Phase – 2 Practice Project: Assisted Practice

23. Configure Hibernate using Annotations in Eclipse IDE.

✓ Database Code

create database ecommerce1;

use ecommerce1;

CREATE TABLE eproduct (

ID bigint primary key auto_increment,

name varchar(100), price decimal(10,2),

date_added timestamp default now()

)

INSERT INTO eproduct(name, price) VALUES ('HP Laptop ABC', 12000);

INSERT INTO eproduct(name, price) VALUES ('Acer Laptop ABC', 14000);

INSERT INTO eproduct(name, price) VALUES ('Lenovo Laptop ABC', 12000);

select * from eproduct;

✓ hibernate.cfg.xml

```
<?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.cj.jdbc.Driver</property>
    <property
name="connection.url">jdbc:mysql://localhost:3306/ecommerce1</property>
    <property name="connection.username">root</property>
    <property name="connection.password">Arun121212</property>

    <mapping resource="com/ecommerce/EProduct.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

✓ HibernateUtil.java

```
package com.simpli;

import org.hibernate.SessionFactory;

import org.hibernate.boot.*;
import org.hibernate.boot.registry.*;

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;
	}

}
```

✓ index.html

```
<title>Hibernate Annotation QueryDemo</title>
<h3>Hibernate Annotation Query Demo</h3>

<a href="HibernateQueryDemo">Hibernate Query Demo</a><br>
```

✓ EProduct.java

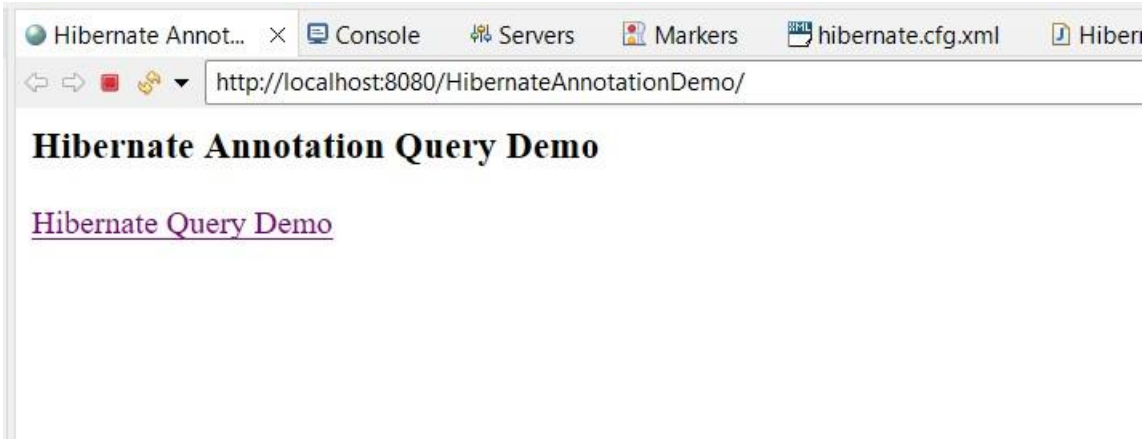
```
package com.ecomerce;

import javax.persistence.Entity;
import javax.persistence.Table;

import java.math.BigDecimal;
import java.util.Date;

import javax.persistence.*;

@Entity
@Table(name= "eproduct")
```

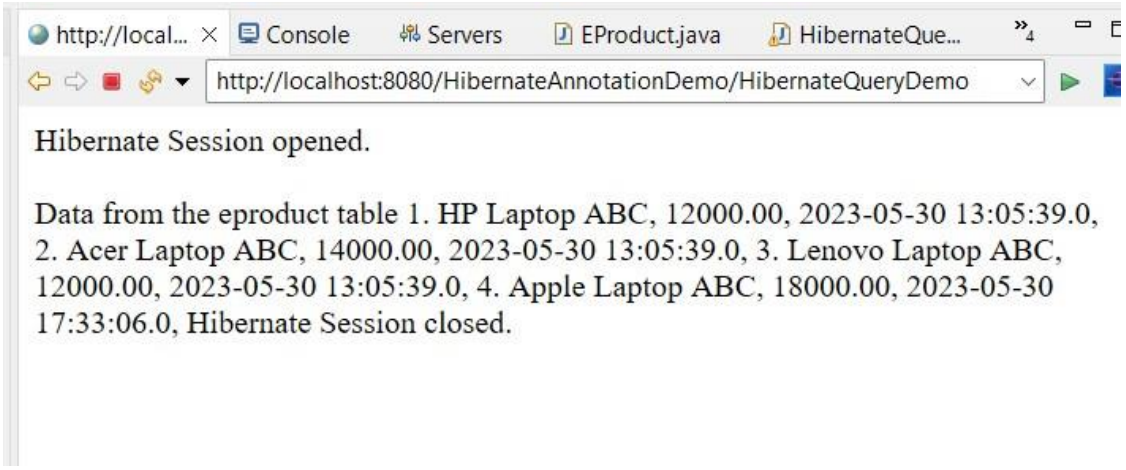
```java
public class EProduct {

	@Id
	@GeneratedValue
	@Column(name="ID")
	private long ID;

	@Column(name="name")
	private String name;

	@Column(name="price")
	private BigDecimal price;

	@Column(name="date_added")
	private Date dateAdded;

	public EProduct() {

	}

	public long getID() {
		return ID;
	}

	public void setID(long iD) {
		ID = iD;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public BigDecimal getPrice() {
		return price;
	}

	public void setPrice(BigDecimal price) {
		this.price = price;
	}

	public Date getDateAdded() {
		return dateAdded;
	}

	public void setDateAdded(Date dateAdded) {
		this.dateAdded = dateAdded;
	}

}
```

✓ **HibernateQueryDemo.java**

```
package com.simpli;

import java.io.*;

import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

import org.hibernate.*;

import com.ecomerce.EProduct;

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

     protected void doGet(HttpServletRequest request,
HttpServletResponse response)
               throws ServletException, IOException {
          PrintWriter out = response.getWriter();
          out.println("<html><body>");

          // STEP 1: Get a Session (connection) from the Session Factory
class
          SessionFactory factory = HibernateUtil.getSessionFactory();

          Session session = factory.openSession();

          out.println("Hibernate Session opened.<br>");

          // STEP 2 execute the HQL commands
          // for now we will only test if the connection is establised
with MySQL server.
          List<EProduct> eproducts = session.createQuery("from
EProduct").list();

          out.println("<br> Data from the eproduct table");
          for(EProduct prod: eproducts) {
               out.println(prod.getID() + ". " + prod.getName() + ", "
          + prod.getPrice() + ", "  + prod.getDateAdded() + ", " );
          }


          session.close();

          out.println("Hibernate Session closed.<br>");

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

}
```

- **Output**


Hibernate Annot...
Console
Servers
Markers
hibernate.cfg.xml
http://localhost:8080/HibernateAnnotationDemo/
Hibernate Annotation Query Demo
Hibernate Query Demo


http://local...
Console
Servers
EProduct.java
HibernateQue...
http://localhost:8080/HibernateAnnotationDemo/HibernateQueryDemo
Hibernate Session opened.
Data from the eproduct table 1. HP Laptop ABC, 12000.00, 2023-05-30 13:05:39.0, 2. Acer Laptop ABC, 14000.00, 2023-05-30 13:05:39.0, 3. Lenovo Laptop ABC, 12000.00, 2023-05-30 13:05:39.0, 4. Apple Laptop ABC, 18000.00, 2023-05-30 17:33:06.0, Hibernate Session closed.