

Phase-2 Practice Project: Assisted Practice

1.Demonstrate a project to set up JDBC environment.

DBConnection:

```
package com.ecommerce;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    private Connection connection;

    public DBConnection(String dbURL, String user, String pwd) throws
ClassNotFoundException, SQLException{

        Class.forName("com.mysql.jdbc.Driver");
        this.connection = DriverManager.getConnection(dbURL, user, pwd);
    }

    public Connection getConnection(){
        return this.connection;
    }

    public void closeConnection() throws SQLException {
        if (this.connection != null)
            this.connection.close();
    }
}
```

DemoJDBC:

```
package com.ecommerce;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.CallableStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
```

```

import java.util.Properties;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

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

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DemoJDBC() {
        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 {
            PrintWriter out = response.getWriter();
            out.println("<html><body>");

            InputStream in =
getServletContext().getResourceAsStream("/WEB-INF/config.properties");
            Properties props = new Properties();
            props.load(in);

            DBConnection conn = new DBConnection(props.getProperty("url"),
props.getProperty("userid"), props.getProperty("password"));
            out.println("DB Connection initialized.<br>");

            conn.closeConnection();
            out.println("DB Connection closed.<br>");
        }
    }
}

```

```

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

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

/**
 * @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);
}
}

```

Classpath:

```

<?xml version="1.0" encoding="UTF-8"?>
<classpath>
    <classpathentry kind="src" path="src/main/java"/>
    <classpathentry kind="con"
path="org.eclipse.jdt.launching.JRE_CONTAINER/org.eclipse.jdt.internal.debug.ui.launcher.
StandardVMType/JavaSE-17">
        <attributes>
            <attribute name="module" value="true"/>
        </attributes>
    </classpathentry>
    <classpathentry kind="con"
path="org.eclipse.jst.server.core.container/org.eclipse.jst.server.tomcat.runtimeTarget/Apach
e Tomcat v10.1">
        <attributes>
            <attribute name="owner.project.facets" value="jst.web"/>
        </attributes>
    </classpathentry>
    <classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.web.container"/>
    <classpathentry kind="con" path="org.eclipse.jst.j2ee.internal.module.container"/>
    <classpathentry kind="lib" path="D:/mysql-connector-j-8.0.33.jar"/>

```

```
        <classpathentry kind="output" path="build/classes"/>
</classpath>
```

Proj descr:

```
<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
    <name>JDBCSetup</name>
    <comment></comment>
    <projects>
</projects>
    <buildSpec>
        <buildCommand>
            <name>org.eclipse.jdt.core.javabuilder</name>
            <arguments>
</arguments>
        </buildCommand>
        <buildCommand>
            <name>org.eclipse.wst.common.project.facet.core.builder</name>
            <arguments>
</arguments>
        </buildCommand>
        <buildCommand>
            <name>org.eclipse.wst.validation.validationbuilder</name>
            <arguments>
</arguments>
        </buildCommand>
    </buildSpec>
    <natures>
        <nature>org.eclipse.jem.workbench.JavaEMFNature</nature>
        <nature>org.eclipse.wst.common.modulecore.ModuleCoreNature</nature>
        <nature>org.eclipse.wst.common.project.facet.core.nature</nature>
        <nature>org.eclipse.jdt.core.javanature</nature>
        <nature>org.eclipse.wst.jsdt.core.jsNature</nature>
    </natures>
</projectDescription>
```

Output:

