**📦 GitHub Repo med demo-API**

Her kommer et forslag til hvordan du setter opp et GitHub-repo med demo-API, front-end og ferdige prompt‑koblinger for **“Reimagine Workplace Learning With AI”**:

**📁 Eksempel på GitHub‑struktur**

**/**

**├── 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 i Python (FastAPI + LangChain)**

- Inkluderer prompt-håndtering, LLM-oppsett og Couchbase-vektor søk integrasjon

- API-endepunkter for å generere modulinnhold, simulere opplæring og gi anbefalingerEt 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 i React (Chat UI + Simulator)**

* Interaktivt chatbot-grensesnitt for brukerdialog og interaksjon med mikroleksjoner
* En egen simulatorkomponent for scenariobaserte opplæringsmoduler
* Responsiv design med moderne webkomponenter

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.

**📝 Prompter definert i prompts.py**

1. **Konsept‑generator** – 3 AI‑konsepter for adaptiv opplæring, simulering og anbefaling
2. **Mikroleksjon** – 5‑min læringsmodul med quiz
3. **Samtalesimulering** – interaktiv dialog med tilbakemelding
4. **Anbefalings‑analyse** – RAG‑drevet oppsummering og modulforslag

(Redigerteksten i PROMPTS‑filen til å inkludere dine egne ønsker.)

**🧩 Integrasjon og neste steg**

* **Vector store**: Bruk Couchbase Capella + LangChain‑integrasjon for embeddings og RAG-anbefaling ([docs.couchbase.com](https://docs.couchbase.com), [langchain.com](https://python.langchain.com), etc.)
* **RAG‑pipeline**: Følg guidene for hybride søk og RAG [couchbase.com](https://www.couchbase.com/blog/rag-applications-with-vector-search-and-couchbase/?utm_source=chatgpt.com).
* **CI/CD & hosting**: Pakk backend i Docker og deploy via Cloud Run eller Vercel.
* **Frontend**: Lag en chatbot‑UI for interaksjon med API‑endepunktene, samt scenario‑simulator.

**🎯 Videre**

1) Lage det faktiske GitHub‑repoet med filstruktur og README

2) Eksempelkode for simulering og quiz‑logikk

3) Deployment‑script og miljø­­variabler

4) Ekstra funksjoner: caching med Couchbase, autentisering, dokument opplasting, videosimulering

Ignacio Tejera  
June 2025