**COLLEGE CODE :** 8203

**COLLEGE NAME:** A.V.C. College of Engneering

**DEPARTMENT:** Computer Science And Engineering

**STUDENT NM-ID:**

| D21AA6EA77E172BB15648DAD458441A9 |
| --- |

**ROLL NO:** 820323104075

**DATE : 08/09/2025**

**Completed the project named as Phase -2 TECHNOLOGY**

**PROJECT NAME:** Student Grading System

**SUBMITTED BY,**

**NAME:** S.Pooja

**MOBILE NO**: 6379 437 780

**Phase 2 — Solution Design & Architecture**

## 1. Tech Stack Selection

**Frontend**: React.js (for dynamic UI and smooth user experience)

**Backend**: Node.js with Express.js (for REST API services)

**Database**: MongoDB (NoSQL database for flexible schema and fast access)

**Authentication**: JWT (JSON Web Tokens) for secure login

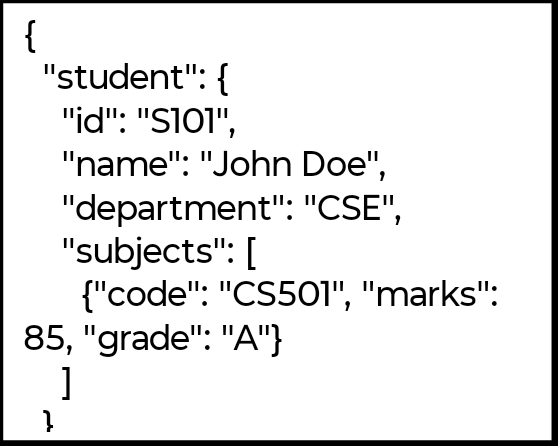
**Deployment**: AWS / Heroku / Render for hosting backend, Netlify/Vercel for frontend

**2. UI Structure / API Schema Design**

1. **UI Structure:**

* Login Page (Admin/Teacher/Student)
* Dashboard (role-based view)
* Student Details Page (view/add/edit student info)
* Grade Entry Page (teachers enter/update grades)
* Grade Report Page (students view performance, GPA)

1. **API Schema Design:**



**3. Data Handling Approach**

* Data validation at both frontend (form validation) and backend (API validation).
* Use of **Mongoose schemas** to ensure structured student and grade records.
* Role-based access control (admin can manage users, teachers manage grades, students view results).
* Error handling with proper HTTP status codes.

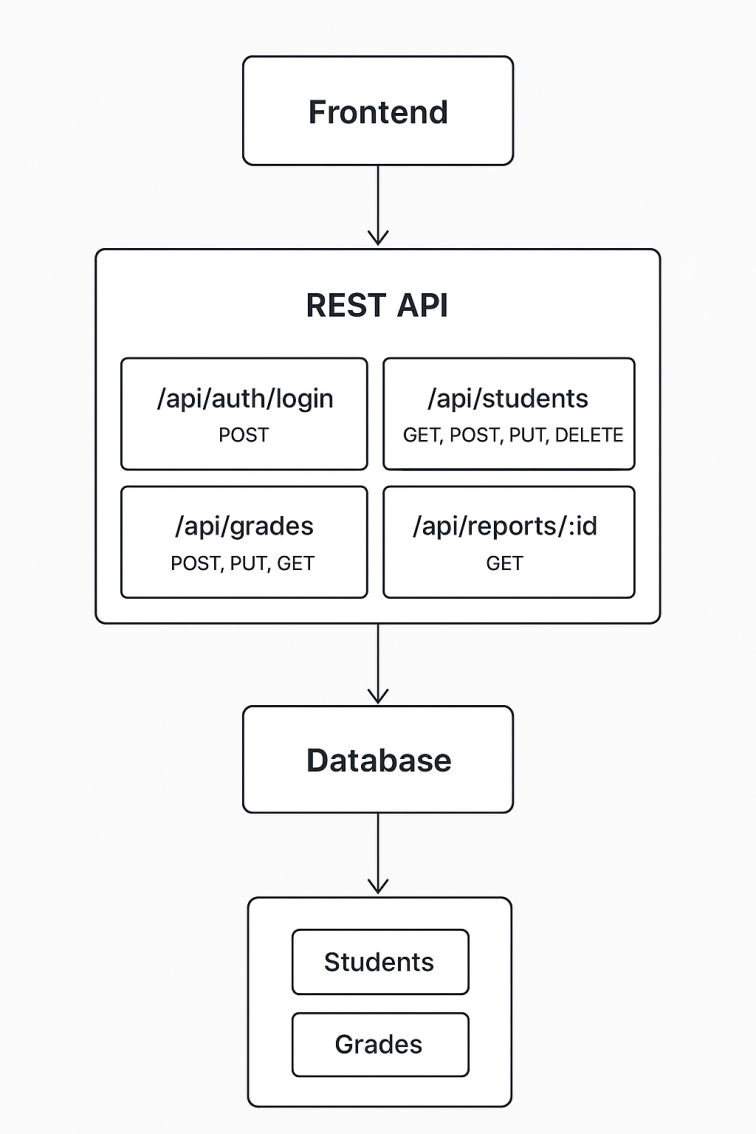
**4. Component / Module Diagram**

**Authentication Module** (login, registration, role management)

**Student Management Module** (add/edit student details)

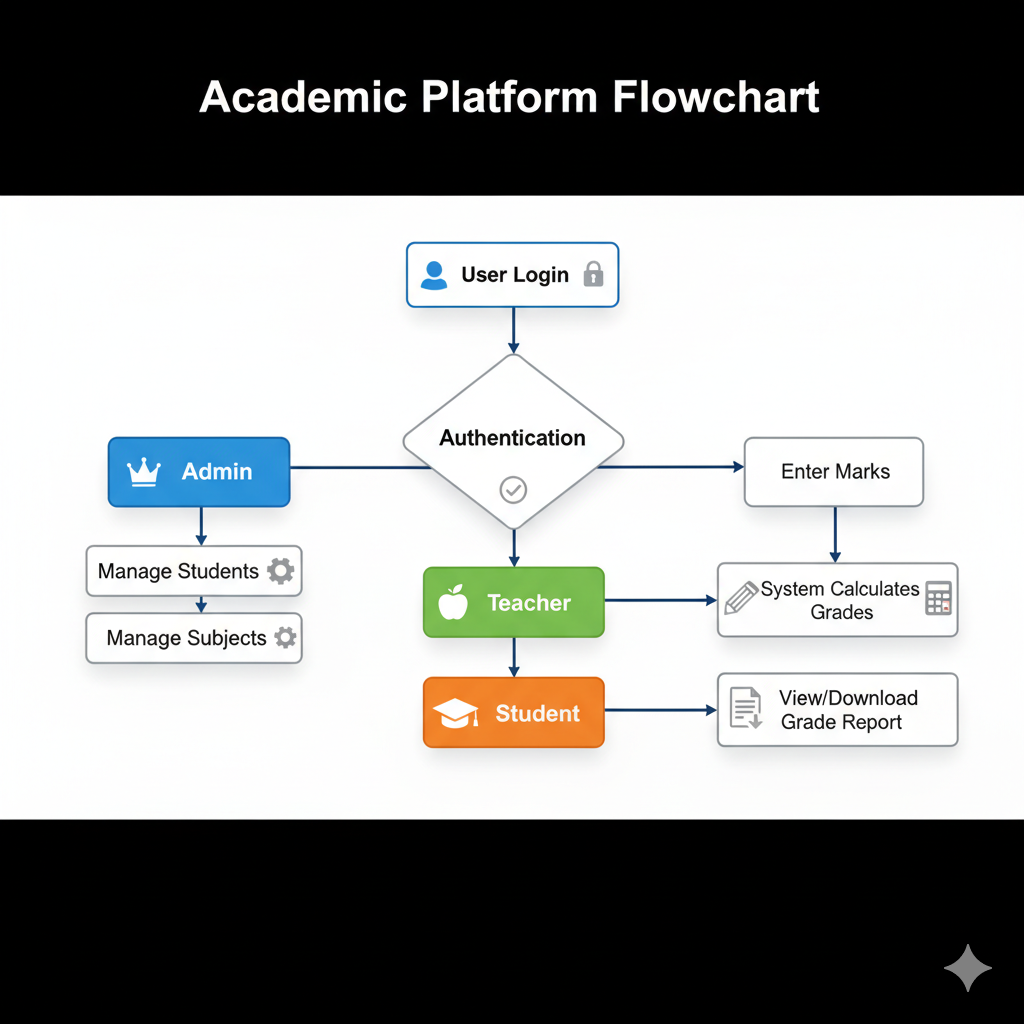
**Grading Module** (input/update grades)

**Report Module** (generate grade reports, GPA, performance analysis)

****

**System architecture**

**5. Basic Flow Diagram**

****