Personalized Education Platform Powered by AI – Documentation

1. Project Overview

This project delivers an AI-powered adaptive learning platform using Streamlit for the frontend, Supabase for user authentication and data management, and advanced NLP techniques for intelligent chatbot functionality. The chatbot provides educational support based on a udemy course dataset and RAG (Retrieval-Augmented Generation) pipeline. All components are designed to be open-source, modular, and user-centric.

---

2. Project Objectives

- Build a chatbot capable of adaptive learning and educational recommendations.
- Authenticate and manage users securely with Supabase.
- Prepare a vectorized course dataset with enriched semantics.
- Implement LLM-powered agents with LangChain and CrewAI.
- Create an interactive frontend using Streamlit.
- Store user learning data and chatbot interactions in a structured format.

---

3. Functional Modules

3.1 User Authentication & Data Management

- **Library**: Supabase (Python SDK)
- **Features**:
  - Sign up / sign in via email/password
  - (Bonus) OAuth2 integration with GitHub
  - Store user profiles, user courses, passive logs, learning progress, learning history, chat interactions.
- **Implemented Files**:
  - **auth/auth_utils.py:** This file contains utility functions for authentication and user management using supabase.
  - **auth/oauth.py:** This file contains the oauth authentication logic for GitHub using supabase

3.2 Data Preparation & Vectorization

- **Steps**:
  - Cleaning (categorize URL, extract Valid URLs, drop duplicates)
  - Enrichment (lemmatizing, tokenizing, NER, semantic tagging using spacy)
  - Embedding generation (all-MiniLM-L6-v2/ BERT based)
  - Indexing with Qdrant (using singleton client)
- **Implemented Files**:
  - **data_prepration/clean_data.py:** cleaning step

- **data_prepration/enrich_data.py:** enrichment and embedding generation steps
- **data_prepration/qdrant.py:** Indexing with Qdrant step
- **pipeline.py:** This file handles the pipeline for data preparation, embedding generation and uploading to Qdrant.

3.3 AI Core – RAG + Agents

- **rag_pipeline.py**
  - Retrieves relevant vectorized courses embeddings.
  - Feeds context to LLM and returns response.
- **langchain_flow.py**
  - LangChain flows for Q&A, course recommendation, career coaching.
  - extracting course name/topic from query.
- **agents.py**
  - Intent classification agent from LangChain flows.
  - Context filtering agent.
- **main-ai.py**
  - Testing the AI pipeline and its components.

3.4 Streamlit Frontend

- **Pages**:
  - **home.py:** Welcome page for the stream-lit app.
  - **login.py:** for both email/password and GitHub.
  - **signup.py:** sign up new users.
  - **chatbot.py:** Main chatbot interface
  - **add_course.py**: add courses to the user's profile.
  - **View_courses.py:** view all courses with their information.
  - **Navigate_course_material.py:** navigate courses' quizzes and videos.
- **App.py:** main entry for streamlit app.

3.5 Monitoring

- **Logging User/Agent Interactions**
  - **monitoring/agentops_logger.py:** track session start/end, prompt, response

---

4. Key Features

- Secure authentication via Supabase.
- Real-time AI chatbot with course knowledge.
- Retrieval-Augmented Generation using Qdrant.
- Modular, scalable code structure.
- Logging for usage tracking (AgentOps-ready).

---

5. Configuration & Setup

- **.env file**:
  - SUPABASE_URL=project_url
  - SUPABASE_KEY=_anon_or_service_role_key
  - QDRANT_URL=qdrant_instance
  - GOOGLE_API_KEY=gemini_api_key
- **Run the App**:
  - streamlit run app.py

---

6. Outstanding Work

- Backend (FastAPI) is not implemented in this version.
- Deployment on cloud not implemented in this version.

---

7. Version Control

- GitHub Repository: [Repo Link]
- Branch Strategy: *main, *master
- Commit Convention: Conventional Commits

---

10. Final Deliverables

- Functional Streamlit chatbot UI.
- AI agents powered by RAG.
- Supabase-based user management.
- Vectorized and enriched course dataset.
- Codebase hosted on GitHub.
- Documented architecture and modules.
- Reusable modular pipeline for LLM apps.

11.Future Work

As the foundation of the Personalized Education Platform has been established, several enhancements and extensions can be pursued in future iterations to enrich the system's functionality, scalability, and user experience:

- **Backend Development & API Integration**
  - Fully implement the FastAPI backend to provide structured endpoints for user, course, and chatbot interactions.
  - Ensure seamless integration between Streamlit frontend and backend logic via RESTful APIs.
  - Add automated testing and error-handling mechanisms for robustness.
- **Deployment & DevOps**
  - Deploy both frontend and backend using containerized services
  - Explore cloud deployment options (e.g., Render, Railway, Hugging Face Spaces) for scalable access.
  - Implement CI/CD pipelines to automate testing and deployment workflows via GitHub Actions.
- **Advanced LLM Features**
  - Integrate additional LLM APIs such as OpenAI's GPT-4 or Anthropic's Claude for diverse response styles.
  - Enable multi-turn memory and deeper personalization using persistent conversation history.
  - Allow real-time fine-tuning or in-context learning for personalized tutoring sessions.
- **Enhanced Course Intelligence**
  - Add course progress tracking analytics and learning recommendations based on behavior.
- **Security & Privacy**
  - Encrypt sensitive user data at rest and in transit.
  - Implement OAuth2-based authentication and session expiration.
  - Add role-based access control (RBAC) for students, instructors, and admins.
- **Knowledge File Uploader**
  - Finalize the feature that allows users to upload PDFs, CSVs, or TXT files to expand chatbot knowledge dynamically during a session.
- **Admin Dashboard**
  - Build an admin panel to manage users, monitor chatbot usage, upload course materials, and generate engagement reports.
- **Monitoring & AgentOps**
  - Enhance AgentOps for real-time monitoring of chatbot performance and failure tracing.