# Assignment 2: Advance Containers

# Documentation

This documentation provides a complete overview of setting up and using the QuoteVault system.

## 1. Project Overview

QuoteVault is a containerized full stack web application built with the following technologies:

* React frontend (Node.js)
* Flask backend (Python)
* PostgreSQL database
* NGINX reverse proxy
* All services are orchestrated using Docker Compose.

## 2. Repository Structure

* backend/: Python Flask API
* frontend/: React client
* db/: Contains init.sql for DB bootstrap
* nginx/: Reverse proxy config (default.conf)
* docker compose.yml: Orchestration file for all services
* .env: Environment variables (excluded from Git)
* README.md: Instructions
* screenshots/: Collected proof of setup and testing

## 3. Setup Instructions

To run the application in GitHub Codespaces or locally:

1. Clone the repository:

*git clone https://github.com/JithinJyothi95/A2 Advance Containers QuoteVault.git*

2. Navigate into the project folder.

3. Run: docker compose up build

Make sure to run `npm install` inside the `frontend/` folder before building if working locally.

## 4. Environment Variables

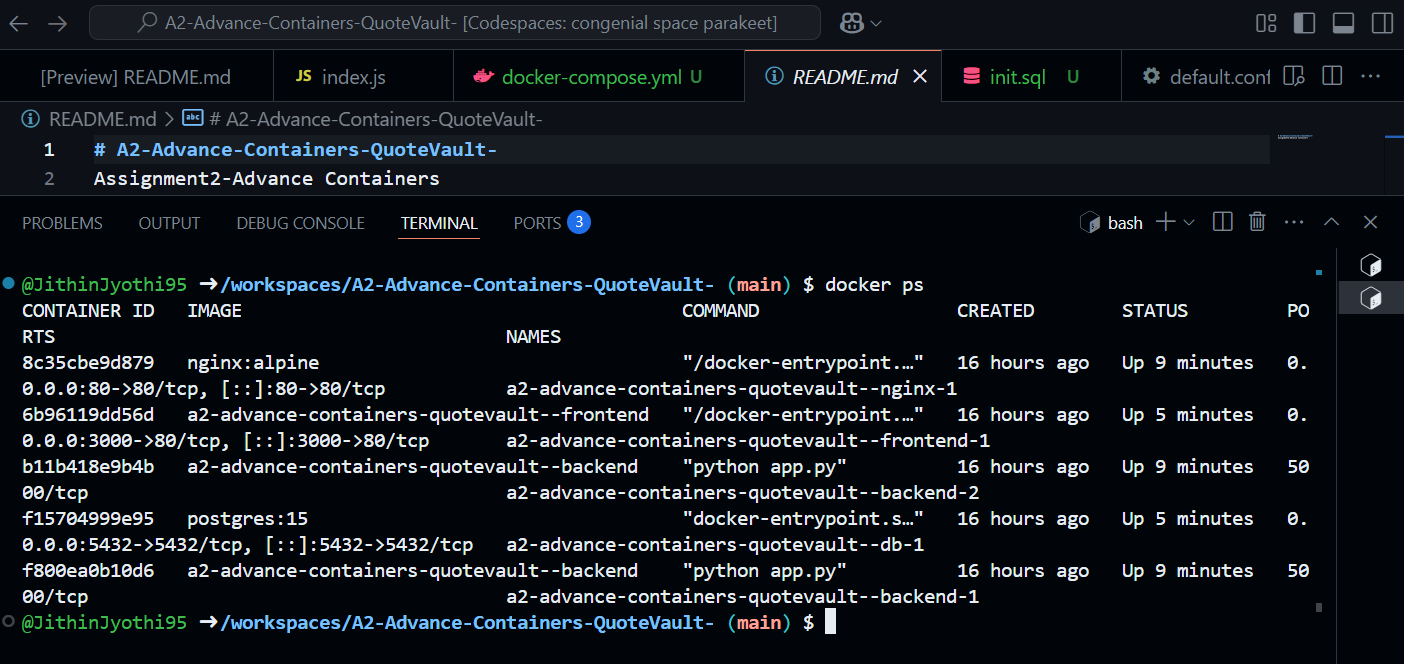
* Create a `.env` file in the root directory with the following content:  
  POSTGRES\_USER=postgres  
  POSTGRES\_PASSWORD=postgres  
  POSTGRES\_DB=quotevault

## 5. Features Demonstrated

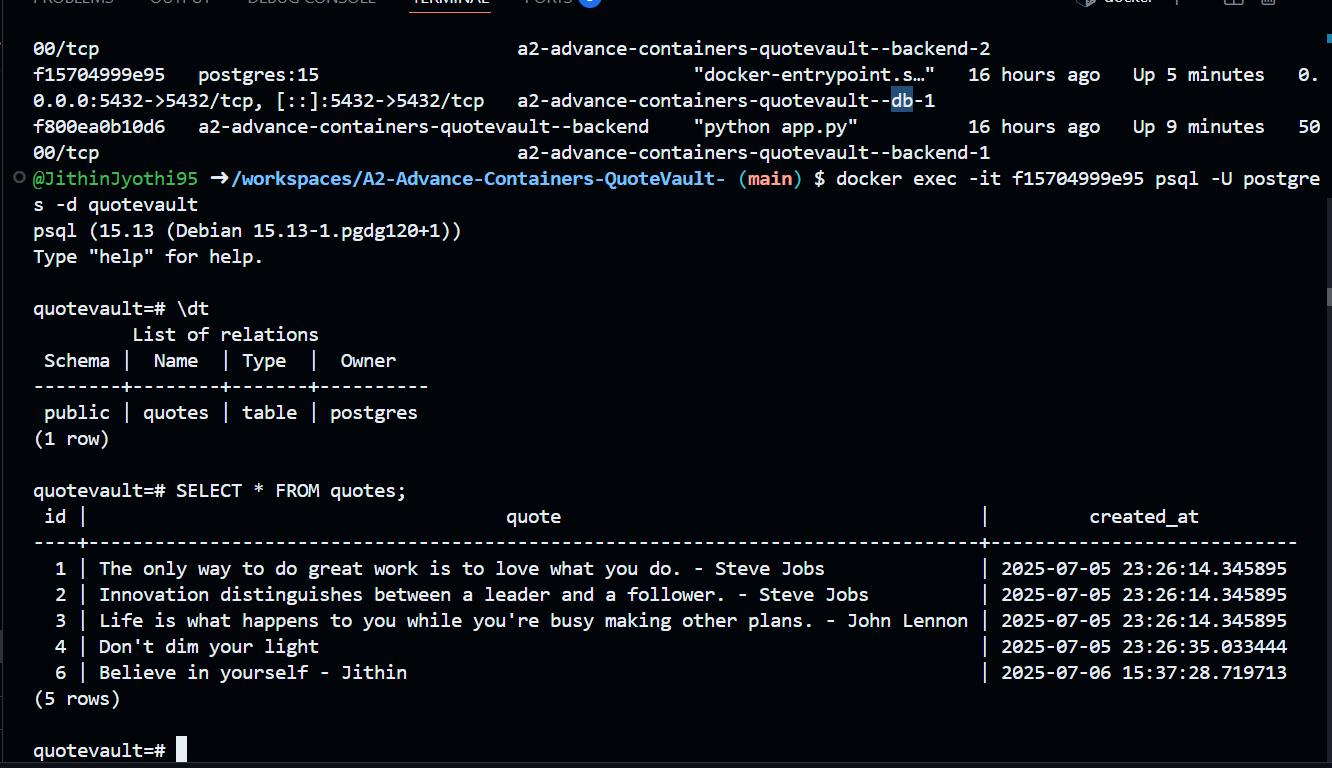
* Multi container orchestration via Docker Compose.
* API endpoints tested via fetch() and browser.
* PostgreSQL query via psql container session.
* UI rendered via React and reverse proxied via NGINX.
* All containers running and connected to common Docker network.

## 6. Screenshots

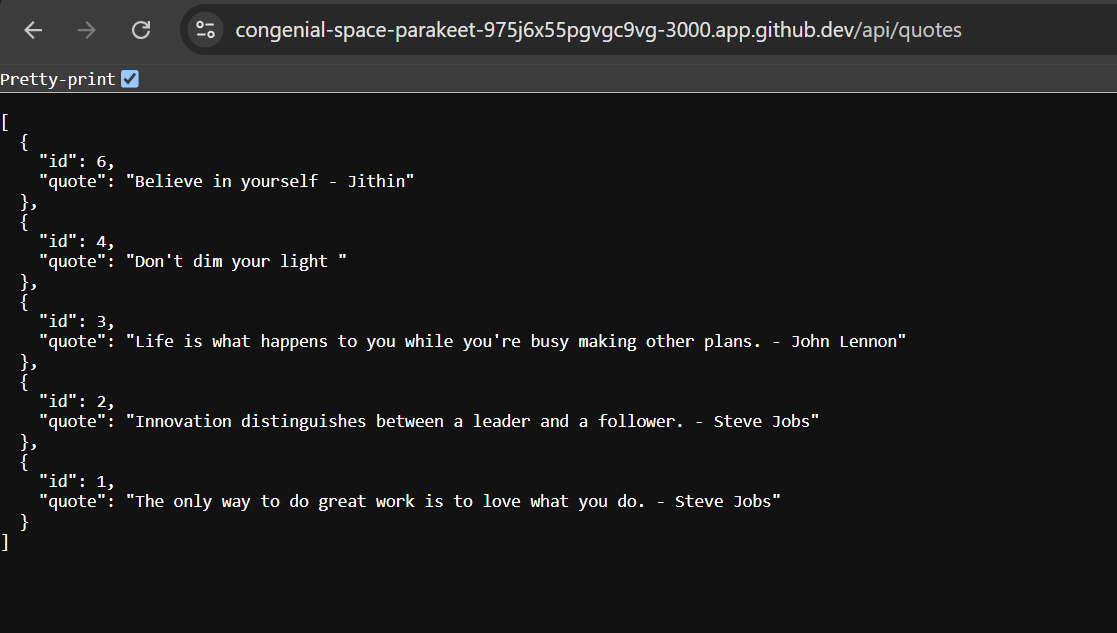
* Docker containers running



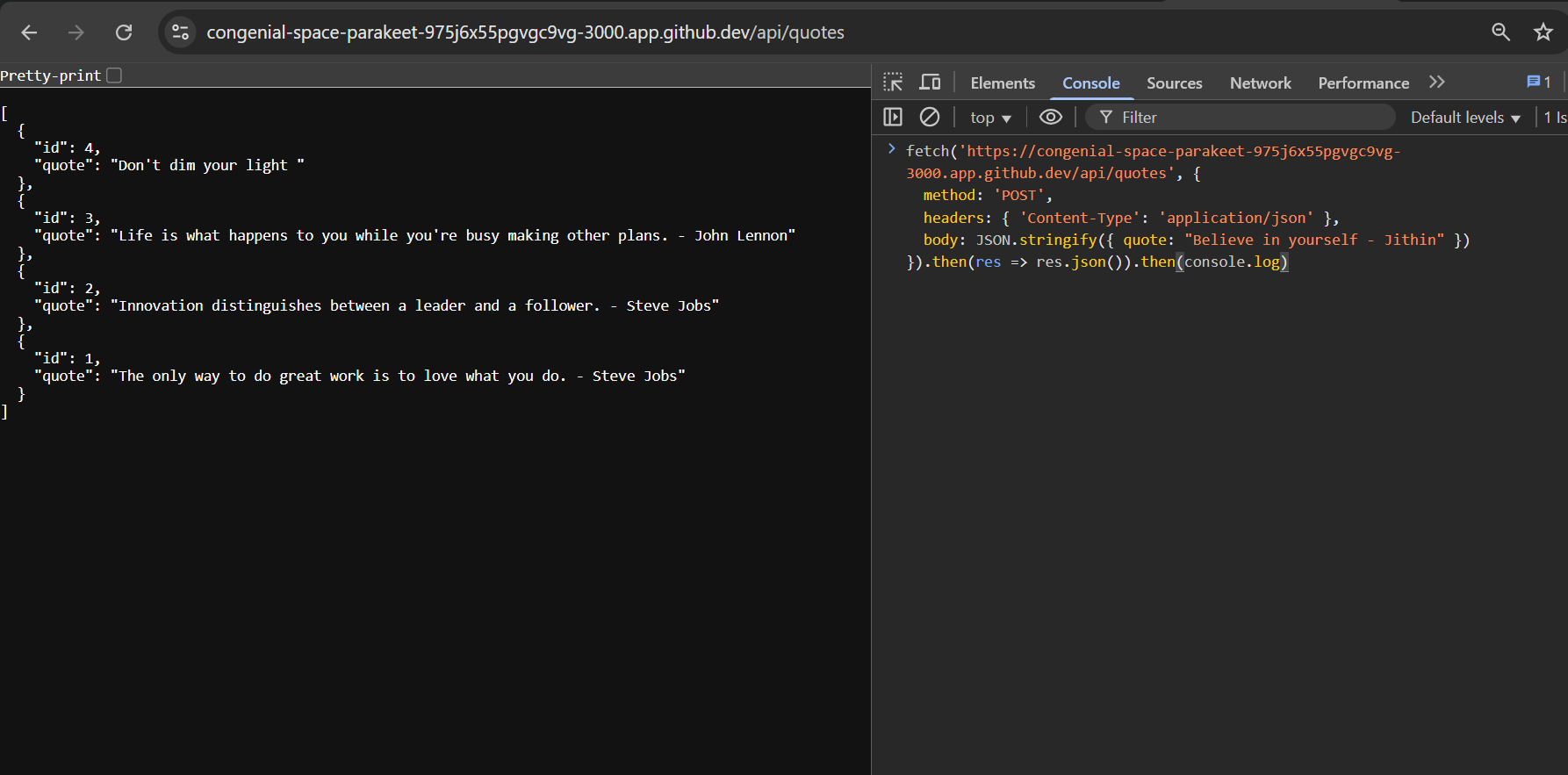
* PostgreSQL database test



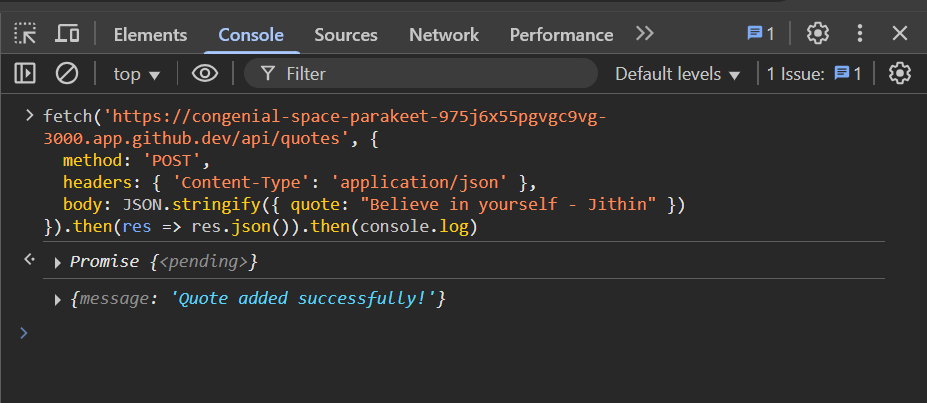
* Backend API JSON response



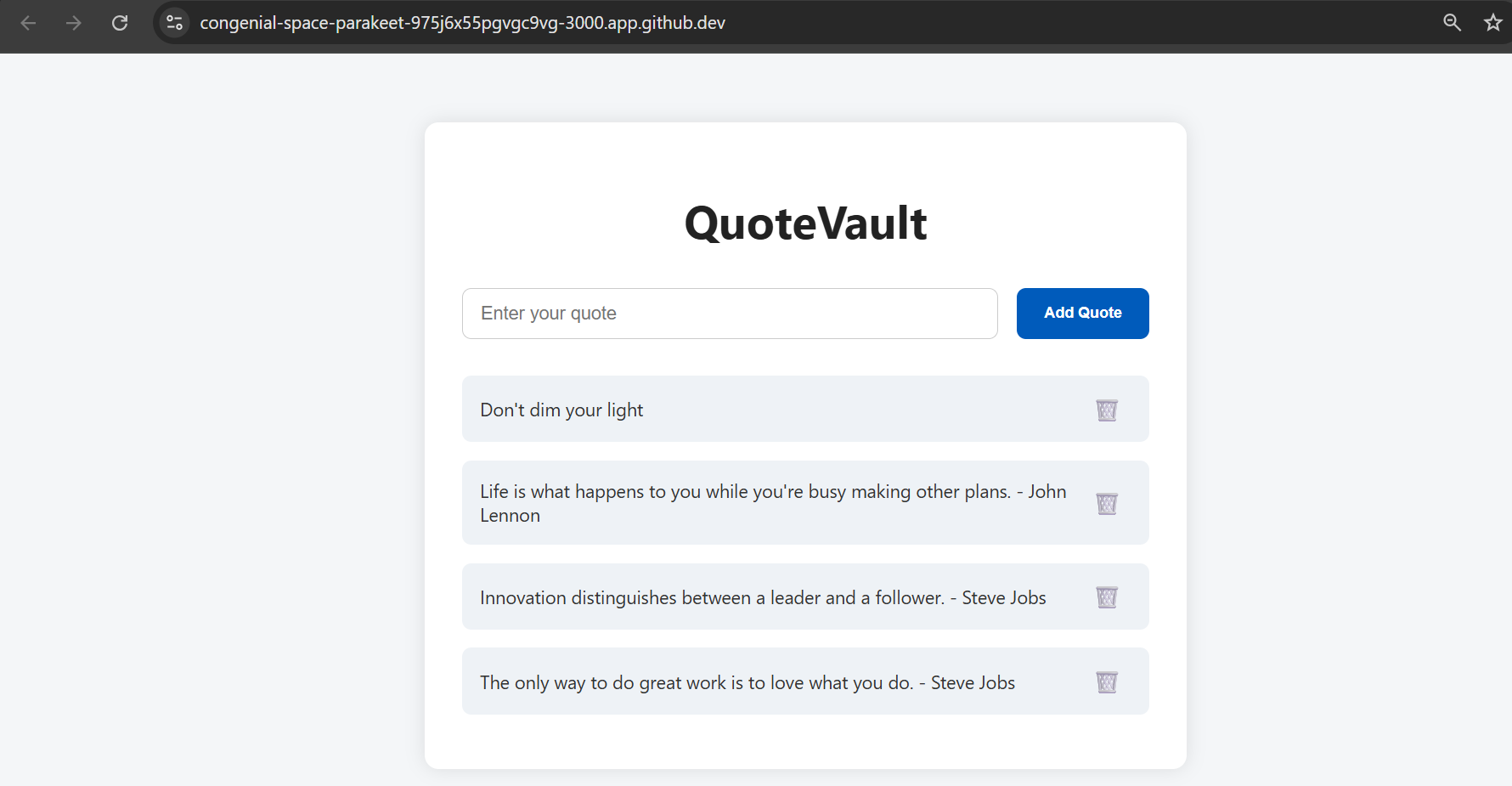
* POST request using fetch()



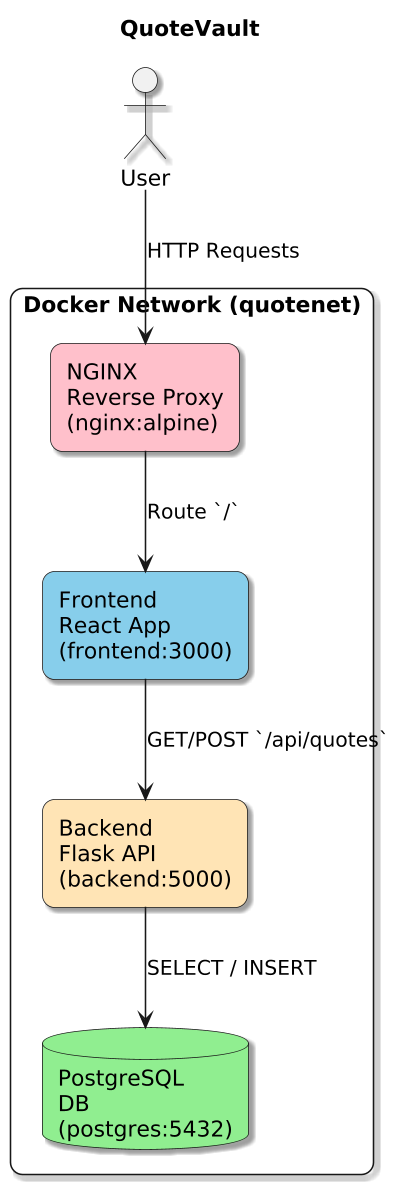
* POST success response



* Frontend UI on browser



* Application Architecture Diagram



## 7. API Endpoints

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/quotes | Get all quotes |
| POST | /api/quotes | Add new quote |
| DELETE | /api/quotes/:id | Delete quote by ID |

## 8. Notes

node\_modules are excluded via .gitignore and must be installed with npm install.  
 The .env file must be created before launching the stack.  
 The Flask backend runs on port 5000 but is accessed via frontend:3000 through NGINX.