Product Details Portal

Introduction:

The provided code constitutes a simple web application for managing product details. It allows users to input product information such as name, description, and price via a form, submits the data to a servlet for processing, and then displays the product information on a separate page. Let's delve into each component of the application:

1. SubmitDetails.jsp: This file contains the HTML form where users input product details. It includes fields for the name, description, and price of the product. Upon submission, the form sends the data to the 'ProductServlet' for processing.

2. ProductInfo.jsp: ProductInfo.jsp is responsible for displaying the product details submitted by the user. It retrieves the product information from the servlet and presents it in a structured format using HTML.

3. Product.java: This Java class defines the 'Product' object, encapsulating properties such as name, description, and price. It includes getter and setter methods for accessing and modifying these properties.

4. ProductServlet.java: ProductServlet is a Java servlet responsible for handling requests sent by the form submission in SubmitDetails.jsp. It extracts the product details from the request parameters, creates a Product object, stores it in a session attribute, and then forwards the request to ProductInfo.jsp for displaying the product information.

5. web.xml: The web.xml file is the deployment descriptor for the web application. It maps servlets to their corresponding URLs and specifies other configuration details. In this case, it maps the 'ProductServlet' to the '/ProductServlet' URL pattern.

Workflow:

1. User accesses the SubmitDetails.jsp page, where they input product details.
2. Upon submitting the form, the data is sent to the ProductServlet for processing.
3. ProductServlet extracts the product details from the request parameters, creates a Product object, and stores it in the session.
4. The request is then forwarded to ProductInfo.jsp.

5. ProductInfo.jsp retrieves the product information from the session attribute and displays it to the user.

Conclusion:

This web application provides a basic framework for managing product details. It demonstrates the use of Java servlets, JSPs, and session management to create a simple yet functional product information system. Further enhancements could include validation of user input, error handling, and integration with a database for persistent storage of product data.

Algorithm for Product Details Portal:

1. **Start:**
    - Begin the execution of the program.
2. **Display SubmitDetails.jsp:**
    - Load the SubmitDetails.jsp page in the web browser.
    - Display a form with fields for the product name, description, and price.
3. **User Input:**
    - User enters the details of the product into the form.
    - Validate the input:
        - Ensure that all fields are filled.
        - Ensure that the price is a valid numeric value.
4. **Submit Form:**
    - User submits the form.
5. **Handle Form Submission (ProductServlet):**
    - ProductServlet receives the submitted form data.
    - Extract the product name, description, and price from the request parameters.

    - Create a new instance of the Product class.
    - Set the attributes of the Product object with the extracted data.
    - Store the Product object in the session attribute.

6. **Redirect to ProductInfo.jsp:**
    - Redirect the user to the ProductInfo.jsp page.
7. **Display Product Information:**
    - Load ProductInfo.jsp in the web browser.
    - Retrieve the Product object from the session attribute.
    - Display the product name, description, and price on the page.
8. **End:**
    - End the execution of the program.