik




IPHONE 12
50000.99 TL

Sepete Ekle




XIOMI 11 LITE
20000.99 TL

Sepete Ekle




XIOMI NOTE 12 PRO
12000.99 TL

Sepete Ekle




XIOMI 11 ULTRA
30000.99 TL

Sepete Ekle




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle




MSI PULSE 64GB
90000.99 TL

Sepete Ekle




APPLE MACBOOK PRO
120000.99 TL

Sepete Ekle




DERIMOD DERİMONT
8000.99 TL

Sepete Ekle



H&M KARGO
PANTOLON
900.99 TL

Sepete Ekle



SWEET DÜZ
700.50 TL

Sepete Ekle

E -Commerce Main


localhost:8080/ECommerceProject/mainPage.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimOrders

FAVORITE PRODUCTS



XIOMI 11 LITE
20000.99 TL

Sepete Ekle

MainPage.jsp:

When the MainPage.jsp page loads, the getMenu and getProducts functions run.

getMenu function takes the Menu names from the ajaxGetMenu.jsp page and displays them on the page.

The getProducts function sends the value in the search box to the ajaxGetProductsSearch.jsp page and displays the products from the ajaxGetProductsSearch.jsp page on the screen.

When one of the menus is clicked, the subMenu function is called and gets the ID of the clicked menu, and shows the submenus connected to that menu from the ajaxGetSubMenu.jsp page.

When one of the submenus is clicked, the getProductsSubMenu function is called and takes the id of the clicked submenu and shows the products connected to that submenu from the ajaxGetSubMenuProducts.jsp page.

When you click on the heart under one of the products, the favProducts function is called and the clicked product ID is sent as ajaxFavProducts.jsp page parameter.

When the sepete ekle button under one of the products is clicked, the addBasket function is called and takes the clicked product ID and sends it as a parameter to the ajaxAddBasket.jsp page.

```
<div style ="display: flex; flex-direction: row; padding-left: 0px;">
<div onclick="location.href='http://localhost:8080/ECommerceProject/mainPage.jsp';" id="image" style="width:250px; height:80px;" class="box">
<nav class="navbar navbar-light">
<form class="form-inline">
<input id="searchingProducts" style="width: 450px" class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search"/>
<button id="searchButton" onclick="getProducts()" class="btn btn-secondary"><i class="fas fa-search"></i></button>
</form>
</nav>
</div>
<div id="hesabimButton" style="margin-left:1px; margin-top: 7px; width:110px;" class="box">
<button onclick="location.href='http://localhost:8080/ECommerceProject/UserProfile.jsp';" class="btn btn-primary"><i class="fas fa-user-circle"></i> Hesabim</button>
</div>
<div id="favoriButton" style="margin-left:1px; margin-top: 7px; width:120px;" class="box">
<button onclick="location.href='http://localhost:8080/ECommerceProject/favoriProducts.jsp';" class="btn btn-danger"><i class="fas fa-heart"></i> Favorilerim</button>
</div>
<div id="sepetButton" style="margin-left: 5px; margin-top: 7px; width:110px;" class="box">
<div><button onclick="location.href='http://localhost:8080/ECommerceProject/basket.jsp';" class="btn btn-warning"><i class="fas fa-shopping-cart"></i> Sepetim</button>
</div>
<div id="ordersButton" style="margin-left:1px; margin-top: 7px; width:110px;" class="box">
<button onclick="location.href='http://localhost:8080/ECommerceProject/orders.jsp';" class="btn btn-success"><i class="fas fa-list"></i> Orders</button>
</div>
</div>
<div style ="display: flex; flex-direction: row; padding-left: 0px;">
<div style="width: 250px; position: static">
<table class="table table-hover">
<thead class="thead-dark">
<tr>
<th scope="col">MENU</th>
</tr>
</thead>
<tbody id="menu"></tbody>
</table>
</div>
<div id="third" style ="display: flex; flex-direction: column;"></div>
</div>
```

```
function getMenu(){
$.ajax({
url: "ajaxGetMenu.jsp",
type: "GET", //
success: function(response) {
$("#menu").html(response);
},
error: function(xhr, status, error) {
console.error(error);
}
});
}

function subMenu(menuNo) {
var menuId = menuNo;
$.ajax({
url: "ajaxGetSubMenu.jsp",
type: "GET",
data: {
menuId: menuId
},
success: function(response) {
$("#submenuseem" + menuId).html(response);
},
error: function(xhr, status, error) {
console.error(error);
}
});
}
```

```
function getProducts(){
var productName = $('#searchingProducts').val();
$.ajax({
url: "ajaxGetProductsSearch.jsp",
type: "GET",
data: {
productName: productName
},
success: function(response) {
$("#third").html(response);
},
error: function(xhr, status, error) {
console.error(error);
}
});
}

function getProductsSubMenu(subMenuId) {
$.ajax({
url: "ajaxGetProductsSubMenu.jsp",
type: "GET",
data: {
subMenuId: subMenuId
},
success: function(response) {
$("#third").html(response);
},
error: function(xhr, status, error) {
console.error(error);
}
});
}
```

```
function favProducts(productId) {
    $.ajax({
        url: "ajaxFavProduct.jsp",
        type: "GET",
        data: {
            productId: productId
        },
        success: function(responseMessage) {
            alert(responseMessage.trim());
        },
        error: function(xhr, status, error) {
            alert("Kayıt Edilemedi Hatalı.");
            console.error(error);
        }
    });
}
```

```
function addBasket(productId) {
    $.ajax({
        url: "ajaxAddBasket.jsp",
        type: "GET",
        data: {
            productId: productId
        },
        success: function(responseMessage) {
            alert(responseMessage.trim());
        },
        error: function(xhr, status, error) {
            alert("Kayıt Edilemedi Hatalı.");
            console.error(error);
        }
    });
}
```

ajaxGetMenu.jsp:

This page takes the menu names and menu ids from the menu table in the database and writes the menu names to the screen with a while loop.

```
<body>
<%
    String url = "jdbc:postgresql://localhost/ecommerce";
    Connection con;
    ResultSet rs;
    String s;
    String sqlstr = "SELECT * FROM menu";
    Statement st;
    Class.forName("org.postgresql.Driver");
    con = DriverManager.getConnection(url, "postgres", "root");
    st = con.createStatement();
    rs = st.executeQuery(sqlstr);
%>
<%int count=0;
    while(rs.next()) {
        count++;
        String submenuseemId = "submenuseem"+rs.getString(1);
    %>
<tr onclick="showHiddenDiv('<%=submenuseemId%>') "class="menu-count">
    <td hidden=""><%=rs.getString(1)%></td>
    <td class="table-secondary" onclick="subMenu('<%=rs.getString(1)%>') "><%=rs.getString(2)%></td>
</tr>
<tr id = "<%=submenuseemId%>">
</tr>
<tbody id="submenuseem"></tbody>
<%}%>
```

ajaxGetProductsSearch.jsp:

This page assigns the product name that comes as a parameter from the getProducts function to a variable and searches for the product name parameter in the products table in the database and returns the products containing this letter in the product name.

```
String productName = request.getParameter("productName");
String url = "jdbc:postgresql://localhost/ecommerce";
Connection con;
ResultSet rs;
String s;
String sqlstr = "SELECT product_name,price,product_image,product_id FROM products WHERE product_name ILIKE '%" + productName + "%'";
Statement st;
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(url,"postgres","root");
st = con.createStatement();
rs = st.executeQuery(sqlstr);

%>
<%int count=0;
while(rs.next()){
count++;
%>

<div style ="display: flex; flex-direction: row;padding-left: 10px; margin-bottom: 10px">

<div style ="display: flex; flex-direction: column;" class="myDiv">
<div>

</div>
<div>
<h4><%=rs.getString(1)%></h4>

</div>
<div>
<h5><%=rs.getString(2)+" TL"%></h5>
</div>
<div><button onclick="addBasket('<%=rs.getString(4)%>')" class="btn btn-warning">
<i class="fas fa-shopping-cart"></i> Sepete Ekle
</button>
<button onclick="favProducts('<%=rs.getString(4)%>')" class="btn btn-danger">
<i class="fas fa-heart"></i>
</button>
</div>

</div>
```

ajaxGetSubMenu.jsp:

This page searches the menuId parameter from the getSubMenu function in the submenu table and brings up the submenus connected to that menu.

```
String menuId = request.getParameter("menuId");
String url = "jdbc:postgresql://localhost/ecommerce";
Connection con;
ResultSet rs;
String s;
String sqlstr = "SELECT * FROM submenu where menu_id="+menuId;
Statement st;
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(url,"postgres","root");
st = con.createStatement();
rs = st.executeQuery(sqlstr);

%>
</head>
<body>

<%int count=0;
while(rs.next()){
count++;
%>

<td style="display: block; margin: 0; padding: 0;">
<table class="table table-hover" style="margin: 0; padding: 0;">
<tr style="height:2px">
<td hidden=""><%=rs.getString(1)%></td>
<td id="searchButton" onclick="getProductsSubMenu('<%=rs.getString(1)%>')" style="height:2px "><%=rs.getString(3)%></td>
</tr>
</table>
</td>

<%}%>
```


ajaxGetProductsSubMenu.jsp:

This page searches the products table for the subMenuId parameter from the getProductsSubMenu function and returns the products linked to that submenu.

```
<%
String subMenuId = request.getParameter("subMenuId");
String url = "jdbc:postgresql://localhost/ecommerce";
Connection con;
ResultSet rs;
String s;
String sqlstr = "SELECT product_name,price,product_image FROM products WHERE submenu_id =" + subMenuId + ";";
Statement st;
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(url,"postgres","root");
st = con.createStatement();
rs = st.executeQuery(sqlstr);
%>

<%int count=0;
while(rs.next()){
count++;
%>

<div style = "display: flex; flex-direction: row; padding-left: 10px; margin-bottom: 10px">

    <div style = "display: flex; flex-direction: column;" class="myDiv">
        <div>
            
        </div>
        <div>
            <h4><%=rs.getString(1)%></h4>
        </div>
        <div>
            <h5><%=rs.getString(2)+ " TL"%></h5>
        </div>
        <div><button class="btn btn-warning">
            <i class="fas fa-shopping-cart"></i> Sepete Ekle
        </button>
        <button class="btn btn-danger">
            <i class="fas fa-heart"></i>
        </button>
    </div>

```

ajaxFavProducts.jsp:

This page inserts the productId parameter from the favProducts function and the userid from the session into the favorite_products table in the database.

```
<%
String hata="";
String userId = (String)session.getAttribute("userId");
String productId = request.getParameter("productId");

try {
String url = "jdbc:postgresql://localhost/ecommerce";
Connection connection;
String s;
Class.forName("org.postgresql.Driver");
connection = DriverManager.getConnection(url,"postgres","root");
String INSERT_USERS_SQL = "INSERT INTO favori_products " +
    " (user_id, product_id) VALUES " +
    " (?, ?);";

PreparedStatement preparedStatement = connection.prepareStatement(INSERT_USERS_SQL);
preparedStatement.setInt(1, Integer.parseInt(userId));
preparedStatement.setInt(2, Integer.parseInt(productId));

int affectedRows = preparedStatement.executeUpdate();
if (affectedRows > 0) {
    // Ekleme başarılı oldu
    out.println("Ürün favorilere eklendi.");
} else {
    // Ekleme başarısız
    out.println("Ürün favorilere eklenemedi.");
}

preparedStatement.close();
connection.close();
}
```

ajaxAddBasket.jsp:

This page sends the productId parameter coming from the favProducts function and the userid from the session as parameters to the basket_add function in the database.

```
%  
String hata="";  
String userId = (String)session.getAttribute("userId");  
String productId = request.getParameter("productId");  
Connection connection = null;  
CallableStatement callableStatement = null;  
  
try {  
    String url = "jdbc:postgresql://localhost/ecommerce";  
    String s;  
    Class.forName("org.postgresql.Driver");  
    connection = DriverManager.getConnection(url, "postgres", "root");  
  
    // CallableStatement kullanarak saklı fonksiyonu çağırma  
    callableStatement = connection.prepareCall("{call basket_add(?, ?)}");  
    callableStatement.setInt(1, Integer.parseInt(userId)); // user_id parametresi  
    callableStatement.setInt(2, Integer.parseInt(productId)); // product_id parametresi  
  
    boolean hadResults = callableStatement.execute();  
  
    if (hadResults) {  
        out.println("Ürün Sepete eklendi.");  
    } else {  
        out.println("Ürün Sepete eklenemedi.");  
    }  
}
```

basket_add function(from database):

The basket_add function checks if there is a basket record for the user whose is_active variable is true. If it exists, it inserts that product into the basket_items table. If not, it first adds a new row for that user to the basket table and then inserts the product into the basket_items table.

```
OR REPLACE FUNCTION IF EXISTS public.basket_add(integer, integer),  
  
CREATE OR REPLACE FUNCTION public.basket_add(  
    user_id1 integer,  
    product_id1 integer)  
    RETURNS void  
    LANGUAGE 'plpgsql'  
    COST 100  
    VOLATILE PARALLEL UNSAFE  
AS $BODY$  
DECLARE  
    basket_idx INTEGER;  
BEGIN  
    -- Kullanıcının aktif bir sepeti var mı kontrol et  
    SELECT basket_id INTO basket_idx  
    FROM basket  
    WHERE user_id = user_id1 AND is_active = true;  
  
    IF basket_idx IS NULL THEN  
        -- Kullanıcının aktif bir sepeti yoksa yeni bir sepet oluştur  
        INSERT INTO basket (user_id, is_active) VALUES (user_id1, true) RETURNING basket_id INTO basket_idx;  
    END IF;  
  
    -- Sepet öğesini ekleyin  
    INSERT INTO basket_items (basket_id, product_id, quantity) VALUES (basket_idx, product_id1, 1);  
END;
```

Basket.jsp :

E -Commerce Main



localhost:8080/ECommerceProject/basket.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimSiparişlerim

Sepetim

	PRODUCT IMAGE	PRODUCT NAME	PRICE	AMOUNT	
1		IPHONE 12	50000.99 TL	1	

Toplam Tutar : 50000,99 TL

Ödeme Yap

E -Commerce Main



localhost:8080/ECommerceProject/basket.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimOrders

Sepetim

	PRODUCT IMAGE	PRODUCT NAME	PRICE	AMOUNT	
1		IPHONE 12	50000.99 TL	1	

Toplam Tutar : 50000,99 TL

Ödeme Yap

PAYING PAGE - İş - Microsoft Edge

localhost:8080/ECommerceProject/Paying.jsp?sellid=26

localhost:8080 şunu diyor:
Kayıt başarılı

Tamam

Credit Card Number
444405555666777

Month
3

Year
2024

CVV/CVC
123

ResetPAY

E -Commerce Main

localhost:8080/ECommerceProject/basket.jsp

NovaNest

Online Alışveriş

HesabımFavorilerimSepetimOrders

Sepetim

	PRODUCT IMAGE	PRODUCT NAME	PRICE	AMOUNT	
--	---------------	--------------	-------	--------	--

Toplam Tutar : 0,00 TL

Ödeme Yap

E -Commerce Main





localhost:8080/ECommerceProject/basket.jsp

NovaNest

Online Alışveriş

FavorilerimSepetimOrders

Sepetim

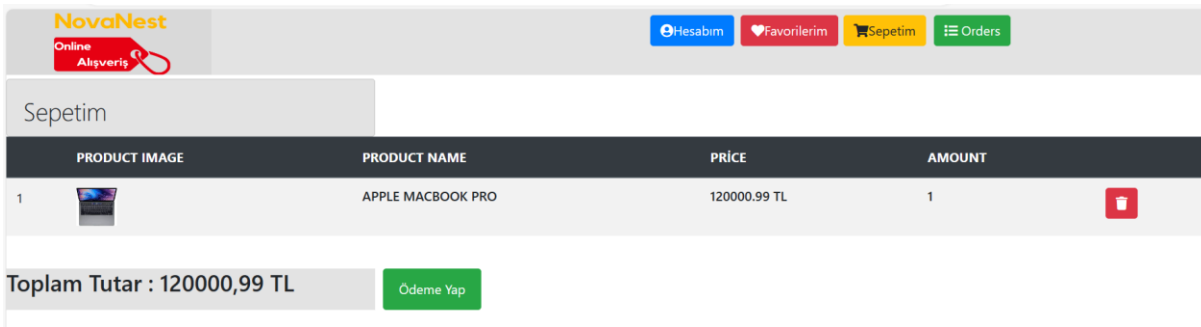
	PRODUCT IMAGE	PRODUCT NAME	PRICE	AMOUNT	
1		APPLE MACBOOK PRO	120000.99 TL	1	
2		XIOMI 11 ULTRA	30000.99 TL	1	

Toplam Tutar : 150001,98 TL

Ödeme Yap

localhost:8080 şunu diyor:
SİLİNDİ.

Tamam



When the basket.jsp page loads, the getBasket function runs. The `getBasket()` function takes the product information from the `ajaxGetBasket.jsp` page and displays it on the page.

When the `pay` button is clicked, the `pay()` function runs and sends the `basketId` of the basket to be purchased as a parameter. And the `basketId` is named `sellId` in the `pay()` function. The `sellId` is sent to the payment page as a parameter. And the `Paying.jsp` page opens on the basket page.

When the `trash` can button is pressed, the `basketProductsDelete()` function runs and sends the `basketItemId` value of the product to be deleted as a parameter. `basketItemId` is sent as a parameter to the `ajaxBasketProductsDel.jsp` page. and prints the message from the `ajaxBasketProductsDel.jsp` page as a warning on the screen.

```
function getBasket() {
    $.ajax({
        url: "ajaxGetBasket.jsp", // Ajax isteğinin yönlendirileceği sayfa
        type: "GET",
        success: function(response) {
            // İstek başarılıysa yapılacak işlemler
            $("#third").html(response); // Sonucu 'result' div'ine ekleme
        },
        error: function(xhr, status, error) {
            // İstek başarısız olursa yapılacak işlemler
            console.error(error); // Hata mesajını konsola yazdırma
        }
    });
}

function pay(sellId) {
    window.open ("http://localhost:8080/ECommerceProject/Paying.jsp?sellId="+sellId,"mywindow","menubar=1,resizable=1,width=550,height=650px");
}

function basketProductsDelete(basketId) {
    $.ajax({
        url: "ajaxBasketProductDel.jsp",
        type: "GET",
        data: {
            basketId: basketId
        },
        success: function(responseMessage) {
            alert(responseMessage.trim());
            window.location.reload();
        },
        error: function(xhr, status, error) {
            alert("Silinemedi Hatalı.");
            console.error(error);
        }
    });
}
```

ajaxGetBasket.jsp:

This page takes the `userId` from the session object and sends this `userId` as a parameter to the `get_user_basket_products` function in the database and lists the results as a table.

```
<%
    String userId = (String)session.getAttribute("userId");

    String url = "jdbc:postgresql://localhost/ecommerce";
    String sqlstr = "{CALL get_user_basket_products(?)}";
    Connection con;
    CallableStatement callableStatement = null;
    ResultSet rs = null;
    Class.forName("org.postgresql.Driver");
    con = DriverManager.getConnection(url, "postgres", "root");
    callableStatement = con.prepareCall(sqlstr);
    callableStatement.setInt(1, Integer.parseInt(userId));
    rs = callableStatement.executeQuery();
%>
```

```
<table class="table table-striped table-hover">
  <thead class="thead-dark">
    <tr>
      <th></th>
      <th>PRODUCT IMAGE</th>
      <th>PRODUCT NAME</th>
      <th>PRICE</th>
      <th>AMOUNT</th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><%=count%></td>
      <td></td>
      <td><h4><%=rs.getString(1)%></h4></td>
      <td><h5><%=rs.getString(2)%> TL</h5></td>
      <td><h5><%=rs.getString(5)%></h5></td>
      <td><button onclick="basketProductsDelete('<%=rs.getString(6)%>')" class="btn btn-danger">
        <i class="fas fa-trash"></i></button></td>
    </tr>
  </tbody>
</table>
float sayi = 12.3456f;
```

get_user_basket_products function(from database):

The function links the `basket`, `basket_items`, and `product` tables together to return the items in the basket as a table with the `userId` as a parameter and the user basket's `is_active` variable is true.

```
CREATE OR REPLACE FUNCTION public.get_user_basket_products(
    user_id1 integer)
  RETURNS TABLE(
    product_name character varying, price numeric,
    product_image character varying, product_id integer,
    amount integer, basket_items_id integer, basket_id integer)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
  RETURN QUERY
  SELECT products.product_name,
    products.price,
    products.product_image,
    products.product_id,
    basket_items.quantity,
    basket_items.basket_item_id,
    basket.basket_id
  FROM basket,basket_items,products
  WHERE basket.user_id = user_id1 AND
    basket.is_active = true AND
    basket.basket_id = basket_items.basket_id AND
    basket_items.product_id =products.product_id;
END;
```

ajaxBasketProductsDel.jsp:

This page takes the `basketItemId` as a parameter and assigns this `basketItemId` to a variable. and deletes the record for that `basketItemId` from the `basketItems` table in the database and sends a message saying it was 'Silindi' or 'Silinemedi'.

```
String basketId = request.getParameter("basketId");

String url = "jdbc:postgresql://localhost/ecommerce";
Connection connection;
Class.forName("org.postgresql.Driver");
connection = DriverManager.getConnection(url, "postgres", "root");

String deleteSQL = "DELETE FROM basket_items WHERE basket_item_id = ?";
PreparedStatement deleteStatement = connection.prepareStatement(deleteSQL);
deleteStatement.setInt(1, Integer.parseInt(basketId));
int rowCount = deleteStatement.executeUpdate();

if (rowCount > 0) {
    out.println("Silindi.");
} else {
    out.println("Silinemedi.");
}
connection.close();
```

Paying.jsp:

This page takes `sellId` as a parameter and assigns this `sellId` to a variable.

There are relevant fields on the page to enter card information. After the card information is entered.

When the `Pay button` is clicked, it sends the `sellId` to the `ajaxCreateOrder.jsp` Page. If the payment is successful, the screen will warn that the 'Ödeme Başarılı'.

```
function createOrder(){
    var sellId = '<%=sellId%>';

    $.ajax({
        url: "ajaxCreateOrder.jsp",
        type: "GET",
        data: {sellId: sellId
        },
        success: function(responseMessage) {
            alert("Kayıt başarılı");
            window.location.reload();
        },
        error: function(xhr, status, error) {
            alert("Kayıt Edilemedi Hatalı.");
            console.error(error);
        }
    });
}
```

ajaxCreateOrder.jsp:

This page takes sellId as a parameter and assigns it to a variable. It passes it as a parameter to the **create_order function** in the database.

```
String hata = "";
String userId = (String)session.getAttribute("userId");
String sellId = request.getParameter("sellId");
Connection connection = null;
CallableStatement callableStatement = null;
try {
    String url = "jdbc:postgresql://localhost/ecommerce";
    String s;
    Class.forName("org.postgresql.Driver");
    connection = DriverManager.getConnection(url, "postgres", "root");

    callableStatement = connection.prepareCall("{call create_order(?)}");
    callableStatement.setInt(1, Integer.parseInt(sellId));

    boolean hadResults = callableStatement.execute();

    if (hadResults) {
        out.println("Sipariş Oluşturuldu.");
    } else {
        out.println("Sipariş Oluşturulamadı.");
    }
} catch (Exception e) {
    // Hata yakalanırsa, hatayı kullanıcıya göster
    hata = "Error: " + e.getMessage();
    out.print(hata);
}
```

create_order function(from database):

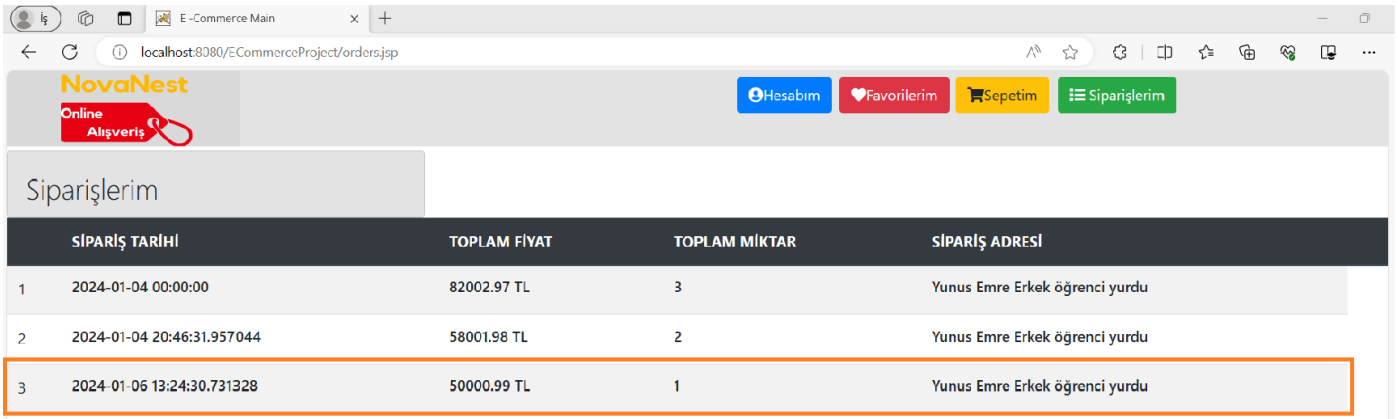
The function finds products that have the basketId provided as a parameter. By connecting the basket_items and product tables, it collects the prices of the products in the basket and assigns them to the total_amount variable. By connecting the basket and users table, it assigns the address of the user in basketId to the shipping_address variable. And inserts shipping_address, order_date, total_amount variables into the orders table. Then it sets the is_active variable of that basket to false because the order is created.

```
CREATE OR REPLACE FUNCTION public.create_order(
    basket_id1 integer)
    RETURNS void
    LANGUAGE 'plpgsql'
    COST 100
    VOLATILE PARALLEL UNSAFE
AS $BODY$
DECLARE
    total_amount1 NUMERIC(10,2);
    shipping_address1 VARCHAR(255);
BEGIN
    SELECT SUM(products.price) INTO total_amount1
    FROM basket_items
    INNER JOIN products ON basket_items.product_id = products.product_id
    WHERE basket_items.basket_id = basket_id1;

    SELECT users.adress INTO shipping_address1
    FROM basket
    INNER JOIN users ON basket.user_id = users.user_id
    WHERE basket.basket_id = basket_id1;

    INSERT INTO orders (basket_id, order_date, total_amount, shipping_address)
    VALUES (basket_id1, CURRENT_TIMESTAMP, total_amount1, shipping_address1);
    -- Basket tablosundaki is_active değerini false yapma işlemi
    UPDATE basket
    SET is_active = false
    WHERE basket_id = basket_id1;
```

Orders.jsp :



	SİPARİŞ TARİHİ	TOPLAM FİYAT	TOPLAM MİKTAR	SİPARİŞ ADRESİ
1	2024-01-04 00:00:00	82002.97 TL	3	Yunus Emre Erkek öğrenci yurdu
2	2024-01-04 20:46:31.957044	58001.98 TL	2	Yunus Emre Erkek öğrenci yurdu
3	2024-01-06 13:24:30.731328	50000.99 TL	1	Yunus Emre Erkek öğrenci yurdu

When the **orders.jsp** page loads, the **getOrders** function runs. The **getOrders()** function takes the orders information from the **ajaxGetOrders.jsp** page and displays it on the page.

```
$(document).ready(function() {
    getOrders();
});

function getOrders() {
    $.ajax({
        url: "ajaxGetOrders.jsp",
        type: "GET",
        success: function(response) {
            $("#third").html(response);
        },
        error: function(xhr, status, error) {
            console.error(error);
        }
    });
}
```

ajaxGetOrders.jsp:

This page takes the **userId** from the session object and sends this **userId** as a parameter to the **get_orders_summary** function in the database and lists the results as a table.

```
String userId = (String)session.getAttribute("userId");

String url = "jdbc:postgresql://localhost/ecommerce";
String sqlstr = "{CALL get_orders_summary(?)}";
Connection con;
CallableStatement callableStatement = null;
ResultSet rs = null;
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(url, "postgres", "root");
callableStatement = con.prepareCall(sqlstr);
callableStatement.setInt(1, Integer.parseInt(userId));
rs = callableStatement.executeQuery();
```



```

<table class="table table-striped table-hover ">
  <thead class="thead-dark">
    <tr>
      <th></th>
      <th>SİPARİŞ TARİHİ</th>
      <th>TOPLAM FİYAT</th>
      <th>TOPLAM MİKTAR</th>
      <th>SİPARİŞ ADRESİ</th>
      <th></th>
    </tr>
  </thead>
  <%
    int count=0;
    while(rs.next()){
      count++;
    %>
    <tr>
      <td><%=count%></td>
      <td><h4><%=rs.getString(1)%></h4></td>
      <td><h5><%=rs.getString(2)+" TL"%></h5></td>
      <td><h5><%=rs.getString(4)%></h5></td>
      <td><h5><%=rs.getString(3)%></h5></td>
    </tr>
    <%}%>
  </table>

```

get_orders_summary function(from database):

The function takes the userId variable as a parameter. It brings the order information as a table by connecting the basket and order tables. Returns orders that the basket is_active variable to false and belong to the submitted userId.

```

CREATE OR REPLACE FUNCTION public.get_orders_summary(
  user_id1 integer)
  RETURNS TABLE(order_date timestamp without time zone,
    total_amount numeric,
    shipping_address character varying,
    total_quantity bigint)
  LANGUAGE 'plpgsql'
  COST 100
  VOLATILE PARALLEL UNSAFE
  ROWS 1000

AS $BODY$
BEGIN
  RETURN QUERY
  SELECT
    orders.order_date,
    orders.total_amount AS total_amount,
    orders.shipping_address AS shipping_address,
    SUM(basket_items.quantity) AS total_quantity
  FROM basket
  JOIN basket_items ON basket.basket_id = basket_items.basket_id
  JOIN orders ON orders.basket_id = basket.basket_id
  WHERE basket.is_active = false
  AND basket.user_id = user_id1
  GROUP BY
    orders.order_date, orders.total_amount, orders.shipping_address;
END;


```

FavoriProducts.jsp :

NovaNest
Online Alışveriş


HesabımFavorilerimSepetimOrders

FAVORITE PRODUCTS



XIOMI 11 LITE
20000.99 TL

Sepete Ekle



LENOVA IDEAPAD 3
25000.99 TL


Sepete Ekle

NovaNest
Online Alışveriş

localhost:8080 şunu diyor:
silindi.


FavorilerimSepetimOrders

FAVORITE PRODUCTS



XIOMI 11 LITE
20000.99 TL

Sepete Ekle




LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle

NovaNest
Online Alışveriş

HesabımFavorilerimSepetimOrders

FAVORITE PRODUCTS



LENOVA IDEAPAD 3
25000.99 TL

Sepete Ekle

When the `favoriProducts.jsp` page loads, the `getFavProducts` function runs. The `getFavProducts()` function takes the product information from the `ajaxGetFavProducts.jsp` page and displays it on the page.

When the trash can button is pressed, the `favProductsDelete()` function runs and sends the `favoriId` value of the product to be deleted as a parameter. `favoriId` is sent as a parameter to the

`ajaxBasketProductsDel.jsp` page. and prints the message from the `ajaxBasketProductsDel.jsp` page as a warning on the screen.

```
function getFavProduct() {
    $.ajax({
        url: "ajaxGetFavProducts.jsp",
        type: "GET",
        success: function(response) {
            $("#third").html(response);
        },
        error: function(xhr, status, error) {
            console.error(error);
        }
    });
}

function favProductsDelete(favId) {
    $.ajax({
        url: "ajaxFavProductsDel.jsp",
        type: "GET",
        data: {
            favId: favId
        },
        success: function(responseMessage) {
            alert(responseMessage.trim());
            window.location.reload();
        },
        error: function(xhr, status, error) {
            alert("Kayıt Edilemedi Hatalı.");
            console.error(error);
        }
    });
}
```

ajaxGetFavProducts.jsp:

This page assigns the `userId` value taken from the session to a variable and finds the products with `userId` by joining the `favorite_products`, and `products` tables in the database. It prints these products to the screen.

```
String userId = (String)session.getAttribute("userId");
String url = "jdbc:postgresql://localhost/ecommerce";
Connection con;
ResultSet rs;
String s;
String sqlstr = "SELECT products.product_name,products.price,products.product_image,products.product_id,favori_products.favori_id FROM "+
    " products,favori_products WHERE "+
    " favori_products.product_id = products.product_id AND favori_products.user_id = "+userId+";";

Statement st;
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(url,"postgres","root");
st = con.createStatement();
rs = st.executeQuery(sqlstr);


<div style ="display: flex; flex-direction: row;padding-left: 10px; margin-bottom: 10px">
    <div style ="display: flex; flex-direction: column;" class="myDiv">
        <div>
            
        </div>
        <div>
            <h4><%=rs.getString(1)%></h4>
        </div>
        <div>
            <h5><%=rs.getString(2)+" TL"%></h5>
        </div>
        <div>
            <button onclick="addBasket('<%=rs.getString(4)%>')" class="btn btn-warning">
                <i class="fas fa-shopping-cart"></i> Sepete Ekle
            </button>
            <button onclick="favProductsDelete('<%=rs.getString(5)%>')" class="btn btn-danger">
                <i class="fas fa-trash"></i>
            </button>
        </div>
    </div>
</div>
```

ajaxFavProductsDel.jsp:

This page takes the `favId` as a parameter and assigns this `favId` to a variable. and deletes the record for that `favoriId` from the `favori_products` table in the database. and sends a message saying it was 'Silindi' or 'Silinemedi'.

```
String favId = request.getParameter("favId");
String url = "jdbc:postgresql://localhost/ecommerce";
Connection connection;
Class.forName("org.postgresql.Driver");
connection = DriverManager.getConnection(url, "postgres", "root");
String deleteSQL = "DELETE FROM favori_products WHERE favori_id = ?";
PreparedStatement deleteStatement = connection.prepareStatement(deleteSQL);
deleteStatement.setInt(1, Integer.parseInt(favId));
int rowCount = deleteStatement.executeUpdate();
if (rowCount > 0) {
    out.println("SİLİNDİ.");
} else {
    out.println("SİLİNEMEDİ.");
}
connection.close();
```

UserProfile.jsp :



[Hesabım](#)
[Favorilerim](#)
[Sepetim](#)
[Siparişlerim](#)

Kullanıcı Bilgileri

Adı

Soyadı

Kullanıcı Adı

Telefon

Adres

Email

Şifre

Kaydet

This page assigns the userId value obtained from the session to a variable when entered. With userId, it takes the user information from the users table in the database and assigns it to variables. It displays this information in text boxes on the screen.

If this information is changed when the **Kaydet** button is pressed, the **userUpdt** function sends this information as a parameter to the **ajaxUserUpdt.jsp** page. If it is updated successfully, the screen warns that the registration is successful and brings updated information.

```
String userId = (String)session.getAttribute("userId");

String name = "";
String surname = "";
String userName = "";
String address = "";
String email = "";
String phoneNumber = "";
String password = "";

String url = "jdbc:postgresql://localhost/ecommerce";
Connection connection;
ResultSet rs;
String sqlstr = "SELECT name,surname,username,adress,email,phone_number,password FROM users WHERE user_id = "+userId+"";
Statement st;
Class.forName("org.postgresql.Driver");
connection = DriverManager.getConnection(url,"postgres","root");
st = connection.createStatement();
rs = st.executeQuery(sqlstr);

rs.next();
name = rs.getString(1);
surname = rs.getString(2);
userName = rs.getString(3);
address = rs.getString(4);
email = rs.getString(5);
phoneNumber = rs.getString(6);
password = rs.getString(7);
```

```
function userUpdt(){
    var address = $('#address').val();
    var email = $('#email').val();
    var phoneNumber = $('#phoneNumber').val();
    var password = $('#password').val();
    $.ajax({
        url: "ajaxUserUpdt.jsp",
        type: "GET",
        data: {address:address,
            email: email,
            phoneNumber:phoneNumber,
            password: password
        },
        success: function(responseMessage) {
            alert("Kayıt başarılı");
            window.location.reload();
        },
        error: function(xhr, status, error) {
            alert("Kayıt Edilemedi Hatalı.");
            console.error(error);
        }
    });
}
```

ajaxUserUpdt.jsp:

This page assigns the userId value received from the session and the user information received as parameters to variables. Accesses the users table in the database with userId. and updates the user information with the user information in the relevant variables.

```
String userId = (String)session.getAttribute("userId");
String hata = "";
String address = request.getParameter("address");
String email = request.getParameter("email");
String phoneNumber = request.getParameter("phoneNumber");
String password = request.getParameter("password");
try {
    String url = "jdbc:postgresql://localhost/ecommerce";
    Connection connection;
    Statement s;
    Class.forName("org.postgresql.Driver");
    connection = DriverManager.getConnection(url,"postgres","root");
    String UPDATE_USERS_SQL = "UPDATE users SET email = ?, password = ?, phone_number = ?, adress = ? WHERE user_id = ?";
    PreparedStatement preparedStatement = connection.prepareStatement(UPDATE_USERS_SQL);
        preparedStatement.setString(1, email);
        preparedStatement.setString(2, password);
        preparedStatement.setString(3, phoneNumber);
        preparedStatement.setString(4, address);
        preparedStatement.setInt(5, Integer.parseInt(userId));

        out.println(preparedStatement);
        preparedStatement.executeUpdate();
    }catch (Exception e) {
        hata = "Error: " + e.getMessage();
        out.print(hata);
    }
}
```