**Bazar.com Design Documentation**

**System Architecture :-**

* **Frontend Service**: Routes requests to backend services.

Frontend Endpoints:-

1. “/search/<topic>” [ GET ] Returns books by topic
2. “/info/<item\_id>” [ GET ] Returns book details
3. “/purchase/<item\_id>” [ POST ] Purchases a book

* **Catalog Service**: Manages book data using SQLite.

Catalog Endpoints:-

1. “/search/<topic>” [ GET ] Internal search by topic
2. “/info/<item\_id>” [ GET ] Internal book details
3. “/update/<item\_id>” [ POST ] Updates stock/price

* **Order Service**: Handles purchases and logs orders.

Order Endpoint:

1. “/purchase/<item\_id>” [ POST ] Processes purchase

**Design Tradeoffs:-**

* **SQLite over MySQL**: Chosen for simplicity and file-based persistence.
* **Flask Concurrency**: Relies on Werkzeug’s thread pool for concurrent requests.
* **Atomicity**: Uses SQL transactions for stock decrement.

**Issues:-**

* No retry mechanism for failed inter-service calls
* No authentication for update operations.

**How to Run:-**

1. Clone the repository

<https://github.com/MusabAbuBshara/Bazar>

1. Run: `docker-compose up --build` command on terminal
2. Test endpoints with `curl` (examples in `Output.docx` file).