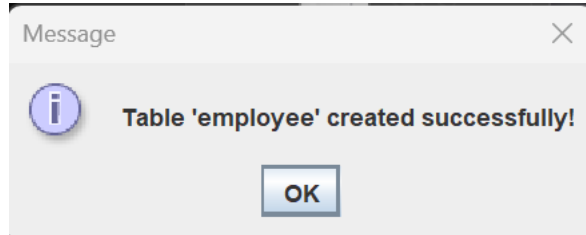
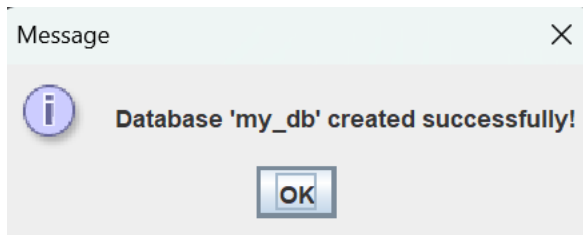






Lab 7



Lab 8

```
"C:\Program Files\Java\jdk-21\bin\java.exe"  
A new user record was inserted successfully!  
  
Process finished with exit code 0
```

				id	username	email	mobile	address
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	binita	binita@yahoo.com	9813169135	Lainchour

CODE

```
package JDBC_Program;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class JDBCSwing {
    // Constructor to handle login functionality by verifying
    // the username/email and password from the database
    public static boolean validateLogin(String email, String
password) {
        String db_url = "jdbc:mysql://localhost:3306/my_db"; //
Database URL
        String db_user = "root"; // Database username
        String db_password = ""; // Database password

        try {
            // Load and register the driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Create a connection
            Connection conn =
DriverManager.getConnection(db_url, db_user, db_password);

            // SQL query for checking user credentials based on
            // either username or email
            String sql = "SELECT * FROM users WHERE email = ?
AND password = ?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, email);
            ps.setString(2, password);
            ResultSet rs = ps.executeQuery();

            // Check if user exists and the credentials are
            // correct
            if (rs.next()) {
                rs.close();
            }
        }
    }
}
```

```

        ps.close();
        conn.close();
        return true; // User found and credentials
match
        } else {
            rs.close();
            ps.close();
            conn.close();
            return false; // User not found or incorrect
credentials
        }

        } catch (ClassNotFoundException e) {
            System.out.println("JDBC Driver not found: " +
e.getMessage());
        } catch (SQLException e) {
            System.out.println("Database error: " +
e.getMessage());
        }

        return false; // Return false in case of any error
    }
}

class LoginForm extends JFrame {
    // Swing UI components for login form
    JLabel emailLabel = new JLabel("Email: ");
    JTextField emailField = new JTextField();

    JLabel passwordLabel = new JLabel("Password: ");
    JPasswordField passwordField = new JPasswordField();

    JButton loginBtn = new JButton("Login");
    JTextArea statusText = new JTextArea("Status: ");

    public LoginForm() {
        setLayout(null);
        setTitle("Login Form");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
    }
}

```

```

emailLabel.setBounds(40, 50, 160, 32);
emailField.setBounds(120, 50, 180, 32);

passwordLabel.setBounds(40, 110, 160, 32);
passwordField.setBounds(120, 110, 180, 32);

loginBtn.setBounds(140, 160, 120, 30);
loginBtn.setBackground(Color.BLUE);
loginBtn.setForeground(Color.WHITE);
loginBtn.setFocusPainted(false);

statusText.setBounds(40, 210, 300, 80);
statusText.setEditable(false);
statusText.setLineWrap(true);
statusText.setWrapStyleWord(true);
statusText.setFont(new Font("Verdana", Font.PLAIN, 15));

add(emailLabel);
add(passwordLabel);
add(loginBtn);
add(statusText);

add(emailField);
add(passwordField);

// Adding action listener for login button
loginBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            // Get email and password from the text
fields
            String email = emailField.getText().trim();
            String password = new
String(passwordField.getPassword()).trim();

            // Validate if fields are empty
            if (email.isEmpty() || password.isEmpty()) {
                statusText.setForeground(Color.RED);
                statusText.setText("Please ensure no

```

```

field is empty.");
        } else {
            // Check the login credentials by
calling validateLogin method
            boolean isLoginValid =
JDBCSwing.validateLogin(email, password);

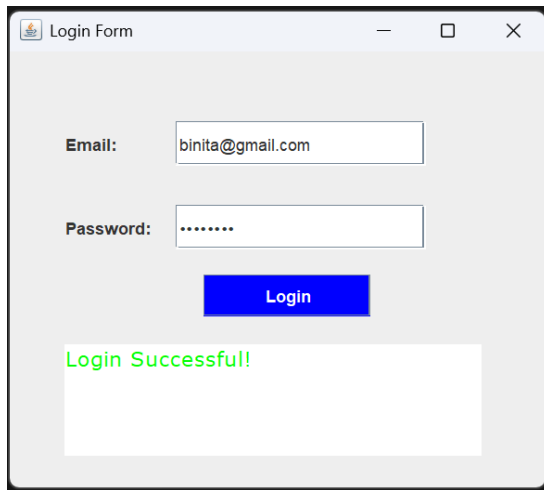
            if (isLoginValid) {

statusText.setForeground(Color.GREEN);
                statusText.setText("Login
Successful!");
            } else {
                statusText.setForeground(Color.RED);
                statusText.setText("Login
Unsuccessful. Invalid credentials.");
            }
        }
    } catch (Exception ex) {
        System.out.println("Error during login: " +
ex.getMessage());
    }
}

public static void main(String[] args) {
    // Create and show the login form
    LoginForm loginForm = new LoginForm();
    loginForm.setVisible(true);
}
}

```

OUTPUT



CONCLUSION

This program demonstrates how to create a simple login page using Java Swing for the graphical user interface and JDBC to interact with a MySQL database for user authentication. It validates the user credentials entered on the Swing form by checking them against the table in the MySQL database. The program provides feedback to the user based on whether the login was successful or not.

CODE

```
package JDBC_Program;

import java.sql.*;

class DbModel {

    public DbModel(String Name, String Password, String Email,
String Address) {

        String url = "jdbc:mysql://localhost:3306/my_db";

        String user = "root";

        String password = "";

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            Connection connection =

DriverManager.getConnection(url, user, password);

            String sql= ("INSERT INTO user (name, password,
email, address) VALUES (?, ?, ?, ?);");

            PreparedStatement ps =

connection.prepareStatement(sql);

            ps.setString(1, Name);

            ps.setString(2, Password);

            ps.setString(3, Email);

            ps.setString(4, Address);

            int rows=ps.executeUpdate();

            if(rows>0) {
```

```

        System.out.println("Data already exists.");
    }else{
        System.out.println("Data sent to database.");
    }

    ps.close();

    connection.close();

    System.out.println("Problem didn't arised.");
} catch (ClassNotFoundException e) {

    System.out.println("JDBC Driver not found: " +
e.getMessage());

    } catch (SQLException e) {

        System.out.println("Database error: " +
e.getMessage());

    }

}

}

class RegisterForm extends JFrame {

    JLabel name = new JLabel("Name : ");

    JTextField nameField = new JTextField();

    JLabel email = new JLabel("Email : ");

    JTextField emailField = new JTextField();

```



```
JLabel address = new JLabel("Address : ");

JTextField addField = new JTextField();


JLabel password = new JLabel("Password : ");

JPasswordField passField = new JPasswordField();


JButton btnSubmit = new JButton("submit");

JTextArea outputText = new JTextArea("Status : ");


public RegisterForm() {

    setLayout(new FlowLayout());

    setLayout(null);

    setTitle("User Register Page");

    setSize(400, 550);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setLocationRelativeTo(null);


    name.setBounds(20, 50, 60, 32);

    nameField.setBounds(100, 50, 180, 32);

    nameField.setBorder(new Border());


    email.setBounds(20, 110, 60, 32);

    emailField.setBounds(100, 110, 180, 32);
```

```
address.setBounds(20,170,120,32);

addField.setBounds(100,170,180,32);


password.setBounds(20,230,120,32);
passField.setBounds(100,230,180,32);


btnSubmit.setBounds(110,290,100,32);
btnSubmit.setBackground(Color.blue);
btnSubmit.setForeground(Color.WHITE);
btnSubmit.setFocusPainted(false);


outputText.setBounds(40,350,280,130);
outputText.setEditable(false);
outputText.setLineWrap(true);
outputText.setWrapStyleWord(true);
outputText.setFont(new Font("Verdana",Font.PLAIN,15));
outputText.setBackground(Color.);


add(name);

add(email);

add(address);

add(password);

add(btnSubmit);

add(outputText);
```

```
add(nameField);

add(emailField);

add(addField);

add(passField);


btnSubmit.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        try {

            String Name =

(nameField.getText()).toLowerCase();

            String Password = new

String(passField.getPassword());

            String Email =

(emailField.getText()).toLowerCase();

            String Address =

(addField.getText()).toLowerCase();


            if (!Name.isEmpty() && !Email.isEmpty() &&

!Address.isEmpty() && !Password.isEmpty()){
```

```

        new DbModel(Name, Password, Email,
Address);

        outputText.setForeground(Color.BLACK);
        outputText.setText("Data successfully
sent to database.");

        outputText.setText("Entered data
are:\n"+ " Name : "+ Name +" \n Email : "+Email+" \n Address :
"+Address);

        System.out.println("Entered data
are:\n\n"+ "Name : "+ Name +" \nEmail : "+Email+" \nAddress :
"+Address);

    }else {

        outputText.setForeground(Color.RED);
        outputText.setText("Please look field is
empty !!!");

    }

    } catch (Exception ex) {

        System.out.println("Error on submission: \t"
+ ex.getMessage());

    }

}

});

```

```

    }
}

public class Signup {

    public static void main(String[] args) {

        RegisterForm loginForm = new RegisterForm();

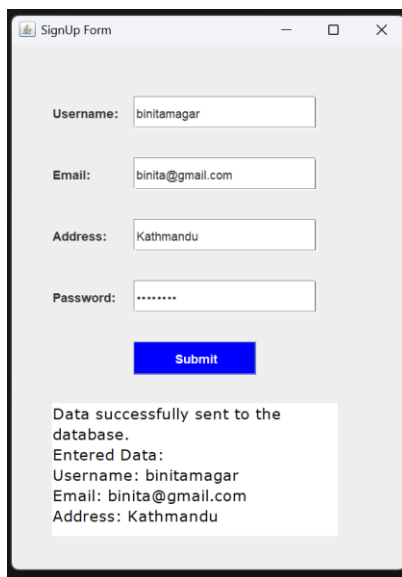
        loginForm.setVisible(true);

    }

}

```

OUTPUT



SignUp Form

Username: binitamagar

Email: binita@gmail.com

Address: Kathmandu

Password:

Submit

Data successfully sent to the database.
Entered Data:
Username: binitamagar
Email: binita@gmail.com
Address: Kathmandu

CONCLUSION

This program demonstrates how to use JDBC to register a user dynamically by taking input from the user and storing the data in a MySQL database. By using PreparedStatement, the program ensures the data is inserted securely and prevents SQL injection attacks. This approach is useful in real-world applications where user registration is required, and it ensures that data is handled efficiently and securely.

