Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Front-End-програмування

Лабораторна робота 2

Варіант №21

Виконав:
студент групи ІА-72
Чорнолоз Денис
Дата здачі
Захищено з балом

Перевірив:
викладач кафедри АУТС
Жереб К. А.

Київ 2021

Завдання

Необхідно встановити та налаштувати інструментальні засоби для front-end розробки. Результатом виконання даної лабораторної роботи є docker-образ з встановленими інструментальними засобами, а також звіт про використані технології та можливості. У docker-образі можна також додати компоненти, що відповідають за backend.

Посилання на проект

https://github.com/Velns/FrontEnd_Lab2/tree/master

Хід роботи

У ході даної лабораторної роботи було створено docker образ, який містить налаштування для розгортання створеного у лабораторній роботі 1 додатку.

Docker являє собою стандартизований пакетний програмний застосунок для створення та розгортання програмних комплексів.

В межах нашого проекту він слугує для розгортання середовища та запуску контейнеру в якому знаходиться розроблений сайт.

Першим кроком необхідно налаштувати майбутній контейнер.

Створюємо пустий файл з іменем «Dockerfile» та заповнюємо його налаштуваннями, де вказуємо початковий образ, задаємо змінні, вказуємо робочу директорію, задаємо початкові команди, передаємо файли проекту всередину контейнеру та задаємо порт.

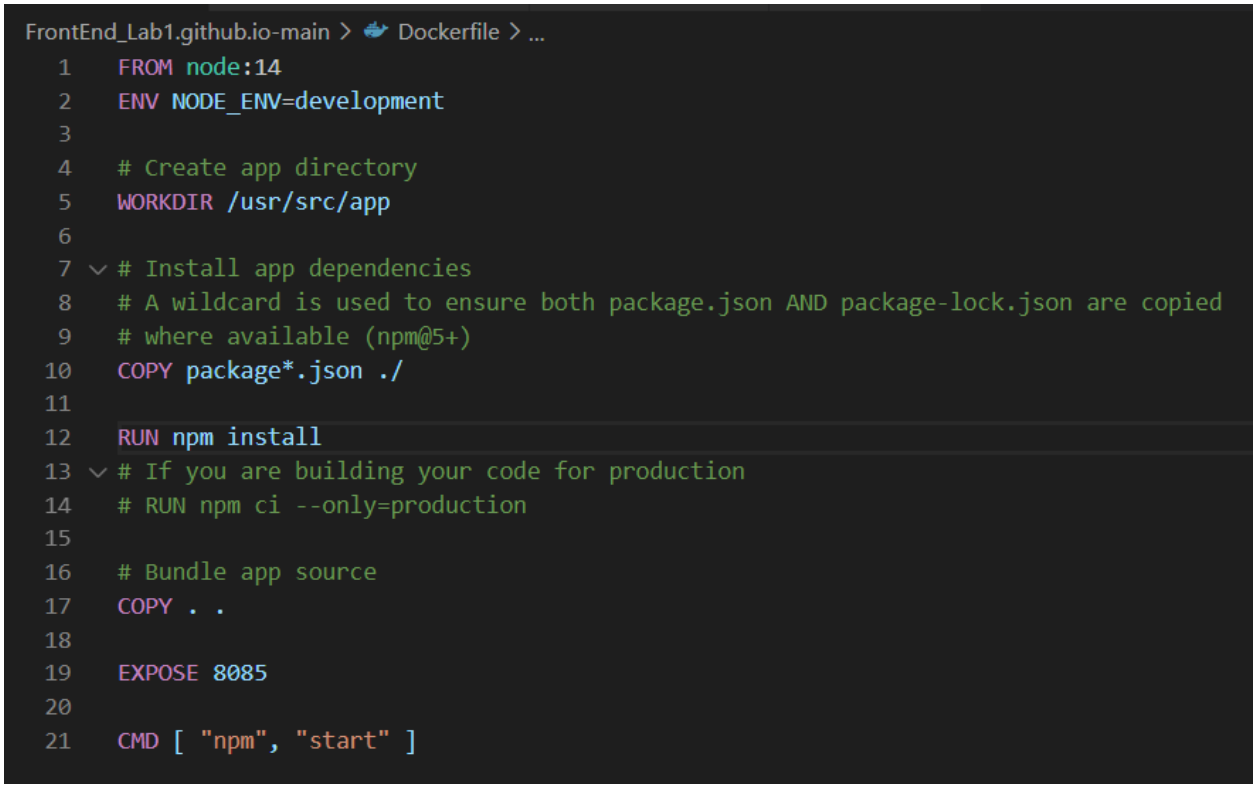

Рисунок 1 – Файл Dockerfile

Наступним кроком необхідно створити файл docker-compose.yml, в якому задаються пов’язані служби.

FrontEnd_Lab1.github.io-main > docker-compose.yml

```yaml
version: '3.4'

services:
  usergallery:
    image: denyshornoloz/frent-end-lab2
    build:
      context: .
      dockerfile: ./Dockerfile
    environment:
      NODE_ENV: development
    ports:
      - 9090:8085
```

Рисунок 2 – Файл docker-compose.yml

Далі створимо файл package.json. У цьому файлі описується наш додаток та вказуються його зв’язки.

FrontEnd_Lab1.github.io-main > {} package.json > ...

```json
{
  "name": "user-gallery",
  "version": "1.0.0",
  "description": "own project",
  "main": "server.js",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "Denys Chornoloz",
  "license": "MIT",
  "dependencies": {
    "bootstrap": "^5.0.0-beta1",
    "connect": "^3.7.0",
    "serve-static": "^1.14.1"
  }
}
```

Рисунок 3 – Файл package.json

Після слід створити файл server.js записи в якому визначають наш веб додаток.

```
FrontEnd_Lab1.github.io-main > JS server.js > ...
1   var connect = require('connect');
2   var serveStatic = require('serve-static');
3   const port = 8085;//8080
4
5   connect()
6       .use(serveStatic("./app"))
7       .listen(port, () => console.log(`Server running on ${port}...`));
```

Рисунок 3 – Файл server.js

В результаті маємо директорію представлену на рисунку 5.

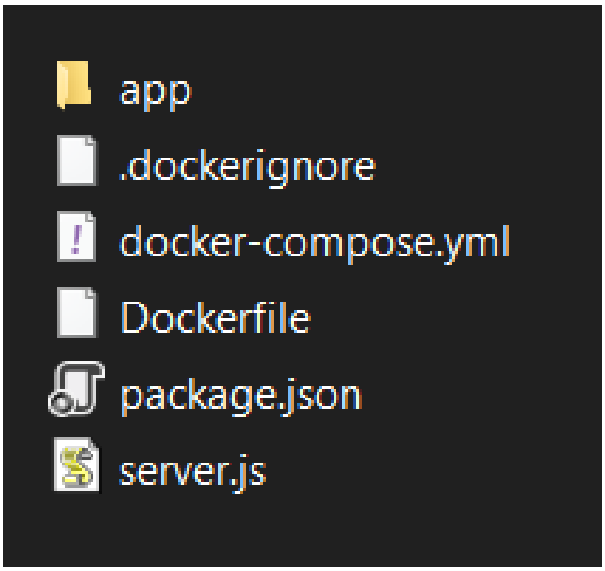

Рисунок 5 – Директорія з підготовленим до розгортання додатком

На наступному етапі виконується побудова контейнеру.

Для цього необхідно:

1. Запустити термінал
2. Перейти в директорію з проектом
3. виконати команду docker-compose up

Результатом описаних вище маніпуляцій виступатиме повідомлення про успішні побудову та запуск контейнеру (рисунок 6).

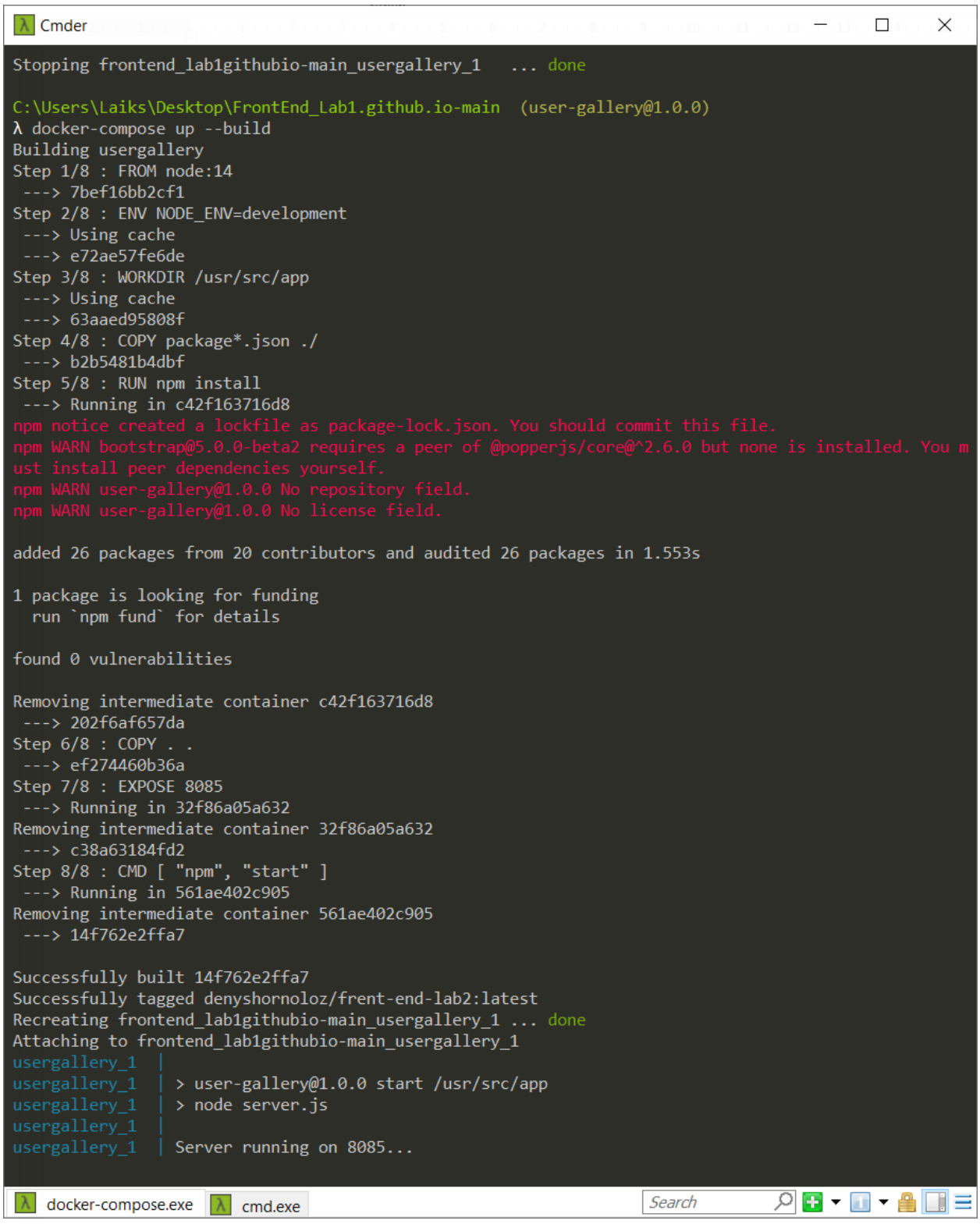

Рисунок 6 – Запущений docker контейнер

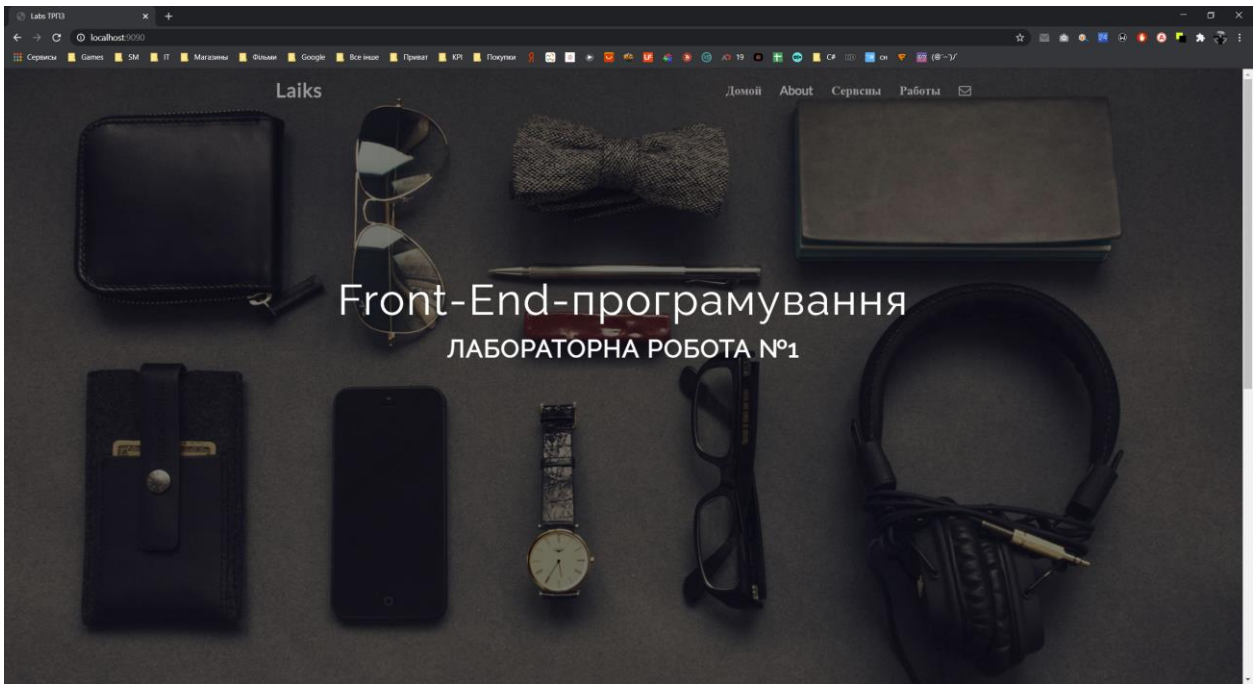

Рисунок 7 – Результат роботи