**DEVOPS PROJECT**

**Docker Containerization**

create a Dockerfile to containerize the application and ensure it runs correctly within a Docker container

I have a project in my git hub repository , I need to run the application in my local environment (optional).

Requirement:

* Python : 3.6 – 3.7
* Postgresql
* Dijango

Step to run the project

Clone the git hub repository

**git clone** [**https://github.com/jameskomo/bus-reservation-system.git**](https://github.com/jameskomo/bus-reservation-system.git)

change the directory to project directory

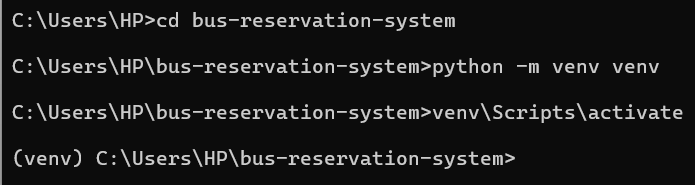
**cd bus-reservation-system**

Create a virtual environment

**python -m venv venv**

here, the python creates a virtual environment.so,we needed to activate the environment

**venv\Scripts\activate**



Then,

Required to run the command

**pip install -r requirements.txt**

this command show that the project required dependencies are install in virtual environment

to run without any errors.

Next step to check:

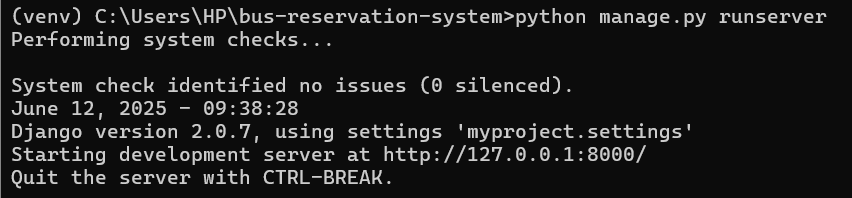
A black screen with white text

AI-generated content may be incorrect.

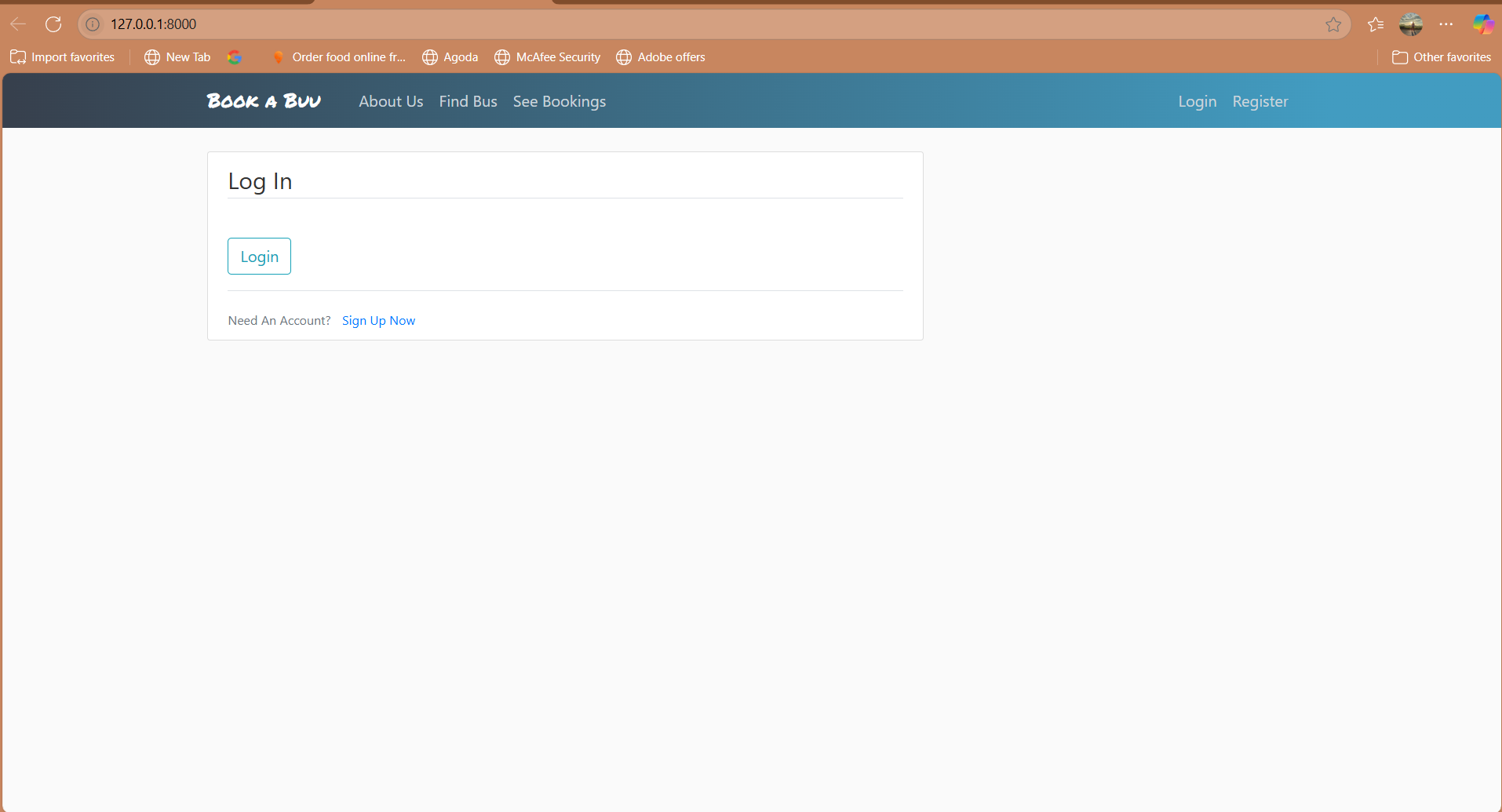
The final step is to run the project .

Using this commad :

python manage.py runserver



The project runs successfully . we can check them in server at [**http://127.0.0.1:8000/**](http://127.0.0.1:8000/)



create a Dockerfile to containerize the application and ensure it runs correctly within a Docker container.

**Create a Dockerfile :**

# Use official Python image

FROM python:3.7-slim

# Set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# Set work directory

WORKDIR /code

# Install system dependencies

RUN apt-get update && apt-get install -y \

    libpq-dev gcc python3-dev \

    && pip install --upgrade pip

# Copy project files

COPY requirements.txt /code/

RUN pip install -r requirements.txt

# Copy the rest of the code

COPY . /code/

# Create static files directory

RUN mkdir -p /code/static

# Collect static files

# RUN python manage.py collectstatic --noinput

# Run migrations (optional for dev)

CMD ["gunicorn", "myproject.wsgi:application", "--bind", "0.0.0.0:8000"]

Above code for build docker images and run the docker image

With this file, I have onemore file that is docker-compose.yml

**version: '3.9'**

**services:**

**web:**

    build: .

    command: gunicorn myproject.wsgi:application --bind 0.0.0.0:8000

    volumes:

      - .:/code

**ports:**

      - "8000:8000"

    env\_file:

      - .env

    depends\_on:

      - db

**db:**

    image: postgres:12

    environment:

      POSTGRES\_DB: ${DB\_NAME}

      POSTGRES\_USER: ${DB\_USER}

      POSTGRES\_PASSWORD: ${DB\_PASSWORD}

    volumes:

      - postgres\_data:/var/lib/postgresql/data

**volumes:**

**postgres\_data:**

There is a another file that is .dockerignore

\_\_pycache\_\_

\*.pyc

\*.pyo

\*.pyd

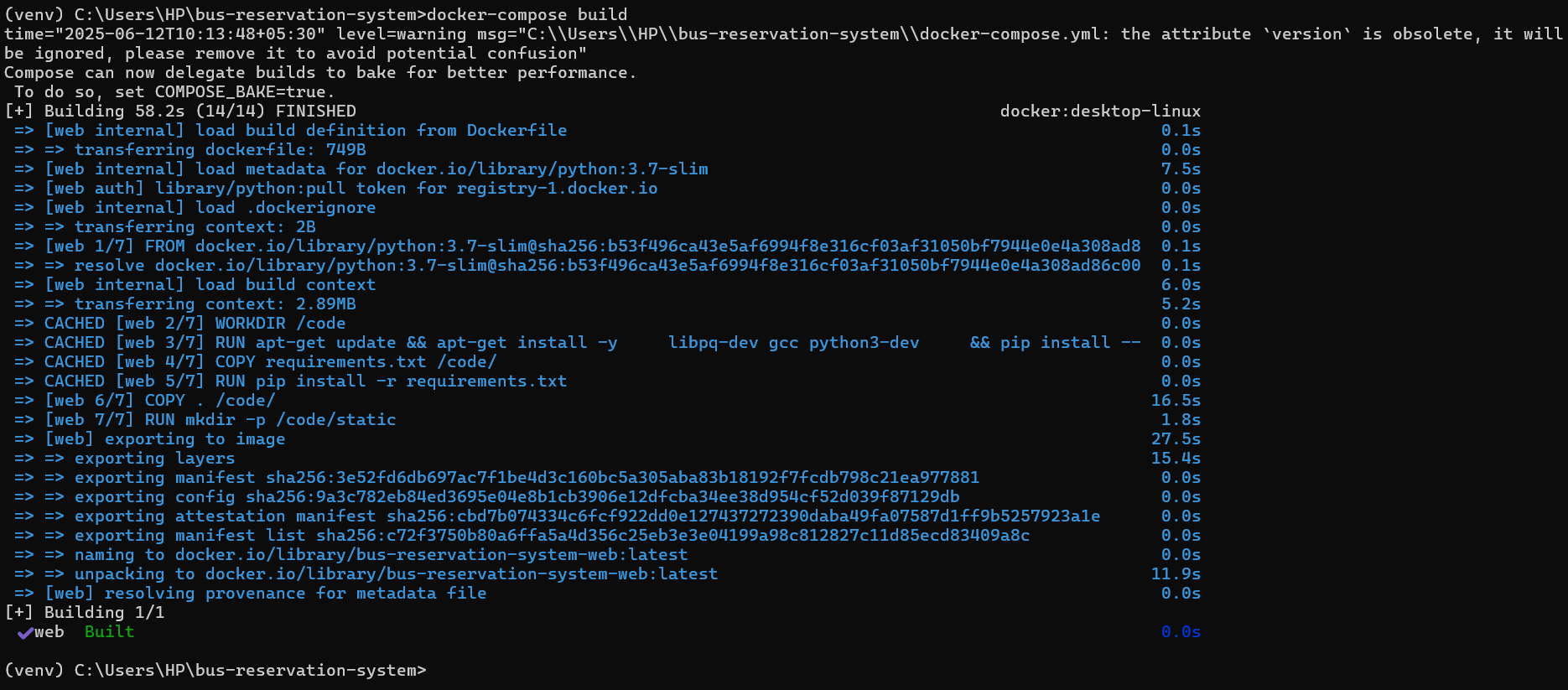
\*.sqlite3

\*.env

.git

Then run the commad for build the image

**docker-compose build**



To check the images in docker hub or containers

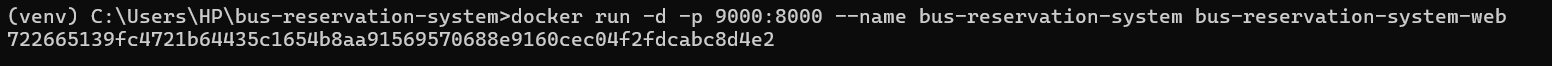
**docker images**

**A computer screen with white text

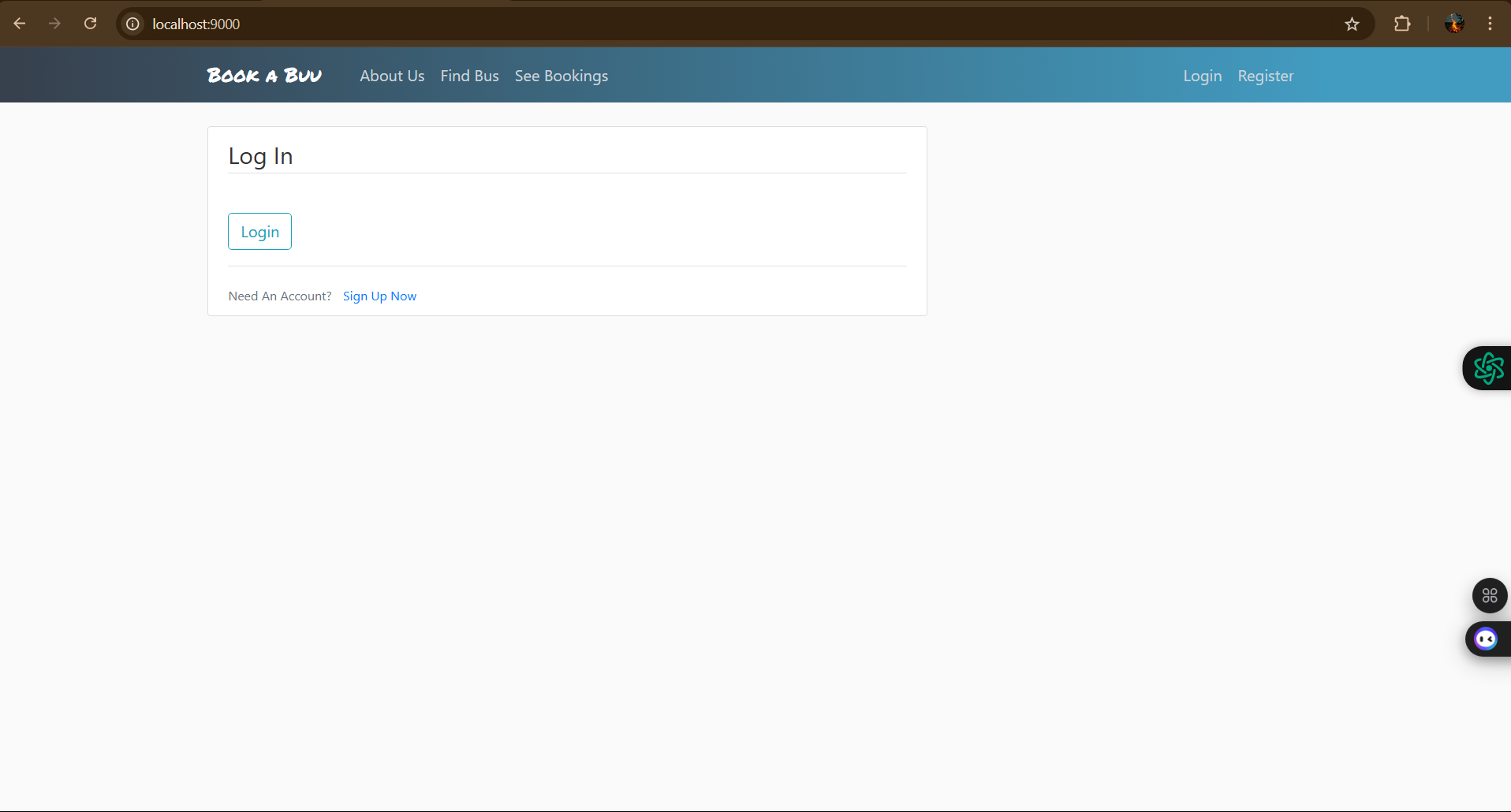
AI-generated content may be incorrect.**

Now, we required to run the container

**docker run -d -p 8000:8000 --name bus-reservation-system bus-reservation-system-web**

****

Finally, the project run in the suitable port number : 9000



To check them in docker hub whether the container is running or not.

A screenshot of a computer

AI-generated content may be incorrect.