



SDLC of LMS Portal

👤 Author	 Dipak Rathod  Ajay Gaikwad
🕒 Created time	@January 1, 2025 12:17 PM
📂 subject	Projects

Software Development Life Cycle (SDLC) Documentation for LLMS Portal

1. Planning

Objective

The LLMS Portal project aims to create a comprehensive Learning and Learning Management System that integrates seamlessly with the Quyl Dashboard. The system will provide role-based access, efficient student and course management, and an analytics dashboard.

Deliverables

- Software Requirements Specification (SRS)
- Project timeline
- Resource allocation plan
- Risk assessment

Stakeholders

Client: Educational institutions

End Users: Administrators, instructors, students

Development Team: Frontend and backend developers, QA engineers

2. Requirements Analysis

Functional Requirements

- Role-based authentication and authorization
- CRUD operations for students and courses
- Real-time analytics dashboard
- Light and dark theme management

Non-functional Requirements

- Page load time under 3 seconds
- Scalability to support 10,000 concurrent users
- High responsiveness across devices

Tools and Technologies

Frontend: Next.js 15, TailwindCSS

Backend: Supabase, Prisma ORM

Hosting: Vercel

```

flowchart TD
    A[Planning] --> B[Requirements Analysis]
    B --> C[System Design]
    C --> D[Implementation]
    D --> E[Testing]
    E --> F[Deployment]
    F --> G[Maintenance]
    G --> H[Future Enhancements]
  
```

3. System Design

High-Level Architecture

```

graph TD
    A[Client Browser] --> B[Next.js Frontend]
    B --> C[API Routes]
    C --> D[Prisma ORM]
    D --> E[Supabase Database]
  
```

```
B --> F[Authentication]
F --> G[Supabase Auth]
```

Component Structure

Layer	Components
Presentation	Navbar, Sidebar, Dashboards, Forms
Business Logic	Authentication, State Management, API Services
Data	Supabase Database, Prisma Models

4. Implementation

Development Phases

1. Frontend Development

- Implement user authentication and role-based login
- Develop responsive UI components
- Integrate state management with Zustand

2. Backend Development

- Configure Supabase database schemas
- Implement Prisma models and migrations
- Create API endpoints

5. Testing Strategy

Testing will be conducted in multiple phases to ensure comprehensive coverage and quality assurance.

▼ Testing Phases

- Unit Testing: Individual component testing
- Integration Testing: API and service integration verification
- System Testing: End-to-end functionality testing

- User Acceptance Testing: Stakeholder validation

6. Deployment Plan

```
graph LR
    A[Code Repository] --> B[CI/CD Pipeline]
    B --> C[Build Process]
    C --> D[Testing]
    D --> E[Staging Deploy]
    E --> F[Production Deploy]
```

7. Maintenance

- ☐ Regular system monitoring
- ☐ Bug fixes and patches
- ☐ Performance optimization
- ☐ Security updates

8. Future Enhancements

Planned features include AI-driven course recommendations, mobile applications, and advanced analytics capabilities.

9. Conclusion

This comprehensive SDLC ensures the successful development and deployment of the LLMS Portal, meeting all stakeholder requirements while maintaining high quality and performance standards.