Scalable Services: Assignment

Library Microservices Application

(Group 8)

Submitted by: Rucha Sandeep Vaidya (2024TM93058)

1. Group Details and Contributions

Member	Contribution
Rucha Sandeep Vaidya (2024TM93058)	Inventory Service, Docker Setup, API Testing
Prashant Raghunath Jagtap (2023TM93591)	Book Service, Docker Setup, API Testing
Praharsha Laxmi Kandragula ()	Borrow Service, Docker Setup, API Testing
Taheri Mehlam Ali Sagar	No Contribution
Roopak Kumar	No Contribution

2. Application Details

This application simulates a digital library system using a microservices-based architecture. The system is composed of four independently deployable services:

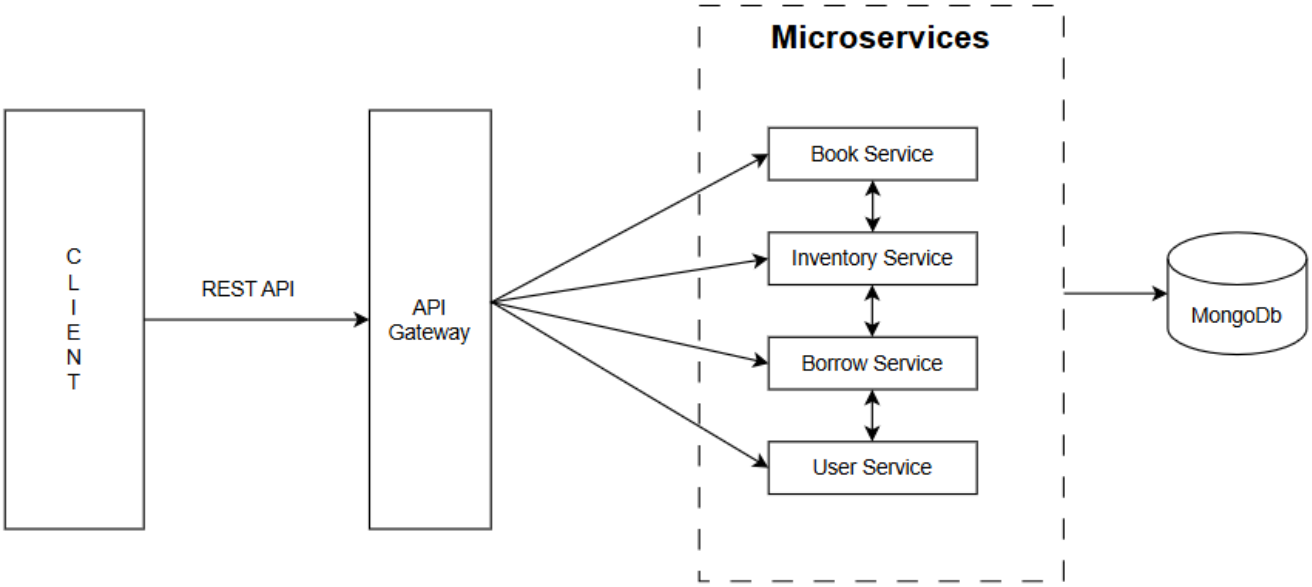
- Book Service: Manages book metadata.
- Borrow Service: Handles borrowing and returning of books.
- Inventory Service: Tracks available stock.
- User Service: Manages user profiles and data.

Each microservice is developed using Flask, uses MongoDB for data persistence, and is containerized using Docker. The app supports full CRUD operations and allows seamless testing via Postman or through the mobile UI.

Tech Stack

- **Frontend**: Flutter
- **Backend**: Python (Flask)
- **Database**: MongoDB (Dockerized)
- **Containerization**: Docker, Docker Compose
- **API Testing**: Postman
- **Architecture**: Microservices with separate Flask apps for each core module

Architecture Diagram



Microservices:

Service	Description	Port
book-service	Manages book catalog	5001
user-service	Manages members/users	5002
borrow-service	Handles book borrowing/returning	5003
inventory-service	Tracks stock count for each book	5004

Below Step – by – Step Execution focuses mainly on "**Inventory Service**" which I have worked on.

1. Local Setup:

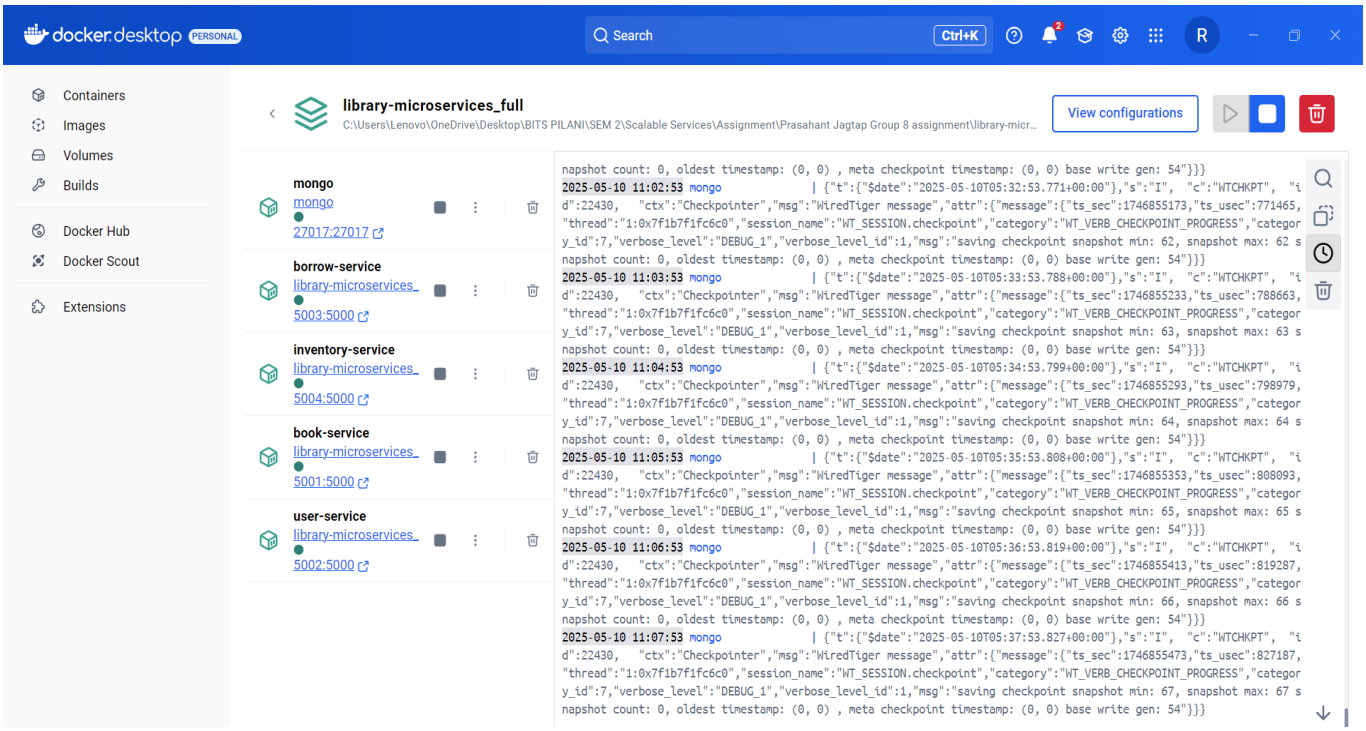
```
PS C:\Users\Lenovo\OneDrive\Desktop\BITS PILANI\SEM 2\Scalable Services\Assignment\Prasahant Jagtap Group 8 assignment\library-microservices-updated\library-microservices_full\
inventory-service> python -m venv venv
```

```
inventory-service> pip install flask pymongo
>>
Requirement already satisfied: flask in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (3.0.3)
Requirement already satisfied: pymongo in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (4.10.1)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (3.0.6)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (3.1.6)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.6.2 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (1.8.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from flask) (8.5.0)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from pymongo) (2.6.1)
Requirement already satisfied: colorama in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: zipp>=3.20 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from importlib-metadata>=3.6.0->flask) (3.20.2)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\lenovo\appdata\local\programs\python\python38\lib\site-packages (from Jinja2>=3.1.2->flask) (2.1.5)
```

```
PS C:\Users\Lenovo\OneDrive\Desktop\BITS PILANI\SEM 2\Scalable Services\Assignment\Prasahant Jagtap Group 8 assignment\library-microservices-updated\library-microservices_full\
inventory-service> docker compose up -d
time="2025-05-10T10:03:48+05:30" level=warning msg="C:\\Users\\Lenovo\\OneDrive\\Desktop\\BITS PILANI\\SEM 2\\Scalable Services\\Assignment\\Prasahant Jagtap Group 8 assignment
\\library-microservices-updated\\library-microservices_full\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential co
nfusion"
[+] Running 6/6
 ✓ Network library-microservices_full_backend  Created                                    0.2s
 ✓ Container mongo                             Started                                    0.7s
 ✓ Container user-service                      Started                                    1.4s
 ✓ Container inventory-service                 Started                                    1.4s
 ✓ Container borrow-service                    Started                                    1.4s
 ✓ Container book-service                      Started                                    1.1s
```

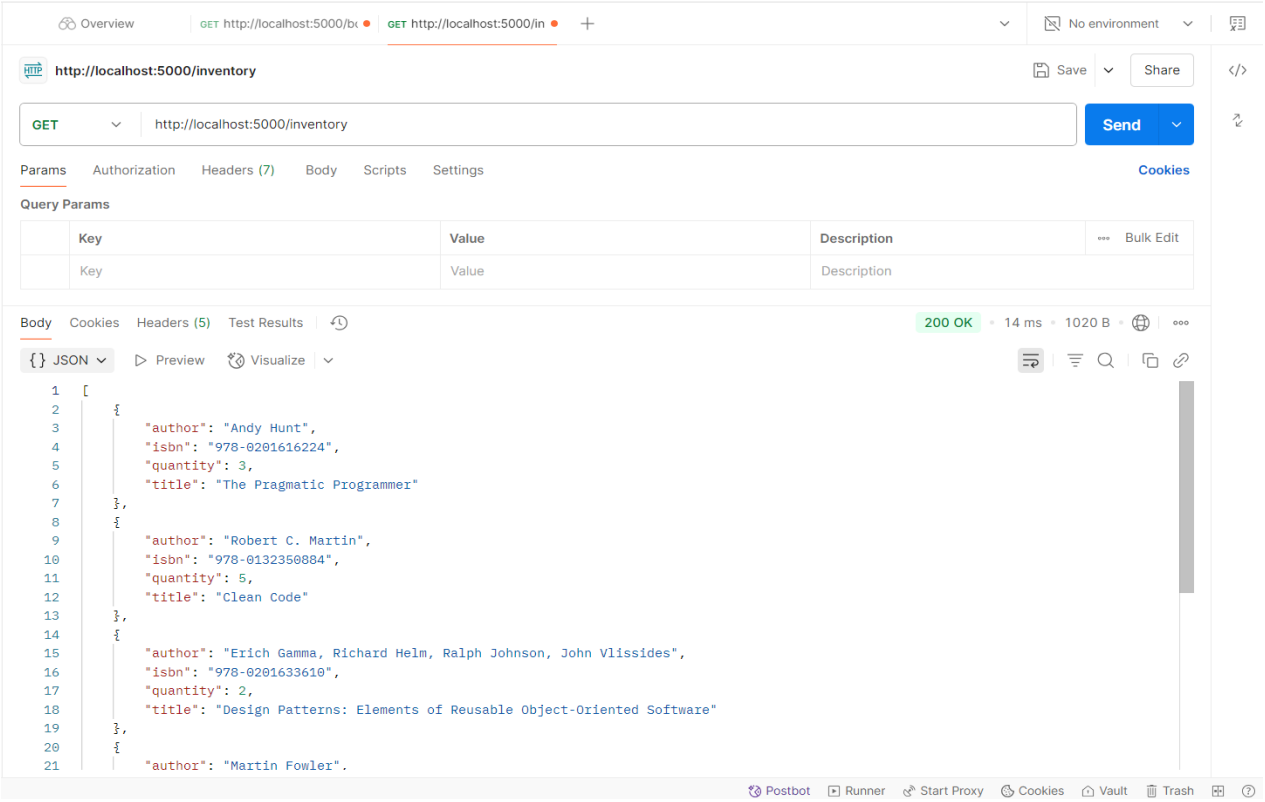
```
inventory-service> python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.31.198:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 444-303-048
127.0.0.1 - - [10/May/2025 10:04:33] "GET /inventory HTTP/1.1" 200 -
127.0.0.1 - - [10/May/2025 10:06:21] "POST /inventory HTTP/1.1" 201 -
127.0.0.1 - - [10/May/2025 10:07:42] "PATCH /inventory/9781234567890 HTTP/1.1" 200 -
127.0.0.1 - - [10/May/2025 10:56:35] "GET /inventory HTTP/1.1" 200 -
```

2. Docker



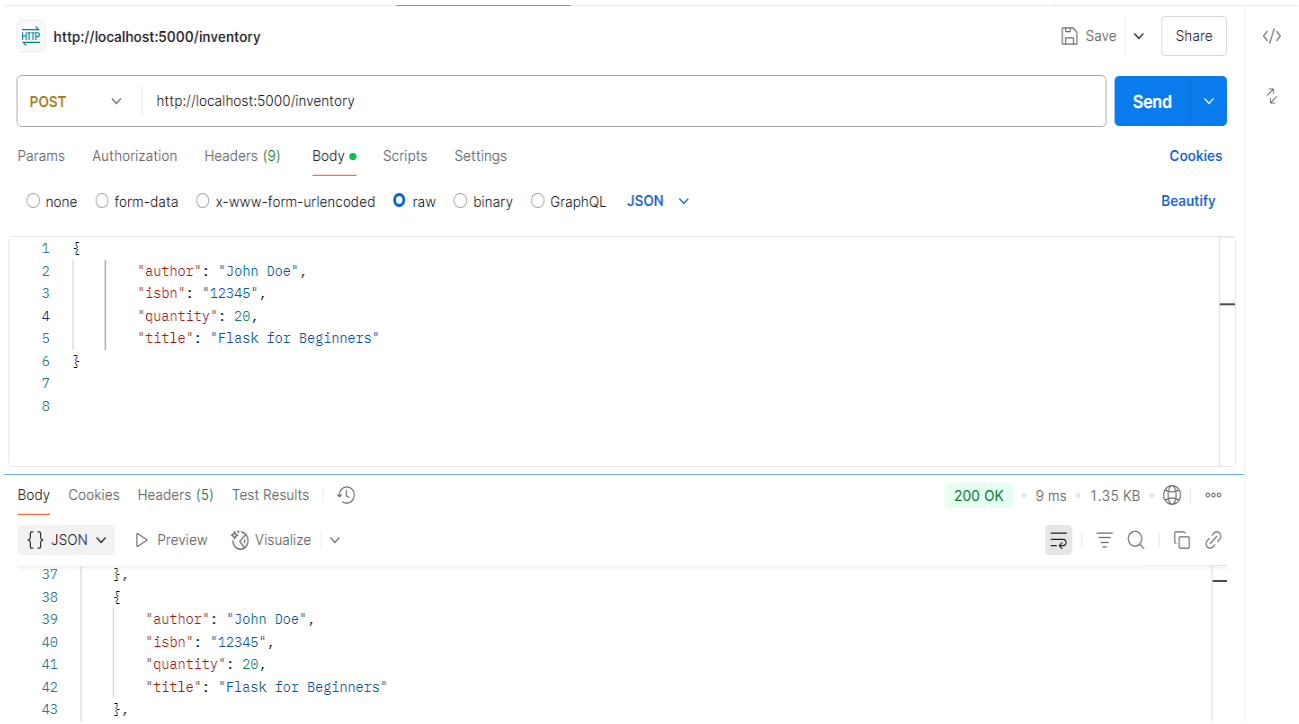
Inventory Service

3. Postman Testing
   - GET: http://localhost:5000/inventory



   - POST: http://localhost:5000/inventory

```
{
    "author": "John Doe",
    "isbn": "12345",
    "quantity": 20,
    "title": "Flask for Beginners"
}
```

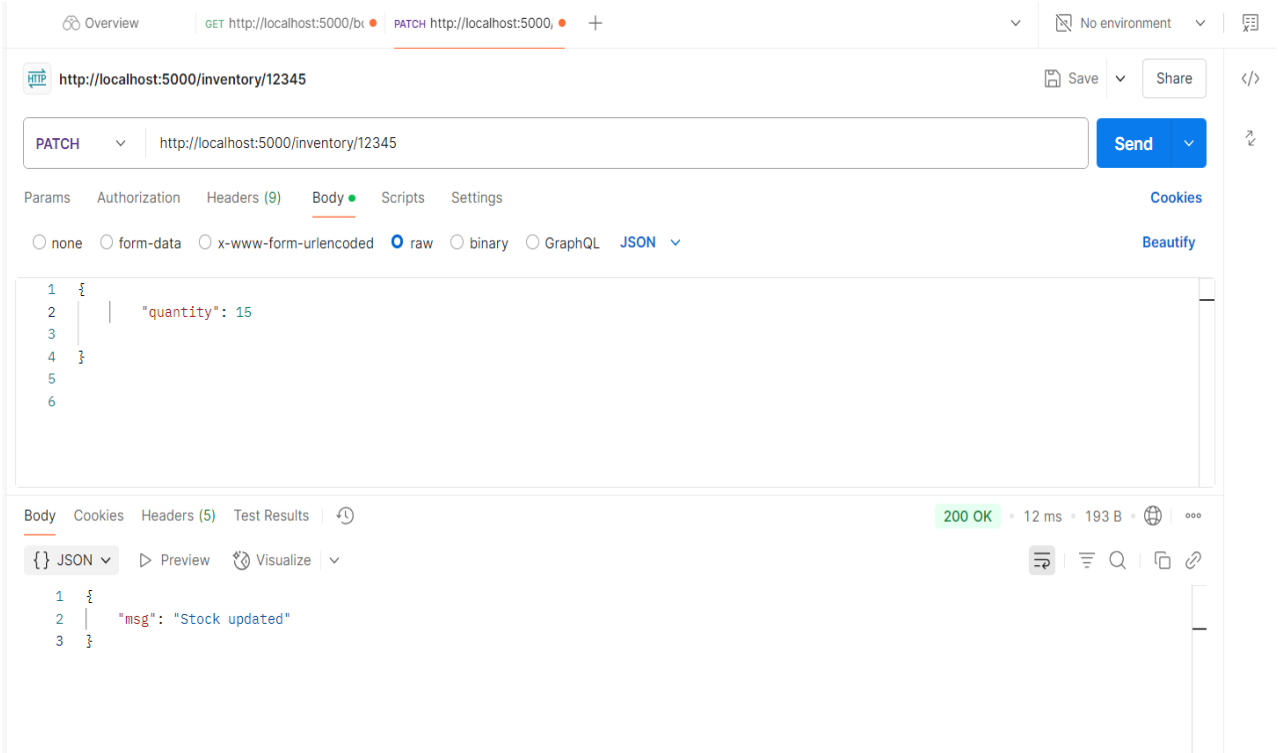


- PATCH: http://localhost:5000/inventory/9781234567890

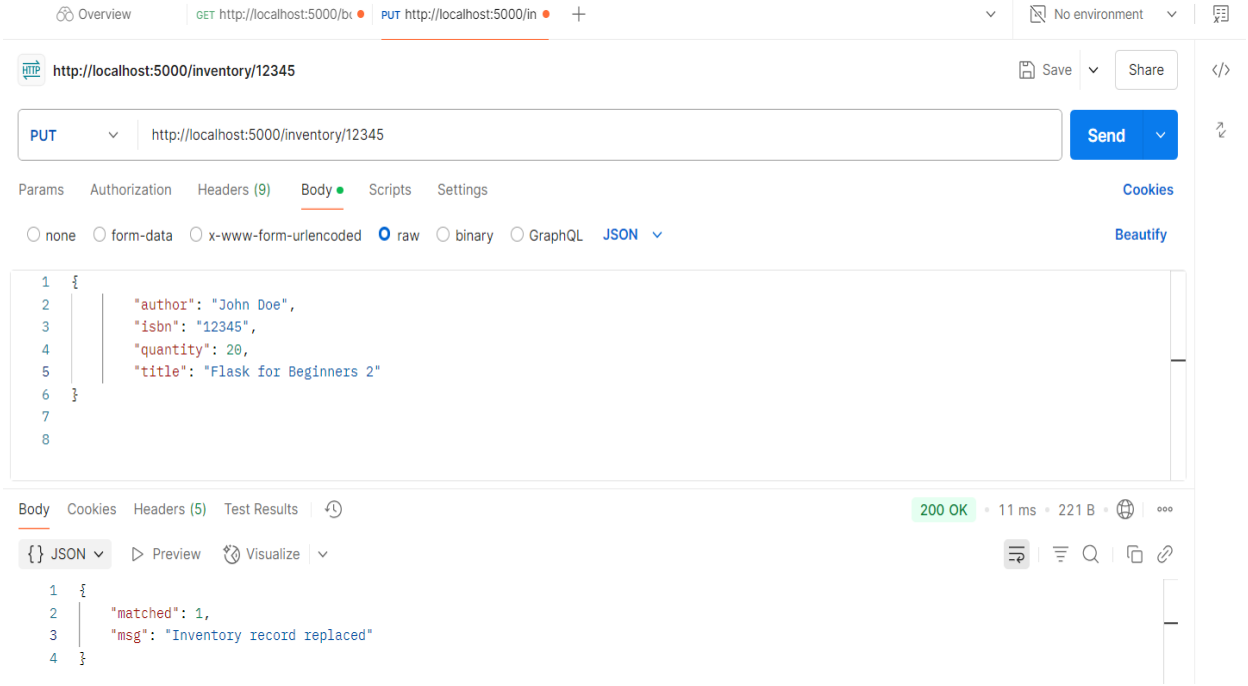
  {

  "quantity": 15

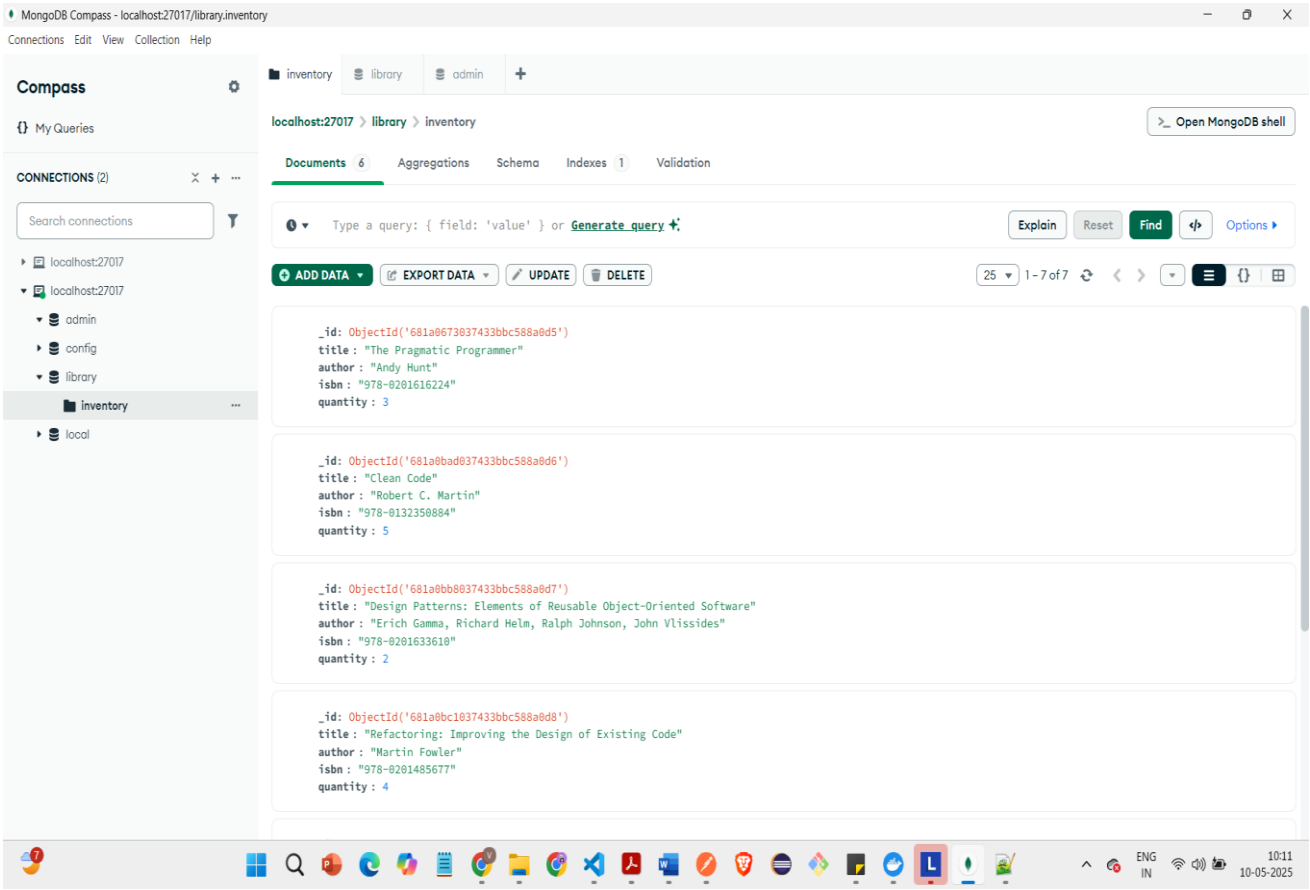
  }



- PUT: http://localhost:5000/inventory/12345



4. MONGODB



5. Browser: http://localhost:5000/inventory

localhost:5000/inventory

Pretty-print


```
[
  {
    "author": "Andy Hunt",
    "isbn": "978-0201616224",
    "quantity": 3,
    "title": "The Pragmatic Programmer"
  },
  {
    "author": "Robert C. Martin",
    "isbn": "978-0132350884",
    "quantity": 5,
    "title": "Clean Code"
  },
  {
    "author": "Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides",
    "isbn": "978-0201633610",
    "quantity": 2,
    "title": "Design Patterns: Elements of Reusable Object-Oriented Software"
  },
  {
    "author": "Martin Fowler",
    "isbn": "978-0201485677",
    "quantity": 4,
    "title": "Refactoring: Improving the Design of Existing Code"
  },
  {
    "author": "Kyle Simpson",
    "isbn": "978-1491904244",
    "quantity": 7,
    "title": "You Don't Know JS"
  },
  {
    "author": "Robert C. Martin",
    "isbn": "978-0137081073",
    "quantity": 3,
    "title": "The Clean Coder"
  },
  {
    "isbn": "9781234567890",
    "stock": 15,
    "title": "Sample Book"
  }
]
```


6. Example Commands Execution on Terminal

i. To list all containers: docker ps

```
PS C:\Users\Lenovo\OneDrive\Desktop\BITS PILANI\SEM 2\Scalable Services\Assignment\Prasahant Jagtap Group 8 assignment\library-microservices-updated\library-microservices_full\
inventory-service> docker ps
CONTAINER ID   IMAGE                                    COMMAND                  CREATED          STATUS          PORTS                    NAMES
713200128d79   library-microservices_full-user-service  "python app.py"          About an hour ago  Up About an hour  0.0.0.0:5002->5000/tcp   user-service
4fff34bad7b0   library-microservices_full-book-service  "python app.py"          About an hour ago  Up About an hour  0.0.0.0:5001->5000/tcp   book-service
4d8f498bab01   library-microservices_full-inventory-service "python app.py"      About an hour ago  Up About an hour  0.0.0.0:5004->5000/tcp   inventory-service
222debe79145   library-microservices_full-borrow-service "python app.py"          About an hour ago  Up About an hour  0.0.0.0:5003->5000/tcp   borrow-service
e62a5521ab5a   mongo                                    "docker-entrypoint.s…"   About an hour ago  Up About an hour  0.0.0.0:27017->27017/tcp mongo
```

ii. To enter into a container and execute commands: docker exec -it <container-id> bash

```
inventory-service> docker exec -it 4d8f498bab01 bash
root@4d8f498bab01:/app#
```

iii. Start a mongosh inside container: docker exec -it <container-id> mongosh

```
inventory-service> docker exec -it e62a5521ab5a mongosh
Current Mongosh Log ID: 681eedf68608b51a09d861df
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.0
Using MongoDB:          8.0.9
Using Mongosh:          2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-05-10T04:33:49.651+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2025-05-10T04:33:50.734+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2025-05-10T04:33:50.734+00:00: For customers running the current memory allocator, we suggest changing the contents of the following sysfsFile
   2025-05-10T04:33:50.734+00:00: We suggest setting the contents of sysfsFile to 0.
   2025-05-10T04:33:50.734+00:00: vm.max_map_count is too low
   2025-05-10T04:33:50.734+00:00: We suggest setting swappiness to 0 or 1, as swapping can cause performance problems.
------

test>
```

iv. Switch to database, eg. Library: use library

```
test> use library
switched to db library
```

v. Show collections

```
library> show collections
inventory
```

vi. db.inventory.find().pretty()

```
library> db.inventory.find().pretty()
...
[
  {
    _id: ObjectId('681ef08cdf2abe6cc0d861e0'),
    isbn: '1234567890',
    stock: 5
  }
]
```

7. Minikube Deployment

```
PS C:\Users\Lenovo> minikube start
😄 minikube v1.35.0 on Microsoft Windows 11 Home Single Language 10.0.22631.5189 Build 22631.5189
✨ Automatically selected the docker driver
📌 Using Docker Desktop driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚜 Pulling base image v0.0.46 ...
💾 Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 2.54 Mi
    > gcr.io/k8s-minikube/kicbase...:  500.31 MiB / 500.31 MiB  100.00% 1.94 Mi
🔥 Creating docker container (CPUs=2, Memory=4000MB) ...
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🐳 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗 Configuring bridge CNI (Container Networking Interface) ...
🔎 Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
💡 kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
🏄 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

```
inventory-service> kubectl apply -f inventory-deployment.yaml
deployment.apps/inventory-service created
service/inventory-service created
```