

Name : Tejaswini Anil Kamble
Email : teju000kamble@gmail.com

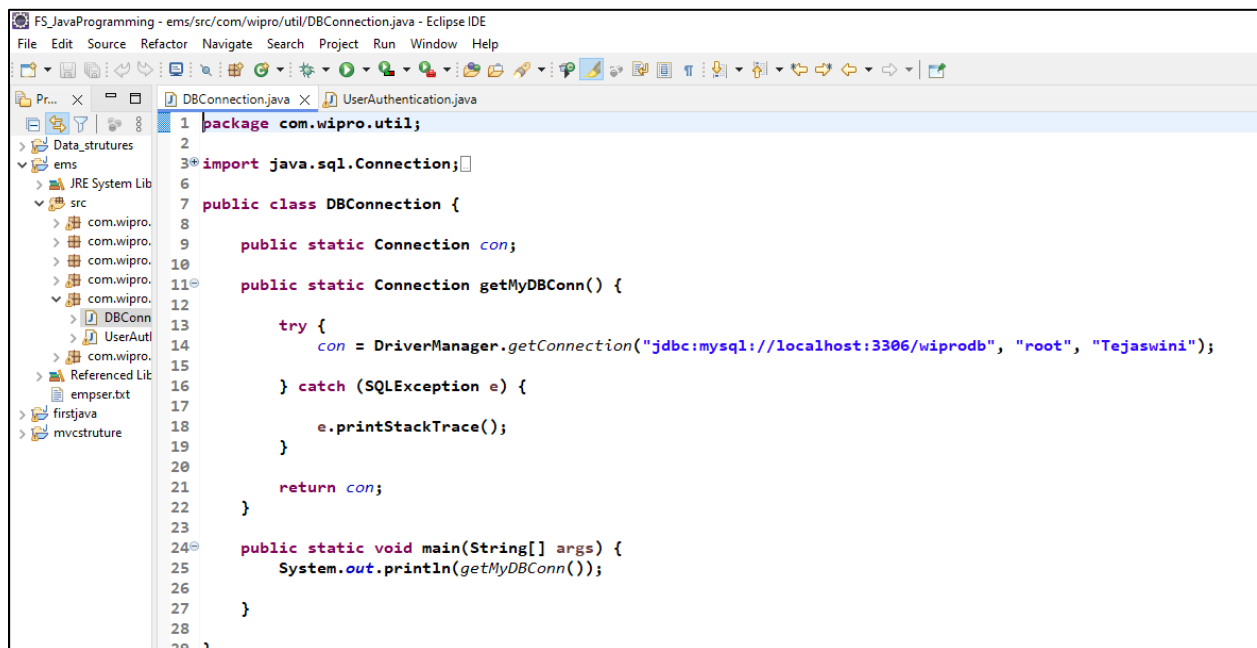
JDBC Assignments : Day 21

Task 1: Establishing Database Connections

Write a Java program that connects to a SQLite database and prints out the connection object to confirm successful connection.

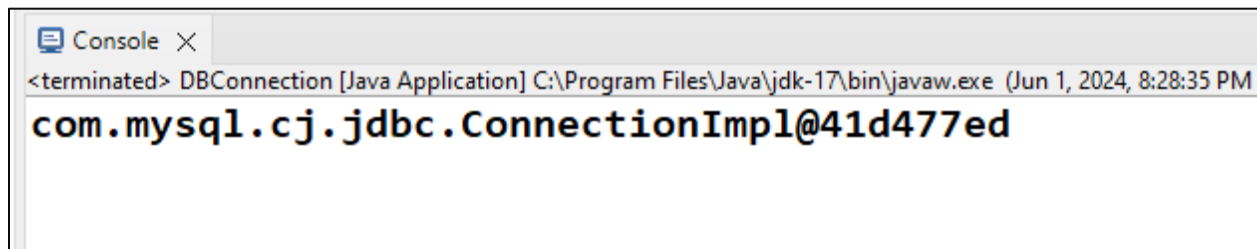
Ans:

Source Code :



```
1 package com.wipro.util;
2
3 import java.sql.Connection;
4
5
6
7 public class DBConnection {
8
9     public static Connection con;
10
11     public static Connection getMyDBConn() {
12
13         try {
14             con = DriverManager.getConnection("jdbc:mysql://localhost:3306/wiprodb", "root", "Tejaswini");
15
16         } catch (SQLException e) {
17
18             e.printStackTrace();
19         }
20
21         return con;
22     }
23
24     public static void main(String[] args) {
25         System.out.println(getMyDBConn());
26     }
27
28
29 }
```

Output :



```
<terminated> DBConnection [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Jun 1, 2024, 8:28:35 PM)
com.mysql.cj.jdbc.ConnectionImpl@41d477ed
```

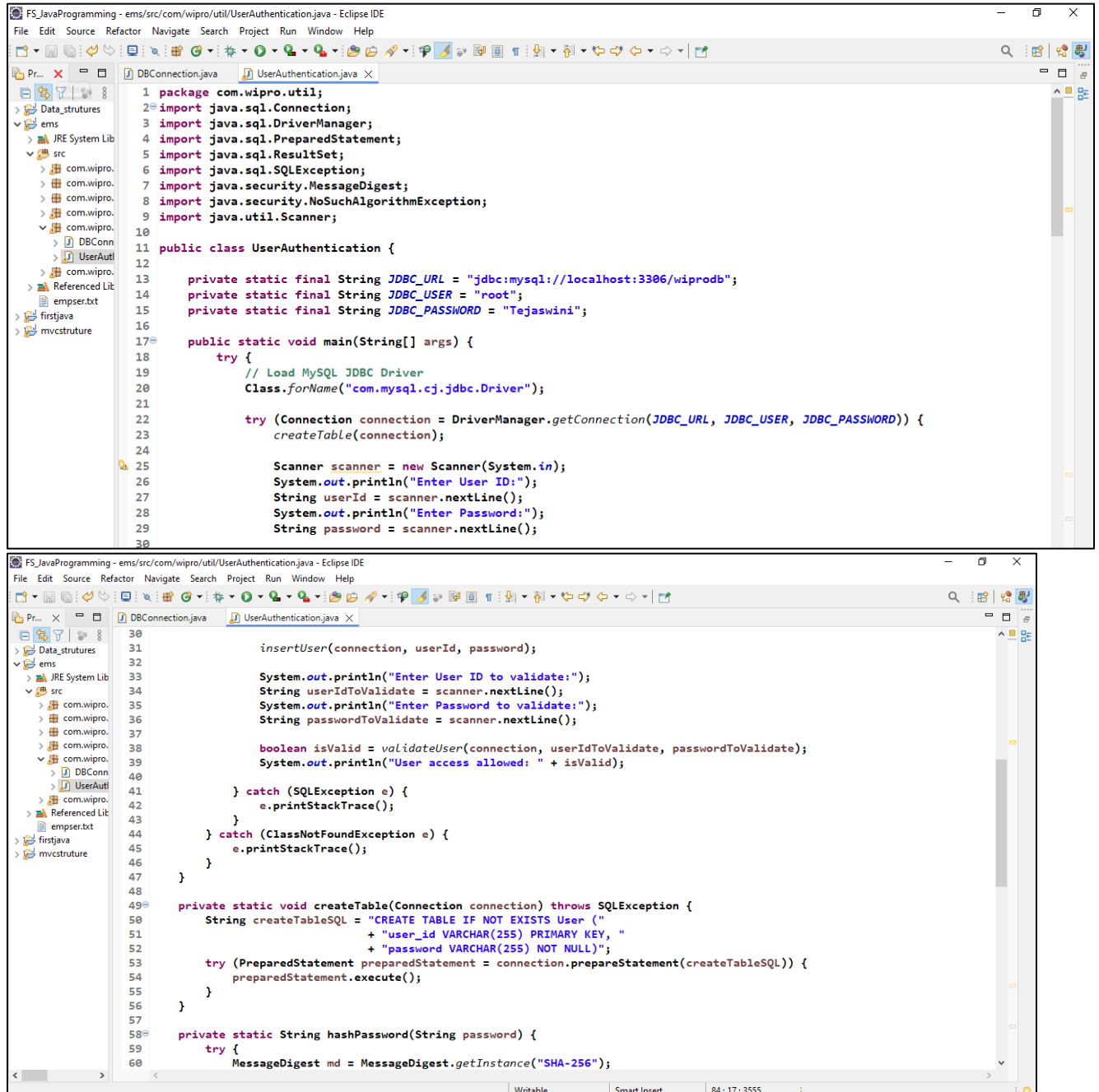
Task 2: SQL Queries using JDBC

Create a table 'User' with a following schema 'User ID' and 'Password' stored as hash format (note you have research on how to generate hash from a

string), accept "User ID" and "Password" as input and check in the table if they match to confirm whether user access is allowed or not.

Ans:

Source Code:



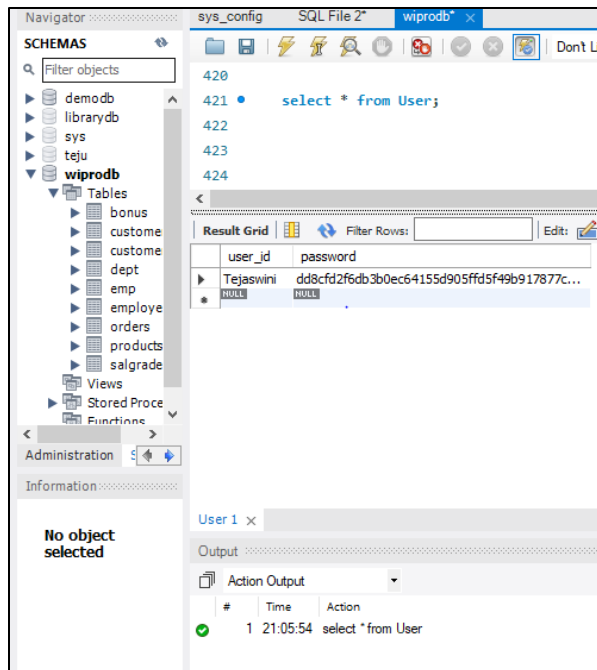
```
1 package com.wipro.util;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.security.MessageDigest;
8 import java.security.NoSuchAlgorithmException;
9 import java.util.Scanner;
10
11 public class UserAuthentication {
12
13     private static final String JDBC_URL = "jdbc:mysql://localhost:3306/wiprodb";
14     private static final String JDBC_USER = "root";
15     private static final String JDBC_PASSWORD = "Tejaswini";
16
17     public static void main(String[] args) {
18         try {
19             // Load MySQL JDBC Driver
20             Class.forName("com.mysql.cj.jdbc.Driver");
21
22             try (Connection connection = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PASSWORD)) {
23                 createTable(connection);
24
25                 Scanner scanner = new Scanner(System.in);
26                 System.out.println("Enter User ID:");
27                 String userId = scanner.nextLine();
28                 System.out.println("Enter Password:");
29                 String password = scanner.nextLine();
30
31                 insertUser(connection, userId, password);
32
33                 System.out.println("Enter User ID to validate:");
34                 String userIdToValidate = scanner.nextLine();
35                 System.out.println("Enter Password to validate:");
36                 String passwordToValidate = scanner.nextLine();
37
38                 boolean isValid = validateUser(connection, userIdToValidate, passwordToValidate);
39                 System.out.println("User access allowed: " + isValid);
40
41             } catch (SQLException e) {
42                 e.printStackTrace();
43             }
44         } catch (ClassNotFoundException e) {
45             e.printStackTrace();
46         }
47     }
48
49     private static void createTable(Connection connection) throws SQLException {
50         String createTableSQL = "CREATE TABLE IF NOT EXISTS User ("
51             + "user_id VARCHAR(255) PRIMARY KEY, "
52             + "password VARCHAR(255) NOT NULL)";
53         try (PreparedStatement preparedStatement = connection.prepareStatement(createTableSQL)) {
54             preparedStatement.execute();
55         }
56     }
57
58     private static String hashPassword(String password) {
59         try {
60             MessageDigest md = MessageDigest.getInstance("SHA-256");
```

```
FS_JavaProgramming - ems/src/com/wipro/util/UserAuthentication.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

DBConnection.java UserAuthentication.java
62     StringBuilder sb = new StringBuilder();
63     for (byte b : hashedPassword) {
64         sb.append(String.format("%02x", b));
65     }
66     return sb.toString();
67 } catch (NoSuchAlgorithmException e) {
68     throw new RuntimeException(e);
69 }
70 }
71
72 private static void insertUser(Connection connection, String userId, String password) throws SQLException {
73     String hashedPassword = hashPassword(password);
74     String insertUserSQL = "INSERT INTO User (user_id, password) VALUES (?, ?)";
75     try (PreparedStatement preparedStatement = connection.prepareStatement(insertUserSQL)) {
76         preparedStatement.setString(1, userId);
77         preparedStatement.setString(2, hashedPassword);
78         System.out.println("Executing query: " + preparedStatement.toString());
79         preparedStatement.executeUpdate();
80     }
81 }
82 private static boolean validateUser(Connection connection, String userId, String password) throws SQLException
83 {
84     String hashedPassword = hashPassword(password);
85     String selectUserSQL = "SELECT password FROM User WHERE user_id = ?";
86     try (PreparedStatement preparedStatement = connection.prepareStatement(selectUserSQL))
87     {
88         preparedStatement.setString(1, userId);
89         try (ResultSet resultSet = preparedStatement.executeQuery())
90         {
91             if (resultSet.next())
92             {
93                 String storedHashedPassword = resultSet.getString("password");
94                 return storedHashedPassword.equals(hashedPassword);
95             } else {
96                 return false;
97             }
98         }
99     }
100 }
101 }
```

Output:

```
Console X
<terminated> UserAuthentication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Jun 11, 2024, 8:52:26 PM - 8:53:34 PM) [pid: 3060]
Enter User ID:
Tejaswini
Enter Password:
Teju@123
Executing query: com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO User (user_id, password) VALUES ('Te
Enter User ID to validate:
Tejaswini
Enter Password to validate:
Teju@123
User access allowed: true
```



Task 3: PreparedStatement

Modify the SELECT query program to use PreparedStatement to parameterize the query and prevent SQL injection.

Ans:

Source Code:

```

private static boolean validateUser(Connection connection, String userId, String password) throws SQLException
{
    String hashedPassword = hashPassword(password);
    String selectUserSQL = "SELECT password FROM User WHERE user_id = ?";
    try (PreparedStatement preparedStatement = connection.prepareStatement(selectUserSQL))
    {
        preparedStatement.setString(1, userId);
        try (ResultSet resultSet = preparedStatement.executeQuery())
        {
            if (resultSet.next())
            {
                String storedHashedPassword = resultSet.getString("password");
                return storedHashedPassword.equals(hashedPassword);
            } else {
                return false;
            }
        }
    }
}

```

Output:

```
<terminated> UserAuthentication [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Jun 1, 2024, 9:17:40 PM – 9:18:20 PM) [pid: 14708]
Enter User ID:
Tejaswini06
Enter Password:
Teju@1234
Executing query: com.mysql.cj.jdbc.ClientPreparedStatement: INSERT INTO User (user_id, password)
Enter User ID to validate:
Tejaswini06
Enter Password to validate:
Teju@1234
User access allowed: true
```

```
421 • select * from User;
422
423
424
```

	user_id	password
▶	Tejaswi	4a99519525764894e8c89e33f6550121ef7b60c...
	Tejaswini	dd8cfd2f6db3b0ec64155d905ffd5f49b917877c...
	Tejaswini06	4f6fc9f518e974a2a6ffe88e1a06b9bedea87e05...
*	NULL	NULL