Student Attendance Marker

FLASK APPLICATION WITH DOCKER AND JENKINS CI/CD PIPELINE

Name : Harshith SK
Usn:4ni22cs073
Class:CSB
Collage: The National institute of engineering.
Devops Project
Github: https://github.com/harshith-kudapali/devops-assignment1.git

Introduction

This project focuses on developing a **Student Attendance System** using the **Flask web framework**, **Docker** for containerization, and **Jenkins** for automating the build, test, and deployment processes (CI/CD). The goal of this system is to provide a simple way for instructors to record and track student attendance. The entire application is containerized using Docker, making it portable and easy to deploy in any environment. We've also integrated **Jenkins** to streamline the process, ensuring that each update to the system is automatically tested and deployed.

Technologies Used

- **Flask**: A lightweight Python web framework to build the application.
- **Docker**: A platform to package the application and its dependencies into a container.
- **Jenkins**: A powerful open-source tool to automate tasks like testing, building, and deploying software.

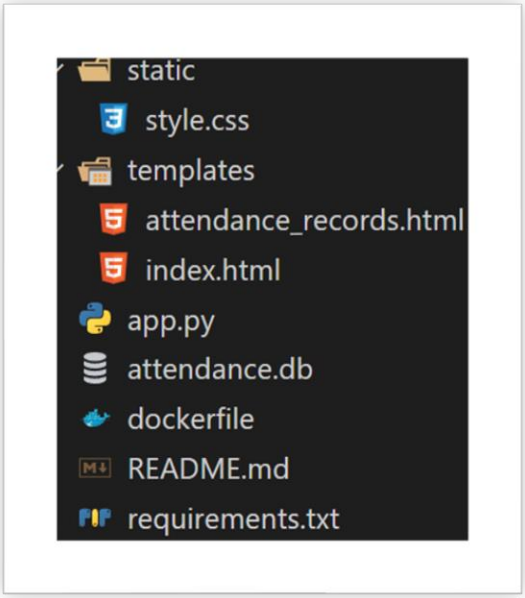
1. Building the Flask Application

Enter a student's name.

Mark that student as "Present" or "Absent."

View all the attendance records in a simple, clean interface

STRUCTURE OF APP.



CODE

App.py

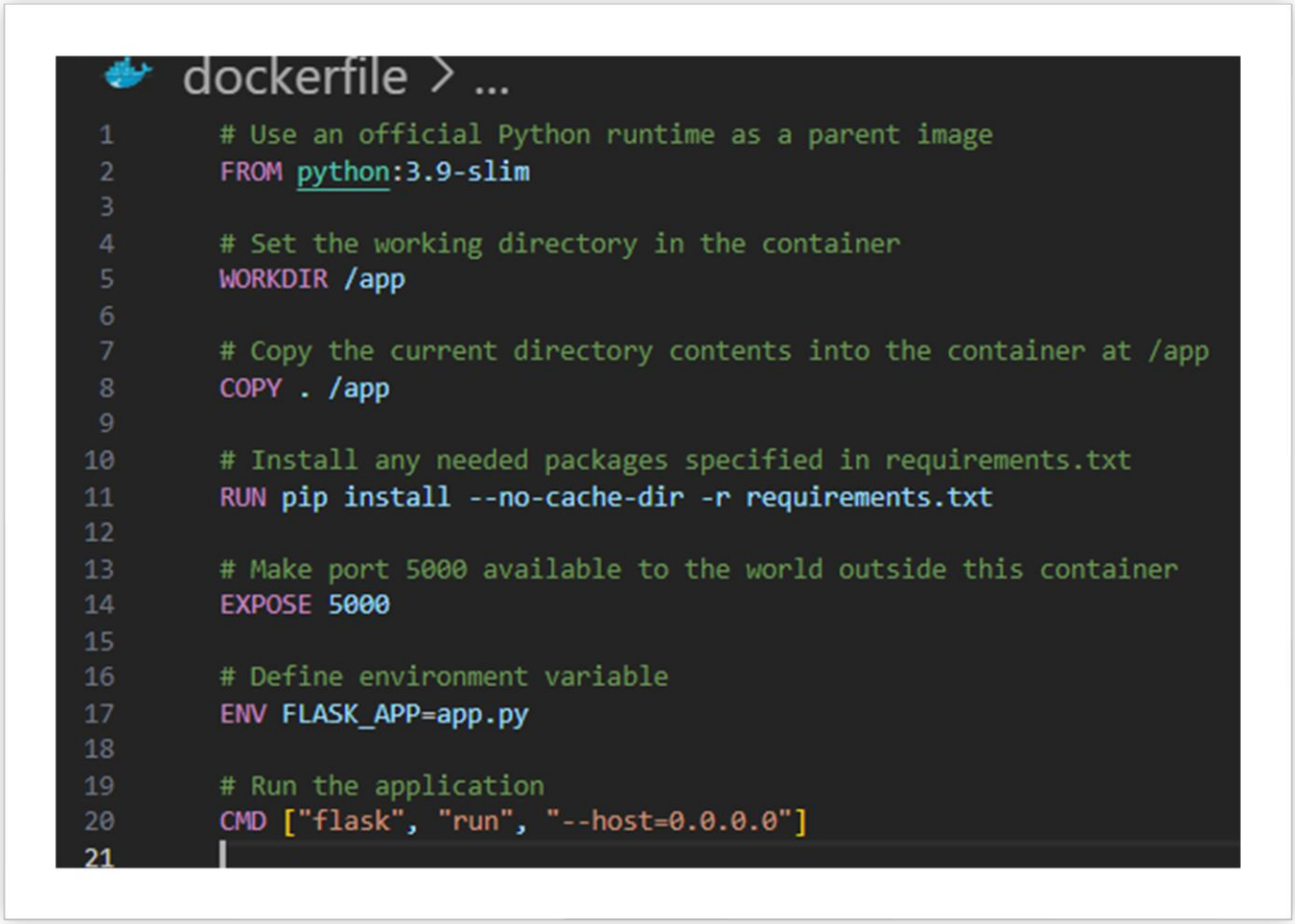
app.py > view_attendance

```python
from flask import Flask, render_template, request, redirect, url_for
import sqlite3
from datetime import datetime
app = Flask(__name__)
# Function to create/connect to the database
def init_db():
    with sqlite3.connect("attendance.db") as conn:
        cursor = conn.cursor()
        cursor.execute('''CREATE TABLE IF NOT EXISTS attendance (
                            id INTEGER PRIMARY KEY AUTOINCREMENT,
                            student_name TEXT NOT NULL,
                            date TEXT NOT NULL,
                            status TEXT NOT NULL
                        )''')
        conn.commit()
# Route to display the attendance page
@app.route('/')
def index():
    return render_template("index.html")
# Route to record attendance
@app.route('/mark_attendance', methods=['POST'])
def mark_attendance():
    if request.method == 'POST':
        student_name = request.form['student_name']
        status = request.form['status']
        date = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

        with sqlite3.connect("attendance.db") as conn:
            cursor = conn.cursor()
            cursor.execute("INSERT INTO attendance (student_name, date, status) VALUES (?, ?, ?)",
                           (student_name, date, status))
            conn.commit()

        return redirect(url_for('index'))
# Route to view attendance records
@app.route('/view_attendance')
def view_attendance():
    with sqlite3.connect("attendance.db") as conn:
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM attendance")
        records = cursor.fetchall()
    return render_template("attendance_records.html", records=records)
if __name__ == '__main__':
    init_db()  # Initialize the database when the app starts
    app.run(debug=True)
```

Dockerfile


dockerfile > ...
Use an official Python runtime as a parent image
FROM python:3.9-slim
Set the working directory in the container
WORKDIR /app
Copy the current directory contents into the container at /app
COPY . /app
Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
Make port 5000 available to the world outside this container
EXPOSE 5000
Define environment variable
ENV FLASK_APP=app.py
Run the application
CMD ["flask", "run", "--host=0.0.0.0"]

Attendance_records.html

templates > attendance_records.html > ...

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Attendance Records</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h1>Attendance Records</h1>

    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Student Name</th>
                <th>Date</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>
            {% for record in records %}
            <tr>
                <td>{{ record[0] }}</td>
                <td>{{ record[1] }}</td>
                <td>{{ record[2] }}</td>
                <td>{{ record[3] }}</td>
            </tr>
            {% endfor %}
        </tbody>
    </table>

    <br>
    <a href="/">Back to Mark Attendance</a>
</body>
</html>
```

Jenkins

```
pipeline {
    agent any

    environment {
        DOCKER_TAG = "latest"   // Docker image tag

    }

    stages {
        stage('Checkout Code') {
            steps {

                git url: 'https://github.com/harshith-kudapali/devops-assignment1.git', branch: 'main'
            }
        }

        stage('Install Dependencies') {
            steps {
                script {
                    bat
'C:\\Users\\harsh\\AppData\\Local\\Programs\\Python\\Python313\\Scripts\\pip.exe install -r
requirements.txt'
                }
            }
        }

        stage('Build Docker Image') {
            steps {
                script {
                    // def imageName = "${DOCKER_IMAGE}".toLowerCase()
                    bat "docker build -t ${DOCKER_IMAGE} ."
                }
            }
        }

        stage('Run Docker Container') {
            steps {
                script {
                     def imageName = "${DOCKER_IMAGE}".toLowerCase()
                     bat "docker run -d -p 5009:5001 ${imageName}"
                }
            }
        }
    }
}
```

Appendices

A. SCREENSHOTS

Running app in CND

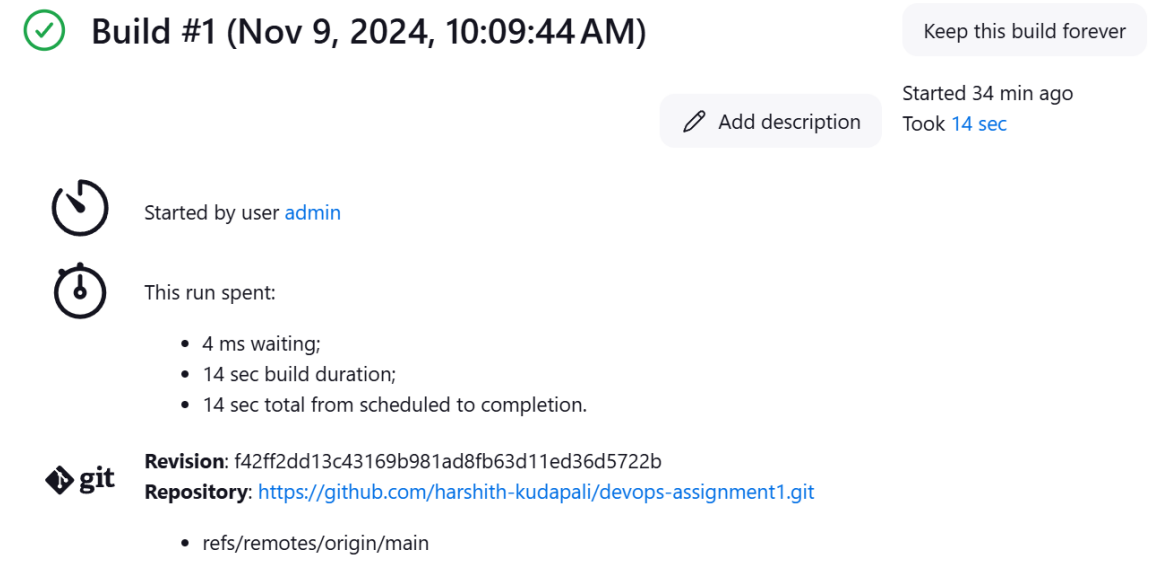
```
PS C:\Users\harsh\Desktop> python -u "c:\Users\harsh\Desktop\app\app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production depl
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 567-807-008
```

Building app in docker and running it

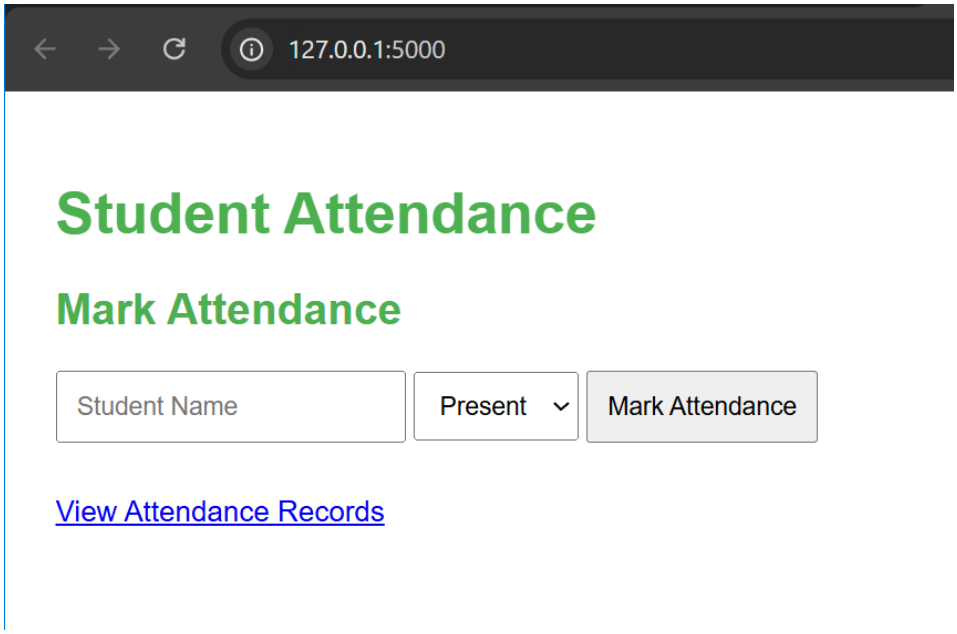
```
PS C:\Users\harsh\Desktop\app> docker build -t flask-attendance-app .
[+] Building 6.3s (10/10) FINISHED                              docker:desktop-linux
 => [internal] load build definition from dockerfile                            0.0s
 => => transferring dockerfile: 570B                                            0.0s
 => [internal] load metadata for docker.io/library/python:3.9-slim              1.7s
 => [auth] library/python:pull token for registry-1.docker.io                   0.0s
 => [internal] load .dockerignore                                               0.0s
 => => transferring context: 2B                                                 0.0s
 => [1/4] FROM docker.io/library/python:3.9-slim@sha256:7a9cd42706c174cdcf578880a  0.0s
 => => resolve docker.io/library/python:3.9-slim@sha256:7a9cd42706c174cdcf578880a  0.0s
 => [internal] load build context                                               0.0s
 => => transferring context: 12.61kB                                            0.0s
 => CACHED [2/4] WORKDIR /app                                                   0.0s
 => [3/4] COPY . /app                                                           0.0s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                    3.6s
 => exporting to image                                                          0.9s
 => => exporting layers                                                         0.5s
 => => exporting manifest sha256:c0d2af2493a08f7f46b9aeeaa9690f36bffda4fb921485d2  0.0s
 => => exporting config sha256:8c86a1f37a01fbddb400d53bd0dffe040a34f7b8394e140644  0.0s
```

```
PS C:\Users\harsh\Desktop\app> docker run -p 5000:5000 flask-attendance-app
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a p
roduction WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
```

Jenkins Build



App-UI



Student Attendance

Mark Attendance

harshith sk | Present | Mark Attendance

Present
Absent

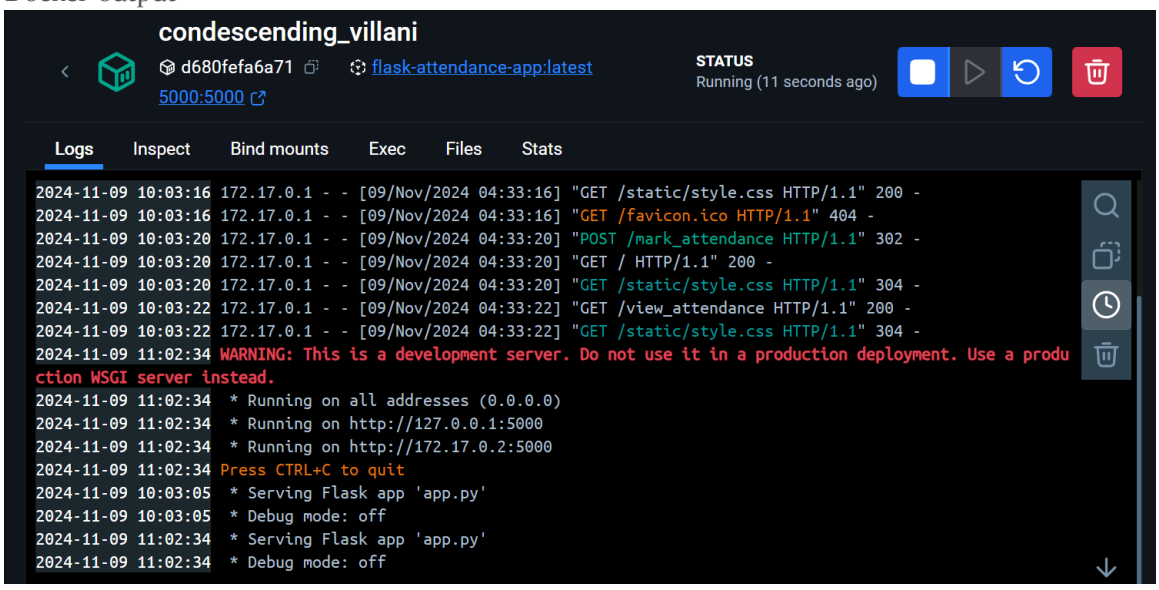
View Attendance Records

Attendance Records

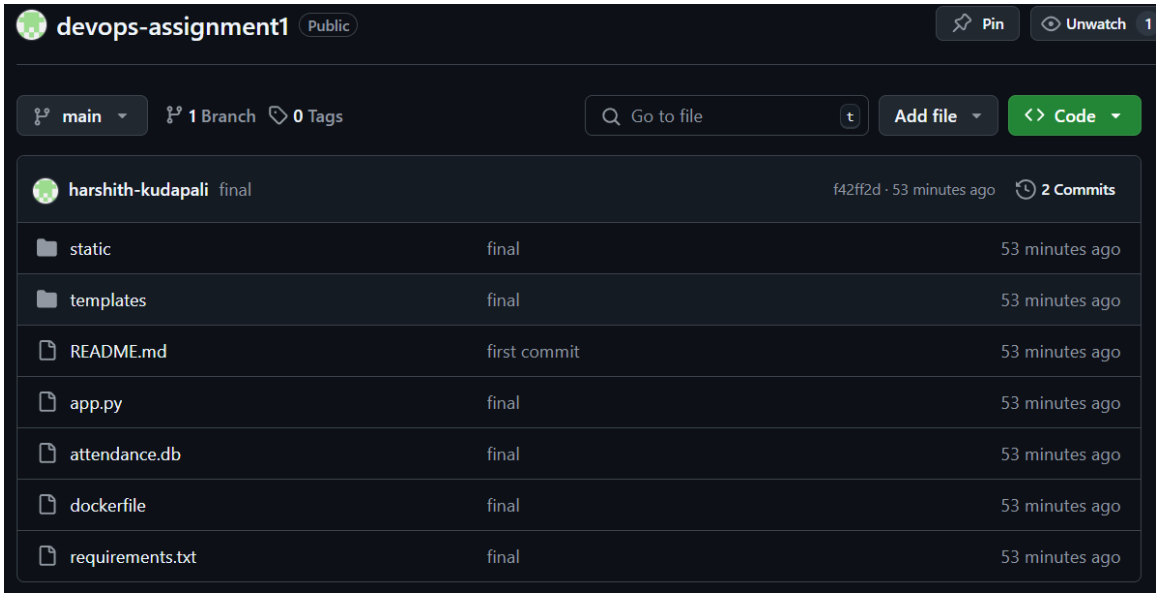
ID	Student Name	Date	Status
1	harshith sk	2024-11-09 10:55:29	Present

Back to Mark Attendance

Docker-output



Github



Thank you