Low Level Design Document

This Low Level Design (LLD) document outlines the core implementation details for **EduVerse - Adaptive Learning Management System**. EduVerse is a scalable frontend platform for adaptive learning, featuring interactive quizzes, real-time progress tracking, and advanced form handling, built with React, Redux Toolkit, and Tailwind CSS.

---

1. System Components

Component	Description / Responsibility
App Shell	Root layout, routing, global providers
Auth Module	Handles authentication, protected routes
Dashboard	Displays user progress, learning paths
Learning Path	Renders adaptive course content
Quiz Engine	Interactive quizzes, answer validation, feedback
Assignment Module	Assignment forms, submission, feedback
Progress Tracker	Real-time progress updates, visualizations
State Management	Redux Toolkit slices for user, course, quiz, progress
UI Components	Reusable UI (buttons, modals, cards) with Tailwind CSS

---

2. Class/Interface Overview

Name	Type	Key Methods / Attributes	Relationships
`User`	Interface	id, name, email, role, progress	Used in Auth, Progress
`Course`	Interface	id, title, modules, adaptiveRules	Used in Learning Path
`Quiz`	Interface	id, questions, userResponses, score	Used in Quiz Engine
`Assignment`	Interface	id, title, status, feedback	Used in Assignment Module
`AuthContext`	Context	login(), logout(), isAuthenticated	Used in Auth, ProtectedRoute
`ProtectedRoute`	Component	-	Wraps secure routes
`useProgress`	Hook	getProgress(), updateProgress()	Used in Progress Tracker

Example: User Interface

```
interface User {
  id: string;
  name: string;
  email: string;
```

```
  role: 'student' | 'instructor';
  progress: Progress[];
}
```

---

3. Data Structure Overview

Model	Fields
User	id, name, email, role, progress[]
Course	id, title, modules[], adaptiveRules[]
Module	id, title, content, quizzes[], assignments[]
Quiz	id, questions[], userResponses[], score
Assignment	id, title, description, status, feedback
Progress	courseId, moduleId, completed, score

---

4. Algorithms / Logic

Adaptive Learning Path Selection (Pseudocode)

```
function getNextModule(userProgress, course) {
  for (const rule of course.adaptiveRules) {
    if (rule.condition(userProgress)) {
      return rule.nextModuleId;
    }
  }
  return course.modules[0].id; // default
}
```

Quiz Submission Flow (Summary)

- Validate answers
- Calculate score
- Update user progress in Redux store
- Show feedback

---

5. Error Handling

Scenario	Handling Approach
Auth failure / token expired	Redirect to login, clear user state
API/network error	Show error toast, retry option
Invalid form input	Inline validation messages, prevent submission

Unauthorized route access	Redirect to login or error page
Data fetch failure	Show fallback UI, log error

End of Document