

AI Agent Chat with Tools + DB (Library Desk Agent)

Goal

Build a local **chat interface** for a **Library Desk Agent** that can answer questions **and** perform actions via tools that read/write a database. No authentication and no deployment are required.

What to Build

1) Database (SQLite recommended; Postgres optional)

Create and seed these tables:

- **Domain Tables**

- `books(isbn PK, title, ...)`
- `customers(id PK, name, email, ...)`
- `orders(id PK, customer_id, ...)`
- `order_items(order_id, isbn, ...)`

- **Chat storage**

- `messages(id, session_id, role , ...)`
- `tool_calls(id, session_id, name, args_json, result_json, created_at)`

- Seed: **10** books, **6** customers, **4** orders.

2) Agent Tools (backend functions)

Implement these callable tools and ensure they change/read the DB correctly:

- `find_books({ q, by: "title" | "author" })` → list of matches
- `create_order({ customer_id, items: [{ isbn, qty }] })` → new `order_id` and **reduce stock**

- `restock_book({ isbn, qty })` → updated stock
- `update_price({ isbn, price })`
- `order_status({ order_id })` → summary
- `inventory_summary()` → low-stock titles

3) LLM

- Use any LLM (OpenAI/Ollama ..etc).

4) Chat UI

- **Left:** session selector (new/load). (Optional - preferred)
- **Center:** Chat conversation.

Sample Scenarios:

1. “We sold 3 copies of *Clean Code* to customer 2 today. Create the order and adjust stock.”
→ calls `create_order`, reduces stock, replies with order id & new stock.
2. “Restock *The Pragmatic Programmer* by 10 and list all books by Andrew Hunt.”
→ `restock_book` then `find_books`.
3. “What’s the status of order 3?” → `order_status`.

Deliverables

Git repo (public or private with read access) containing:

```
/app          (frontend)
/server       (agent, tools, REST endpoints)
/db           (schema.sql + seed.sql or seed script)
/prompts      (system prompt, ...)
.env.example  # OPENAI_KEY / OLLAMA toggle ..etc
```

README.md

- Provide simple run scripts.
- Include seed instructions.