

Student Management System

This document explains the Student Management System logically from a system perspective. Each screenshot is followed by a short, clean explanation focusing on how the system works, how data flows, and how different components interact. The explanations are not code-heavy, but small syntax references are used where necessary for clarity.

Student List Dashboard

Student Management System				
ID	Name	Email	Course	Actions
8	Riya	bhujel@gmail.com	BBA	Edit Delete
5	Nabin Lama	nabin@gmail.com	BCA	Edit Delete
3	Hari Thapa	hari@gmail.com	BIT	Edit Delete
2	Sita Karki	sita@gmail.com	BCA	Edit Delete
1	Ram Sharma	ram@gmail.com	BSc CSIT	Edit Delete

This is the main dashboard of the system. It displays all student records retrieved from the database. The system performs a READ operation by fetching student data using a database connection. Each row represents one student entity, showing structured data like name, email, and course. The Actions column allows navigation to update or delete operations, making this page the central control point.

Add New Student Form

The screenshot shows a modal window titled "Add New Student". It contains three input fields: "Name", "Email", and "Course", each with a corresponding text input box below it. A large blue button labeled "Add Student" is centered at the bottom. At the bottom left of the modal, there is a link "← Back to Student List". The background of the entire image is a light grey gradient.

Add New Student

Name

Email

Course

Add Student

[← Back to Student List](#)

This screen handles the CREATE operation. The system collects user input through a form interface. Once submitted, the data is validated and sent to the backend for insertion into the database. This separation between form display and data processing improves system clarity and maintainability.

Edit Student Form

Edit Student

Name

Email

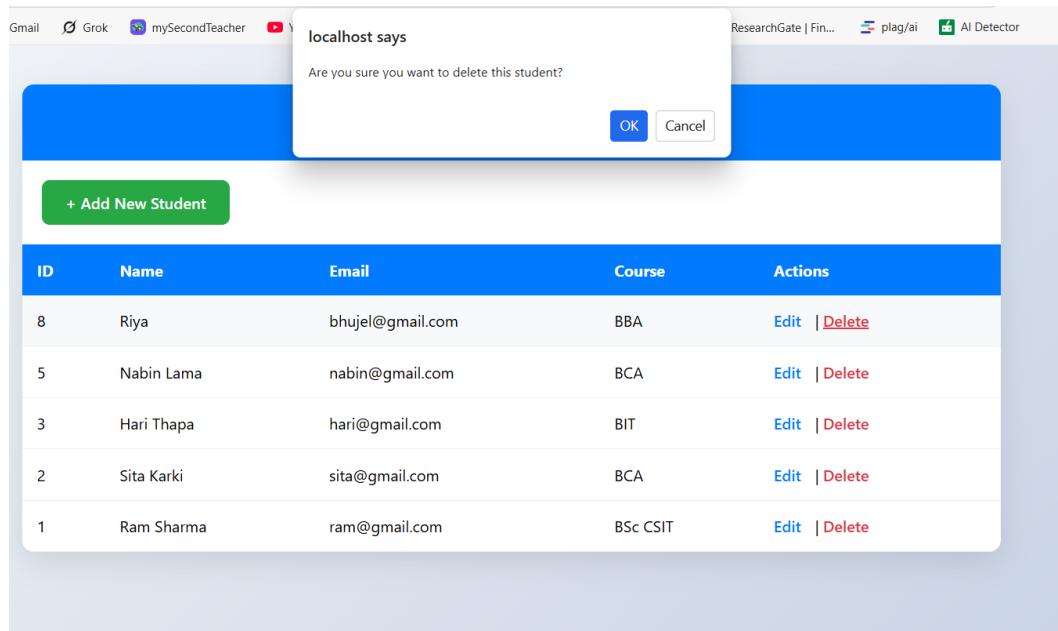
Course

[Update Student](#)

[← Back to Student List](#)

This page supports the UPDATE operation. Existing student data is first retrieved using a unique identifier (ID). The form is pre-filled to avoid data loss and improve user experience. After modification, the system updates only the selected record, ensuring controlled data consistency.

Delete Confirmation Dialog



This confirmation dialog acts as a safety mechanism before performing the DELETE operation. The system uses a confirmation check to prevent accidental data loss. Only after user approval does the system execute the delete request and refresh the student list.

System Flow Summary

The system follows a clear CRUD-based architecture. User actions from the interface trigger backend logic, which interacts with the database and returns updated results to the interface. This logical separation between presentation, processing, and data storage makes the system efficient, easy to understand, and suitable for beginner-level backend development projects.