

# **JAVA LAB ASSIGNMENT**

**Name : VISHAKHA JOSHI**

**PRN : 22070126132**

**CLASS: AIML- B3**

**Github Link**

**<https://github.com/Vishakhajoshi24/JAVA-CODE>**

# Java\_Assignment\_8\_10

## Assignment.java

```
package application;

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;

import java.sql.*;

public class Assignment extends Application {

    public static void main(String[] args) {
        launch(args);
    }

    @Override
    public void start(Stage stage) {
        stage.setTitle("Employee Management System");

        stage.setWidth(500);
        stage.setHeight(500);
        stage.setResizable(false);

        VBox mainLayout = new VBox(10);
        mainLayout.setPadding(new Insets(10));
```

```

        TextField employeeIDField = new TextField();
employeeIDField.setPromptText("Employee ID");
TextField nameField = new TextField();
nameField.setPromptText("Name");           TextField
ageField = new TextField();
ageField.setPromptText("Age");           TextField
emailField = new TextField();
emailField.setPromptText("Email");           TextField
departmentField = new TextField();
departmentField.setPromptText("Department");

        Button createTableButton = new Button("Create Employee
e Table");           Button deleteTableButton = new Button("Delete
Employee e Table");           Button registerButton = new
Button("Register Employee e");           Button
viewEmployeesButton = new Button("View Employees");
Button updateButton = new Button("Update Employee Info");

        createTableButton.setOnAction(event -> createEmployee
Table());
        deleteTableButton.setOnAction(event ->
deleteEmployee Table());
registerButton.setOnAction(event -> registerEmployee(
employeeIDField.getText(),
nameField.getText(),           ageField.getText(),
emailField.getText(),
departmentField.getText()           ));
viewEmployeesButton.setOnAction(event -> viewEmployee s());

```

```

        updateButton.setOnAction(event -> updateEmployeeInfo
(employeeIDField.getText(),
nameField.getText(),                ageField.getText(),
emailField.getText(),
departmentField.getText()));

        mainLayout.getChildren().addAll(
employeeIDField, nameField, ageField, emailField,
departmentField,
                createTableButton, deleteTableButton, registerButton, viewEmployeesButton, updateButton
                );

        Scene scene = new Scene(mainLayout, 400,
300);                stage.setScene(scene);
stage.show();        }

        private void createEmployeeTable() {                try
(Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/assignment_8_10", "root", "002cd592"));

                Statement statement = con.createStatement() {
String sql = "CREATE TABLE IF NOT EXISTS employee s (" +
"id INT PRIMARY KEY," +
                "name VARCHAR(100) NOT NULL," +
                "age INT NOT NULL," +
                "email VARCHAR(100) NOT NULL," +
                "department VARCHAR(100) NOT NULL" +
                ") ";

                statement.executeUpdate(sql);
System.out.println("Employee table created!");

```

```
} catch (SQLException e) {
```

```

        e.printStackTrace();
    }
}

private void deleteEmployeeTable() {
    try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/assignment_8_10", "root", "002cd592"));

        Statement statement = con.createStatement() {
String sql = "DROP TABLE IF EXISTS employees";

        statement.executeUpdate(sql);
System.out.println("Employee table deleted!");
        }
    catch (SQLException e) {
        e.printStackTrace();
    }
}

private void registerEmployee(String id, String name, String age, String email, String department) {
    if (!isTableExists()) {
        displayError("Error: Employee table does not exist!");
        return;
    }

    try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/assignment_8_10", "root", "002cd592"));

        PreparedStatement statement = con.prepareStatement(

            "INSERT INTO employees (id, name, age, email, department) VALUES (?, ?, ?, ?, ?)") {

            if (employeeExists(con, id)) {
displayError("Error: Employee ID already exists");
            }
        }
    }
}

```

```

ts!");
return;                                }

        statement.setInt(1, Integer.parseInt(id));
statement.setString(2, name);
statement.setInt(3, Integer.parseInt(age));
statement.setString(4, email);
statement.setString(5, department);

        statement.executeUpdate();

        System.out.println("Employee registered successfully.");

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private boolean employeeExists(Connection con, String id)
throws SQLException {
    PreparedStatement statement =
con.prepareStatement("S ELECT id FROM employees WHERE id =
?");

    statement.setInt(1, Integer.parseInt(id));
ResultSet resultSet = statement.executeQuery();

```

```
return resultSet.next(); // Returns true if ID exist s,  
false otherwise    }
```

```
    private boolean isTableExists() {          try (Connection  
con = DriverManager.getConnection("jd  
bc:mysql://localhost:3306/assignment_8_10", "root", "002cd59  
2")) {          DatabaseMetaData metaData =  
con.getMetaData();          try (ResultSet resultSet =  
metaData.getTables(nul
```



```

l, null, "employees", null)) {
return resultSet.next();
} catch (SQLException e) {
    e.printStackTrace();
return false;
}

private void displayError(String errorMessage) {
Label errorLabel = new Label(errorMessage);

    Stage errorStage = new Stage();
errorStage.setTitle("Error");
errorStage.setScene(new
Scene(new StackPane(errorLabel), 300, 100));
errorStage.show();

private void viewEmployees() {
    StringBuilder
employeeDetails = new StringBuilder();

    if (!isTableExists()) {
displayError("Error: Employee table does not exist!");
return;
}

    try (Connection con = DriverManager.getConnection("jdbc:
mysql://localhost:3306/assignment_8_10", "root", "002cd59
2");

        Statement statement = con.createStatement();
        ResultSet resultSet = statement.executeQuery("SE
LECT * FROM employees")) {

        employeeDetails.append("Registered Employees:\n

```



```

\n");

        while (resultSet.next()) {
employeeDetails.append("ID: ").append(resultSet
et.getInt("id")).append("\n");
            employeeDetails.append("Name:
").append(resultSet.getString("name")).append("\n");
            employeeDetails.append("Age:
").append(resultSet.getInt("age")).append("\n");
employeeDetails.append("Email: ").append(resu
ltSet.getString("email")).append("\n");
            employeeDetails.append("Department: ").append
(resultSet.getString("department")).append("\n");
employeeDetails.append("-----\n");
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }

    TextArea textArea = new TextArea(employeeDetails.toSt
ring());
    textArea.setEditable(false);

    Stage employeeDetailsStage = new Stage();
employeeDetailsStage.setTitle("Registered Employee s");
employeeDetailsStage.setScene(new Scene(new StackPane
(textArea), 400, 300));
employeeDetailsStage.show();
}

    private void updateEmployeeInfo(String id, String name,
String age, String email, String department) {
        if
(!isTableExists()) {

```

```

        displayError("Error: Employee table does not exist!");
        return;
    }

    try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/assignment_8_10", "root", "002cd592");
        PreparedStatement statement = con.prepareStatement(
            "UPDATE employees SET name = ?, age = ?, email = ?, department = ? WHERE id = ?")) {

        statement.setString(1, name);
        statement.setInt(2, Integer.parseInt(age));
        statement.setString(3, email);
        statement.setString(4, department);
        statement.setInt(5, Integer.parseInt(id));

        int rowsUpdated = statement.executeUpdate();

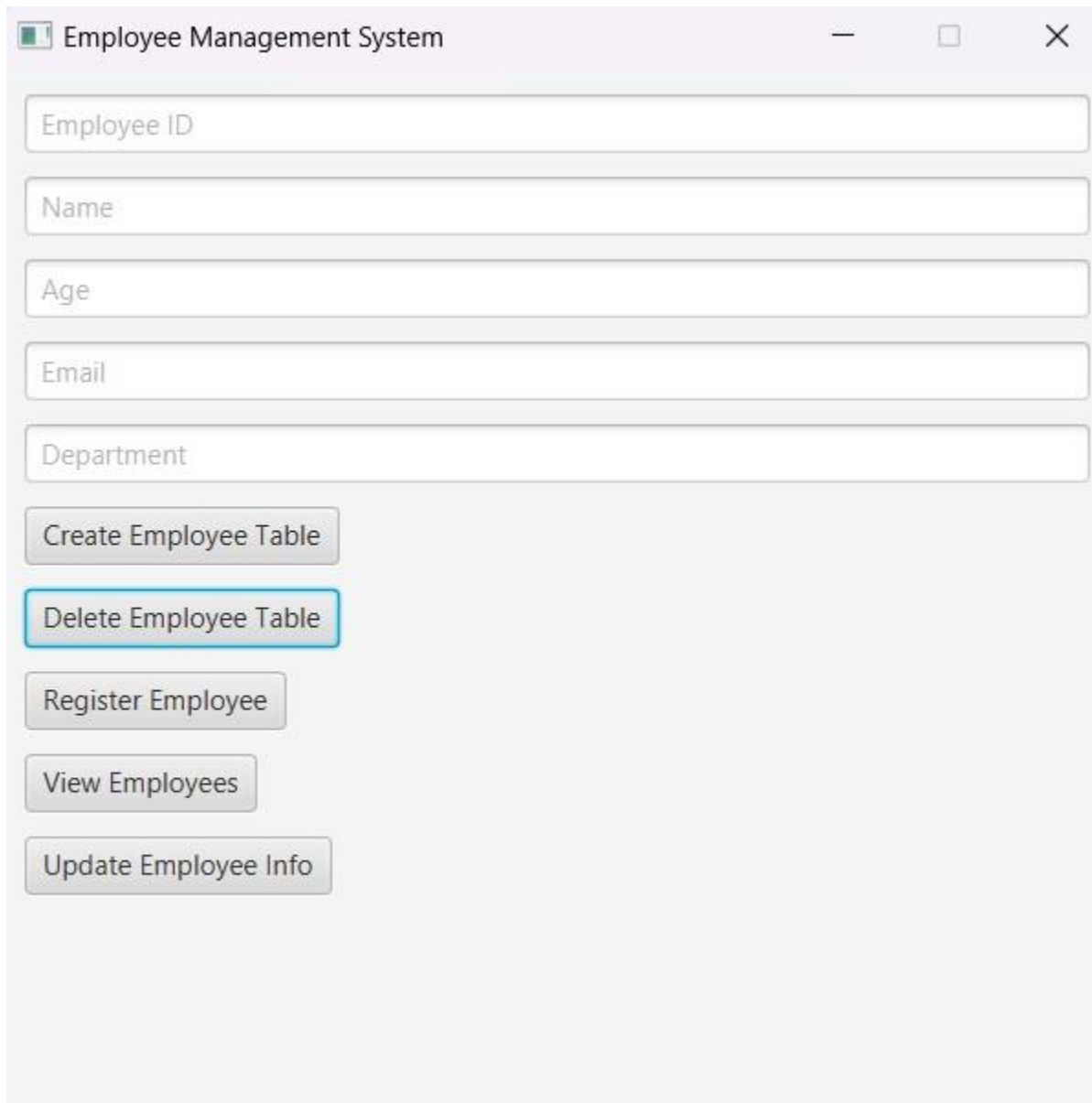
        if (rowsUpdated > 0) {
            System.out.println("Employee information updated successfully.");
        } else {
            System.out.println("No employee found with ID: " + id);
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

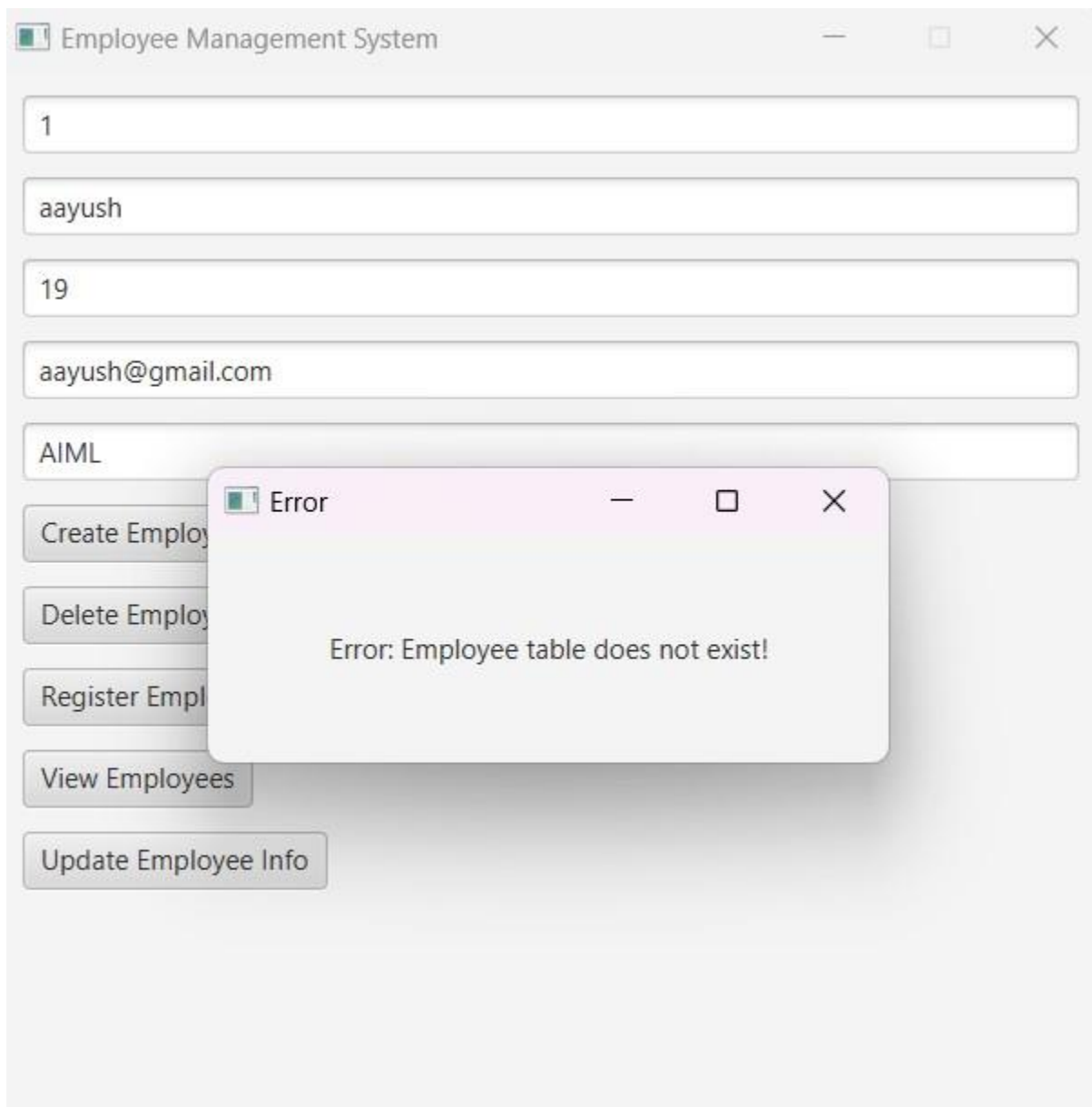
# Output

Front end:

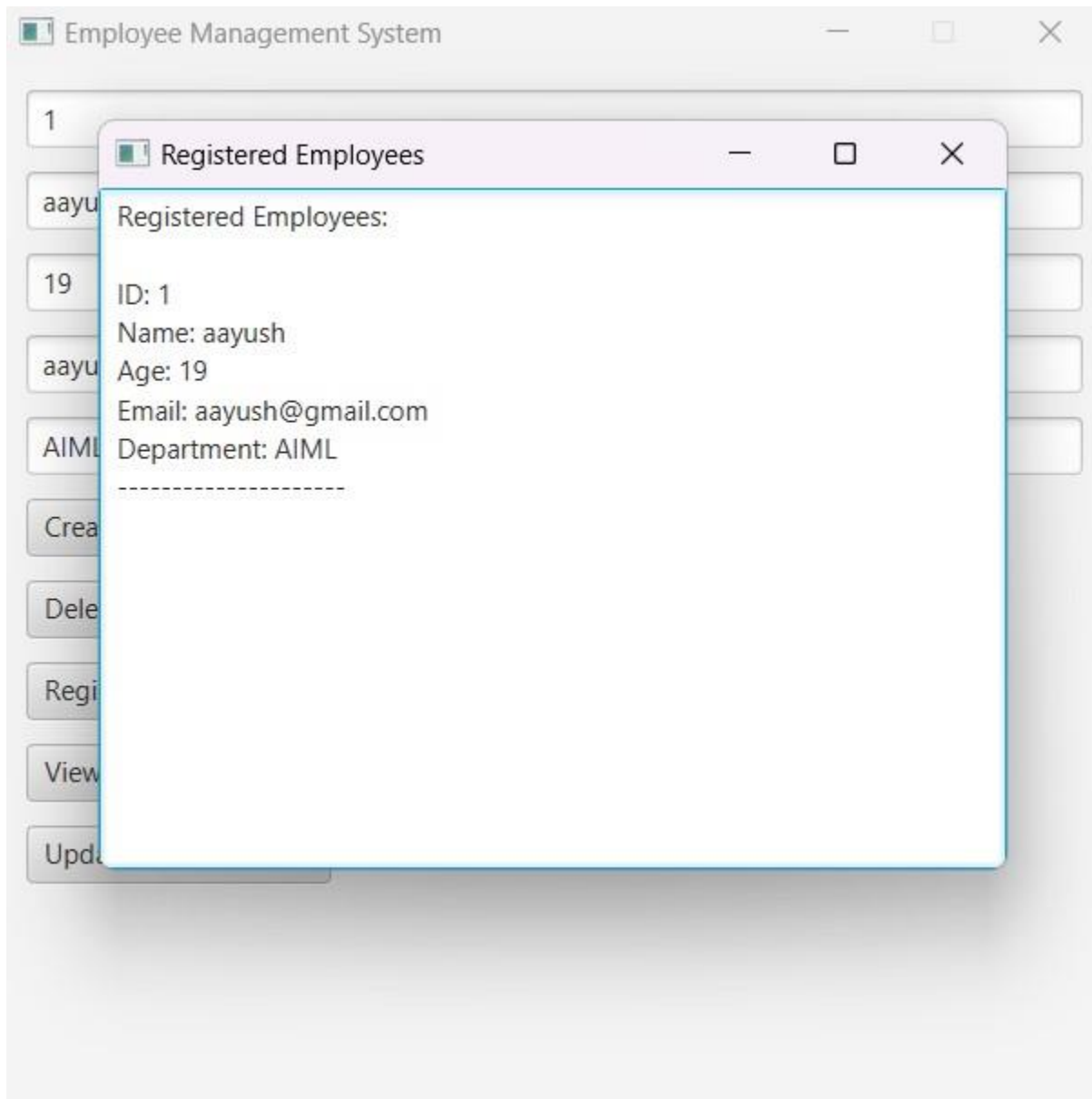


The screenshot shows a web application window titled "Employee Management System". It features five text input fields for "Employee ID", "Name", "Age", "Email", and "Department". Below these fields are five buttons: "Create Employee Table", "Delete Employee Table", "Register Employee", "View Employees", and "Update Employee Info". The "Delete Employee Table" button is highlighted with a blue border.

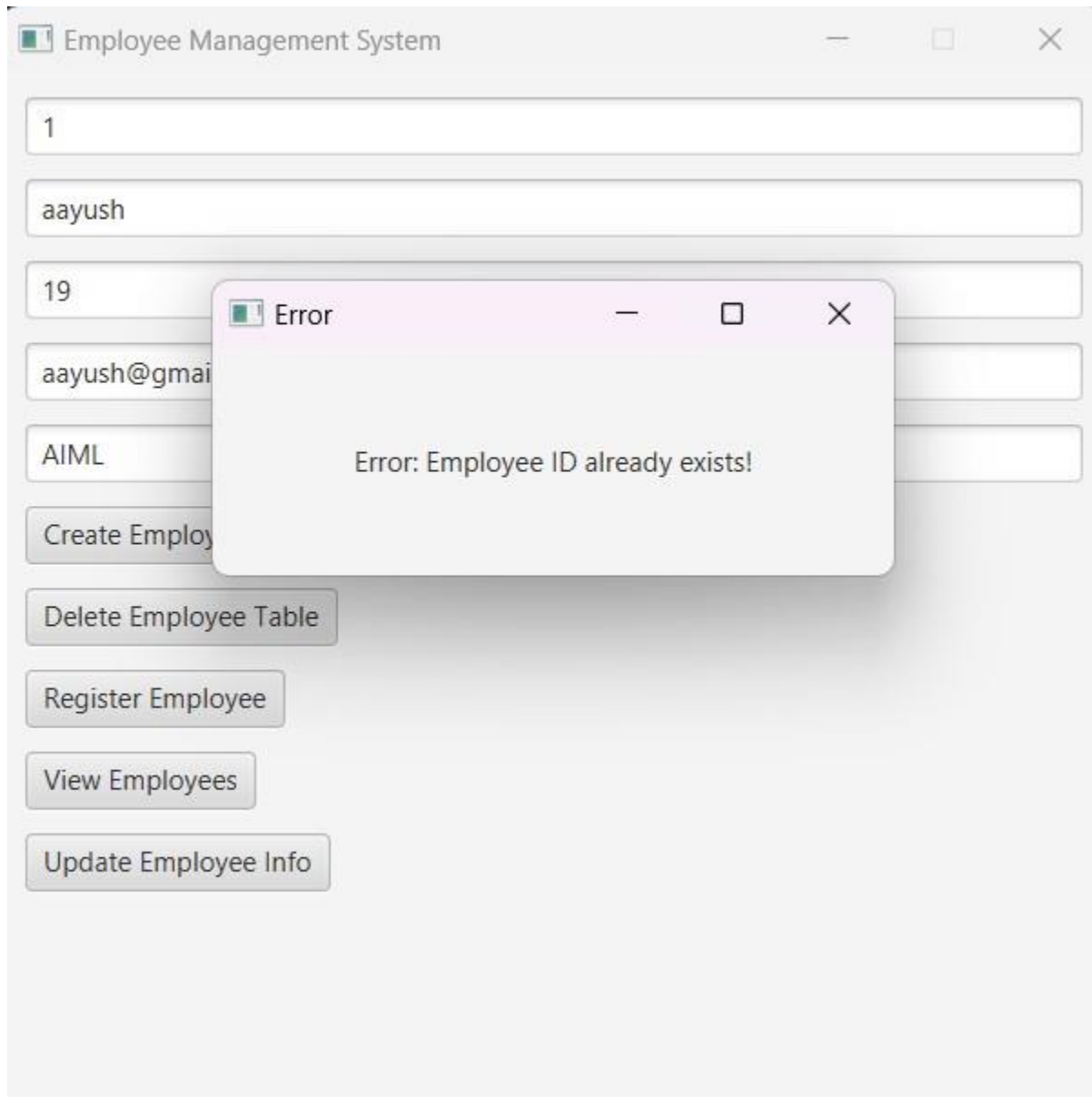
**Trying to register employee without creating table:**



**Creating table and then registering employee and viewing employees:**

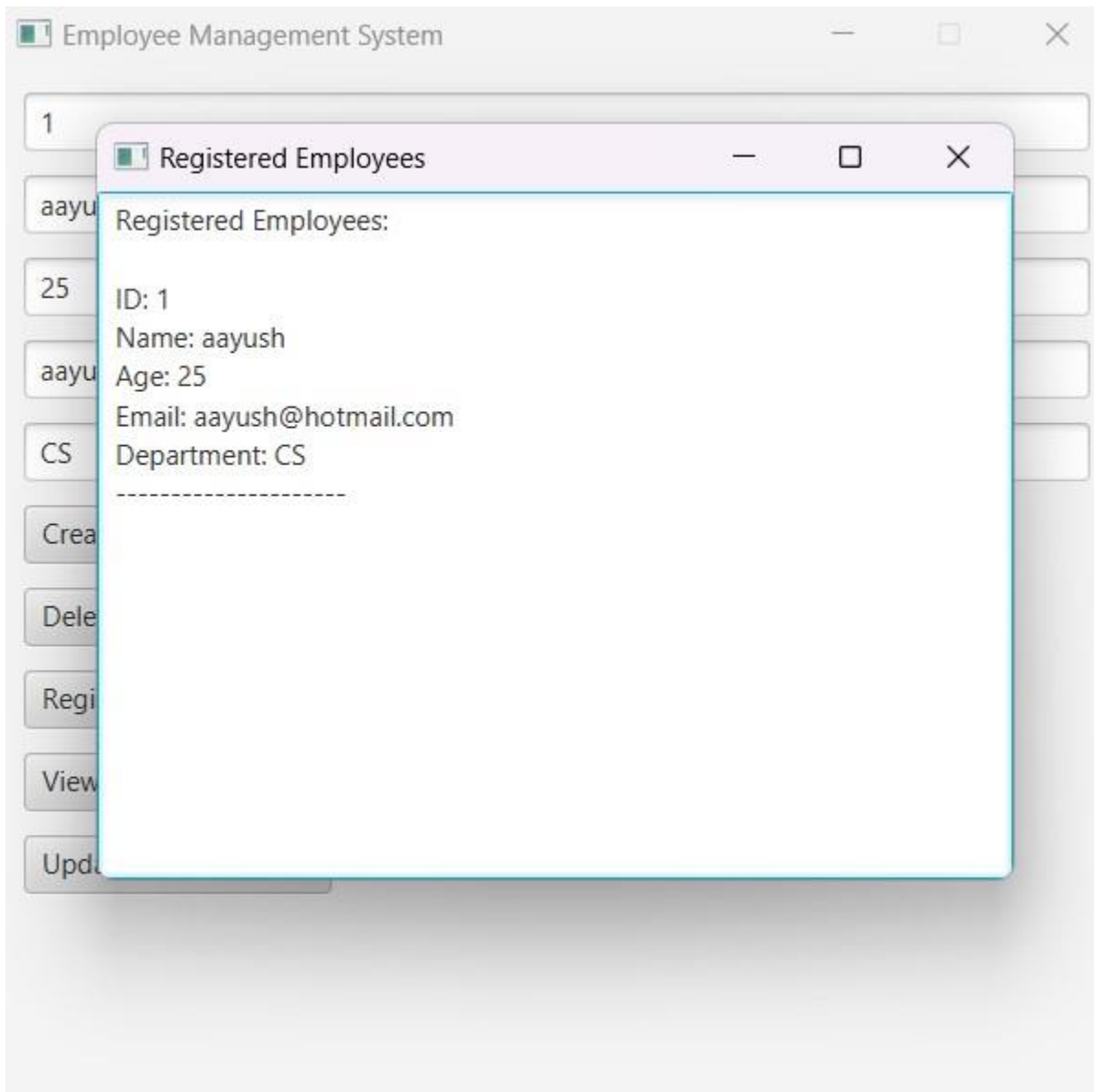


**Trying to register employee with same id:**

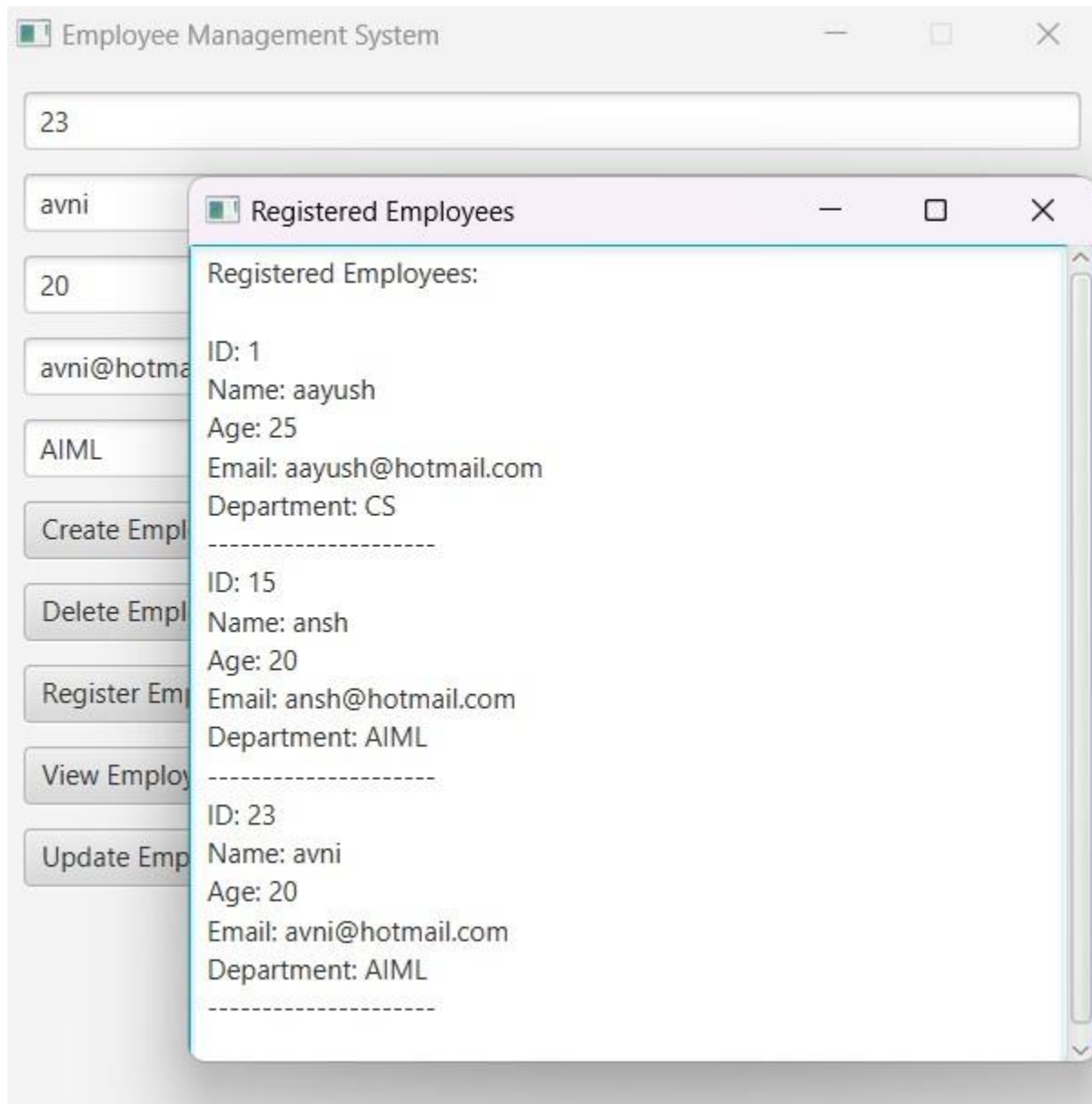


**Updating employee details base don employee id and viewing employees:**

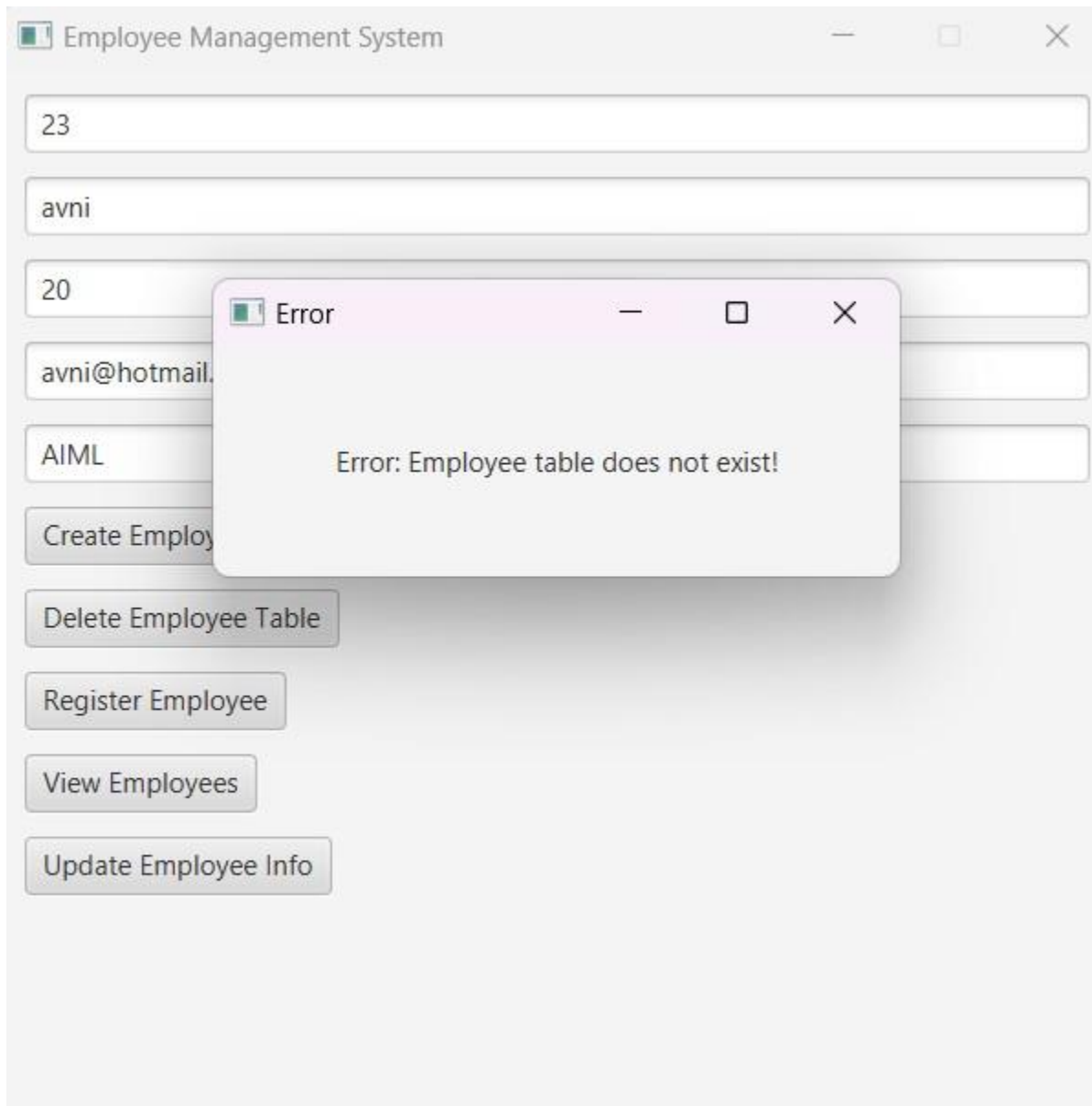




**Adding multiple employees and viewing:**



**Deleting the table and then viewing employees:**



<https://github.com/Vishakhajoshi24/JAVA-CODE>