BAZAR.COM

PART 1

Sema Hodali 12029448

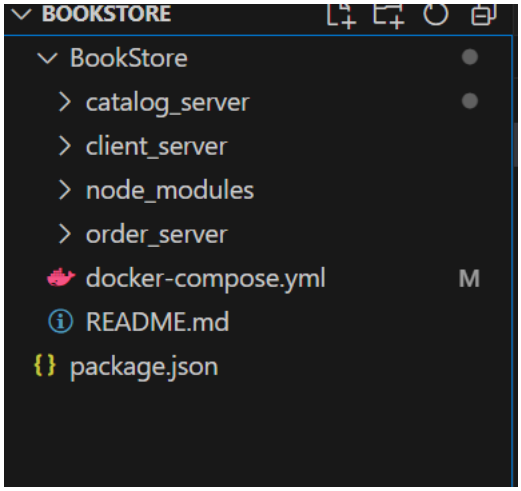
Leema Abu-Aladel 12029320

Introduction:

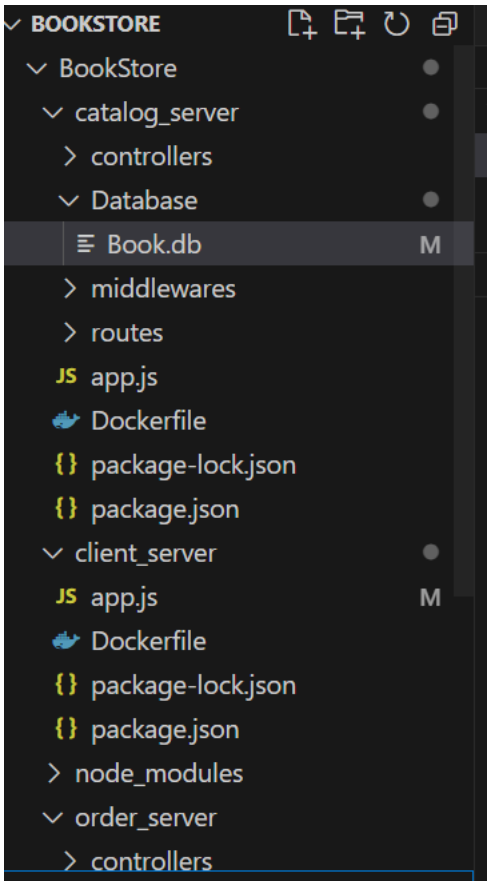
Bazar.com is online bookstore designed as two tier web microservices, backend implemented with node.js, database is sqlite3, backend has two microservices (catalog and order), docker and docker compose tool are utilize to deploy the microservices.

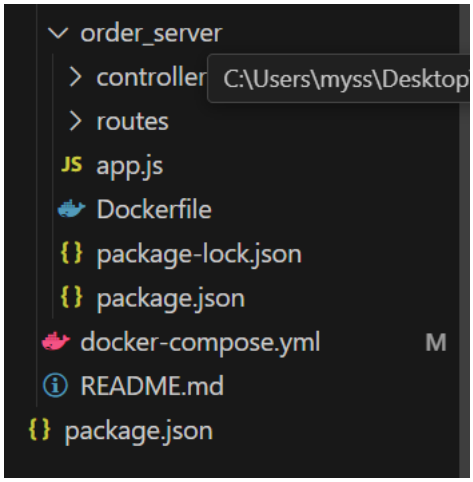
Project hierarchy:

Bazar book store consists of three services catalog, order and client(front end)



Each service has its own packages and docker file (as complete independent service)





Microservices:

Catalog service:

This microservice handle query operation like info and search operations, and update operation like update count operation.

-Info by id:

```
http://localhost:3000/CATALOG_WEBSERVICE_IP/info/id
```

-search by book topic:

```
http://localhost:3000/search/bookTopic
```

-update book count by id:

```
http://localhost:3000/updateCount /id
```

Order service:

This microservice support purchase operation which depend on catalog service (info and update count).

-Purchase by book id:

```
http://localhost:4000/purchase/id
```

Client service :

This microsevice represent the front end of the system which receive user request and call catalog and order operation according to request.

Running the project:

Cd the directory where docker-compose.yml file located

To build all containers → docker-compose up -d –build

To stop all containers → docker-compose down

```
PS C:\Users\myss\Desktop\BookStore\BookStore> docker compose up --build
time="2024-10-28T20:58:33+02:00" level=warning msg="C:\\Users\\myss\\Desktop\\BookStore\\BookStore\\docker-compose.yml:
the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 3.2s (27/27) FINISHED                                                                  docker:desktop-linux
 => [catalog_server internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 330B                                                                               0.0s
 => [client_server internal] load metadata for docker.io/library/node:14                                           1.4s
 => [catalog_server internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                                    0.0s
 => [catalog_server internal] load build context                                                                   0.0s
 => => transferring context: 423B                                                                                  0.0s
 => [client_server 1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce574  0.0s
```

Commands and Output:

Inside client server (front end) VScode terminal

To get book info

-Node app.js info-book-item-number

Then enter book id

```
PS C:\Users\myss\Desktop\BookStore\BookStore\client_server> node app.js info-book-item-number
? please enter item number to get info about it:  Cloent server running on port 5000...
? please enter item number to get info about it:  4
Response Data: {
  information: { name: 'Cooking for the Impatient Undergrad', count: 50, cost: 70 }
}
```

To purchase book

-node app.js purchase-book-by-item-number

Then enter book id

```
PS C:\Users\myss\Desktop\BookStore\BookStore\client_server> node app.js purchase-book-by-item-number
? please enter book item number to purchase it:  Cloent server running on port 5000...
? please enter book item number to purchase it:  4
Response Data: {
  message: 'Purchase successful!',
  item: { name: 'Cooking for the Impatient Undergrad', cost: 70 }
}
```

Note that after purchase the book with id 4 the count decreased by 1

```
PS C:\Users\myss\Desktop\BookStore\BookStore\client_server> node app.js search-book-title
? please enter book topic to get details about it:  Cloent server running on port 5000...
? please enter book topic to get details about it:  Distributed systems
  information: { name: 'Cooking for the Impatient Undergrad', count: 49, cost: 70 }
}
```

To search for book

-node app.js search-book-title

Then enter book topic

```
PS C:\Users\myss\Desktop\BookStore\BookStore\client_server> node app.js search-book-title
? please enter book topic to get details about it:  Cloent server running on port 5000...
? please enter book topic to get details about it:  Distributed systems
Response Data: {
  information: [
    { id: 1, name: 'RPCs for Noobs' },
    {
      id: 2,
      name: 'How to get a good grade in DOS in 40 minutes a day'
    }
  ]
}
```

possible improvements and extensions:

add the user budget to purchase request, if budget is enough to buy the book the operation done successfully else error message appear.

```
http://localhost:4000/purchase/id/budget
```

known issues:

Race Conditions (concurrency Issues): If multiple users attempt to purchase the same book simultaneously, race conditions might occur, leading to inconsistencies in stock levels.