

Software Design Document for EduLearn Pro

- High-Level Design Principles
- Scalability:
 - Design to handle thousands of concurrent users during peak hours.
 - Use load balancers and distributed systems to ensure seamless scaling
- Maintainability:
 - Modular codebase with clear separation of concerns.
 - Comprehensive documentation for each module and API.
- Flexibility:
 - Component-based architecture to allow easy feature addition.
 - Support for multiple content types (video, audio, quizzes).
- Architectural Patterns
 - Microservices Architecture:
 - Break down the platform into independent, loosely-coupled services (e.g., User Management, Content Delivery, Analytics).
 - Benefits:
 - Independent scalability of services.
 - Simplified deployment and maintenance.
 - Event-Driven Architecture:
 - Use an asynchronous messaging system (e.g., RabbitMQ, Kafka) for inter-service communication.
 - Benefits:
 - Real-time updates for features like live quizzes and notifications.
- Cloud-Native Approach:
 - Host the platform on cloud infrastructure (e.g., AWS, Azure, GCP) for high availability and disaster recovery.

- Design Patterns
 - Model-View-Controller (MVC):
 - Separation of concerns for the front-end web and mobile applications.
 - Benefits:
 - Easy maintainability and testing.
 - Factory Pattern:
 - Dynamically create objects for different types of content (e.g., courses, quizzes, assignments).
 - Benefits:
 - Simplified object creation process.
 - Singleton Pattern:
 - Manage shared resources like database connections and configuration settings.
 - Benefits:
 - Prevent redundancy and ensure resource efficiency.
- Security Measures
 - Data Encryption:
 - Encrypt sensitive user data in transit (TLS) and at rest (AES).
 - Access Control:
 - Implement role-based access control (RBAC).
 - Regular Audits:
 - Periodically perform security audits and vulnerability scans.