# IT21237904 – Gunawardana U. G. C. D

## Github link - https://github.com/ch4mi2/CSTE-Lab-1

# Containerizing a Simple Application.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen with blue text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Using Docker Volumes

A black screen with white text

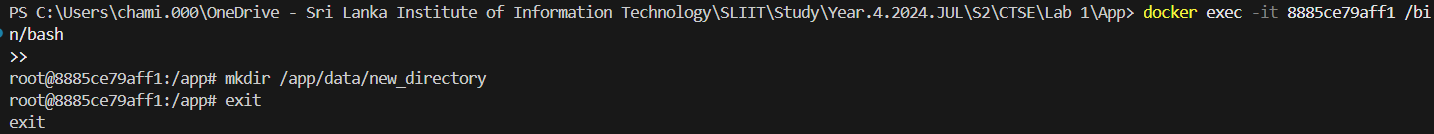
AI-generated content may be incorrect.



A screen shot of a computer

AI-generated content may be incorrect.

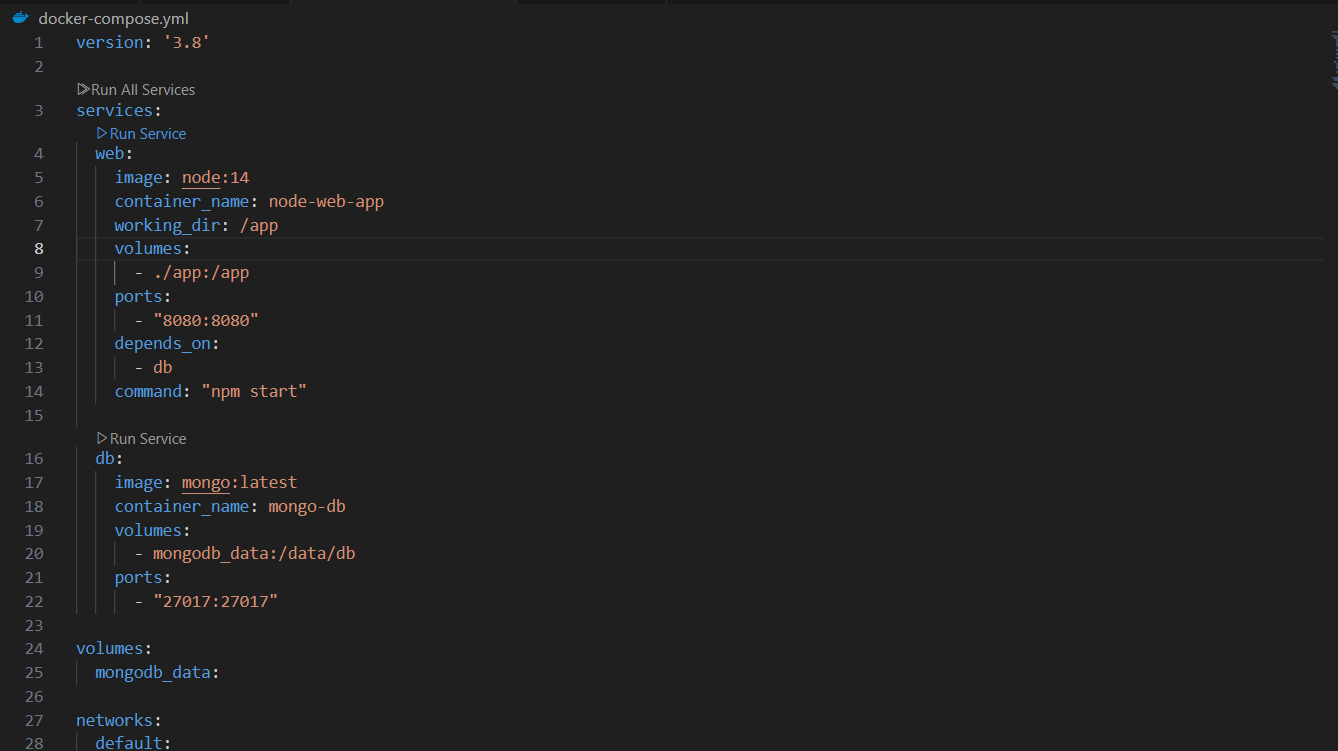
# Using Bind Mounts



A screenshot of a computer

AI-generated content may be incorrect.

# Using Docker Compose



A computer screen with white text

AI-generated content may be incorrect.

# Understanding Docker Layers

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Learnings:

Docker Layers: Each command in a Dockerfile creates a layer in the image. By understanding how layers work, you can optimize Dockerfiles to avoid unnecessary rebuilding and speed up the development cycle.

Efficient Caching: By copying requirements.txt first, Docker can cache the dependency installation step, so it doesn't need to re-run it when only the application code changes.

Faster Development Cycles: Optimizing Dockerfiles can save a significant amount of time in development, especially when you're iterating frequently on application code.