

Project: Product Details Portal

My GitHub link :

<https://github.com/SK-NOORALAM-RAHAMAN/Phase-2/tree/master/Practise%20Projects/Product%20Details%20Portal>

Problem statement:

As a part of developing an ecommerce web application, you have to prototype a form for adding products into the system entered by the users. There is no database involved here, so the product is just captured and displayed without storing it anywhere.

Development Environment

- Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
- Apache Tomcat Server v9.0
- JRE: OpenJDK Runtime Environment 11.0.2

This lab has ten subsections, namely:

- 1.1.1 Creating a dynamic web project
- 1.1.2 Creating an HTML page
- 1.1.3 Creating a servlet GetHandler.java
- 1.1.4 Creating a servlet PostHandler.java
- 1.1.5 Configuring web.xml
- 1.1.6 Checking for servlet-api.jar
- 1.1.7 Building the project
- 1.1.8 Publishing and starting the project
- 1.1.9 Running the project
- 1.1.10 Pushing the code to GitHub repositories

Step 1.1.1: Creating a dynamic web project

- Open Eclipse
- Go the **File** menu. Choose **New->Dynamic Web Project**
- Enter the project name as ProductDetailsPortal. Click on **Next**
- Enter nothing in the next screen and click on **Next**
- Check the checkbox **Generate web.xml deployment descriptor** and click on **Finish**
- This will create the project files in the Project Explorer

Step 1.1.2: Creating an JSP page

- In the Project Explorer, expand the project **ServletGetPost**
- Expand **WebContent**. Right click on **WebContent**. Choose **New->JSP File**
- Enter the filename as **display.jsp** and click on **Finish**

Enter the following code:

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Display</title>
</head>
<body bgcolor="Teal">
<h1>Displaying the Product Details</h1>
<hr>
<%= "Product Id : " + session.getAttribute("pid")
%> <br> <br>
<%= "Product Name : " +
session.getAttribute("pname") %> <br> <br>
<%= "Product Price : " +
session.getAttribute("pprice") %>
</body>
</html>
```

Step 1.1.3: Creating an JSP page

- In the Project Explorer, expand the project **ServletGetPost**
- Expand **WebContent**. Right click on **WebContent**. Choose **New->JSP File**
- Enter the filename as **product.jsp** and click on **Finish**
- Enter the following code:

```
<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Product Details</title>
</head>
<body bgcolor="Purple">
<h1>Enter Product Details</h1>
<hr>
<form action="App">
<input type="text" name="productId"
placeholder="PRODUCT ID"><br>
<input type="text" name="productName"
placeholder="PRODUCT NAME"><br>
<input type="text" name="productPrice"
placeholder="PRODUCT PRICE"><br>
<input type="submit" value="ENTER">
</form>
</body>
```

</html>

Step 1.1.4: Creating an Servlet page

- In the Project Explorer, expand the project **ServletGetPost**
- Expand **WebContent**. Right click on **WebContent**. Choose **New->Servlet File**
- Enter the filename as App.java and click on **Finish**
- Enter the following code:

```
package com;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/app")
public class App extends HttpServlet {
    private static final long serialVersionUID =
    1L;
    protected void doGet(HttpServletRequest
request, HttpServletResponse response) throws
ServletException, IOException {
    // TODO Auto-generated method stub
    String pId =
request.getParameter("productId");
    String pName =
request.getParameter("productName");
    String pPrice =
request.getParameter("productPrice");
```

```

HttpSession theSession =
request.getSession();
theSession.setAttribute("pid", pId);
theSession.setAttribute("pname", pName);
theSession.setAttribute("pprice", pPrice);
response.sendRedirect("display.jsp");
}
}

```

Step 1.1.5: Configuring web.xml

- In the Project Explorer, expand **ServletGetPost->WebContent->WEB-INF**
- Double click **web.xml** to open it in the editor
- Enter the following script:

```

<?xml version="1.0" encoding="UTF-8"?>

<web-app >
  <display-name>Product details portal</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
  </welcome-file-list>
  <servlet>
    <description></description>
    <display-name>App</display-name>
    <servlet-name>App</servlet-name>
    <servlet-class>com.App</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>App</servlet-name>
    <url-pattern>/App</url-pattern>
  </servlet-mapping>
</web-app>

```

Step 1.1.6: Checking for servlet-api.jar

- Before building the project, we need to add servlet-api.jar to the project
- Servlet-api.jar file is already present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
- To add it to the project, follow the below mentioned steps:
 - In the Project Explorer, right click on **ProductDetailsPortal** and choose **Properties**
 - Select **Java Build Path** from the options on the left

- Click on **Libraries** tab on the right
- Under **ClassPath**, expand the node that says **Apache Tomcat**
- If there is an existing entry for **servlet-api.jar**, then click on **Cancel** and exit the window
- If it is not there, then click on **Classpath** entry and click on **Add External JARs** button on the right
- From the file list, select **servlet-api.jar** file and click **Ok**
- Click on **Apply and Close**

Step 1.1.7: Building the project

- From the **Project** menu at the top, click on **Build**
- If any compile errors are shown, fix them as required

Step 1.1.8: Publishing and starting the project

- If you do not see the **Servers** tab near the bottom of the IDE, go to the Window menu and click **Show View->Servers**
- Right click on the **Server** entry and choose **Add and Remove**
- Click the **Add** button to move **ServletGetPost** from the **Available** list to the **Configured List**
- Click **Finish**
- Right click on the **Server** entry and click on **Publish**
- Right click on the **Server** entry and click on **Start**
- This will start the server

Step 1.1.9: Running the project

- To run the project, open a web browser and type: **<http://localhost:8080/ServletGetPost>**

Step 1.1.10: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

- Initialize your repository using the following command:

git init

- Add all the files to your git repository using the following command:

git add .

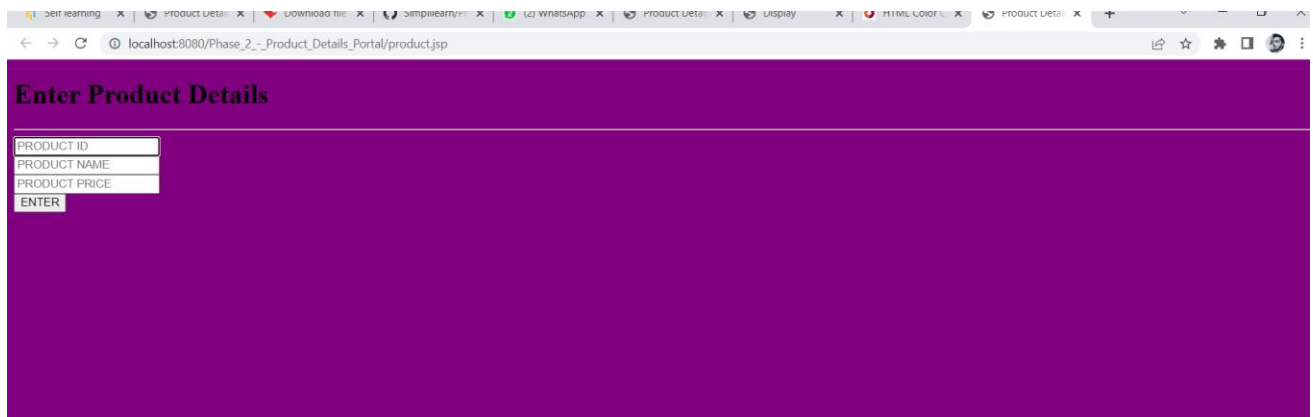
- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

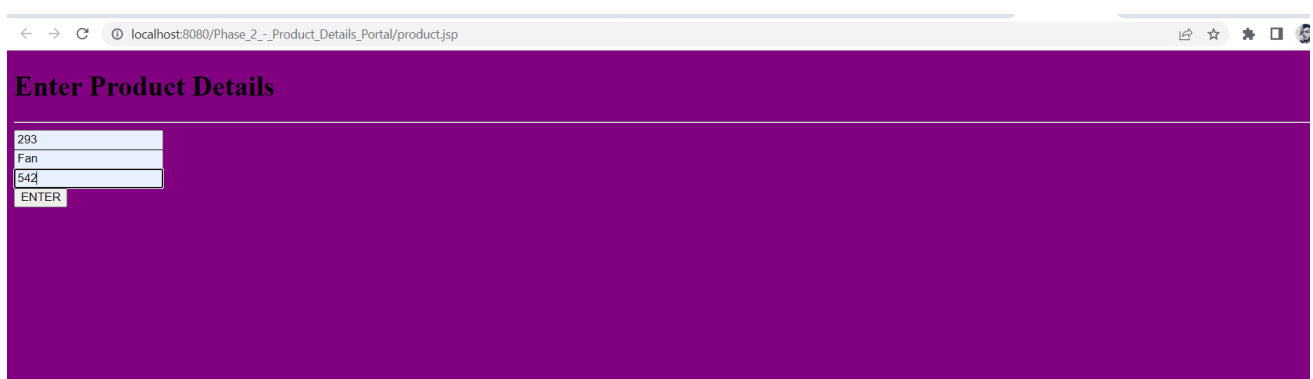
- Push the files to the folder you initially created using the following command.

git push -u origin master

Output:



A screenshot of a web browser window displaying a form titled "Enter Product Details". The form is located on a purple background. It contains four input fields: "PRODUCT ID", "PRODUCT NAME", "PRODUCT PRICE", and an "ENTER" button. The browser's address bar shows the URL "localhost:8080/Phase_2_-_Product_Details_Portal/product.jsp".



A screenshot of a web browser window displaying the same form titled "Enter Product Details". The form now contains data: "293" in the "PRODUCT ID" field, "Fan" in the "PRODUCT NAME" field, and "542" in the "PRODUCT PRICE" field. The "ENTER" button is still present. The browser's address bar shows the same URL: "localhost:8080/Phase_2_-_Product_Details_Portal/product.jsp".

