**ai-quiz-frontend/public/vite.svg:**

**ai-quiz-frontend/src/api/axiosClient.ts:**

import axios from 'axios';

const axiosClient = axios.create({

  baseURL: import.meta.env.VITE\_API\_URL,

  headers: {

    'Content-Type': 'application/json',

  },

});

axiosClient.interceptors.request.use((config) => {

  const token = localStorage.getItem('access');

  if (token) {

    config.headers['Authorization'] = `Bearer ${token}`;

  }

  return config;

});

export default axiosClient;

**ai-quiz-frontend/src/api/chapterApi.ts:**

import axiosClient from './axiosClient';

export interface Chapter {

  id: number;

  name: string;

}

export const fetchChapters = (subjectId: number) => {

  return axiosClient.get(`/chapters/?subject=${subjectId}`);

};

**ai-quiz-frontend/src/api/materialApi.ts:**

import axiosClient from './axiosClient';

export interface StudyMaterial {

  id: number;

  title: string;

  document: string;

}

export const fetchMaterials = (subchapterId: number): Promise<StudyMaterial[]> => {

  return axiosClient.get(`/materials/?subchapter=${subchapterId}`);

};

export const uploadMaterial = (formData: FormData): Promise<any> => {

  return axiosClient.post('/materials/', formData, {

    headers: { 'Content-Type': 'multipart/form-data' }

  });

};

**ai-quiz-frontend/src/api/quizApi.ts:**

import axiosClient from './axiosClient';

export const generateAIQuiz = (materialId: number, level: string): Promise<{ quiz: string }> => {

  return axiosClient.post(`/generate-quiz/${materialId}/`, { level });

};

export const submitQuizScore = (data: {

  material: number;

  score: number;

  level: string;

  time\_taken: string; // Format: "HH:MM:SS"

}) => {

  return axiosClient.post("/submit-score/", data);

};

export const getLeaderboard = (materialId: number) => {

  return axiosClient.get(`/leaderboard/${materialId}/`);

};

export const getStudentReport = () => {

  return axiosClient.get("/student-report/");

};

**ai-quiz-frontend/src/api/subchapterApi.ts:**

import axiosClient from './axiosClient';

export interface Subchapter {

  id: number;

  name: string;

}

export const fetchSubchapters = (chapterId: number): Promise<Subchapter[]> => {

  return axiosClient.get(`/subchapters/?chapter=${chapterId}`);

};

**ai-quiz-frontend/src/api/subjectApi.ts:**

import axiosClient from './axiosClient.ts';

export interface Subject {

  id: number;

  name: string;

}

export const fetchSubjects = (): Promise<Subject[]> => {

  return axiosClient.get('/subjects/');

};

**ai-quiz-frontend/src/auth/LoginPage.tsx:**

import React, { useEffect, useState } from 'react';

import { useNavigate, useLocation } from 'react-router-dom';

import axios from 'axios';

const LoginPage: React.FC = () => {

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const [role, setRole] = useState<string | null>(null);

  const [error, setError] = useState('');

  const navigate = useNavigate();

  const location = useLocation();

  useEffect(() => {

    const queryParams = new URLSearchParams(location.search);

    const userRole = queryParams.get('role');

    setRole(userRole);

  }, [location]);

  const handleLogin = async (e: React.FormEvent) => {

    e.preventDefault();

    setError('');

    try {

      const response = await axios.post('http://localhost:8000/api/token/', {

        email,

        password,

      });

      const { access, refresh } = response.data;

      localStorage.setItem('access\_token', access);

      localStorage.setItem('refresh\_token', refresh);

      axios.defaults.headers.common['Authorization'] = `Bearer ${access}`;

      const userResponse = await axios.get('http://localhost:8000/api/user/');

      const userRole = userResponse.data.role;

      if (userRole === 'student') {

        navigate('/subjects');

      } else if (userRole === 'teacher' || userRole === 'admin') {

        navigate('/admin/dashboard');

      } else {

        setError('Unknown user role.');

      }

    } catch (err) {

      setError('Invalid credentials. Please try again.');

    }

  };

  const handleNavigateToRegister = () => {

    navigate(`/register?role=${role}`);

  };

  return (

    <div className="min-h-screen bg-blue-50 flex items-center justify-center px-4">

      <div className="max-w-md w-full bg-white p-8 rounded-xl shadow-md">

        <h2 className="text-2xl font-bold text-center text-blue-600 mb-6">Login ({role})</h2>

        <form onSubmit={handleLogin} className="space-y-4">

          <input

            type="email"

            placeholder="Email"

            className="w-full px-4 py-2 border rounded-md focus:ring-2 focus:ring-blue-400"

            value={email}

            onChange={(e) => setEmail(e.target.value)}

            required

          />

          <input

            type="password"

            placeholder="Password"

            className="w-full px-4 py-2 border rounded-md focus:ring-2 focus:ring-blue-400"

            value={password}

            onChange={(e) => setPassword(e.target.value)}

            required

          />

          {error && <p className="text-red-500 text-sm">{error}</p>}

          <button

            type="submit"

            className="w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700 transition"

          >

            Login

          </button>

        </form>

        <div className="mt-4 text-center text-sm">

          Don't have an account?{' '}

          <button onClick={handleNavigateToRegister} className="text-blue-500 hover:underline">

            Register here

          </button>

        </div>

      </div>

    </div>

  );

};

export default LoginPage;

**ai-quiz-frontend/src/auth/RegisterPage.tsx:**

import React, { useEffect, useState } from 'react';

import { useNavigate, useLocation } from 'react-router-dom';

import axios from 'axios';

const RegisterPage: React.FC = () => {

  const [email, setEmail] = useState('');

  const [password, setPassword] = useState('');

  const [role, setRole] = useState('student');

  const [error, setError] = useState('');

  const navigate = useNavigate();

  const location = useLocation();

  useEffect(() => {

    const queryParams = new URLSearchParams(location.search);

    const roleParam = queryParams.get('role');

    if (roleParam) {

      setRole(roleParam);

    }

  }, [location]);

  const handleRegister = async (e: React.FormEvent) => {

    e.preventDefault();

    setError('');

    try {

      await axios.post('http://localhost:8000/api/register/', {

        email,

        password,

        role,

      });

      navigate(`/login?role=${role}`);

    } catch (err) {

      setError('Registration failed. Please try again.');

    }

  };

  return (

    <div className="min-h-screen bg-green-50 flex items-center justify-center px-4">

      <div className="max-w-md w-full bg-white p-8 rounded-xl shadow-md">

        <h2 className="text-2xl font-bold text-center text-green-600 mb-6">Register as {role}</h2>

        <form onSubmit={handleRegister} className="space-y-4">

          <input

            type="email"

            placeholder="Email"

            className="w-full px-4 py-2 border rounded-md focus:ring-2 focus:ring-green-400"

            value={email}

            onChange={(e) => setEmail(e.target.value)}

            required

          />

          <input

            type="password"

            placeholder="Password"

            className="w-full px-4 py-2 border rounded-md focus:ring-2 focus:ring-green-400"

            value={password}

            onChange={(e) => setPassword(e.target.value)}

            required

          />

          {error && <p className="text-red-500 text-sm">{error}</p>}

          <button

            type="submit"

            className="w-full bg-green-600 text-white py-2 rounded-md hover:bg-green-700 transition"

          >

            Register

          </button>

        </form>

      </div>

    </div>

  );

};

export default RegisterPage;

**ai-quiz-frontend/src/components/Navbar.tsx:**

// src/components/Navbar.tsx

import { useAuth } from "../context/authContext";

const Navbar = () => {

  const { user, logout, isAuthenticated } = useAuth(); // Use the hook

  return (

    <nav className="bg-gray-100 p-4 flex justify-between">

      <div>📚 AI Quiz</div>

      {isAuthenticated ? (

        <div>

          Hello, {user?.username}! {/\* Access username safely \*/}

          <button onClick={logout}>Logout</button>

        </div>

      ) : (

        <div>

          <a href="/login" className="mr-4">Login</a>

          <a href="/register">Register</a>

        </div>

      )}

    </nav>

  );

};

export default Navbar;

**ai-quiz-frontend/src/context/authContext.tsx:**

// src/context/authContext.tsx

import { createContext, useState, useEffect, useContext, ReactNode } from "react";

import { jwtDecode } from "jwt-decode";

// Define the shape of the user object decoded from JWT

interface User {

  id: number;

  username: string;

  role: "student" | "admin";

}

// Define the context type

interface AuthContextType {

  user: User | null;

  isAuthenticated: boolean;

  login: (accessToken: string) => void;

  logout: () => void;

}

// Create context with default values

const AuthContext = createContext<AuthContextType>({

  user: null,

  isAuthenticated: false,

  login: () => {},

  logout: () => {},

});

// Define props for AuthProvider

interface AuthProviderProps {

  children: ReactNode;

}

// Auth Provider component

export const AuthProvider: React.FC<AuthProviderProps> = ({ children }) => {

  const [user, setUser] = useState<User | null>(null);

  // Load user from token on initial render

  useEffect(() => {

    const access = localStorage.getItem("access");

    if (access) {

      try {

        const decoded: any = jwtDecode(access);

        const userObj: User = {

          id: decoded.user\_id,

          username: decoded.username,

          role: decoded.role,

        };

        setUser(userObj);

      } catch (error) {

        console.error("Invalid token:", error);

        localStorage.removeItem("access");

      }

    }

  }, []);

  // Login function

  const login = (accessToken: string) => {

    try {

      const decoded: any = jwtDecode(accessToken);

      const userObj: User = {

        id: decoded.user\_id,

        username: decoded.username,

        role: decoded.role,

      };

      setUser(userObj);

      localStorage.setItem("access", accessToken);

    } catch (error) {

      console.error("Login failed:", error);

    }

  };

  // Logout function

  const logout = () => {

    localStorage.removeItem("access");

    localStorage.removeItem("refresh");

    setUser(null);

  };

  return (

    <AuthContext.Provider

      value={{

        user,

        isAuthenticated: !!user,

        login,

        logout,

      }}

    >

      {children}

    </AuthContext.Provider>

  );

};

// Custom hook for using auth context

export const useAuth = () => {

  const context = useContext(AuthContext);

  if (!context) {

    throw new Error("useAuth must be used within an AuthProvider");

  }

  return context;

};

**ai-quiz-frontend/src/pages/RoleSelectionPage.tsx:**

import React from 'react';

import { useNavigate } from 'react-router-dom';

const RoleSelectionPage: React.FC = () => {

  const navigate = useNavigate();

  const handleSelect = (role: string) => {

    navigate(`/login?role=${role}`);

  };

  return (

    <div className="min-h-screen bg-gradient-to-r from-indigo-200 to-blue-300 flex items-center justify-center px-4">

      <div className="bg-white shadow-xl rounded-lg p-8 max-w-md w-full text-center space-y-6">

        <h1 className="text-3xl font-bold text-indigo-700">Select Your Role</h1>

        <p className="text-gray-600">Choose your role to continue to login</p>

        <div className="flex flex-col gap-4">

          <button

            className="py-3 bg-indigo-600 text-white rounded-md hover:bg-indigo-700 transition"

            onClick={() => handleSelect('student')}

          >

            Student

          </button>

          <button

            className="py-3 bg-indigo-600 text-white rounded-md hover:bg-indigo-700 transition"

            onClick={() => handleSelect('teacher')}

          >

            Teacher / Admin

          </button>

        </div>

      </div>

    </div>

  );

};

export default RoleSelectionPage;

**ai-quiz-frontend/src/pages/SubjectPage.tsx:**

import { useEffect, useState } from "react";

import { fetchSubjects } from "../api/subjectApi";

import { Link } from "react-router-dom";

import { motion } from "framer-motion";

interface Subject {

  id: number;

  name: string;

}

export default function SubjectPage() {

  const [subjects, setSubjects] = useState<Subject[]>([]);

  useEffect(() => {

    fetchSubjects()

      .then((res) => setSubjects(res))

      .catch(() => {

        // Fallback data

        setSubjects([

          { id: 1, name: "Mathematics" },

          { id: 2, name: "Science" },

          { id: 3, name: "Bahasa Malaysia" },

          { id: 4, name: "English" }

        ]);

      });

  }, []);

  return (

    <div className="min-h-screen p-8 bg-gradient-to-b from-blue-50 to-white">

      <h1 className="text-3xl font-bold text-center mb-8 text-gray-800">

        Choose Your Subject

      </h1>

      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6 max-w-6xl mx-auto">

        {subjects.map((subject, index) => (

          <motion.div

            key={subject.id}

            initial={{ opacity: 0, y: 20 }}

            animate={{ opacity: 1, y: 0 }}

            transition={{ delay: index \* 0.1 }}

          >

            <Link

              to={`/subjects/${subject.id}/chapters`}

              className="group relative bg-white rounded-2xl shadow-xl p-6 transition-all duration-300 hover:scale-105 hover:shadow-2xl block h-full"

            >

              <div className="absolute inset-0 bg-gradient-to-br from-blue-100 to-transparent opacity-0 group-hover:opacity-100 rounded-2xl transition-opacity" />

              <div className="relative z-10">

                <div className="w-16 h-16 mb-4 mx-auto bg-blue-100 rounded-xl flex items-center justify-center">

                  <span className="text-2xl">📘</span>

                </div>

                <h2 className="text-xl font-semibold text-center text-gray-800">

                  {subject.name}

                </h2>

              </div>

            </Link>

          </motion.div>

        ))}

      </div>

    </div>

  );

}

**ai-quiz-frontend/src/pages/ChapterPage.tsx:**

import { useParams, Link } from "react-router-dom";

import { useEffect, useState } from "react";

import { fetchChapters } from "../api/chapterApi";

import { motion } from "framer-motion";

interface Chapter {

  id: number;

  name: string;

}

export default function ChapterPage() {

  const { subjectId } = useParams<{ subjectId: string }>();

  const [chapters, setChapters] = useState<Chapter[]>([]);

  useEffect(() => {

    if (subjectId) {

      fetchChapters(parseInt(subjectId))

        .then(res => setChapters(res.data))

        .catch(() => {

          // Sample chapters

          const sampleChapters = Array.from({ length: 10 }, (\_, i) => ({

            id: i + 1,

            name: `Chapter ${i + 1}`

          }));

          setChapters(sampleChapters);

        });

    }

  }, [subjectId]);

  return (

    <div className="min-h-screen p-8 bg-gray-50">

      <h1 className="text-3xl font-bold text-center mb-8">Chapters</h1>

      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 max-w-6xl mx-auto">

        {chapters.map((chapter, index) => (

          <motion.div

            key={chapter.id}

            initial={{ opacity: 0, x: -20 }}

            animate={{ opacity: 1, x: 0 }}

            transition={{ delay: index \* 0.1 }}

          >

            <Link

              to={`/chapters/${chapter.id}/subchapters`}

              className="bg-white p-6 rounded-xl shadow-md hover:shadow-lg transition-shadow duration-300 block h-full"

            >

              <div className="flex items-center gap-4">

                <div className="bg-blue-100 p-3 rounded-lg">

                  <span className="text-2xl">📖</span>

                </div>

                <h2 className="text-xl font-semibold">{chapter.name}</h2>

              </div>

            </Link>

          </motion.div>

        ))}

      </div>

    </div>

  );

}

**ai-quiz-frontend/src/pages/SubchapterPage.tsx:**

import { useParams, Link } from "react-router-dom";

import { useEffect, useState } from "react";

import { fetchSubchapters } from "../api/subchapterApi";

interface Subchapter {

  id: number;

  name: string;

}

export default function SubchapterPage() {

  const { chapterId } = useParams<{ chapterId: string }>();

  const [subs, setSubs] = useState<Subchapter[]>([]);

  useEffect(() => {

    if (chapterId) {

      fetchSubchapters(parseInt(chapterId))

        .then((res: Subchapter[]) => setSubs(res))

        .catch((err: Error) => console.error(err));    }

  }, [chapterId]);

  return (

    <div className="p-6">

      <h2 className="text-xl font-semibold mb-4">Subchapters</h2>

      <div className="grid grid-cols-1 gap-3">

        {subs.map(sub => (

          <Link

            key={sub.id}

            to={`/subchapter/${sub.id}`}

            className="p-4 bg-yellow-100 hover:bg-yellow-200 rounded-xl shadow"

          >

            {sub.name}

          </Link>

        ))}

      </div>

    </div>

  );

}

**ai-quiz-frontend/src/pages/StudyMaterialPage.tsx:**

import { useParams } from "react-router-dom";

import { useEffect, useState } from "react";

import { fetchMaterials, StudyMaterial, uploadMaterial } from "../api/materialApi";

import { generateAIQuiz } from "../api/quizApi";

import { motion } from "framer-motion";

export default function StudyMaterialPage() {

  const { subchapterId } = useParams<{ subchapterId: string }>();

  const [materials, setMaterials] = useState<StudyMaterial[]>([]);

  const [title, setTitle] = useState<string>('');

  const [file, setFile] = useState<File | null>(null);

  const [selectedLevel, setSelectedLevel] = useState<string>("Beginner");

  const [aiQuiz, setAiQuiz] = useState<string>("");

  const [loadingQuiz, setLoadingQuiz] = useState<boolean>(false);

  const [error, setError] = useState<string | null>(null);

  const isTeacher = localStorage.getItem("is\_staff") === "true";

  useEffect(() => {

    if (subchapterId) {

      fetchMaterials(parseInt(subchapterId))

        .then(res => setMaterials(res))

        .catch(err => console.error(err));

    }

  }, [subchapterId]);

  const handleUpload = async () => {

    if (!title || !file) {

      alert("Please provide both title and file.");

      return;

    }

    const formData = new FormData();

    formData.append("title", title);

    formData.append("document", file);

    formData.append("subchapter", subchapterId!);

    try {

      await uploadMaterial(formData);

      alert("Uploaded!");

      setTitle("");

      setFile(null);

      if (subchapterId) {

        fetchMaterials(parseInt(subchapterId)).then(res => setMaterials(res));

      }

    } catch (e) {

      alert("Error uploading file");

    }

  };

  const handleGenerateQuiz = async (materialId: number) => {

    setLoadingQuiz(true);

    setAiQuiz("");

    setError(null);

    try {

      const res = await generateAIQuiz(materialId, selectedLevel);

      setAiQuiz(res.quiz);

    } catch (err) {

      setError("Failed to generate quiz. Please try again.");

    } finally {

      setLoadingQuiz(false);

    }

  };

  return (

    <div className="p-6">

      <h2 className="text-xl font-semibold mb-4">Study Materials</h2>

      {materials.length === 0 ? (

        <motion.div

          initial={{ scale: 0.8 }}

          animate={{ scale: 1 }}

          className="text-center p-8 bg-white rounded-xl shadow-sm max-w-md mx-auto"

        >

          <div className="text-6xl mb-4">📭</div>

          <h3 className="text-xl font-semibold mb-2">No Materials Available</h3>

          <p className="text-gray-500">

            The teacher hasn't uploaded any study materials yet.

          </p>

        </motion.div>

      ) : (

        <motion.div

          initial={{ opacity: 0 }}

          animate={{ opacity: 1 }}

          className="space-y-4"

        >

          {materials.map((m, index) => (

            <motion.div

              key={m.id}

              initial={{ y: 20, opacity: 0 }}

              animate={{ y: 0, opacity: 1 }}

              transition={{ delay: index \* 0.1 }}

              className="p-3 border rounded bg-white shadow-sm"

            >

              <div className="mb-2">

                📄 <a href={m.document} target="\_blank" rel="noopener noreferrer">{m.title}</a>

              </div>

              <div className="flex flex-col md:flex-row items-start md:items-center gap-2 mb-2">

                <select

                  className="border p-1 rounded"

                  value={selectedLevel}

                  onChange={(e) => setSelectedLevel(e.target.value)}

                >

                  <option value="Beginner">Beginner</option>

                  <option value="Intermediate">Intermediate</option>

                  <option value="Advanced">Advanced</option>

                </select>

                <button

                  onClick={() => handleGenerateQuiz(m.id)}

                  className="bg-indigo-600 text-white px-4 py-2 rounded hover:bg-indigo-700"

                >

                  Generate AI Quiz

                </button>

              </div>

            </motion.div>

          ))}

        </motion.div>

      )}

      {loadingQuiz && <p className="text-blue-500 mt-4">Generating quiz...</p>}

      {error && <p className="text-red-500 mt-2">{error}</p>}

      {aiQuiz && (

        <div className="mt-6 p-4 border rounded bg-gray-100 whitespace-pre-wrap">

          <h3 className="font-bold mb-2">Generated Quiz:</h3>

          <div>{aiQuiz}</div>

        </div>

      )}

      {isTeacher && (

        <div className="p-4 border rounded bg-gray-100 mt-8">

          <h3 className="font-bold mb-2">Upload New Material</h3>

          <input

            type="text"

            placeholder="Title"

            className="border px-2 py-1 mb-2 block w-full"

            value={title}

            onChange={(e) => setTitle(e.target.value)}

          />

          <input

            type="file"

            className="mb-2"

            onChange={(e) => setFile(e.target.files ? e.target.files[0] : null)}

          />

          <button onClick={handleUpload} className="bg-blue-500 text-white px-4 py-2 rounded">

            Upload

          </button>

        </div>

      )}

    </div>

  );

}

**ai-quiz-frontend/src/pages/QuizPage.tsx:**

import { useState, useEffect } from "react";

import { useParams } from "react-router-dom";

import axiosClient from "../api/axiosClient";

const QuizPage = () => {

  const { subchapterId } = useParams();

  const [quiz, setQuiz] = useState<any>(null);

  const [answers, setAnswers] = useState<any>({});

  const [submitted, setSubmitted] = useState(false);

  useEffect(() => {

    fetchQuiz();

  }, [subchapterId]);

  const fetchQuiz = async () => {

    try {

      const res = await axiosClient.get(`/quizzes/?subchapter=${subchapterId}`);

      setQuiz(res.data);

    } catch (err) {

      console.error("Error fetching quiz", err);

    }

  };

  const handleAnswerChange = (questionId: number, answer: string) => {

    setAnswers((prevAnswers: any) => ({

      ...prevAnswers,

      [questionId]: answer,

    }));

  };

  const handleSubmit = async () => {

    setSubmitted(true);

    try {

      // Send answers to the backend for evaluation

      await axiosClient.post("/quizzes/submit", { subchapterId, answers });

    } catch (err) {

      console.error("Error submitting quiz", err);

    }

  };

  return (

    <div className="p-6 bg-gray-50 min-h-screen">

      <h2 className="text-2xl font-bold mb-4">📚 Quiz</h2>

      {quiz ? (

        <div>

          {quiz.questions.map((question: any) => (

            <div key={question.id} className="mb-6">

              <h3 className="font-semibold text-lg">{question.text}</h3>

              <div className="space-y-2">

                {question.options.map((option: string, index: number) => (

                  <label key={index} className="block">

                    <input

                      type="radio"

                      name={`question-${question.id}`}

                      value={option}

                      onChange={() => handleAnswerChange(question.id, option)}

                      disabled={submitted}

                      className="mr-2"

                    />

                    {option}

                  </label>

                ))}

              </div>

            </div>

          ))}

          {!submitted ? (

            <button

              onClick={handleSubmit}

              className="bg-blue-600 text-white px-4 py-2 rounded mt-6"

            >

              Submit Quiz

            </button>

          ) : (

            <p className="mt-4 text-green-500">Quiz Submitted!</p>

          )}

        </div>

      ) : (

        <p>Loading quiz...</p>

      )}

    </div>

  );

};

export default QuizPage;

**ai-quiz-frontend/src/pages/StudentReport.tsx:**

import { useEffect, useState } from "react";

import { getStudentReport } from "../api/quizApi";

interface ReportEntry {

  attempts: number;

  avg\_score: number;

  levels: string[];

  recommendation: string;

}

const StudentReport = () => {

  const [report, setReport] = useState<Record<string, ReportEntry>>({});

  useEffect(() => {

    getStudentReport().then(res => setReport(res.data));

  }, []);

  return (

    <div className="p-6">

      <h2 className="text-xl font-bold mb-4">📊 Your Performance Report</h2>

      {Object.entries(report).map(([title, data]) => (

        <div key={title} className="mb-4 p-4 border rounded bg-white shadow-sm">

          <h3 className="text-lg font-semibold">{title}</h3>

          <p>📈 Average Score: {data.avg\_score.toFixed(2)}%</p>

          <p>🎯 Attempts: {data.attempts}</p>

          <p>⚙️ Levels Tried: {data.levels.join(", ")}</p>

          <p className="text-blue-600 font-medium mt-2">💡 AI Tip: {data.recommendation}</p>

        </div>

      ))}

    </div>

  );

};

export default StudentReport;

**ai-quiz-frontend/src/pages/AdminChaptersPage.tsx:**

import { useEffect, useState } from "react";

import { useParams } from "react-router-dom";

import axiosClient from "../api/axiosClient";

import { Link } from "react-router-dom";

const AdminChaptersPage = () => {

  const { subjectId } = useParams();

  const [chapters, setChapters] = useState([]);

  const [newChapterName, setNewChapterName] = useState("");

  useEffect(() => {

    fetchChapters();

  }, [subjectId]);

  const fetchChapters = async () => {

    try {

      const res = await axiosClient.get(`/chapters/?subject=${subjectId}`);

      setChapters(res.data);

    } catch (err) {

      console.error("Failed to fetch chapters", err);

    }

  };

  const handleAddChapter = async () => {

    if (!newChapterName.trim()) return;

    try {

      await axiosClient.post("/chapters/", {

        name: newChapterName,

        subject: subjectId,

      });

      setNewChapterName("");

      fetchChapters();

    } catch (err) {

      console.error("Failed to add chapter", err);

    }

  };

  const handleDeleteChapter = async (id: number) => {

    try {

      await axiosClient.delete(`/chapters/${id}/`);

      fetchChapters();

    } catch (err) {

      console.error("Failed to delete chapter", err);

    }

  };

  return (

    <div className="p-6 bg-gray-50 min-h-screen">

      <h1 className="text-2xl font-bold mb-4">📚 Manage Chapters</h1>

      <div className="mb-4 flex gap-2">

        <input

          type="text"

          value={newChapterName}

          onChange={(e) => setNewChapterName(e.target.value)}

          placeholder="Enter chapter name"

          className="border p-2 rounded w-64"

        />

        <button

          onClick={handleAddChapter}

          className="bg-green-500 text-white px-4 py-2 rounded hover:bg-green-600"

        >

          ➕ Add Chapter

        </button>

      </div>

      <ul className="space-y-3">

        {chapters.map((chapter: any) => (

            <><Link

                to={`/admin/chapters/${chapter.id}/subchapters`}

                className="text-blue-500 hover:underline"

            >

                📂 Manage Subchapters

            </Link><li

                key={chapter.id}

                className="bg-white p-4 shadow rounded flex justify-between items-center"

            >

                    <span>{chapter.name}</span>

                    <button

                        onClick={() => handleDeleteChapter(chapter.id)}

                        className="text-red-500 hover:underline"

                    >

                        ❌ Delete

                    </button>

                </li></>

        ))}

      </ul>

    </div>

  );

};

export default AdminChaptersPage;

**ai-quiz-frontend/src/pages/AdminDashboardPage.tsx:**

import { useNavigate } from "react-router-dom";

const AdminDashboardPage = () => {

  const navigate = useNavigate();

  return (

    <div className="min-h-screen bg-gray-50 p-8">

      <h1 className="text-3xl font-bold mb-6">👩‍🏫 Admin Dashboard</h1>

      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">

        <button

          onClick={() => navigate("/admin/subjects")}

          className="bg-blue-500 text-white p-4 rounded shadow hover:bg-blue-600"

        >

          📘 Manage Subjects

        </button>

        <button

          onClick={() => navigate("/admin/chapters")}

          className="bg-green-500 text-white p-4 rounded shadow hover:bg-green-600"

        >

          📚 Manage Chapters

        </button>

        <button

          onClick={() => navigate("/admin/subchapters")}

          className="bg-yellow-500 text-white p-4 rounded shadow hover:bg-yellow-600"

        >

          🧾 Manage Subchapters

        </button>

        <button

          onClick={() => navigate("/admin/materials")}

          className="bg-purple-500 text-white p-4 rounded shadow hover:bg-purple-600"

        >

          📂 Manage Study Materials

        </button>

        <button

          onClick={() => navigate("/admin/reports")}

          className="bg-red-500 text-white p-4 rounded shadow hover:bg-red-600"

        >

          📊 View Student Reports

        </button>

      </div>

    </div>

  );

};

export default AdminDashboardPage;

**ai-quiz-frontend/src/pages/AdminMaterialsPage.tsx:**

import React, { useEffect, useState } from "react";

import { useParams } from "react-router-dom";

import axiosClient from "../api/axiosClient";

import { generateAIQuiz } from "../api/quizApi";

const AdminMaterialsPage = () => {

  const { subchapterId } = useParams();

  const [materials, setMaterials] = useState([]);

  const [file, setFile] = useState<File | null>(null);

  const [quiz, setQuiz] = useState<any>(null);

  useEffect(() => {

    fetchMaterials();

  }, [subchapterId]);

  const fetchMaterials = async () => {

    try {

      const res = await axiosClient.get(`/materials/?subchapter=${subchapterId}`);

      setMaterials(res.data);

    } catch (err) {

      console.error("Failed to load materials", err);

    }

  };

  const handleUpload = async (e: React.FormEvent) => {

    e.preventDefault();

    if (!file) return;

    const formData = new FormData();

    formData.append("file", file);

    formData.append("subchapter", subchapterId!);

    try {

      await axiosClient.post("/materials/", formData, {

        headers: { "Content-Type": "multipart/form-data" },

      });

      setFile(null);

      fetchMaterials();

    } catch (err) {

      console.error("Upload failed", err);

    }

  };

  const handleDelete = async (id: number) => {

    try {

      await axiosClient.delete(`/materials/${id}/`);

      fetchMaterials();

    } catch (err) {

      console.error("Delete failed", err);

    }

  };

  const handleGenerateQuiz = async (material: any) => {

    try {

      const response = await generateAIQuiz(material.id, "Beginner");

      setQuiz(response.quiz);

    } catch (err) {

      console.error("Quiz generation failed", err);

    }

  };

  return (

    <div className="p-6 bg-gray-50 min-h-screen">

      <h2 className="text-2xl font-bold mb-4">📑 Manage Study Materials</h2>

      <form onSubmit={handleUpload} className="mb-4 flex gap-4 items-center">

        <input

          type="file"

          onChange={(e) => setFile(e.target.files?.[0] || null)}

          className="border p-2"

        />

        <button

          type="submit"

          className="bg-blue-600 text-white px-4 py-2 rounded hover:bg-blue-700"

        >

          ⬆️ Upload Material

        </button>

      </form>

      <ul className="space-y-3">

        {materials.map((material: any) => (

          <li

            key={material.id}

            className="bg-white p-4 shadow rounded flex justify-between items-center"

          >

            <a

              href={material.file}

              target="\_blank"

              rel="noopener noreferrer"

              className="text-blue-600 hover:underline"

            >

              {material.file.split("/").pop()}

            </a>

            <div className="space-x-4">

              <button

                onClick={() => handleGenerateQuiz(material)}

                className="text-green-500 hover:underline"

              >

                🔄 Generate Quiz

              </button>

              <button

                onClick={() => handleDelete(material.id)}

                className="text-red-500 hover:underline"

              >

                ❌ Delete

              </button>

            </div>

          </li>

        ))}

      </ul>

      {quiz && (

        <div className="mt-6">

          <h3 className="text-xl font-semibold">AI-Generated Quiz:</h3>

          <pre className="bg-white p-4 rounded shadow mt-2 whitespace-pre-wrap">{quiz}</pre>

        </div>

      )}

    </div>

  );

};

export default AdminMaterialsPage;

**ai-quiz-frontend/src/pages/AdminSubchapterPage.tsx:**

import { useEffect, useState } from "react";

import { useParams } from "react-router-dom";

import axiosClient from "../api/axiosClient";

import { Link } from "react-router-dom";

const AdminSubchaptersPage = () => {

  const { chapterId } = useParams();

  const [subchapters, setSubchapters] = useState([]);

  const [newSubchapterName, setNewSubchapterName] = useState("");

  useEffect(() => {

    fetchSubchapters();

  }, [chapterId]);

  const fetchSubchapters = async () => {

    try {

      const res = await axiosClient.get(`/subchapters/?chapter=${chapterId}`);

      setSubchapters(res.data);

    } catch (err) {

      console.error("Failed to fetch subchapters", err);

    }

  };

  const handleAddSubchapter = async () => {

    if (!newSubchapterName.trim()) return;

    try {

      await axiosClient.post("/subchapters/", {

        name: newSubchapterName,

        chapter: chapterId,

      });

      setNewSubchapterName("");

      fetchSubchapters();

    } catch (err) {

      console.error("Failed to add subchapter", err);

    }

  };

  const handleDeleteSubchapter = async (id: number) => {

    try {

      await axiosClient.delete(`/subchapters/${id}/`);

      fetchSubchapters();

    } catch (err) {

      console.error("Failed to delete subchapter", err);

    }

  };

  return (

    <div className="p-6 bg-gray-50 min-h-screen">

      <h1 className="text-2xl font-bold mb-4">📂 Manage Subchapters</h1>

      <div className="mb-4 flex gap-2">

        <input

          type="text"

          value={newSubchapterName}

          onChange={(e) => setNewSubchapterName(e.target.value)}

          placeholder="Enter subchapter name"

          className="border p-2 rounded w-64"

        />

        <button

          onClick={handleAddSubchapter}

          className="bg-green-500 text-white px-4 py-2 rounded hover:bg-green-600"

        >

          ➕ Add Subchapter

        </button>

      </div>

      <ul className="space-y-3">

        {subchapters.map((subchapter: any) => (

          <li

            key={subchapter.id}

            className="bg-white p-4 shadow rounded flex justify-between items-center"

          >

            <span>{subchapter.name}</span>

            <Link

              to={`/admin/subchapters/${subchapter.id}/materials`}

              className="text-blue-500 hover:underline"

              >

              📁 Manage Materials

            </Link>

            <button

              onClick={() => handleDeleteSubchapter(subchapter.id)}

              className="text-red-500 hover:underline"

            >

              ❌ Delete

            </button>

          </li>

        ))}

      </ul>

    </div>

  );

};

export default AdminSubchaptersPage;

**ai-quiz-frontend/src/pages/AdminSubjectsPage.tsx:**

import { useEffect, useState } from "react";

import axiosClient from "../api/axiosClient";

import {Link } from 'react-router-dom';

const AdminSubjectsPage = () => {

  const [subjects, setSubjects] = useState([]);

  const [newSubject, setNewSubject] = useState("");

  // Fetch subjects on page load

  useEffect(() => {

    fetchSubjects();

  }, []);

  const fetchSubjects = async () => {

    try {

      const res = await axiosClient.get("/subjects/");

      setSubjects(res.data);

    } catch (err) {

      console.error("Failed to fetch subjects", err);

    }

  };

  const handleAddSubject = async () => {

    if (!newSubject.trim()) return;

    try {

      await axiosClient.post("/subjects/", { name: newSubject });

      setNewSubject("");

      fetchSubjects();

    } catch (err) {

      console.error("Failed to add subject", err);

    }

  };

  const handleDeleteSubject = async (id: number) => {

    try {

      await axiosClient.delete(`/subjects/${id}/`);

      fetchSubjects();

    } catch (err) {

      console.error("Failed to delete subject", err);

    }

  };

  return (

    <div className="p-6 bg-gray-100 min-h-screen">

      <h1 className="text-2xl font-bold mb-6">📘 Manage Subjects</h1>

      <div className="mb-4 flex gap-2">

        <input

          type="text"

          value={newSubject}

          onChange={(e) => setNewSubject(e.target.value)}

          placeholder="Enter subject name"

          className="border p-2 rounded w-64"

        />

        <button

          onClick={handleAddSubject}

          className="bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-600"

        >

          ➕ Add Subject

        </button>

      </div>

      <ul className="space-y-3">

        {subjects.map((subject: any) => (

  <li

    key={subject.id}

    className="bg-white p-4 shadow rounded flex justify-between items-center"

  >

    <span>{subject.name}</span>

    <div className="flex gap-2">

      <Link

        to={`/admin/subjects/${subject.id}/chapters`}

        className="text-blue-500 hover:underline"

      >

        🧩 Manage Chapters

      </Link>

      <button

        onClick={() => handleDeleteSubject(subject.id)}

        className="text-red-500 hover:underline"

      >

        ❌ Delete

      </button>

    </div>

  </li>

))}

      </ul>

    </div>

  );

};

export default AdminSubjectsPage;

**ai-quiz-frontend/src/pages/Leaderboard.tsx:**

import { useEffect, useState } from "react";

import { useParams } from "react-router-dom";

import { getLeaderboard } from "../api/quizApi";

interface ScoreEntry {

  id: number;

  student: number;

  score: number;

  level: string;

  time\_taken: string;

  created\_at: string;

}

const Leaderboard = () => {

  const { materialId } = useParams<{ materialId: string }>();

  const [entries, setEntries] = useState<ScoreEntry[]>([]);

  useEffect(() => {

    if (materialId) {

      const materialIdNum = parseInt(materialId);

      getLeaderboard(materialIdNum).then(res => setEntries(res.data));

    }

  }, [materialId]);

  return (

    <div className="p-6">

      <h2 className="text-xl font-bold mb-4">🏆 Leaderboard</h2>

      <table className="w-full border table-auto">

        <thead>

          <tr className="bg-gray-200">

            <th className="p-2">#</th>

            <th className="p-2">Score</th>

            <th className="p-2">Level</th>

            <th className="p-2">Time</th>

            <th className="p-2">Date</th>

          </tr>

        </thead>

        <tbody>

          {entries.map((entry: ScoreEntry, index) => (

            <tr key={entry.id} className="text-center border-t">

              <td>{index + 1}</td>

              <td>{entry.score}</td>

              <td>{entry.level}</td>

              <td>{entry.time\_taken}</td>

              <td>{new Date(entry.created\_at).toLocaleDateString()}</td>

            </tr>

          ))}

        </tbody>

      </table>

    </div>

  );

};

export default Leaderboard;

**ai-quiz-frontend/src/pages/NoutfoundPage.tsx:**

import React from 'react';

const NotFoundPage: React.FC = () => {

  return (

    <div className="min-h-screen flex items-center justify-center">

      <h1 className="text-3xl font-bold">404 - Page Not Found</h1>

    </div>

  );

};

export default NotFoundPage;

**ai-quiz-frontend/src/App.tsx:**

import { Routes, Route, useLocation } from "react-router-dom";

// Pages

import RoleSelectionPage from "./pages/RoleSelectionPage";

import SubjectPage from "./pages/SubjectPage";

import ChapterPage from "./pages/ChapterPage";

import SubchapterPage from "./pages/SubchapterPage";

import StudyMaterialPage from "./pages/StudyMaterialPage";

import LoginPage from "./auth/LoginPage";

import RegisterPage from "./auth/RegisterPage";

import AdminDashboardPage from "./pages/AdminDashboardPage";

import AdminSubjectsPage from "./pages/AdminSubjectsPage";

import AdminChaptersPage from "./pages/AdminChaptersPage";

import AdminSubchaptersPage from "./pages/AdminSubchaptersPage";

import AdminMaterialsPage from "./pages/AdminMaterialsPage";

import Leaderboard from "./pages/Leaderboard";

import StudentReport from "./pages/StudentReport";

import NotFoundPage from "./pages/NotFoundPage";

// Auth and Navbar

import { AuthProvider } from "./context/authContext";

import Navbar from "./components/Navbar";

import ProtectedRoute from "./utils/ProtectedRoute";

export default function App() {

  const location = useLocation();

  // Hide navbar on login/register/landing pages

  const hideNavbar = [

    "/",

    "/login",

    "/register",

  ].some(path => location.pathname.startsWith(path));

  return (

    <AuthProvider>

      {!hideNavbar && <Navbar />}

      <Routes>

        {/\* 🎯 Role Selection \*/}

        <Route path="/" element={<RoleSelectionPage />} />

        {/\* 🔐 Unified Auth Flow \*/}

        <Route path="/login" element={<LoginPage />} />

        <Route path="/register" element={<RegisterPage />} />

        {/\* 📚 Student Learning Workflow \*/}

        <Route

          path="/subjects"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <SubjectPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/subjects/:subjectId/chapters"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <ChapterPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/chapters/:chapterId/subchapters"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <SubchapterPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/subchapters/:subchapterId/materials"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <StudyMaterialPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/leaderboard/:materialId"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <Leaderboard />

            </ProtectedRoute>

          }

        />

        <Route

          path="/student-report"

          element={

            <ProtectedRoute allowedRoles={['student']}>

              <StudentReport />

            </ProtectedRoute>

          }

        />

        {/\* 🧑🏫 Admin Dashboard \*/}

        <Route

          path="/admin/dashboard"

          element={

            <ProtectedRoute allowedRoles={['admin', 'teacher']}>

              <AdminDashboardPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/admin/subjects"

          element={

            <ProtectedRoute allowedRoles={['admin', 'teacher']}>

              <AdminSubjectsPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/admin/subjects/:subjectId/chapters"

          element={

            <ProtectedRoute allowedRoles={['admin', 'teacher']}>

              <AdminChaptersPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/admin/chapters/:chapterId/subchapters"

          element={

            <ProtectedRoute allowedRoles={['admin', 'teacher']}>

              <AdminSubchaptersPage />

            </ProtectedRoute>

          }

        />

        <Route

          path="/admin/subchapters/:subchapterId/materials"

          element={

            <ProtectedRoute allowedRoles={['admin', 'teacher']}>

              <AdminMaterialsPage />

            </ProtectedRoute>

          }

        />

        {/\* 404 Not Found \*/}

        <Route path="\*" element={<NotFoundPage />} />

      </Routes>

    </AuthProvider>

  );

}

**ai-quiz-frontend/src/utils/extractContent.ts:**

import \* as pdfjsLib from 'pdfjs-dist';

import mammoth from 'mammoth';

export const extractTextFromPdf = async (file: File) => {

  const reader = new FileReader();

  return new Promise<string>((resolve, reject) => {

    reader.onload = async () => {

      const typedArray = new Uint8Array(reader.result as ArrayBuffer);

      const pdf = await pdfjsLib.getDocument(typedArray).promise;

      let textContent = '';

      const numPages = pdf.numPages;

      for (let pageNum = 1; pageNum <= numPages; pageNum++) {

        const page = await pdf.getPage(pageNum);

        const text = await page.getTextContent();

        textContent += text.items.map((item: any) => item.str).join(' ') + '\n';

      }

      resolve(textContent);

    };

    reader.onerror = (error) => reject(error);

    reader.readAsArrayBuffer(file);

  });

};

export const extractTextFromDocx = async (file: File) => {

  const reader = new FileReader();

  return new Promise<string>((resolve, reject) => {

    reader.onload = async () => {

      const arrayBuffer = reader.result as ArrayBuffer;

      try {

        const result = await mammoth.extractRawText({ arrayBuffer });

        resolve(result.value);

      } catch (error) {

        reject(error);

      }

    };

    reader.onerror = (error) => reject(error);

    reader.readAsArrayBuffer(file);

  });

};

**ai-quiz-frontend/src/utils/ProtectedRoute.tsx:**

import React from 'react';

import { Navigate } from 'react-router-dom';

import  useAuth  from '../context/authContext';

interface ProtectedRouteProps {

  children: React.ReactNode;

  allowedRoles: string[];

}

const ProtectedRoute: React.FC<ProtectedRouteProps> = ({ children, allowedRoles }) => {

  const { user } = useAuth();

  if (!user) {

    // User is not authenticated

    return <Navigate to="/login" replace />;

  }

  if (!allowedRoles.includes(user.role)) {

    // User does not have the required role

    return <Navigate to="/" replace />;

  }

  return <>{children}</>;

};

export default ProtectedRoute;

**ai-quiz-frontend/.env:**

VITE\_API\_URL=http://localhost:8000