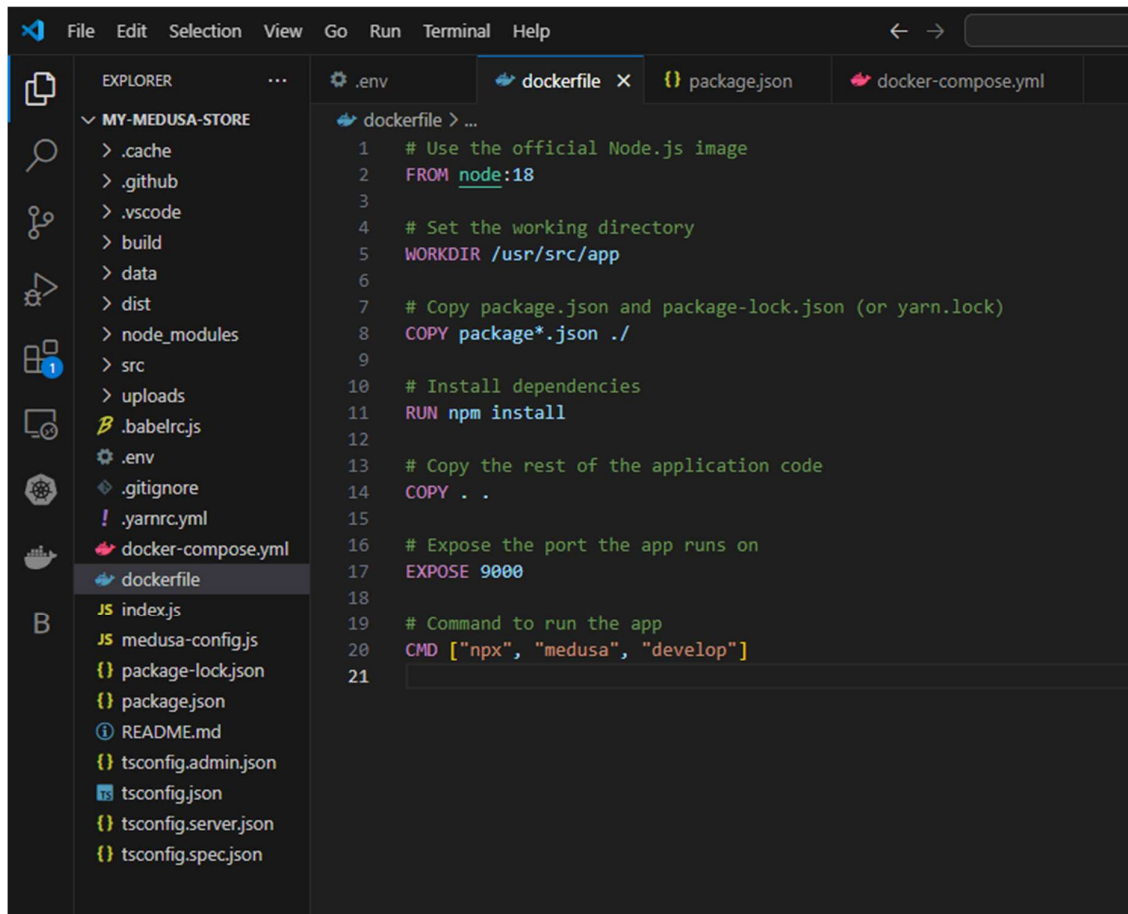


Name : Yash Anil Mutatkar.

1. Create a Dockerfile for Medusa

This file will define the image for your Medusa server.

Create a file named Dockerfile in your Medusa project directory:



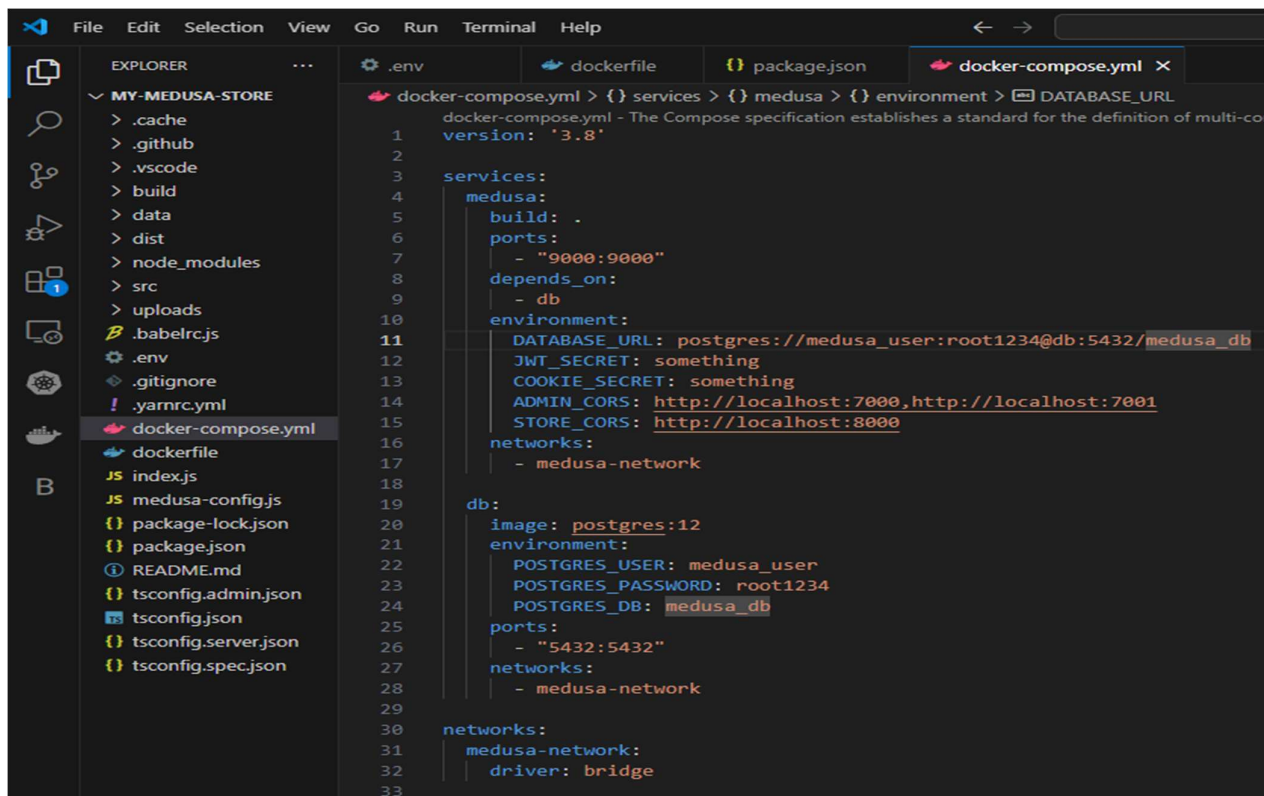
The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the file structure of a project named 'MY-MEDUSA-STORE'. The file 'dockerfile' is selected and highlighted. The main editor area shows the content of the 'dockerfile' file, which is a Dockerfile for a Node.js application. The Dockerfile includes instructions to use the official Node.js image, set the working directory, copy package files, install dependencies, copy the application code, expose port 9000, and run the application using npx.

```
dockerfile > ...
1  # Use the official Node.js image
2  FROM node:18
3
4  # Set the working directory
5  WORKDIR /usr/src/app
6
7  # Copy package.json and package-lock.json (or yarn.lock)
8  COPY package*.json ./
9
10 # Install dependencies
11 RUN npm install
12
13 # Copy the rest of the application code
14 COPY . .
15
16 # Expose the port the app runs on
17 EXPOSE 9000
18
19 # Command to run the app
20 CMD ["npx", "medusa", "develop"]
21
```

2. Create a docker-compose.yml file

This file will define the services for Medusa and PostgreSQL.

Create a file named docker-compose.yml in your project directory:



The screenshot shows a VS Code editor window with the 'docker-compose.yml' file open. The Explorer sidebar on the left shows the project structure for 'MY-MEDUSA-STORE'. The main editor area displays the following YAML configuration:

```
1 version: '3.8'
2
3 services:
4   medusa:
5     build: .
6     ports:
7       - "9000:9000"
8     depends_on:
9       - db
10    environment:
11      DATABASE_URL: postgres://medusa_user:root1234@db:5432/medusa_db
12      JWT_SECRET: something
13      COOKIE_SECRET: something
14      ADMIN_CORS: http://localhost:7000,http://localhost:7001
15      STORE_CORS: http://localhost:8000
16    networks:
17      - medusa-network
18
19 db:
20   image: postgres:12
21   environment:
22     POSTGRES_USER: medusa_user
23     POSTGRES_PASSWORD: root1234
24     POSTGRES_DB: medusa_db
25   ports:
26     - "5432:5432"
27   networks:
28     - medusa-network
29
30 networks:
31   medusa-network:
32     driver: bridge
```

3. Build and Run Docker Containers

Run the following commands from your project directory:

1. Build the Docker images:
docker-compose build
2. Start the containers:
docker-compose up

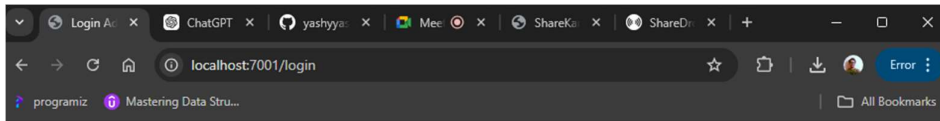
4. Run Migrations

Once your containers are up and running, you need to run Medusa migrations. Open a new terminal and run:

```
docker-compose run --rm medusa npm run migrate
```

5. Access Medusa

You should now be able to access your Medusa server at <http://localhost:7001/login>



Log in to Medusa

Continue

[Forgot your password?](#)

Conclusion

- **Dockerfile:** Defines how to build the Medusa image.
- **docker-compose.yml:** Defines the Medusa and PostgreSQL services and their configuration.
- **Build and Run:** Use docker-compose build and docker-compose up to start the application.