Ebuddy AI Intern Take-Home Assignment

Background

Ebuddy is building an AI travel assistant that helps users **search, book, and manage** business trips while following company travel policies.

This assignment checks your understanding of:

- Retrieval-Augmented Generation (RAG)
- Using LLMs to trigger actions or “agents” from user language
- Basic prompt engineering and personalization

You don’t need a full production system — focus on **clear thinking, simple code, and readable explanations.**

What to Submit

- A short **README.md** (1–2 pages): explain your approach, main ideas, and how to run your demo.
- Code that runs locally (any language is fine; Python preferred).
- Example input/output to show your idea works.
- Optional: one short demo video or notebook.

Problem 1: Build a Small RAG Demo (≈40 points)

Goal: Show that you understand how a Retrieval-Augmented Generation system works.

Tasks

1. Create a small dataset (5–10 short texts) about business travel — for example:
    - Company travel policy
    - Airline ticket change rules
    - Hotel cancellation policy
2. Choose a simple way to store and search these texts (for example: list of documents + embeddings using `sentence-transformers` or `faiss`).
3. Write a small script or notebook where:
    - The user asks a question, e.g. *“Can I refund my ticket within 24 hours?”*
    - The program retrieves the most relevant text and combines it with the question to answer using an LLM (e.g. GPT or open-source model).
4. Show a few example questions and answers.
5. In your README, explain briefly:
    - How you retrieve documents
    - Why retrieval improves answer quality

Bonus (optional): add a field like “region” or “vendor” and show you can filter results (e.g. “Airline A only”).

Problem 2: From User Language to Action (≈30 points)

Goal: Make a simple agent trigger system that maps user requests to structured actions.

Tasks

1. Define **3–4 simple intents**, for example:
    - `SearchFlight`

- `BookHotel`
- `CancelFlight`
- `CheckPolicy`

2. For each intent, define what parameters are needed. Example:

```
{
  "name": "SearchFlight",
  "parameters": ["from_city", "to_city", "date"]
}
```

3. Write a small function or prompt so that, given a user sentence, the system can identify which intent to use and extract the parameters.
   Example input:

   "Book a flight from Shanghai to Tokyo next Monday."
   Example output:

   {"intent": "SearchFlight", "from_city": "Shanghai", "to_city": "Tokyo", "date": "next Monday"}

4. Show at least 3 example cases.

Bonus (optional): before confirming an action, check if it follows a "company policy" rule (e.g. "Economy class only").

Problem 3: Prompt Design & Personalization (≈30 points)

Goal: Think about how user preferences can change the AI's behavior.

Tasks

1. Design a simple **user profile** with a few fields, for example:

   {

```
  "home_city": "Shanghai",
  "preferred_airline": "Air China",
  "budget_limit": 2000,
  "language": "en"
}
```

2. Show how you would **include this profile in a prompt** when the model answers or performs an action.
   Example system prompt:

   "You are a travel assistant. Always prefer flights by {{preferred_airline}} and keep price under {{budget_limit}} RMB."

3. Compare two responses — one **with** and one **without** profile data — and explain the difference.

4. In your README, explain what kind of user data is helpful for personalization and how to use it safely (no sensitive data in prompts).

Bonus (optional): add one "guardrail" rule, e.g.
If the model is not sure or missing info, it should ask a clarifying question instead of guessing.

Recommended Structure

```
/project
  rag_demo.py / notebook.ipynb
  agent_demo.py
  prompts/
  data/
  README.md
```