# Project Simulation

Project Simulation

# Cake Catalog

Cake Catalog

Report Submitted to

KrishnaveniMohana By

Karthik Donepalli

13597031

# Introduction

## Objective

1. Implementing business capability using microservices.
2. Communicate via Application program Interface or messaging.
3. Container adoption to support application deployment and self-contained execution environment.

## Tools and Technologies

1. Visual Studio Code
2. Mongodb
3. Postman
4. Docker

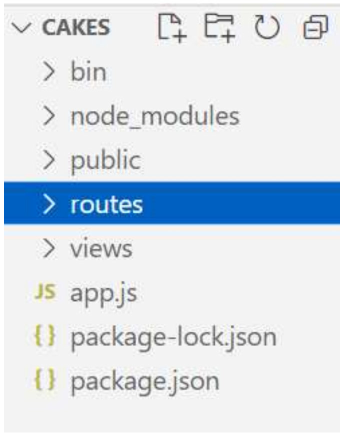
## Features

1. Browse through the cake items
2. Apply filters for selection
3. Add cakes to the shopping basket
4. Display items in the basket

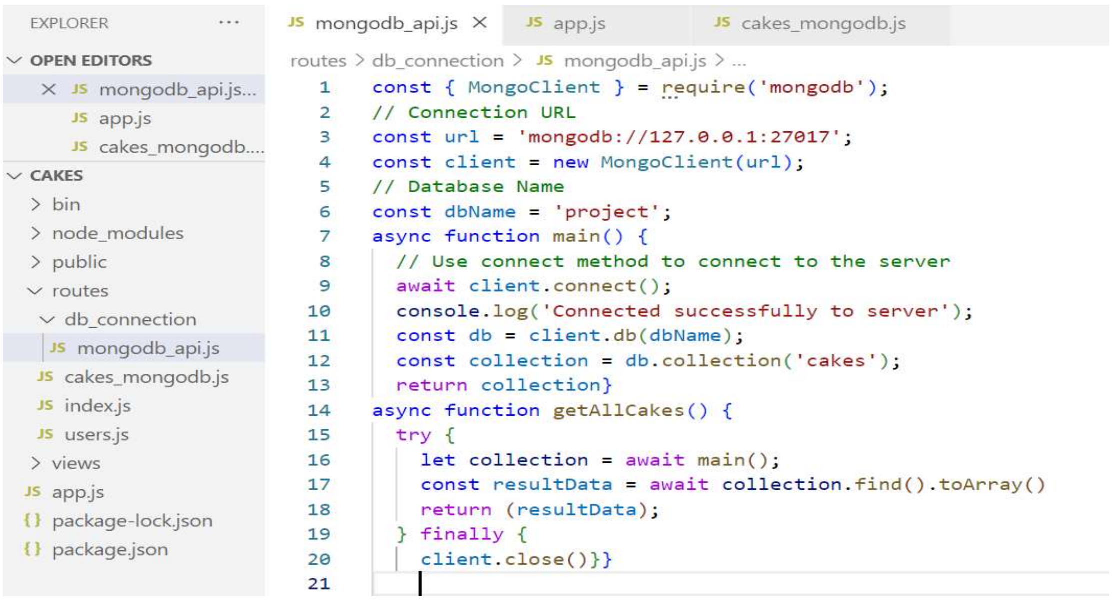
Cake delight- Cakes Microservice

1. List the available cake items
   a. Create the project body: npx express-generator

   b. Install dependencies: npm install

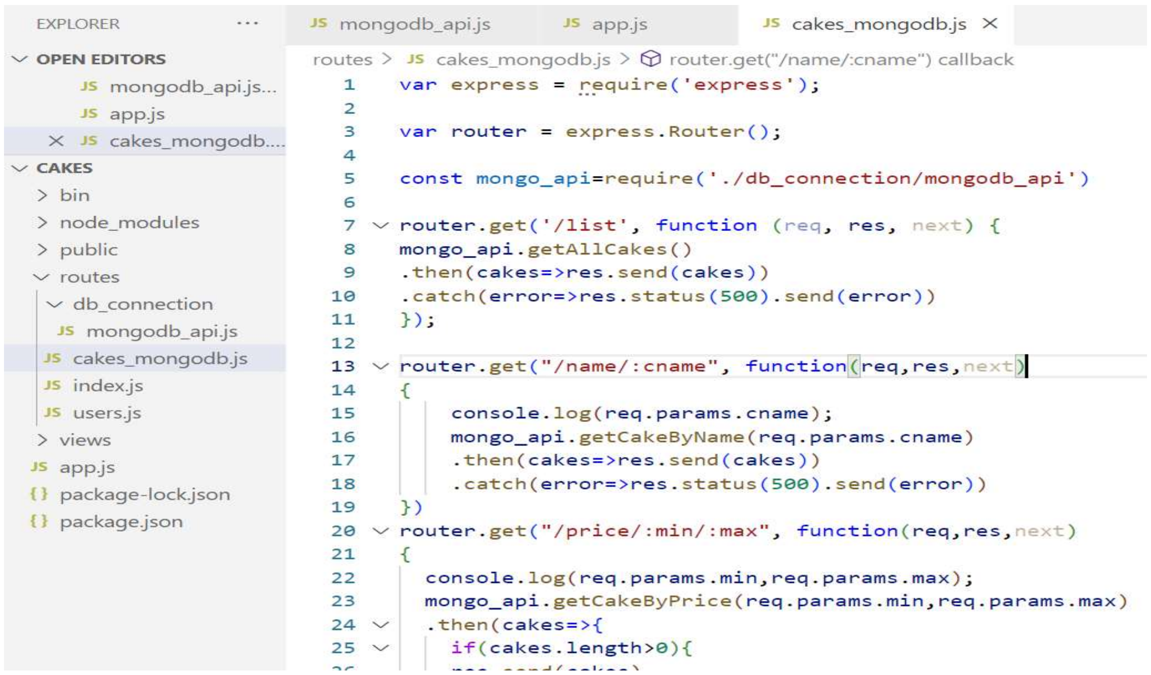


c. Create js file to connect Mongodb database:



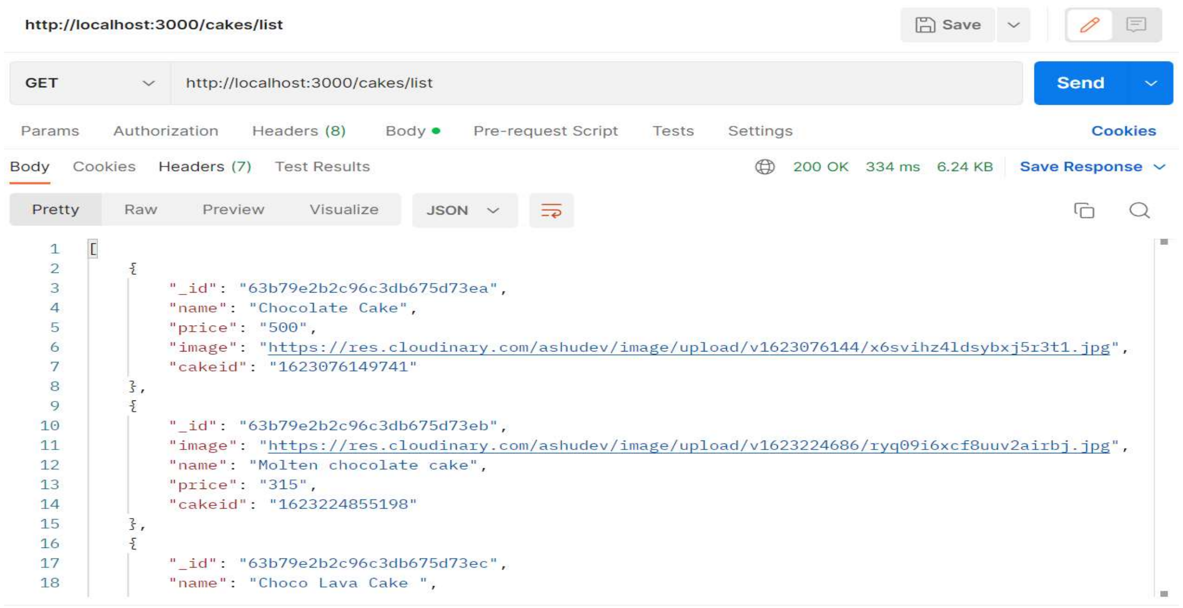
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';
async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cakes');
  return collection}
async function getAllCakes() {
  try {
    let collection = await main();
    const resultData = await collection.find().toArray()
    return (resultData);
  } finally {
    client.close()}}

```

## d. Create the route file for the end points:



```
var express = require('express');

var router = express.Router();

const mongo_api=require('./db_connection/mongodb_api')

router.get('/list', function (req, res, next) {
mongo_api.getAllCakes()
.then(cakes=>res.send(cakes))
.catch(error=>res.status(500).send(error))
});

router.get("/name/:cname", function(req,res,next)
{
    console.log(req.params.cname);
    mongo_api.getCakeByName(req.params.cname)
    .then(cakes=>res.send(cakes))
    .catch(error=>res.status(500).send(error))
})
router.get("/price/:min/:max", function(req,res,next)
{
  console.log(req.params.min,req.params.max);
  mongo_api.getCakeByPrice(req.params.min,req.params.max)
  .then(cakes=>{
    if(cakes.length>0){
```

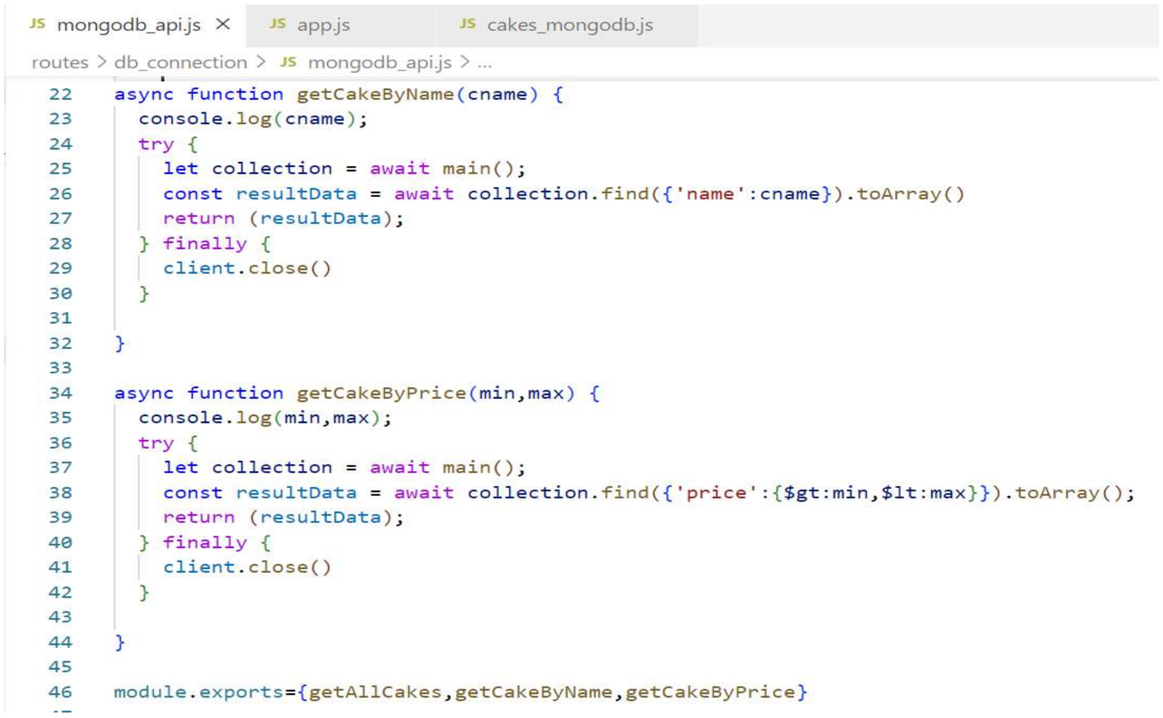
## e. Results:



```
[
    {
        "_id": "63b79e2b2c96c3db675d73ea",
        "name": "Chocolate Cake",
        "price": "500",
        "image": "https://res.cloudinary.com/ashudev/image/upload/v1623076144/x6svihz4ldsybxj5r3t1.jpg",
        "cakeid": "1623076149741"
    },
    {
        "_id": "63b79e2b2c96c3db675d73eb",
        "image": "https://res.cloudinary.com/ashudev/image/upload/v1623224686/ryq09i6xcf8uuv2airbj.jpg",
        "name": "Molten chocolate cake",
        "price": "315",
        "cakeid": "1623224855198"
    },
    {
        "_id": "63b79e2b2c96c3db675d73ec",
        "name": "Choco Lava Cake ",
```

2. Filter items by name and price range

a. Again create access from database:

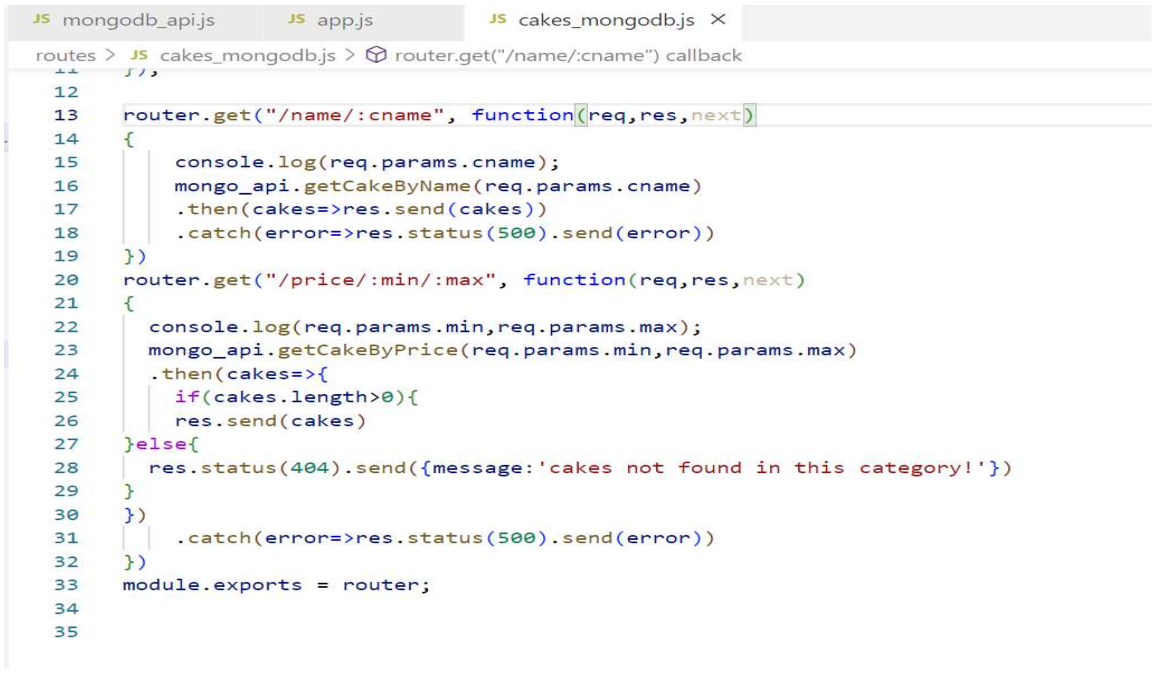


mongodb_api.js × | app.js | cakes_mongodb.js

routes > db_connection > mongodb_api.js > ...

```javascript
async function getCakeByName(cname) {
  console.log(cname);
  try {
    let collection = await main();
    const resultData = await collection.find({'name':cname}).toArray()
    return (resultData);
  } finally {
    client.close()
  }

}

async function getCakeByPrice(min,max) {
  console.log(min,max);
  try {
    let collection = await main();
    const resultData = await collection.find({'price':{$gt:min,$lt:max}}).toArray();
    return (resultData);
  } finally {
    client.close()
  }

}

module.exports={getAllCakes,getCakeByName,getCakeByPrice}
```

b. Give necessary end points in route:

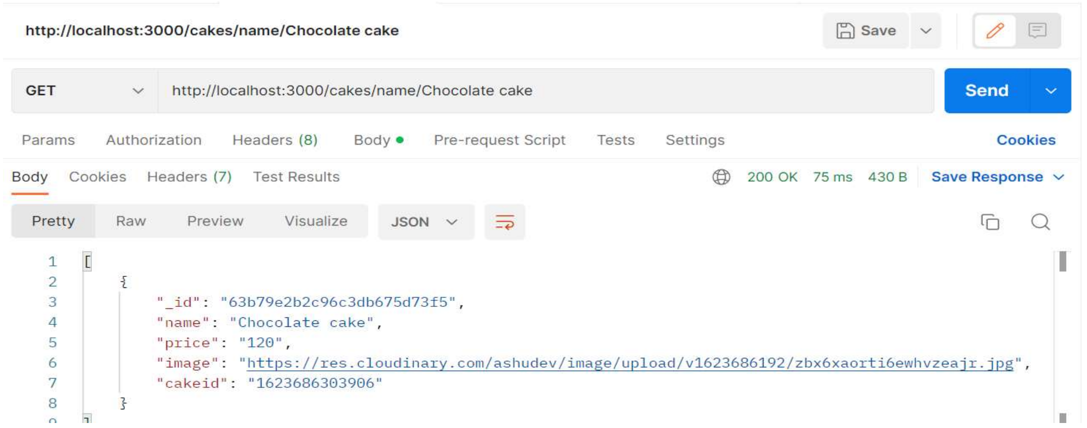


mongodb_api.js | app.js | cakes_mongodb.js ×

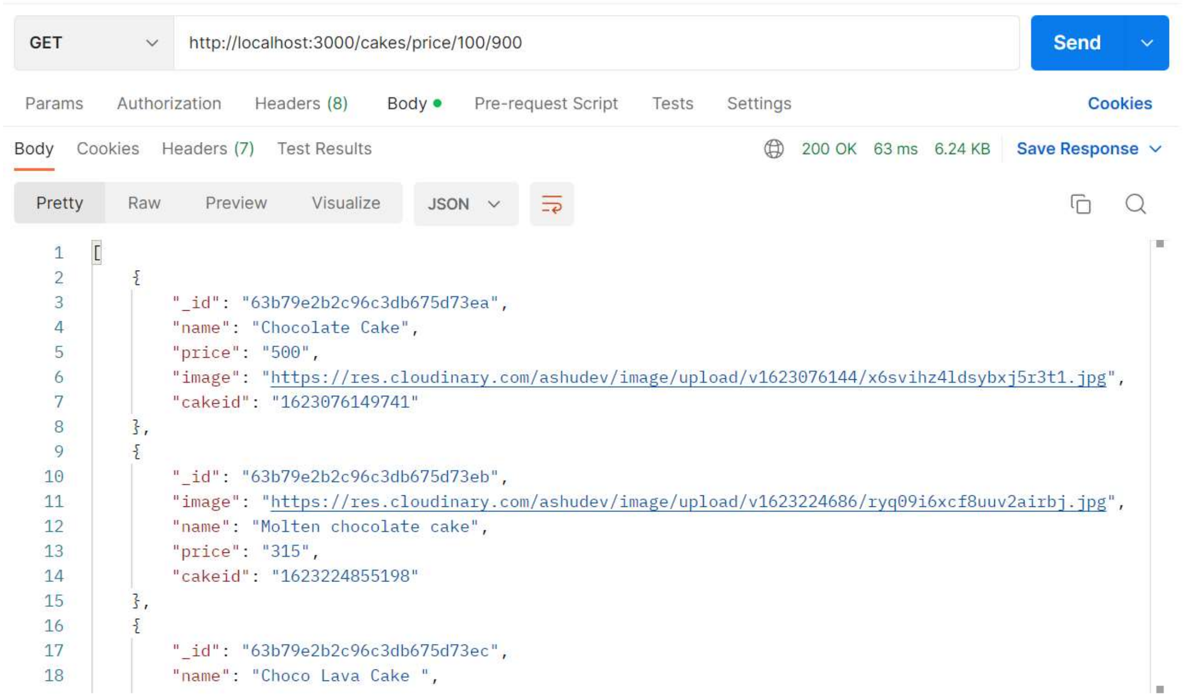
routes > cakes_mongodb.js > router.get("/name/:cname") callback

```javascript

router.get("/name/:cname", function(req,res,next)
{
    console.log(req.params.cname);
    mongo_api.getCakeByName(req.params.cname)
    .then(cakes=>res.send(cakes))
    .catch(error=>res.status(500).send(error))
})
router.get("/price/:min/:max", function(req,res,next)
{
  console.log(req.params.min,req.params.max);
  mongo_api.getCakeByPrice(req.params.min,req.params.max)
  .then(cakes=>{
    if(cakes.length>0){
    res.send(cakes)
}else{
  res.status(404).send({message:'cakes not found in this category!'})
}
})
    .catch(error=>res.status(500).send(error))
})
module.exports = router;
```

c. Results:


http://localhost:3000/cakes/name/Chocolate cake
Save
GET
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies
Body Cookies Headers (7) Test Results
200 OK 75 ms 430 B Save Response
Pretty Raw Preview Visualize JSON

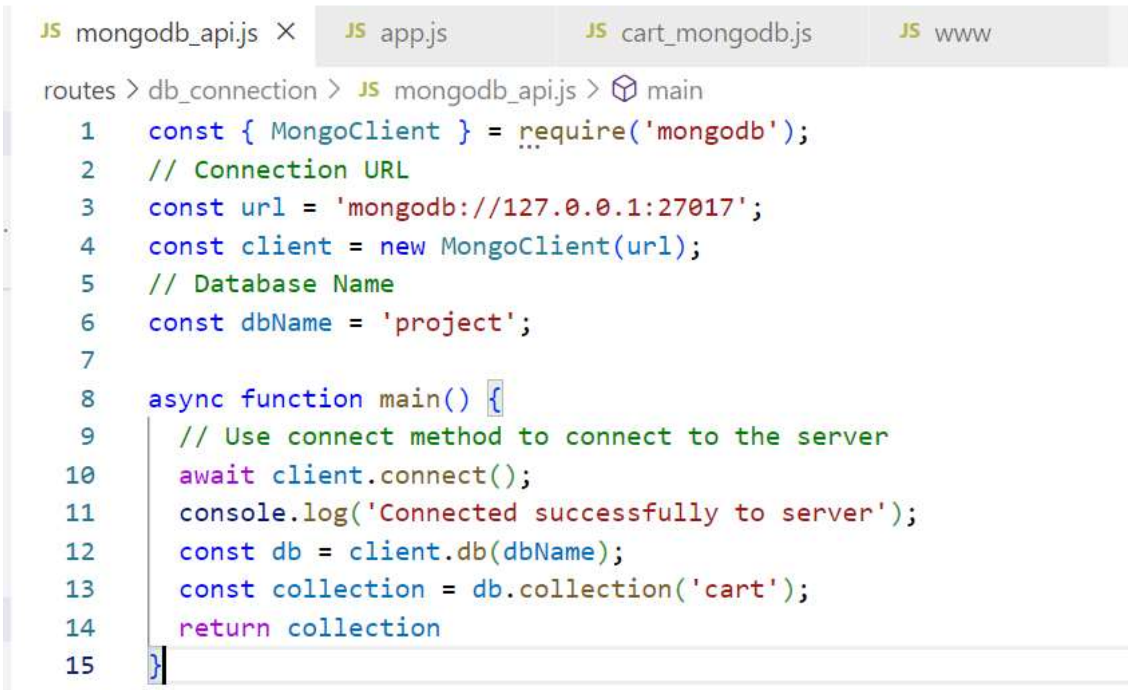
```
[
    {
        "_id": "63b79e2b2c96c3db675d73f5",
        "name": "Chocolate cake",
        "price": "120",
        "image": "https://res.cloudinary.com/ashudev/image/upload/v1623686192/zbx6xaorti6ewhvzeajr.jpg",
        "cakeid": "1623686303906"
    }
]
```


GET
http://localhost:3000/cakes/price/100/900
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies
Body Cookies Headers (7) Test Results
200 OK 63 ms 6.24 KB Save Response
Pretty Raw Preview Visualize JSON

```
[
    {
        "_id": "63b79e2b2c96c3db675d73ea",
        "name": "Chocolate Cake",
        "price": "500",
        "image": "https://res.cloudinary.com/ashudev/image/upload/v1623076144/x6svihz4ldsybxj5r3t1.jpg",
        "cakeid": "1623076149741"
    },
    {
        "_id": "63b79e2b2c96c3db675d73eb",
        "image": "https://res.cloudinary.com/ashudev/image/upload/v1623224686/ryq09i6xcf8uuv2airbj.jpg",
        "name": "Molten chocolate cake",
        "price": "315",
        "cakeid": "1623224855198"
    },
    {
        "_id": "63b79e2b2c96c3db675d73ec",
        "name": "Choco Lava Cake ",
```

Cake delight- Cart Microservices

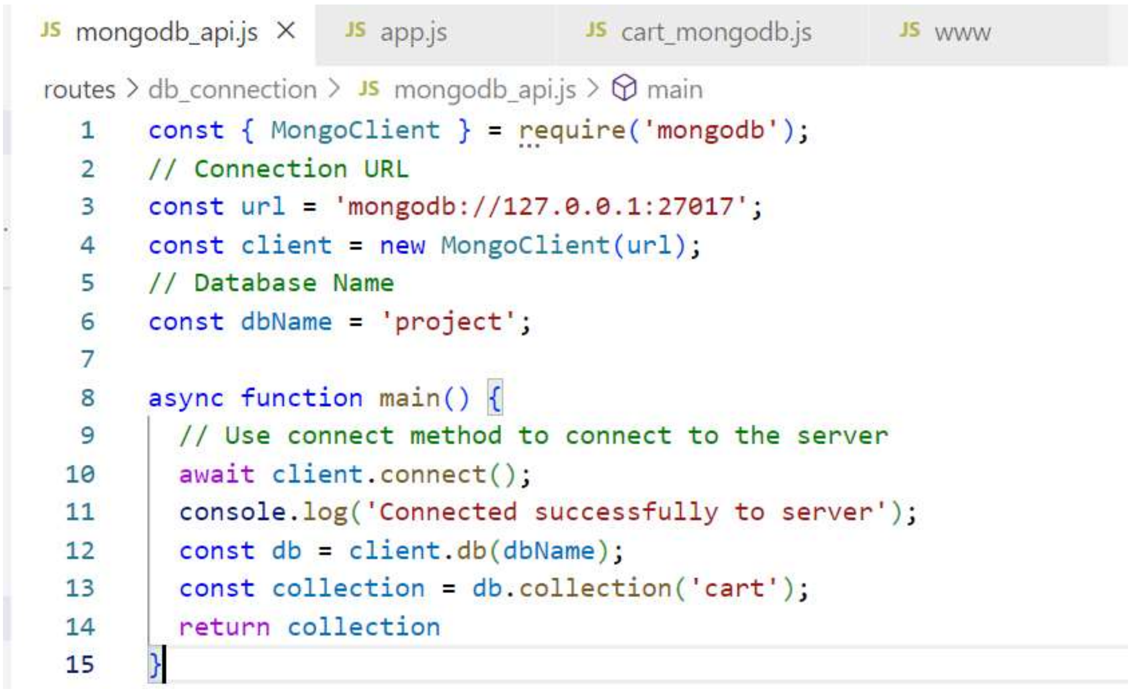
1. Add items to the cart
   a. Again create the connection.



mongodb_api.js × | app.js | cart_mongodb.js | www

routes > db_connection > mongodb_api.js > main

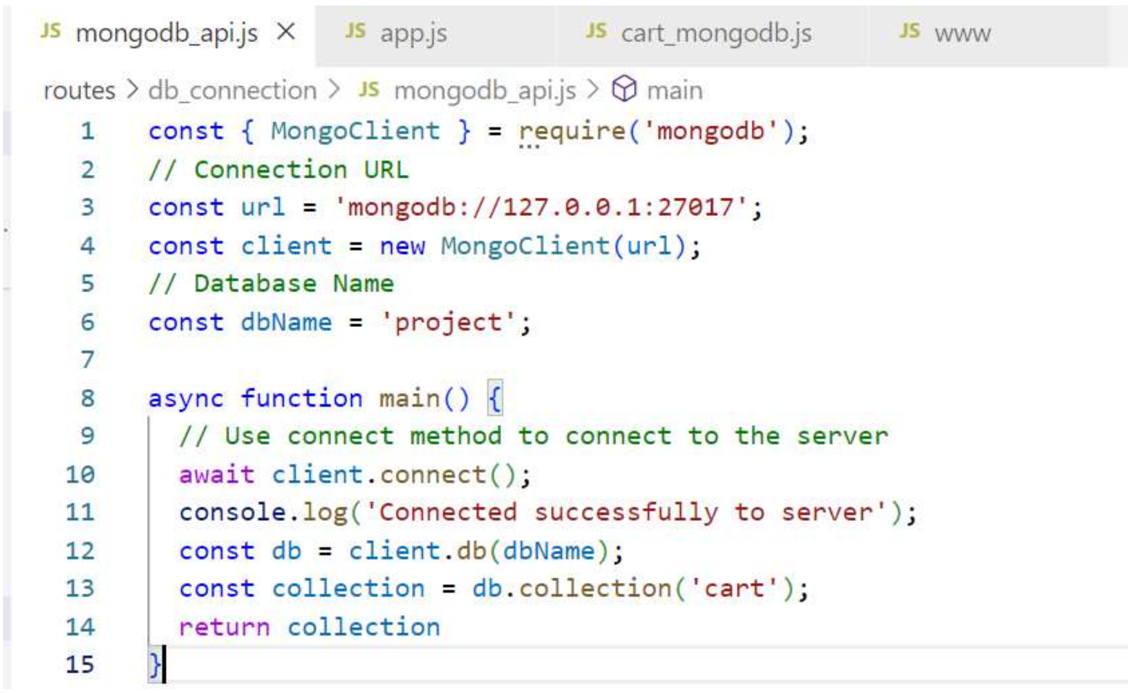
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



mongodb_api.js × | app.js | cart_mongodb.js | www

routes > db_connection > mongodb_api.js > main

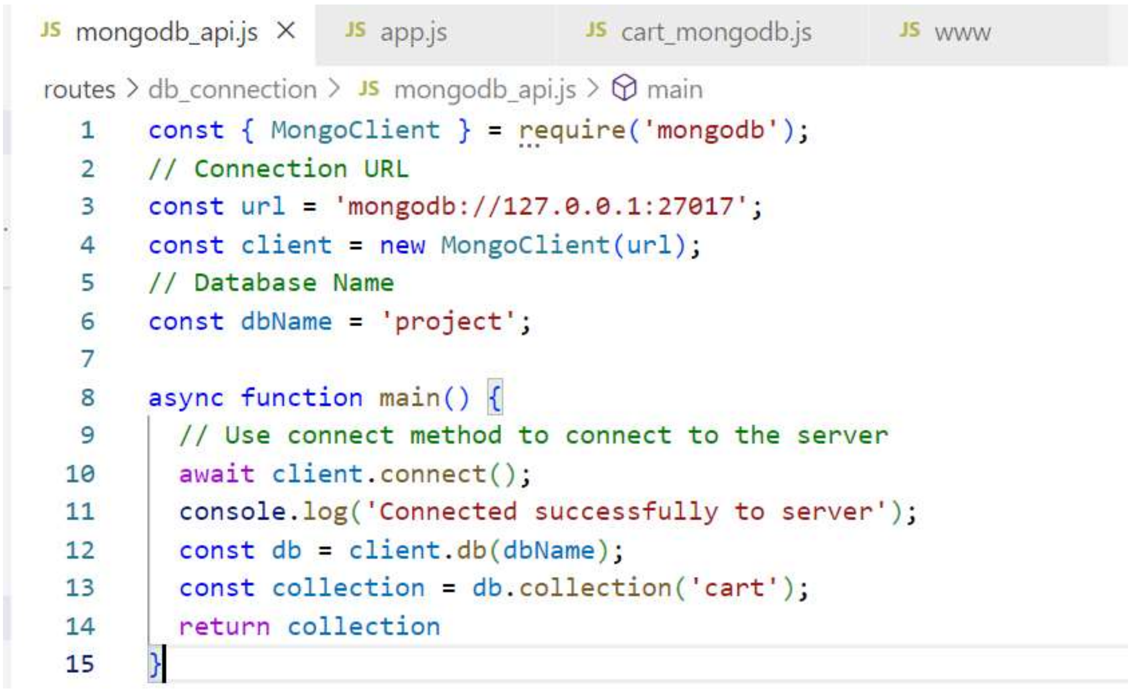
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```


JS mongodb_api.js × JS app.js JS cart_mongodb.js JS www

routes > db_connection > JS mongodb_api.js > main

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```

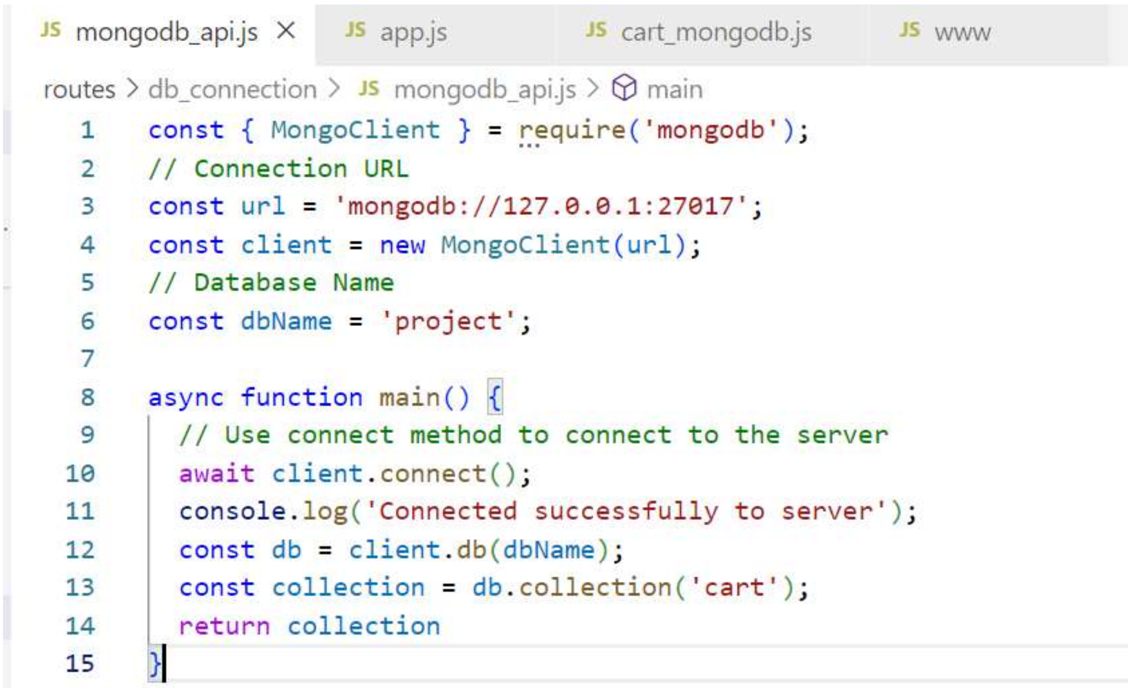
V



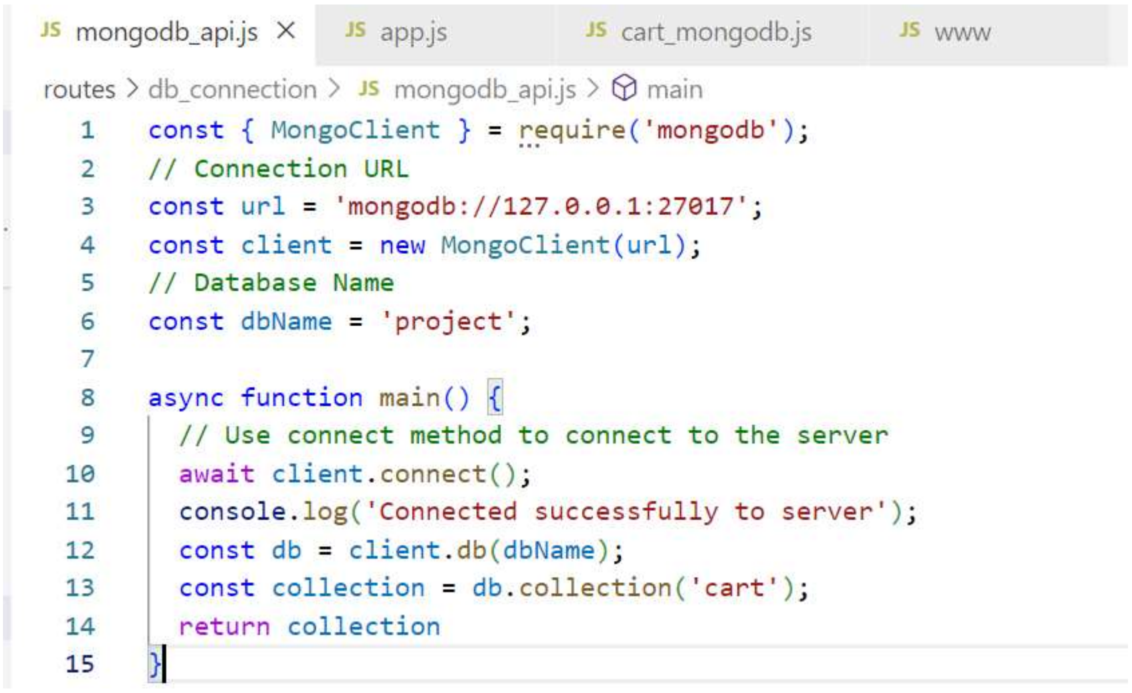
JS mongodb_api.js × JS app.js JS cart_mongodb.js JS www

routes > db_connection > JS mongodb_api.js > main

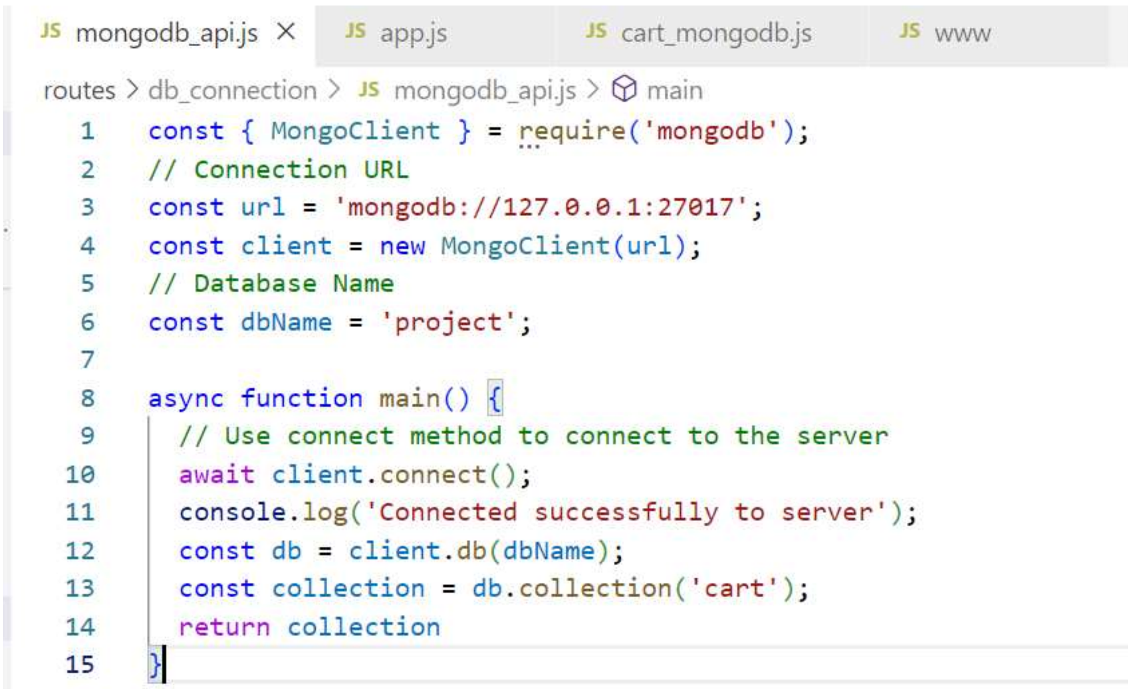
```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



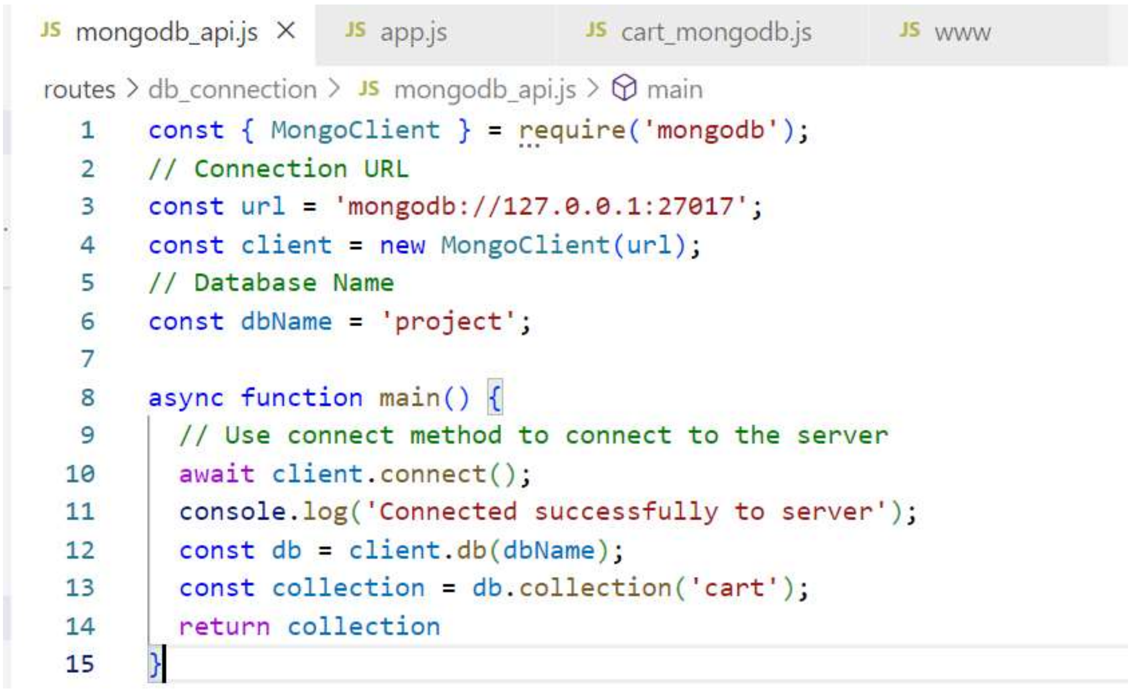
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



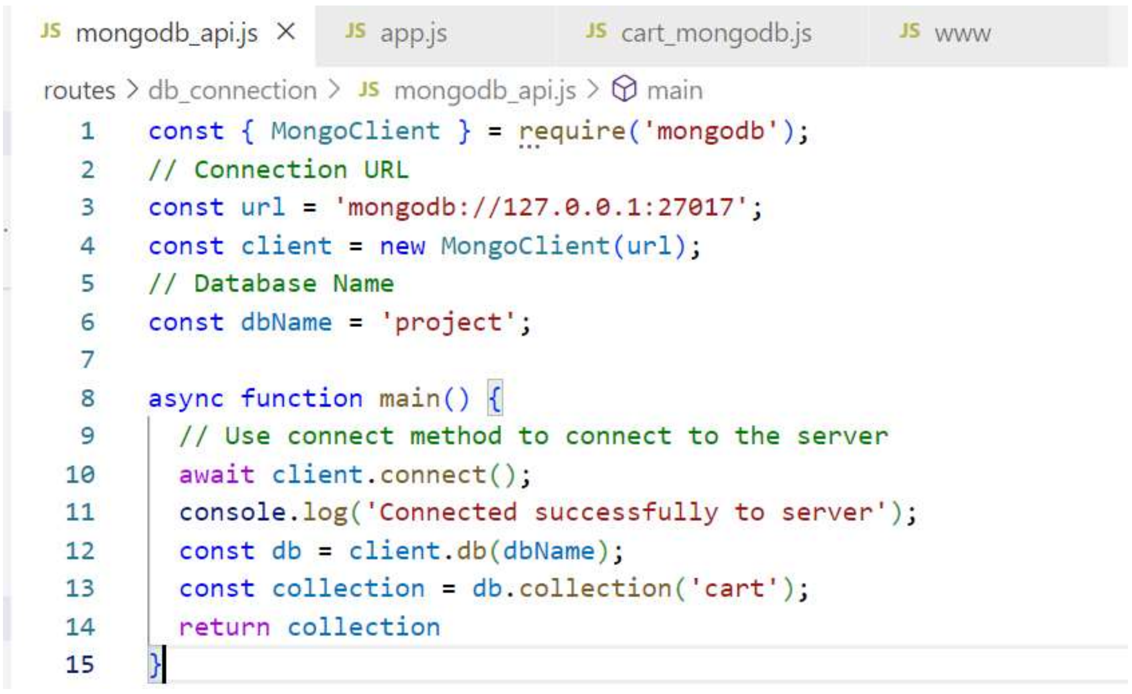
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



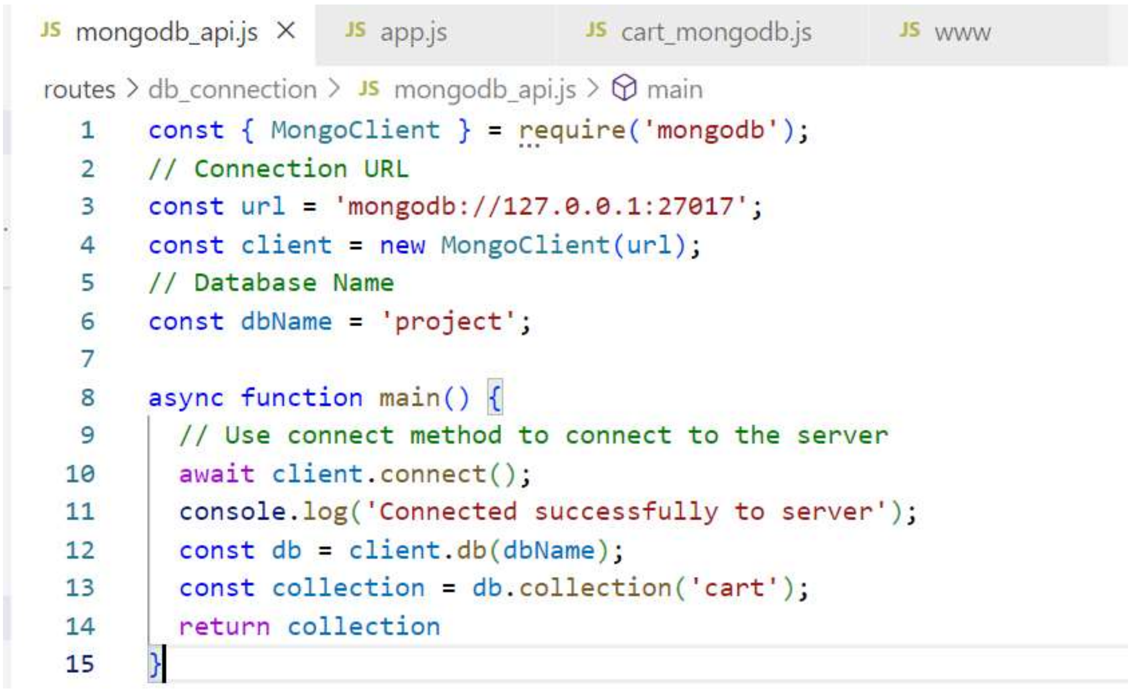
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



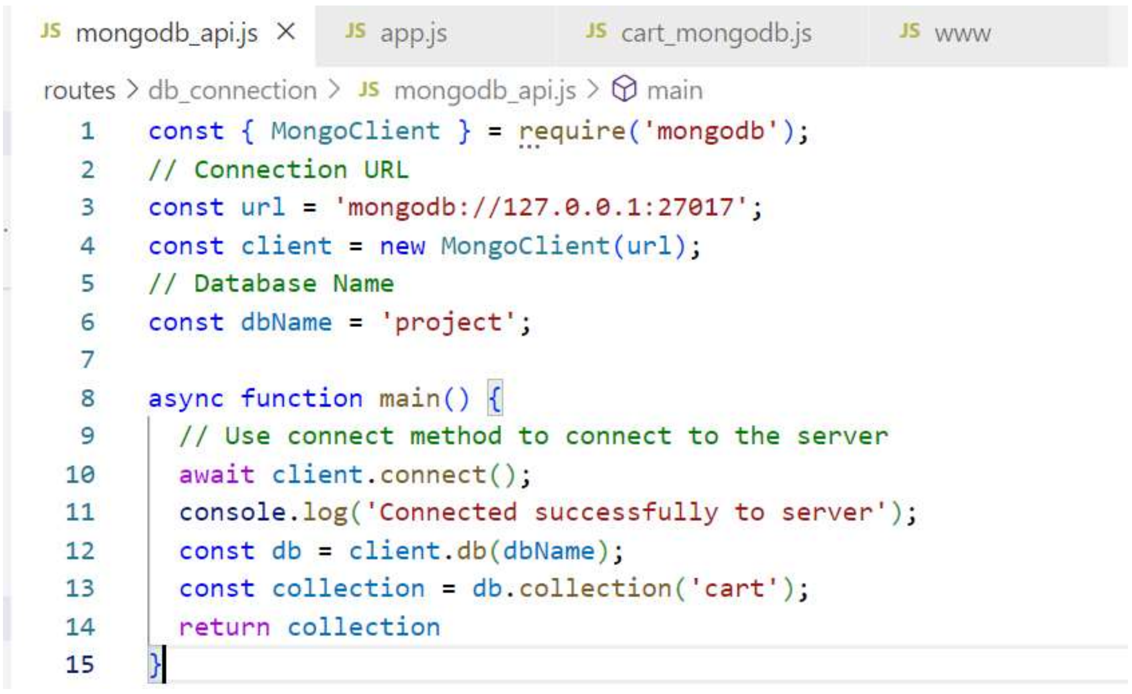
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



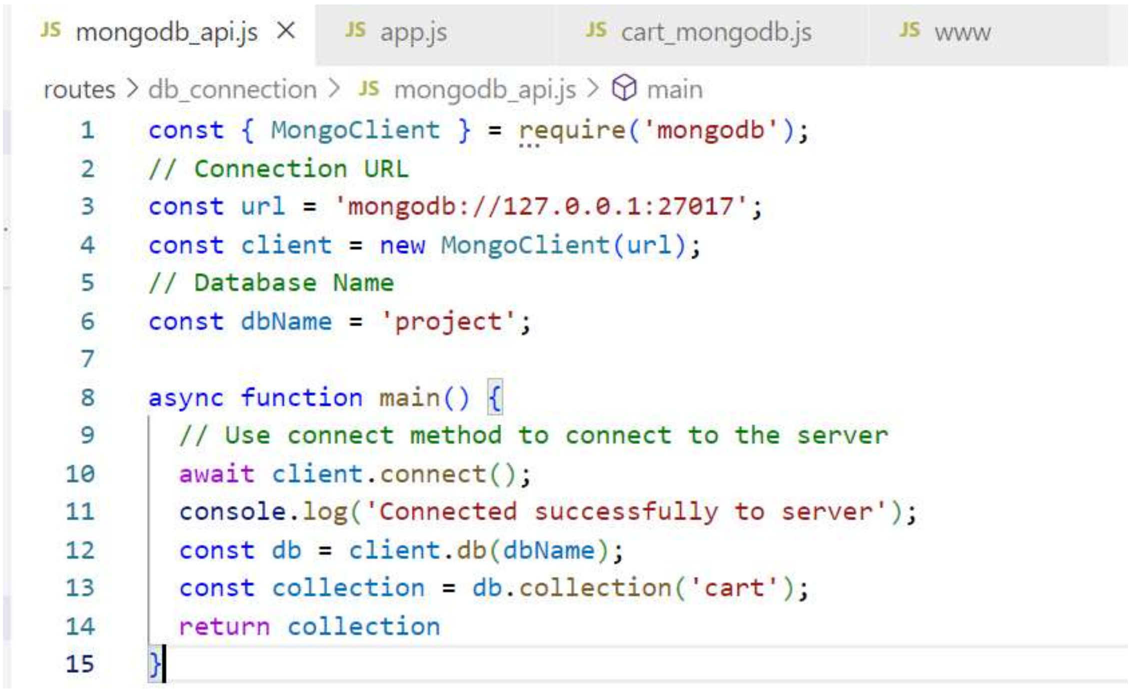
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



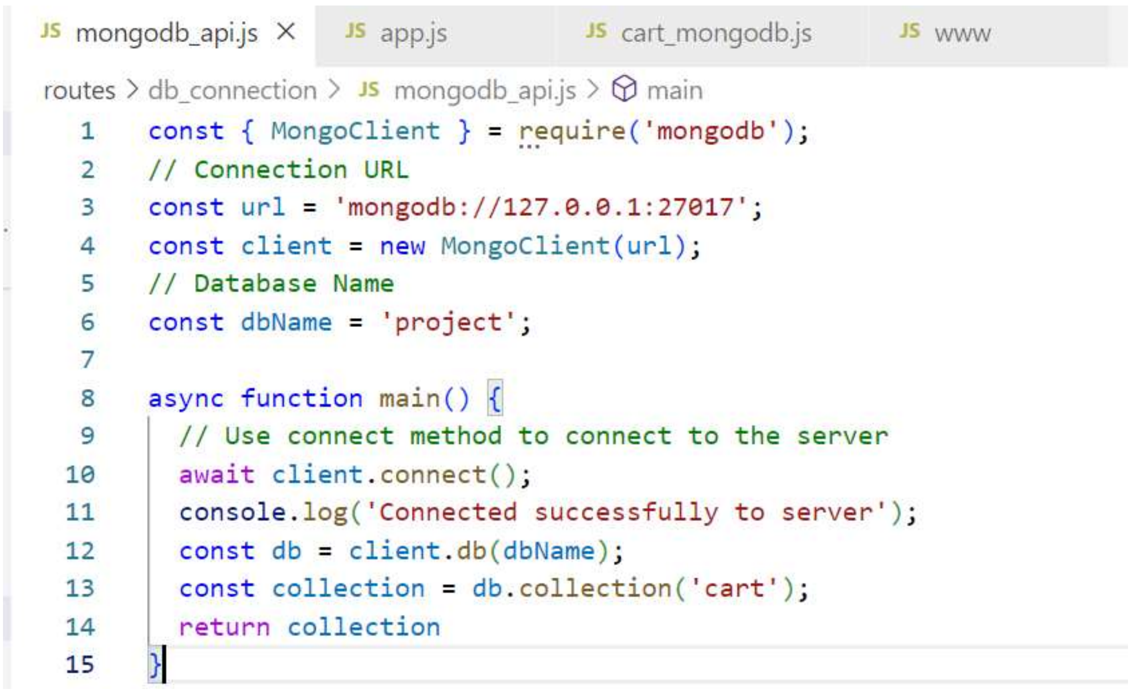
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```



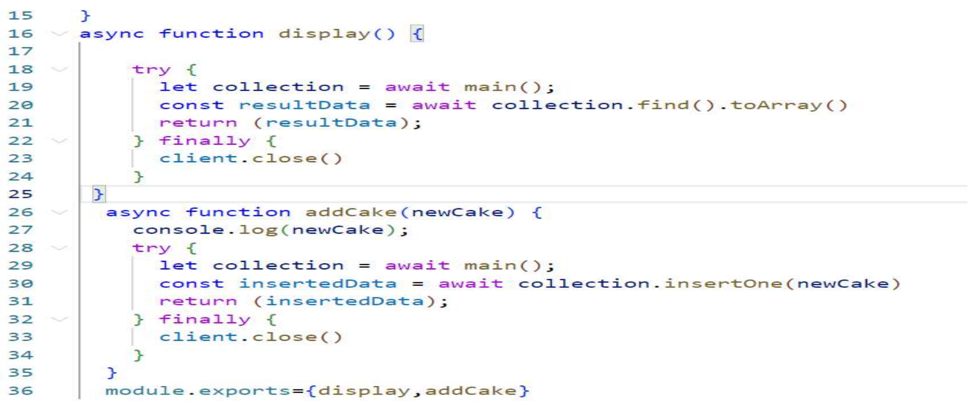
```javascript
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```


mongodb_api.js × app.js cart_mongodb.js www

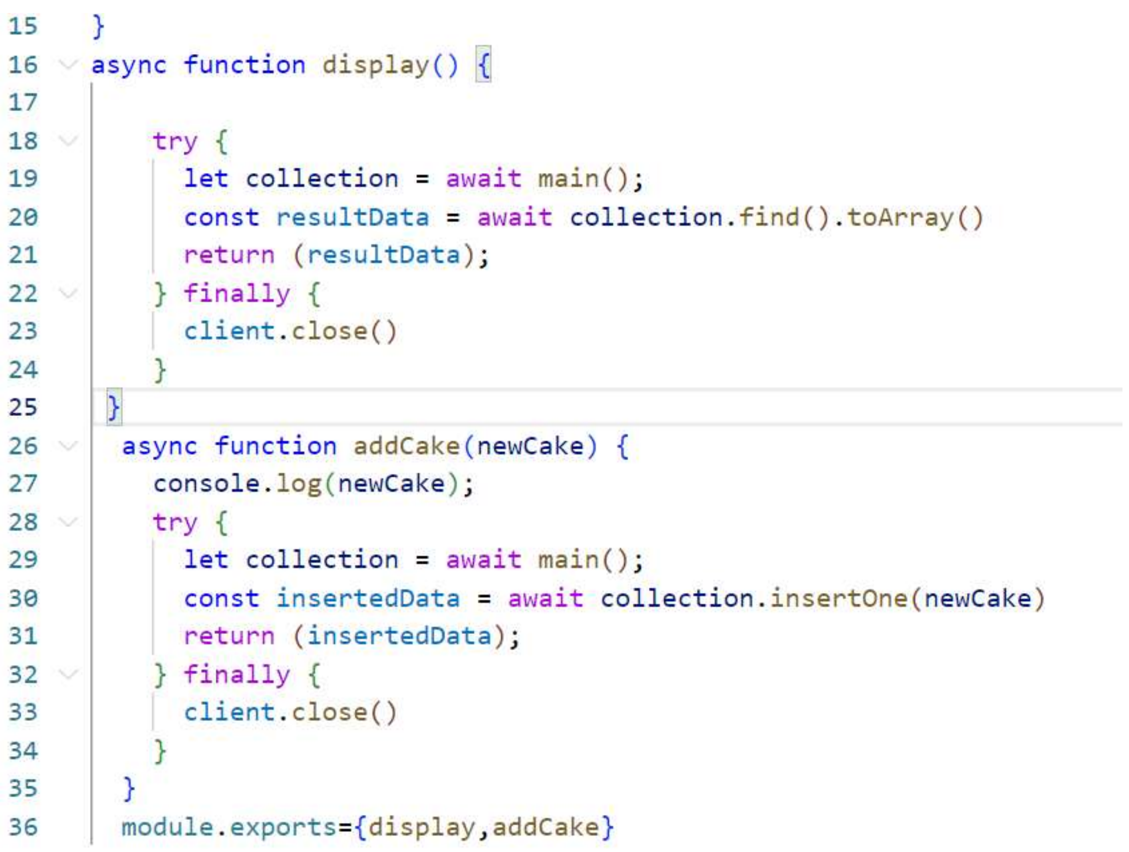
routes > db_connection > mongodb_api.js > main

```
const { MongoClient } = require('mongodb');
// Connection URL
const url = 'mongodb://127.0.0.1:27017';
const client = new MongoClient(url);
// Database Name
const dbName = 'project';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('cart');
  return collection
}
```

b. Add cakes into cart and display cart

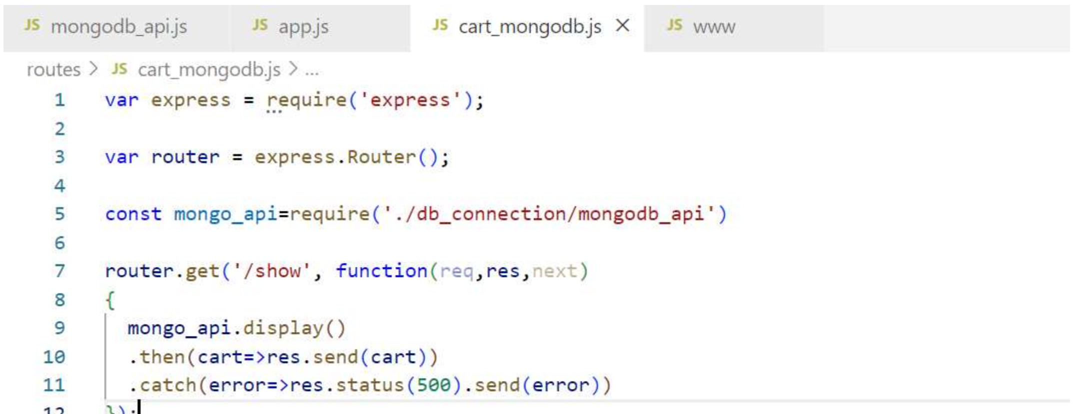


```
}
async function display() {

    try {
      let collection = await main();
      const resultData = await collection.find().toArray()
      return (resultData);
    } finally {
      client.close()
    }
}
 async function addCake(newCake) {
    console.log(newCake);
    try {
      let collection = await main();
      const insertedData = await collection.insertOne(newCake)
      return (insertedData);
    } finally {
      client.close()
    }
 }
 module.exports={display,addCake}
```



```
}
async function display() {

  try {
   let collection = await main();
   const resultData = await collection.find().toArray()
   return (resultData);
  } finally {
   client.close()
  }
}
 async function addCake(newCake) {
  console.log(newCake);
  try {
   let collection = await main();
   const insertedData = await collection.insertOne(newCake)
   return (insertedData);
  } finally {
   client.close()
  }
 }
 module.exports={display,addCake}
```

c. Call the routers

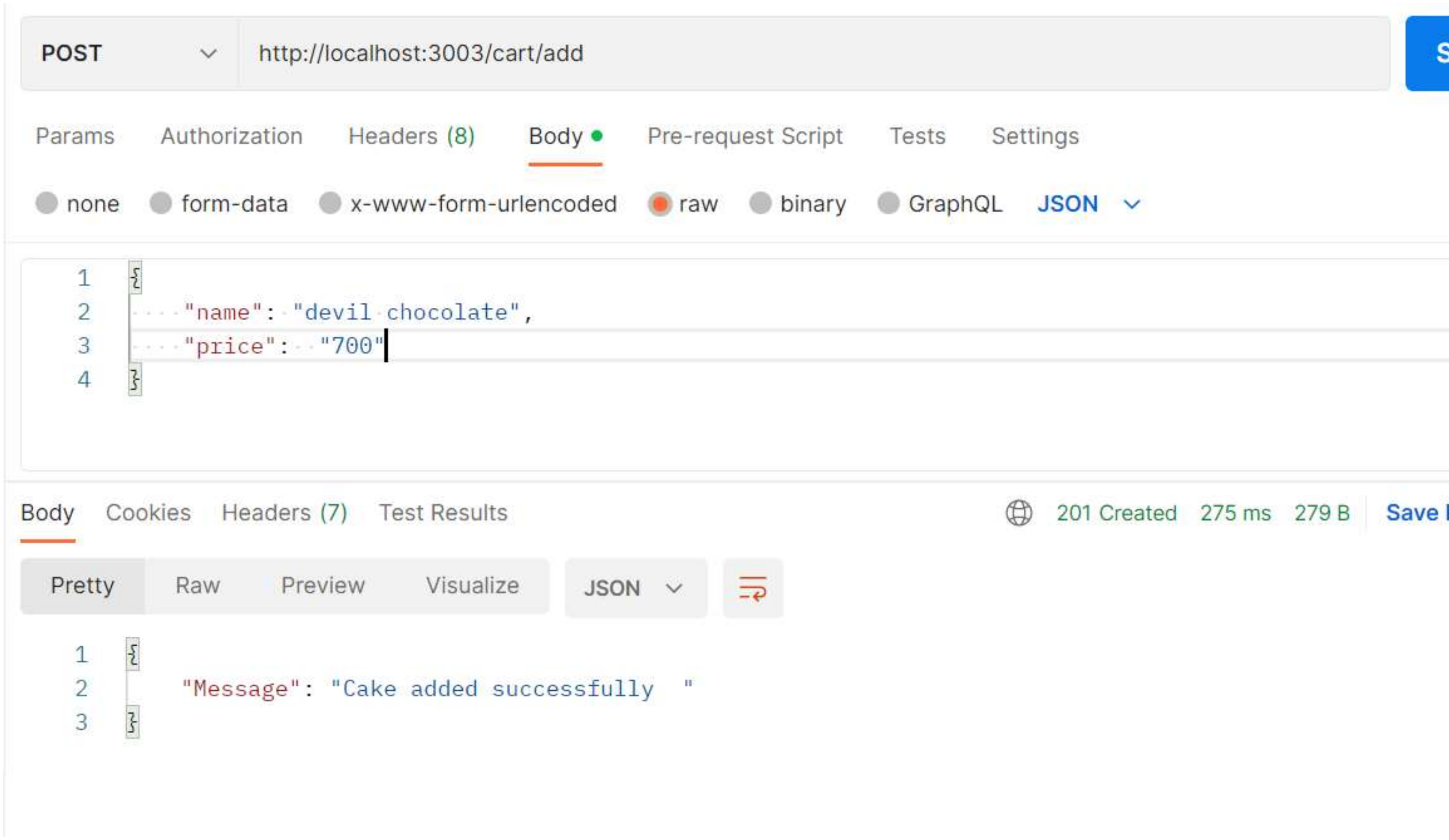


JS mongodb_api.js   JS app.js   JS cart_mongodb.js ×   JS www

routes > JS cart_mongodb.js > ...

```
var express = require('express');

var router = express.Router();

const mongo_api=require('./db_connection/mongodb_api')

router.get('/show', function(req,res,next)
{
  mongo_api.display()
  .then(cart=>res.send(cart))
  .catch(error=>res.status(500).send(error))
});
```

d.result



POST http://localhost:3003/cart/add

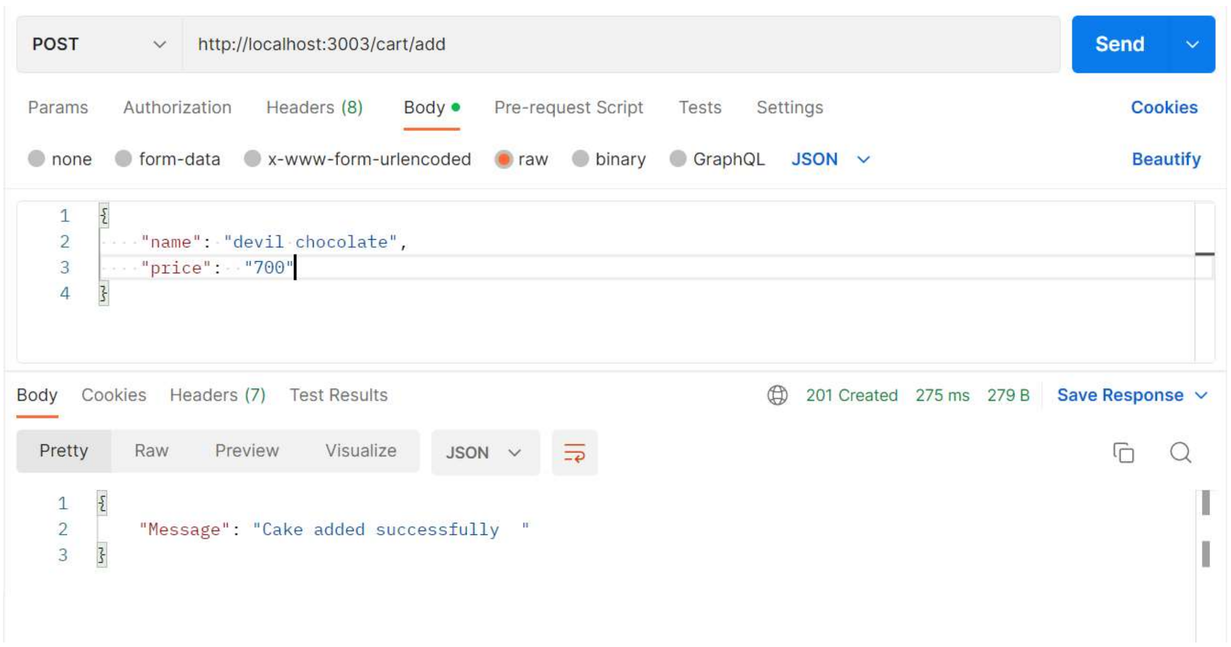
Body: raw JSON

```
{
    "name": "devil chocolate",
    "price":  "700"
}
```

201 Created 275 ms 279 B

JSON

```
{
    "Message": "Cake added successfully  "
}
```


POST http://localhost:3003/cart/add

Send

Body: raw JSON

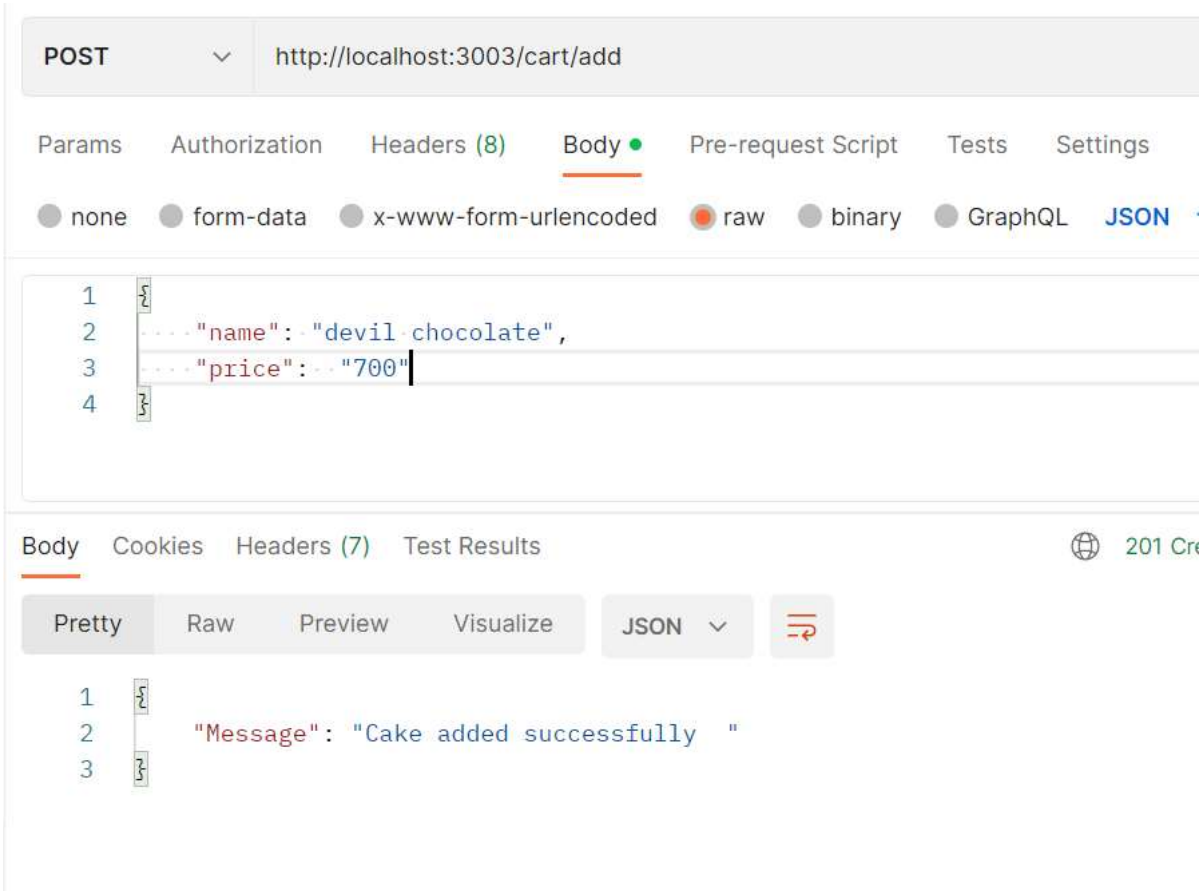
```
{
    "name": "devil chocolate",
    "price":  "700"
}
```

201 Created 275 ms 279 B

Save Response

JSON

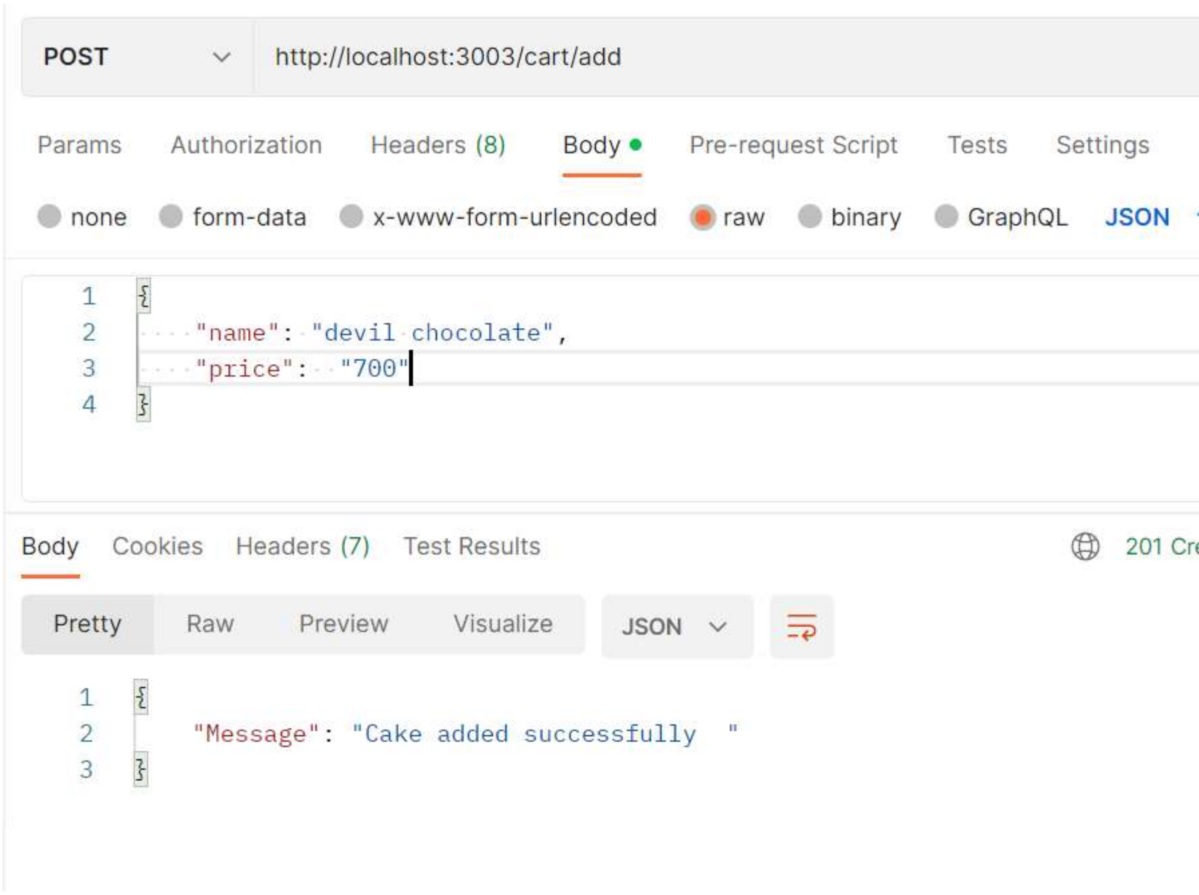
```
{
    "Message": "Cake added successfully  "
}
```



POST http://localhost:3003/cart/add

```
{
    "name": "devil chocolate",
    "price":  "700"
}
```

```
{
    "Message": "Cake added successfully  "
}
```


POST http://localhost:3003/cart/add

Params Authorization Headers (8) Body Pre-request Script Tests Settings

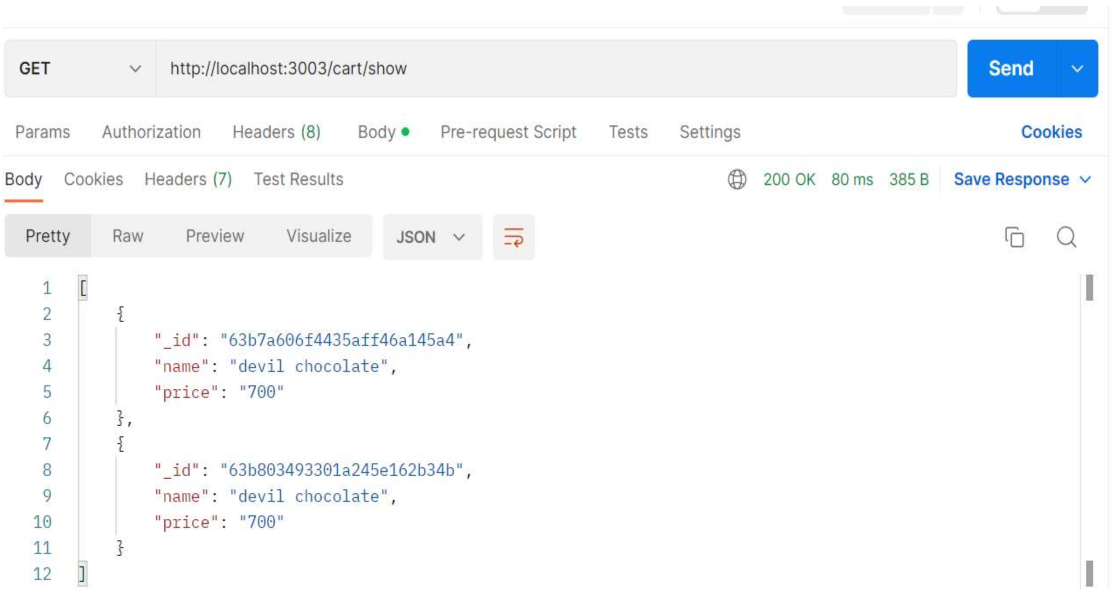
none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "name": "devil chocolate",
    "price":  "700"
}
```

Body Cookies Headers (7) Test Results 201 Cre

Pretty Raw Preview Visualize JSON

```
{
    "Message": "Cake added successfully  "
}
```


GET http://localhost:3003/cart/show Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

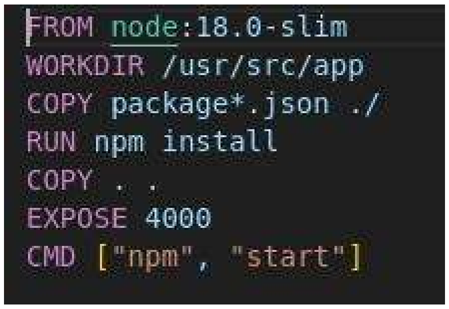
Body Cookies Headers (7) Test Results 200 OK 80 ms 385 B Save Response

Pretty Raw Preview Visualize JSON

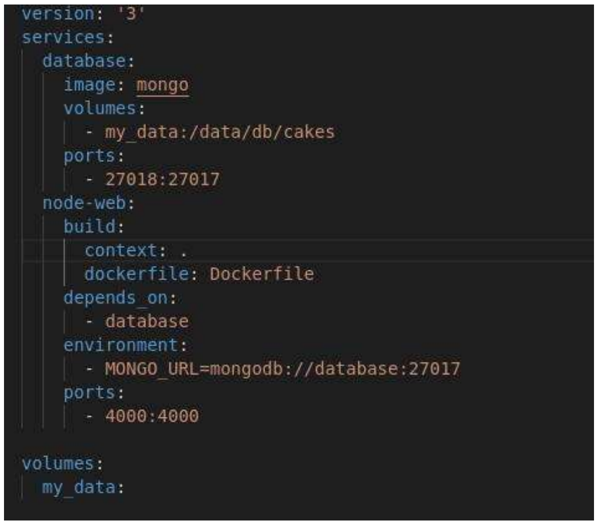
```
[
    {
        "_id": "63b7a606f4435aff46a145a4",
        "name": "devil chocolate",
        "price": "700"
    },
    {
        "_id": "63b803493301a245e162b34b",
        "name": "devil chocolate",
        "price": "700"
    }
]
```

Containerization

1. Create docker and docker-compose file for all:

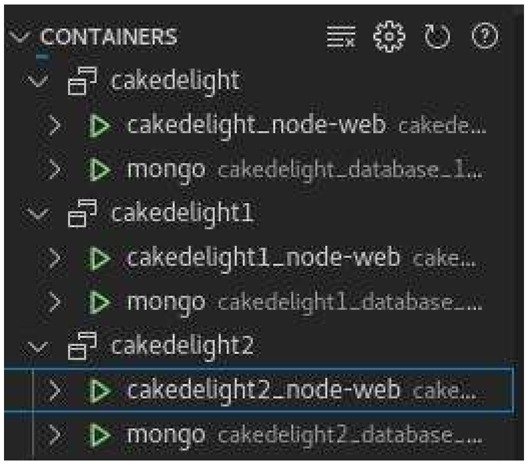


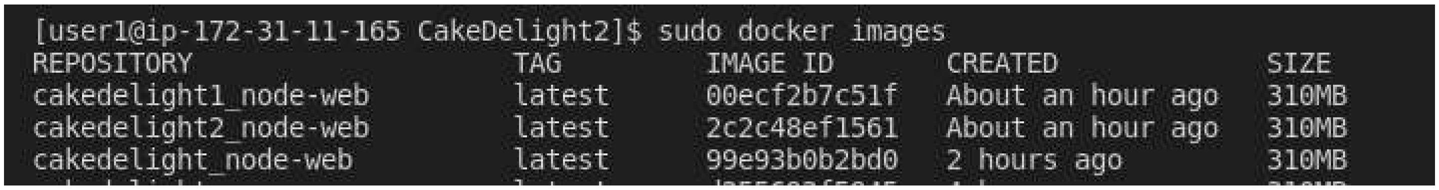
```
FROM node:18.0-slim
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 4000
CMD ["npm", "start"]
```



```
version: '3'
services:
  database:
    image: mongo
    volumes:
      - my_data:/data/db/cakes
    ports:
      - 27018:27017
  node-web:
    build:
      context: .
      dockerfile: Dockerfile
    depends_on:
      - database
    environment:
      - MONGO_URL=mongodb://database:27017
    ports:
      - 4000:4000

volumes:
  my_data:
```

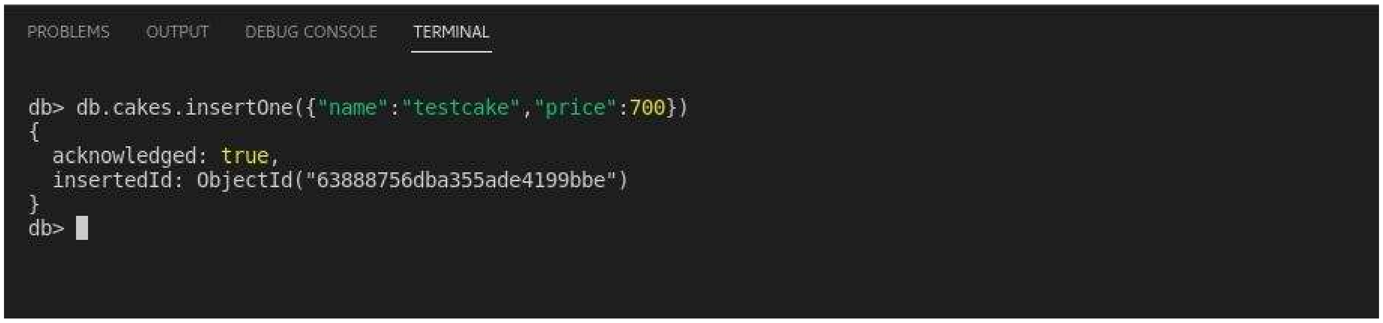
2. Compose up the docker-compose file:


CONTAINERS
cakedelight
cakedelight_node-web cakede...
mongo cakedelight_database_1...
cakedelight1
cakedelight1_node-web cake...
mongo cakedelight1_database_...
cakedelight2
cakedelight2_node-web cake...
mongo cakedelight2_database_...



```
[user1@ip-172-31-11-165 CakeDelight2]$ sudo docker images
REPOSITORY                    TAG          IMAGE ID       CREATED             SIZE
cakedelight1_node-web         latest       00ecf2b7c51f   About an hour ago   310MB
cakedelight2_node-web         latest       2c2c48ef1561   About an hour ago   310MB
cakedelight_node-web          latest       99e93b0b2bd0   2 hours ago         310MB
```

3. Run mongo shell and populate the database:



```
db> db.cakes.insertOne({"name":"testcake","price":700})
{
  acknowledged: true,
  insertedId: ObjectId("63888756dba355ade4199bbe")
}
db>
```