 

**COLLEGE CODE : 9222**

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

**TECHNOLOGY**

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

**STUDENT NM-ID : 6D112234381ADB68ED2489F91797AE06**

**ROLL NO : 922223205040**

**DATE : 08/10/2025**

**Completed The Project Named As**

**PHASE 4**

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

**SUBMITTED BY,**

**NAME : RITTHIK K**

**MOBILE NO : 9751631967**

# Enhancements & Deployment

**Title:Student Greading System**

**🔧 1. Additional Features**

* **Student Performance Reports: Automatically generate detailed reports for each student.**
* **Role-Based Access: Different interfaces and permissions for Admin, Teacher, and Student roles.**
* **Notifications: Alerts for new grades, updates, or feedback through email or in-app messages.**
* **Export Options: Download grade sheets in PDF, Excel formats.**
* **Search & Filter: Easily search and filter students by name, grade, or subject.**

**🎨 2. UI/UX Improvements**

* **Responsive Design: The interface adjusts to all screen sizes (mobile, tablet, desktop).**
* **Clean Dashboard Layout: Organized dashboard for easy navigation and information access.**
* **Theme Options: Support for Light and Dark modes.**
* **Accessibility: Improved support for screen readers and keyboard navigation.**
* **User-Centered Design: Easy-to-understand icons, labels, and actions for a smooth user experience.**

**🔌 3. API Enhancements**

* **RESTful API Structure: Well-defined routes for handling grading, student info, and authentication.**
* **Secure Authentication: JWT-based login system for safe API access.**
* **Input Validation: Strong checks to ensure only valid data is entered.**
* **Search, Sort, and Pagination: Enhanced API capabilities for large datasets.**
* **API Documentation: Swagger/Postman docs provided for developers to test and integrate easily.**

**🔐 4. Performance & Security Checks**

* **Data Protection: All sensitive data encrypted and stored securely.**
* **Input Sanitization: Prevents SQL Injection and Cross-Site Scripting (XSS).**
* **Optimized Queries: Improved database performance using indexing and efficient queries.**
* **HTTPS Protocol: Ensures all data transmission is encrypted.**
* **Session Management: Auto logout after inactivity to protect accounts.**

**🧪 5. Testing of Enhancements**

* **Unit Testing: All core functions tested individually.**
* **Integration Testing: Ensures all modules work together properly.**
* **UI Testing: Front-end tested using tools like Selenium or Cypress.**
* **API Testing: Postman or automated scripts used to test all API endpoints.**
* **User Testing: Feedback collected from sample users to identify issues before launch.**

**☁️ 6. Deployment (Netlify, Vercel, or Cloud Platform)**

* **Frontend Deployment:**
  + **Hosted on Netlify or Vercel for fast and global content delivery.**
  + **Automatic deployment from GitHub on every update.**
* **Backend Deployment:**
  + **Hosted on Render, Railway, or Heroku with environment variables and scaling options.**
* **Database:**
  + **MongoDB Atlas, Firebase, or Supabase used for secure cloud database storage.**
* **CI/CD Integration:**
  + **Continuous Integration and Deployment setup to ensure the latest changes are always live.**
* **Monitoring Tools:**
  + **Tools like Sentry, LogRocket, or Google Analytics used to track performance and fix issues quickly.**