VeidaLabs Software Developer Hiring Assignment

Project: Learn with Jiji - The AI Learning Companion

Goal: Assess hands-on server-side development skills, including Supabase usage (DB, Auth, Storage), API design, data handling, code quality, and basic security awareness, in a product-led AI application context.

Duration & Deadline: 2 Hours; Feb 9, end of day

Submission**:** Need all the details together in the below format in an email

1. **Name:** Add Name
2. **Subject**: VL Software Developer Hiring Assignment
3. **LinkedIn Profile:** Add link
4. **GitHub Profile:** Add link
5. **Resume:** Attach or link
6. **Assignment GitHub Repo + Short Readme:** Add link
7. **Supabase schema / SQL (or migration file):** Attach or link
8. **Working Demo Video:** Attach or link

Email: hello@veidalabs.com

Context

VeidaLabs is building Learn with Jiji, an AI-driven learning companion that personalizes how professionals, founders, young adults, and teams learn about AI.

VeidaLabs serves: Corporates and Universities and plans to serve B2C users.

Learning content can include text, presentations (PPTs), images, session-recorded videos, and AI-generated avatar videos.

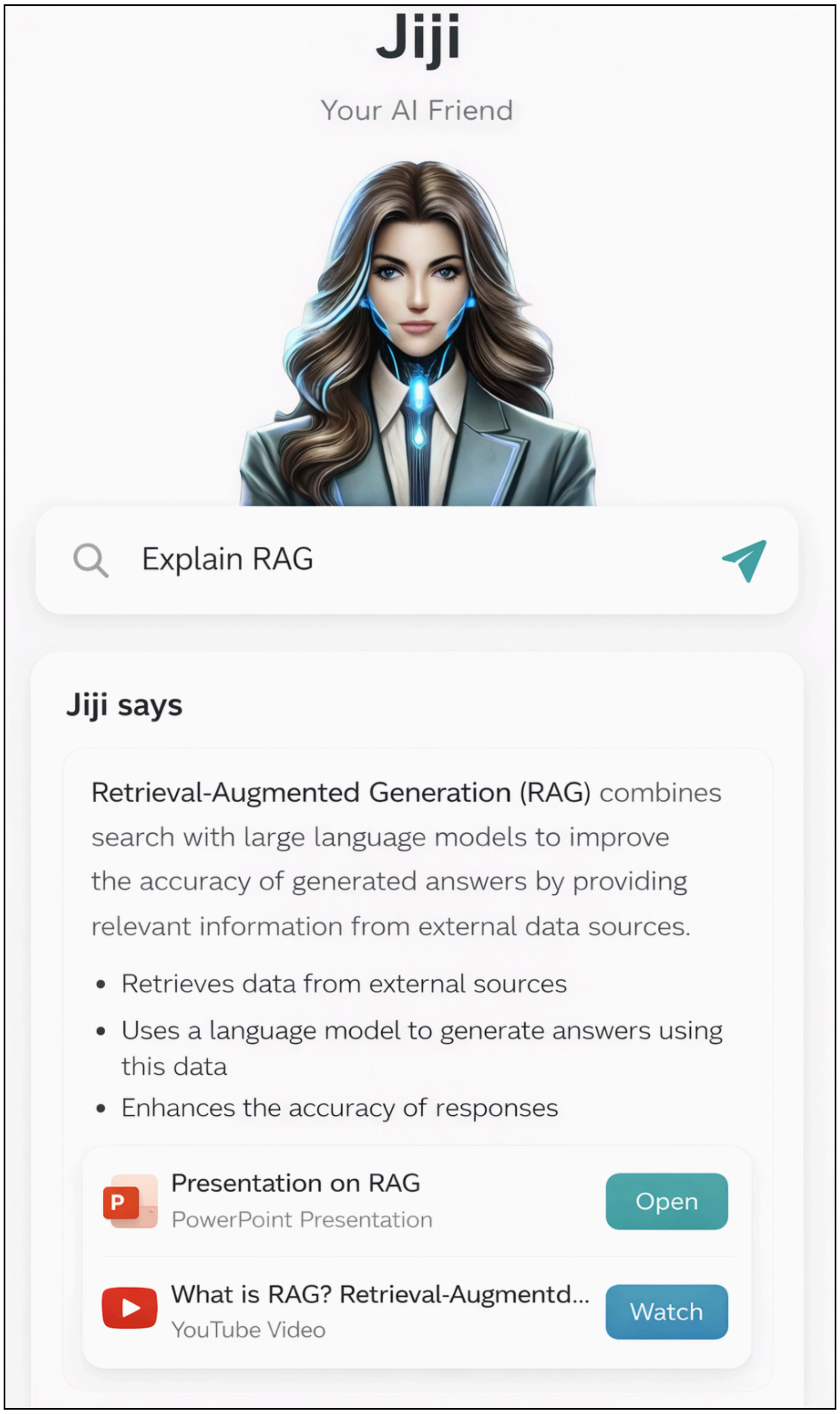
Internet content is out of scope for this assignment.

The frontend (Flutter / Web) consumes backend APIs that:

- Accept user queries
- Fetch relevant learning content
- Return structured responses (text + resources)

This assignment focuses only on backend implementation.

Here is a sample design to relate to how the app looks.



Jiji

Your AI Friend

Explain RAG

Jiji says

Retrieval-Augmented Generation (RAG) combines search with large language models to improve the accuracy of generated answers by providing relevant information from external data sources.

- Retrieves data from external sources
- Uses a language model to generate answers using this data
- Enhances the accuracy of responses

Presentation on RAG
PowerPoint Presentation
Open

What is RAG? Retrieval-Augmentd...
YouTube Video
Watch

VeidaLabs
Make AI your Friend

Your 2-Hour Challenge

Assignment

Build a backend service that powers Jiji’s *search & respond* flow.

The backend should:

1. Accept a user query
2. Retrieve relevant learning content
3. Return a structured response consumable by the frontend

No real AI integration is required. Mocked responses are acceptable.

One Flow – Creation & Delivery

API Flow

1. User sends a query (e.g., *“Explain RAG”*)
2. Backend validates request
3. Backend fetches matching resources from Supabase
4. Backend responds with:
    - Answer text
    - Resource links (PPT + Video)

Technical Expectations

Backend & Supabase

- Use Supabase for:
    - Database
    - Auth (simple / mocked is fine)
    - Storage (for PPT / Video links)

API

- One endpoint (example):
    - `POST /ask-jiji`
- Clean request & response contracts
- Proper error handling

Security

- Implement **Row Level Security (RLS)** in Supabase
- No secrets in code
- Basic input validation

Data (Simple & Realistic)

Minimum suggested tables:

- `profiles`
- `queries`
- `resources` (ppt / video)

Storage:

- 1 sample PPT file
- 1 sample video file (or placeholder)

Deliverables

Please include:

1. **Backend code** (repo)
2. **Supabase schema / SQL**
3. **README** covering:
    - How to run
    - API endpoint(s)
    - How auth & RLS work
    - One improvement you'd make with more time

Role Evaluation Criteria

- Backend fundamentals
- Supabase understanding (DB, Auth, RLS, Storage)
- API design clarity
- Code structure & cleanliness
- Basic security awareness