****

**CoDeKu Paid Mini Tasks**

|  |  |
| --- | --- |
| **Project Code** | **CPT-DC-006** |
| **Project Name** | **Simple web app with flask** |
| **Your Name** | **Amalsha Fernando** |
| **Your Email** | **amalshaf6@gmail.com** |
| **Your WhatsApp Number** | **0712586284** |
| **Submission Date** | **29-07-2024** |
| **Task** | **Creating a simple Flask web application that displays "Hello, CoDeKu!", containerize it with Docker, and test it by running the container.** |

Contents

[**Summary** 3](#_Toc173152839)

[**Project Objectives** 3](#_Toc173152840)

[**Project Timeline** 3](#_Toc173152841)

[**Resources** 3](#_Toc173152842)

[**Expected Outcomes** 3](#_Toc173152843)

[**Methodology** 4](#_Toc173152844)

[**1. Setting Up the Project Environment:** 4](#_Toc173152845)

[**2. Developing the Flask Application:** 4](#_Toc173152846)

[**3. Containerizing the Application:** 5](#_Toc173152847)

[**4. Building and Running the Docker Container:** 6](#_Toc173152848)

[**5. Testing the Application:** 8](#_Toc173152849)

[**Conclusion** 8](#_Toc173152850)

# **Summary**

This project aims to develop a simple web application using Flask, a lightweight Python web framework, and containerize it using Docker. The application will display a "Hello,CoDeKu!" message on a web page. The primary goal is to demonstrate the process of building, containerizing, and deploying a web application in a consistent and reproducible environment.

In modern software development, containerization has become a standard practice for ensuring applications run consistently across different environments. Docker is a popular containerization platform that allows developers to package applications and their dependencies into a portable container. This project addresses the need for understanding and implementing a basic web application using Flask and Docker, which are widely used technologies in the industry.

# **Project Objectives**

## **Project Timeline**

|  |  |
| --- | --- |
| **Task** | **Spent Time Duration** |
| Set up project environment | 2 min |
| Develop Flask application | 10 min |
| Create Dockerfile | 2 min |
| Build and test Docker image | 1 hour |

## **Resources**

- Personnel: 1 developer

- Software: Python, Flask, Docker, vscode

- Hardware: Development machine with Docker installed

## **Expected Outcomes**

- A working Flask web application that displays "Hello, World!".

- A Docker container image for the Flask application.

- Documentation of the development and containerization process

# **Methodology**

## **1. Setting Up the Project Environment:**

- Create a project directory named flask-docker-app.

- Set up a virtual environment and activate it:

python -m venv venv

source venv/bin/activate

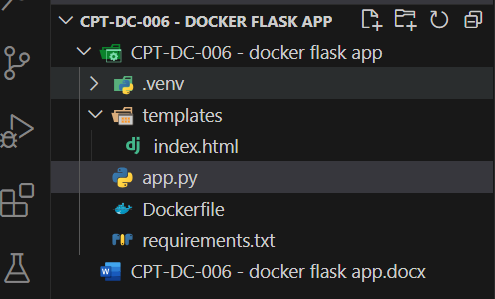


Figure 1- project structure

## **2. Developing the Flask Application:**

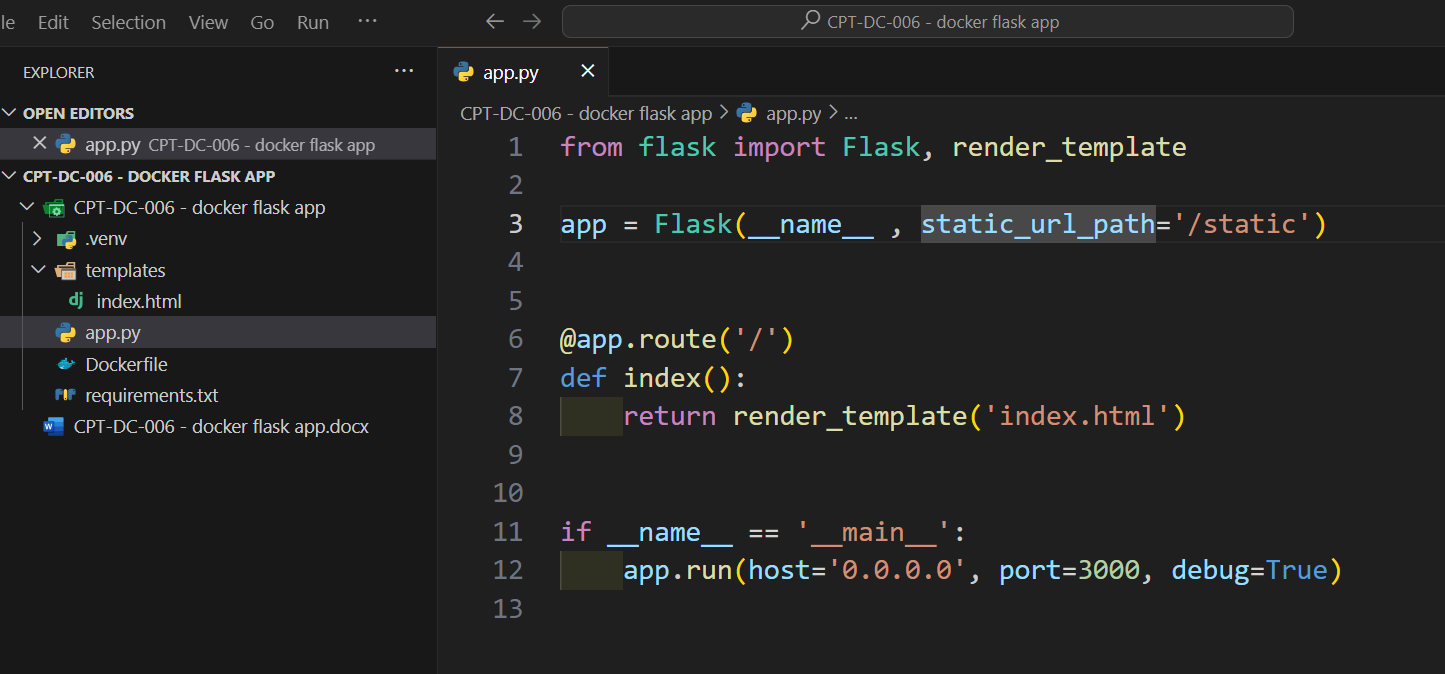
- Create app.py with the following content:

Figure 2

- Create templates/index.html:

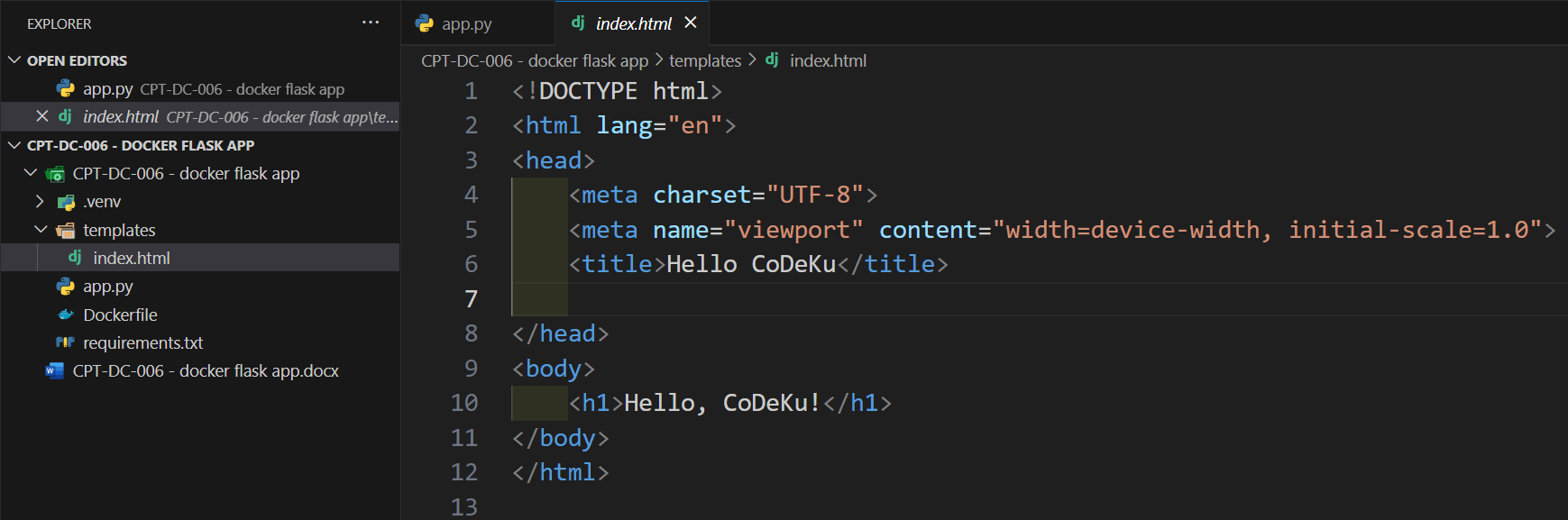


Figure 3

## **3. Containerizing the Application:**

- Create a Dockerfile:

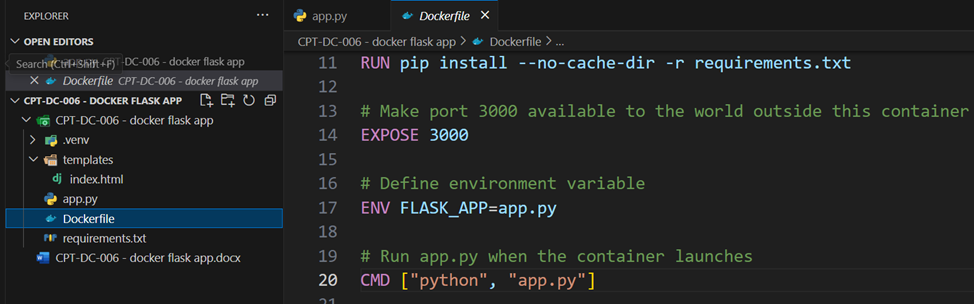


Figure 4

## **4. Building and Running the Docker Container:**

**- Build the Docker image:**

dockerbuild -t amalsha/hello\_codeku:latest .

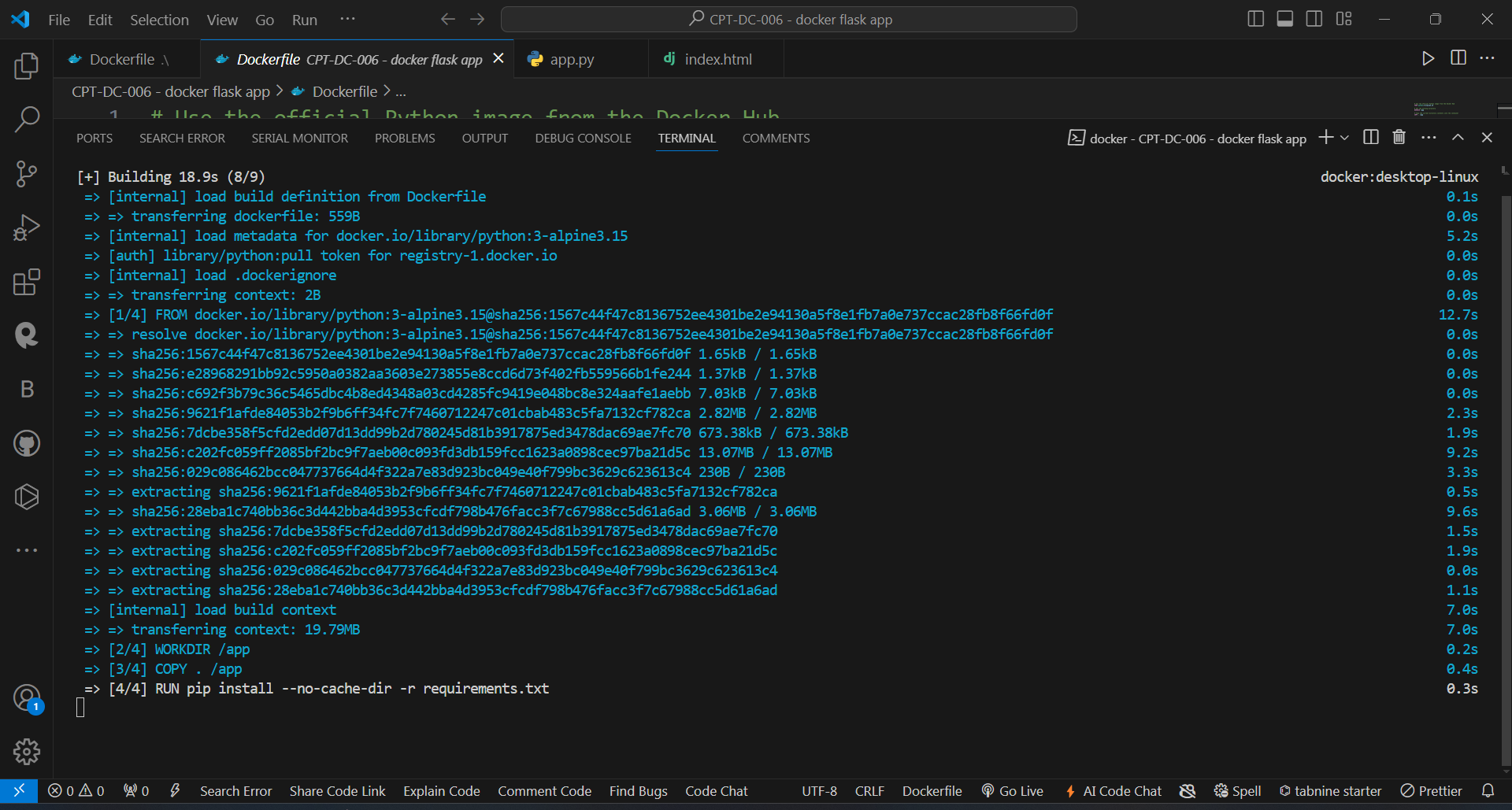


Figure 5

**- Run the Docker container:**

* docker run -p 3000:3000 flask-docker-app
* here first port, 3000 is dedicated for local pc port and second port, 3000 is dedicated for second port

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Figure 6

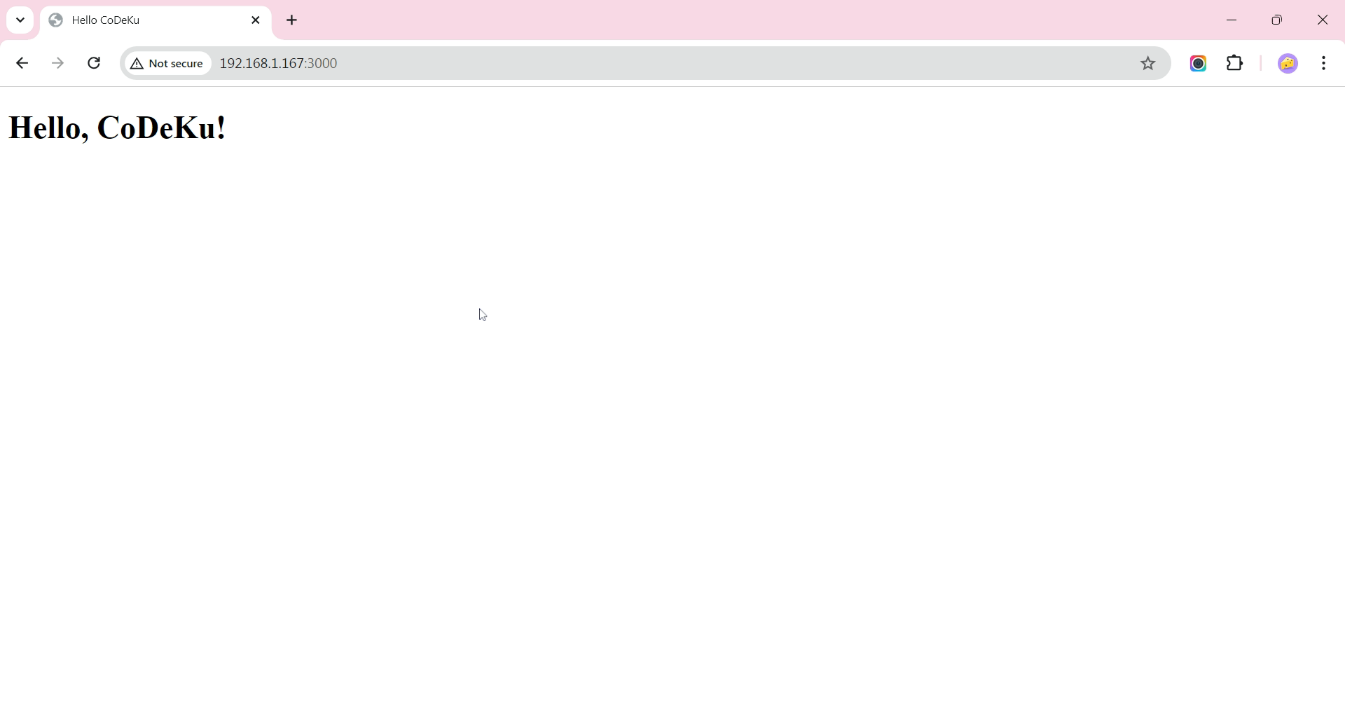
The docker ps command can show currently running containers. At first, it displays running containers, and after stopping containers, it shows nothingA screenshot of a computer program

Description automatically generated

Figure 7

## **5. Testing the Application:**

- When open a web browser and visit http://localhost:5000 to see the "Hello, CoDeKu!" message rendered from the HTML file.



# **Conclusion**

This project will demonstrate the fundamental steps of developing and containerizing a simple web application using Flask and Docker. It serves as a foundational example for more complex applications and containerization processes.