EMP No: 2577108
Name: Pramod

Algorithm: Adding a New Product in the Database

1. Create a class named Hibernate Util. Declare a private static variable session Factory of type Session Factory. Create a static block.
   - Inside the static block, initialize session Factory as follows:
     a. Create a Standard Service Registry using Standard Service Registry Builder and configure it with "hibernate.cfg.xml".
     b. Build the Metadata using Meta data Sources and the standard Registry.
     c. Build the session Factory using the meta-Data.
   - If any exception occurs, throw an Exception In Initializer Error. Create a public static method named get Session Factory that returns session Factory.

2. Create a class named E Product. Declare private instance variables: ID of type long, name of type String, price of type Big Decimal, and model of type String. Create a default constructor. Create a parameterized constructor that initializes the instance variables. Create getter and setter methods for all the instance variables.

3. Create a class named add Product which extends Http Servlet. Override the doGet method.
   - Inside the doGet method, do the following:
     a. Get the Session Factory instance using Hibernate Util.get Session Factory().
     b. Open a new Session using the Session Factory.
     c. Get the request parameters for "product Name", "price", and "model".
     d. Parse the "price" parameter to a Double and convert it to a Big Decimal.
     e. Create a new instance of E Product and set its properties using the request parameters.
     f. Begin a new transaction using the Session.
     g. Save the E Product instance using the Session.
     h. Commit the transaction.
     i. Query all the E Product objects using the Session and store them in a list.
     j. Close the Session.
     k. Get the Print Writer from the response and write the HTML output.
   - Write the product listing by iterating over the list of products. Override the doPost method and call the doGet method with the same parameters.

4. Create a JSP file named "index.jsp". Define the HTML markup for the form to add a new product. Use the "temp" URL as the form action.

5. Create an XML file named "web.xml". Define the web-app element with the necessary attributes and namespaces. Define a servlet element for the add Product servlet. Define a servlet-mapping element to map the servlet to the "/temp" URL.

6. Create an XML file named "hibernate.cfg.xml". Define the hibernate-configuration element. Inside the session-factory element, set the necessary properties for the database connection. Include the mapping resource for the E Product class.
7. Create an XML file named "Eproduct.hbm.xml". Define the hibernate-mapping element. Inside the class element, define the ID property, name property, price property, and model property.