Edututor AI: PERSONALIZED LEARNING WITH GENERATIVE AI And LMS INTEGRATION PROJECT

✅ 1. Project Folder Structure

```
pgsql
CopyEdit
edututor-ai/
├── backend/
│   ├── main.py
│   ├── models.py
│   ├── schemas.py
│   ├── auth.py
│   └── ai_engine.py
├── frontend/
│   └── React components (Login, Dashboard, etc.)
└── requirements.txt
```

✅ 2. Backend Code (FastAPI)

main.py

```
python
CopyEdit
from fastapi import FastAPI, Depends, HTTPException
from fastapi.middleware.cors import CORSMiddleware
from auth import get_current_user
from ai_engine import generate_quiz
from schemas import User, QuizRequest, QuizResponse

app = FastAPI()

# CORS setup
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
```

```
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

@app.get("/")
def root():
    return {"message": "Welcome to Edututor AI"}

@app.post("/generate-quiz", response_model=QuizResponse)
def generate_ai_quiz(data: QuizRequest, user: User =
Depends(get_current_user)):
    return generate_quiz(data.topic)
```

---

schemas.py

```
python
CopyEdit
from pydantic import BaseModel

class User(BaseModel):
    email: str

class QuizRequest(BaseModel):
    topic: str

class QuizResponse(BaseModel):
    questions: list[str]
```

---

auth.py (Mocked)

```
python
CopyEdit
from fastapi import Depends

def get_current_user():
    # In real life, verify JWT or OAuth
    return {"email": "user@example.com"}
```

---

ai_engine.py (Mocked AI Quiz Generator)

```
python
CopyEdit
def generate_quiz(topic: str):
    # Placeholder for OpenAI or LLM call
    return {
        "questions": [
            f"What is {topic}?",
            f"Explain key concepts in {topic}.",
            f"Why is {topic} important in learning?"
        ]
    }
```

✅ 3. Frontend Code (React – simplified)

App.js

```
jsx
CopyEdit
import React, { useState } from 'react';

function App() {
  const [topic, setTopic] = useState("");
  const [quiz, setQuiz] = useState([]);

  const handleGenerate = async () => {
    const res = await fetch("http://localhost:8000/generate-quiz", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ topic })
    });
    const data = await res.json();
    setQuiz(data.questions);
  };

  return (
    <div className="App">
      <h1>Edututor AI</h1>
      <input
        type="text"
        value={topic}
        onChange={(e) => setTopic(e.target.value)}
        placeholder="Enter topic"
      />
      <button onClick={handleGenerate}>Generate Quiz</button>
      <ul>
        {quiz.map((q, index) => <li key={index}>{q}</li>)}
      </ul>
    </div>
  );
}

export default App;
```

✅ 4. LMS Integration Example

```
python
CopyEdit
@app.get("/lms-content/{course_id}")
def get_lms_content(course_id: str):
    # Mock LMS content fetch
    return {
        "course_id": course_id,
        "modules": ["Intro", "Module 1", "Module 2"]
    }
```

✅ 5. **requirements.txt**

```
css
CopyEdit
fastapi
uvicorn
pydantic
```

```
aiohttp
python-jose[cryptography]
```

✅ How to Run

1. **Backend**:

   ```
   bash
   CopyEdit
   uvicorn main:app --reload
   ```

2. **Frontend**:

   ```
   bash
   CopyEdit
   npm install
   npm start
   ```

🚀 Next Steps You Can Add

- Google OAuth login (`react-google-login`)
- Real OpenAI integration (use `openai` Python SDK)
- PostgreSQL with `SQLAlchemy`
- Dockerize the app for deployment