



Assignment 3

create JDBC connection with the database ,verify Data with Database using Selenium and MySQL, store the data into the database.

```
package demo;

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

public class jdbcConnection {

    public static void main(String[] args) throws SQLException {
        // TODO Auto-generated method stub
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/factory", "root",
"root");

        Statement st = con.createStatement();

        ResultSet rs = st.executeQuery("select id, firstName, lastName from
worker");

        while(rs.next()) {
            int id = rs.getInt("id");
            String firstName = rs.getString("firstName");
            String lastName = rs.getString("lastName");
            System.out.println(id + " => " + firstName + " " + lastName);
        }
    }
}
```



A screenshot of the Eclipse IDE interface. The Project Explorer on the left shows a project named 'demo' with a 'src' folder containing 'jdbcConnection.java'. The main editor displays the code for 'jdbcConnection.java'. The code imports 'java.sql.Connection', 'java.sql.DriverManager', 'java.sql.ResultSet', 'java.sql.SQLException', and 'java.sql.Statement'. It defines a 'main' method that connects to a MySQL database, executes a query to select employee details, and prints the results. The Console window at the bottom shows the output of the program: '1 => Jeethendra S R', '2 => Monika B A', and '3 => Shivkumar Panchal'.

```
1 package demo;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8
9
10 public class jdbcConnection {
11
12     public static void main(String[] args) throws SQLException {
13         // TODO Auto-generated method stub
14         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/factory", "root", "root");
15         Statement st = con.createStatement();
16
17         ResultSet rs = st.executeQuery("select id, firstName, lastName from worker");
18
19         while(rs.next()) {
20             int id = rs.getInt("id");
21             String firstName = rs.getString("firstName");
22             String lastName = rs.getString("lastName");
23             System.out.println(id + " => " + firstName + " " + lastName);
24         }
25     }
26 }
27
28
```

A screenshot of the Eclipse IDE interface, similar to the one above, but with additional Selenium imports added to the code. The imports now include 'org.openqa.selenium.By', 'org.openqa.selenium.WebDriver', and 'org.openqa.selenium.firefox.FirefoxDriver'. The 'main' method is updated to create a 'WebDriver' object using 'FirefoxDriver()' and to find an element by its ID using 'driver.findElement(By.xpath(...))'. The rest of the code, including the database connection and query execution, remains the same. The Console window still shows the same output as the previous screenshot.

```
1 package demo;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8
9
10 import org.openqa.selenium.By;
11 import org.openqa.selenium.WebDriver;
12 import org.openqa.selenium.firefox.FirefoxDriver;
13
14 public class jdbcConnection {
15
16     public static void main(String[] args) throws SQLException {
17         // TODO Auto-generated method stub
18         Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/factory", "root", "root");
19         Statement st = con.createStatement();
20
21         ResultSet rs = st.executeQuery("select id, firstName, lastName from worker");
22
23         while(rs.next()) {
24
25             WebDriver driver = new FirefoxDriver();
26             driver.get("https://infosys.in/login");
27             driver.findElement(By.xpath("//*[@id='root']/div/div[1]/div/div[2]/div[2]/div[2]/div[2]/div[1]/div/div/field
28             int id = rs.getInt("id");
29             String firstName = rs.getString("firstName");
30             String lastName = rs.getString("lastName");
31             System.out.println(id + " => " + firstName + " " + lastName);
32         }
33     }
34 }
35
```



