# Requirements Document for E-Learning Platform

|  |  |
| --- | --- |
| Chapter | Description |
| Preface | This document defines the scope, objectives, and functionality of the proposed e-learning system, including version history and a rationale for development. |
| Introduction | The e-learning platform aims to provide accessible, scalable, and engaging online education tools for students, instructors, and administrators. |
| Glossary | Defines key terms like User Authentication, Role-Based Access, Progress Tracking, and Wireframes. |
| User Requirements Definition | Describes the services for students (course enrollment, tracking), instructors (content management), and admins (platform oversight). Nonfunctional requirements include performance, security, and usability. |
| System Architecture | Highlights the three-tier architecture (frontend, backend, database) and reusable modules like authentication and course management. |
| System Requirements Specification | Includes functional requirements such as user authentication, course management, content delivery, and progress tracking. Nonfunctional requirements cover performance benchmarks and usability standards. |
| System Models | Contains class diagrams, use case diagrams, and sequence diagrams for enrollment, content access, course management, and feedback. |
| System Evolution | Describes assumptions about future changes, including scalability, integration of real-time features, and enhancements based on user feedback. |
| Appendices | Details include hardware requirements and database descriptions, such as the use of PostgreSQL for storing user and course data. |
| Index | Provides indexes for diagrams, features, and modules described in the document. |

**Preface**

This document defines the scope, objectives, and functionality of the proposed e-learning management system. It includes version history, a rationale for the system, and highlights the challenges addressed by the platform.

**Version History:**

* **Version 1.0:** Initial proposal outlining requirements and architecture.

**Rationale for Development:** The platform is designed to address key challenges in online education, such as accessibility, scalability, and engagement. It leverages modern web technologies to ensure a seamless user experience.

**Introduction**

The e-learning management system aims to revolutionize online education by offering a user-friendly platform for students, instructors, and administrators. The system will provide tools for course creation, content delivery, progress tracking, and user management, aligning with the strategic objective of democratizing quality education.

**Glossary**

* **User Authentication:** Process of verifying the identity of a user.
* **Role-Based Access Control:** System that restricts access based on a user’s role.
* **Progress Tracking:** Monitoring and reporting on students’ progress in courses.
* **Wireframes:** Visual blueprints of the user interface.
* **SQL Database:** Structured database for storing and managing platform data.

**User Requirements Definition**

**Services Provided:**

1. **Students:**
   * Enroll in courses.
   * Access course materials and track progress.
   * Receive feedback on assignments and tests.
2. **Instructors:**
   * Create, manage, and deliver courses.
   * Monitor student progress and engagement.
   * Provide feedback through assessments.
3. **Administrators:**
   * Oversee platform operations.
   * Manage user roles and permissions.
   * Ensure system reliability and data security.

**Nonfunctional Requirements:**

* **Performance:** Support up to 100 concurrent users with minimal latency.
* **Security:** Secure user data using Subspace for authentication and HTTPS..
* **Usability:** Responsive design optimized for web devices.

**System Architecture  
The e-learning platform will follow a three-tier architecture:**

1. **Frontend (Presentation Layer):**
   * Technologies: C JavaScript, React.
   * Components: Navigation bar, dashboards, and course pages.
2. **Backend (Application Layer):**
   * Technology: Subapase (for backend functionality).
   * Functionalities: Authentication, course management, progress tracking.
3. **Database (Data Layer):**
   * Technology: PostgreSQL (using Subapase).
   * Stores: User data, courses, and progress reports**.**

**Highlighted Architectural Components:**

* **Reusable Modules:**
  + Authentication system.
  + Content delivery components.

**System Requirements Specification**

**High-Level Functional Requirements**

1. **User Authentication and Role Management**:
   * Users can register as students, instructors, or administrators.
   * Secure login and session management.
   * Role-based access control (students, instructors, admins).
2. **Course Management**:
   * Instructors can create, edit, and delete courses.
   * Students can enroll in courses based on academic level and view materials.
   * Admins can manage and approve courses.
3. **Content Delivery**:
   * Instructors can upload materials such as PDFs, videos, and quizzes.
   * Students can download materials and submit assignments.
4. **Progress Tracking**:
   * Students can view their course progress and grades.
   * Instructors can track student performance through analytics.(curve of grades of students is high or low )
5. **Communication Tools**:
   * Message boards for students and instructors.(optional not important)
   * Notification system for course updates and deadlines.

**5. High-Level Non-Functional Requirements**

1. **Performance**:
   * The platform should handle up to 100 concurrent users with a response time of less than 2 seconds for most actions.
2. **Security**:
   * Use Django’s built-in authentication and data encryption for sensitive information.
   * Enforce HTTPS for secure communication.
3. **Usability**:
   * A responsive design that works seamlessly on both desktop and mobile devices.
   * An intuitive user interface for all user roles.
4. **Maintainability**:
   * Follow Django coding standards and modular design for easy updates and bug fixes.

**System Models**

The platform includes the following diagrams:

1. **Class Diagram:** Depicts relationships between users, courses, and other entities.
2. **Use Case Diagram:** Illustrates system interactions for each user role.
3. **Sequence Diagrams:** Scenarios include enrollment, accessing course materials, and managing courses.

**System Evolution**

The system is designed with future scalability in mind, allowing integration of new features such as real-time chat and AI-based progress analytics. Anticipated updates will focus on improving performance and incorporating user feedback.

**Appendices**

* **Hardware Requirements:** Any modern browser and a stable internet connection.
* **Database Details:** PostgreSQL for storing user, course, and progress data.

**Index**

* **Diagrams:** Class diagram, use case diagram, sequence diagrams.
* **Features:** Authentication, course management, progress tracking.