

Student Management System (MERN CRUD)

A simple MERN stack web app for managing student records (Create, Read, Update, Delete).

Features

- Add a student (Full Name, Email, Course)
- View students in a table/card layout
- Edit student details
- Delete a student

Tech Stack

- MongoDB
- Express.js
- React.js (Html + Css)
- Node.js

Project Structure (example)

```
student-management-system/
|
|   └── backend/      # Node.js + Express + MongoDB API
|       |
|       |   └── config/
|       |       └── db.js      # MongoDB connection setup
|       |
|       └── controllers/
```

```
  | | └ studentController.js # Logic for CRUD operations
  | |
  | └─ models/
  |   └ Student.js      # Mongoose schema for students
  | |
  | └─ routes/
  |   └ studentRoutes.js # API routes for student endpoints
  | |
  | └─ server.js        # Main backend entry point
  | └─ .env.example      # Example environment variables
  | └─ package.json      # Backend dependencies and scripts
  └─ .gitignore

  └─ frontend/          # React (Vite) Frontend
    └─ src/
      └─ components/
        └─ StudentForm.jsx # Form to add/edit students
        └─ StudentList.jsx # Displays students table
      └─ api/
        └─ studentApi.js   # Axios configuration & API calls
      └─ App.jsx           # Main application layout
      └─ main.jsx          # React entry point
      └─ index.css         # Styling
    └─ public/
```

```
|   |-- package.json      # Frontend dependencies
|   └── vite.config.js
|
|   ├── README.md
|
|   └── .gitignore
```

Database Schema (MongoDB)

Each student document looks like:

```
{  
  _id: ObjectId,  
  fullName: String,  // required  
  email: String,    // required, unique  
  course: String,   // required  
  createdAt: Date,  
  updatedAt: Date  
}
```

##Setup & Run

1) Backend

```
cd backend
```

```
npm install
```

Create a .env file:

PORT=5000

MONGO_URI=mongodb://127.0.0.1:27017/student_management

Run:

npm run dev

2) Frontend

cd .. frontend

npm install

npm run dev