📘 STUDENT INFORMATION SYSTEM

Java Console Application

Submitted By:

Name: Samruddhi Auti

Internship Program: The Developers Arena

---

1. Project Title

Student Information System using Java

TABLE OF CONTENTS

---

1. Introduction

The Student Information System is a Java-based, menu-driven console application developed as part of an internship task.
It allows efficient management of student records using Object-Oriented Programming principles.

---

2. Project Objective

The objective of this project is to:

- Store student details securely

- Perform CRUD operations (Create, Read, Update, Delete)
- Demonstrate Java OOP concepts such as encapsulation and collections
- Build a user-friendly menu-driven system

---

3. Problem Statement

Manual handling of student records is inefficient and prone to errors.
This project provides an automated solution to manage student information such as:

- Student ID
- Name
- Age
- Grade
- Contact details

---

4. System Requirements

Hardware Requirements

- Computer / Laptop
- Minimum 4 GB RAM

Software Requirements

- Java JDK 8 or higher
- IDE: VS Code / Eclipse / IntelliJ
- Operating System: Windows / Linux / macOS

---

5. Technologies Used

- **Programming Language:** Java
- **Concepts:** Object-Oriented Programming (OOP)
- **Collections Framework:** ArrayList
- **Input Handling:** Scanner class

---

6. Project Description

The application consists of two main classes:

1. **Student Class**

- Stores student data
- Implements encapsulation using private variables and public getters/setters

2. **StudentInformationSystem Class**
    - Provides menu-driven interface
    - Handles user interaction and operations

Student data is stored dynamically using an ArrayList.

---

7. System Features

- Add Student
- View All Students
- Update Student Details
- Delete Student Record
- Search Student by ID or Name
- Input Validation for Age
- Formatted Output Display

---

8. Program Design (Classes Used)

Student Class

- Attributes: studentID, name, age, grade, contact
- Methods: getters, setters, display()

StudentInformationSystem Class

- Menu-driven main method
- CRUD operation methods
- Uses ArrayList for storage

---

9. How to Run the Application

1. Compile the program:
2. javac Student.java StudentInformationSystem.java
3. Run the application:
4. java StudentInformationSystem
5. Select options from the menu

6. Enter required details
7. Exit safely using option 6

---

10. Sample Data

Student ID	Name	Age	Grade	Contact
101	Rahul	20	85.00	9876543210
102	Priya	19	91.82	9123456789
103	Amit	21	78.50	9988776655

---

11. Test Cases

Test Case	Input	Expected Output
Add Student	Valid data	Student added successfully
Add Student	Age = -5	Invalid Age
View Students	Menu option 2	Displays all records
Search by ID	ID = 101	Student details shown
Search by Name	Name = Priya	Student details shown
Update Student	Valid ID	Student updated successfully
Delete Student	Valid ID	Student deleted successfully
Delete Student	Invalid ID	Student not found

---

12. Conclusion

The Student Information System successfully fulfills the requirements by providing a simple yet effective solution for managing student records.
The project demonstrates practical implementation of Java programming concepts and prepares the developer for real-world application development.

---

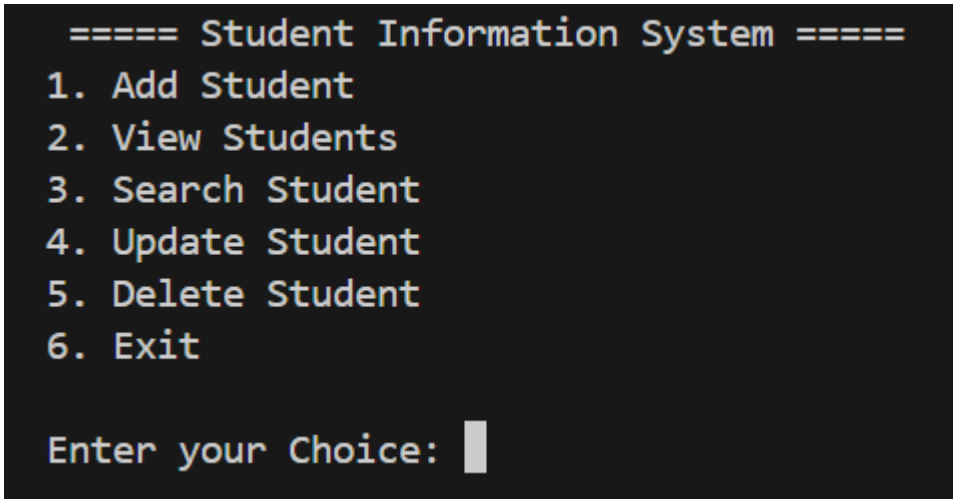
13. Screenshots / Output Snapshots

This section contains screenshots of the working Student Information System application, demonstrating different functionalities of the system.

This section contains screenshots of the working Student Information System application, demonstrating different functionalities of the system.

Screenshot 1: Main Menu Display

Description:
This screenshot shows the main menu of the Student Information System when the program starts. The menu provides options to add, view, update, delete, search student records, and exit the application.



Screenshot 2: Add Student

Description:
This screenshot shows the process of adding a new student by entering student ID, name, age, grade, and contact details.

```
=== ADD NEW STUDENT ===
Enter Student ID: 101
Enter Name: Rahul Sharma
Enter Age: 20
Enter Grade: 85.00
Enter Contact: 9876543210

Student Added Successfully

 ===== Student Information System =====
1. Add Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit

Enter your Choice: 1

=== ADD NEW STUDENT ===
Enter Student ID: 102
Enter Name: Priya Patil
Enter Age: 19
Enter Grade: 91.67
Enter Contact: 9123456789

Student Added Successfully
```

Screenshot 3: View All Students

Description:
This screenshot displays all stored student records in a tabular format using formatted output.

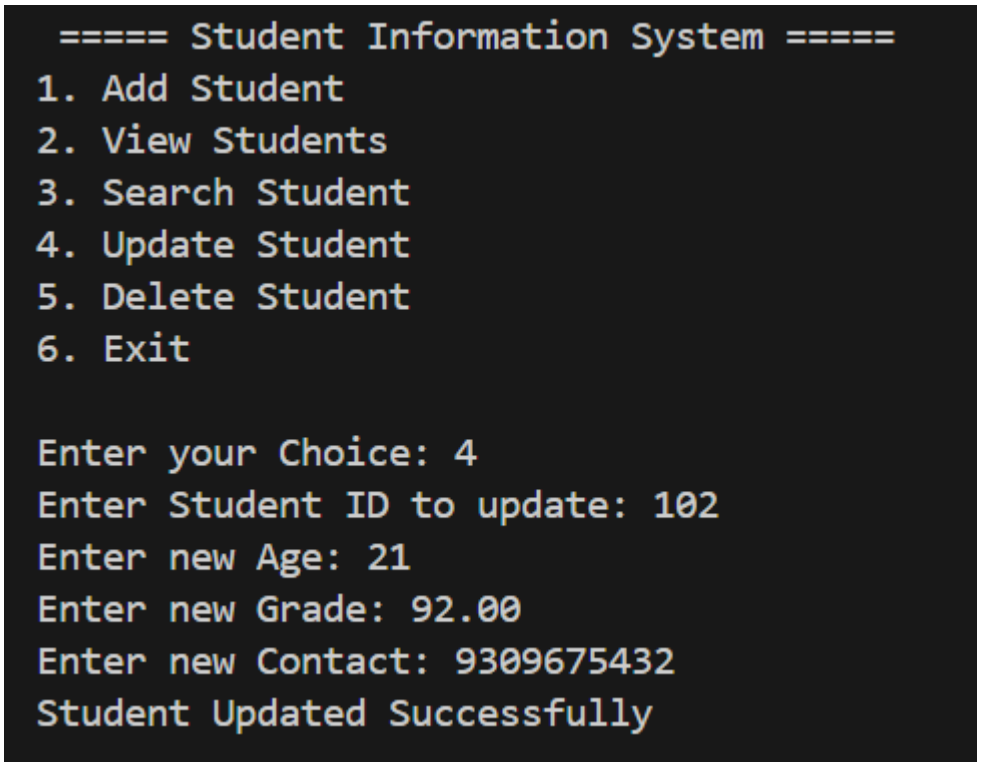
```
Enter your Choice: 2
ID          Name            Age   Grade Contact
-------------------------------------------------
101         Rahul Sharma    20    85.00 9876543210
102         Priya Patil     19    91.67 9123456789
```

Screenshot 4: Search Student by ID

Description:
This screenshot demonstrates searching a student record using the student ID and displaying the corresponding details.

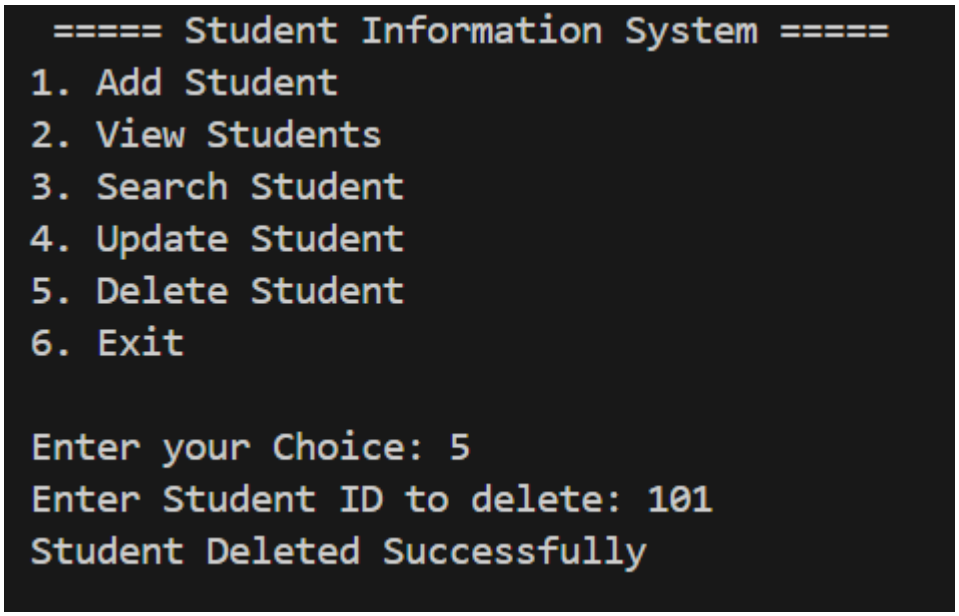
Screenshot 5: Update Student

Description:
This screenshot shows updating an existing student's age, grade, and contact details using the student ID.



Screenshot 6: Delete Student

Description:
This screenshot demonstrates deletion of a student record using the student ID.



Screenshot 7: Validation / Error Handling

Description:
This screenshot shows validation handling when invalid input (such as negative age) is entered.

```
 ===== Student Information System =====
1. Add Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit

Enter your Choice: 1

=== ADD NEW STUDENT ===
Enter Student ID: 104
Enter Name: Sneha Kulkarni
Enter Age: -22
Invalid Age

 ===== Student Information System =====
1. Add Student
2. View Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit

Enter your Choice: 6
Exiting Program...
```

Screenshots Summary

The above screenshots verify that:

- The application runs successfully
- All menu options work correctly
- Input validation is implemented
- CRUD operations function as expected