DON BOSCO INSTITUTE OF TECHNOLOGY
Kumbalagodu, Mysore Road, Bengaluru – 560074

Department of Information Science and Engineering

Software Engineering & Project Management (BCS501)

On

"System Requirements Specification on

Edu-Learn Platform"

Submitted By

Spandana– 1DB22IS117

Under the Guidance of:

Mrs. Deepika A.B
Assistant Professor Dept. of ISE

Department of Information Science and Engineering
DON BOSCO INSTITUTE OF TECHNOLOGY
Kumbalagodu, Mysuru Road, Bengaluru-560074
2024-25

Table of Contents

1.Introduction

1) Purpose

The purpose of this SRS is to define the functional and nonfunctional requirements for an E-Learning platform that enables students and educators to interact, share resources, and track progress. The system will be developed using the MERN stack.

2) Product Scope

The E-Learning platform is a dynamic, web-based system designed to enhance modern education by offering a versatile and interactive environment for students, educators, and institutions. This platform enables *educators* to create and manage courses, upload multimedia content such as videos, PDFs, and quizzes, and monitor student progress through advanced analytics. *Students* can access course materials, participate in discussions, and track their learning journey with features like performance dashboards and real-time notifications. Key highlights of the platform include:

-*Live Classes* powered by video conferencing tools.

-*Gamification Elements*, such as badges and certificates, to boost engagement.

-*Cross-Device Accessibility* for seamless learning on desktops, tablets, and smartphones.

-*AI-Driven Recommendations* to personalize the learning experience.

Built using the *MERN stack*, the platform ensures scalability, robust performance, and secure user authentication. Designed for diverse use cases, it supports interactive, collaborative, and self-paced learning, catering to learners and professionals alike. Institutions can also digitize their curriculum efficiently using administrative tools and comprehensive reporting capabilities.

3) Definitions, Acronyms, and Abbreviations

1. **MERN**: MongoDB,Express,React,NodeJs
2. **UI**: User Interface
3. **CRUD**: Creat,Repeat,Update,Delete
4. **DB**: MongoDB Database

4) References

•IEEE 830-1998: Software Requirements Specification Standard

•ISO/IEC 27001: Information Security Management.

2. Overall Description

1) Product Functions

The system will include the following primary functions:

1. User authentication and role-based access control (students, educators, administrators).
2. Course creation and management.
3. Multimedia content delivery.
4. Real-time chat and discussion forums.
5. Progress tracking and reporting.

2) Operating Environment

-***Client Side***: React for responsive, dynamic web interfaces.

-***Server Side***: Node.js with Express for RESTful APIs.

-***Database***: MongoDB for data storage.

-***Deployment***: Cloud-based (e.g., AWS, Heroku).

3) Design and Implementation Constraints

-Limited bandwidth for users in remote areas.

-Adherence to accessibility standards (WCAG).

-Scalability to support thousands of concurrent users.

4) Assumptions and Dependencies

•- Internet connectivity is available for all users.

•- Third-party libraries will remain stable and supported.

3. External Interface Requirements

1) User Interfaces

1.Login Page: A secure login interface with username, password fields, SSO integration, and a "Forgot Password" option for easy recovery.

2.Dashboard: A role-specific dashboard showcasing course progress for students and performance analytics for educators, with a navigation bar and quick search.

3.Course Management Page: A streamlined interface for educators to create, update, or delete courses and upload multimedia content easily.

4.Reports Page: A comprehensive page displaying student progress and course performance with options to generate, visualize, and export reports.

5.Settings Page: A configuration page for managing account settings, language preferences, and platform policies for personalized experiences.

2) Hardware Interfaces

1. Server: Quad-core processor, 16 GB RAM, 500 GB SSD, Gigabit Ethernet.
2. Workstations: Dual-core processor, 4 GB RAM, 1280x720 resolution, keyboard/mouse.
3. Network: Gigabit routers/switches, Cat6 cables, dual-band Wi-Fi.
4. Backup Devices: External 1 TB HDD/SSD or cloud storage for data backups.
5. Authentication Devices: RFID/card readers and biometric scanners.

3) Software Interfaces

1. **Operating System**: Linux-based server OS for hosting.
2. **Database**: MySQL or PostgreSQL for data storage.
3. **Web Server**: Apache or Nginx for serving the application.
4. **APIs**: RESTful APIs for communication and integration.
5. **Authentication**: SSO or LDAP for secure user access.

4) Communication Interfaces

1. Network Protocols: HTTP/HTTPS for web communication and secure data transfer.
2. API Communication: RESTful APIs for interaction between system components and external systems.
3. Email Notifications: SMTP integration for automated alerts and notifications.

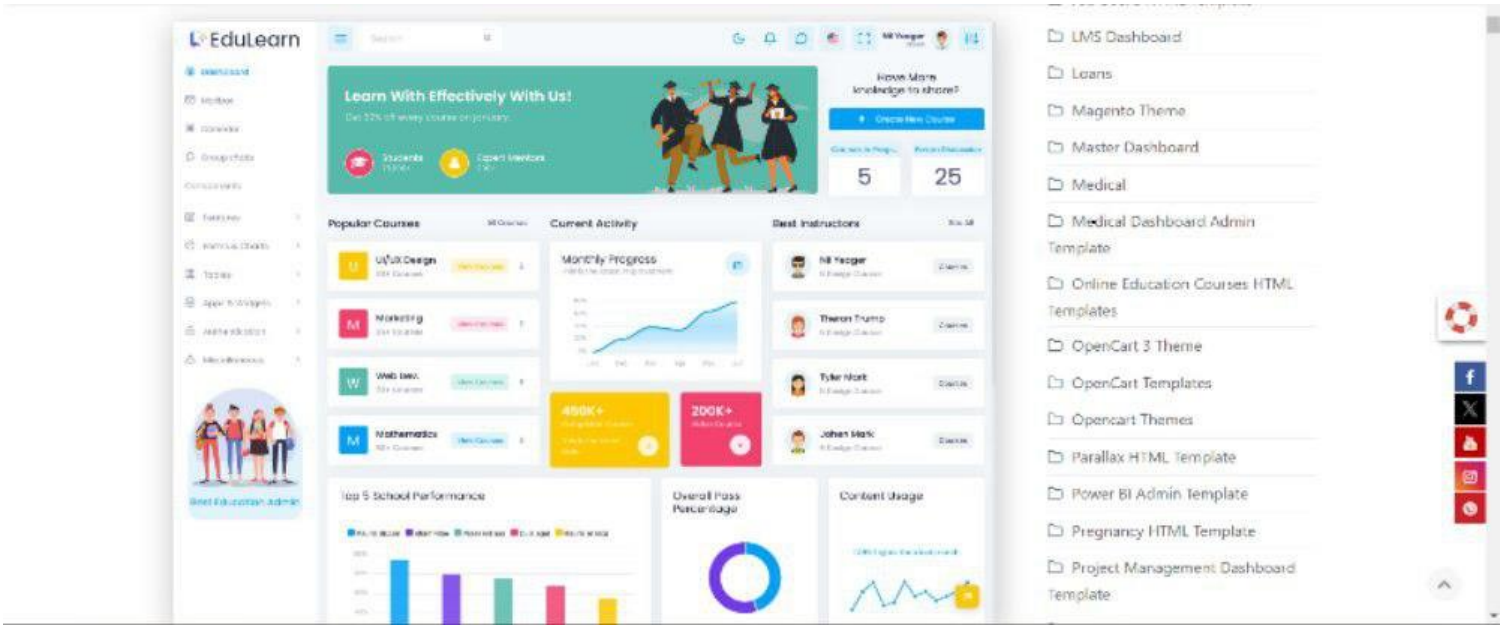

Figure : Home Page

4. Functional Requirements

-User Management: Register, login, and manage user roles.

-Course Management: Add, update, and delete courses. Secure registration and profile management facilities for different users.

-Content Delivery: Upload multimedia materials and quizzes.

-Analytics: Track student performance and generate reports.

5. Nonfunctional Requirements

5.1 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.2 Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.3 Software Quality Attributes

- AVAILABILITY: Since we are hosting our project on the server it will be available all the time.
- CORRECTNESS: The system should generate an appropriate report about different activities of the lab and should keep track of all records.
- MAINTAINABILITY: The system should maintain correct schedules of labs and the documentation of all the lab equipment.
- USABILITY: The system should satisfy the maximum number of users needs.

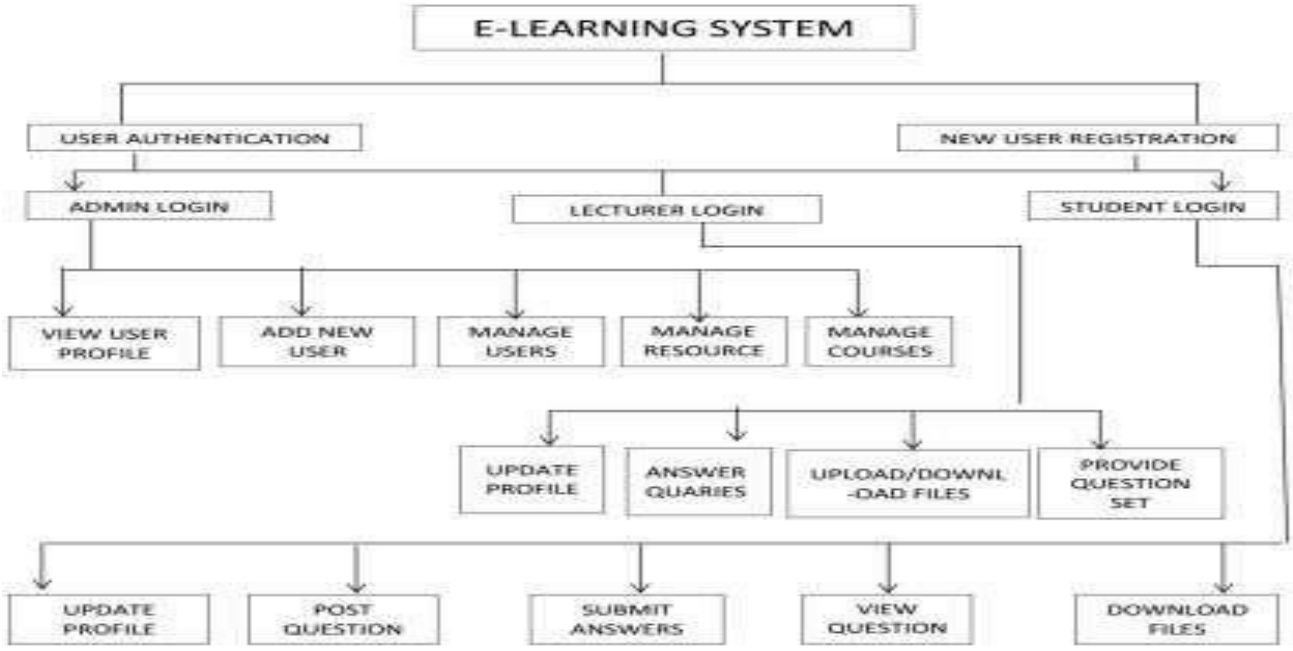

Figure: System Architecture of Edu-Learn platform

6. Other Requirements

1. **Interoperability**: The system should integrate seamlessly with existing systems like student information systems (SIS) and other administrative tools.
2. **User Training and Support**: Provide documentation, training, and helpdesk support for users to effectively use the system.
3. **Customizability**: The system should offer customization options for user preferences, roles, and labspecific settings.
4. **Error Handling**: Implement robust error handling to manage system failures and unexpected situations gracefully.
5. **Logging and Auditing**: Track and log user activities and system actions for security and compliance purposes.
6. **Accessibility**: The system should be accessible to users with disabilities, adhering to WCAG guidelines.

Appendix A: Glossary

1. **SSO (Single Sign-On):** Enables users to log in once and access multiple applications without signing in again.
2. **LDAP (Lightweight Directory Access Protocol):** Allows secure access and management of user directory information over a network.
3. **RESTful API:** A web API that follows the principles of REST, using HTTP requests to perform operations like GET, POST, PUT, and DELETE.
4. **CRUD (Create, Read, Update, Delete)**: The basic operations of a database - creating a new record, reading a record, updating a record, and deleting a record.

5. **Backend:** The server-side part of an application responsible for managing data storage, processing logic, and communication with the database.
6. **Frontend:** The client-side part of an application that handles user interface and experience, interacting with

   the backend.
7. **DBMS (Database Management System):** Software that manages, stores, and retrieves data in a database, such as MySQL or PostgreSQL.
8. **API Key:** A unique identifier used to authenticate requests made to a web API, ensuring that only authorized users can access the API.
9. **SDK (Software Development Kit):** A collection of software development tools used to create applications for a specific platform.
10. **UML (Unified Modeling Language):** A standardized odeling language used for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.
11. **SSO (Single Sign-On):** A process that enables a user to authenticate once and gain access to multiple applications without needing to sign in again.

Appendix B: Analysis Models

1. **Use Case Model:** Defines actors (students, instructors, admins) and their interactions like course enrollment, content management, and progress tracking
2. **Data Model:** Entities include users, courses, content, assessments, and progress reports with relationships connecting the data flow.
3. **Activity Diagrams:** Illustrates processes like course enrollment, content upload, and quiz completion with clear step-by-step workflows.
4. **State Diagrams:** Defines the lifecycle states of entities like courses (draft, published, archived) and user sessions (logged in, logged out).
5. **Sequence Diagrams:** Shows interactions between users, the frontend, and the backend for scenarios like enrolling in a course or submitting assessments.
6. **Logical Architecture:** Three-tier system with client (frontend), application logic (backend), and database (MongoDB) tiers ensuring secure communication.

   Inter-component Communication: Defines how different components interact within the system.
8. **Non-Functional Requirements:** Address scalability, performance, reliability, security, usability, and maintainability.
9. **Error Handling:** Robust error management processes for unexpected situations.
10. **Customizability:** Allows configuration of user roles, course settings, grading policies, and platform themes
11. **Integration:** Seamless integration with systems like Learning Management Systems (LMS), payment gateways, and external databases.
12. **Backup and Recovery:** Automatic daily backups and recovery processes to safeguard data.

13. **Compliance:** Ensures compliance with relevant regulations such as GDPR for data protection and privacy.

Appendix C: To Be Determined List

1. **User Interface Design:** Finalizes layout, color scheme, and user experience to ensure intuitive navigation for students, instructors, and admins.

2. **Detailed Data Model Attributes:** Defines attributes for entities like users (name, email, role), courses (title, duration), content (type, format), and assessments (scores, deadlines).

3. **Notification Mechanisms:** Includes system-generated alerts like email, SMS, and in-app notifications for deadlines, announcements, and activity updates.

4. **Custom Reports:** Design customizable reports for different roles, such as student progress, instructor performance, and course engagement metrics.

5. **System Performance Benchmarks:** Defines metrics like page load times (<2s), concurrent users (10,000+), and acceptable server response times.

6. **Error Handling Scenarios:** Scenarios for failed logins, incomplete submissions, API timeouts, and unexpected system errors with clear user feedback.

   Accessibility Features: Implements WCAG-compliant features like screen reader support, keyboard navigation, and adjustable font sizes for accessibility.

8. **Backup and Recovery Procedures:** Details automatic daily backups and recovery steps for user data, course content, and progress history.

9. **Security Measures:** Specifies data encryption (e.g., AES-256), secure user authentication (JWT), access controls, and role-based permissions.

10. **Scalability Testing:** Defines load testing benchmarks to handle increasing users, content uploads, and concurrent course access efficiently.

11. **Data Migration Plan:** Outlines the process to migrate existing courses, user data, and records into the new elearning platform seamlessly.

12. **Training Materials:** Provides user manuals, video tutorials, and onboarding guides for students, instructors, and admins.

13. **Compliance Requirements**: Ensures GDPR compliance for data privacy, user consent, and legal protection of sensitive information.

14. **Integration with SIS:** Details integration with Student Information Systems for seamless enrollment, grades, and user data synchronization.