JAVA

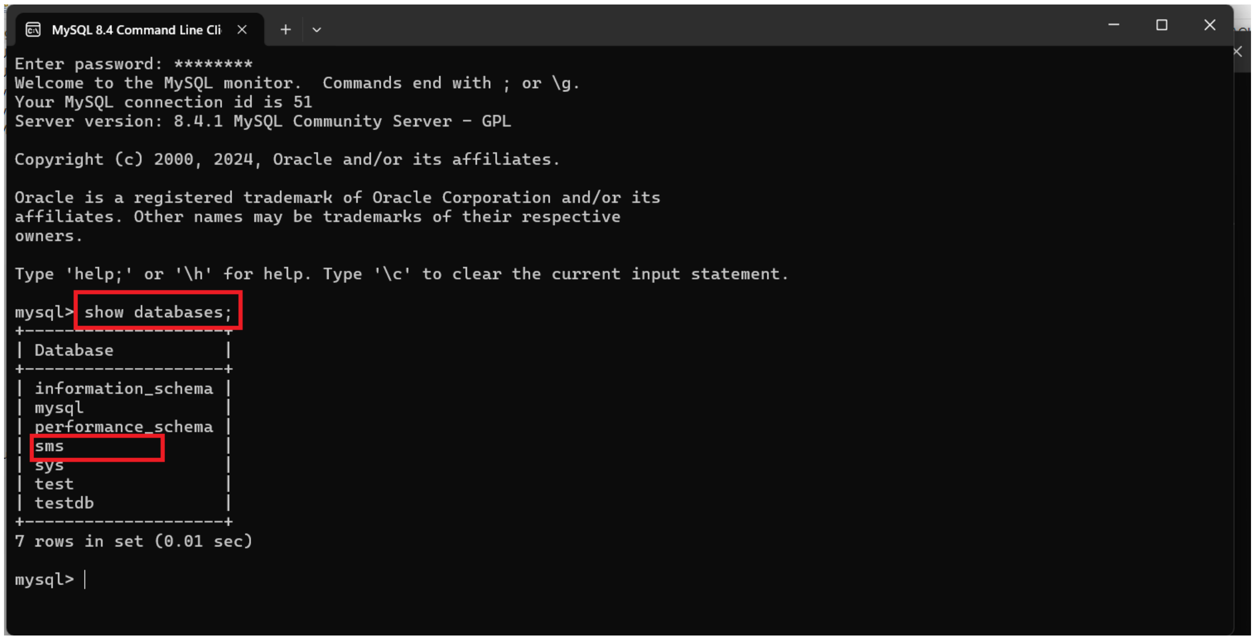
Student ID : C0930321

Student Name : Shreejana Shrestha

# Student Management System

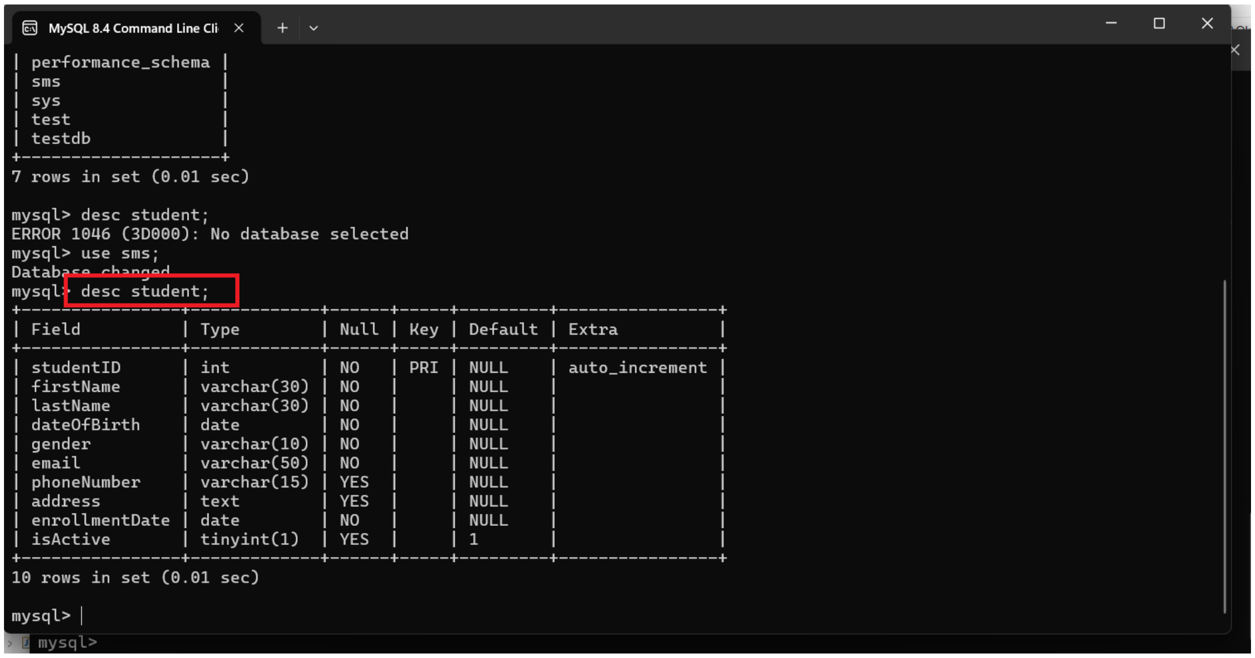
# Database and its connectivity to the SMS project

# showing database



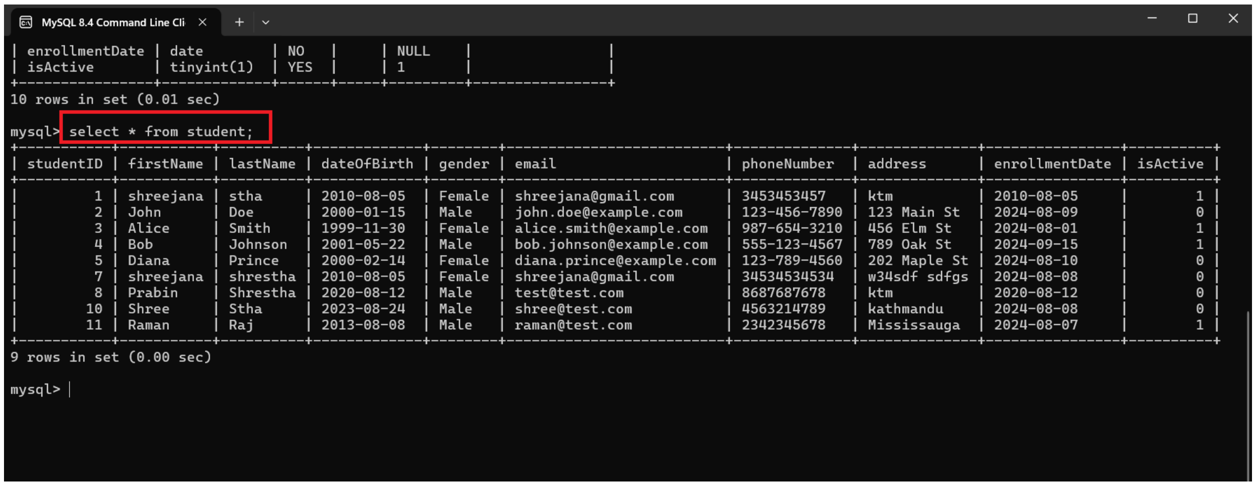
```
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 51
Server version: 8.4.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sms                |
| sys                |
| test               |
| testdb             |
+--------------------+
7 rows in set (0.01 sec)

mysql> |
```

# Structure of student table



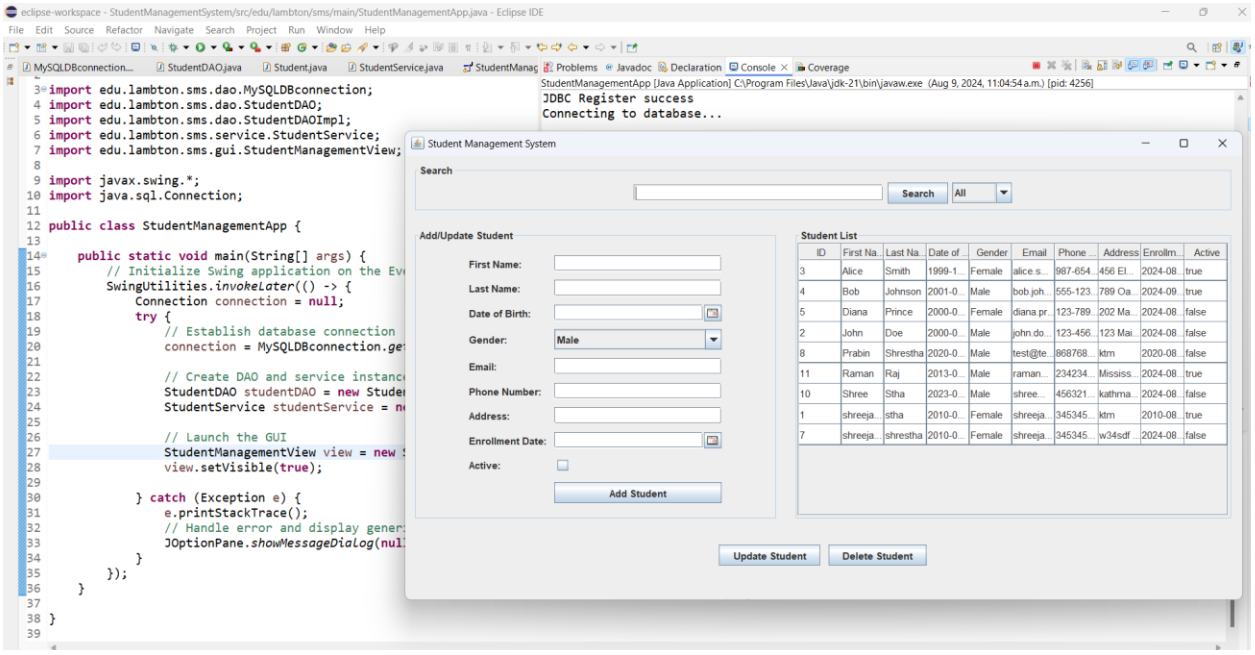
```
| performance_schema |
| sms                |
| sys                |
| test               |
| testdb             |
+--------------------+
7 rows in set (0.01 sec)

mysql> desc student;
ERROR 1046 (3D000): No database selected
mysql> use sms;
Database changed
mysql> desc student;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| studentID      | int         | NO   | PRI | NULL    | auto_increment |
| firstName      | varchar(30) | NO   |     | NULL    |                |
| lastName       | varchar(30) | NO   |     | NULL    |                |
| dateOfBirth    | date        | NO   |     | NULL    |                |
| gender         | varchar(10) | NO   |     | NULL    |                |
| email          | varchar(50) | NO   |     | NULL    |                |
| phoneNumber    | varchar(15) | YES  |     | NULL    |                |
| address        | text        | YES  |     | NULL    |                |
| enrollmentDate | date        | NO   |     | NULL    |                |
| isActive       | tinyint(1)  | YES  |     | 1       |                |
+----------------+-------------+------+-----+---------+----------------+
10 rows in set (0.01 sec)

mysql>
```

# Existing Records on the student table



```
| enrollmentDate | date        | NO   |     | NULL    |                |
| isActive       | tinyint(1)  | YES  |     | 1       |                |
+----------------+-------------+------+-----+---------+----------------+
10 rows in set (0.01 sec)

mysql> select * from student;
+-----------+-----------+----------+-------------+--------+--------------------------+-------------+--------------+----------------+----------+
| studentID | firstName | lastName | dateOfBirth | gender | email                    | phoneNumber | address      | enrollmentDate | isActive |
+-----------+-----------+----------+-------------+--------+--------------------------+-------------+--------------+----------------+----------+
|         1 | shreejana | stha     | 2010-08-05  | Female | shreejana@gmail.com      | 3453453457  | ktm          | 2010-08-05     |        1 |
|         2 | John      | Doe      | 2000-01-15  | Male   | john.doe@example.com     | 123-456-7890| 123 Main St  | 2024-08-09     |        0 |
|         3 | Alice     | Smith    | 1999-11-30  | Female | alice.smith@example.com  | 987-654-3210| 456 Elm St   | 2024-08-01     |        1 |
|         4 | Bob       | Johnson  | 2001-05-22  | Male   | bob.johnson@example.com  | 555-123-4567| 789 Oak St   | 2024-09-15     |        1 |
|         5 | Diana     | Prince   | 2000-02-14  | Female | diana.prince@example.com | 123-789-4560| 202 Maple St | 2024-08-10     |        0 |
|         7 | shreejana | shrestha | 2010-08-05  | Female | shreejana@gmail.com      | 34534534534 | w34sdf sdfgs | 2024-08-08     |        0 |
|         8 | Prabin    | Shrestha | 2020-08-12  | Male   | test@test.com            | 8687687678  | ktm          | 2020-08-12     |        0 |
|        10 | Shree     | Stha     | 2023-08-24  | Male   | shree@test.com           | 4563214789  | kathmandu    | 2024-08-08     |        0 |
|        11 | Raman     | Raj      | 2013-08-08  | Male   | raman@test.com           | 2342345678  | Mississauga  | 2024-08-07     |        1 |
+-----------+-----------+----------+-------------+--------+--------------------------+-------------+--------------+----------------+----------+
9 rows in set (0.00 sec)

mysql>
```

# Making connection to the database



```
import edu.lambton.sms.dao.MySQLDBconnection;
import edu.lambton.sms.dao.StudentDAO;
import edu.lambton.sms.dao.StudentDAOImpl;
import edu.lambton.sms.service.StudentService;
import edu.lambton.sms.gui.StudentManagementView;

import javax.swing.*;
import java.sql.Connection;

public class StudentManagementApp {

    public static void main(String[] args) {
        // Initialize Swing application on the Eve
        SwingUtilities.invokeLater(() -> {
            Connection connection = null;
            try {
                // Establish database connection
                connection = MySQLDBconnection.ge

                // Create DAO and service instanc
                StudentDAO studentDAO = new Studer
                StudentService studentService = n

                // Launch the GUI
                StudentManagementView view = new 
                view.setVisible(true);

            } catch (Exception e) {
                e.printStackTrace();
                // Handle error and display gener
                JOptionPane.showMessageDialog(nul
            }
        });
    }

}
```

```
StudentManagementApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Aug 9, 2024, 11:04:54 a.m.) [pid: 4256]
JDBC Register success
Connecting to database...
```

Student Management System

Search

Search field: All

Add/Update Student

First Name:

Last Name:

Date of Birth:

Gender: Male

Email:

Phone Number:

Address:

Enrollment Date:

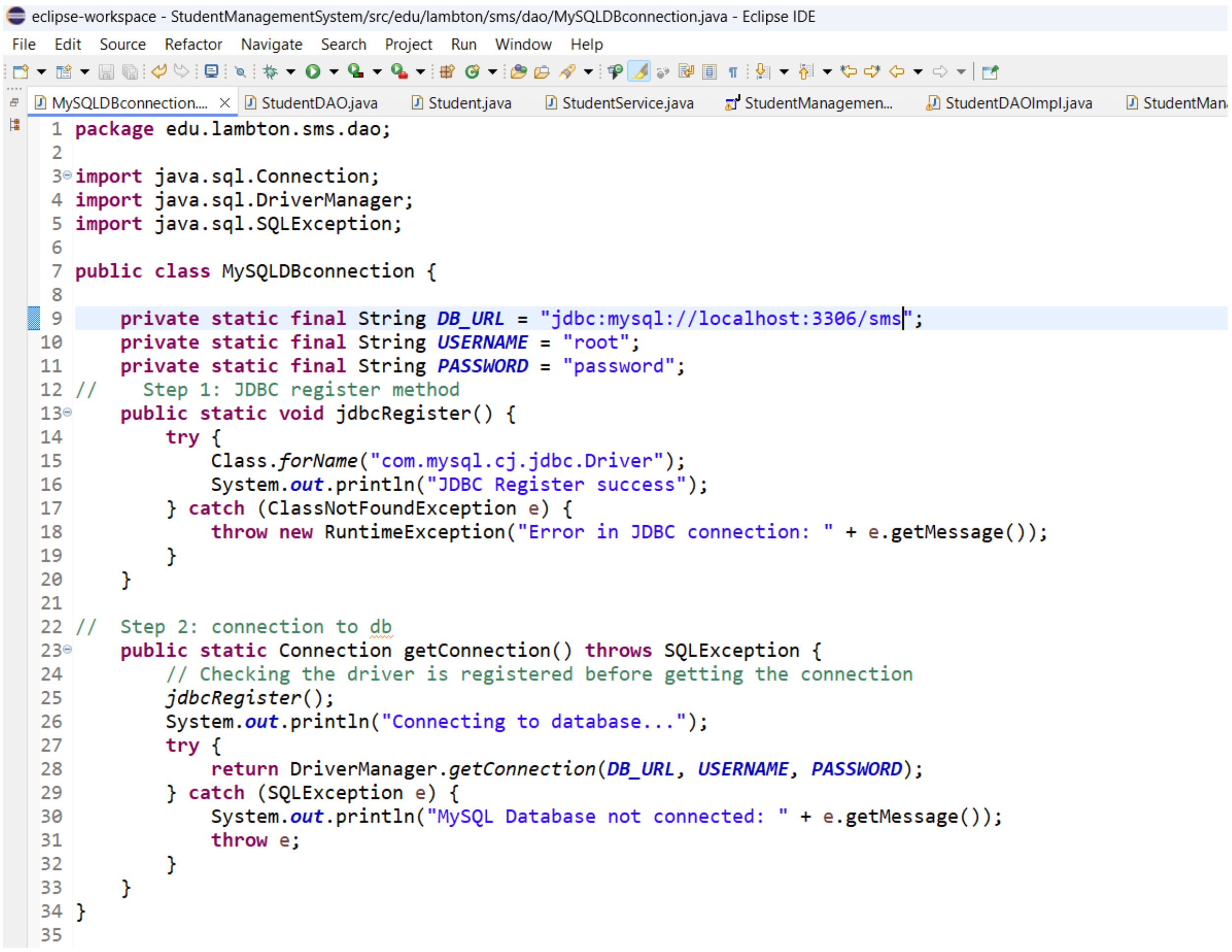
Active:

Student List

ID	First Na...	Last Na...	Date of ...	Gender	Email	Phone ...	Address	Enrollm...	Active
3	Alice	Smith	1999-1...	Female	alice.s...	987-654...	456 El...	2024-08...	true
4	Bob	Johnson	2001-0...	Male	bob.joh...	555-123...	789 Oa...	2024-09...	true
5	Diana	Prince	2000-0...	Female	diana.pr...	123-789...	202 Ma...	2024-08...	false
2	John	Doe	2000-0...	Male	john.do...	123-456...	123 Mai...	2024-08...	false
8	Prabin	Shrestha	2020-0...	Male	test@te...	868768...	ktm	2020-08...	false
11	Raman	Raj	2013-0...	Male	raman...	234234...	Mississ...	2024-08...	true
10	Shree	Stha	2023-0...	Male	shree...	456321...	kathma...	2024-08...	false
1	shreeja...	stha	2010-0...	Female	shreeja...	345345...	ktm	2010-08...	true
7	shreeja...	shrestha	2010-0...	Female	shreeja...	345345...	w34sdf ...	2024-08...	false

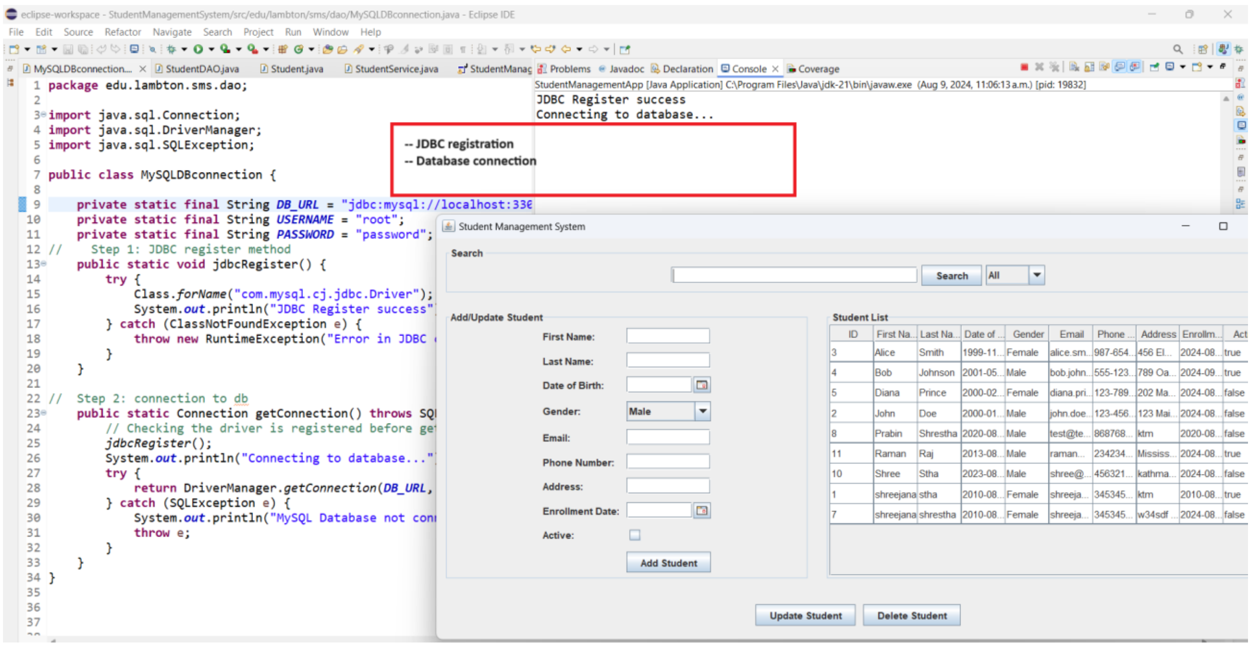
eclipse-workspace - StudentManagementSystem/src/edu/lambton/sms/main/StudentManagementApp.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MySQLDBconnection.... StudentDAO.java Student.java StudentService.java StudentManag
Problems Javadoc Declaration Console Coverage
Add Student
Update Student
Delete Student

# Database connectivity code snippet


eclipse-workspace - StudentManagementSystem/src/edu/lambton/sms/dao/MySQLDBconnection.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MySQLDBconnection.... StudentDAO.java Student.java StudentService.java StudentManagemen... StudentDAOImpl.java StudentMan

```java
package edu.lambton.sms.dao;

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

public class MySQLDBconnection {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/sms";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "password";
//    Step 1: JDBC register method
    public static void jdbcRegister() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("JDBC Register success");
        } catch (ClassNotFoundException e) {
            throw new RuntimeException("Error in JDBC connection: " + e.getMessage());
        }
    }

//  Step 2: connection to db
    public static Connection getConnection() throws SQLException {
        // Checking the driver is registered before getting the connection
        jdbcRegister();
        System.out.println("Connecting to database...");
        try {
            return DriverManager.getConnection(DB_URL, USERNAME, PASSWORD);
        } catch (SQLException e) {
            System.out.println("MySQL Database not connected: " + e.getMessage());
            throw e;
        }
    }
}

```

⇨ **MySQLDBconnection** class handles the database connection setup which is initialized from the **StudentManagementApp** class. Here, we have registered the jdbc and make database connection using DriverManager.getConnection(DB_URL, USERNAME, PASSWORD).


eclipse-workspace - StudentManagementSystem/src/edu/lambton/sms/dao/MySQLDBconnection.java - Eclipse IDE

```
package edu.lambton.sms.dao;

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

public class MySQLDBconnection {

    private static final String DB_URL = "jdbc:mysql://localhost:33
    private static final String USERNAME = "root";
    private static final String PASSWORD = "password";
//    Step 1: JDBC register method
    public static void jdbcRegister() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("JDBC Register success"
        } catch (ClassNotFoundException e) {
            throw new RuntimeException("Error in JDBC 
        }
    }

//  Step 2: connection to db
    public static Connection getConnection() throws SQ
        // Checking the driver is registered before ge
        jdbcRegister();
        System.out.println("Connecting to database..."
        try {
            return DriverManager.getConnection(DB_URL,
        } catch (SQLException e) {
            System.out.println("MySQL Database not con
            throw e;
        }
    }
}
```

StudentManagementApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Aug 9, 2024, 11:06:13 a.m.) [pid: 19832]

```
JDBC Register success
Connecting to database...
```

-- JDBC registration
-- Database connection

Student Management System

Search

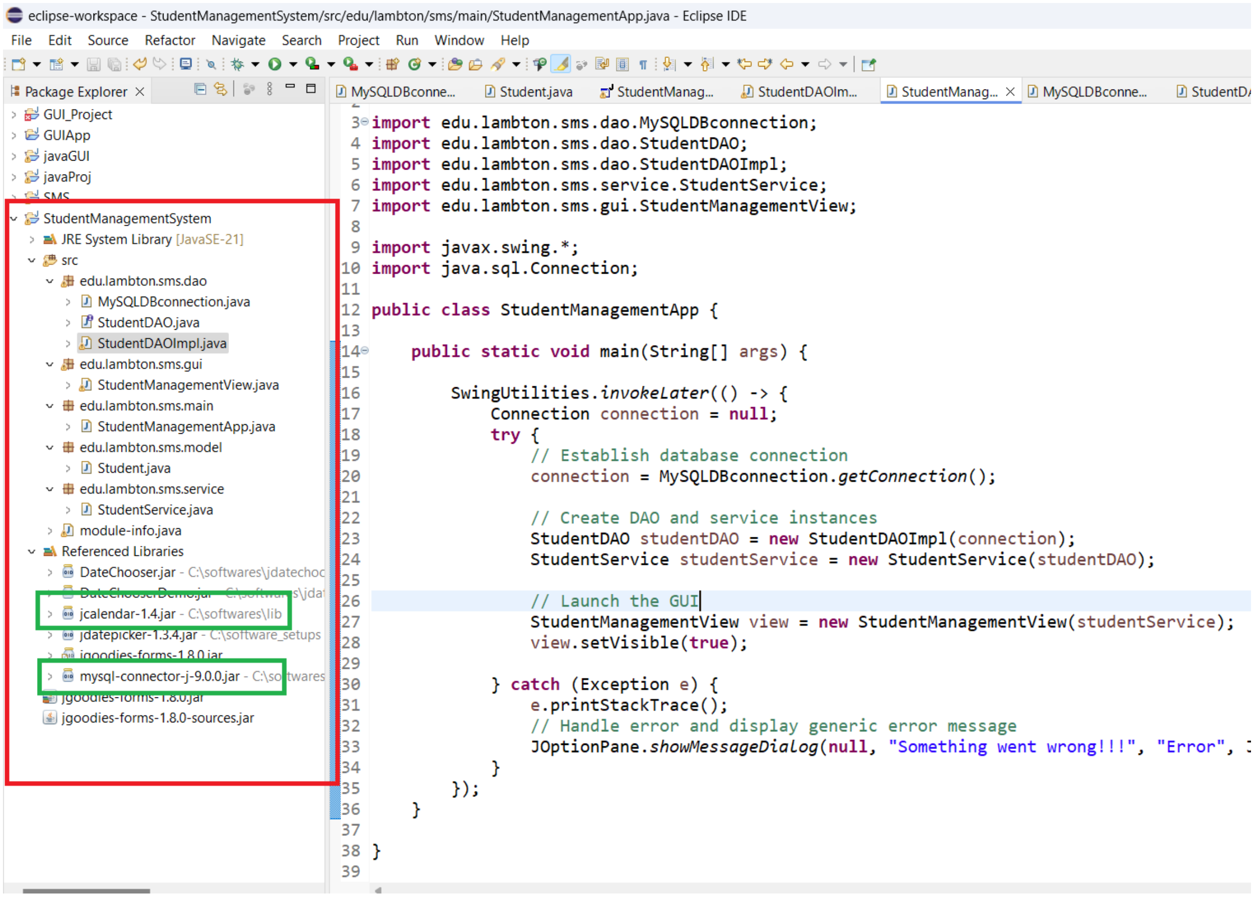
Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	

Student List

ID	First Na...	Last Na...	Date of ...	Gender	Email	Phone ...	Address	Enrollm...	Act
3	Alice	Smith	1999-11...	Female	alice.sm...	987-654...	456 El...	2024-08...	true
4	Bob	Johnson	2001-05...	Male	bob.john...	555-123...	789 Oa...	2024-09...	true
5	Diana	Prince	2000-02...	Female	diana.pri...	123-789...	202 Ma...	2024-08...	false
2	John	Doe	2000-01...	Male	john.doe...	123-456...	123 Mai...	2024-08...	false
8	Prabin	Shrestha	2020-08...	Male	test@te...	868768...	ktm	2020-08...	false
11	Raman	Raj	2013-08...	Male	raman...	234234...	Mississ...	2024-08...	true
10	Shree	Stha	2023-08...	Male	shree@...	456321...	kathma...	2024-08...	false
1	shreejana	stha	2010-08...	Female	shreeja...	345345...	ktm	2010-08...	true
7	shreejana	shrestha	2010-08...	Female	shreeja...	345345...	w34sdf ...	2024-08...	false

# Folder structure in the project


eclipse-workspace - StudentManagementSystem/src/edu/lambton/sms/main/StudentManagementApp.java - Eclipse IDE

Package Explorer

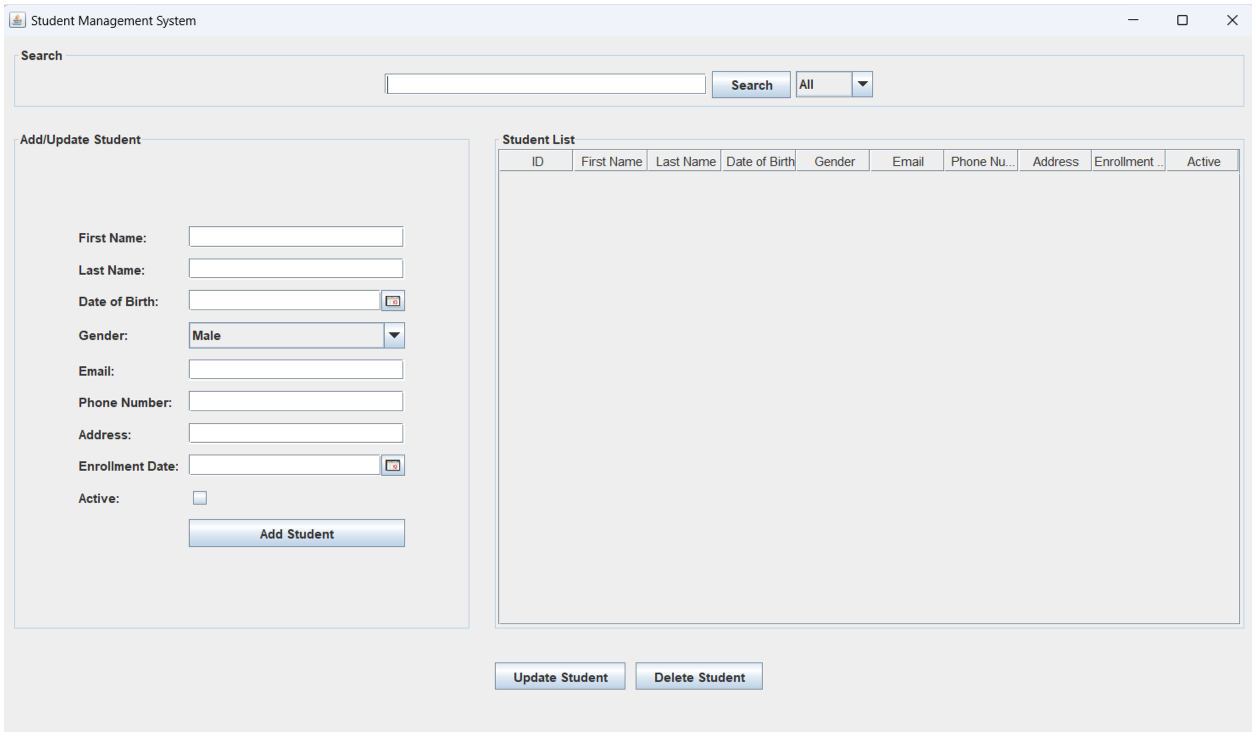
```
> GUI_Project
> GUIApp
> javaGUI
> javaProj
> SMS
v StudentManagementSystem
    > JRE System Library [JavaSE-21]
    v src
        v edu.lambton.sms.dao
            > MySQLDBconnection.java
            > StudentDAO.java
            > StudentDAOImpl.java
        v edu.lambton.sms.gui
            > StudentManagementView.java
        v edu.lambton.sms.main
            > StudentManagementApp.java
        v edu.lambton.sms.model
            > Student.java
        v edu.lambton.sms.service
            > StudentService.java
        > module-info.java
    v Referenced Libraries
        > DateChooser.jar - C:\softwares\jdatechoo
        > DateChooserDemo.jar - C:\softwares\jda
        > jcalendar-1.4.jar - C:\softwares\lib
        > jdatepicker-1.3.4.jar - C:\software_setups
        > jgoodies-forms-1.8.0.jar
        > mysql-connector-j-9.0.0.jar - C:\softwares
    jgoodies-forms-1.8.0.jar
    jgoodies-forms-1.8.0-sources.jar
```

```java
import edu.lambton.sms.dao.MySQLDBconnection;
import edu.lambton.sms.dao.StudentDAO;
import edu.lambton.sms.dao.StudentDAOImpl;
import edu.lambton.sms.service.StudentService;
import edu.lambton.sms.gui.StudentManagementView;

import javax.swing.*;
import java.sql.Connection;

public class StudentManagementApp {

    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> {
            Connection connection = null;
            try {
                // Establish database connection
                connection = MySQLDBconnection.getConnection();

                // Create DAO and service instances
                StudentDAO studentDAO = new StudentDAOImpl(connection);
                StudentService studentService = new StudentService(studentDAO);

                // Launch the GUI
                StudentManagementView view = new StudentManagementView(studentService);
                view.setVisible(true);

            } catch (Exception e) {
                e.printStackTrace();
                // Handle error and display generic error message
                JOptionPane.showMessageDialog(null, "Something went wrong!!!", "Error", 
            }
        });
    }

}
```

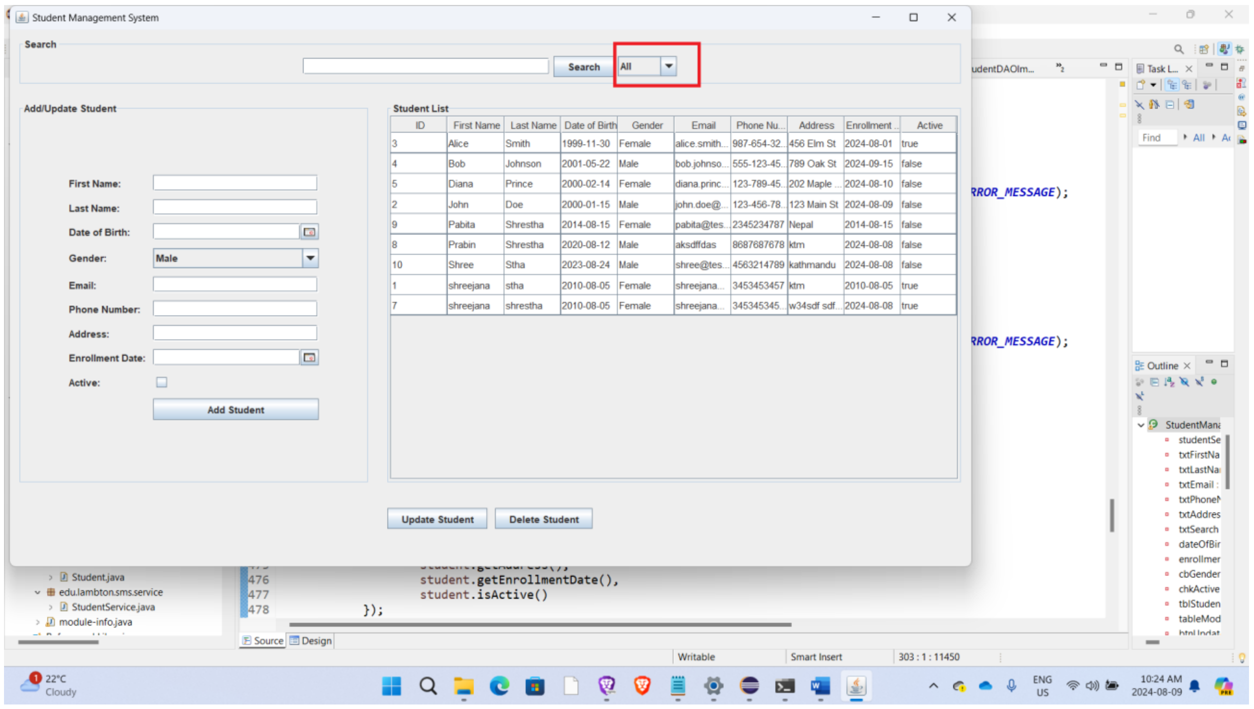
Here, we have used edu.lambton.sms as the root package for the project SMS. All the other packages gui, dao, service, main, model are organized under this. We have created a separate package and respective class for each functionality.

- ➔ Gui contains the components viewed by the admin/user. It manages the user interface, including forms and table views in our application.
- ➔ Dao contains data access object interface and implementation. It defines the data access methods for the student entity and contains all the SQL queries and database interaction logic.
- ➔ Model contains domain classes that represent the data model i.e Student in our application
- ➔ Service contains the service layer where the main business logic resides such as adding, updating, retrieving and deleting. It interacts with DAO layer to perform these operations.
- ➔ Main contains the main entry point of our application

# First GUI of Student Management System

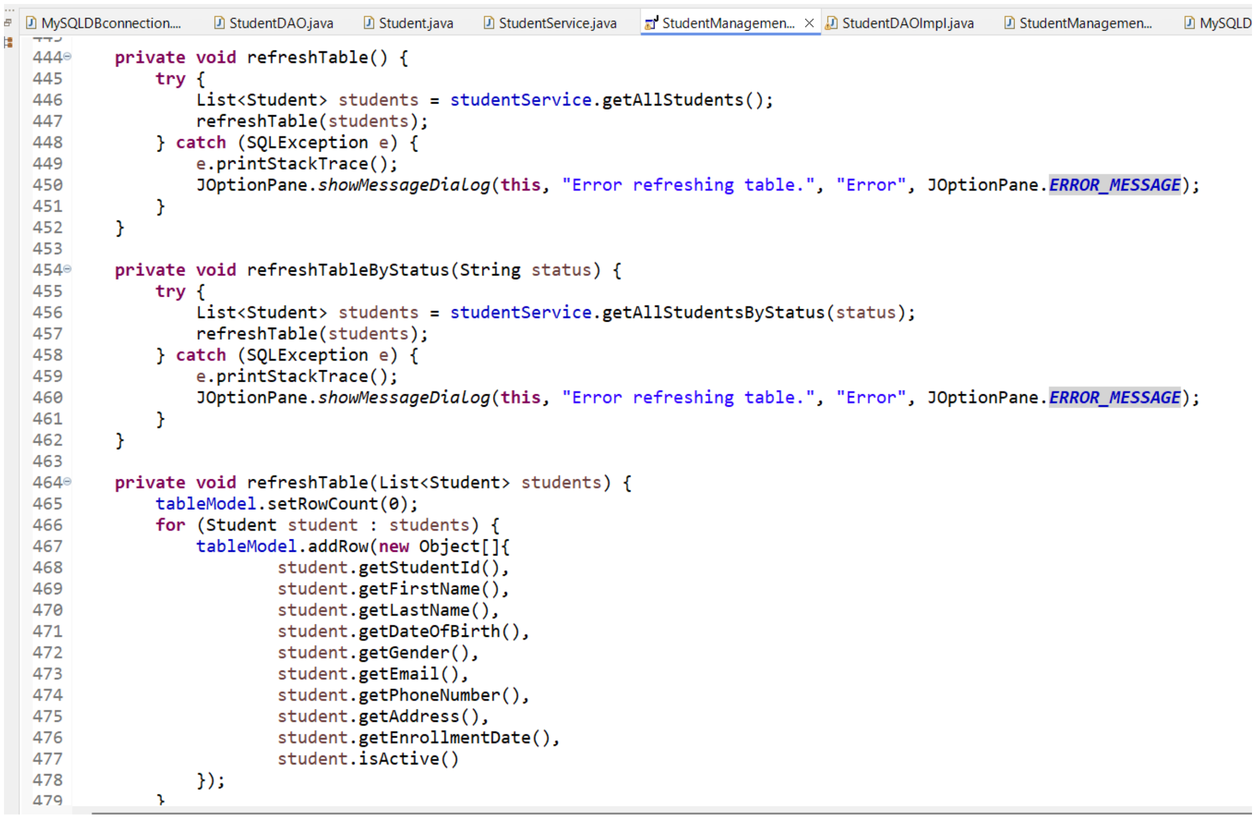


# GUI with all from the student table listed below



ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

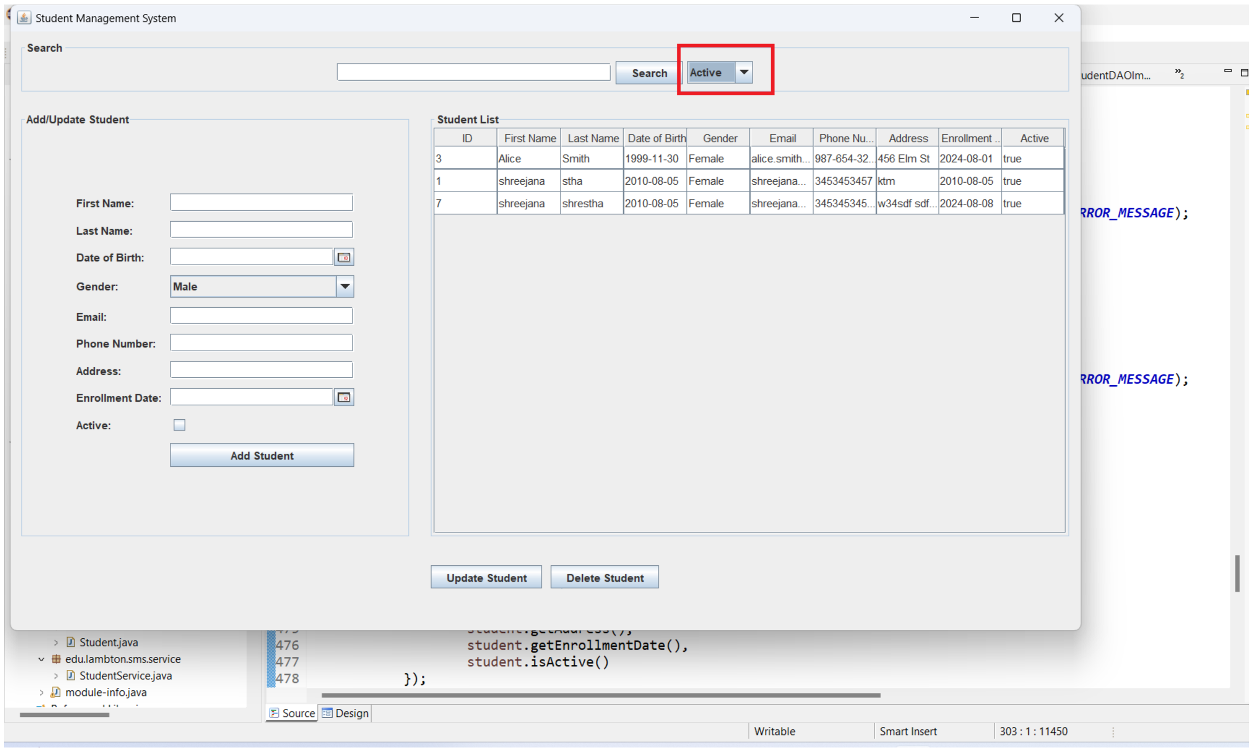
```
                student.getEnrollmentDate(),
                student.isActive()
        });
```



```java
    private void refreshTable() {
        try {
            List<Student> students = studentService.getAllStudents();
            refreshTable(students);
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error refreshing table.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void refreshTableByStatus(String status) {
        try {
            List<Student> students = studentService.getAllStudentsByStatus(status);
            refreshTable(students);
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error refreshing table.", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void refreshTable(List<Student> students) {
        tableModel.setRowCount(0);
        for (Student student : students) {
            tableModel.addRow(new Object[]{
                    student.getStudentId(),
                    student.getFirstName(),
                    student.getLastName(),
                    student.getDateOfBirth(),
                    student.getGender(),
                    student.getEmail(),
                    student.getPhoneNumber(),
                    student.getAddress(),
                    student.getEnrollmentDate(),
                    student.isActive()
            });
        }
```

⇨ This is the logic implemented to display all the records from the student table. The table is listed with data if exists when the application is first run.

# Table listing only the active students record



Active

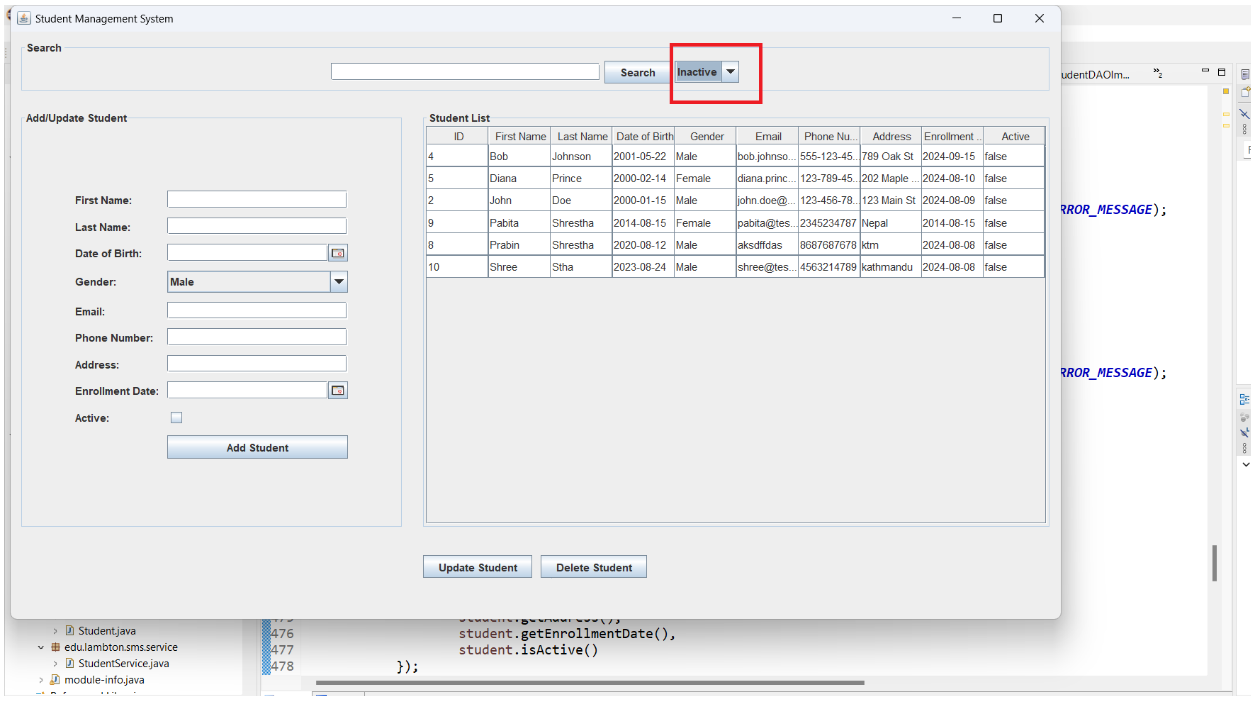
Gender: Male

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

```
            student.getEnrollmentDate(),
            student.isActive()
        });
```

# Table listing only the inactive students record



Inactive

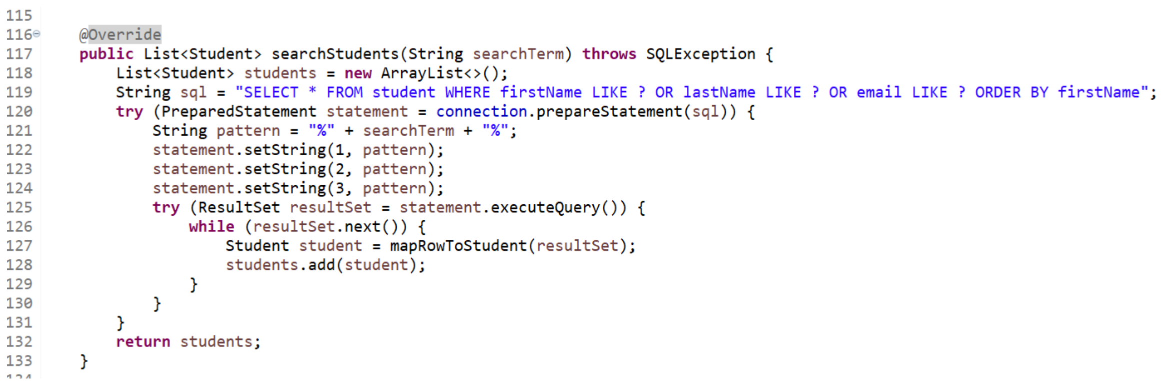
Gender: Male

Student List

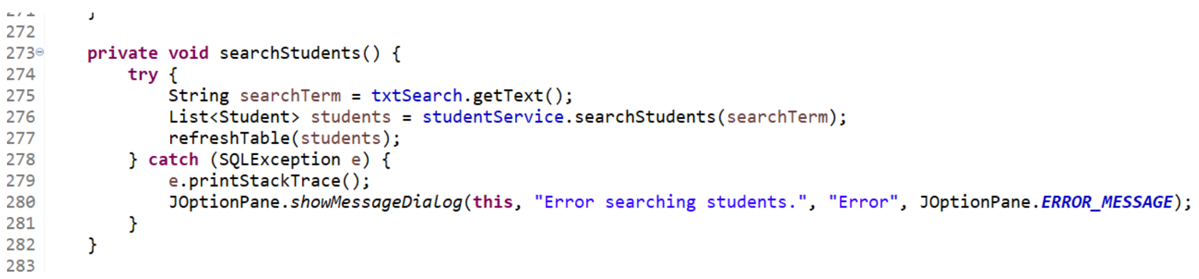
ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false

```
            student.getEnrollmentDate(),
            student.isActive()
        });
```

# Search filter

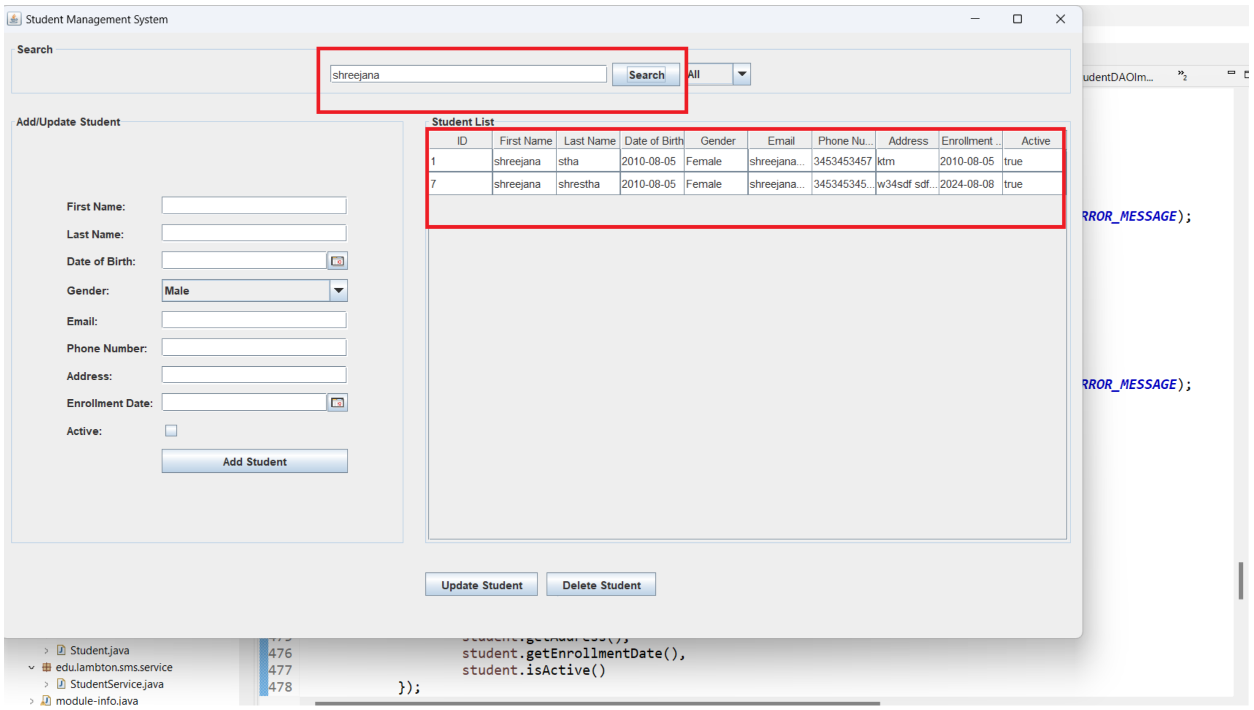


```java
@Override
public List<Student> searchStudents(String searchTerm) throws SQLException {
    List<Student> students = new ArrayList<>();
    String sql = "SELECT * FROM student WHERE firstName LIKE ? OR lastName LIKE ? OR email LIKE ? ORDER BY firstName";
    try (PreparedStatement statement = connection.prepareStatement(sql)) {
        String pattern = "%" + searchTerm + "%";
        statement.setString(1, pattern);
        statement.setString(2, pattern);
        statement.setString(3, pattern);
        try (ResultSet resultSet = statement.executeQuery()) {
            while (resultSet.next()) {
                Student student = mapRowToStudent(resultSet);
                students.add(student);
            }
        }
    }
    return students;
}
```



```java
private void searchStudents() {
    try {
        String searchTerm = txtSearch.getText();
        List<Student> students = studentService.searchStudents(searchTerm);
        refreshTable(students);
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error searching students.", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

➔ The search filter button enables the functionality to filter and retrieve student records based on specific criteria such as students’ firstName, lastName and email. When a search query is entered, the system queries the database and returns a list of students that match the criteria and only the relevant student information is displayed.



Student Management System

Search

shreejana **Search** All

Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

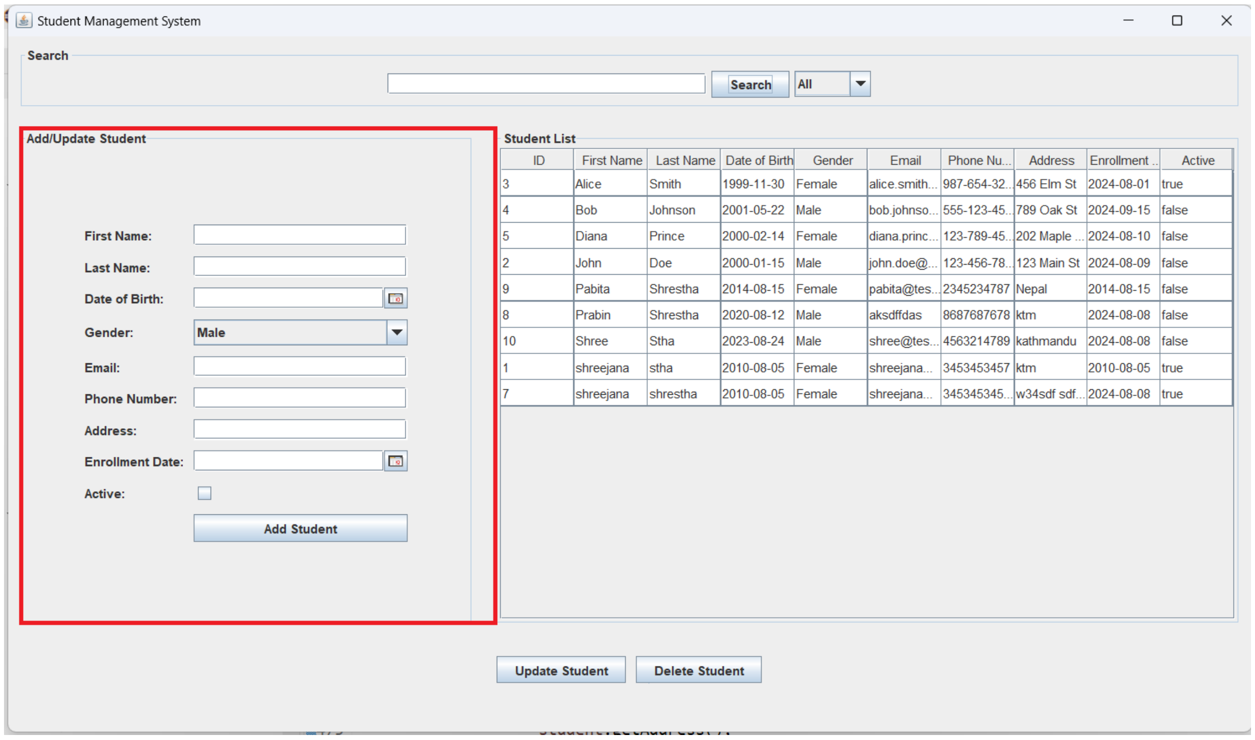
Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

```
RROR_MESSAGE);
RROR_MESSAGE);
```

```
            student.getEnrollmentDate(),
            student.isActive()
    });
```

# GUI to get the student data



Student Management System

Search

Search All

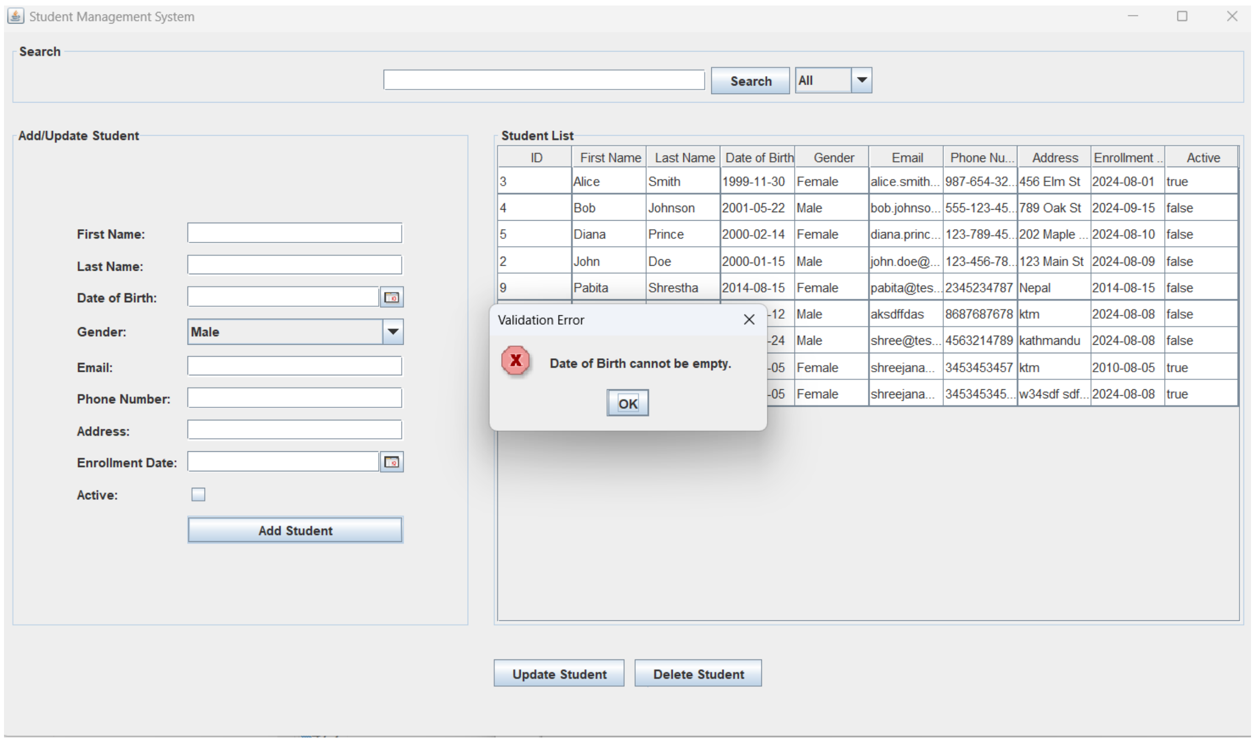
Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

Input validation for the form fields



Student Management System

Search

Search | All

Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

Add Student

Student List

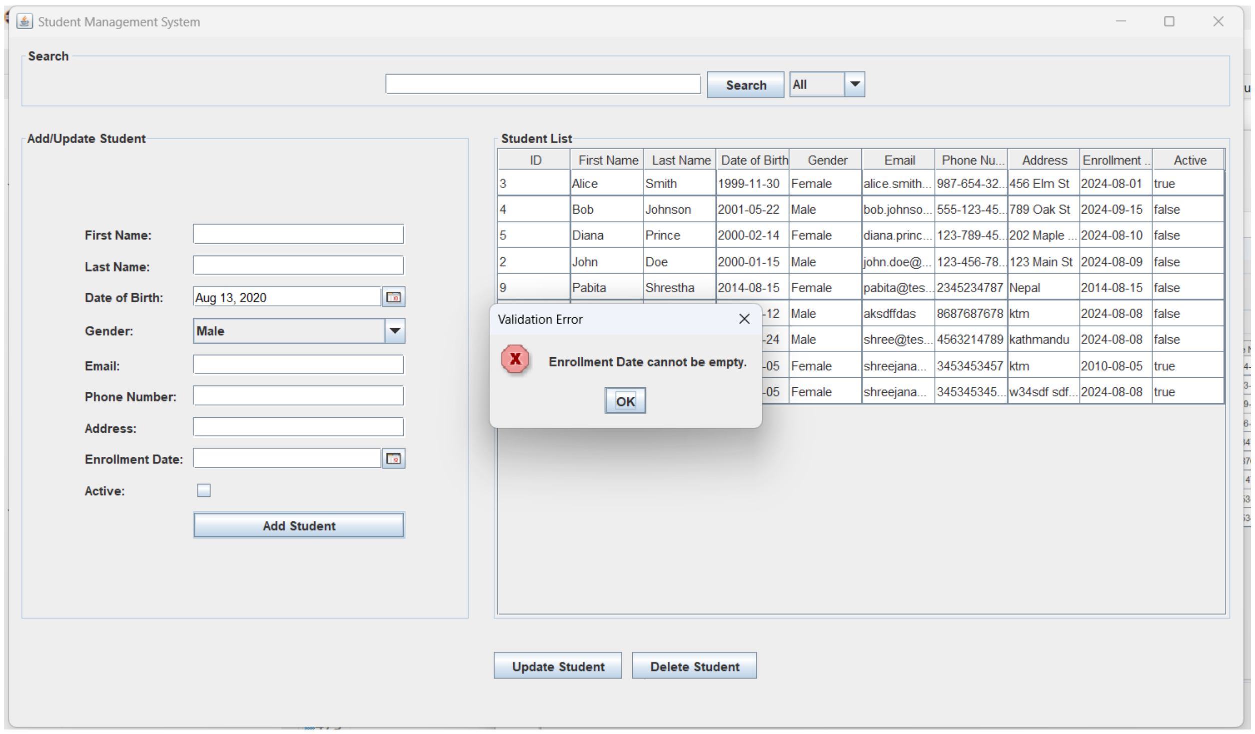
ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

Validation Error

Date of Birth cannot be empty.

OK

Update Student | Delete Student



Student Management System

Search

Search | All

Add/Update Student

First Name:	
Last Name:	
Date of Birth:	Aug 13, 2020
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

Add Student

Student List

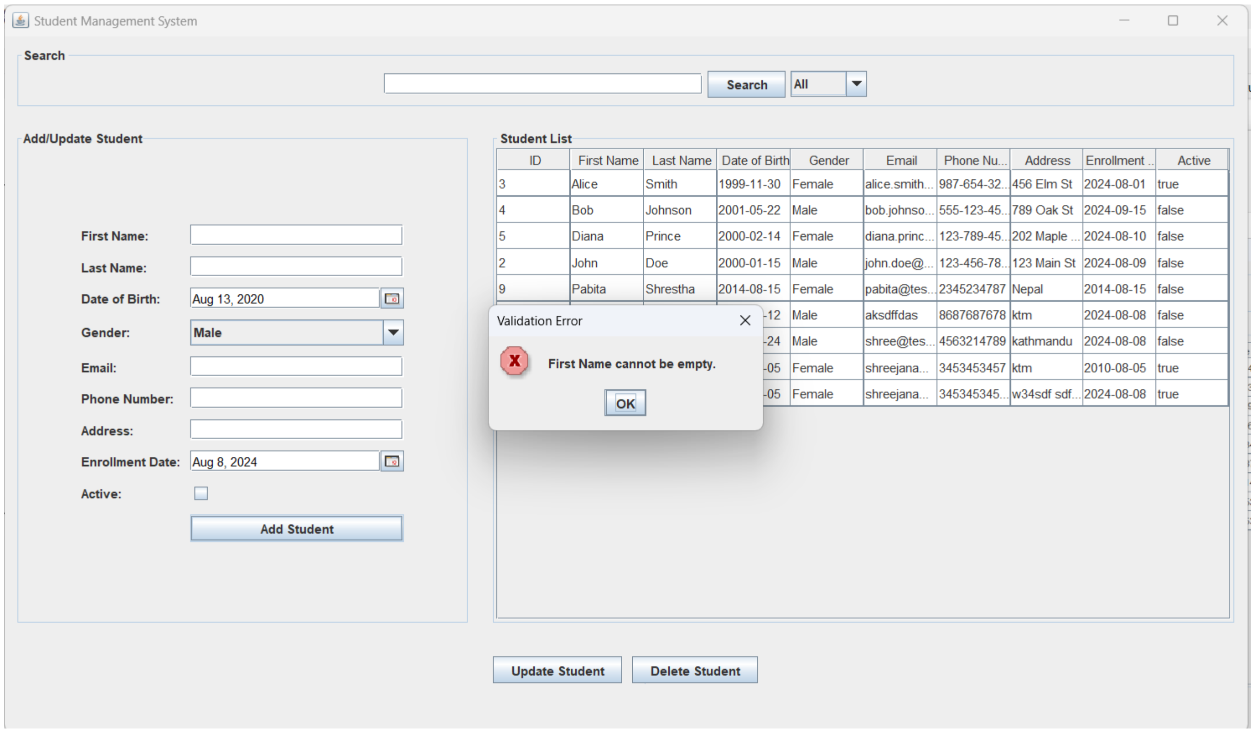
ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

Validation Error

Enrollment Date cannot be empty.

OK

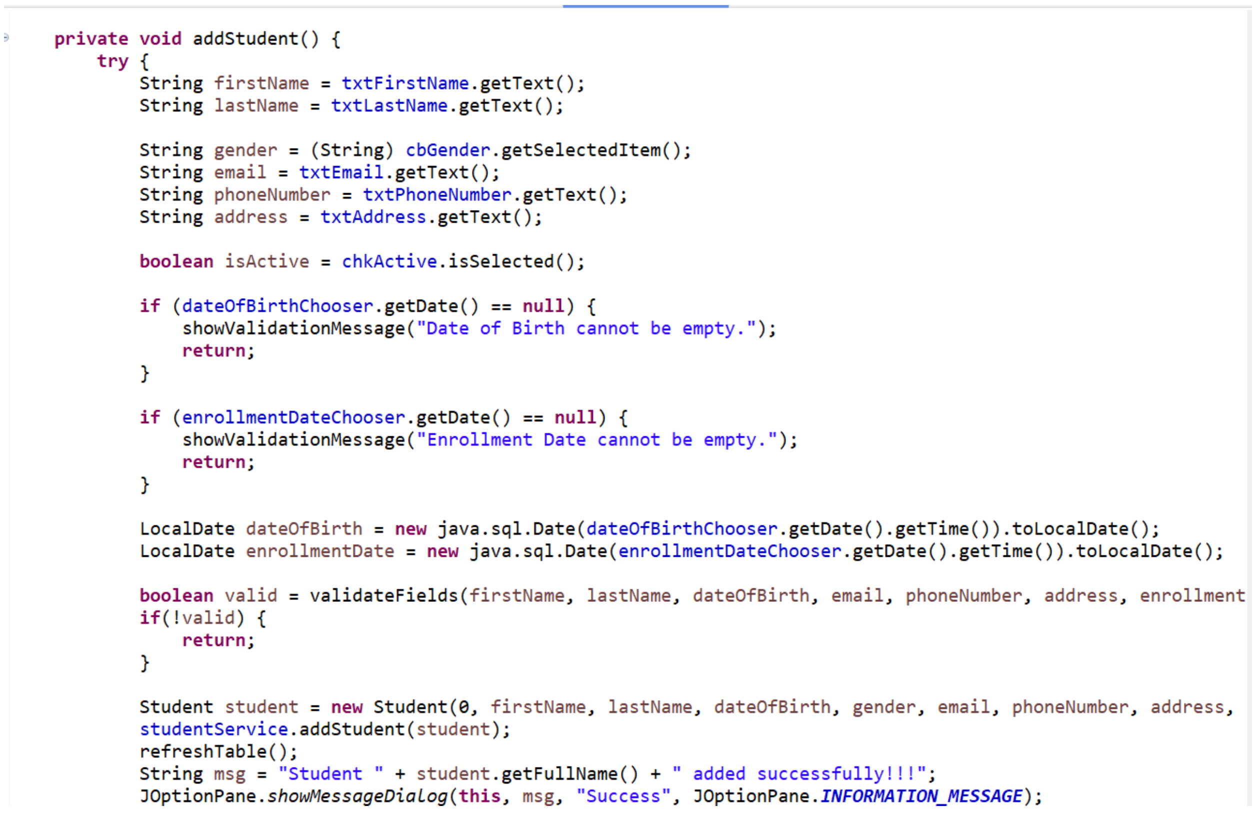
Update Student | Delete Student



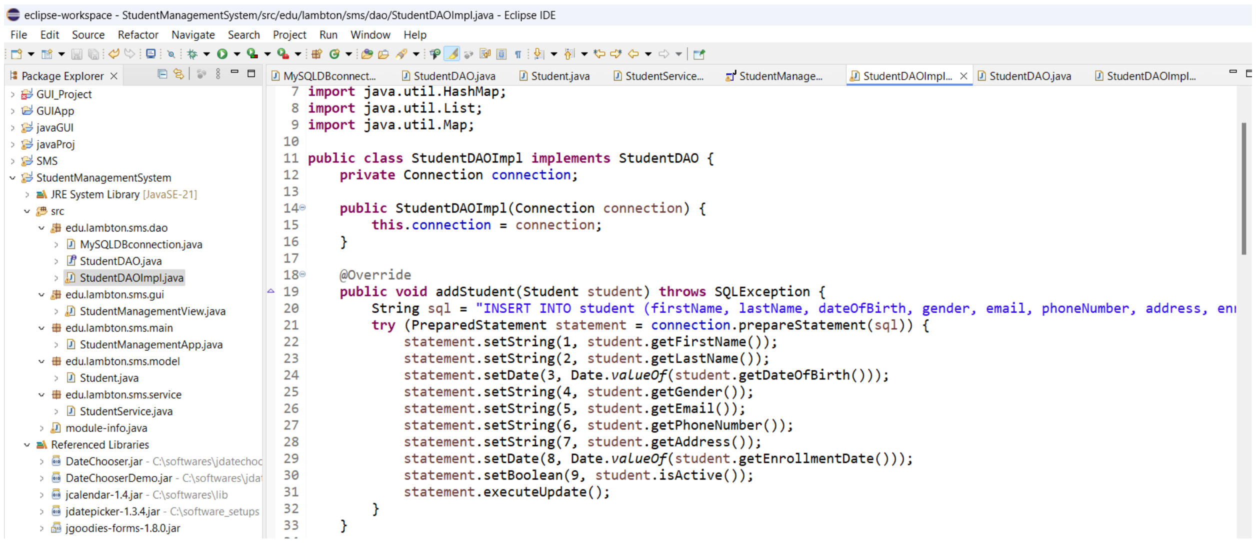
ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

First Name cannot be empty.

# Adding functionality

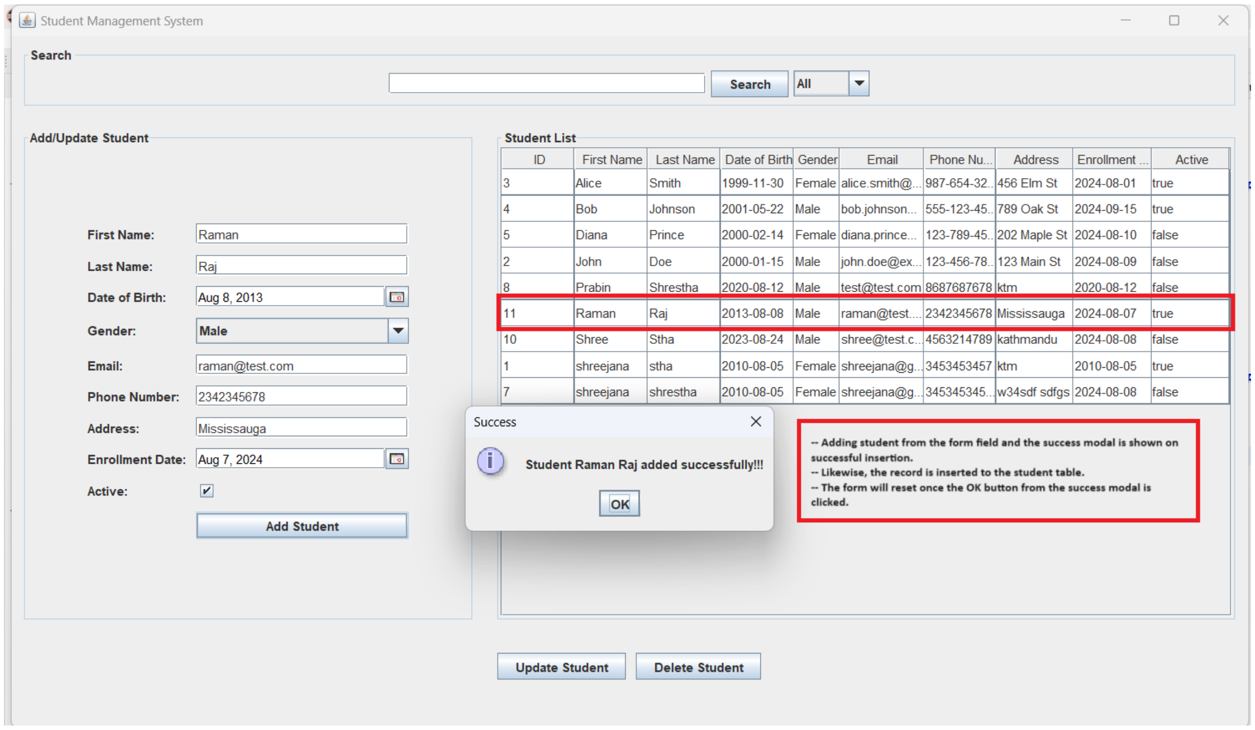


```java
    private void addStudent() {
        try {
            String firstName = txtFirstName.getText();
            String lastName = txtLastName.getText();

            String gender = (String) cbGender.getSelectedItem();
            String email = txtEmail.getText();
            String phoneNumber = txtPhoneNumber.getText();
            String address = txtAddress.getText();

            boolean isActive = chkActive.isSelected();

            if (dateOfBirthChooser.getDate() == null) {
                showValidationMessage("Date of Birth cannot be empty.");
                return;
            }

            if (enrollmentDateChooser.getDate() == null) {
                showValidationMessage("Enrollment Date cannot be empty.");
                return;
            }

            LocalDate dateOfBirth = new java.sql.Date(dateOfBirthChooser.getDate().getTime()).toLocalDate();
            LocalDate enrollmentDate = new java.sql.Date(enrollmentDateChooser.getDate().getTime()).toLocalDate();

            boolean valid = validateFields(firstName, lastName, dateOfBirth, email, phoneNumber, address, enrollment
            if(!valid) {
                return;
            }

            Student student = new Student(0, firstName, lastName, dateOfBirth, gender, email, phoneNumber, address,
            studentService.addStudent(student);
            refreshTable();
            String msg = "Student " + student.getFullName() + " added successfully!!!";
            JOptionPane.showMessageDialog(this, msg, "Success", JOptionPane.INFORMATION_MESSAGE);
```



eclipse-workspace - StudentManagementSystem/src/edu/lambton/sms/dao/StudentDAOImpl.java - Eclipse IDE

```java
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class StudentDAOImpl implements StudentDAO {
    private Connection connection;

    public StudentDAOImpl(Connection connection) {
        this.connection = connection;
    }

    @Override
    public void addStudent(Student student) throws SQLException {
        String sql = "INSERT INTO student (firstName, lastName, dateOfBirth, gender, email, phoneNumber, address, en
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, student.getFirstName());
            statement.setString(2, student.getLastName());
            statement.setDate(3, Date.valueOf(student.getDateOfBirth()));
            statement.setString(4, student.getGender());
            statement.setString(5, student.getEmail());
            statement.setString(6, student.getPhoneNumber());
            statement.setString(7, student.getAddress());
            statement.setDate(8, Date.valueOf(student.getEnrollmentDate()));
            statement.setBoolean(9, student.isActive());
            statement.executeUpdate();
        }
    }
```



Student Management System

Search

Add/Update Student

First Name:	Raman
Last Name:	Raj
Date of Birth:	Aug 8, 2013
Gender:	Male
Email:	raman@test.com
Phone Number:	2342345678
Address:	Mississauga
Enrollment Date:	Aug 7, 2024
Active:	✓

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	true
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.com	8687687678	ktm	2020-08-12	false
11	Raman	Raj	2013-08-08	Male	raman@test....	2342345678	Mississauga	2024-08-07	true
10	Shree	Stha	2023-08-24	Male	shree@test.c...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana@g...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	false

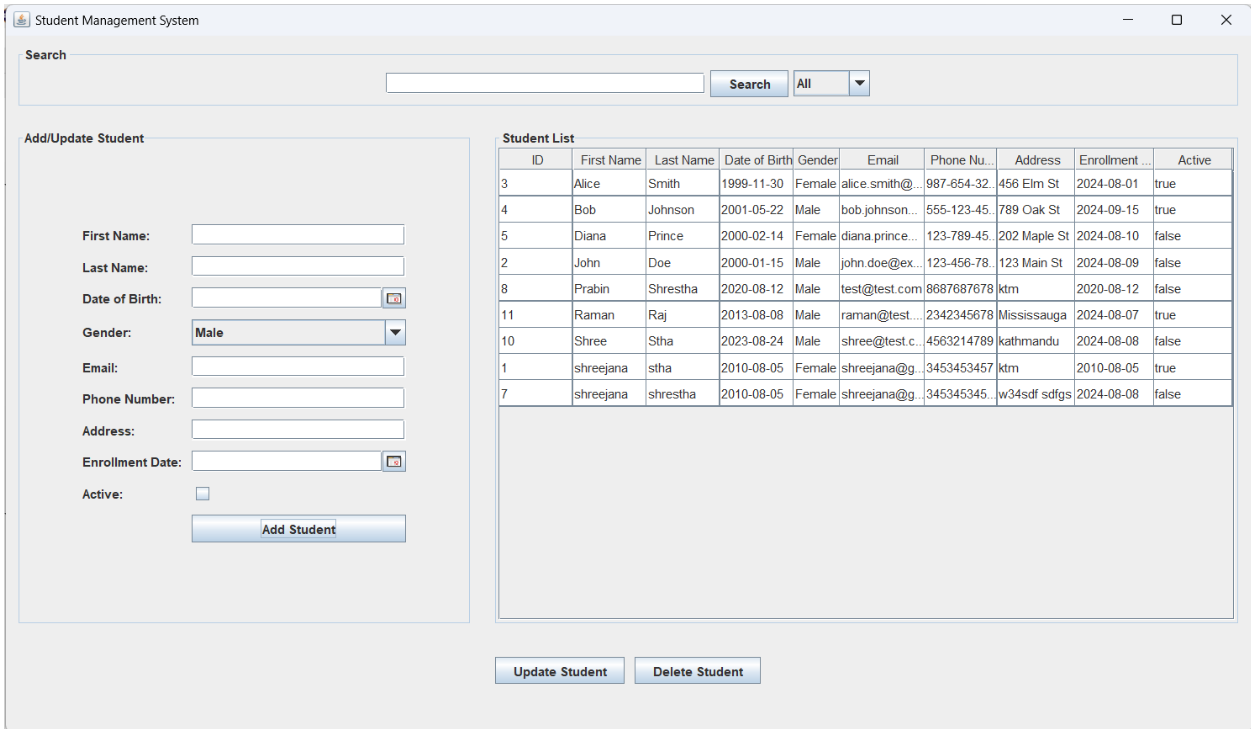
Success

Student Raman Raj added successfully!!!

-- Adding student from the form field and the success modal is shown on successful insertion.
-- Likewise, the record is inserted to the student table.
-- The form will reset once the OK button from the success modal is clicked.

➔ The "Add Student" functionality in our project allows to input student details through a form in the GUI and save them to the database. When the admin user submits the form, the input data is validated, ensuring that all required fields are filled out correctly. Once validated, a Student object is created using the form data, and the StudentService interacts with the StudentDAO to insert the new student record into the database. If the insertion is successful, the form is reset, and the student table in the GUI is refreshed to display the newly added student.

Table after successful addition of record


Student Management System

Search

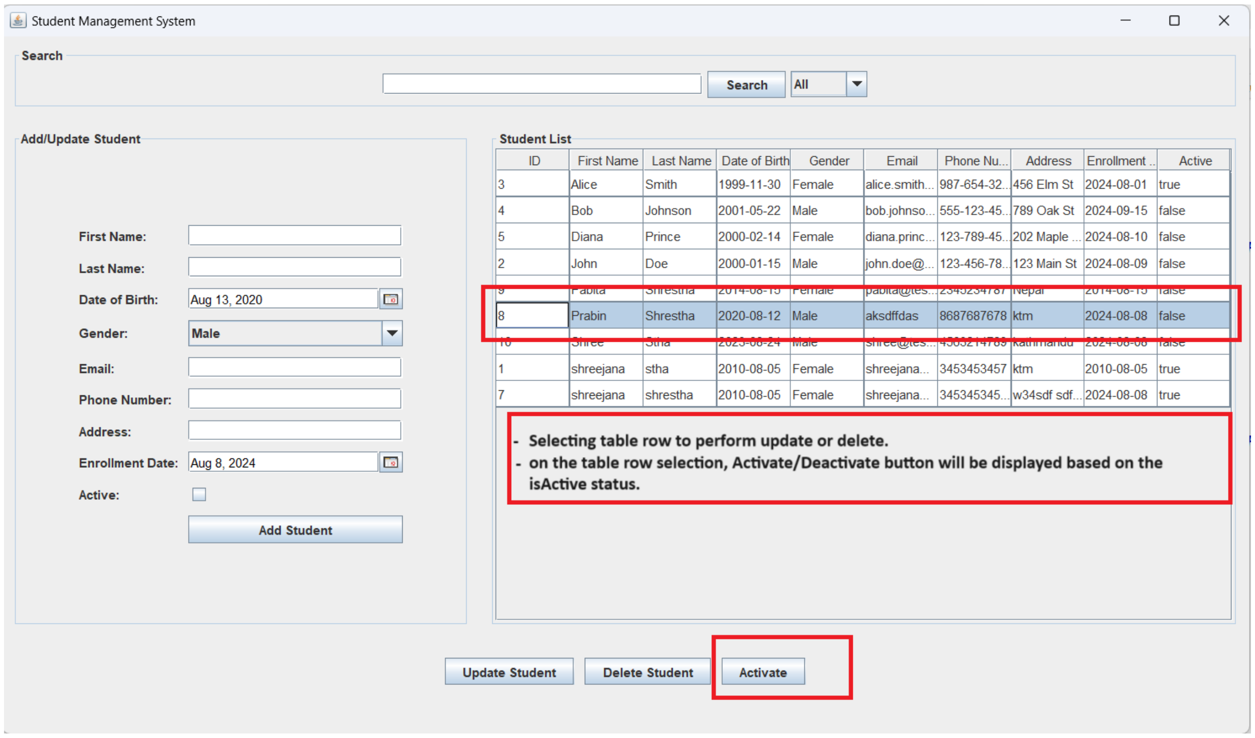
Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	true
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.com	8687687678	ktm	2020-08-12	false
11	Raman	Raj	2013-08-08	Male	raman@test....	2342345678	Mississauga	2024-08-07	true
10	Shree	Stha	2023-08-24	Male	shree@test.c...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana@g...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	false

# Selecting the record to perform update/delete or activate/deactivate operation



Student Management System

Search

Search | All

Add/Update Student

First Name:	
Last Name:	
Date of Birth:	Aug 13, 2020
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	Aug 8, 2024
Active:	☐

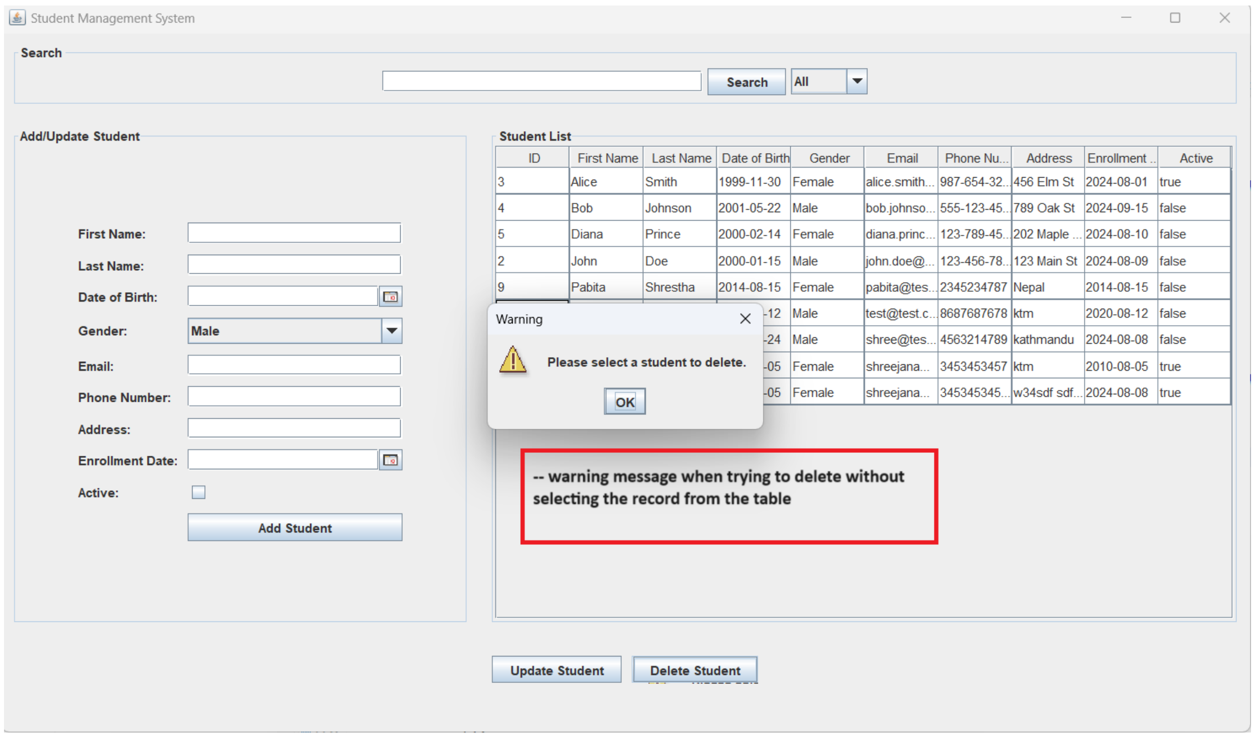
Add Student

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

- **Selecting table row to perform update or delete.**
- **on the table row selection, Activate/Deactivate button will be displayed based on the isActive status.**

# Error message if no record is not selected



Student Management System

Search

Search | All

Add/Update Student

First Name:	
Last Name:	
Date of Birth:	
Gender:	Male
Email:	
Phone Number:	
Address:	
Enrollment Date:	
Active:	☐

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	test@test.c...	8687687678	ktm	2020-08-12	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

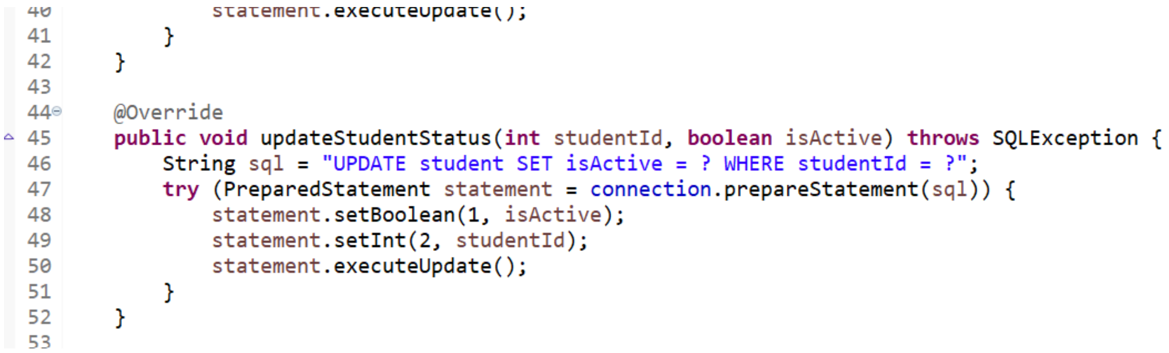
Warning

Please select a student to delete.

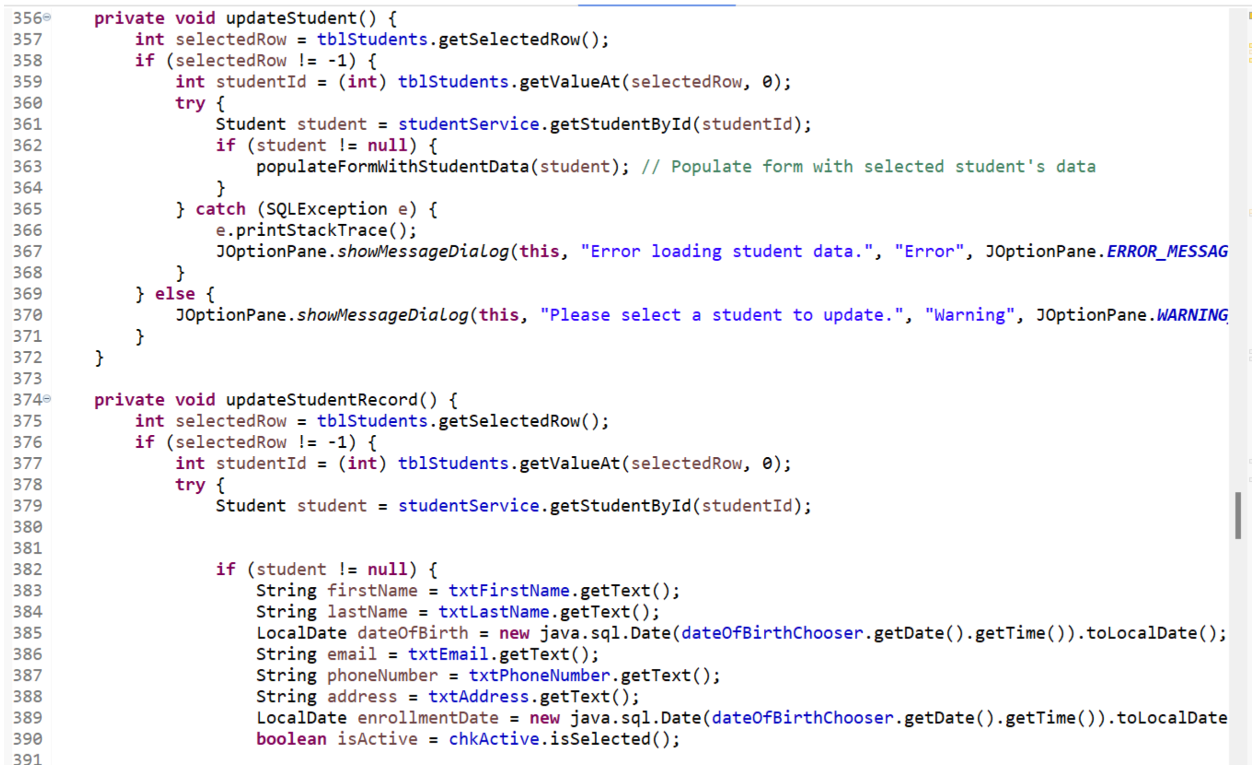
OK

-- warning message when trying to delete without selecting the record from the table

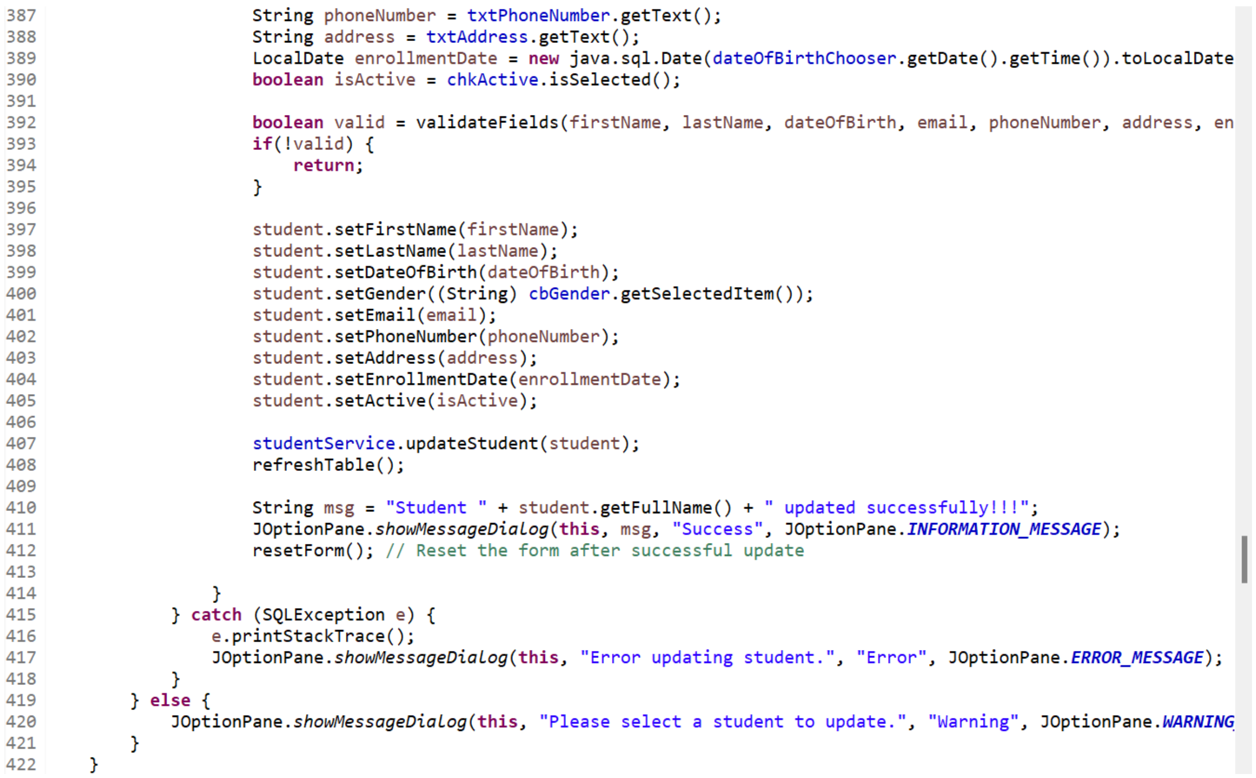
# Update Operation



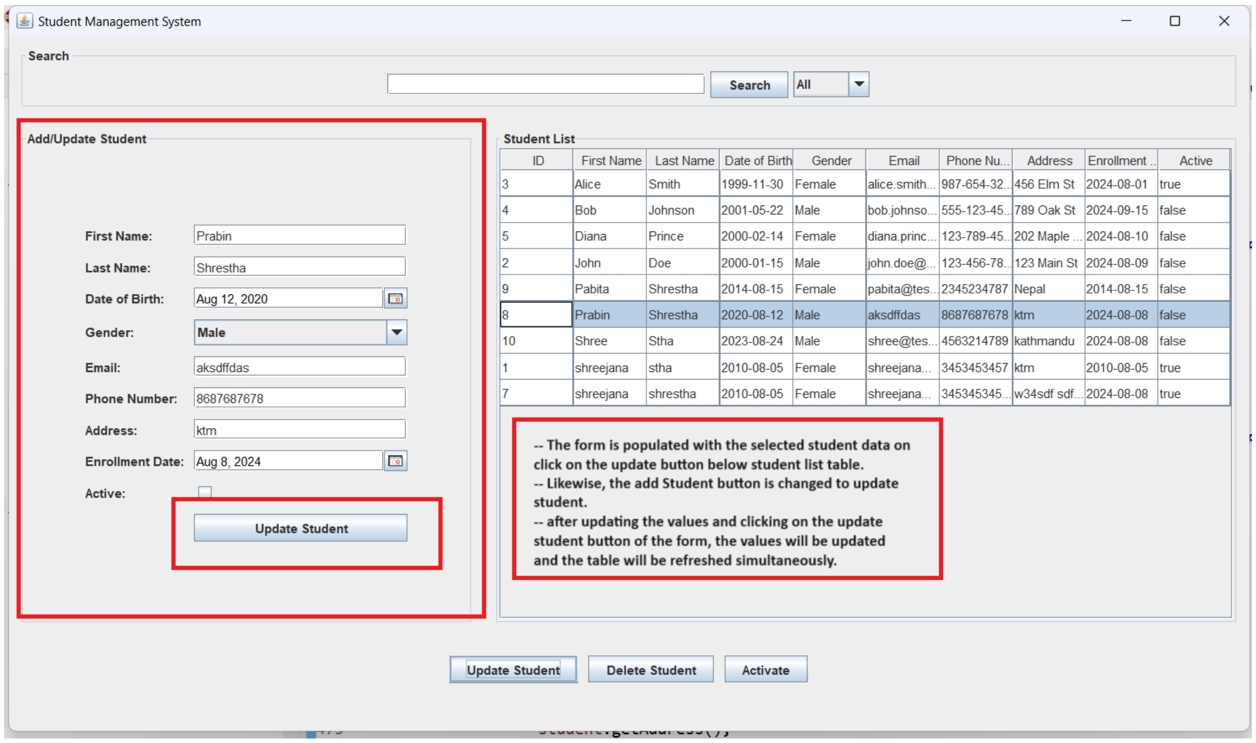
```
                statement.executeUpdate();
            }
        }

        @Override
        public void updateStudentStatus(int studentId, boolean isActive) throws SQLException {
            String sql = "UPDATE student SET isActive = ? WHERE studentId = ?";
            try (PreparedStatement statement = connection.prepareStatement(sql)) {
                statement.setBoolean(1, isActive);
                statement.setInt(2, studentId);
                statement.executeUpdate();
            }
        }

```



```java
    private void updateStudent() {
        int selectedRow = tblStudents.getSelectedRow();
        if (selectedRow != -1) {
            int studentId = (int) tblStudents.getValueAt(selectedRow, 0);
            try {
                Student student = studentService.getStudentById(studentId);
                if (student != null) {
                    populateFormWithStudentData(student); // Populate form with selected student's data
                }
            } catch (SQLException e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(this, "Error loading student data.", "Error", JOptionPane.ERROR_MESSAG
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select a student to update.", "Warning", JOptionPane.WARNING
        }
    }

    private void updateStudentRecord() {
        int selectedRow = tblStudents.getSelectedRow();
        if (selectedRow != -1) {
            int studentId = (int) tblStudents.getValueAt(selectedRow, 0);
            try {
                Student student = studentService.getStudentById(studentId);


                if (student != null) {
                    String firstName = txtFirstName.getText();
                    String lastName = txtLastName.getText();
                    LocalDate dateOfBirth = new java.sql.Date(dateOfBirthChooser.getDate().getTime()).toLocalDate();
                    String email = txtEmail.getText();
                    String phoneNumber = txtPhoneNumber.getText();
                    String address = txtAddress.getText();
                    LocalDate enrollmentDate = new java.sql.Date(dateOfBirthChooser.getDate().getTime()).toLocalDate
                    boolean isActive = chkActive.isSelected();

```



```java
                    String phoneNumber = txtPhoneNumber.getText();
                    String address = txtAddress.getText();
                    LocalDate enrollmentDate = new java.sql.Date(dateOfBirthChooser.getDate().getTime()).toLocalDate
                    boolean isActive = chkActive.isSelected();

                    boolean valid = validateFields(firstName, lastName, dateOfBirth, email, phoneNumber, address, en
                    if(!valid) {
                        return;
                    }

                    student.setFirstName(firstName);
                    student.setLastName(lastName);
                    student.setDateOfBirth(dateOfBirth);
                    student.setGender((String) cbGender.getSelectedItem());
                    student.setEmail(email);
                    student.setPhoneNumber(phoneNumber);
                    student.setAddress(address);
                    student.setEnrollmentDate(enrollmentDate);
                    student.setActive(isActive);

                    studentService.updateStudent(student);
                    refreshTable();

                    String msg = "Student " + student.getFullName() + " updated successfully!!!";
                    JOptionPane.showMessageDialog(this, msg, "Success", JOptionPane.INFORMATION_MESSAGE);
                    resetForm(); // Reset the form after successful update

                }
            } catch (SQLException e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(this, "Error updating student.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select a student to update.", "Warning", JOptionPane.WARNING
        }
    }
```



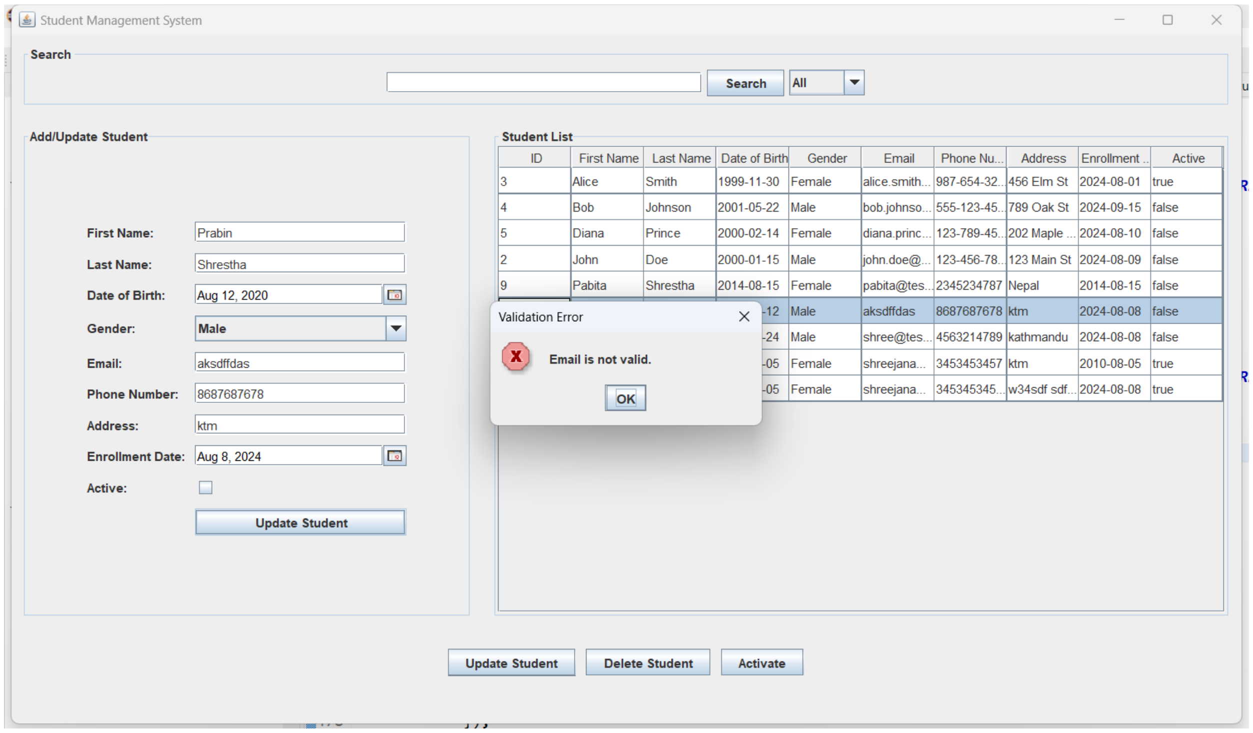
First Name:	Prabin
Last Name:	Shrestha
Date of Birth:	Aug 12, 2020
Gender:	Male
Email:	aksdffdas
Phone Number:	8687687678
Address:	ktm
Enrollment Date:	Aug 8, 2024
Active:	☐

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

-- The form is populated with the selected student data on click on the update button below student list table.
-- Likewise, the add Student button is changed to update student.
-- after updating the values and clicking on the update student button of the form, the values will be updated and the table will be refreshed simultaneously.

➔ With he "Update Student" functionality, admin user can modify an existing student's details. When a row in the student table is selected, the student's current data is populated into the form fields. The user can then edit these fields and submit the form. Upon submission, the data is validated to ensure correctness. The StudentService then interacts with the StudentDAO to update the student's record in the database using the modified data. After a successful update, the form is reset, the table is refreshed to show the updated information, and the button text reverts to "Add Student."

# Input validation



Student Management System

Search

Search | All

Add/Update Student

First Name:	Prabin
Last Name:	Shrestha
Date of Birth:	Aug 12, 2020
Gender:	Male
Email:	aksdffdas
Phone Number:	8687687678
Address:	ktm
Enrollment Date:	Aug 8, 2024
Active:	☐

Update Student

Student List

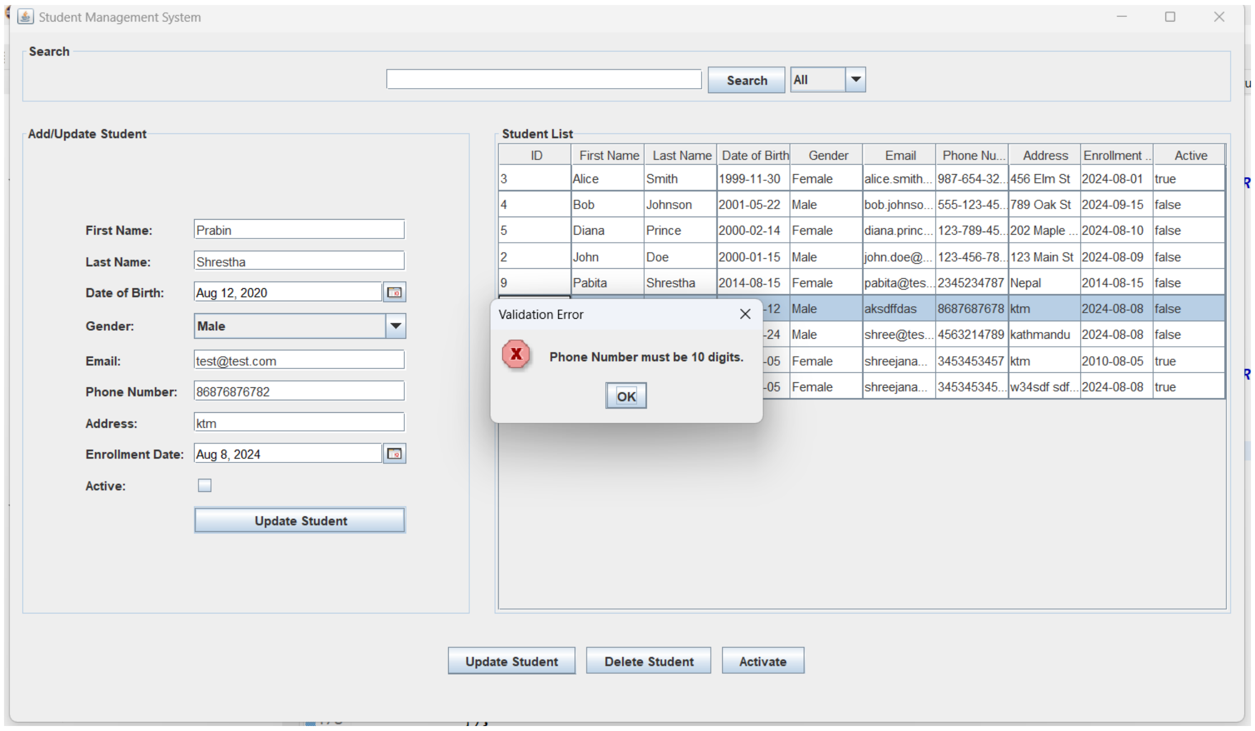
ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

Validation Error

Email is not valid.

OK

Update Student | Delete Student | Activate



Student Management System

Search

Search | All

Add/Update Student

First Name:	Prabin
Last Name:	Shrestha
Date of Birth:	Aug 12, 2020
Gender:	Male
Email:	test@test.com
Phone Number:	86876876782
Address:	ktm
Enrollment Date:	Aug 8, 2024
Active:	☐

Update Student

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
			-12	Male	aksdffdas	8687687678	ktm	2024-08-08	false
			-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

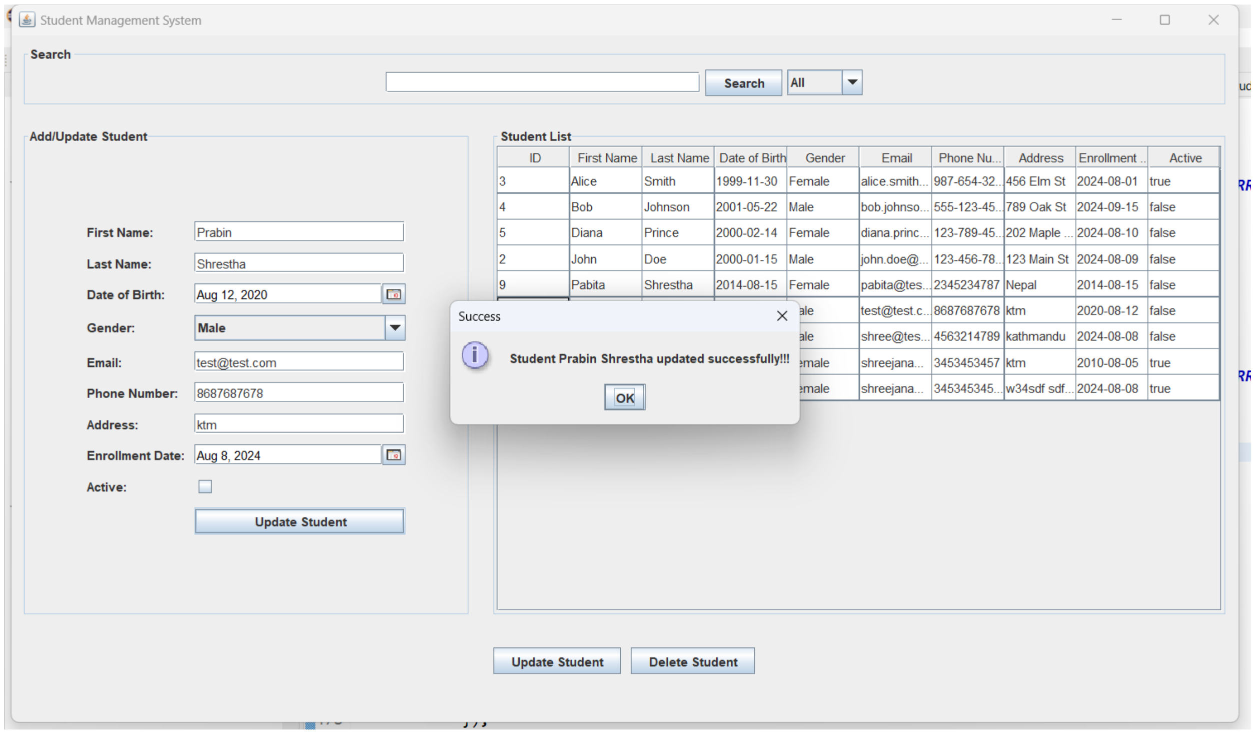
Validation Error

Phone Number must be 10 digits.

OK

Update Student | Delete Student | Activate

Successful update of data



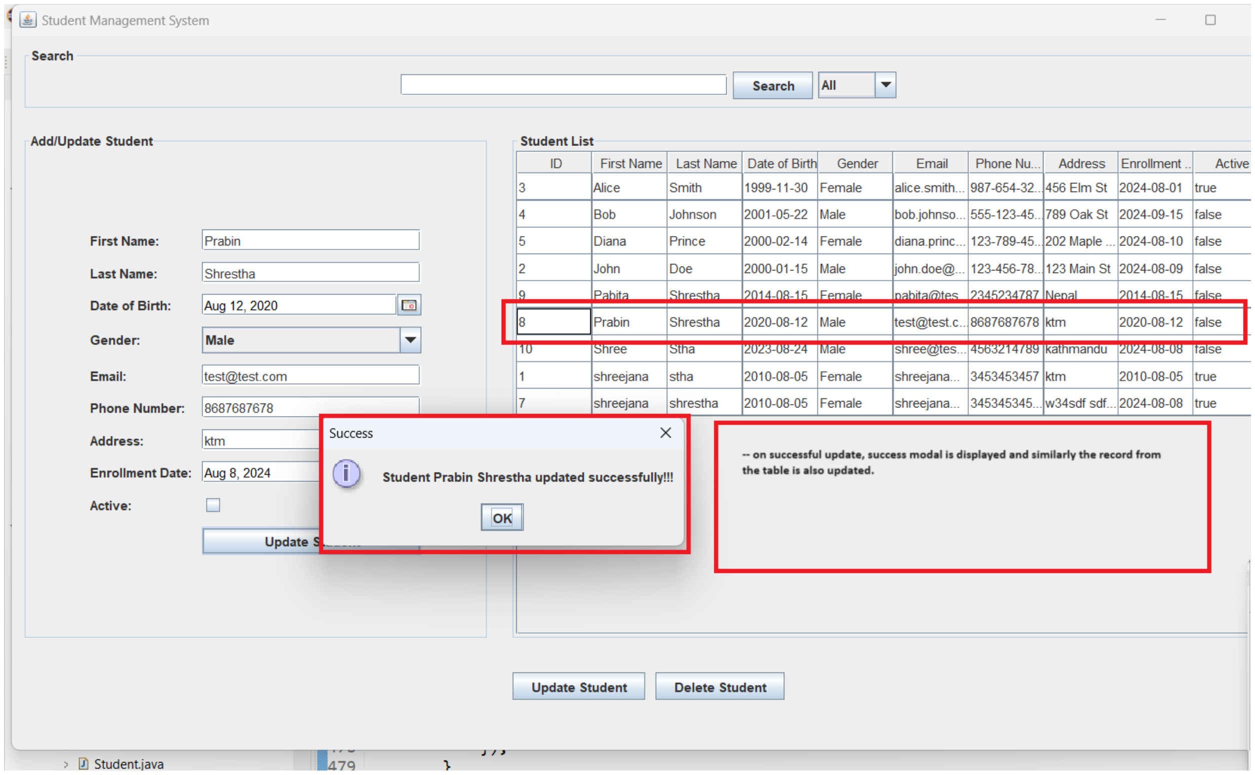
First Name:	Prabin
Last Name:	Shrestha
Date of Birth:	Aug 12, 2020
Gender:	Male
Email:	test@test.com
Phone Number:	8687687678
Address:	ktm
Enrollment Date:	Aug 8, 2024
Active:	☐

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
				ale	test@test.c...	8687687678	ktm	2020-08-12	false
				ale	shree@tes...	4563214789	kathmandu	2024-08-08	false
				emale	shreejana...	3453453457	ktm	2010-08-05	true
				emale	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

Success

Student Prabin Shrestha updated successfully!!!

OK



First Name:	Prabin
Last Name:	Shrestha
Date of Birth:	Aug 12, 2020
Gender:	Male
Email:	test@test.com
Phone Number:	8687687678
Address:	ktm
Enrollment Date:	Aug 8, 2024
Active:	☐

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnso...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.princ...	123-789-45...	202 Maple ...	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@tes...	2345234787	Nepal	2014-08-15	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.c...	8687687678	ktm	2020-08-12	false
10	Shree	Stha	2023-08-24	Male	shree@tes...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana...	345345345...	w34sdf sdf...	2024-08-08	true

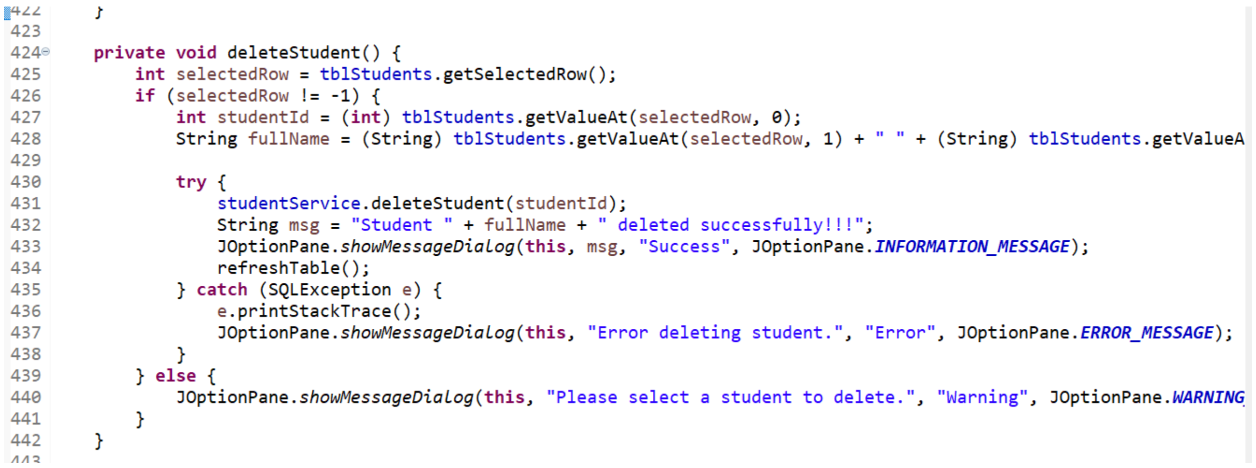
Success

Student Prabin Shrestha updated successfully!!!

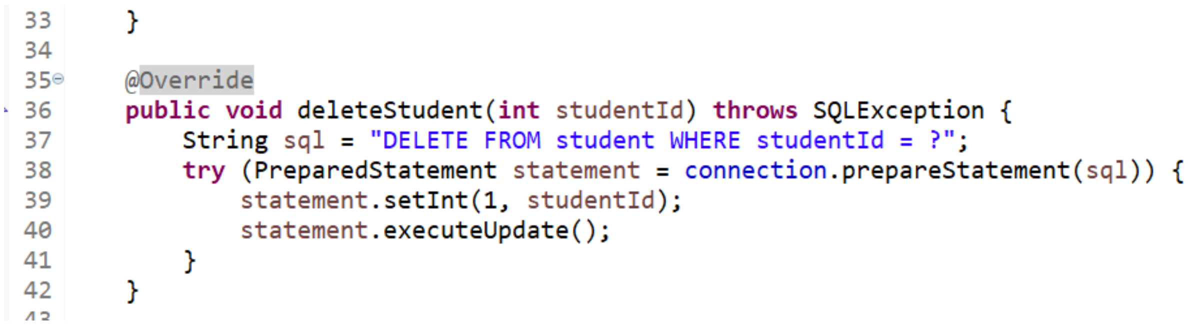
OK

-- on successful update, success modal is displayed and similarly the record from the table is also updated.

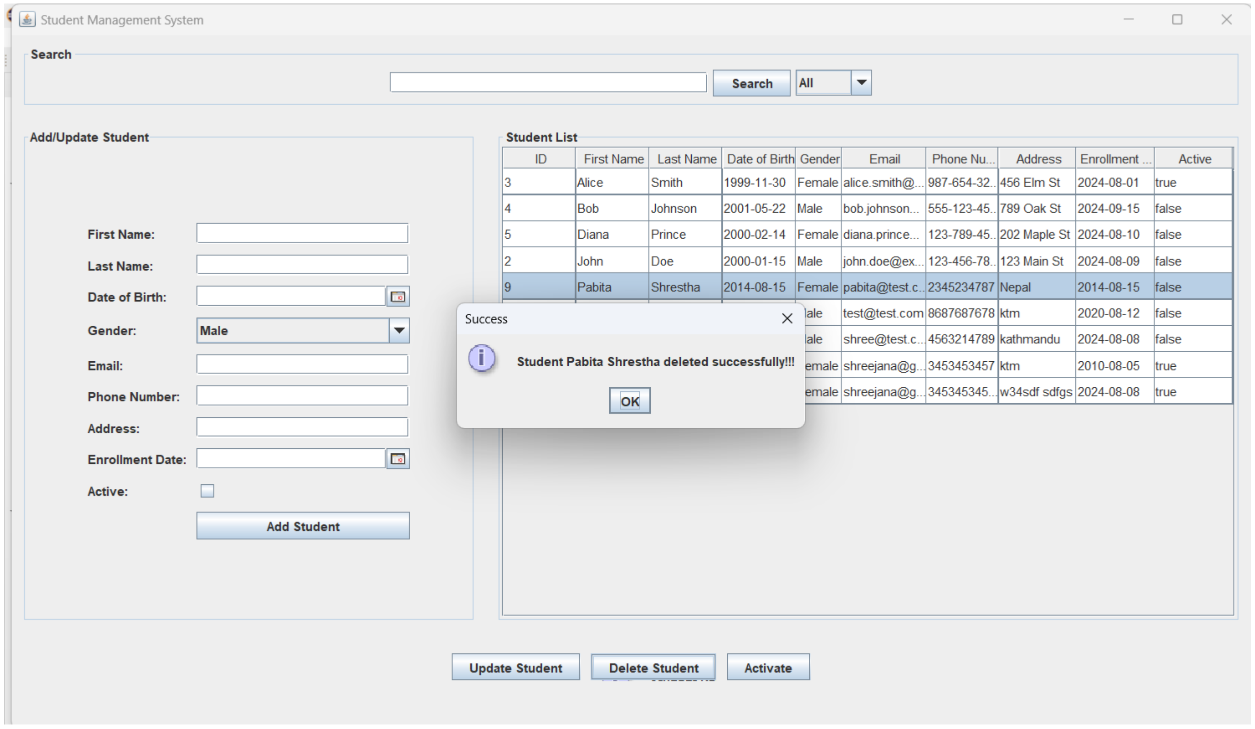
Delete operation



```java
    }

    private void deleteStudent() {
        int selectedRow = tblStudents.getSelectedRow();
        if (selectedRow != -1) {
            int studentId = (int) tblStudents.getValueAt(selectedRow, 0);
            String fullName = (String) tblStudents.getValueAt(selectedRow, 1) + " " + (String) tblStudents.getValueA

            try {
                studentService.deleteStudent(studentId);
                String msg = "Student " + fullName + " deleted successfully!!!";
                JOptionPane.showMessageDialog(this, msg, "Success", JOptionPane.INFORMATION_MESSAGE);
                refreshTable();
            } catch (SQLException e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(this, "Error deleting student.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select a student to delete.", "Warning", JOptionPane.WARNING
        }
    }
```



```java
    }

    @Override
    public void deleteStudent(int studentId) throws SQLException {
        String sql = "DELETE FROM student WHERE studentId = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, studentId);
            statement.executeUpdate();
        }
    }
```

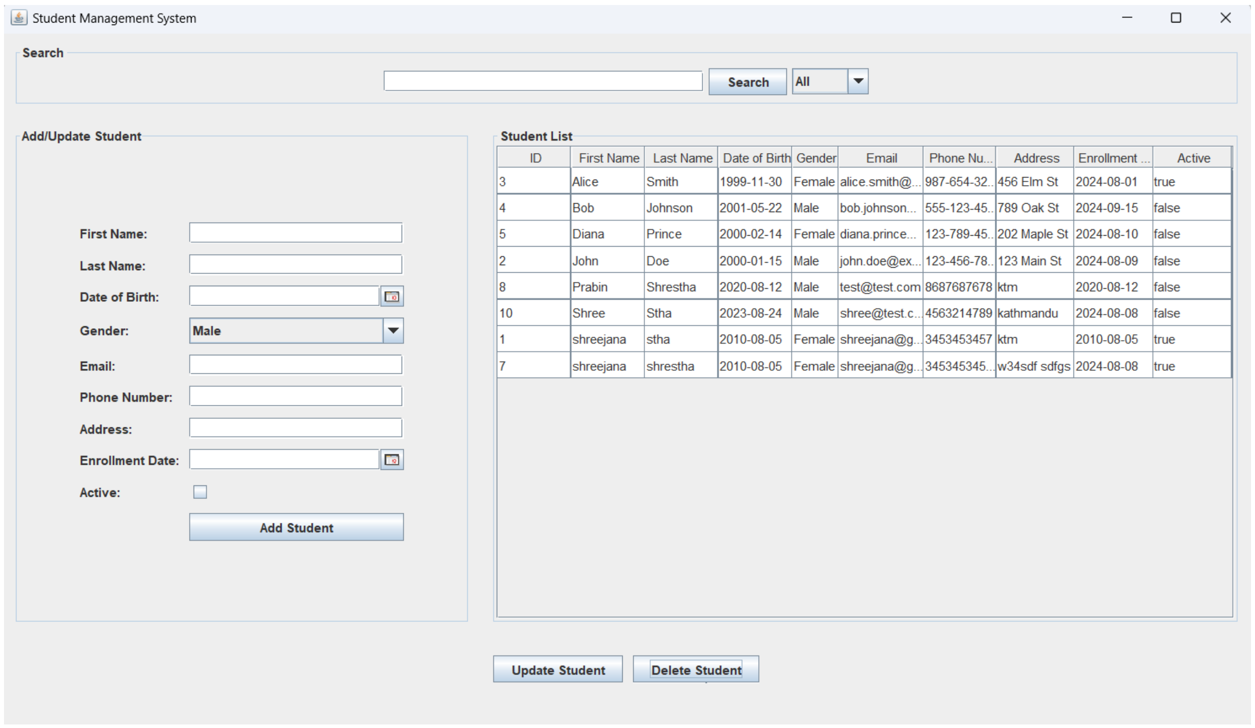


ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
9	Pabita	Shrestha	2014-08-15	Female	pabita@test.c...	2345234787	Nepal	2014-08-15	false
				ale	test@test.com	8687687678	ktm	2020-08-12	false
				ale	shree@test.c...	4563214789	kathmandu	2024-08-08	false
				emale	shreejana@g...	3453453457	ktm	2010-08-05	true
				emale	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	true

Student Pabita Shrestha deleted successfully!!!

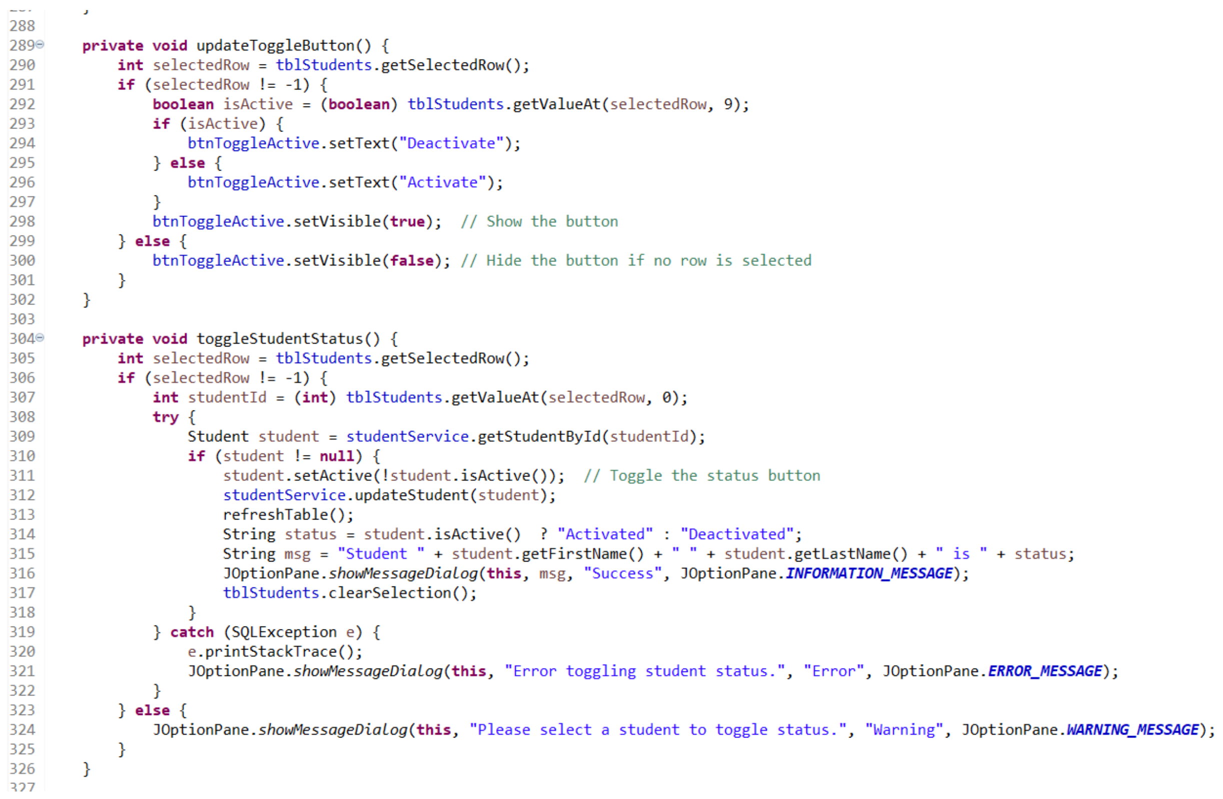
➔ The ‘Delete Student’ functionality remove a student's record from the database permanently. When a user selects a row in the student table and chooses the delete option, the system prompts the user for confirmation to prevent accidental deletions. If confirmed, the StudentService interacts with the StudentDAO to execute a delete operation on the selected student's record in the database. Once the record is deleted, the table is refreshed to reflect the removal, and the form is cleared to reset the interface for the next operation.

# Table after successful deletion

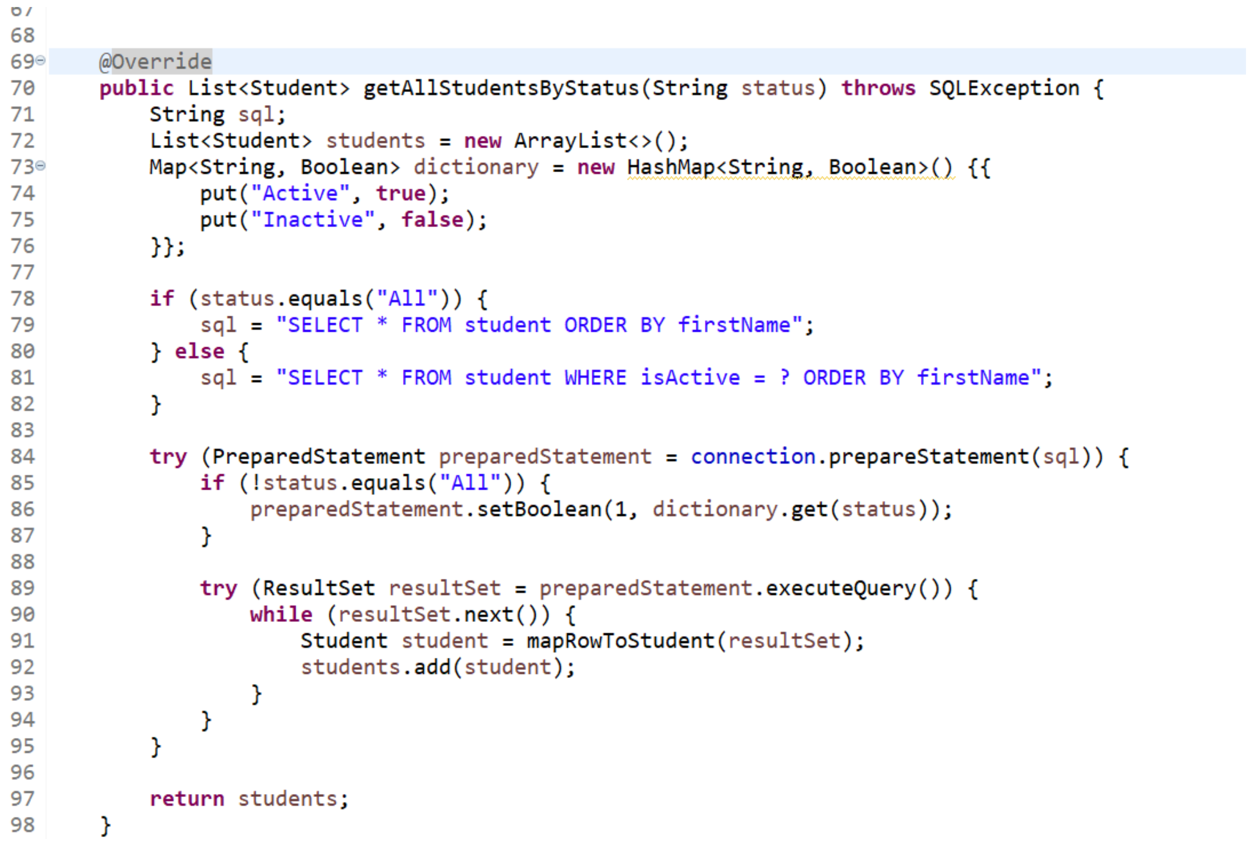


ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	false
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.com	8687687678	ktm	2020-08-12	false
10	Shree	Stha	2023-08-24	Male	shree@test.c...	4563214789	kathmandu	2024-08-08	false
1	shreejana	stha	2010-08-05	Female	shreejana@g...	3453453457	ktm	2010-08-05	true
7	shreejana	shrestha	2010-08-05	Female	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	true

Activate / Deactivate functionality

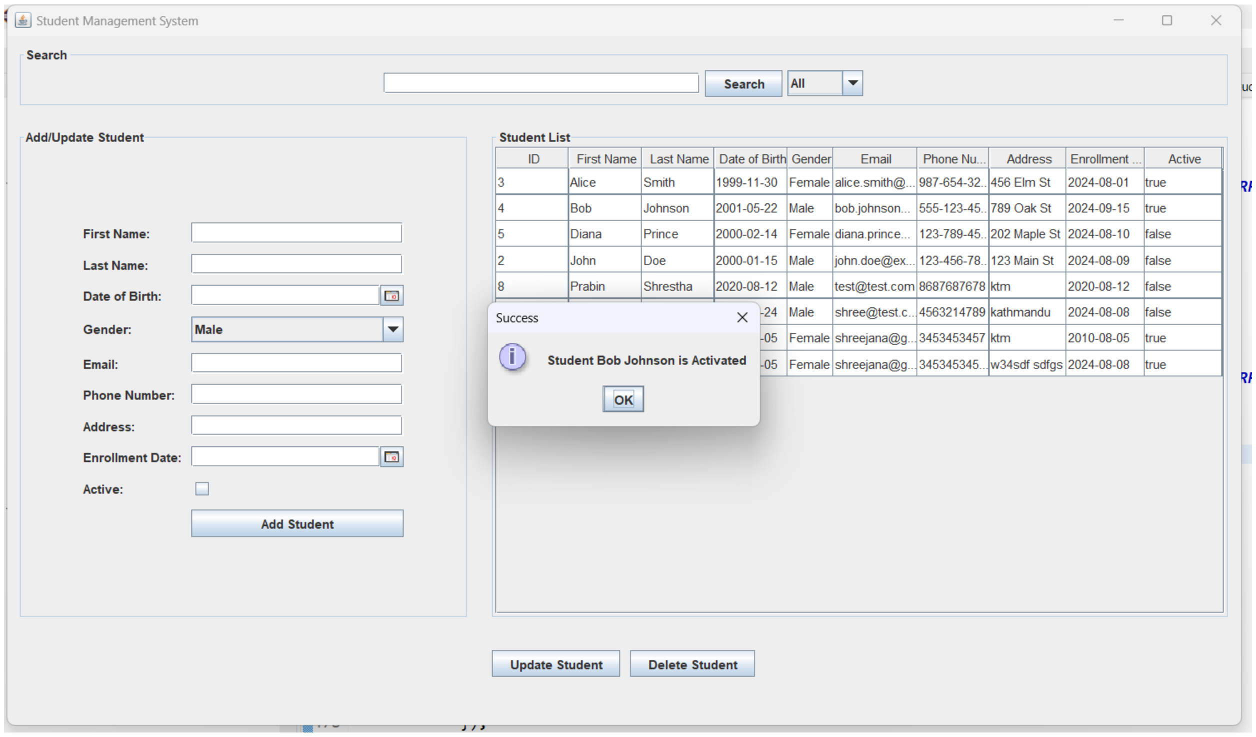


```java

    private void updateToggleButton() {
        int selectedRow = tblStudents.getSelectedRow();
        if (selectedRow != -1) {
            boolean isActive = (boolean) tblStudents.getValueAt(selectedRow, 9);
            if (isActive) {
                btnToggleActive.setText("Deactivate");
            } else {
                btnToggleActive.setText("Activate");
            }
            btnToggleActive.setVisible(true);  // Show the button
        } else {
            btnToggleActive.setVisible(false); // Hide the button if no row is selected
        }
    }

    private void toggleStudentStatus() {
        int selectedRow = tblStudents.getSelectedRow();
        if (selectedRow != -1) {
            int studentId = (int) tblStudents.getValueAt(selectedRow, 0);
            try {
                Student student = studentService.getStudentById(studentId);
                if (student != null) {
                    student.setActive(!student.isActive());  // Toggle the status button
                    studentService.updateStudent(student);
                    refreshTable();
                    String status = student.isActive()  ? "Activated" : "Deactivated";
                    String msg = "Student " + student.getFirstName() + " " + student.getLastName() + " is " + status;
                    JOptionPane.showMessageDialog(this, msg, "Success", JOptionPane.INFORMATION_MESSAGE);
                    tblStudents.clearSelection();
                }
            } catch (SQLException e) {
                e.printStackTrace();
                JOptionPane.showMessageDialog(this, "Error toggling student status.", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Please select a student to toggle status.", "Warning", JOptionPane.WARNING_MESSAGE);
        }
    }
```



```java

    @Override
    public List<Student> getAllStudentsByStatus(String status) throws SQLException {
        String sql;
        List<Student> students = new ArrayList<>();
        Map<String, Boolean> dictionary = new HashMap<String, Boolean>() {{
            put("Active", true);
            put("Inactive", false);
        }};

        if (status.equals("All")) {
            sql = "SELECT * FROM student ORDER BY firstName";
        } else {
            sql = "SELECT * FROM student WHERE isActive = ? ORDER BY firstName";
        }

        try (PreparedStatement preparedStatement = connection.prepareStatement(sql)) {
            if (!status.equals("All")) {
                preparedStatement.setBoolean(1, dictionary.get(status));
            }

            try (ResultSet resultSet = preparedStatement.executeQuery()) {
                while (resultSet.next()) {
                    Student student = mapRowToStudent(resultSet);
                    students.add(student);
                }
            }
        }

        return students;
    }
```

- ➔ We have activate/deactivate functionality and the corresponding logic to show the data in the table based on the status selected from the drop down filter.
- ➔ It allows admin to activate or deactivate the students which can be achieved from the toggle of the button on the bottom section. If the student is currently active, clicking the "Deactivate" button will set the isActive field in the database to false. Conversely, if the student is inactive, clicking the "Activate" button will set the isActive field to true. The system uses the StudentService to call the StudentDAO, which executes the update query to toggle the isActive status in the database.



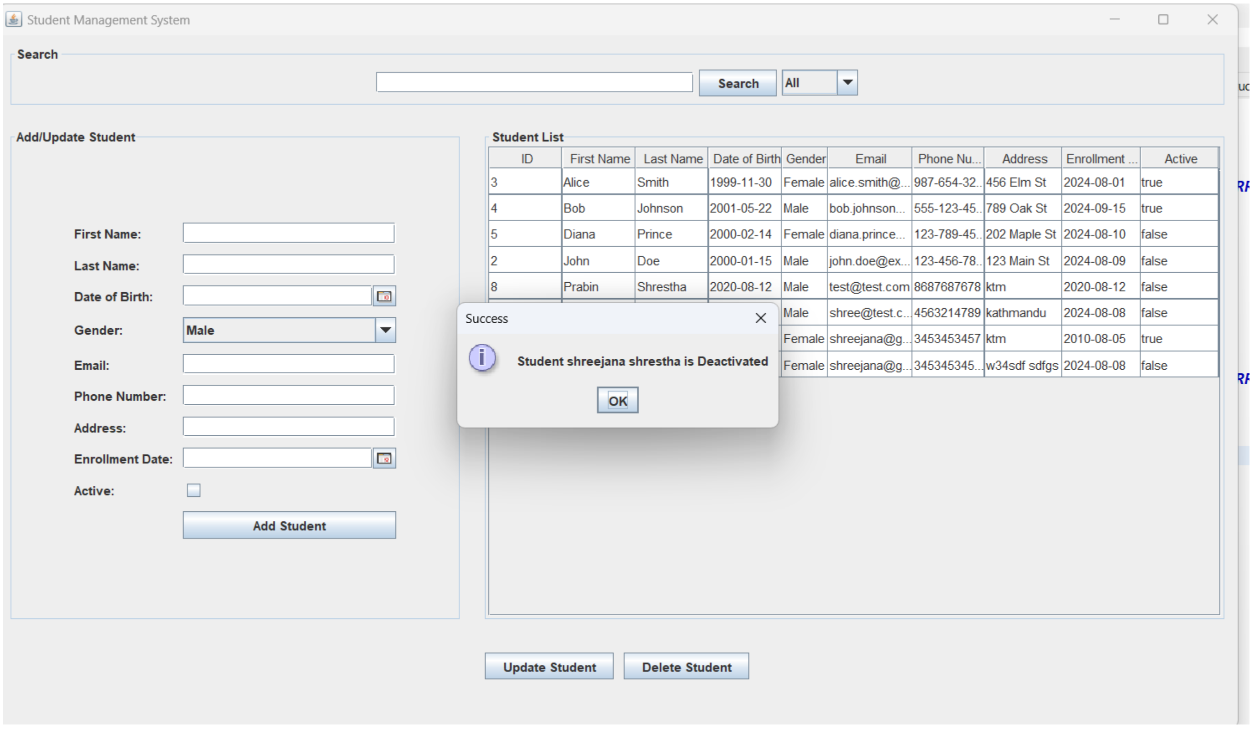
First Name:
Last Name:
Date of Birth:
Gender: Male
Email:
Phone Number:
Address:
Enrollment Date:
Active:

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	true
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.com	8687687678	ktm	2020-08-12	false
			-24	Male	shree@test.c...	4563214789	kathmandu	2024-08-08	false
			-05	Female	shreejana@g...	3453453457	ktm	2010-08-05	true
			-05	Female	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	true

Success

Student Bob Johnson is Activated



First Name:
Last Name:
Date of Birth:
Gender: Male
Email:
Phone Number:
Address:
Enrollment Date:
Active:

Student List

ID	First Name	Last Name	Date of Birth	Gender	Email	Phone Nu...	Address	Enrollment ...	Active
3	Alice	Smith	1999-11-30	Female	alice.smith@...	987-654-32...	456 Elm St	2024-08-01	true
4	Bob	Johnson	2001-05-22	Male	bob.johnson...	555-123-45...	789 Oak St	2024-09-15	true
5	Diana	Prince	2000-02-14	Female	diana.prince...	123-789-45...	202 Maple St	2024-08-10	false
2	John	Doe	2000-01-15	Male	john.doe@ex...	123-456-78...	123 Main St	2024-08-09	false
8	Prabin	Shrestha	2020-08-12	Male	test@test.com	8687687678	ktm	2020-08-12	false
				Male	shree@test.c...	4563214789	kathmandu	2024-08-08	false
				Female	shreejana@g...	3453453457	ktm	2010-08-05	true
				Female	shreejana@g...	345345345...	w34sdf sdfgs	2024-08-08	false

Success

Student shreejana shrestha is Deactivated