SMARTBRIDGE JAVA BOOTSTRAP

20MIS0346

LAKSHMI G MENON

VIT, VELLORE

ASSIGNMENT -03

JDBC MySQL Connection

To connect to MySQL using JDBC (Java Database Connectivity):

Step 1: Set up the MySQL JDBC driver

- Download the MySQL Connector/J JDBC driver from the MySQL website (https://dev.mysql.com/downloads/connector/j/). Extract the downloaded archive and add the JAR file to your Java project's classpath.

Step 2: Import required classes

- In your Java code, import the necessary JDBC classes to work with the database:

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

Step 3: Define connection parameters

- Set up the connection parameters such as the database URL, username, and password:

```
String url="jdbc:mysql://localhost:3306/mydatabase";
String username = "your-username";
String password = "your-password";
```

- Replace "mydatabase" with the name of your MySQL database, and provide your actual username and password.

Step 4: Establish a connection

- Create a connection object using the **DriverManager.getConnection()** method:

```
Connection connection = null;
```

```
try {
 connection = DriverManager.getConnection(url,
username, password);
System.out.println("Connection successful!");
} catch (SQLException e)
{
System.out.println("Connection failed. Error: " +
e.getMessage());
}
```

Step 5: Handle connection and cleanup

- After you're done with the database operations, close the connection and any other resources:

```
if (connection != null)
{
Try
{
connection.close();
System.out.println("Connection closed.");
} catch (SQLException e)
{
System.out.println("Failed to close connection. Error:
" + e.getMessage());
}
}
```

EXAMPLE

```
import java.sql.*;
```

```
public class JdbcExample {
    public static void main(String[] args) {
        String url =
"jdbc:mysql://localhost:3306/mydatabase";
        String username = "your-username";
        String password = "your-password";

        try {
            Connection connection =
DriverManager.getConnection(url, username, password);
            System.out.println("Connection
successful!");

            DatabaseMetaData metaData =
connection.getMetaData();
            ResultSet resultSet =
metaData.getTables(null, null, null, new
String[]{"TABLE"});

            System.out.println("Tables in the
database:");
            while (resultSet.next()) {
                String tableName =
resultSet.getString("TABLE_NAME");
                System.out.println(tableName);
            }

            resultSet.close();
            connection.close();
```

```
            System.out.println("Connection closed.");
        } catch (SQLException e) {
            System.out.println("Connection failed.
Error: " + e.getMessage());
        }
    }
}
```

In this code, after establishing the connection,

- We obtain the **DatabaseMetaData** object from the connection, which provides information about the database.
- Then, we execute the **getTables** method to retrieve a **ResultSet** containing the table names from the database.
- We iterate over the **ResultSet** and retrieve the value of the "TABLE_NAME" column for each row, which represents the table name.
- We then print out the table names.

OUTPUT:

Connection successful!

Tables in the database:

table1

table2

table3

...

Connection closed.