

Adding a New Product in the Database

Writeup

1. Create the "hibernate.cfg.xml" file:
 - Create a new XML file named "hibernate.cfg.xml"
 - Inside the file, define the Hibernate configuration properties, such as database connection details, dialect, and mapping information.
 - Include the necessary dependencies.
2. Create the "DBUtil" class:
 - Create a new Java class named "DBUtil".
 - Inside the class, define a method ("dbConn") that establishes a database connection using Hibernate.
 - Initialize a Hibernate SessionFactory and return an open Session from it.
 - Use the "configure" method of the SessionFactoryBuilder to load the Hibernate configuration from the "hibernate.cfg.xml" file.
3. Create the "Product" class:
 - Create a new Java class named "Product".
 - Annotate the class with the "@Entity" annotation to mark it as an entity for Hibernate.
 - Define the class properties (pID, pName, pCost) and provide the necessary getter and setter methods.
 - Use appropriate annotations ("@Id", "@GeneratedValue") to define the primary key property and its generation strategy.
4. Create the "index.jsp" file:
 - Create a new JSP file named "index.jsp"
 - Design form to capture the product details (product name and cost).
 - Include two buttons in the form: one for adding a product and another for viewing available products.
5. Create the "AddProductServlet" servlet:
 - Create a new Java servlet named "AddProductServlet".
 - Override the "doPost" method to handle the form submission.
 - Retrieve the product details from the request parameters.
 - Create an instance of the "Product" class and set the retrieved details.
 - Use the "DBUtil" class to establish a database connection.

- Open a Hibernate session, begin a transaction, and save the product instance using the session's "save" method.
- Commit the transaction, close the session, and redirect the user to a success page.

6. Create the "ShowProductsServlets" servlet:

- Create a new Java servlet named "ShowProductsServlets".
- Override the "doPost" method to handle the button click event for viewing available products.
- Use the "DBUtil" class to establish a database connection.
- Open a Hibernate session, begin a transaction, and execute an HQL query to fetch all products.
- Retrieve the query result as a list of "Product" objects.
- Commit the transaction, close the session, and generate an HTML page to display the product list.
- Include a button or link to go back to the "index.jsp" page.

7. Create the "success.jsp" file:

- Create a new JSP file named "success.jsp".
- Design to display a success message.
- Include an inclusion directive to display the "index.jsp" page as well.