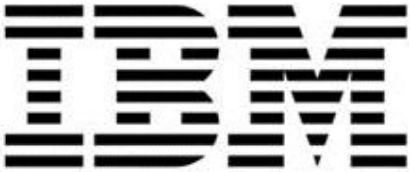

COLLEGE CODE : 8203

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

DEPARTMENT: Computer Science And Engineering

STUDENT NM-ID: B0808D888884AEABB134C815048F6CE0

ROLL NO: 820323104078

DATE: 08/09/2025

Completed the project named as Phase -3 TECHNOLOGY

PROJECT NAME: Student Grading System

SUBMITTED BY,

NAME: M.PRASANNA DEVI

MOBILE NO:9095267006

Phase-3 MVP Implementation

1. Project Setup

- Initialized **Node.js + Express.js** backend project structure.
- Installed required dependencies (**express, mysql2, sequelize/knex, cors, dotenv**).
- Configured **MySQL database connection** and created required schemas (students, subjects, grades, users)
- Setup project folder structure (controllers, routes, models, middlewares)..

2. Core Features Implementation

- Implemented **user roles**: Admin, Teacher, Student.
- Teachers can **add/update student grades**.
- Students can **view their results** securely.
- Admin can **manage users, subjects, and grading policies**.
- RESTful API endpoints created for CRUD operations (Create, Read, Update, Delete).

3. Data Storage (Local State / Database)

- Used **MySQL** database for persistent storage.
- Designed relational schema:
  - ✓ **Users** table (Admin, Teacher, Student).
  - ✓ **Students** table (student details).
  - ✓ **Subjects** table.
  - ✓ **Grades** table (student_id, subject_id, marks, grade).
- Local state handled via **Express session/JWT authentication**.
- Data relationships implemented (One-to-Many: Student → Grades).

Example Table Design:

Table : Students

Student_ID	Name	Class	Roll _No	Email	Mobile_No
101	Rahul	10-A	12	rahul02@gmail.com	9577883210
102	Priya	10-B	25	priya56@gmail.com	9346231678

Table :Marks

Mark_ID	Student_ID	Subject	Marks	Grade
1	101	Maths	80	A+
2	102	Physics	96	O

4. Testing Core Features

- **Postman testing** done for all API endpoints (e.g., /students, /grades, /login).
- Verified CRUD operations for student and grade management.
- Checked authentication flow with **JWT token validation**.
- Unit testing with **Mocha/Jest** for core backend functions.

5. Version Control (GitHub)

- Used **GitHub repository** for code version control.
- Regular commits with proper messages (e.g., “Added Student API”, “Integrated MySQL schema”).
- Created separate **branches for features** (feature/authentication, feature/grades).
- Enabled GitHub **collaboration and pull requests** for team workflow.