

# Functional Requirements

## 1 .Student Management

- **Registration**
  - Enable students to register by entering details.
  - Upload photo or scanned ID.
  - Validate and store data on submission.
- **Management**
  - Display a searchable and filterable list of students.
  - Update student details .
  - View payment history and academic progress.
- **Tracking**
  - Allow administrators to update and monitor academic status.
  - Provide tools to contact students.
  - Collect and visualize student data over time.

## 2 .Course Management

- **Create New Course**
  - Enter course name, description, and prerequisites.
  - Assign a lecturer and set schedule.
  - Upload course materials.
- **Manage Existing Courses**
  - Display a list of all courses.
  - Edit, delete, or view course details.
  - View registered students for each course.

## 3 .Lecturer Management

- **Adding a Lecturer**
  - Enter lecturer details .
  - Assign lecturers to courses.
- **Managing Lecturers**
  - View a list of all lecturers.
  - Update lecturer details.
- **Course Allocation**

- Assign lecturers to specific courses.

#### **4. Payments and Invoices**

- **Invoice Creation**
  - Enter payment details, including amount and type.
  - Apply discounts or special offers.
- **Sending Invoices**
  - Send invoices to students via email.
- **Payment Tracking**
  - Monitor payment statuses .
- **Invoice History**
  - View and manage previous invoices.
  - Resend invoices when needed.

#### **5 .Evaluations and Grades**

- **Grade Entry**
  - Input grades and comments for students.
- **Student Ranking**
  - Rank students based on attendance, group work, and academic performance.
- **Feedback**
  - Provide ongoing feedback to students.
- **Reports**
  - Generate grade lists and detailed reports.
  - Allow students to download personalized grade reports.

#### **6. Reporting**

- **Generate Reports**
  - Choose report types .
  - Set report parameters .
  - Visualize data.

#### **7. User Management and Permissions**

- **User Management**
  - Display a list of all users in the system.
  - Update user details or change passwords.

- **Permissions**
  - Allocate access permissions based on roles (administrator, lecturer, student).
  - Manage access to different modules.

## **8. Support and Feedback**

- **Technical Support**
  - Submit technical inquiries to a support system.
- **Feedback Mechanism**
  - Allow users to provide feedback for system improvement.

# **Non-Functional Requirements**

## **1. Usability**

- The system must be user-friendly with a clean, intuitive interface.
- Provide easy navigation between modules.

## **2. Security**

- Implement role-based access control to protect sensitive data.
- Secure integration with external systems (e.g., payment gateways, email servers).
- Encrypt stored and transmitted data.

## **3 .Performance**

- Ensure system responsiveness under high user loads.
- Generate reports and queries within 2 seconds.

## **4. Scalability**

- Design a modular system that supports future feature additions.
- Handle increased user and data loads as the college grows.

## **5. Integration**

- Support seamless integration with external systems .
- Ensure compatibility with popular operating systems and browsers.

## **6.Availability**

- Achieve 99.9% system uptime.
- Provide a backup mechanism for data recovery.

## **7 .Accessibility**

- Ensure system compliance with accessibility standards .
- Provide multi-language support for diverse user bases.

## **8 .Maintainability**

- Use clear and well-documented code.
- Ensure ease of updates for bug fixes or new features.