Title: Building a Product Details Portal - A Servlet-Based Web Application

Introduction:

In this project, we've developed a servlet-based web application called the "Product Details Portal." The primary goal of this application is to create a simple yet functional prototype for entering and displaying product details, a fundamental feature of an e-commerce web application. Importantly, this prototype captures and displays the product information in real-time without the need for database storage.

Background:

One of the essential features of such an application is the ability for users to add and view product details. However, for this prototype, we won't be dealing with database storage. Instead, we capture and display the product data immediately using session management.

Technologies Used:

To build this project, we've utilized the following technologies and tools:

1. **Eclipse IDE:** We selected Eclipse as our Integrated Development Environment (IDE) for developing the web application.

2.**Apache Tomcat:** Apache Tomcat serves as our chosen web server to host and execute the servlet-based application.

3. JSP (JavaServer Pages): JSP pages were employed to create the frontend user interfaces, enabling the design of product detail entry forms.

4. Servlets: The backend processing, including data capture and storage, is handled by servlets, which receive, process, and store the product data.

Requirements and Implementation:

Here's a breakdown of the key requirements and how they were implemented:

1. Product Class:

- A Java class named "Product" was created to represent the product details. This class contains attributes such as product name, description, and price, along with corresponding getters and setters.

2. Servlet for Product Entry:

- A servlet was developed to manage the product entry process. In the servlet's "doPost" method, we extracted the product details submitted through a form.

- An instance of the "Product" class was created to hold the form data.

- The product object was stored in the session to make it accessible for later retrieval.

3. JSP Page for Product Entry Form:

-We designed a JSP file named "ProductEntry.jsp" to create the user interface for entering product details. This form collects product name, description, and price from users.

4. JSP Page for Displaying Product Info:

- Another JSP file named "ProductDisplay.jsp" was created to display product information. This page retrieved product details from the session and presented them to the user.

- Conditional statements were used to handle scenarios where no product data was available.

5. Running the Application:

- Eclipse was configured to work seamlessly with Apache Tomcat, and the project was deployed on the Tomcat server.

- Users accessed the product entry form by visiting the appropriate URL.

- After entering product details, the data was instantly displayed on the "ProductDisplay.jsp" page.

Conclusion:

In this project, we successfully created a "Product Details Portal" using servlets and JSP. This portal allows users to input product information through a form and immediately see the data displayed on a separate page. Although this is a simplified prototype, it lays the groundwork for more complex e-commerce web applications.