Quick Trip planning Assistant from New York

Presented by,
Soorya Nivedha Ashokan

Outline:

- Abstract
- Introduction
- Motivation
- How this works?
- About Django
- Directory Structure
- Web application Snippet
- Demo
- Future Enhancement

Abstract

In this project I have created a web application that quickly provide an overall idea about the destination the user plans to visit anywhere in the world from New York having direct flights.

The application would provide information about weather, closest airports and cheapest air fares for destination with no layover or hops.

Based on the airfares the user could decide their vacation destination as per their budget with minimum travel time.

Also with the knowledge of weather they could plan their activity during the vacation.

INTRODUCTION

The aim of this project is to provide user a quick and handy information about the destination they dream to visit and to make their planning profitable.

Input from the user:

- Destination: where to go
- Travel start date : departure date
- Travel End date : arrival date

Output:

- Weather condition of the destination
- Closest Airports and city
- Current air fares for the direct flights, if there is no direct flights then displays nothing(enhanced this for hops now, but not updated in github)

Motivation

Most of us love travelling, whenever we begin to plan a vacation, the main thing we would consider is how much the trip would cost and how satisfactory our trip would be, if we travel in particular period of the year.

Because psychologically season and weather has effects on travel related mood and travel satisfaction[1]. So here I took two parameters such as weather and air fare mainly to get a basic idea before planning a trip.

How this Works?

- Used Django 2.2 framework to create web application for travel recommendation
- Used Django forms to get user input - Location, Date range
- Used weather API to obtain temperatures for the given date range
- Used flight API to identify closest airports to the destination
- Also check for the cheapest flight available for the travel period

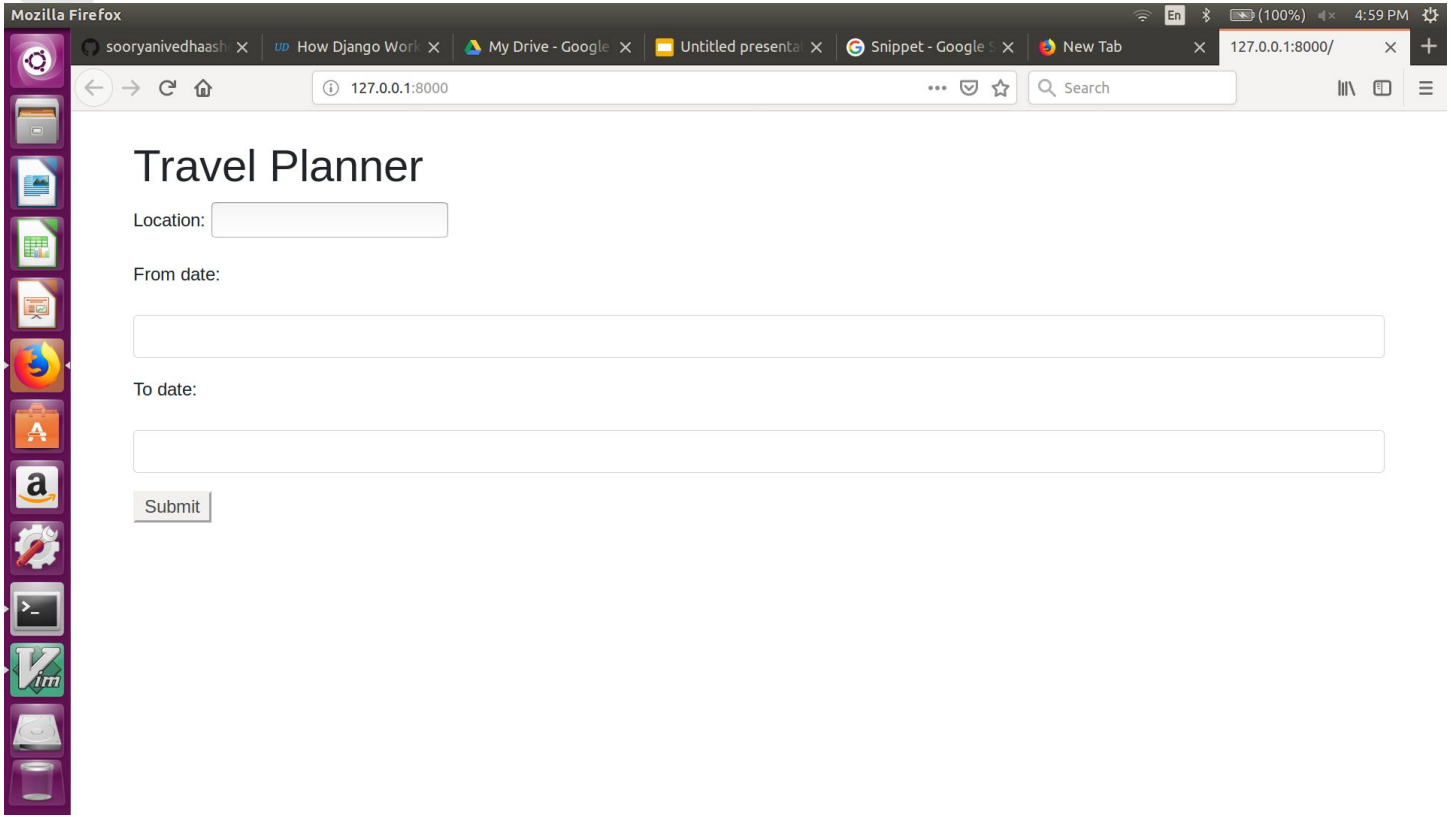
About Django

- Django can accept requests for URLs like www.example.com, and return all of the HTML needed for a web browser to render a page. That page could be plain text or something much richer.
- **Web Requests Enter Django Applications via URLs:** When a user accesses a URL, Django will pass it to a view for processing.
- **Requests are processed by views:** Django Views are custom Python code that get executed when a certain URL is accessed. Once a view is done processing, a web response is provided back to the user.
- **Web responses are returned:** When a user accesses a URL in a browser, what is shown in the window is the web response. Most often this is a HTML web page, showing a combination of text and images. These pages are created using Django's templating system.

Directory Structure

```
|__mysite              # Django Project created called "mysite"
  |__kunju             # Application created inside Django Project
    |__ migrations
    |__ templates
    |__ tests
      |__ test.py   # Test command: (Top) cd mysite , $python3 -m unittest kunju/tests/test.py
      |__ test_air.py
      |__ test_flight.py
      |__ test_geolocate.py
      |__ test_weather.py
    |__ air.py
    |__ apps.py
    |__ flight.py
    |__ forms.py
    |__ geolocate.py
    |__ urls.py
    |__ views.py
    |__ weather.py
    |__ widgets.py
  |__ manage.py
  |__ requiement.txt    # contains all the required installation to access the application
  |__ test              # Executable file for quick testing of the application
  |__ mysite            # Django Project created called "mysite", folder contains required file for project to run
    |__ settings.py
    |__ urls.py
    |__ wsgi.py
|__README.md
```

Web Application Snippet



Future Enhancement

We could enhance the project by simulating various models for forecasting the future air fares and classifying whether to buy or wait for the reduction in air ticket fare based on historical trends.

THANK YOU !!