Assignment No. 6

Deploy a User Registration and Login Web Page using Docker Containers

1. Aim

To deploy a complete web application consisting of a user registration and login page along with a database service, all running inside Docker containers.

2. Objectives

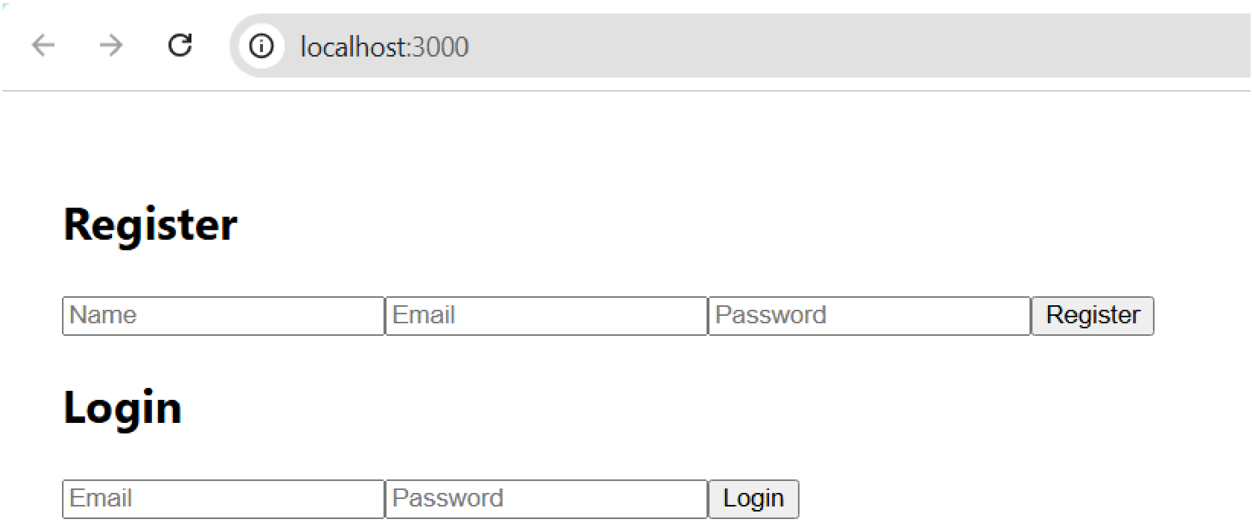
- Deploy a **frontend web page** using Docker.
- Deploy a **database service** using Docker.
- Integrate both services through a **backend API** to provide user registration and login functionality.

3. Tools and Technologies

- **Docker Desktop (Windows)**
- **Docker Compose**
- **Node.js + Express** (backend)
- **MongoDB 6.0** (database)

Browser → frontend→ Express (backend) → MongoDB (database)

Frontend: Static HTML

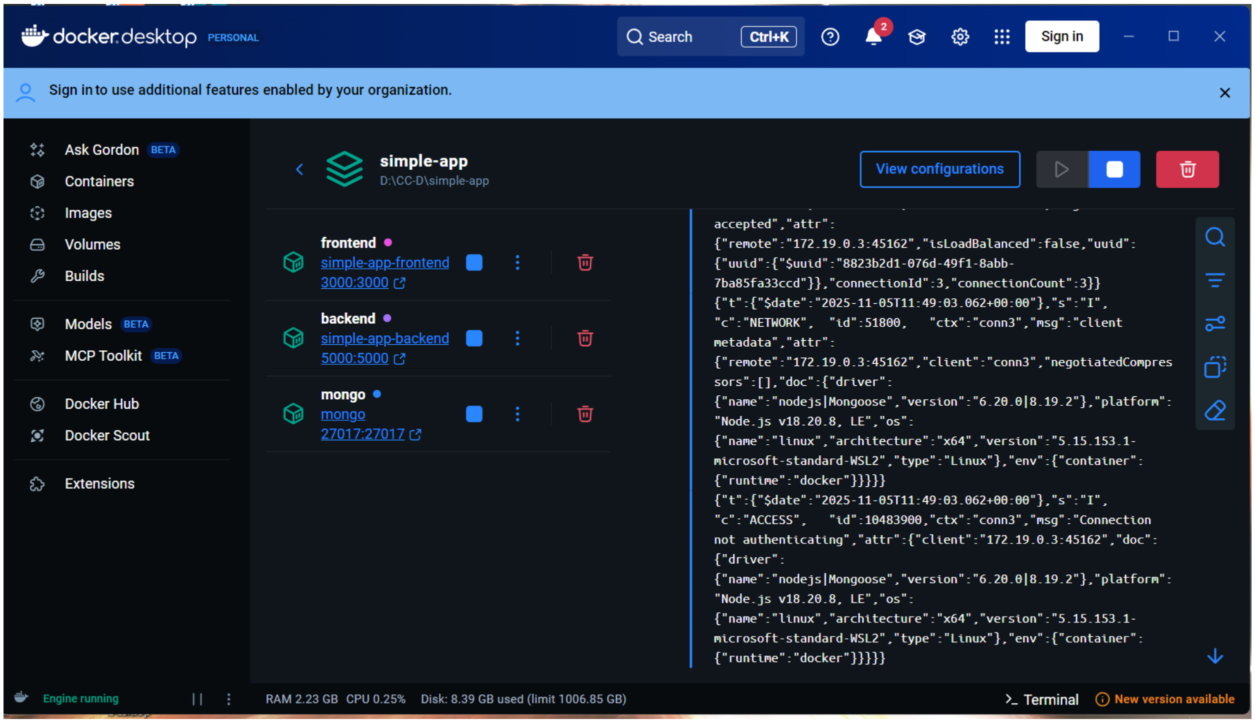


localhost:3000

Register

Login

Networking: All services connected through a single Docker network created by Docker Compose.



frontend	simple-app-frontend	3000:3000
backend	simple-app-backend	5000:5000
mongo	mongo	27017:27017

```
accepted","attr":
{"remote":"172.19.0.3:45162","isLoadBalanced":false,"uuid":
{"uuid":{"$uuid":"8823b2d1-076d-49f1-8abb-
7ba85fa33ccd"}},"connectionId":3,"connectionCount":3}}
{"t":{"$date":"2025-11-05T11:49:03.062+00:00"},"s":"I",
"c":"NETWORK",  "id":51800,   "ctx":"conn3","msg":"client
metadata","attr":
{"remote":"172.19.0.3:45162","client":"conn3","negotiatedCompres
sors":[],"doc":{"driver":
{"name":"nodejs|Mongoose","version":"6.20.0|8.19.2"},"platform":
"Node.js v18.20.8, LE","os":
{"name":"linux","architecture":"x64","version":"5.15.153.1-
microsoft-standard-WSL2","type":"Linux"},"env":{"container":
{"runtime":"docker"}}}}}
{"t":{"$date":"2025-11-05T11:49:03.062+00:00"},"s":"I",
"c":"ACCESS",   "id":10483900,"ctx":"conn3","msg":"Connection
not authenticating","attr":{"client":"172.19.0.3:45162","doc":
{"driver":
{"name":"nodejs|Mongoose","version":"6.20.0|8.19.2"},"platform":
"Node.js v18.20.8, LE","os":
{"name":"linux","architecture":"x64","version":"5.15.153.1-
microsoft-standard-WSL2","type":"Linux"},"env":{"container":
{"runtime":"docker"}}}}}
```

RAM 2.23 GB CPU 0.25% Disk: 8.39 GB used (limit 1006.85 GB)

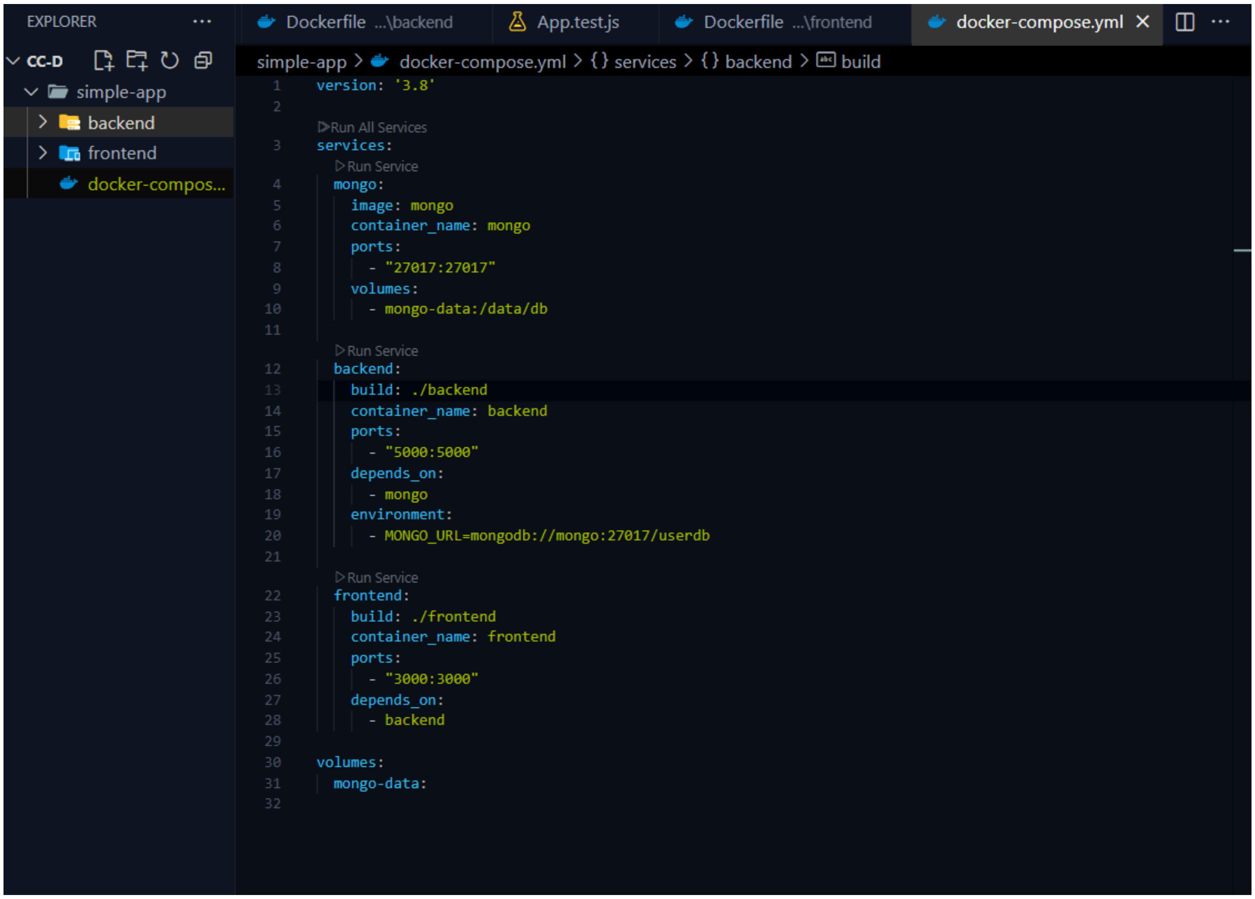
5. Configuration Files

docker-compose.yml

- Defines 3 services: `frontend`, `backend`, `mongo`
- Maps ports:
  - `8080:80` → Frontend
  - `5000:5000` → Backend

    - `27017:27017` → MongoDB

- Uses environment variables from `.env` for credentials and database name.



```yaml
version: '3.8'

services:
  mongo:
    image: mongo
    container_name: mongo
    ports:
      - "27017:27017"
    volumes:
      - mongo-data:/data/db

  backend:
    build: ./backend
    container_name: backend
    ports:
      - "5000:5000"
    depends_on:
      - mongo
    environment:
      - MONGO_URL=mongodb://mongo:27017/userdb

  frontend:
    build: ./frontend
    container_name: frontend
    ports:
      - "3000:3000"
    depends_on:
      - backend

volumes:
  mongo-data:
```

6. Procedure

1. **Create project folder** `Simple-app` with sub-folders `backend` and `frontend`.

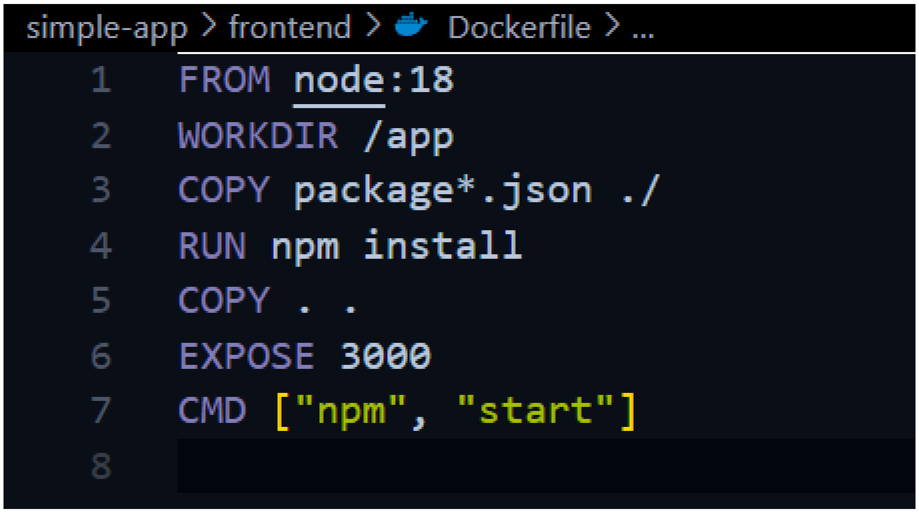
2. **Write the backend code** for registration & login using Express and MongoDB.

3. **Add Dockerfile** inside backend folder.

4. **Write docker-compose.yml** to connect backend, frontend, and database services.

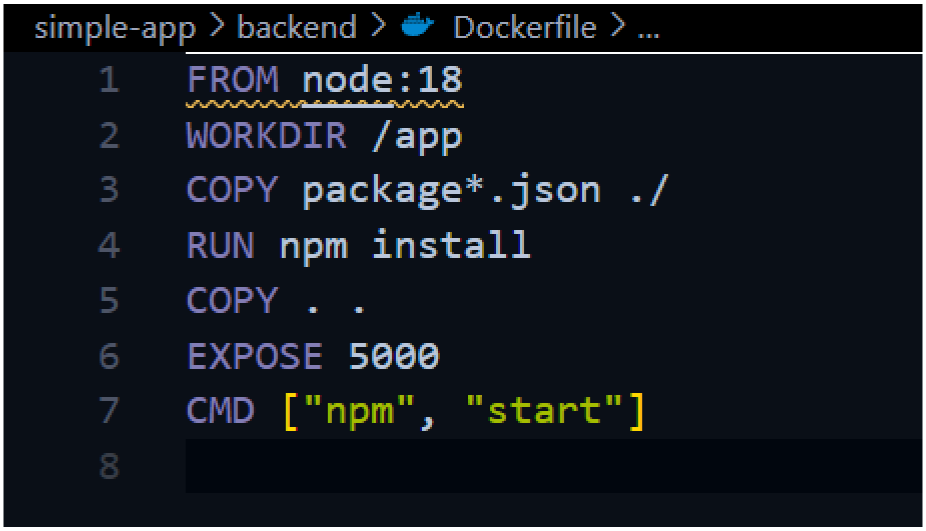
5. **Create separate Dockerfiles for both frontend and backend**

   -frontend Dockerfile



```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

   -backend Dockerfile



```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD ["npm", "start"]
```

6. **Run the project:**

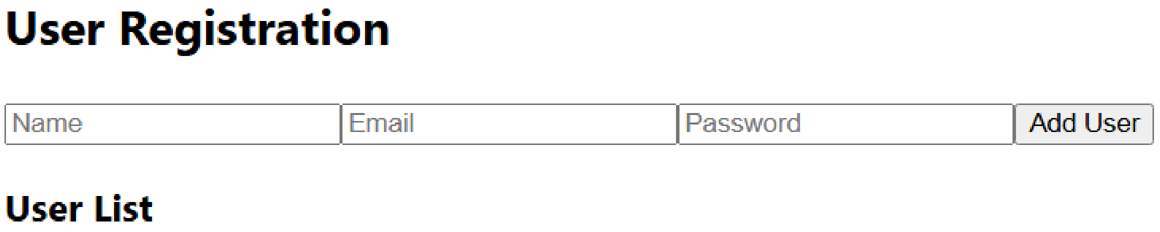
docker compose up --build

Access services:

- Frontend → http://localhost:3000

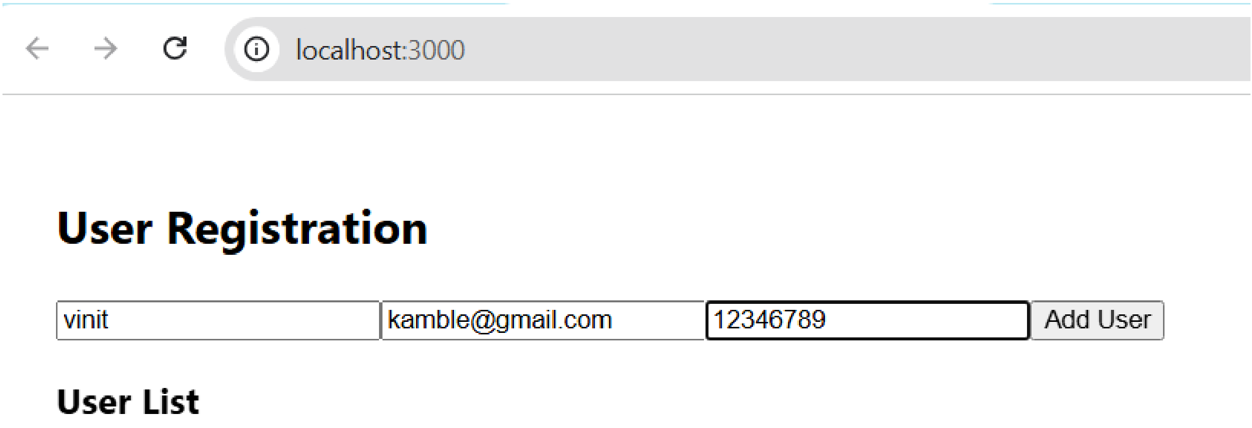
- Backend → http://localhost:5000

- Database → MongoDB Compass at localhost:27017



User Registration

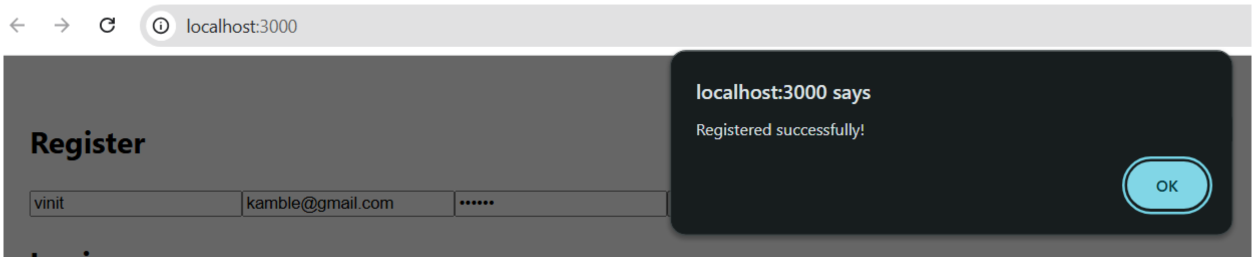
User List



User Registration

vinit | kamble@gmail.com | 12346789 | Add User

User List

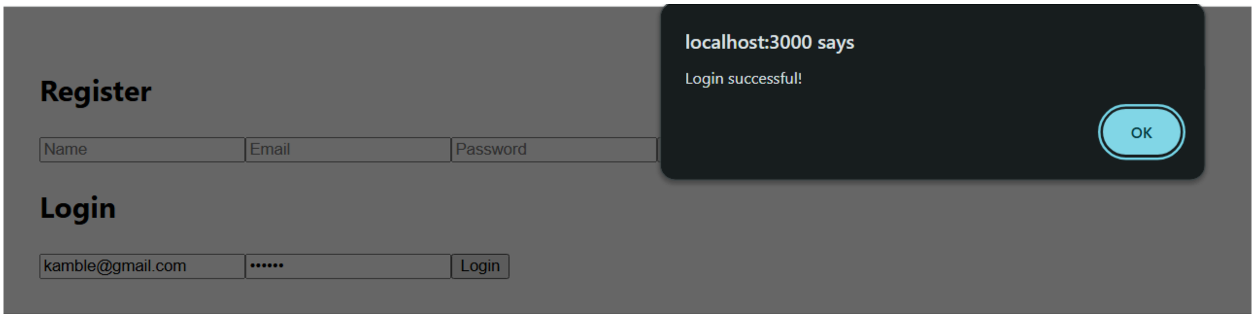


Register

vinit | kamble@gmail.com | ••••••

localhost:3000 says

Registered successfully!

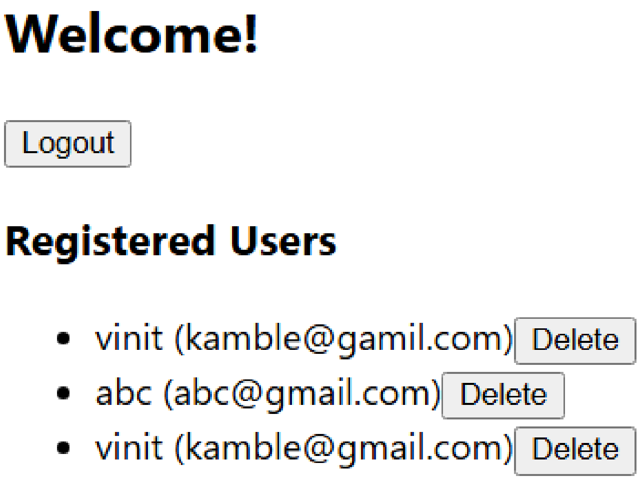

Register

Login

kamble@gmail.com | •••••• | Login

localhost:3000 says

Login successful!



Welcome!

Registered Users

- vinit (kamble@gamil.com)
- abc (abc@gmail.com)
- vinit (kamble@gmail.com)