Functional Requirements (FR)

1. User Registration and Login

- Users (**students** and **instructors**) can register using acadymic email and password.
- Users can log in and log out.
- **Password reset** functionality for forgotten passwords.

2. Course Browsing

- Students can view a list of all **available courses**.
- Courses are categorized by department (e.g., Engineering, Computer Science) or academic level.

3. Course Details

- Each course has a dedicated page showing its description, instructor, and credit hours.

4. Registration Cart

- Students can add desired courses to a temporary **registration cart**.
- Students can remove courses from the cart before finalizing their registration.

5. Checkout Process

- Students review their selected courses.

6. Enrollment Management

- Students can view their current enrollment status.
- Admins can update the enrollment status of students in courses.

8. Admin Dashboard

- Admins can **add, update, or delete** courses.
- Admins can manage student accounts, instructor accounts, and enrollments.

9. User Profile

- Users can update their personal information and change their password.

Non-Functional Requirements (NFR)

1. Performance

- The website should load in **less than 5 seconds** on average internet connections.

2. Scalability

- The system must handle increased traffic during peak registration periods without performance degradation.

3. Availability

- The system must be available **99.99%** of the time (high uptime).

4. Security

- User password must be **encrypted** and stored securely.
- Protect against common web vulnerabilities like **SQL Injection & CSRF**

5. Usability

- The platform must be **intuitive** and easy to navigate for all users.

6. Maintainability

- The code should follow **clean architecture** principles to allow for easy maintenance and updates.

7. Compatibility

- The website must be fully **responsive** and functional across all major browsers and devices (PC, tablet, mobile).

I. Risks

1. **Performance Crash at Peak Time:**
   - **Description:** This is the classic problem. When registration opens (e.g., at 9:00 AM), thousands of students will try to log in at the same second. If the server and database are not designed to handle this high concurrency, the system will crash.
2. **Security Breach:**
   - **Description:** Student data and grades are highly sensitive. Any security vulnerability (like SQL Injection or Broken Access Control) could allow a

student to view another student's data, modify their grades, enroll in a course without meeting its prerequisites.

3. **Data Integrity Errors:**
   - **Example (Prerequisites):** The system incorrectly blocks a student from registering, claiming they haven't passed a prerequisite when they actually have.

4. **Poor Usability (UX)**