



# 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.

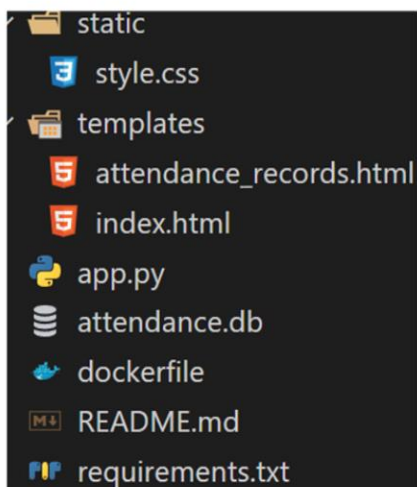
## 1. Building the Flask Application

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.




```
static
style.css
templates
attendance_records.html
index.html
app.py
attendance.db
dockerfile
README.md
requirements.txt
```


## CODE

App.py

```
app.py > view_attendance
1  from flask import Flask, render_template, request, redirect, url_for
2  import sqlite3
3  from datetime import datetime
4  app = Flask(__name__)
5  # Function to create/connect to the database
6  def init_db():
7      with sqlite3.connect("attendance.db") as conn:
8          cursor = conn.cursor()
9          cursor.execute('''CREATE TABLE IF NOT EXISTS attendance (
10                          id INTEGER PRIMARY KEY AUTOINCREMENT,
11                          student_name TEXT NOT NULL,
12                          date TEXT NOT NULL,
13                          status TEXT NOT NULL
14                          )''')
15      conn.commit()
16  # Route to display the attendance page
17  @app.route('/')
18  def index():
19      return render_template("index.html")
20  # Route to record attendance
21  @app.route('/mark_attendance', methods=['POST'])
22  def mark_attendance():
23      if request.method == 'POST':
24          student_name = request.form['student_name']
25          status = request.form['status']
26          date = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
27
28          with sqlite3.connect("attendance.db") as conn:
29              cursor = conn.cursor()
30              cursor.execute("INSERT INTO attendance (student_name, date, status) VALUES (?, ?, ?)",
31                             (student_name, date, status))
32              conn.commit()
33
34      return redirect(url_for('index'))
35  # Route to view attendance records
36  @app.route('/view_attendance')
37  def view_attendance():
38      with sqlite3.connect("attendance.db") as conn:
39          cursor = conn.cursor()
40          cursor.execute("SELECT * FROM attendance")
41          records = cursor.fetchall()
42      return render_template("attendance_records.html", records=records)
43  if __name__ == '__main__':
44      init_db() # Initialize the database when the app starts
45      app.run(debug=True)
```

 **dockerfile** > ...  

```
1      # Use an official Python runtime as a parent image
2      FROM python:3.9-slim
3
4      # Set the working directory in the container
5      WORKDIR /app
6
7      # Copy the current directory contents into the container at /app
8      COPY . /app
9
10     # Install any needed packages specified in requirements.txt
11     RUN pip install --no-cache-dir -r requirements.txt
12
13     # Make port 5000 available to the world outside this container
14     EXPOSE 5000
15
16     # Define environment variable
17     ENV FLASK_APP=app.py
18
19     # Run the application
20     CMD ["flask", "run", "--host=0.0.0.0"]
21
```

templates >  attendance\_records.html > ...

```

1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <title>Attendance Records</title>
7          <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
8      </head>
9      <body>
10         <h1>Attendance Records</h1>
11
12         <table>
13             <thead>
14                 <tr>
15                     <th>ID</th>
16                     <th>Student Name</th>
17                     <th>Date</th>
18                     <th>Status</th>
19                 </tr>
20             </thead>
21             <tbody>
22                 {% for record in records %}
23                 <tr>
24                     <td>{{ record[0] }}</td>
25                     <td>{{ record[1] }}</td>
26                     <td>{{ record[2] }}</td>
27                     <td>{{ record[3] }}</td>
28                 </tr>
29                 {% endfor %}
30             </tbody>
31         </table>
32
33         <br>
34         <a href="/">Back to Mark Attendance</a>
35     </body>
36 </html>
37

```

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 deployment
* 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:c0d2af2493a08f7f46b9aeaa9690f36bffd4fb921485d2 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

✓ Build #1 (Nov 9, 2024, 10:09:44 AM)

Keep this build forever

 Add description

Started 34 min ago  
Took [14 sec](#)



Started by user [admin](#)



This run spent:

- 4 ms waiting;
- 14 sec build duration;
- 14 sec total from scheduled to completion.







**Revision:** f42ff2dd13c43169b981ad8fb63d11ed36d5722b

**Repository:** <https://github.com/harshith-kudapali/devops-assignment1.git>


- refs/remotes/origin/main

App-UI

    127.0.0.1:5000

# Student Attendance

## Mark Attendance



[View Attendance Records](#)



# Student Attendance

## Mark Attendance

Present ▾

Present

Absent

Mark Attendance

[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

condescending\_villani

<  d680fefa6a71  flask-attendance-app:latest

5000:5000 

STATUS  
Running (11 seconds ago)

Logs

Inspect

Bind mounts

Exec

Files

Stats

```
2024-11-09 10:03:16 172.17.0.1 - - [09/Nov/2024 04:33:16] "GET /static/style.css HTTP/1.1" 200 -
2024-11-09 10:03:16 172.17.0.1 - - [09/Nov/2024 04:33:16] "GET /favicon.ico HTTP/1.1" 404 -
2024-11-09 10:03:20 172.17.0.1 - - [09/Nov/2024 04:33:20] "POST /mark_attendance HTTP/1.1" 302 -
2024-11-09 10:03:20 172.17.0.1 - - [09/Nov/2024 04:33:20] "GET / HTTP/1.1" 200 -
2024-11-09 10:03:20 172.17.0.1 - - [09/Nov/2024 04:33:20] "GET /static/style.css HTTP/1.1" 304 -
2024-11-09 10:03:22 172.17.0.1 - - [09/Nov/2024 04:33:22] "GET /view_attendance HTTP/1.1" 200 -
2024-11-09 10:03:22 172.17.0.1 - - [09/Nov/2024 04:33:22] "GET /static/style.css HTTP/1.1" 304 -
2024-11-09 11:02:34 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
2024-11-09 11:02:34 * Running on all addresses (0.0.0.0)
2024-11-09 11:02:34 * Running on http://127.0.0.1:5000
2024-11-09 11:02:34 * Running on http://172.17.0.2:5000
2024-11-09 11:02:34 Press CTRL+C to quit
2024-11-09 10:03:05 * Serving Flask app 'app.py'
2024-11-09 10:03:05 * Debug mode: off
2024-11-09 11:02:34 * Serving Flask app 'app.py'
2024-11-09 11:02:34 * Debug mode: off
```

Github

The screenshot shows a GitHub repository page for 'devops-assignment1' (Public). The repository is on the 'main' branch, with 1 branch and 0 tags. The commit history shows a commit by 'harshith-kudapali' with the message 'final', committed 53 minutes ago. The commit hash is 'f42ff2d'. The repository contains the following files and folders:

File/Folder	Commit Message	Commit Time
static	final	53 minutes ago
templates	final	53 minutes ago
README.md	first commit	53 minutes ago
app.py	final	53 minutes ago
attendance.db	final	53 minutes ago
dockerfile	final	53 minutes ago
requirements.txt	final	53 minutes ago

Thank you