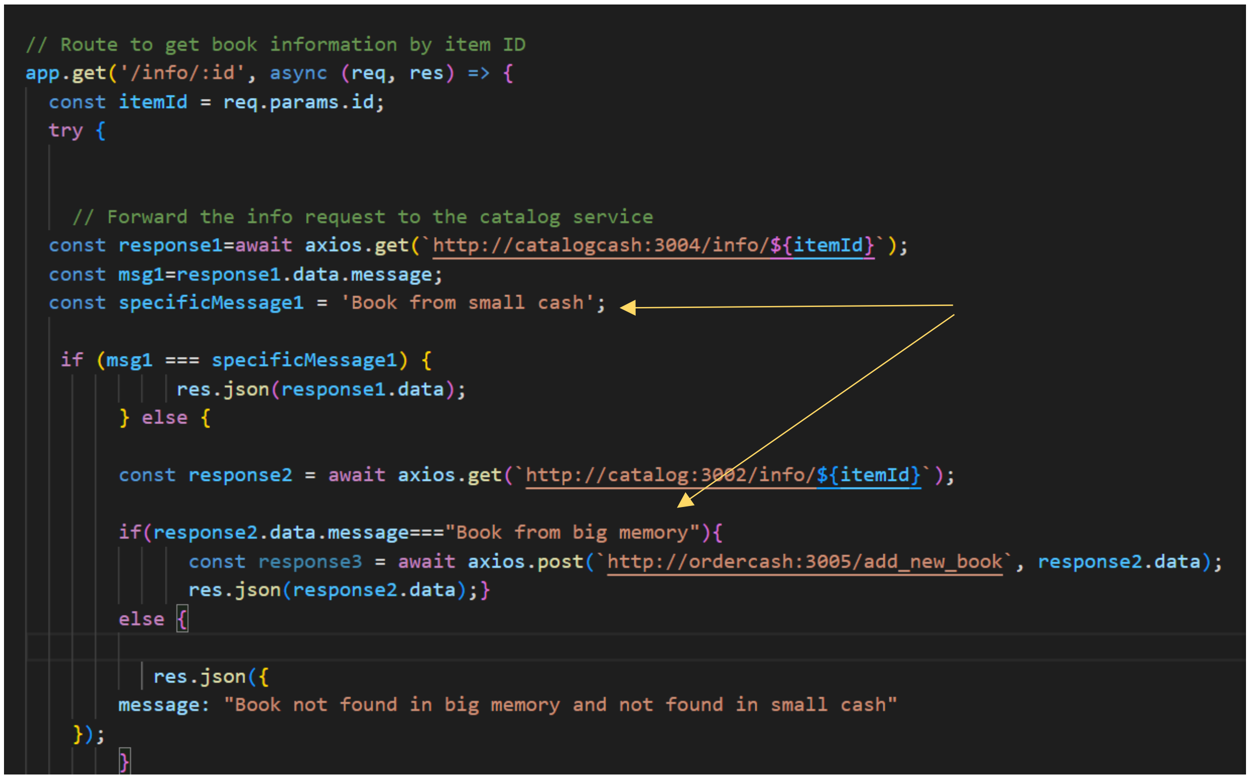
DOS Project Report Part 2

Student Name: Muna Khwaireh Stu#: 12028473

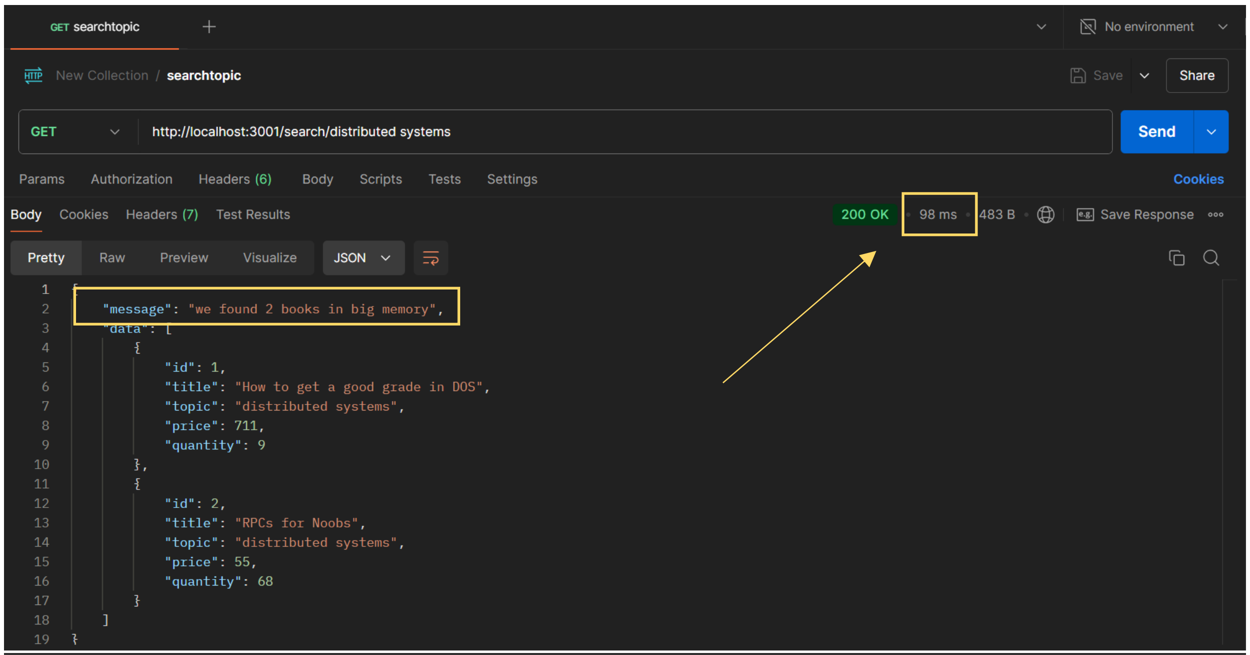
Student Name: Samah Qaradeh Stu#: 12028923

Part1: Cache Consistency



```
// Route to get book information by item ID
app.get('/info/:id', async (req, res) => {
  const itemId = req.params.id;
  try {


    // Forward the info request to the catalog service
  const response1=await axios.get(`http://catalogcash:3004/info/${itemId}`);
  const msg1=response1.data.message;
  const specificMessage1 = 'Book from small cash';

   if (msg1 === specificMessage1) {
           res.json(response1.data);
        } else {

        const response2 = await axios.get(`http://catalog:3002/info/${itemId}`);

        if(response2.data.message==="Book from big memory"){
              const response3 = await axios.post(`http://ordercash:3005/add_new_book`, response2.data);
              res.json(response2.data);}
        else {

           res.json({
        message: "Book not found in big memory and not found in small cash"
    });
        }
```

- When i send `GET` request the **first time**, **before Caching the data**:



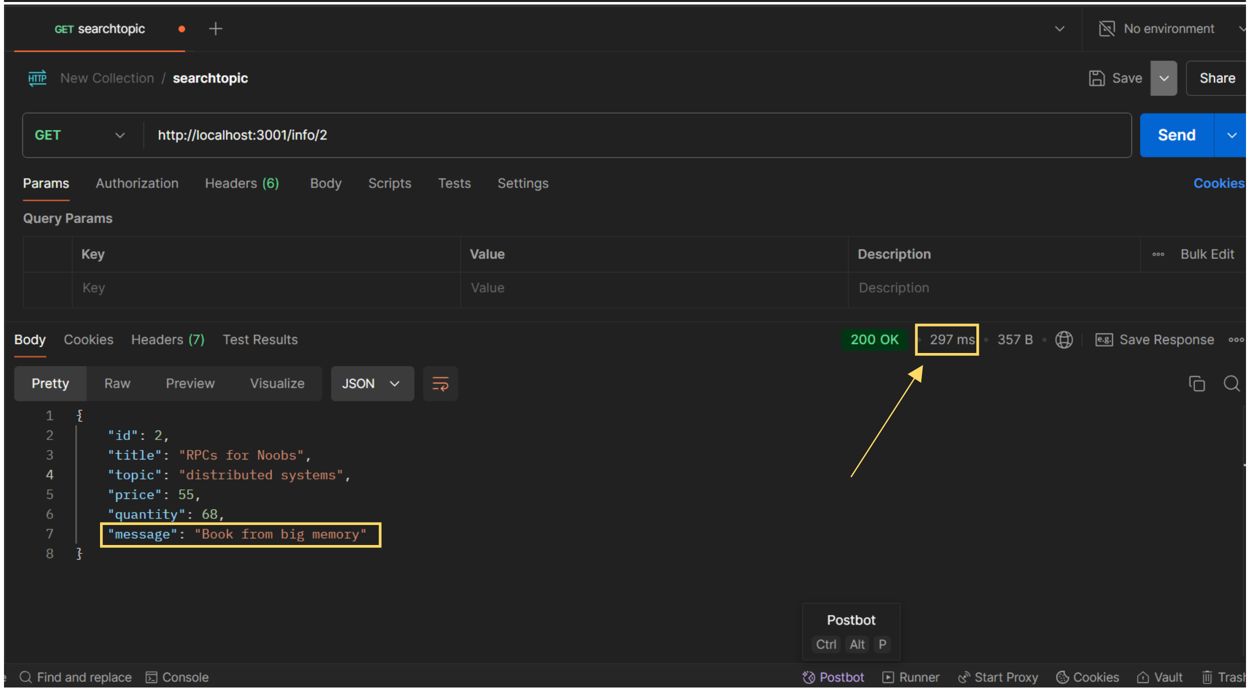
GET searchtopic

New Collection / searchtopic

GET http://localhost:3001/search/distributed systems

200 OK · 98 ms · 483 B

```
{
    "message": "we found 2 books in big memory",
    "data": [
        {
            "id": 1,
            "title": "How to get a good grade in DOS",
            "topic": "distributed systems",
            "price": 711,
            "quantity": 9
        },
        {
            "id": 2,
            "title": "RPCs for Noobs",
            "topic": "distributed systems",
            "price": 55,
            "quantity": 68
        }
    ]
}
```



GET searchtopic

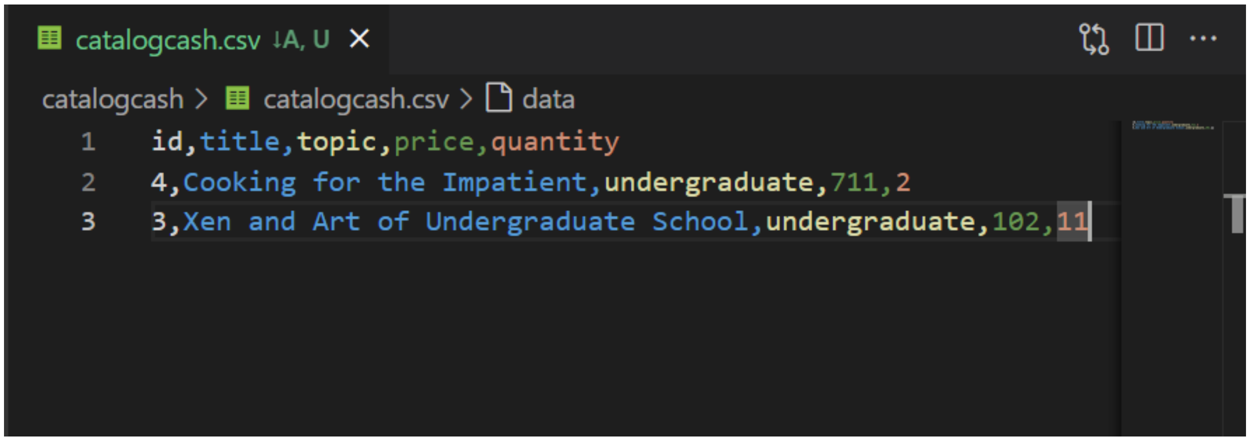
New Collection / searchtopic

GET http://localhost:3001/info/2

200 OK · 297 ms · 357 B

```
{
    "id": 2,
    "title": "RPCs for Noobs",
    "topic": "distributed systems",
    "price": 55,
    "quantity": 68,
    "message": "Book from big memory"
}
```

Our cache capacity is 2 books:



catalogcash.csv ↓A, U

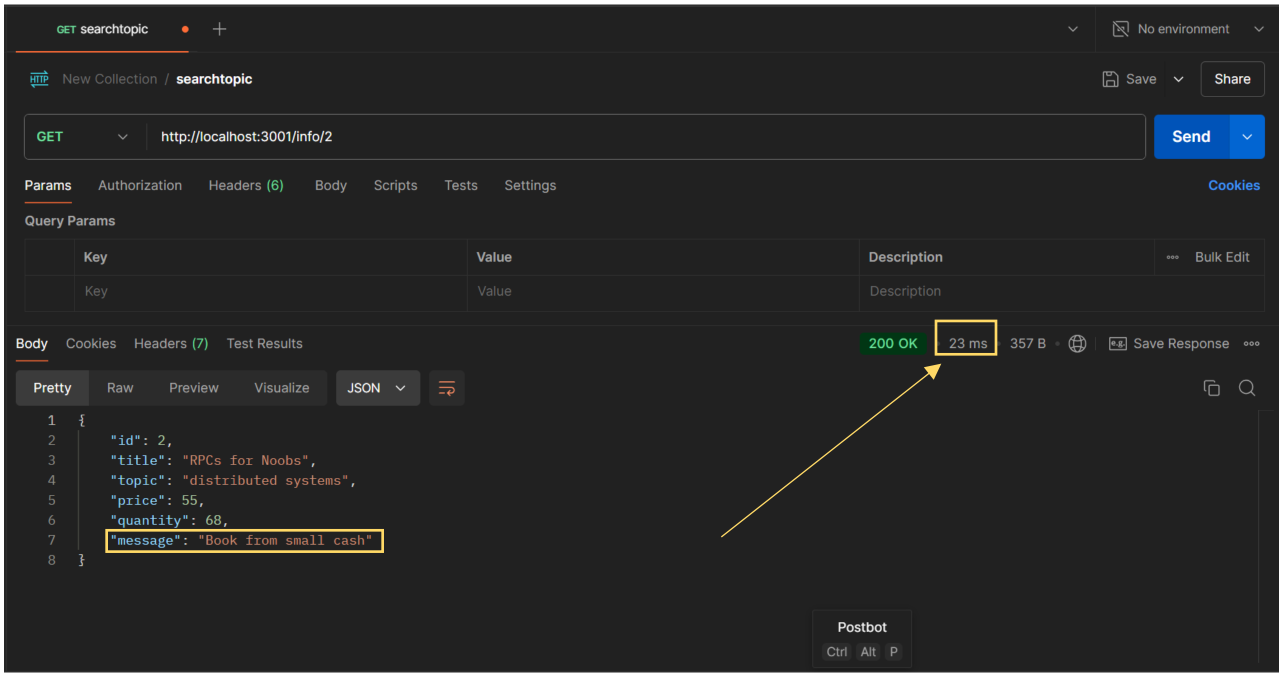
catalogcash > catalogcash.csv > data

```
id,title,topic,price,quantity
4,Cooking for the Impatient,undergraduate,711,2
3,Xen and Art of Undergraduate School,undergraduate,102,11
```

Q1) Compute the average response time (query/buy) of your new systems. What is the response time with and without caching?

- Answers
  - for info: **297ms** for
  - search: **98ms**

with cash:



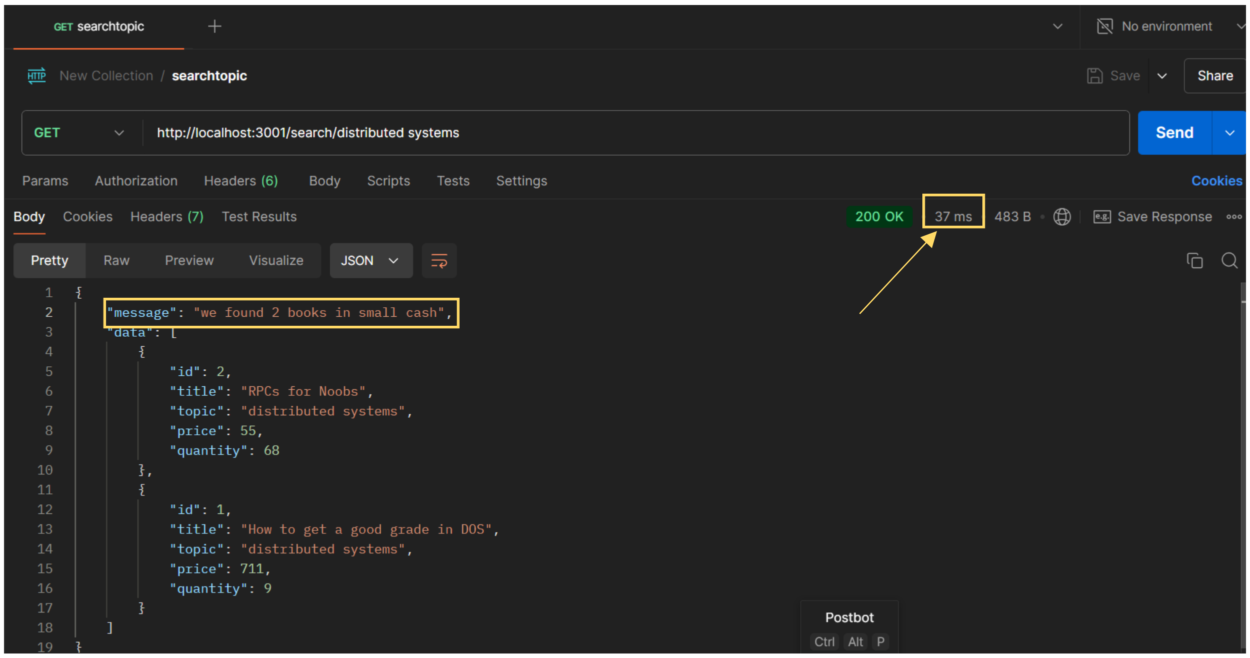
GET searchtopic

New Collection / searchtopic

GET http://localhost:3001/info/2

200 OK 23 ms 357 B

```
{
  "id": 2,
  "title": "RPCs for Noobs",
  "topic": "distributed systems",
  "price": 55,
  "quantity": 68,
  "message": "Book from small cash"
}
```


GET searchtopic
No environment
New Collection / searchtopic
Save
Share
GET
http://localhost:3001/search/distributed systems
Send
Params Authorization Headers (6) Body Scripts Tests Settings
Cookies
Body Cookies Headers (7) Test Results
200 OK
37 ms
483 B
Save Response
Pretty Raw Preview Visualize JSON
Postbot
Ctrl Alt P

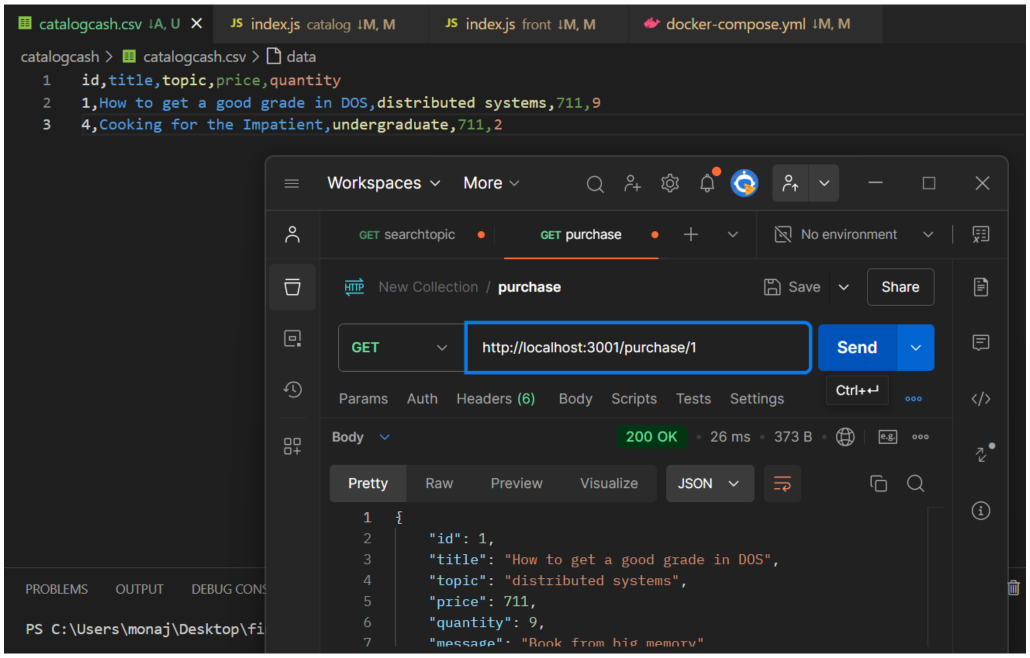
```
{
    "message": "we found 2 books in small cash",
    "data": [
        {
            "id": 2,
            "title": "RPCs for Noobs",
            "topic": "distributed systems",
            "price": 55,
            "quantity": 68
        },
        {
            "id": 1,
            "title": "How to get a good grade in DOS",
            "topic": "distributed systems",
            "price": 711,
            "quantity": 9
        }
    ]
}
```

Q2) How much does caching help?

- Answers for info: **23ms , 297/23 —> 12.9 Faster than without cache**
  - search: **37ms, 98/37—> 2.65 Faster than without using cache**
  -

Invalidate Message

At the first I will purchase a book that is already at the cache:
This is before make the request:



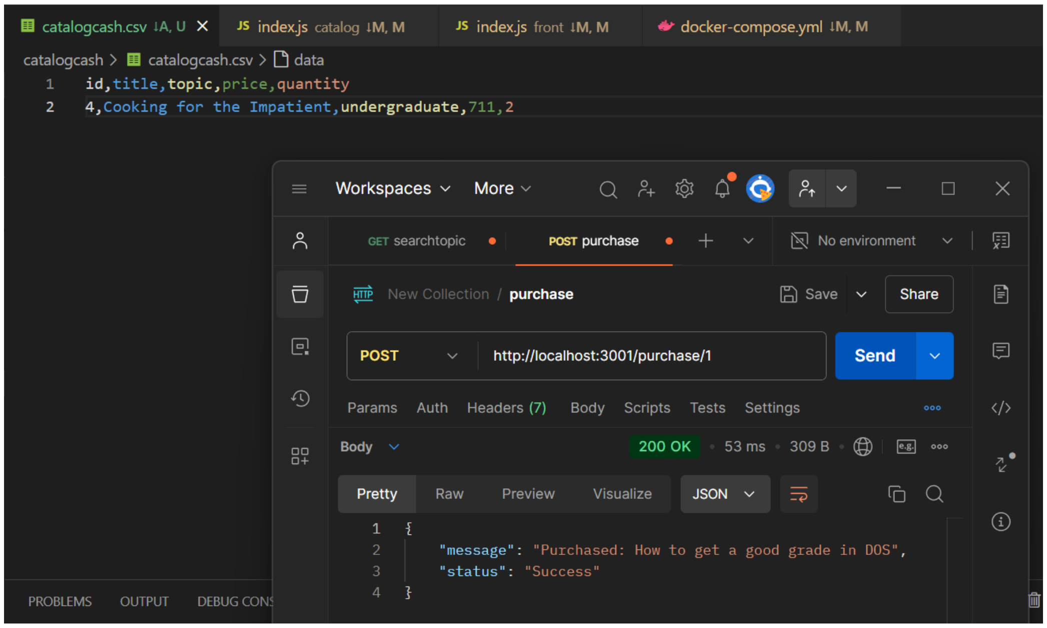
```
id,title,topic,price,quantity
1,How to get a good grade in DOS,distributed systems,711,9
4,Cooking for the Impatient,undergraduate,711,2
```

GET http://localhost:3001/purchase/1 — 200 OK • 26 ms • 373 B

```
{
    "id": 1,
    "title": "How to get a good grade in DOS",
    "topic": "distributed systems",
    "price": 711,
    "quantity": 9,
    "message": "Book from big memory"
```

```
PS C:\Users\monaj\Desktop\fi
```

And here after the request, the book just get out of the cache:

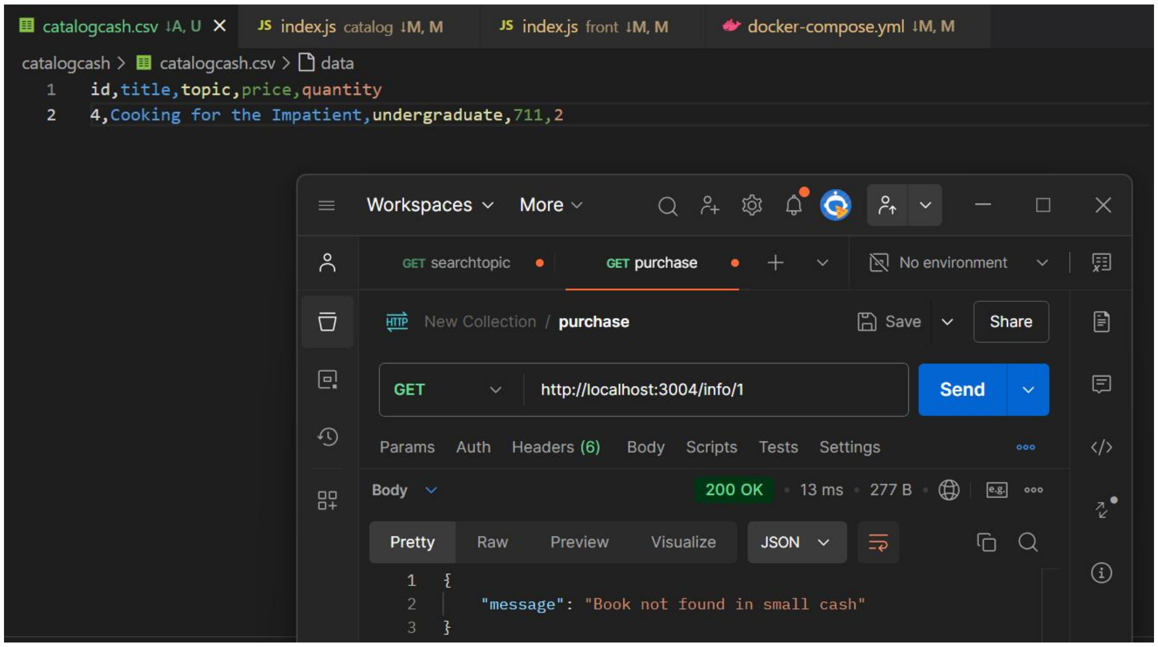


```
id,title,topic,price,quantity
4,Cooking for the Impatient,undergraduate,711,2
```

POST http://localhost:3001/purchase/1 — 200 OK • 53 ms • 309 B

```
{
    "message": "Purchased: How to get a good grade in DOS",
    "status": "Success"
}
```

And if I search at cache for it:



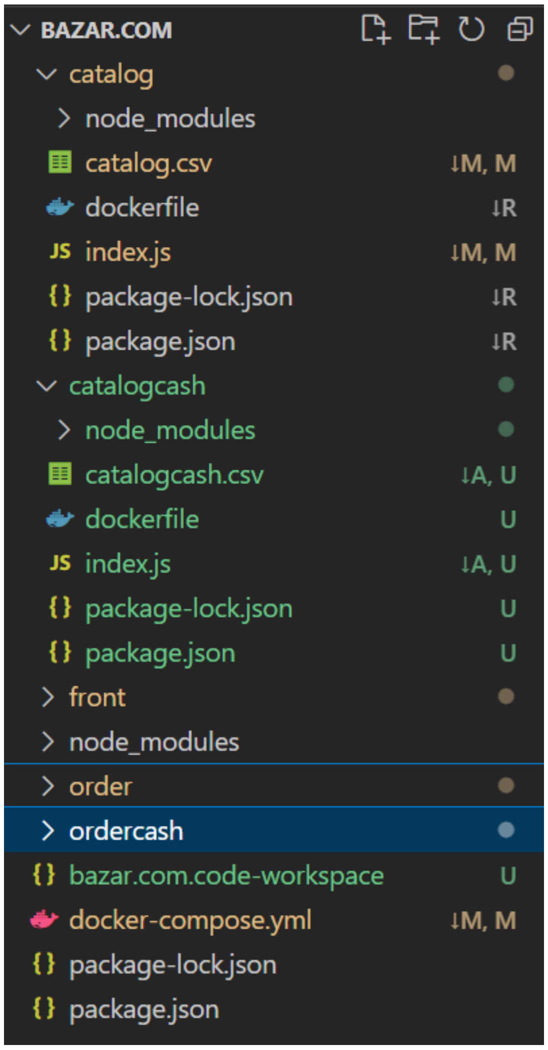
```
id,title,topic,price,quantity
4,Cooking for the Impatient,undergraduate,711,2
```

GET http://localhost:3004/info/1

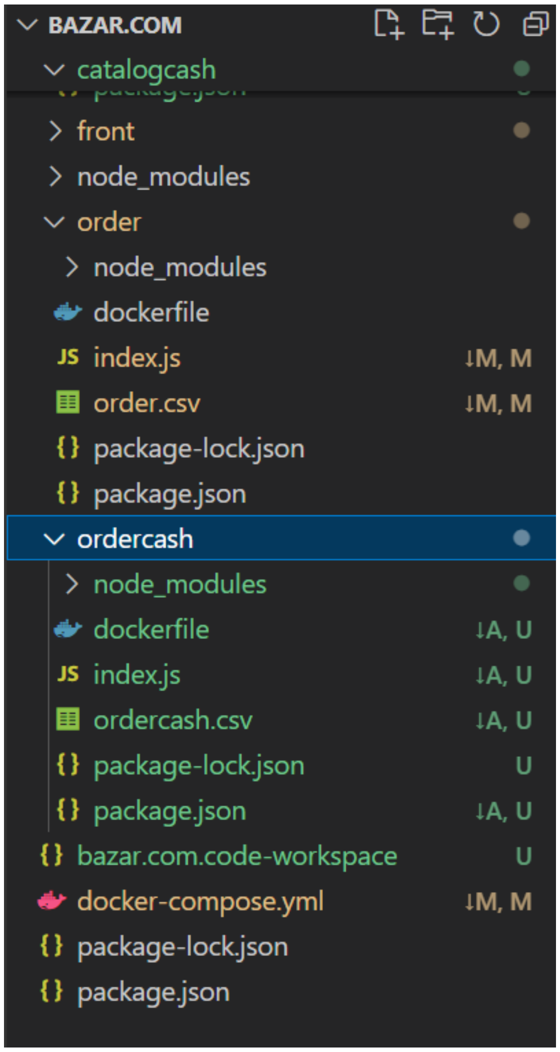
```
{
    "message": "Book not found in small cash"
}
```

Expirement Run	Without Cache	With Cache	#of Times when using Cache speed
1	46	9	46/9 = 5.11 Times
2	26	7	26/7 = 3.71 Times
3	27	9	27/7 = 3.85 Times

For Replication Services & Database:



```
BAZAR.COM
  catalog
    node_modules
    catalog.csv                    ↓M, M
    dockerfile                     ↓R
    index.js                       ↓M, M
    package-lock.json              ↓R
    package.json                   ↓R
  catalogcash
    node_modules
    catalogcash.csv                ↓A, U
    dockerfile                     U
    index.js                       ↓A, U
    package-lock.json              U
    package.json                   U
  front
  node_modules
  order
  ordercash
  bazar.com.code-workspace         U
  docker-compose.yml               ↓M, M
  package-lock.json
  package.json
```



```
BAZAR.COM
  catalogcash
  front
  node_modules
  order
    node_modules
    dockerfile
    index.js                       ↓M, M
    order.csv                      ↓M, M
    package-lock.json
    package.json
  ordercash
    node_modules
    dockerfile                     ↓A, U
    index.js                       ↓A, U
    ordercash.csv                  ↓A, U
    package-lock.json              U
    package.json                   ↓A, U
  bazar.com.code-workspace         U
  docker-compose.yml               ↓M, M
  package-lock.json
  package.json
```

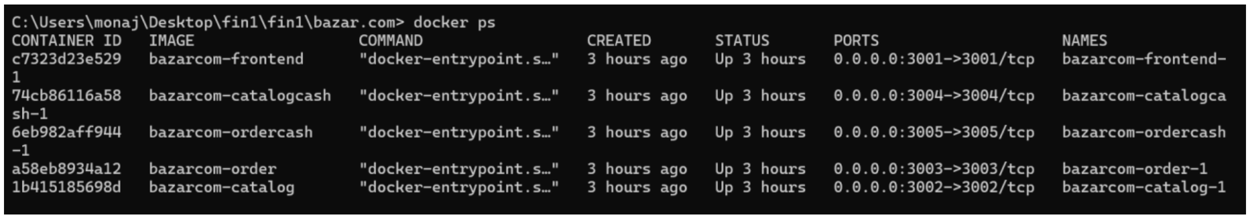
Part2: Dockerize your Application (Optional part)

I Construct my project Dockerized from scratch, i create docker container (image) for each service, and each service has it’s own port, and i use volume for sharing data between the Guest OS (Docker Containers) & Host OS , and i create docker-compose file to run all containers at the same time with 1 simple command .

My compose.yml file:

```
version: '3.8'

services:
  frontend:
    build: ./front
    ports:
      - "3001:3001"
    depends_on:
      - catalog
      - order
      - catalogcash
      - ordercash
    networks:
      - app-network

  catalog:
    build: ./catalog
    ports:
      - "3002:3002"
    volumes:
      - ./catalog:/catalog  # Mount the catalog directory
    networks:
      - app-network
order:
    build: ./order
    ports:
      - "3003:3003"
    volumes:
      - ./catalog:/catalog  # Mount the same catalog directory for access
      - ./order:/order
    networks:
      - app-network
  catalogcash:
    build: ./catalogcash
    ports:
      - "3004:3004"
    depends_on:
      - catalog
      - order
    volumes:
      - ./catalogcash:/catalogcash  # Mount the catalog directory
    networks:
      - app-network

  ordercash:
    build: ./ordercash
    ports:
      - "3005:3005"
    depends_on:
      - catalog
      - order
    volumes:
      - ./catalogcash:/catalogcash  # Mount the same catalog directory for
access
      - ./ordercash:/ordercash
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

Docker Containers(images) while running:



```
C:\Users\monaj\Desktop\fin1\fin1\bazar.com> docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED       STATUS        PORTS                    NAMES
c7323d23e529   bazarcom-frontend      "docker-entrypoint.s…"   3 hours ago   Up 3 hours    0.0.0.0:3001->3001/tcp   bazarcom-frontend-
1
74cb86116a58   bazarcom-catalogcash   "docker-entrypoint.s…"   3 hours ago   Up 3 hours    0.0.0.0:3004->3004/tcp   bazarcom-catalogca
sh-1
6eb982aff944   bazarcom-ordercash     "docker-entrypoint.s…"   3 hours ago   Up 3 hours    0.0.0.0:3005->3005/tcp   bazarcom-ordercash
-1
a58eb8934a12   bazarcom-order         "docker-entrypoint.s…"   3 hours ago   Up 3 hours    0.0.0.0:3003->3003/tcp   bazarcom-order-1
1b415185698d   bazarcom-catalog       "docker-entrypoint.s…"   3 hours ago   Up 3 hours    0.0.0.0:3002->3002/tcp   bazarcom-catalog-1
```

For Running The Project, it’s the same for Part1:

```
docker-compose up -d -- build
```