

Low Level Design

eLearning – Skill Development And Learning Website

Written By Ankur Das, Rahul Pidkalwar

Document Version 0.3

Last Revised Date 23 – May -2021

Document Control

Change Record:

Version	Date	Author	Comments
0.1	19 – May - 2021	Ankur Das	Introduction & Architecture defined
0.2	20 – May - 2021	Rahul Pidkalwar	Architecture & Architecture Description appended and updated
0.3	21 – May - 2021	Ankit Singh Chauhan	Unit Test Cases defined and appended

Reviews:

Version	Date	Reviewer	Comments
0.2	21 – May - 2021	Khusali	Document Content , Version Control and Unit Test Cases to be added

Approval Status:

Version	Review Date	Reviewed By	Approved By	Comments
---------	-------------	-------------	-------------	----------

Contents

1. [Introduction](#) 1
 1. [What is Low-Level design document?](#) 1
 2. [Scope](#) 1
2. [Architecture](#) 2
3. [Architecture Description](#) 3
 1. [Database Design](#) 3
 2. [User Authentication System](#) 3
 3. [Course Management Module](#) 3
 4. [Content Upload & Storage](#) 3
 5. [Student Management System](#) 3

6. [Examination Module](#) 4
 7. [Result Processing & Analytics](#) 4
 8. [File Management System](#) 4
 9. [User Interface Layer](#) 4
 10. [API Gateway & Services](#) 4
 11. [Session Management](#) 4
 12. [Progress Tracking System](#) 4
 13. [Notification System](#) 4
 14. [Security & Access Control](#) 4
 15. [Deployment Architecture](#) 5
4. [Unit Test Cases](#) 5

Introduction

What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for eLearning – Skill Development And Learning Website. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document. This document focuses on the detailed implementation of the eLearning platform using MongoDB, Express.js, Node.js, HTML, CSS, and JavaScript technologies.

Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step [refinement](#) process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms for the eLearning platform. Overall, the data organization may be defined during requirement analysis and then refined during data design work. The scope includes detailed implementation of both Admin and User modules with their respective functionalities.

Architecture Description

Database Design

The eLearning platform uses MongoDB as the primary database. The database schema includes collections for Users, Courses, Content, Examinations, Results, and System Logs. MongoDB's document-oriented structure allows for

flexible storage of course materials, user profiles, and educational content in various formats including text, multimedia, and structured data.

Key Collections:

- **Users Collection:** Stores admin and student profiles, authentication data
- **Courses Collection:** Course metadata, structure, and enrollment information
- **Content Collection:** Educational materials, documents, videos, presentations
- **Examinations Collection:** Question banks, exam configurations, and settings
- **Results Collection:** Student performance data, scores, and progress tracking
- **Sessions Collection:** User session management and activity tracking

User Authentication System

A secure authentication system implemented using JWT (JSON Web Tokens) and bcrypt for password hashing. The system supports role-based access control distinguishing between Admin and User roles with different permission levels and access rights.

Components:

- Login/Registration controllers
- Password encryption and validation
- JWT token generation and verification
- Session management and timeout handling
- Role-based middleware for route protection

Course Management Module

Comprehensive course management system allowing administrators to create, update, and manage educational courses. The module supports hierarchical course structure with chapters, lessons, and individual content items.

Key Features:

- Course creation and metadata management
- Chapter and lesson organization
- Course enrollment and capacity management
- Course prerequisites and sequencing
- Course analytics and reporting

Content Upload & Storage

Multi-format content upload system supporting documents (PDF, DOC, PPT), videos, and multimedia files. The system includes file validation, compression, and organized storage with efficient retrieval mechanisms.

Implementation Details:

- Multer middleware for file upload handling
- File type validation and size restrictions
- Cloud storage integration for scalability
- Content versioning and backup systems
- Streaming capabilities for large video files

Student Management System

Complete student lifecycle management including registration, profile management, course enrollment, and progress tracking. The system maintains detailed student records and learning analytics.

Core Functions:

- Student registration and profile creation
- Course enrollment and waitlist management
- Learning path customization
- Progress monitoring and reporting
- Communication and notification preferences

Examination Module

Comprehensive examination system supporting multiple question types, timed assessments, and automated grading. The module includes question bank management and exam scheduling capabilities.

Features:

- Question bank creation and categorization
- Multiple choice, true/false, and essay questions
- Exam scheduling and time management
- Auto-grading and manual review options
- Proctoring and anti-cheating measures

Result Processing & Analytics

Advanced result processing system that calculates scores, generates reports, and provides detailed analytics on student performance and learning outcomes.

Analytics Components:

- Real-time score calculation
- Performance trend analysis
- Comparative analytics and benchmarking

- Learning outcome assessment
- Predictive analytics for learning paths

File Management System

Robust file management system handling storage, organization, and delivery of educational content across multiple formats and devices.

System Features:

- Hierarchical file organization
- Content delivery optimization
- Backup and recovery mechanisms
- File sharing and permission management
- Mobile-optimized content delivery

User Interface Layer

Responsive web interface built with HTML5, CSS3, and JavaScript providing intuitive user experience across desktop and mobile devices.

UI Components:

- Responsive design framework
- Interactive dashboards for admin and users
- Real-time updates and notifications
- Accessibility compliance (WCAG guidelines)
- Cross-browser compatibility

API Gateway & Services

RESTful API architecture built with Express.js providing secure and scalable endpoints for all system functionalities.

API Structure:

- Authentication and authorization endpoints
- Course and content management APIs
- Examination and result processing services
- User management and profile services
- Analytics and reporting endpoints

Session Management

Secure session management system handling user authentication, session persistence, and timeout management across the platform.

Session Features:

- Secure session creation and validation
- Activity-based session extension
- Multi-device session management
- Session analytics and monitoring
- Graceful session termination

Progress Tracking System

Comprehensive progress tracking system monitoring student learning journey, course completion, and skill development metrics.

Tracking Elements:

- Course completion percentage
- Time spent on materials
- Assessment performance trends
- Learning milestone achievements
- Skill development progression

Notification System

Automated notification system keeping users informed about course updates, examination schedules, results, and system announcements.

Notification Types:

- Real-time in-app notifications
- Email notifications for important updates
- Course reminder and deadline alerts
- Achievement and milestone notifications
- System maintenance and update notices

Security & Access Control

Multi-layered security framework ensuring data protection, secure access control, and compliance with educational data privacy standards.

Security Measures:

- Role-based access control (RBAC)
- Data encryption in transit and at rest

- Input validation and sanitization
- SQL injection and XSS prevention
- Regular security audits and updates

Deployment Architecture

Cloud-based deployment architecture ensuring scalability, reliability, and optimal performance for the eLearning platform.

Deployment Components:

- Load balancing for high availability
- Auto-scaling based on user demand
- Content delivery network (CDN) integration
- Database replication and backup
- Monitoring and logging systems

Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the eLearning Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the eLearning Application loads completely for the user when the URL is accessed	1. Application URL is accessible2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the eLearning application	1. Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the eLearning application	1. Application is accessible2. User is signed up to the application	User should be able to successfully login to the application
Verify whether admin can access the admin dashboard after login	1. Application is accessible2. Admin credentials are valid3. Admin is logged in to the application	Admin should be able to access admin dashboard after login
Verify whether admin can add new courses to the system	1. Application is accessible2. Admin is logged in3. Course creation form is accessible	Admin should be able to add new courses successfully
Verify whether admin can upload course materials (doc, pdf, ppt, video)	1. Application is accessible2. Admin is logged in3. Course exists in the system	Admin should be able to upload course materials in supported formats

Test Case Description	Pre-Requisite	Expected Result
Verify whether admin can manage existing courses (edit/delete)	1. Application is accessible2. Admin is logged in3. Courses exist in the system	Admin should be able to manage existing courses successfully
Verify whether admin can add and manage student details	1. Application is accessible2. Admin is logged in to the application	Admin should be able to add and manage student details
Verify whether admin can create and manage exam questions	1. Application is accessible2. Admin is logged in3. Course exists	Admin should be able to create and manage exam questions
Verify whether students can view available subjects/courses after login	1. Application is accessible2. Student is registered and logged in3. Courses are available in the system	Students should be able to view available subjects/courses
Verify whether students can download course materials	1. Application is accessible2. Student is logged in3. Student is enrolled in the course4. Materials are uploaded	Students should be able to download course materials successfully
Verify whether students can attempt examinations	1. Application is accessible2. Student is logged in3. Exam is scheduled and available	Students should be able to attempt examinations
Verify whether students can view their exam results	1. Application is accessible2. Student is logged in3. Student has attempted the exam	Students should be able to view their exam results
Verify whether the system displays accurate course progress for students	1. Application is accessible2. Student is logged in3. Student has accessed course materials	System should display accurate course progress for students
Verify whether users can successfully logout from the application	1. Application is accessible2. User is logged in to the application	Users should be able to logout successfully and session should be terminated
Verify whether the system handles file upload size restrictions properly	1. Application is accessible2. Admin is logged in3. File exceeds size limit	System should display appropriate error message for oversized files
Verify whether the system validates supported file formats during upload	1. Application is accessible2. Admin is logged in3. Unsupported file format is uploaded	System should reject unsupported file formats with appropriate error message
Verify whether the examination timer works correctly	1. Application is accessible2. Student is logged in3. Timed exam is available	Examination timer should work correctly and auto-submit when time expires

Test Case Description	Pre-Requisite	Expected Result
Verify whether the system maintains user session security	1. Application is accessible 2. User is logged in 3. Session timeout is configured	System should maintain secure sessions and handle timeouts appropriately