LAB EXPERIMENT 3

Create a simple web application and set up a multi container environment using Docker compose to run the application alongside a database container

Step 1: Set Up Project Structure

1. Create a Project Directory

mkdir multi-container-app

cd multi-container-app

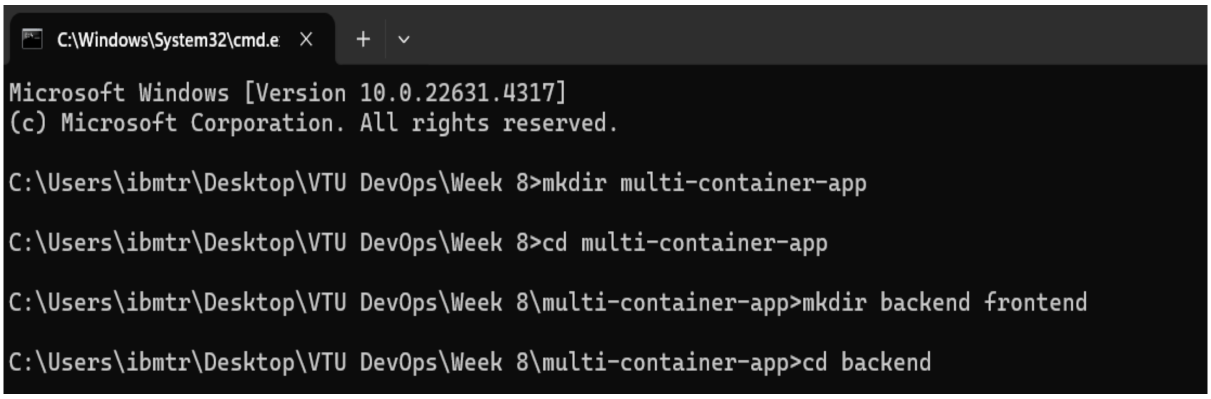
2. Define Folder Structure Create subdirectories for the frontend, backend, and database configurations.

mkdir backend frontend

Step 2: Build the Backend Application

1. Navigate to Backend Folder

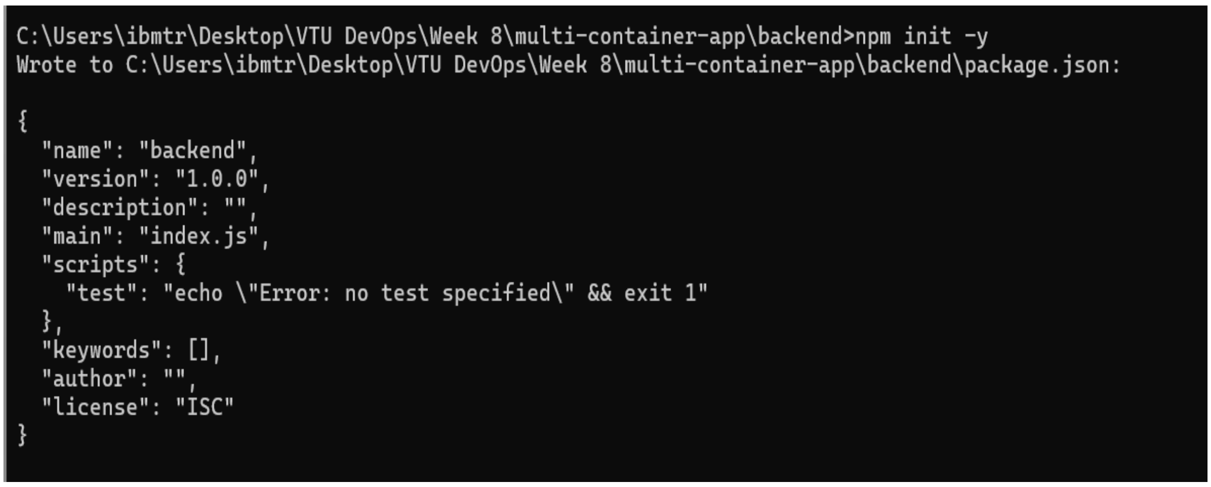
cd backend



```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8>mkdir multi-container-app

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8>cd multi-container-app

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app>mkdir backend frontend

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app>cd backend
```

2. **Initialize Node.js Project**

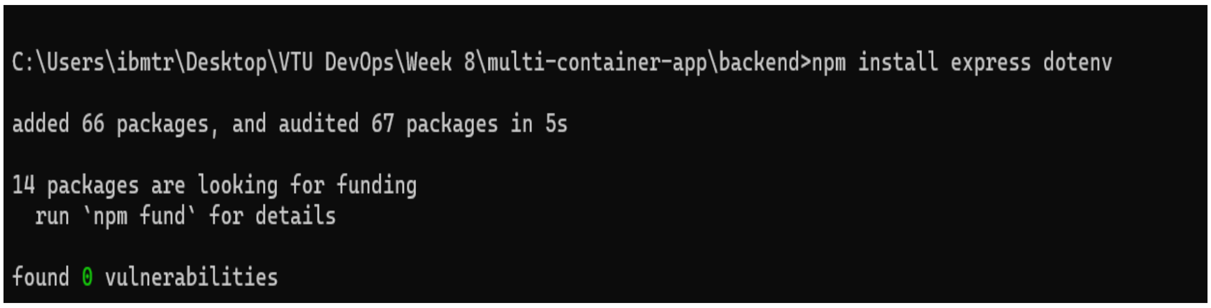
npm init -y



```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>npm init -y
Wrote to C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend\package.json:

{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

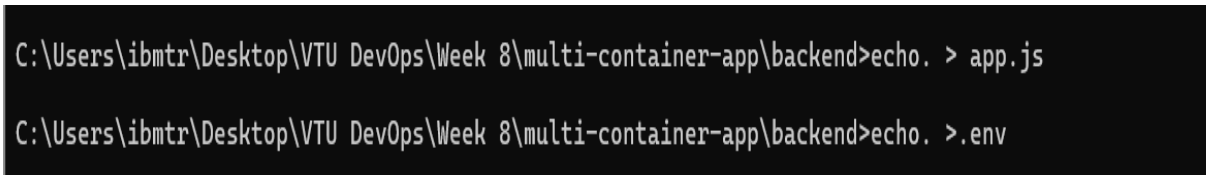
3. **Install Dependencies**
   Install Express, dotenv, and any other dependencies you need.

npm install express dotenv



```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>npm install express dotenv

added 66 packages, and audited 67 packages in 5s

14 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

4. **Create Main Backend Files**
   - Create app.js and .env files in the backend folder.



```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>echo. > app.js

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>echo. >.env
```

5. **Configure Backend Server**
   In app.js, set up a basic Express server:

```
const express = require('express');
const mongoose = require('mongoose');
require('dotenv').config();
const app = express();
const PORT = process.env.PORT || 5000;
// Connect to MongoDB
mongoose.connect(process.env.MONGO_URI, {
   useNewUrlParser: true,
   useUnifiedTopology: true
}).then(() => console.log("Connected to MongoDB"))
  .catch(err => console.error(err));
app.get('/', (req, res) => res.send('Hello from Backend!'));
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

6. **Set Up Environment Variables** In .env, configure the database connection:

```
PORT=5000

MONGO_URI=mongodb://mongo:27017/mydb
```

7. **Create Dockerfile in Backend**

```
FROM node:16

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 5000

CMD ["node", "app.js"]
```

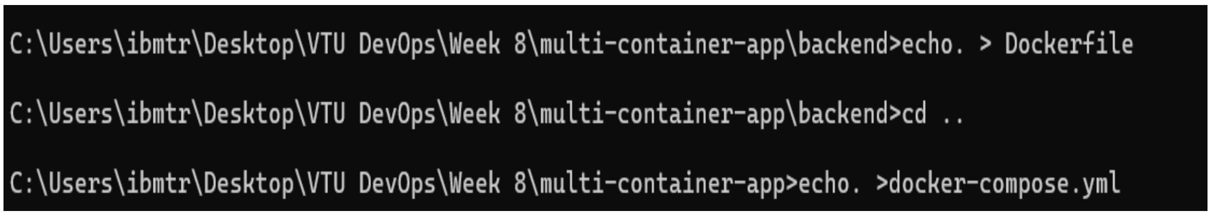
Step 3: Build the Database Container (MongoDB)

No additional setup is required for MongoDB since Docker will pull the image. The Docker Compose file will define MongoDB as a service.

Step 4: Set Up Docker Compose File

1. **Navigate to Project Root**
   Go back to the root directory.

cd ..



```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>echo. > Dockerfile

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>cd ..

C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app>echo. >docker-compose.yml
```

2. **Create docker-compose.yml File** Define services for both the backend and MongoDB in this file:

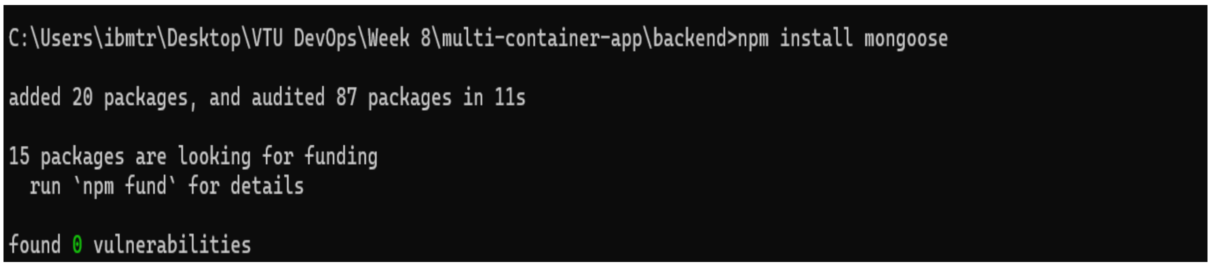
```
version: '3.8'

services:

 backend:

  build: ./backend

  ports:

   - "5000:5000"

  environment:
      - MONGO_URI=mongodb://mongo:27017/mydb
    depends_on:
      - mongo
  mongo:
    image: mongo:latest
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db
volumes:
  mongo-data:
```

Step 5: Install Mongoose in the Backend

1. In the backend folder, install mongoose locally.

npm install mongoose

2. Verify that mongoose is listed in the dependencies section of your package.json file in the backend folder.

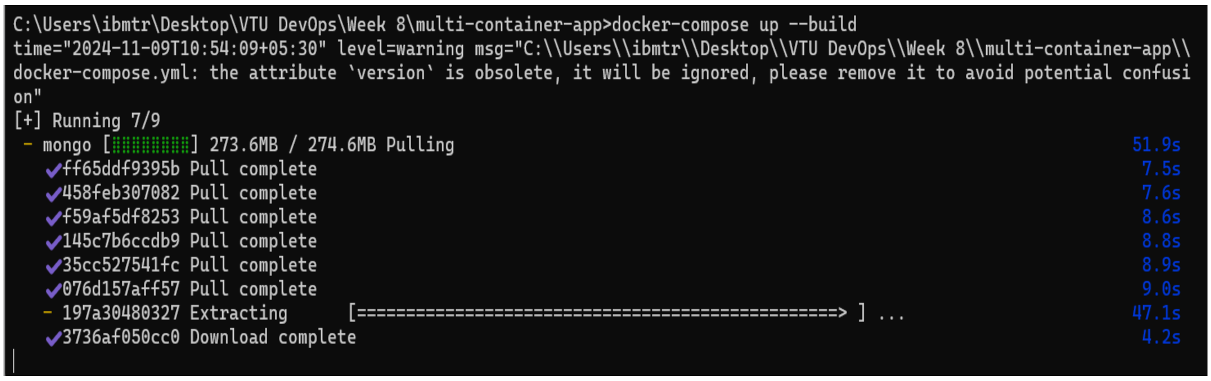


```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app\backend>npm install mongoose

added 20 packages, and audited 87 packages in 11s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Step 6: Build and Run the Containers

1. **Run Docker Compose** In the root directory, start Docker Compose:

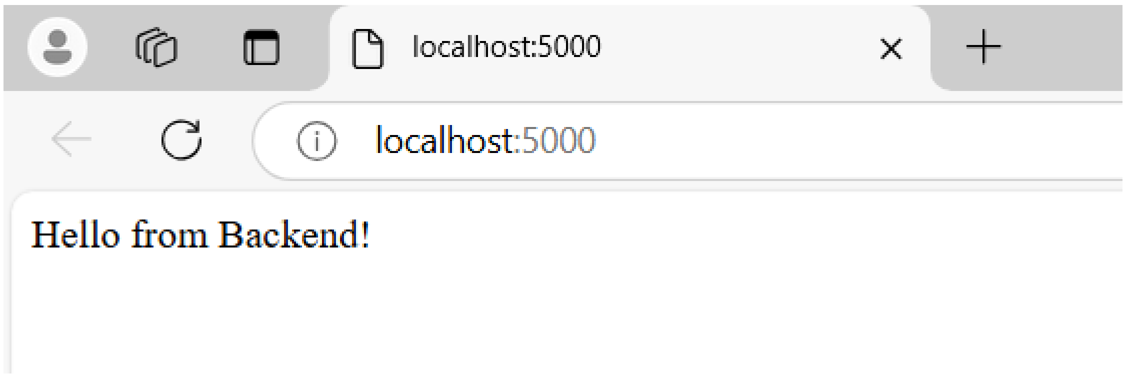
docker-compose up –build



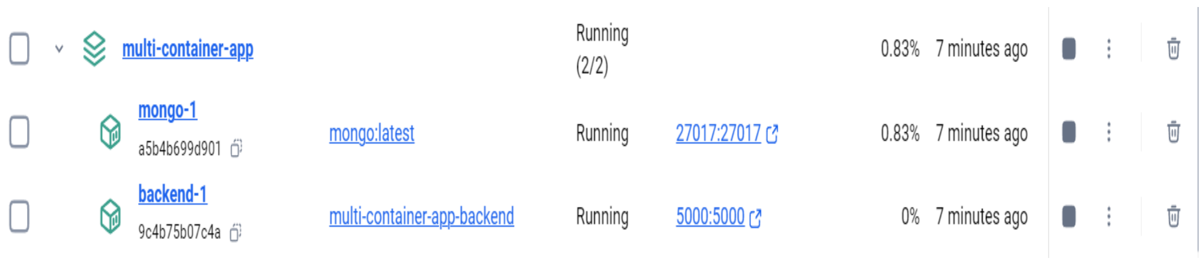
```
C:\Users\ibmtr\Desktop\VTU DevOps\Week 8\multi-container-app>docker-compose up --build
time="2024-11-09T10:54:09+05:30" level=warning msg="C:\\Users\\ibmtr\\Desktop\\VTU DevOps\\Week 8\\multi-container-app\\
docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusi
on"
[+] Running 7/9
 - mongo [⣿⣿⣿⣿⣿⣿⣿⣿⣿] 273.6MB / 274.6MB Pulling                                                              51.9s
   ✔ff65ddf9395b Pull complete                                                                               7.5s
   ✔458feb307082 Pull complete                                                                               7.6s
   ✔f59af5df8253 Pull complete                                                                               8.6s
   ✔145c7b6ccdb9 Pull complete                                                                               8.8s
   ✔35cc527541fc Pull complete                                                                               8.9s
   ✔076d157aff57 Pull complete                                                                               9.0s
   - 197a30480327 Extracting      [==================================================> ] ...              47.1s
   ✔3736af050cc0 Download complete                                                                           4.2s
```

Docker Compose will build the backend image, start the backend and MongoDB services, and link them together.

2. **Verify Setup**
   - Visit http://localhost:5000 to see the backend response.
   - Confirm that MongoDB is running by connecting to mongodb://localhost:27017.



Hello from Backend!



multi-container-app		Running (2/2)		0.83%	7 minutes ago	
mongo-1 a5b4b699d901	mongo:latest	Running	27017:27017	0.83%	7 minutes ago	
backend-1 9c4b75b07c4a	multi-container-app-backend	Running	5000:5000	0%	7 minutes ago	