Interview Assignment: AI- Powered Shopify Analytics App

Objective

Design and build a **mini AI- powered analytics application** that connects to a Shopify store, reads customer/order/inventory data, and allows a user to ask **natural- language questions** (e.g., inventory projection, sales trends). The system should translate these questions into **ShopifyQL**, fetch data from Shopify, and return answers in **simple, layman- friendly language**.

This assignment evaluates: - System design & API architecture - Rails API design - Python + LLM orchestration - Agentic workflows - Shopify API & ShopifyQL understanding - Practical problem- solving

---

Problem Statement

Build an application with the following capabilities:

1. **Connect to a Shopify Store**
    - Authenticate with Shopify using OAuth
    - Read store data (orders, customers, products, inventory)
2. **Rails API (Backend Gateway)**
    - Expose APIs that accept:
        - User questions in natural language
        - Store context (store_id / shop name)
    - Forward requests to a Python AI service
    - Handle authentication, validation, and response formatting
3. **Python AI Service (LLM- Powered Agent)**
    - Accepts a question from Rails
    - Uses an LLM to:
        - Understand user intent
        - Decide **what Shopify data is needed**
        - Generate the appropriate **ShopifyQL query**
    - Execute the query against Shopify APIs
    - Post- process results
    - Convert output into **simple, business- friendly language**
4. **Agentic Workflow**
    - Implement an agent that:
        - Interprets the question
        - Chooses the right data source (orders, inventory, customers)

- Builds ShopifyQL
- Validates query correctness
- Explains results clearly

Example User Questions

- “How many units of Product X will I need next month?”
- “Which products are likely to go out of stock in 7 days?”
- “What were my top 5 selling products last week?”
- “How much inventory should I reorder based on last 30 days sales?”
- “Which customers placed repeat orders in the last 90 days?”

Functional Requirements

1. Shopify Integration

- OAuth- based authentication
- Ability to query:
  - Orders
  - Products
  - Inventory levels
- Use **ShopifyQL** for analytics queries

2. Rails API

Create a Rails- only API application with endpoints such as:

`POST /api/v1/questions`

Request Body:

```
{
  "store_id": "example-store.myshopify.com",
  "question": "How much inventory should I reorder for next week?"
}
```

Responsibilities: - Validate input - Store request logs (optional bonus) - Forward question to Python service - Return formatted response

---

3. Python AI Service

Responsibilities: - Receive question + store context - Use LLM to: - Classify intent (inventory, sales, customers) - Generate ShopifyQL - Handle ambiguous questions gracefully - Call Shopify API with generated query - Convert raw data into insights - Return human- readable explanation

Sample Output:

```
{
  "answer": "Based on the last 30 days, you sell around 10 units per day. You
should reorder at least 70 units to avoid stockouts next week.",
  "confidence": "medium"
}
```

Agent Design Expectations

The agent should:

1. **Understand intent**
    - Identify metrics (sales, inventory, time period)
2. **Plan**
    - Decide which Shopify tables/fields are required
3. **Generate ShopifyQL**
    - Ensure syntactically correct queries
4. **Execute & Validate**
    - Handle empty or partial data
5. **Explain Results**
    - Convert technical metrics into business language

Non- Functional Requirements

- Clean API design
- Proper error handling
- Clear separation of concerns (Rails vs Python)
- Reasonable prompt design for LLM
- Secure handling of Shopify tokens

Tech Stack (Suggested)

- **Backend API:** Ruby on Rails (API- only)
- **AI Service:** Python (FastAPI / Flask)
- **LLM:** OpenAI / Claude / Gemini (mock allowed)
- **Database:** PostgreSQL (optional)
- **Auth:** Shopify OAuth

Bonus (Optional)

- Caching Shopify responses

- Conversation memory for follow- up questions
- Query validation layer for ShopifyQL
- Metrics dashboard
- Retry & fallback logic in agent

Deliverables

1. GitHub repository (or zipped project)
2. README with:
    - Setup instructions
    - Architecture explanation
    - Agent flow description
3. Sample API requests & responses
4. (Optional) Architecture diagram

Evaluation Criteria

- Correctness of Shopify integration
- Quality of API design
- Agent reasoning clarity
- Practical handling of real- world data issues
- Code readability and structure
- Ability to explain results simply

Time Expectation

 48 hours(depth over completeness)

Focus on **design clarity and reasoning**, not production polish.

Good luck!