Adding a New Product in the Database

Index.html

Written a simple code to take Product name, Product description and Product price from the user code given below.

```
<form align="center" action="save" method="post" >
        Product Name: <input type="text" name="Pname" placeholder="Name">
        <br>
        <br>
Product Description: <input type="text" name="Pdesc" placeholder="Discripiton">
        <br>
        <br>
        Product Price: <input type="text" name="Pprice" placeholder="Price">
        <br>
        <br>
        <input type="submit" value="Add">
</form>
```

Hibernate.cfg.xml

In this file we are providing the details of our driver and sql database in which we are going to store all the data. For ex: driver details, localhost, username, password, and mapping the resource file in which details of the object file present etc.

```
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/productdata</property>
        <property name="connection.username">root</property>
        <property name="connection.password">978213</property>
        <property name="hbm.auto">update</property>
        <property name="show_sql">true</property>
        <mapping resource="product.hbm.xml"/>
```

product.hbm.xml

Providing the details of the table , and object class and mapping them with the product.java file in which we added the object details. Auto increment and generating the ID.

```
<hibernate-mapping package="com.simplilearn.demo">
<class name="Product" table="productinfo">
<id name="ID" column="ID">
<generator class="increment"></generator>
</id>
<property name="name" type="string"></property>
<property name="description" type="string"></property>
<property name="price" type="string"></property>
</class>
```

HibernateUtils.java

StandardServiceRegistry. This package contains the contracts that make up the Hibernate native bootstrapping API (building a SessionFactory). Defines service registry contracts application are likely to want to utilize for configuring Hibernate behavior.

```
StandardServiceRegistry registry=new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
Metadata metadata= new MetadataSources(registry).getMetadataBuilder().build();
```

Product.java

This is an object class described the all the product details with there getter & setters so later we used these details as a object so user enter data in them .

```
package com.simplilearn.demo;
public class Product {
        private int ID;
        private String name;
        private String description;
        private String price;
        public Product(String name, String description, String price) {
                this.name = name;
```

```
		this.description = description;
		this.price = price;
	}
	public int getID() {
		return ID;
	}
	public void setID(int iD) {
		ID = iD;
	}
Etc.....
}
```

SaveServlet.java

create configuration

```
	Configuration configuration = new Configuration();
	configuration.configure("hibernate.cfg.xml");
```

sessionFactory - session object we are getting from factory, eg. i you want to buy car u go to car factory

sessionFactory is your datasource means- how u will be connecting to your databse

to create sessionFactory you will be needing configuration

```
	SessionFactory factory= HibernateUtils.getSessionFactory();
	Session session= factory.openSession();
	Transaction  tx= session.beginTransaction();
```

Taking all the parameters from the user and storing them in different variables.

```
String prodname= req.getParameter("Pname");
String proddesc= req.getParameter("Pdesc");
String prodprice= req.getParameter("Pprice");
```

Then inserting all the details from user side in the Product object, commit the transection closed the current session and printing the eding message to user if data saved successfully or not

```
Product s1=new Product(prodname, proddesc, prodprice);
        session.save(s1);
        tx.commit();
        session.close();
        out.print("<h4>Data inserted Successfully</h4>");
```