

# Trip Planner

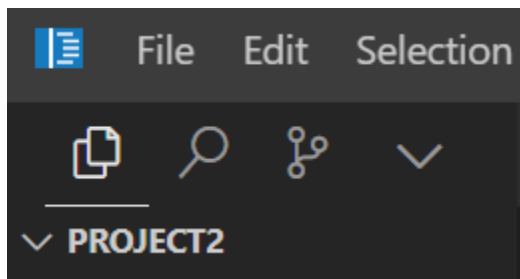
Project Made By

Mukul Sharma-22162581008

Darshan Singh Chauhan-22162581001

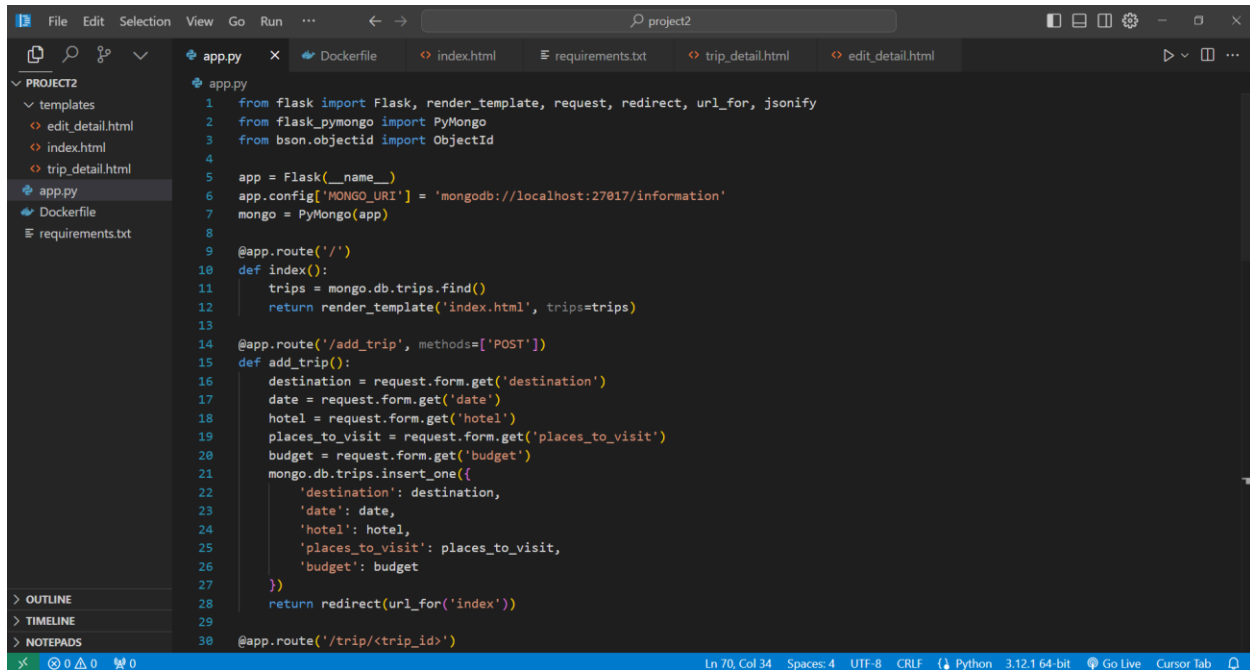
Khushang Patel--=22162581005

Create a project directory:



## Create the Flask App

1. **app.py**: Create the main Flask application.
  - Handles CRUD operations.
  - Connects to MongoDB via Flask-PyMongo.



```
1 from flask import Flask, render_template, request, redirect, url_for, jsonify
2 from flask_pymongo import PyMongo
3 from bson.objectid import ObjectId
4
5 app = Flask(__name__)
6 app.config['MONGO_URI'] = 'mongodb://localhost:27017/information'
7 mongo = PyMongo(app)
8
9 @app.route('/')
10 def index():
11     trips = mongo.db.trips.find()
12     return render_template('index.html', trips=trips)
13
14 @app.route('/add_trip', methods=['POST'])
15 def add_trip():
16     destination = request.form.get('destination')
17     date = request.form.get('date')
18     hotel = request.form.get('hotel')
19     places_to_visit = request.form.get('places_to_visit')
20     budget = request.form.get('budget')
21     mongo.db.trips.insert_one({
22         'destination': destination,
23         'date': date,
24         'hotel': hotel,
25         'places_to_visit': places_to_visit,
26         'budget': budget
27     })
28     return redirect(url_for('index'))
29
30 @app.route('/trip/<trip_id>')
```

### 3. Templates Directory

1. Create the templates folder and add the following files:

#### **index.html**

Handles trip listings and adding new trips.

#### **trip\_detail.html**

Displays trip details.

#### **edit\_detail.html**

Displays and edits trip information.

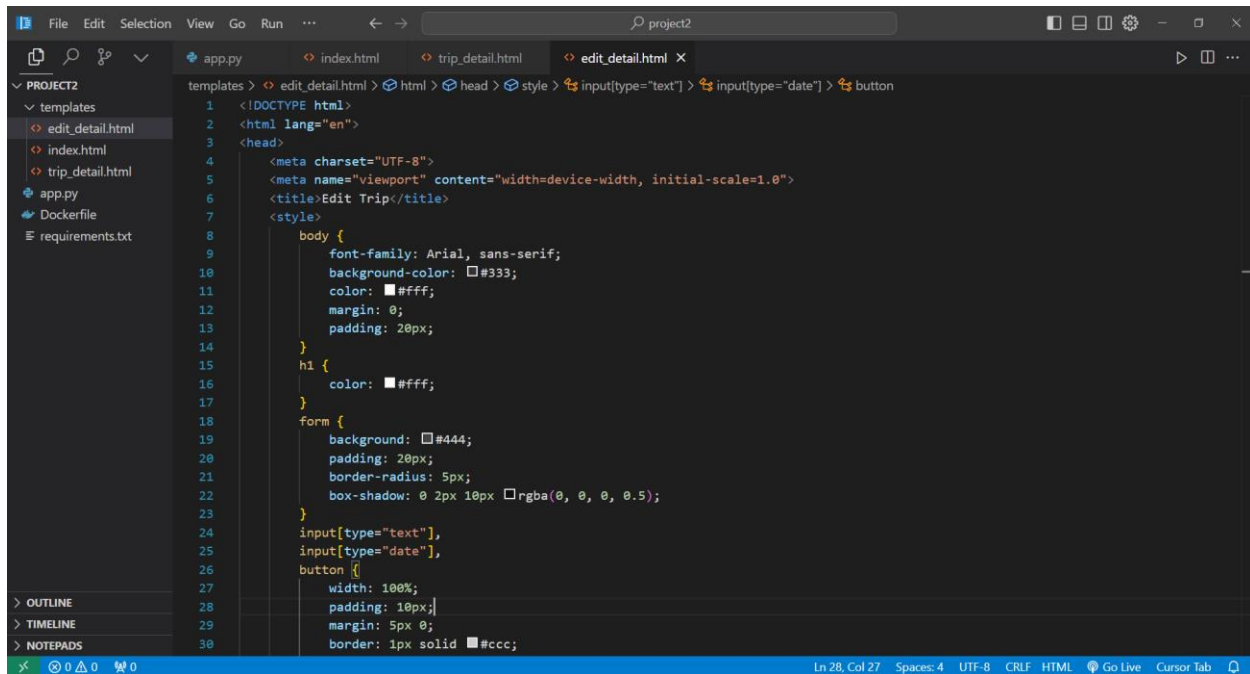
The image displays two screenshots of a Visual Studio Code editor window, showing the development of a web application. The top screenshot shows the `index.html` file, which includes a head section with a DOCTYPE declaration, language attribute, charset, viewport meta tag, and title "Travel Planner". The body contains a form with a text input and a date input, styled with a dark background and light text. The bottom screenshot shows the `trip_detail.html` file, which includes a head section with a DOCTYPE declaration, language attribute, charset, viewport meta tag, and title "Trip Detail". The body contains a `.trip-info` section with a background, padding, border-radius, and box-shadow, and a `button` with a background-color, color, border, padding, cursor, and border-radius.

```
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>Travel Planner</title>
7   <style>
8     body {
9       background-color: #121212; /* Dark background */
10      color: #ffffff; /* Light text color */
11      font-family: Arial, sans-serif; /* Font style */
12      margin: 0;
13      padding: 20px;
14    }
15    h1, h2 {
16      color: #ffcc00; /* Highlighted heading color */
17    }
18    form {
19      background-color: #1e1e1e; /* Form background */
20      padding: 20px;
21      border-radius: 8px; /* Rounded corners */
22      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Shadow effect */
23      margin-bottom: 20px;
24    }
25    input[type="text"], input[type="date"] {
26      width: calc(100% - 22px); /* Full width minus padding */
27      padding: 10px;
28      margin: 10px 0;
29      border: 1px solid #444; /* Border color */
30      border-radius: 4px; /* Rounded corners */

```

```
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>Trip Detail</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      background-color: #333;
11      color: #fff;
12      margin: 0;
13      padding: 20px;
14    }
15    h1 {
16      color: #fff;
17    }
18    .trip-info {
19      background: #444;
20      padding: 20px;
21      border-radius: 5px;
22      box-shadow: 0 2px 10px rgba(0, 0, 0, 0.5);
23    }
24    button {
25      background-color: #5cb85c;
26      color: white;
27      border: none;
28      padding: 10px;
29      cursor: pointer;
30      border-radius: 5px;

```



## dependencies

1. Install Flask, Flask-PyMongo, and other required packages:

bash

Copy code

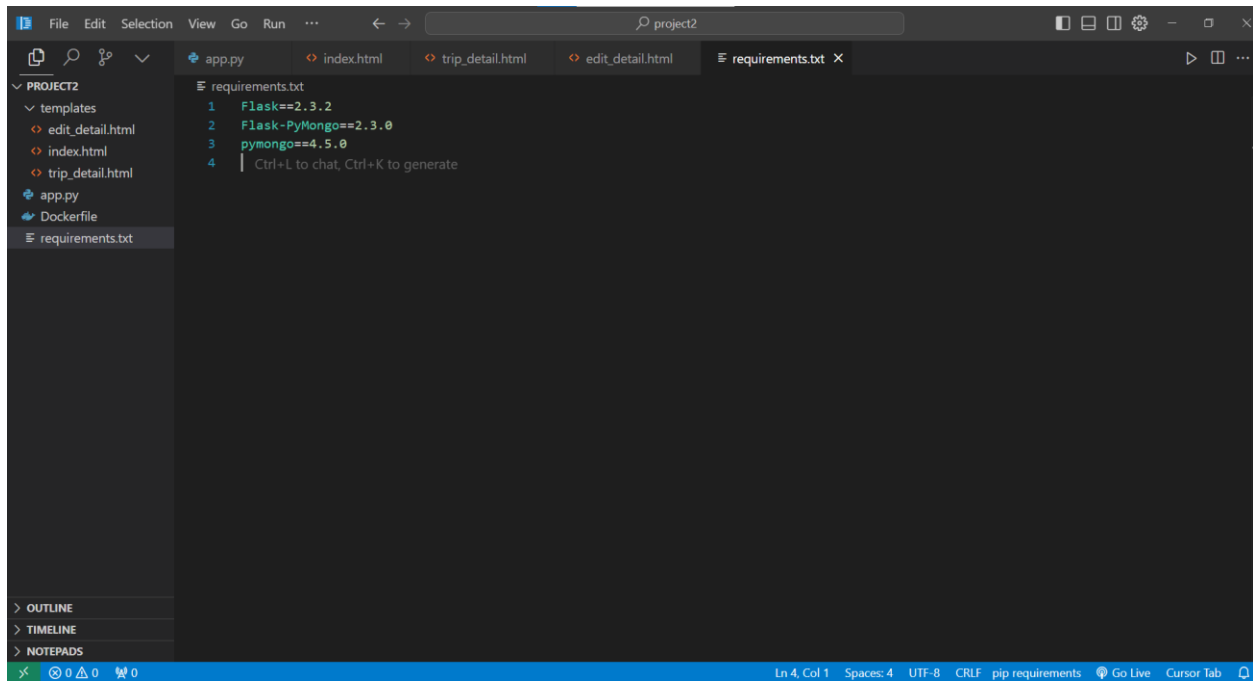
pip install flask flask-pymongo pymongo

2. Save dependencies in requirements.txt:

bash

Copy code

pip freeze > requirements.txt



## Run the Application

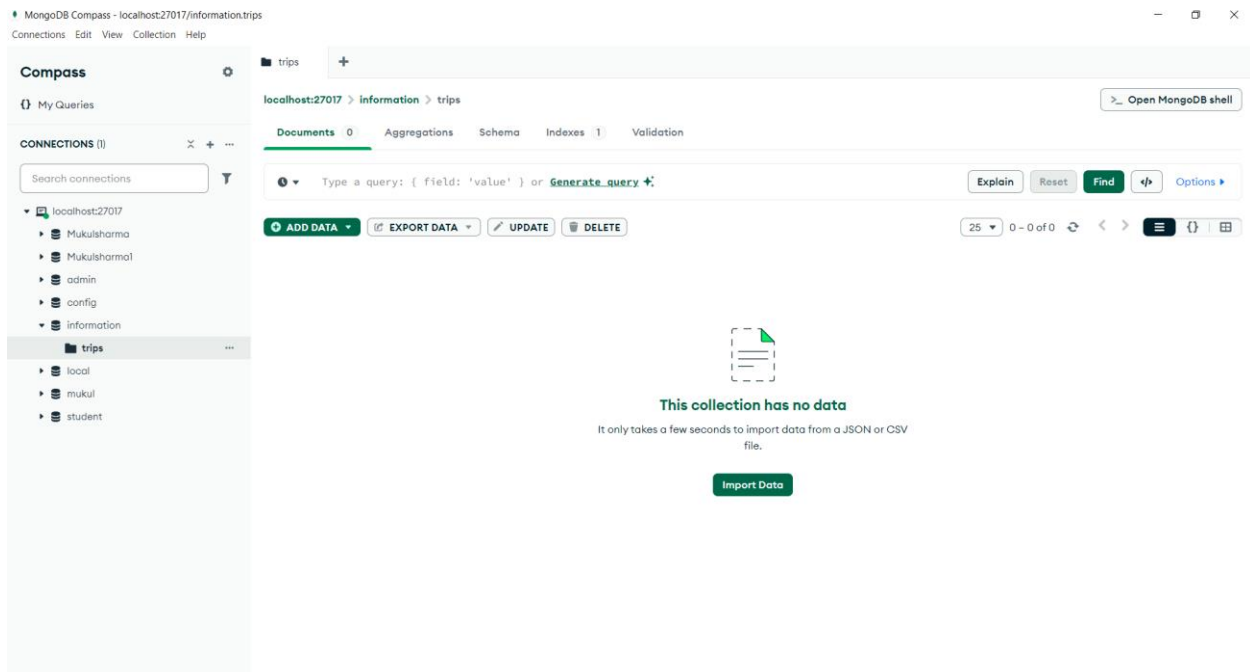
1. Start MongoDB on localhost:27017.
2. Run the Flask app:

bash

Copy code

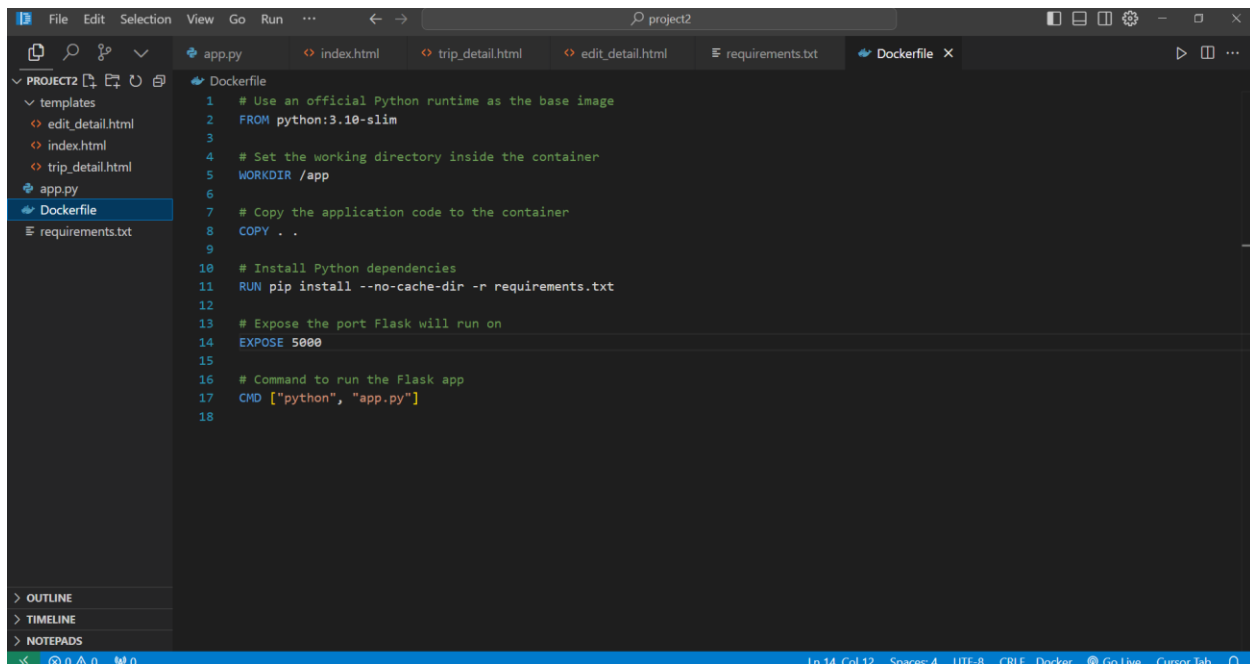
python app.py

3. Visit <http://localhost:5000> to access the application.



## Create Dockerfile

A Dockerfile to containerize the application.



## 7. Build and Run the Docker Container

1. Build the Docker image:

bash

```
docker build -t flask-travel-planner .
```

bash

```
docker run -d -p 3000:5000 flask-travel-planner
```

The screenshot shows the Visual Studio Code interface with a file explorer on the left containing files like templates, index.html, trip\_detail.html, app.py, Dockerfile, and requirements.txt. The main editor displays a Dockerfile with two instructions: # Use an official python runtime as the base image and FROM python:3.10-slim. Below the editor, the TERMINAL tab is active, showing the execution of docker login, docker build -t mukul2107/flask-travel-planner, and docker:desktop-linux. The build process logs show various steps like loading definitions, transferring files, and installing dependencies. At the bottom, there are status bars indicating 'Ln 3, Col 1' and other window details.

```
# PROJECT2 [Python]
templates
edit_detail.html
index.html
trip_detail.html
app.py
Dockerfile
requirements.txt
```

```
Dockerfile
1 # Use an official python runtime as the base image
2 FROM python:3.10-slim
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS D:\SEM 5\Microservices\project2> docker login
>>
Authenticating with existing credentials...
Login Succeeded

PS D:\SEM 5\Microservices\project2> docker build -t mukul2107/flask-travel-planner .
[*] Building 8.9s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 435B                                              0.1s
=> [internal] load metadata for docker.io/library/python:3.10-slim              7.4s
=> [auth] library/python:pull token for registry-1.docker.io                   0.0s
=> [internal] load .dockerignore                                                 0.1s
=> => transferring context: 2B                                                  0.0s
=> [1/4] FROM docker.io/library/python:3.10-slim@sha256:af6f1b19eae3408ea3a569ba92d4819a527be4662971d51bb798c923bba38a81 0.0s
=> [internal] load build context                                                0.1s
=> => transferring context: 2.93kB                                             0.0s
=> CACHED [2/4] WORKDIR /app                                                    0.0s
=> CACHED [3/4] COPY . .                                                        0.0s
=> CACHED [4/4] RUN pip install --no-cache-dir -r requirements.txt             0.0s
=> exporting to image                                                            0.1s
=> => exporting layers                                                         0.0s
=> writing image sha256:10a6e89f4df34a7db9d9b6c38672e96804c5c653037a53e51b095d399042c3d 0.0s
=> naming to docker.io/mukul2107/flask-travel-planner                        0.0s
```

```
What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview

PS D:\SEM 5\Microservices\project2>
```

The screenshot displays the Docker Desktop application window. The top navigation bar includes 'Containers', 'Images', 'Volumes', 'Builds', 'Docker Scout', and 'Extensions'. The 'Containers' tab is active, showing a summary of container usage: CPU at 1.92% (400% available) and memory at 50.62MB (1.81GB available). Below this is a table of containers. The 'travel-planner' container is selected, showing its details: Name 'travel-planner', ID '388152abc3ff', Image 'flask-travel-plan', Port '3000:5000', CPU usage '0.33%', and '56 minutes ago' since last started. The bottom panel shows the terminal output for the 'travel-planner' container, displaying the command 'docker run -p 3000:5000 --name travel-planner mukul2107/flask-travel-planner' and the resulting logs, including a warning about using a development server in production.



Travel Planner

127.0.0.1:5000

## Travel Planner

Add Trip

### Your Trips

**Destination:** jammu  
**Date:** 2025-01-01  
**Hotel:** xyz  
**Places to Visit:** srinagar  
**Budget:** 1000000  

EditDelete

The screenshot displays the MongoDB Compass web interface. On the left, the 'CONNECTIONS' sidebar shows a list of databases and collections, with 'trips' selected under the 'information' database. The main panel shows the 'Documents' tab with a single document in the 'trips' collection. The document is a JSON object with the following fields: `_id` (ObjectId), `destination` (string), `date` (string), `hotel` (string), `places_to_visit` (string), and `budget` (string). A modal window is open, displaying the document details in a dark theme. The modal contains the same JSON object and two buttons: 'Edit' and 'Delete'.

```
{
  "_id": "ObjectId('674ddc63ff61837373549477')",
  "destination": "jammu",
  "date": "2025-01-01",
  "hotel": "xyz",
  "places_to_visit": "srinagar",
  "budget": "100000"
}
```

Destination: mumbai  
Date: 2025-01-01  
Hotel: xyz  
Places to Visit: srinagar  
Budget: 100000

Edit Delete

Pushed To repository

PROJECT2

templates

edit\_detail.html

index.html

trip\_detail.html

app.py

Dockerfile

requirements.txt

OUTLINE

TIMELINE

NOTEPADS

Dockerfile

1 # Use an official Python runtime as the base image

2 FROM python:3.10-slim

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS D:\SEM 5\Microservices\project2> docker login

>>

Authenticating with existing credentials...

Login Succeeded

PS D:\SEM 5\Microservices\project2> docker push mukul2107/flask-travel-planner

Using default tag: latest

The push refers to repository [docker.io/mukul2107/flask-travel-planner]

b564dec8ee26: Pushed

d8c5906b9920: Pushed

0ce8ca9d906c: Pushed

1062cd4d071e: Mounted from library/python

e5d8b619f2ce: Mounted from library/python

6a4c9147f069: Mounted from library/python

c3548211b826: Mounted from library/python

latest: digest: sha256:b3b4cd9f2ccac0da6c23aa9bc1dc1a4a52e794a7ad318abe021e9aa9c670f36b size: 1784

PS D:\SEM 5\Microservices\project2>

Docker Hub Container Image 1

hub.docker.com

New

More Docker. Easy Access. New Streamlined Plans. Learn more. →

dockerhub

Explore

Repositories

Organizations

Usage

Search Docker Hub

ctrl+K

Go Live

Cursor Tab

mukul2107

Search by repository name

All content

Create a repository

Name	Last Pushed ↑	Contains	Visibility	Scout
mukul2107/flask-travel-planner	1 minute ago	IMAGE	Public	Inactive
mukul2107/data	17 days ago	IMAGE	Public	Inactive
mukul2107/my_app	19 days ago	IMAGE	Public	Inactive
mukul2107/my_flask_app	19 days ago	IMAGE	Public	Inactive
mukul2107/mukul21	24 days ago		Public	Inactive

1-5 of 5

[Create an organization](#)

[Create and manage users and grant access to your repositories.](#)

https://hub.docker.com/repository/docker/mukul2107/flask-travel-planner