REQUIREMENTS

1. Functional Requirements

User Registration and Authentication

- Users (Students and Admins) must register using a unique email, name, and secure password.
- The Admin account is created only once at the beginning by the system developer, as the system is role-based. Admin credentials are predefined and do not require registration through the user interface.
- All login credentials must be securely encrypted and stored.

Profile Management

- Students can view and update their personal information, including name, email, password, and profile photo.
- Admins can access and update student profiles for verification or correction.
- All profile changes are reflected immediately across the system.

Role-Based Access Control

- The platform supports multiple user roles with specific access rights:
    - *Student*: Access to enrolled courses, quizzes, progress tracking, and AI suggestions.
    - *Admin*: Full control over content management, student accounts, system configuration, and reporting.
- Role-based dashboards ensure each user sees tools and data relevant to their role.

Course and Topic Management

- Admins can create and manage courses with the following hierarchy:
    - Course Title
    - Description
    - Category (e.g., Math, Science)
    - Subcategory or Level (e.g., Algebra, Biology)
    - Topics (e.g., Linear Equations)

    - Topic Content (videos, PDFs, exercises)
    - Linked Quizzes and Assignments
- Topics can be reordered and updated independently.
- Admins can tag topics for AI-driven relevance and recommendations.
- Students can navigate courses by topic and access content in a logical, guided sequence.
- Progress tracking is available per topic.
- Courses, topics, and content blocks can be deactivated without being permanently deleted.

AI-Driven Personalized Learning

- An integrated AI engine monitors user activity and performance by analyzing:
    - Time spent on each topic
    - Assessment performance (quizzes and assignments)
    - User interaction patterns (e.g., skipped videos, repeated views)
- Based on this analysis, the AI engine:
    - Generates a customized learning path upon course enrollment
    - Adjusts topic sequence and difficulty level in real time
    - Recommends advanced, remedial, or supplementary resources
    - Suggests reattempts or easier explanations for weak areas
- Personalized progress and recommendations are dynamically displayed on the student dashboard.

Quizzes and Assignments

- Admins can:
    - Create quizzes and assignments linked to specific topics
    - Define time limits, difficulty levels, and grading criteria
- Students can:
    - Attempt assessments within set deadlines
    - View instant results, detailed performance, and feedback

- The system supports AI-based adaptive testing, where difficulty adjusts based on past performance.
- Students may be prompted to retry weak areas or attempt similar practice material.

Progress Tracking and Feedback

- Students can view:
    - Real-time course progress through visual indicators
    - Score breakdowns and completion percentages
    - AI-generated suggestions for improvement
- Admins can access:
    - Performance analytics by student, course, and topic
    - Reports highlighting weak areas and improvement trends
    - Statistics on course completion and dropout rates
- The system can flag underperforming students and generate actionable alerts.

2. Non-Functional Requirements

Performance

- The system must support concurrent access by 15 users without performance degradation.
- All pages must load within 20 seconds under normal load conditions.

Security

- All user credentials must be stored using secure encryption algorithms.
- Role-based access must prevent unauthorized access to restricted features.
- Data transmission must be encrypted using SSL/TLS.
- Admin credentials must be predefined and securely managed.

Usability

- The interface must be intuitive and user-friendly for both students and admins.
- Navigation should require no more than 3 clicks to access any primary feature.
- The system should provide meaningful feedback messages for user actions.

Maintainability

- The system must be modular and follow best coding practices to allow efficient updates and scalability.
- Documentation should be provided for both the frontend and backend to support future maintenance.