

بسمه تعالی

پروژه داکرایز CURD

دانشجو : امیر صالح رضایی 4001277318
استاد: دکتر طاهری

سیستم عامل استفاده شده arch linux می باشد.
برای نصب داکر روی ارچ ابتدا سورس ان را دانلود میکنیم سپس نصب میکنیم .
دستورات لازم در زیر آمده :

```
wget https://download.docker.com/linux/static/stable/x86_64/docker-27.4.0.tgz -qO- | tar xvfz  
- docker/docker --strip-components=1
```

```
mv ./docker /usr/local/bin
```

```
sudo pacman -U ./docker-desktop-x86_64.pkg.tar.zst
```

بعد از نصب برای تست اولیه ان دستور زیر را وارد میکنیم و چیزی طبق تصویر زیر مشاهده می کنیم:

docker run hello-world

```
[amir@myarch ~]$ docker run hello-world  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

برای اینکه بعد از هر لاگین در سیستم داکر فعال شود نیاز است ان را در **systemctl** فعال کنیم برای این کار دستورات زیر را وارد میکنیم:

```
systemctl --user start docker-desktop  
systemctl --user enable docker-desktop
```

برای ادامه کار به docker compose نیز نیاز است که بطور خودکار نصب می شود که با دستور زیر نصب بودن آن را چک میکنیم :

```
docker compose version
```

در صورت نصب نبودن آن را طبق دستورات زیر نصب میکنیم:

```
sudo curl -L
"https://github.com/docker/compose/releases/download/v2.18.0/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
docker-compose --version
```

حالا یک پروژه CRUD با node js ایجاد می کنیم به این صورت :

```
mkdir cloud project
cd cloud project
```

```
npm init -y
```

```
npm install express pg-promise dotenv body-parser cors
npm i
```

و بعد از آن پروژه را در فایل app.js می نویسیم.

برای داکرایز کردن این پروژه ابتدا یک داکر فایل می سازیم و دستورات زیر را در آن می نویسیم :

```
# Use the Node.js 18 image as the base image
FROM node:18
```

```
# Set the working directory inside the container
WORKDIR /app
```

```
# Copy package.json and package-lock.json
COPY package*.json ./
```

```
# Install dependencies
RUN npm install
```

```
# Copy the application code, including the .env file
COPY . .
```

```
# Expose the application port
EXPOSE 3000
```

```
# Define the command to run the application
CMD ["npm", "start"]
```

یک فایل به نام `docker-compose.app.yml` ایجاد می کنیم:

```
version: '3.8'
```

```
services:
```

```
  server:
```

```
    build:
```

```
      context: .
```

```
    container_name: restaurant
```

```
    ports:
```

```
      - '3000:3000' # استفاده از پورت 3000 برای دسترسی به سرویس
```

```
    environment:
```

```
      - PORT=3000
```

```
      - PG_HOST=db # اشاره می کند که از سرویس دوم در فایل دیگر می آید db به
```

```
      - PG_PORT=5432
```

```
      - PG_DATABASE=restaurant
```

```
      - PG_USER=postgres
```

```
      - PG_PASSWORD=12345
```

```
    volumes:
```

```
      - ./app
```

```
      - /app/node_modules
```

```
    networks:
```

```
      - saleh-network
```

```
networks:
```

```
  saleh-network:
```

```
    external: true
```

و یک فایل `docker-compose.db.yml`:

```
version: '3.8'
```

```
services:
```

```
  db:
```

```
image: postgres:15
container_name: postgres
ports:
  - '5432:5432' # استفاده از پورت پیش فرض PostgreSQL
environment:
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: 12345
  POSTGRES_DB: restaurant
volumes:
  - pg_data:/var/lib/postgresql/data
networks:
  - saleh-network
```

```
volumes:
  pg_data:

networks:
  saleh-network:
    external: true
```

ابتدا شبکه‌ای به نام **saleh-network** ایجاد می‌کنیم که بین دیتابیس و اپلیکیشن مشترک باشد:

```
docker network create saleh-network
```

فایل **docker-compose.db.yml** را اجرا می‌کنیم تا دیتابیس در کانتینر خود راه‌اندازی شود:

```
docker-compose -f docker-compose.db.yml up -d
```

فایل **docker-compose.app.yml** را برای اپلیکیشن اجرا می‌کنیم. این کار شامل ساخت ایمیج و اجرای کانتینر اپلیکیشن است:

```
docker-compose -f docker-compose.app.yml up --build
```

برای اطمینان از اجرای درست کانتینرها، وضعیت آن‌ها را بررسی می‌کنیم:

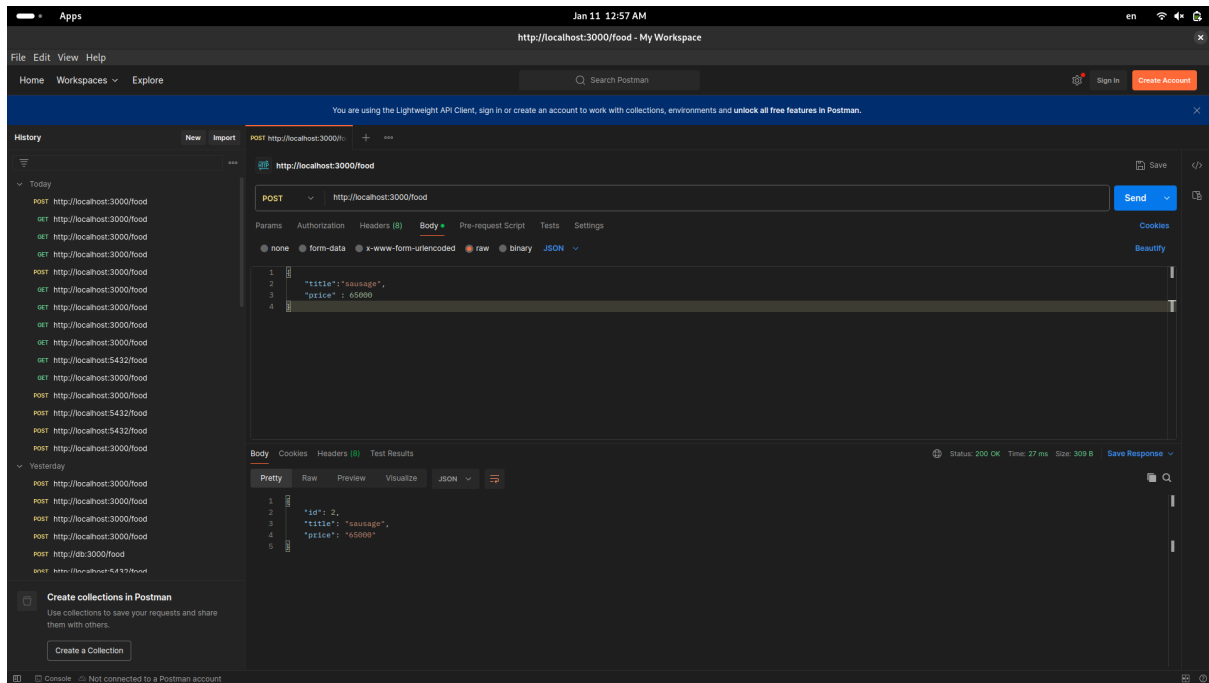
```
docker ps
```

با وارد کردن آدرس زیر در **postman** می‌توانیم پروژه را تست کنیم:

<http://localhost:3000/food>

برای اینکار ما یک نمونه دیتا رو **post** کردیم و دوباره **get** کردیم که به درستی کار میکنه:

POST



GET

