# SMARTBRIDGE – SUMMER EXTERNSHIP

## MODERN APPLICATION DEVELOPMENT (JAVA SPRING BOOT)

## WEEK 3 ASSIGNMENT

1. **Implement JDBC using java.**

**CODE:**

```
import java.sql.*;

public class JDBCExample {
    public static void main(String[] args) {
        // Database credentials
        String url = "jdbc:mysql://localhost:3306/mydatabase";
        String username = "username";
        String password = "password";

        // Connection and statement variables
        Connection connection = null;
        Statement statement = null;

        try {
            // Step 1: Register the JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Step 2: Open a connection
            connection = DriverManager.getConnection(url, username, password);

            // Step 3: Create a statement
            statement = connection.createStatement();
    // Step 4: Execute a query
    String sql = "SELECT * FROM employees";
    ResultSet resultSet = statement.executeQuery(sql);

    // Step 5: Process the result set
    while (resultSet.next()) {
        int id = resultSet.getInt("id");
        String name = resultSet.getString("name");
        int age = resultSet.getInt("age");
        String department = resultSet.getString("department");

        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Department: " + department);
        System.out.println("--------------------------");
    }

    // Step 6: Clean up resources
    resultSet.close();
    statement.close();
    connection.close();

} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try {
        if (statement != null)
            statement.close();
            if (connection != null)
                connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
  }
}
```

**EXPLANATION:**

This code demonstrates the basic steps involved in JDBC:

- 
- Registering the JDBC driver using Class.forName().
- Opening a connection to the database using DriverManager.getConnection().
- Creating a statement object using connection.createStatement().
- Executing a SQL query using statement.executeQuery().
- Processing the result set using resultSet.next() to iterate over the rows and resultSet.getXXX() methods to retrieve column values.
- Closing the resources (result set, statement, and connection) in the finally block.

JDBC needs to have the MySQL Connector/J library in the project's class path for the code to compile and run successfully.

**OUTPUT:**

Since the JDBC code provided in the previous response is a command-line application, the output will be displayed in the console or terminal window where you run the Java program. The output will show the retrieved data from the "employees" table in the database. The output, as shown, displays the values of each row in the "employees" table, including the ID, name, age, and department. Each row is separated by a line of dashes here.

```
ID: 1

Name: John Doe

Age: 30

Department: IT
```

```
--------------------------
ID: 2
Name: Jane Smith
Age: 28
Department: HR
--------------------------
ID: 3
Name: Mark Johnson
Age: 35
Department: Sales
--------------------------
```