**📦GitHub Repo with Demo API**

Here is a suggestion for how to set up a GitHub repository with a demo API, frontend, and ready-to-use prompt integrations for the project **“Reimagine Workplace Learning With AI”**:

**📁 Example GitHub Structure**

**/**

**├── backend/**

**│ ├── app.py # FastAPI or Flask with LangChain chains**

**│ ├── prompts.py # Stored prompts for each module**

**│ ├── llm.py # OpenAI GPT-4 (or Claude) configuration**

**│ └── vector\_store.py # Connection to Couchbase Capella**

**├── frontend/**

**│ ├── src/**

**│ │ ├── App.jsx # React app with Chat UI and module display**

**│ │ └── Simulator.jsx # Component for interactive scenario simulation**

**│ └── package.json**

**├── deployment/**

**│ ├── Dockerfile # Container for backend**

**│ └── cloudrun.yaml # Deployment config for Google Cloud Run**

**└── README.md**

**⚙️ Backend Demo in Python (FastAPI + LangChain)**

* Includes prompt handling, LLM setup, and Couchbase vector search integration
* API endpoints for generating module content, simulating training, and providing recommendations

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

Et bilde som inneholder tekst, Font, skjermbilde

KI-generert innhold kan være feil.

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

**💻 Frontend Demo in React (Chat UI + Simulator)**

* Interactive chatbot interface for user dialogue and micro-lesson interaction
* A separate simulator component for scenario-based training modules
* Responsive design using modern web components

Et bilde som inneholder tekst, skjermbilde, Font

KI-generert innhold kan være feil.

Et bilde som inneholder tekst, skjermbilde, Font, programvare

KI-generert innhold kan være feil.

**📝 Prompts Defined in prompts.py**

1. Concept Generator – Three AI-based ideas for adaptive learning, simulation, and recommendations
2. Micro-Lesson – Five-minute learning module with quiz
3. Conversation Simulation – Interactive customer dialogue with AI feedback
4. Recommendation Analysis – RAG-powered module suggestions and content analysis  
   (*Edit the PROMPTS file to include your custom use cases.*)

**🧩 Integration & Next Steps**

* **Vector Store:** Use Couchbase Capella + LangChain integration for embedding storage and retrieval, suitable for RAG-based recommendations ([docs.couchbase.com](https://docs.couchbase.com), [langchain.com](https://python.langchain.com), etc.)
* **RAG Pipeline**: Follow guides for hybrid search and Retrieval-Augmented Generation (RAG) implementations using Couchbase
* **CI/CD & Hosting:** Package backend with Docker and deploy via Google Cloud Run or Vercel
* **Frontend:** Build a chatbot UI for interacting with the backend APIs, and add a visual simulator for training scenarios

**Prepared by:** Ignacio Tejera  
**Date:** June 2025