# CVR COLLEGE OF ENGINEERING

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A Major Project Review-II
on
Web based Tutoring Application
by

Abdul Rab (21B81A05K3)
Koundinya S(21B81A05M4)
Rithvika K (21B81A05N9)

Under guidance of
Mr. M. Hanimi Reddy

# Contents

# Abstract

This project proposes a MERN stack (MongoDB, Express.js, React, Node.js) web platform to bridge the gap between students and tutors by offering integrated services such as Assignment Help, Performance Review, 1-on-1 Mentoring, Tutor Feedback, Exam Preparation, Resource Sharing, and Progress Tracking. The platform aims to streamline educational support through a centralized, user-friendly interface with features like real-time collaboration, and analytics. By addressing fragmentation in existing solutions, the platform enhances accessibility, personalization, and accountability in tutoring services.

# Motivation

- Growing Demand: Rising need for personalized education and remote learning post-pandemic.
- Fragmented Solutions: Existing platforms focus on single services (e.g., assignment help or tutoring), lacking holistic support.
- Quality Concerns: Students struggle to find verified tutors, while tutors lack structured feedback for improvement.

# Literature review

- Existing Platforms:
  - Chegg: Focuses on homework help but lacks interactive mentoring.
  - Wyzant: Offers 1-on-1 tutoring but no performance analytics.
  - Khan Academy: Free resources but no live tutor interaction.
- Research Insights:
  - Studies highlight the importance of real-time feedback and personalized learning paths (Smith et al., 2020).
  - Fragmented services lead to user dissatisfaction (Gupta & Patel, 2019).

# Limitations of the Existing Work- Research Gaps

1. Isolated Services: No platform integrates assignment help, mentoring, and feedback.
2. Poor Matching: Manual tutor selection leads to mismatched expertise.
3. Lack of Analytics: Students/tutors cannot track long-term progress.
4. Scalability Issues: Legacy systems struggle with real-time features (e.g., video calls).
5. Limited Feedback Loops: Tutors lack structured reviews to improve teaching quality.

# Proposed Problem statement

It is an interface between a student and a tutor to help students  by providing various services like:

1. Assignment help
2. Performance review
3. One on One mentoring
4. etc.

# Proposed Approach of the work

1. Agile Methodology: Sprints for iterative development.
2. Modular Architecture:
    - Frontend: React with Redux for state management.
    - Backend: RESTful APIs (Node.js/Express.js) + MongoDB (NoSQL for scalability).
    - Authentication: JWT + OAuth.
3. Third-Party Integrations:
    - Payment Gateway (Stripe/Razorpay).
    - Video API (Zoom/Agora).

# User/Stakeholder Requirement Analysis

Students
Post assignments, book sessions, track progress, review tutors.

Tutors
Create profiles, set availability, accept requests, view feedback.

Admin
Verify tutors, moderate content, generate reports.

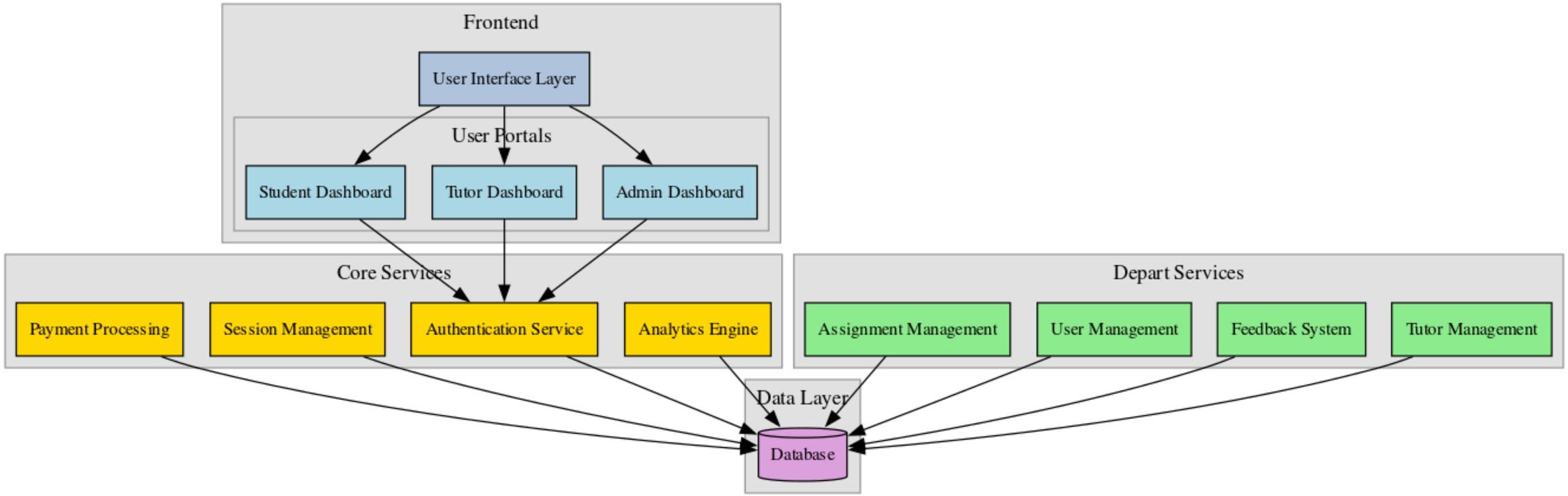
# Software  Requirements/Technology stack

- Frontend: React.js,  Axios.
- Backend: Node.js, Express.js, MongoDB, Mongoose.
- APIs: RESTful for CRUD operations, WebSocket for chat.
- Tools: Postman, Git, VS Code, Figma (UI/UX).

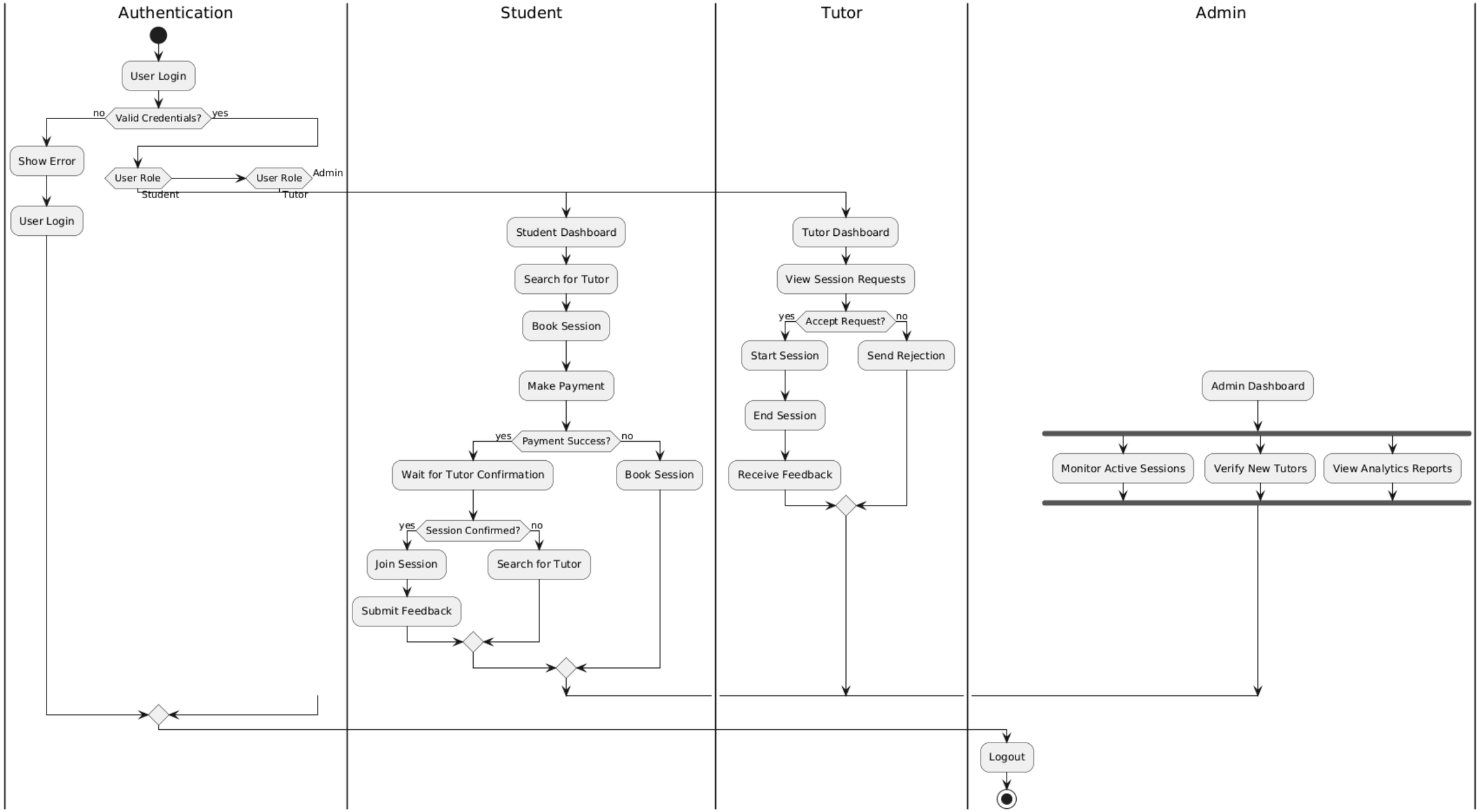
# Hardware requirements

- Server: Cloud hosting (AWS EC2/Heroku) with 4GB RAM, 2vCPU.
- Client: Modern browsers (Chrome/Firefox) with JavaScript enabled.
- Storage: MongoDB Atlas (scalable cloud database).

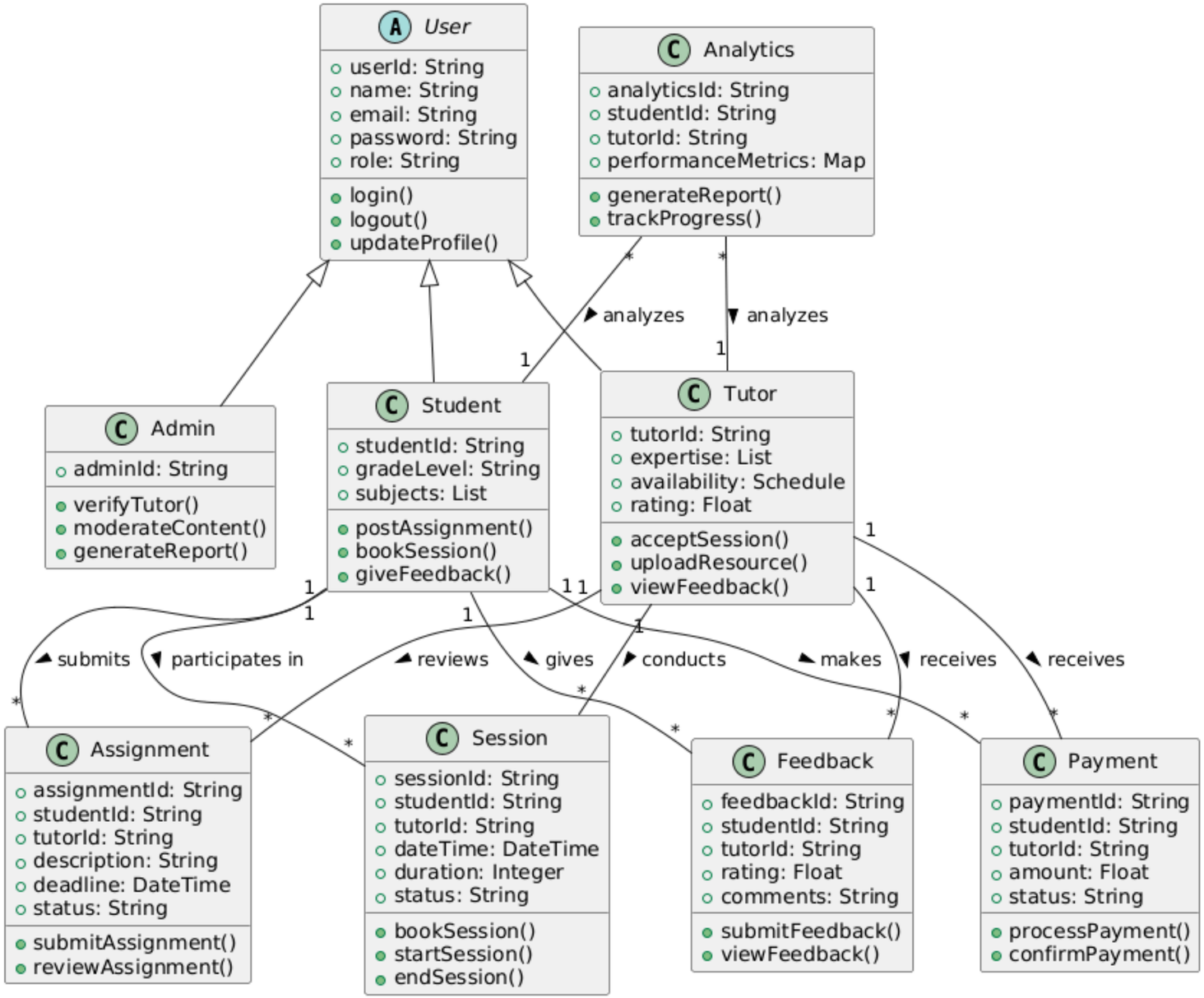
# Architecture of the Model



# Flow / Activity Diagram



# Class Diagram



# Modules in the Code

## Backend

1. **Middleware**
2. **Models**
3. **Routes**
4. **Server.js**

## Frontend

1. **Components**
2. **Context**
3. **Pages**
4. **Utils**

# Main functions and its implementation in the Code

# Snippets of the Code ( can show in laptop)

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';
import { ThemeProvider, createTheme } from '@mui/material';
import CssBaseline from '@mui/material/CssBaseline';
import { AuthProvider, useAuth } from './context/AuthContext';
import Navbar from './components/Navbar';
import Home from './pages/Home';
import Blog from './pages/Blog';
import BookOnline from './pages/BookOnline';
import Login from './pages/Login';
import Register from './pages/Register';
import Profile from './pages/Profile';
import ForgotPassword from './pages/ForgotPassword';
import Dashboard from './pages/Dashboard';
import Footer from './components/Footer';

// Protected Route component
const ProtectedRoute = ({ children }) => {
 const { user, loading } = useAuth();

 if (loading) {
 return null; // or a loading spinner
 }

 if (!user) {
 return <Navigate to="/login" />;
 }

 return children;
};

const theme = createTheme({
 palette: {
 primary: {
 main: '#1976d2',
 },
 secondary: {
 main: '#dc004e',
 },
 },
 typography: {
 fontFamily: '"Roboto", "Helvetica", "Arial", sans-serif',
 h1: {
 fontSize: '2.5rem',
 fontWeight: 600,
 MuiPaper: {
 styleOverrides: {
 root: {
 borderRadius: 12,
 },
 },
 },
 },
});

function App() {
 return (
 <ThemeProvider theme={theme}>
 <CssBaseline />
 <AuthProvider>
 <Router>
 <div className="App">
 <Navbar />
 <Routes>
 <Route path="/" element={<Home />} />
 <Route path="/blog" element={<Blog />} />
 <Route path="/book-online" element={<BookOnline />} />
 <Route path="/login" element={<Login />} />
 <Route path="/register" element={<Register />} />
 <Route path="/forgot-password" element={<ForgotPassword />} />
 <Route
 path="/profile"
 element={
 <ProtectedRoute>
 <Profile />
 </ProtectedRoute>
 }
 />
 <Route
 path="/dashboard"
 element={
 <ProtectedRoute>
 <Dashboard />
 </ProtectedRoute>
 }
 />
 </Routes>
 <Footer />
 </div>
```

# Innovation in the project

1. Unified Dashboard: Integrates tutoring, assignments, and analytics.
2. Real-Time Collaboration Tools: Built-in code editor for coding mentorship.
3. Tutor Skill Analytics: Identifies strengths/weaknesses via feedback.

# Plan of action to complete the project

| S.NO | ITEM | Percentage of Completion |
|---|---|---|
| 1 | Modules | 50 % |
| 2 | Documentation | 30 % |

# References

[1] IEEE Standard for Digitizing Waveform Recorders—IEEE Std. 1057- 1994, SH94245, Dec. 1994.
[2] F. C. Alegria, P. Arpaia, P. Daponte, and A. C. Serra, “An ADC histogram test based on small-amplitude waves,” Measurement, vol. 31, no. 4, pp. 271–279, Jun. 2002.
[3] F. Alegria, P. Arpaia, A. M. da Cruz Serra, and P. Daponte, “Performance analysis of an ADC histogram test using small triangular waves,” IEEE Trans. Instrum. Meas., vol. 51, no. 4, pp. 723–729, Aug. 2002.
[4] F. C. Alegria, P. Arpaia, P. Daponte, and A. C. Serra, “ADC histogram test by triangular small-waves,” in Proc. IEEE Instrumentation Measurement Technology Conf., Budapest, Hungary, May 21–23, 2001, pp. 1690–1695.
[5] F. C. Alegria and A. C. Serra, “Uncertainty in the ADC transition voltages determined with the histogram method,” in Proc. 6th Workshop ADC Modeling Testing, Lisbon, Portugal, Sep. 13–14, 2001, pp. 28–32.
[6] F. C. Alegria, P. Arpaia, P. Daponte, and A. C. Serra, “ADC histogram test using small-amplitude input waves,” in Proc. XVI IMEKO World Congr., vol. X, Vienna, Austria, Sep. 25–28, 2000, pp. 9–14.
[7] J. Blair, “Histogram measurement of ADC nonlinearities using sine waves,” IEEE Trans. Instrum. Meas., vol. 43, no. 3, pp. 373–383, Jun. 1994