Phase – 2 Practice Project: Assisted Practice

24. Demonstrate Hibernate logging by Log4j.

- **Code**

✓ **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>
```

✓ **log4j.properties**

```
log4j.rootLogger=DEBUG, Appender1,Appender2

log4j.appender.Appender1=org.apache.log4j.ConsoleAppender
log4j.appender.Appender1.layout=org.apache.log4j.PatternLayout
log4j.appender.Appender1.layout.ConversionPattern=%-7p %d [%t] %c %x -
%m%n

log4j.appender.Appender2=org.apache.log4j.FileAppender
log4j.appender.Appender2.File=D:\applog.txt
log4j.appender.Appender2.layout=org.apache.log4j.PatternLayout
log4j.appender.Appender2.layout.ConversionPattern=%-7p %d [%t] %c %x -
%m%n


# Log everything. Good for troubleshooting
log4j.logger.org.hibernate=INFO

# Log all JDBC parameters
log4j.logger.org.hibernate.type=ALL
```

✓ index.html

```
<title>Hibernate Configuration Example</title>
<h3>Hibernate Configuration Example</h3>

<a href="init">Initialize Hibernate</a><br>
```

✓ HibernateUtil.java

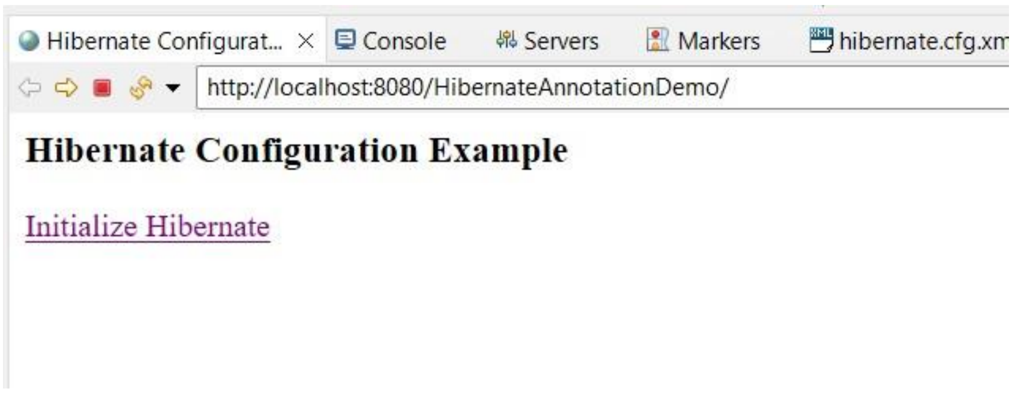
```
package com.simpli;

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;
     }

}
```

✓ InitDemo.java

```
package com.simpli;

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

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```java
import org.hibernate.Session;
import org.hibernate.SessionFactory;

/**
 * Servlet implementation class InitDemo
 */
@WebServlet("/init")
public class InitDemo 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>");

          session.close();

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

          // STEP 2 execute the HQL commands
          // for now we will only test if the connection is established
with MySQL server.


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

     }

}
```

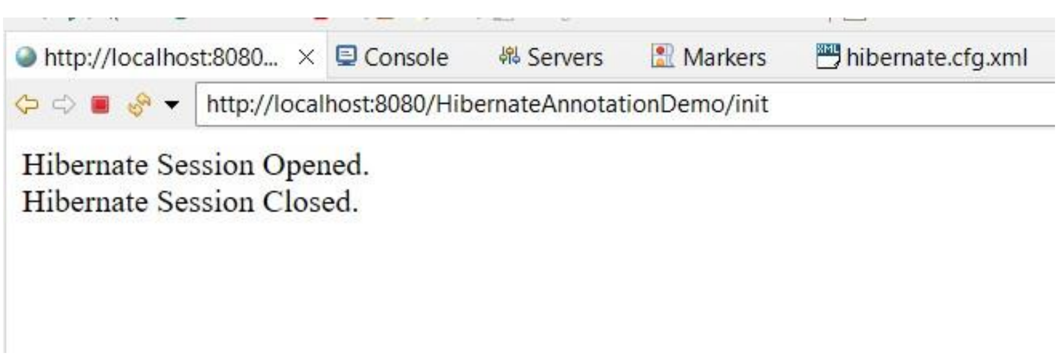
- **Output**



http://localhost:8080/HibernateAnnotationDemo/

Hibernate Configuration Example

Initialize Hibernate



```
http://localhost:8080/HibernateAnnotationDemo/init
```

```
Hibernate Session Opened.
Hibernate Session Closed.
```

- This will generate applog.txt file in the specified path, which contains log files.

applog.txt