**Algorithm:**

1)Import the necessary packages and libraries for servlets, JSP, and Hibernate.

2)Create a JSP page (index.jsp) to display the form for adding new products.

a) Design the form with input fields for id, name, color, and price.

b) Include JavaScript or HTML5 validation for form fields.

3)Create a servlet (ProductServlet) to handle form submission and database operations.

a) Retrieve the form data from the request parameters.

b) Validate the form data for any missing or incorrect information.

c) Create a Hibernate session factory using HibernateUtil.

d) Start a Hibernate session.

e) Create a new Product object and set its properties with the form data.

f) Save the Product object to the database using Hibernate's session.save() method.

g) Commit the transaction and close the session.

4)Display a success message and redirect back to the index.jsp page.

a) Add a success message in the response output stream.

b) Use response.sendRedirect() to redirect back to the index.jsp page.

5)Create the Product class to represent the product entity.

a) Define private instance variables for id, name, color, and price.

b) Generate getters and setters for the instance variables.

6)Create the HibernateUtil class to configure Hibernate and provide a session factory.

a) Use Hibernate's Configuration class to load the Hibernate configuration file (hibernate.cfg.xml).

b) Build the session factory using the loaded configuration.

7)Configure Hibernate using the hibernate.cfg.xml file.

a) Specify the database connection details, such as URL, username, and password.

b) Set the dialect and other Hibernate properties.

8)Create the Hibernate mapping file (product.hbm.xml) to map the Product class to the database table.

a) Define the class, properties, and their mappings to table columns.

9)Set up the web application in the web.xml deployment descriptor.

a) Map the ProductServlet to a URL pattern ("/Product" in this case).

10)Deploy the web application on the Apache Tomcat server.