Product Requirements & Specification Document

Project Name

EduVerse - Adaptive Learning Management System

Description

EduVerse is a scalable, open-source frontend for an education platform, enabling adaptive learning paths, interactive quizzes, and real-time progress tracking. Built with React, Redux Toolkit, and Tailwind CSS, it features dynamic routing, protected routes, and advanced form handling for assignments and feedback.

---

1. Goals & Objectives

Goal	Description
Adaptive Learning	Personalize content and learning paths per user progress
Interactive Quizzes	Enable real-time, auto-graded quizzes with instant feedback
Progress Tracking	Visualize and update user progress in real time
Scalability & Maintainability	Modular, reusable, and open-source codebase
Accessibility & Responsiveness	Fully responsive and accessible UI

---

2. Core Features

Feature	Description
User Authentication	Secure login, registration, and protected routes
Adaptive Dashboard	Personalized dashboard with learning path and progress overview
Course & Module Views	Dynamic routing for courses, modules, and lessons
Quiz Engine	Interactive quizzes with instant feedback and adaptive question flow
Assignment Handling	Advanced forms for assignment submission and feedback
Real-Time Progress	Live progress tracking and visualization
Admin Panel	Manage users, courses, modules, and content

---

3. User Roles & Permissions

Role	Permissions
Student	View courses, take quizzes, submit assignments, track progress

Instructor	Create/edit courses, modules, quizzes, assignments, view student analytics
Admin	Full access: manage users, content, and platform settings

---

4. Technical Specifications

4.1 Stack

Layer	Technology
Frontend	React, Redux Toolkit
Styling	Tailwind CSS
Language	TypeScript, JavaScript
Routing	React Router
Forms	React Hook Form / Formik
State Mgmt	Redux Toolkit
Auth	JWT / OAuth2 (integration ready)
Testing	Jest, React Testing Library

4.2 Architecture

- **Component-based**: Modular, reusable React components
- **State Management**: Centralized via Redux Toolkit
- **Routing**: Dynamic and protected routes using React Router
- **Styling**: Utility-first with Tailwind CSS
- **Type Safety**: TypeScript for all components and state

---

5. Functional Requirements

5.1 Authentication & Authorization

- Secure login, registration, and password reset
- Role-based access control for routes and features

5.2 Adaptive Learning Paths

- Personalized dashboard showing recommended modules
- Dynamic adjustment of learning path based on quiz/assignment performance

5.3 Course & Module Management

- List, view, and navigate courses/modules/lessons
- Instructors can create, edit, and organize content

5.4 Quizzes & Assignments

- Interactive quizzes with multiple question types
- Auto-grading and instant feedback

- Assignment submission with advanced form validation
- Instructor feedback and grading

5.5 Progress Tracking

- Real-time progress bars and analytics
- Visual indicators for completed/incomplete modules

5.6 Admin Panel

- User, course, and content management
- Analytics dashboard

---

6. Non-Functional Requirements

Requirement	Specification
Performance	<200ms UI response time
Scalability	Support 10,000+ concurrent users
Accessibility	WCAG 2.1 AA compliance
Responsiveness	Mobile, tablet, and desktop support
Security	XSS/CSRF protection, secure storage of tokens
Internationalization	Ready for i18n/l10n

---

7. UI/UX Guidelines

- Clean, intuitive, and distraction-free interface
- Consistent use of Tailwind CSS for styling
- Accessible navigation and forms
- Responsive layouts for all devices

---

8. Key Screens (Wireframe References)

Screen	Description
Login/Register	Auth forms with validation
Dashboard	Personalized learning overview
Course/Module View	Content navigation and progress
Quiz/Assignment	Interactive forms and feedback
Admin Panel	Management and analytics

---

9. API & Integration

- RESTful API endpoints for all data operations (to be integrated)
- JWT/OAuth2 authentication support
- WebSocket/Server-Sent Events for real-time updates (optional)

---

10. Acceptance Criteria

- All core features implemented and tested
- 100% TypeScript coverage for components and state
- Responsive and accessible UI
- All routes protected as per user roles
- Open-source codebase with clear documentation

---

11. Milestones & Timeline

Milestone	Timeline
Project Setup	Week 1
Auth & Routing	Week 2-3
Dashboard & Courses	Week 4-5
Quizzes & Assignments	Week 6-7
Progress Tracking	Week 8
Admin Panel	Week 9
Testing & QA	Week 10
Documentation & Release	Week 11-12

---

12. Open Source & Contribution

- MIT License
- Contribution guidelines and code of conduct included
- Public repository with issue tracking

Appendix: Example Component Structure

```
src/
  components/
    Auth/
    Dashboard/
    Courses/
    Quizzes/
    Assignments/
    Admin/
  store/
```

```
    routes/
    utils/
    types/
```
