 

**COLLEGE CODE : 9222**

**COLLEGE NAME : THENI KAMMAVAR SANGAM COLLEGE OF**

**TECHNOLOGY**

**DEPARTMENT : B.TECH(IT)**

**STUDENT NM-ID : 6D112234381ADB68ED2489F91797AE06**

**ROLL NO : 922223205040**

**DATE : 03/10/2025**

**Completed The Project Named As**

**PHASE 3**

**NAME : IBM-NJ-STUDENT GRADING SYSTEM**

**SUBMITTED BY,**

**NAME : RITTHIK K**

**MOBILE NO : 9751631967**

# MVP Implementation

**Title:Student Greading System**

**🔧 Project Setup**

**This step involves preparing the development environment and tools needed to build the project.**

* **Tools & Technologies Used:**
  + **Programming Language: e.g., Python / JavaScript / Java / PHP**
  + **Frontend: HTML, CSS, JavaScript (or frameworks like React, Angular)**
  + **Backend: Node.js / Django / Flask / PHP / Java Spring**
  + **Database: MySQL / MongoDB / Local Storage**
  + **Version Control: Git & GitHub**
* **Initial Setup:**
  + **Create a project folder and initialize it.**
  + **Setup package managers (like npm, pip, etc.).**
  + **Create a file structure: /frontend, /backend, /database, etc.**
  + **Install necessary dependencies/libraries.**

**🧩 Core Features Implementation**

**This part includes building the main features of the grading system.**

1. **Student Registration & Login (optional, if login is needed)**
2. **Add / Edit / Delete Student Records**
   * **Student ID, Name, Class, Subject(s), Marks**
3. **Grade Calculation**
   * **Automatically calculate grades based on marks.**
   * **Example:**
     + **90-100: A**
     + **80-89: B**
     + **etc.**
4. **View Report Card**
   * **Display individual student marks and grades.**
5. **Search & Filter**
   * **Search students by name, ID, grade, etc.**

**💾 Data Storage (Local State / Database)**

* **Local State (For small projects or frontend-only apps):**
  + **Data stored temporarily using JavaScript variables or state management (e.g., React useState, Redux).**
  + **May use localStorage or sessionStorage to save data locally.**
* **Database (For scalable apps):**
  + **Store student data in a proper database.**
  + **Use SQL (MySQL, PostgreSQL) or NoSQL (MongoDB) depending on the project.**
  + **Backend connects to the database using APIs to perform CRUD operations.**

**🧪 Testing Core Features**

* **Manual Testing:**
  + **Check each feature by entering data and verifying outputs.**
  + **Test edge cases (e.g., missing marks, invalid inputs).**
* **Automated Testing (Optional but preferred):**
  + **Use testing tools (like Jest for JavaScript, PyTest for Python).**
  + **Write test cases for:**
    - **Grade calculation logic**
    - **Data insertion/deletion**
    - **Input validation**

**🌐 Version Control (GitHub)**

* **Initialize Git Repository:**
  + **Run git init in your project folder.**
  + **Add .gitignore file to avoid committing unnecessary files.**
* **Basic Git Commands:**
  + **git add . — Add all changes**
  + **git commit -m "Initial commit" — Save changes with a message**
  + **git branch — Manage branches**
  + **git push origin main — Upload code to GitHub**
* **Using GitHub:**
  + **Create a new repository on GitHub.**
  + **Push local repo to GitHub using remote URL.**
  + **Use branches to manage features.**
  + **Collaborate with others through pull requests and commits.**