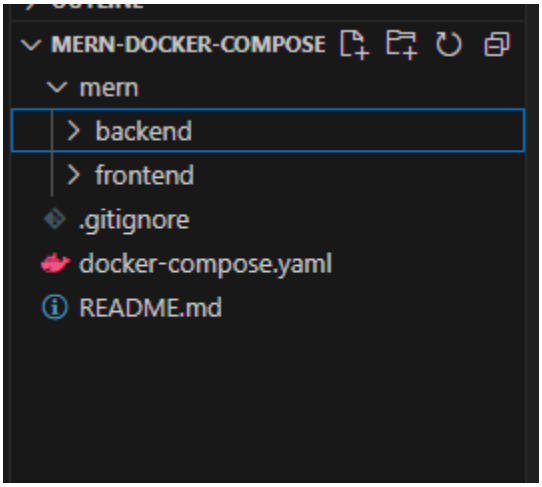
MERN PROJECT DEPLOYMENT USING DOCKER

STEP 1:

File structure:



STEP 2:

Now we have to create a docker file for the front-end and containerize the front-end.

```
FROM node:18.9.1

WORKDIR /app

COPY package.json .

RUN npm install

EXPOSE 5173

COPY . .

CMD ["npm", "run", "dev"]
```

STEP 3:

Now build the docker file which we have created

Docker build –t mern-fornt .

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\frontend>docker build -t mern-front .
[+] Building 24.0s (4/9)                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                        1.7s
 => => transferring dockerfile: 687B                                                        0.3s
 => [internal] load metadata for docker.io/library/node:18.9.1                              9.9s
 => [internal] load .dockerignore                                                           0.9s
 => => transferring context: 2B                                                             0.0s
 => [1/5] FROM docker.io/library/node:18.9.1@sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502eb5e6c45088ce  11.0s
 => => resolve docker.io/library/node:18.9.1@sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502eb5e6c45088ce6  1.2s
 => => sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502eb5e6c45088ce68e8 1.21kB / 1.21kB  0.0s
 => => sha256:48c6c0054de0a80275f9ff67010ab7b555848e126bc35893bba8a77bf871550a 2.21kB / 2.21kB  0.0s
 => => sha256:5eb5251ba1b16c9549a23229d4fb1211e2a7f4ca6ce7b91fb86d939fd886440e 7.74kB / 7.74kB  0.0s
```

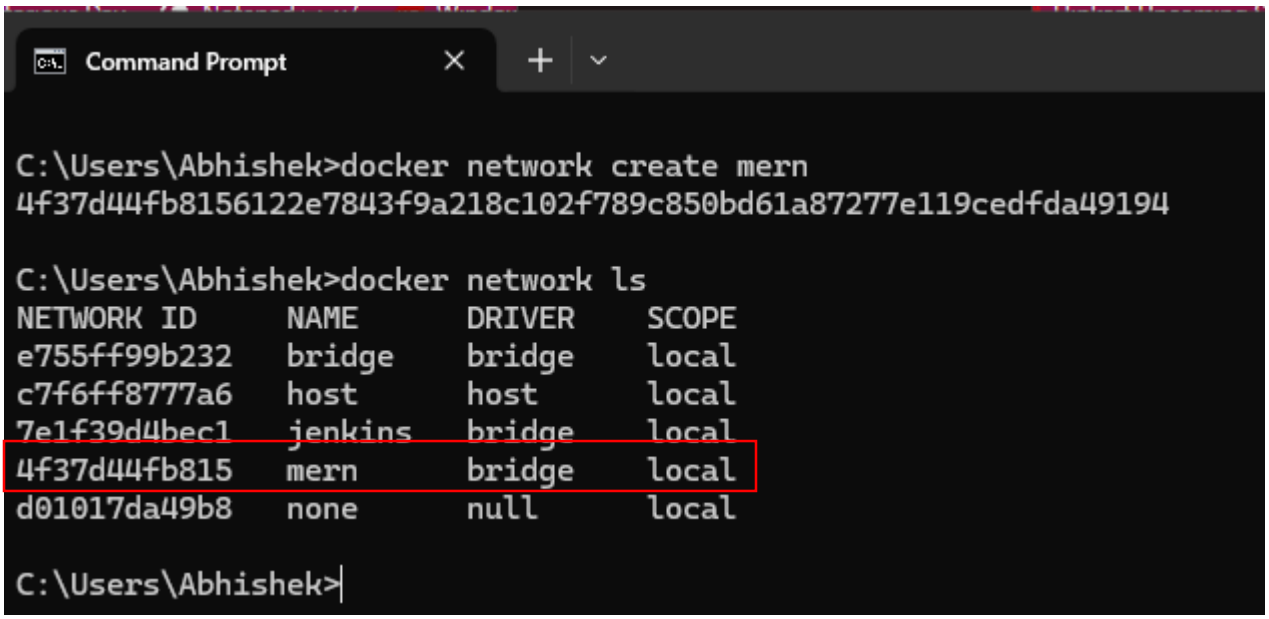
```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\frontend>docker images
REPOSITORY                  TAG         IMAGE ID       CREATED          SIZE
mern-front                  latest      da14db5f471b   38 minutes ago   1.82GB
<none>                      <none>      543174972357   39 hours ago     1.05GB
jenkins/jenkins             latest      c60c715424d9   4 days ago       472MB
jenkins/jenkins             lts-jdk17   58550a76f713   3 weeks ago      472MB
bash                        latest      0d8232586933   2 months ago     14.4MB
gcr.io/k8s-minikube/kicbase v0.0.45     aeed0e1d4642   2 months ago     1.28GB
```

STEP 4:

After building we have to create a network where the front-end and back-end will be able to interact with each other it is important to create a common network for the container to interact with each other.

Docker network create mern

Docker network ls

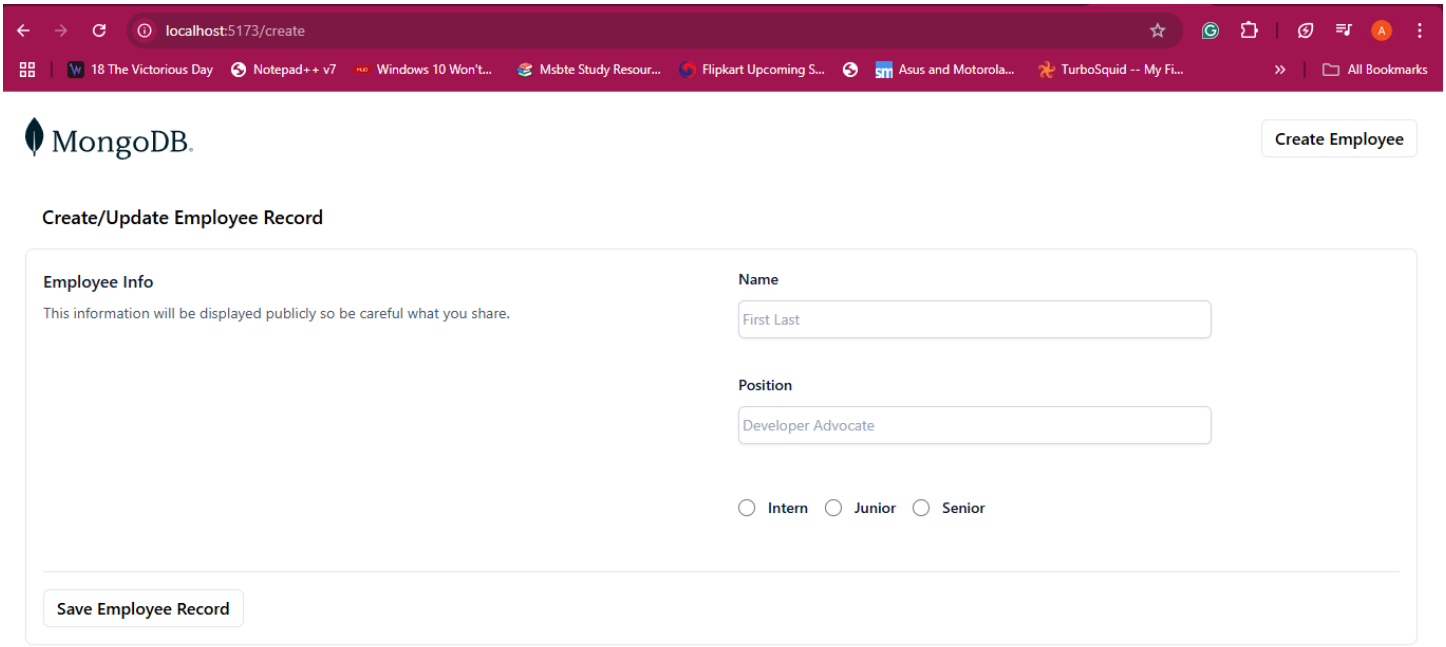


When ever we create network with define network name it will take default network which is bridge.

STEP 5:

Docker run –name=frontend –network=mern –d –p 5173:5173 mern-front

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\frontend>docker run --name=frontend --network=mern -d -p 5173:5173 mern-front
9d3482021a95c8848fbe1fad0e926c64840866e982fdfd6b62ce3bb3d455bd5b
```



STEP 6:

But if we try to add data it will not work because the backend container is not created.

docker run --network=mern --name mongodb -d -p 27017:27017 -v ~/opt/data:/data/db mongo:latest

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\frontend>docker run --network=mern --name mongodb -d -p 27017:27017 -v ~/opt/data:/data/db mongo:latest
342c588d604446401965ed3abd79826a084374cd3c1c83c6477f9d216e04a22f
```

STEP 7:

Now we have to create a docker file for the back-end and containerize the front-end.

```
FROM node:18.9.1

WORKDIR /app

COPY package.json .

RUN npm install

COPY . .
```

```
EXPOSE 5050

CMD ["npm", "start"]
```

STEP 8:

Now it’s time to build the backend container.

Docker build –t mern-backend

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\backend>docker build -t mern-backend  .
[+] Building 42.3s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 168B
 => [internal] load metadata for docker.io/library/node:18.9.1
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/node:18.9.1@sha256:d6ed353d022f6313aa7c3f3df69f3a216f1c9f8c3374502
 => [internal] load build context
 => => transferring context: 35.98kB
 => CACHED [2/5] WORKDIR /app
 => [3/5] COPY package.json .
 => [4/5] RUN npm install
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:e9d7728636b20429c95ab06a8e20058d720efa7df39473b4bdd3910d87f6ab25
 => => naming to docker.io/library/mern-backend

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/bx8y3901c2iozz6kqc

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview

X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\backend>
```

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\backend>docker images
REPOSITORY                TAG         IMAGE ID       CREATED          SIZE
mern-backend              latest      e9d7728636b2   2 minutes ago    1.01GB
mern-front                latest      da14db5f471b   2 hours ago      1.82GB
<none>                    <none>      543174972357   40 hours ago     1.05GB
jenkins/jenkins           latest      c60c715424d9   4 days ago       472MB
jenkins/jenkins           lts-jdk17   58550a76f713   3 weeks ago      472MB
mongo                     latest      df3f01eba940   4 weeks ago      854MB
```

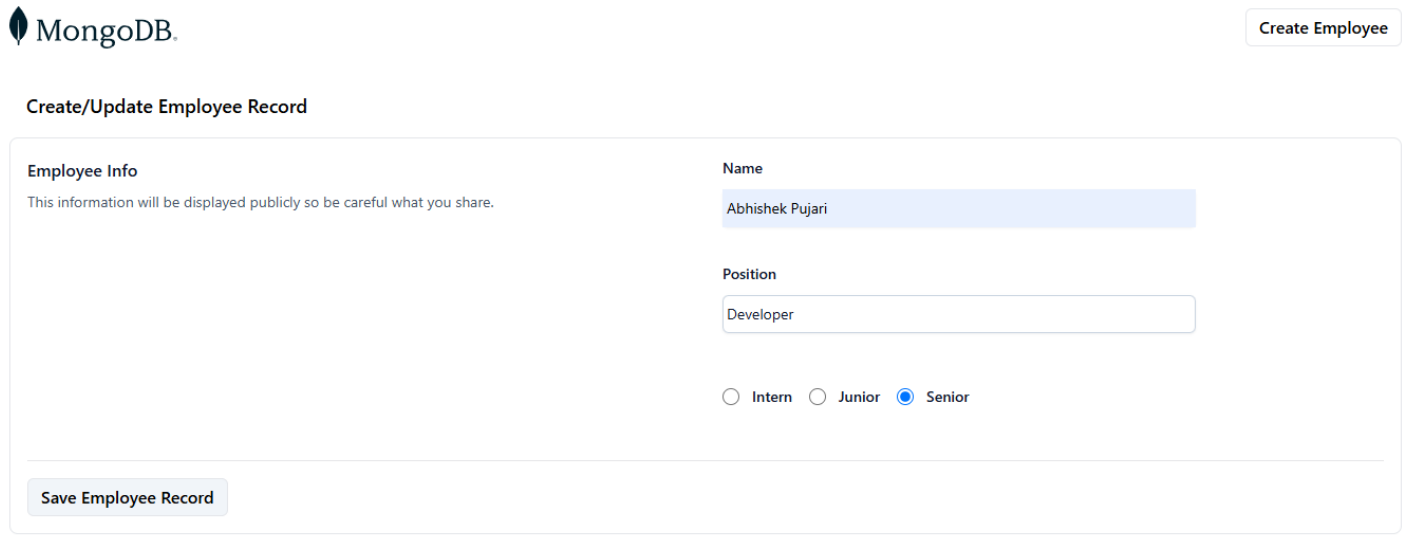
STEP 9:

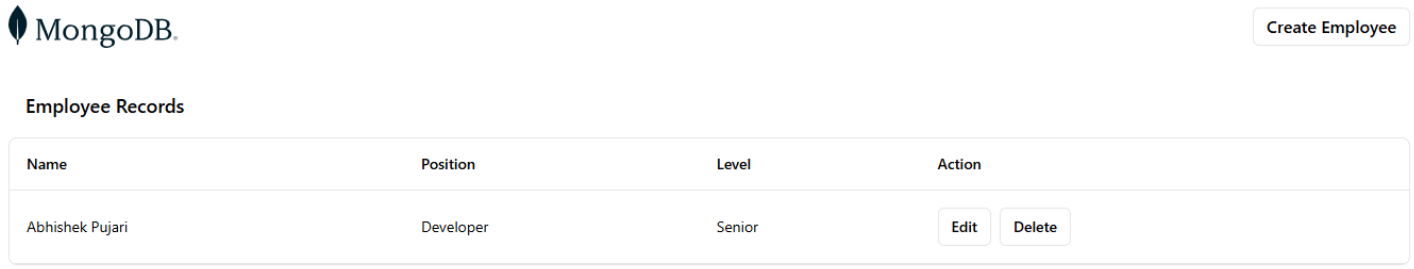
Create the container from the image file

Docker run –name=backend –network=mern –d –p 5050:5050 mern-backend

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose\mern\backend>docker run --name=backend --network=mern -d -p 5050:5050 mern-backend
a512924d5784c3bc40becb1a82650c1b81ad0a6fc57606730eb99df8af381b45
```

Now we can add data to the mongoDB.





Creating a docker compose file

Now creating a docker compose fileso we don’t have to write all the command we just have to run the compose file and the container will be created.

```yaml
services:
  backend:
    build: ./mern/backend
    ports:
      - "5050:5050"
    networks:
      - mern_network
    environment:
      MONGO_URI: mongodb://mongo:27017/mydatabase
    depends_on:
      - mongodb

  frontend:
    build: ./mern/frontend
    ports:
      - "5173:5173"
    networks:
      - mern_network
    environment:
      REACT_APP_API_URL: http://backend:5050

  mongodb:
    image: mongo:latest
    ports:
      - "27017:27017"
    networks:
      - mern_network
    volumes:
      - mongo-data:/data/db

networks:
  mern_network:
    driver: bridge

volumes:
  mongo-data:
    driver: local
```

Now executing the docker compose file

Docker compose up –d

```
X:\AWS PRATICE\DOCKER\MERN-docker-compose>docker compose up -d
[+] Running 4/4
 ⠿ Network mern-docker-compose_mern_network  Created
 ⠿ Container mern-docker-compose-mongodb-1   Started
 ⠿ Container mern-docker-compose-frontend-1  Started
 ⠿ Container mern-docker-compose-backend-1   Started
```