Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering
Semester: (Spring, Year:2024), B.Sc. in CSE (Day)

LAB REPORT NO #04
Course Title: Database Lab
Course Code: CSE 210 Section: 221_D9

Experiment Name: Implementation of Login-signup in NetBeans using Java swing with MySQL Database Connection.

Student Details

Name		ID
1.	Jahidul Islam	221002504

Lab Date : 23 – 03 – 2024
Submission Date : 30 – 03 – 2024
Course Teacher's Name : Md. Nazmus Shakib

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

1. TITLE OF THE LAB EXPERIMENT:

Implementation of Login-signup in NetBeans using Java swing with MySQL Database Connection.

2. OBJECTIVES:

After complementing this lab experiment, we will gain practical knowledge and the outcomes of this experiment are

1. Database creation and Data Insertion.
2. Database creation and Insert Data in NetBeans project using jdbc Driver.
3. Hands on experience on Java Swing Design Interface.
4.

3. PROCEDURE:

To successfully complete the outcomes, we have done the following.

1. **Database Design:** Accounts
2. **Table Cration:** We need only one Table for Registration for keeping the record of the account holder.

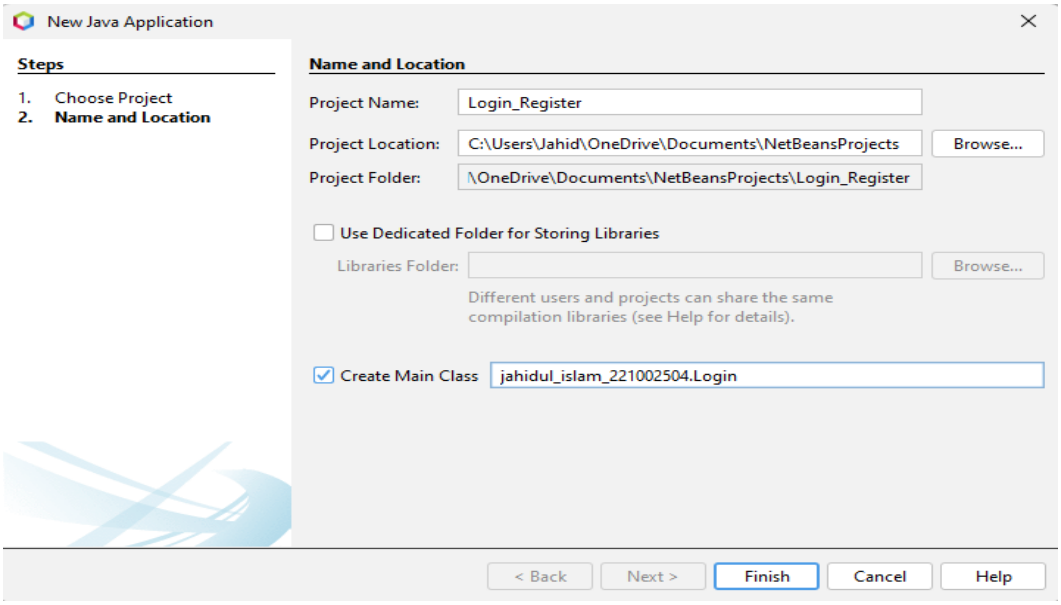
    a. Registration table:

```
CREATE TABLE registration(
    id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL,
    username VARCHAR(50) NOT NULL,
    PASSWORD VARCHAR(50) NOT NULL,
    gender VARCHAR(100),
    country VARCHAR(100),
    phone_number VARCHAR(20),
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Figure 1: registration table's fields datatypes

3. **Project Creation:** Login_Register

   a. We first created a project named **Login_Register** using NetBeans Java Application.



Firgur 2: Project Creation.

4. **Login Form:**

   a. **Login Page Code:**

   We used a userLogin() function to separate the login code for better readability.

```
private void userLogin(String email, String password) {
    Connection dbconn = DBConnection.connectDB();
    try{
        PreparedStatement st;
        st = dbconn.prepareStatement("Select * from registration Where email = ? AND password = ?;");
        st.setString(1, email);
        st.setString(2, password);
        ResultSet result = st.executeQuery();

        if(result.next()){
            dispose();
            HomePage home = new HomePage();
              home.setTitle("HomePage");
            home.setVisible(true);
        }else{
            System.out.println("eamil : " + email);
            System.out.println("pass : " + password);
            JOptionPane.showMessageDialog(this, "Email/Pass not found", "Error", JOptionPane.ERROR_MESSAG
        }

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

Figure 03: Login Page Code. Included Eror checking also.

b. Registration Page code:

```
private void userRegister(String fullName, String Email, String userName, String Password, String Gender, Stri
    Connection dbconn = DBConnection.connectDB();
    try{
        PreparedStatement st;
        st = dbconn.prepareStatement("INSERT INTO registration Values(?, ?, ?, ?, ?, ?, ?)");

        st.setString(1, fullName);
        st.setString(2, Email);
        st.setString(3, userName);
        st.setString(4, Password);
        st.setString(5, Gender);
        st.setString(6, Country);
        st.setString(7, phoneNumber);

        int result = st.executeUpdate();

        JOptionPane.showMessageDialog(this, "Successfully Registered...", "Success", JOptionPane.ERROR_MESSAGE

    }catch(SQLException ex){
/           Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
        System.err.println("Error opening HomePage: " + ex.getMessage());
    }
}
```

Figure 04: Registration Page.

4. IMPLEMENTATION:

a. Button: btnLogin
   The following code is triggered when **login** button is clicked.
   It checks the Text fields and shows error or otherwise call userLogin() function.
   It takes email and password parameters given by the user.

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String email = Email_Login.getText();
    String password = Password_Login.getPassword().toString();
    if(email.isEmpty() || password.isEmpty()){
        JOptionPane.showMessageDialog(this, "Email/Pass empty", "Error", JOptionPane.ERROR_MESSAGE);
    }else{
        //lg starts here
        userLogin(email, password);


    }
```

Figure 5: Extract the given text in the login form field in btnLogin Action Event code.

.

b. button: signup

This signup button navigates to the registration page.

```
    private void btnRegisterActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String fullName = full_name.getText();
        String Email = email.getText();
        String userName = username.getText();
//          String Password = password.getText();
        String Password = String.valueOf(password.getPassword());
        String Gender = gender.getText();
        String Country = country.getText();
        String phoneNumber = phone_number.getText();

        if(Email.isEmpty() || Password.isEmpty() || userName.isEmpty() || fullName.isEmpty()){
            JOptionPane.showMessageDialog(this, "Email/Pass/Name/User Name is empty", "Error", JOptionPane.ERROR_ME
        }else{
            //lg starts here
            userRegister(fullName, Email,userName, Password, Gender, Country, phoneNumber);
        }
```

Figure 6: Extract the text in the registration form field the passes them to userRegister() method.

5. OUTPUT

a. **Design:** login page

Figure 07: Login form with signup btn navigation.

b. Design: Registration Form

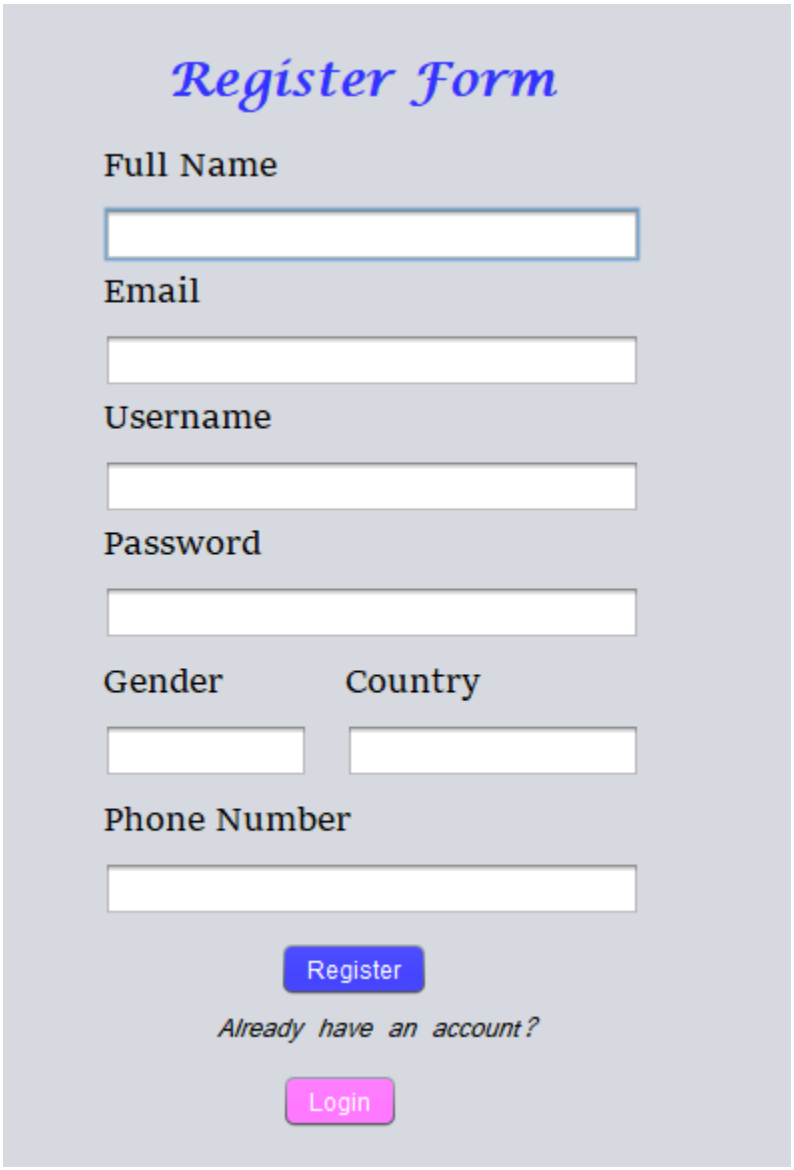


Figure 08: Register form with login button navigation.

c. **Design:** HomePage

- After successful login we navigate to the homepage. I included a exit/signout opttioon for going back to login page.

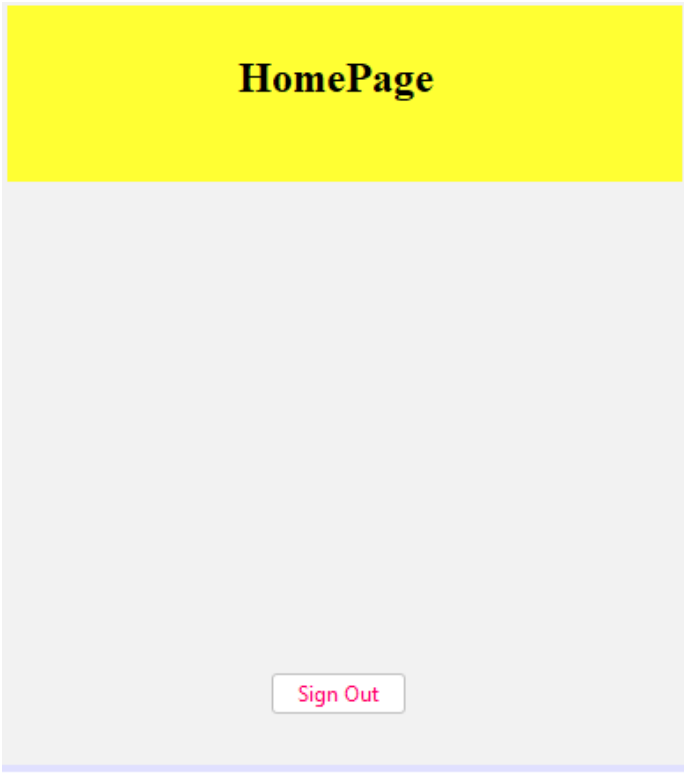


Figure 09: Homepage after login with a sign out option.

6. ANALYSIS AND DISCUSSION:

Despite challenges, I found the implementation efficient, with scope for future enhancements.

a. Logging in:

Here email: **admin** and password: **admin** in the database but somehow password get converted to encrypted string. Due to this we couldn't login successfully.

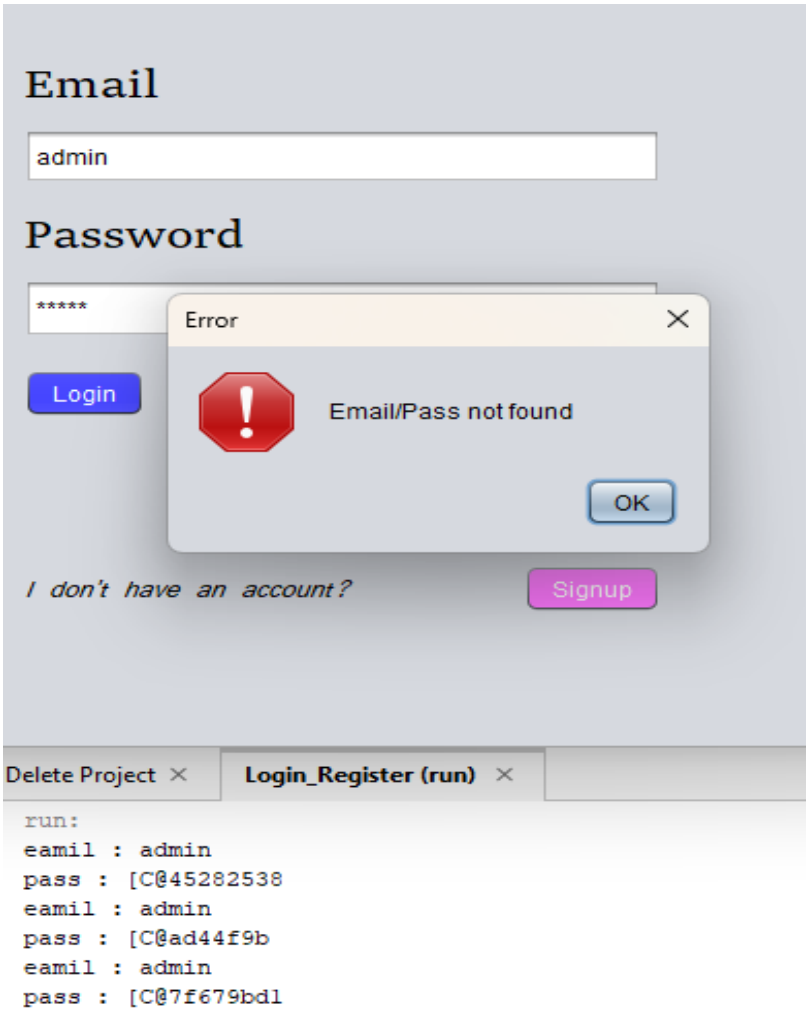


Figure 10: unsuccessful login reason and debugging to find solution.

Solved it by changing the following password string in btnLoginAction

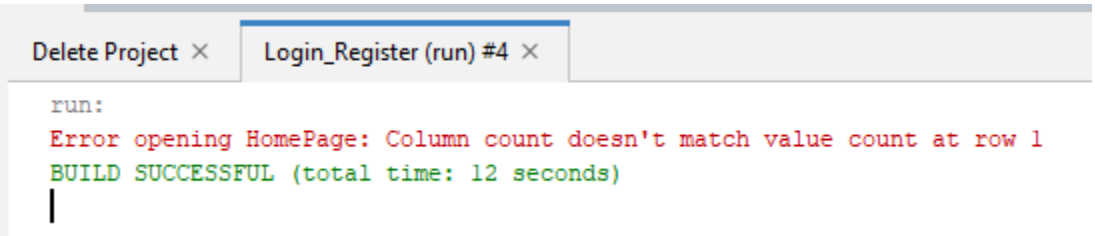
```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String email = Email_Login.getText();
    String password = String.valueOf(Password_Login.getPassword());
```

Figure 11: Code Segment represents formal password extracting method

b. Register page: User Input data insertion in DB:

1st I faced the following error. This is because I didn't add column name in the values SQL query code.

- **Error:**



- **Source of the error:**

```
try{
    PreparedStatement st;
    st = dbconn.prepareStatement("INSERT INTO registration Values(?, ?, ?, ?, ?, ?, ?)");
```

- **Solution:**

  st = dbconn.prepareStatement("INSERT INTO registration (full_name, email, username, password, gender, country, phone_number) Values(?, ?, ?, ?, ?, ?, ?)");

We had to debug some more error along the way. After a long hour of trying I finally able to complete the whole lab report.

7. SUMMARY:

The lab experiment is successfully completed on creating and inseting data in the database with one tables: "registration".
We successfully implemented the whole insertion and Select query for login in in GUI. using netbeans.

7. SOURCE CODE:

- https://github.com/jahidulzaid/login_register-NetBeans.git