**Project Plan: AI-Powered Course Module Generator**

**1. Overview**

This project develops an AI chatbot that generates **structured course modules** based on **teacher-provided resources**(PDFs, articles, web links) and **example formats**. The AI will **automate lesson creation**, refine content based on teacher feedback, and dynamically **update a website** with new modules.

**2. Timeline (8 Weeks)**

| **Phase** | **Task** | **Duration** | **Deliverable** |
| --- | --- | --- | --- |
| **Week 1** | Model Selection & Setup | 1 week | LLM (Vicuna/Llama 2) running locally |
| **Week 2** | PDF & Web Extraction | 1 week | Text extraction pipeline (PyMuPDF, BeautifulSoup) |
| **Week 3-4** | AI-Powered Module Generation | 2 weeks | API for structured lesson generation |
| **Week 5** | Website Development | 1 week | Website prototype (React/Next.js) |
| **Week 6** | Integration & Auto-Updates | 1 week | AI-generated content deployment to website |
| **Week 7** | Feedback & Refinement | 1 week | Teacher feedback system |
| **Week 8** | Testing & Final Deployment | 1 week | Fully functional MVP |

**3. Core Features**

**(a) Resource Processing**

✔️ Extract text from **PDFs, articles, web pages**.  
✔️ Use **PyMuPDF or BeautifulSoup for text extraction**

**(b) AI Module Generation**

✔️ Generate structured lessons (Topic, Examples, Q&A).  
✔️ Use **Vicuna/Mistral** for generation via **llama.cpp or API**.

**(c) Website Integration**

✔️ Deploy modules automatically.  
✔️ Allow teacher feedback & updates.

**4. Deployment Strategy**

* **Local Prototype (Weeks 1-5)**: Run models locally using **llama.cpp**.
* **Cloud Deployment (Weeks 6-8)**:
  + API: **FastAPI or Flask**
  + Hosting: **Vercel, Netlify, or AWS EC2**
  + Storage: **Firebase/PostgreSQL**

**5. Next Steps**

1️⃣ Set up **Vicuna/Mistral model** with llama.cpp.  
2️⃣ Build **text extraction pipeline** (PyMuPDF, BeautifulSoup).  
3️⃣ Develop **AI-generated module API**.  
4️⃣ Deploy **website & feedback system**.

**5. Flowchart**

