7. Demonstrate component mapping in Hibernate.

Code:

E product:

```
package com.ecommerce;

import java.math.BigDecimal;
import java.util.Collection;
import java.util.Date;
import java.util.List;
import java.util.Set;
import java.util.Map;


public class EProduct {
        private long ID;
        private String name;
        private BigDecimal price;
        private Date dateAdded;
        private ProductParts parts;

        public EProduct() {

        }

        public long getID() {return this.ID; }
        public String getName() { return this.name;}
        public BigDecimal getPrice() { return this.price;}
        public Date getDateAdded() { return this.dateAdded;}
        public ProductParts getParts() { return this.parts;}

        public void setID(long id) { this.ID = id;}
        public void setName(String name) { this.name = name;}
        public void setPrice(BigDecimal price) { this.price = price;}
        public void setDateAdded(Date date) { this.dateAdded = date;}
        public void setParts(ProductParts parts) { this.parts = parts;}
}
```

Hibernate util:

```
package com.ecommerce;
import org.hibernate.SessionFactory;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {
      private static final SessionFactory sessionFactory;

    static {
            try {
                    StandardServiceRegistry standardRegistry = new
StandardServiceRegistryBuilder()

.configure("hibernate.cfg.xml").build();
                    Metadata metaData = new
MetadataSources(standardRegistry).getMetadataBuilder().build();
```

```
                    sessionFactory =
metaData.getSessionFactoryBuilder().build();
            } catch (Throwable th) {
                    throw new ExceptionInInitializerError(th);
            }
    }

    public static SessionFactory getSessionFactory() {
            return sessionFactory;
    }

}
```

Product details:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.transaction.*;
import javax.xml.bind.*;

import java.io.Serializable;
import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collection;
import java.util.List;
import java.util.Map;
import java.util.Set;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import com.ecommerce.EProduct;
import com.ecommerce.HibernateUtil;
import com.ecommerce.ProductParts;


/**
* Servlet implementation class ProductDetails
*/
@WebServlet("/ProductDetails")
public class ProductDetails extends HttpServlet {
        private static final long serialVersionUID = 1L;



    /**
* @see HttpServlet#HttpServlet()
*/
    public ProductDetails() {
        super();
        // TODO Auto-generated constructor stub
```

```
    }

        /**
         * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                 try {
                        SessionFactory factory =
HibernateUtil.getSessionFactory();

                        Session session = factory.openSession();



                        List<EProduct> list = session.createQuery("from
EProduct").list();

                         PrintWriter out = response.getWriter();
                         out.println("<html><body>");

                         out.println("<b>Component Mapping</b><br>");
                         for(EProduct p: list) {
                                out.println("ID: " +
String.valueOf(p.getID()) + ", Name: " + p.getName() +
                                                ", Price: " +
String.valueOf(p.getPrice()) + ", Date Added: " +
p.getDateAdded().toString());
                                  ProductParts parts = p.getParts();
                                  out.println("Parts =" + parts.getCpu() +
", " + parts.getHdd() + ", " + parts.getRam());
                                  out.println("<hr>");
                         }

                                 session.close();

                    out.println("</body></html>");


                 } catch (Exception ex) {
                        throw ex;
                 }

        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}
```

Eproduct:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping package="com.ecommerce">
    <class name="EProduct" table="eproduct">
        <id name="ID" type="long" column="ID">
            <generator class="identity"/>
        </id>
        <property name="name" type="string" column="NAME"/>
        <property name="price" type="big_decimal" column="PRICE"/>
        <property name="dateAdded" type="timestamp" column="DATE_ADDED"/>

                <component name="parts" class="com.ecommerce.ProductParts">
                        <property name="hdd" column="parts_hdd"
type="string" />
                        <property name="cpu" column="parts_cpu"
type="string" />
                        <property name="ram" column="parts_ram"
type="string" />
                </component>
    </class>
</hibernate-mapping>
```

Driver:

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
      <session-factory>
            <!-- Database connection settings -->
            <property
name="connection.driver_class">com.mysql.jdbc.Driver</property>
            <property
name="connection.url">jdbc:mysql://localhost:3306/mydb</property>
            <property name="connection.username">root</property>
            <property name="connection.password">23@Swetha</property>
            <!-- SQL dialect -->
            <property
name="dialect">org.hibernate.dialect.MySQL57Dialect</property>

            <!-- Echo all executed SQL to stdout -->
            <property name="show_sql">false</property>

            <!--  Use annotation basaed mapping metadata -->
            <mapping class="com.samples.domain.Message"/>
      </session-factory>
</hibernate-configuration>
```

Web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
version="4.0">
  <display-name>ph2-As-set3-prj7</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
    <servlet>
    <servlet-name>ProductDetails</servlet-name>
    <servlet-class>ProductDetails</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ProductDetails</servlet-name>
    <url-pattern>/details</url-pattern>
  </servlet-mapping>
</web-app>
```

Index.html:

```
<html>

<head>

<meta charset="UTF-8">

<title>Hibernate Collection Mapping</title>

</head>

<body> <a href "details">details</a>

<br>

</body>

</html>
```