# A Project Report on

**WEATHER PREDICTION**

Submitted by **Ashwati Rao(18IT2018) Sagar Nirmal(18IT1095) Sahil Satav(18IT1007)**

Under the guidance of

**Ms. Anita Senathi Assistant Professor**

**Department of Information Technology Ramrao Adik Institute of Technology**

**Nerul ,Navi Mumbai**

**Certificate**

This is to certify that the project entitled **Weather Prediction** is being submitted to the Department of Information Technology, Ramrao Adik Institute of Technology,Navi Mumbai.

Project Guide External

Examiner(Ms.Anita Senathi) ( )

# Acknowledgement

We owe our gratitude to many people who have supported us throughout this journey.

We would,firstly like to express our heartfelt gratitude towards our respected Principal Dr. Mukesh D Patil and our Head of Department Dr. Ashish Jadhav for providing us immense facilities, guidance and never ending support.

The completion of any inter-disciplinary project depends upon cooperation and combined efforts of several sources of knowledge. We take this opportunity to express our profound gratitude and deep regards to our guide Ms.Anita Patil for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

Lastly, we thank our parents, family, friends and well wishers who always looked for the chance to help us in whatever means came forth and for their constant encouragement without which the project would not be a distant reality.

# Introduction

Weather Report project application is a web based application through which you will able to get all the reports related to weather forecasting of any locations. Its geographical locator which will be received through your browser setting and server configuration will identify the location and able to present its weather details such as temperature and atmospheric conditions.

This project’s main idea is to develop a website which will predict the weather of a place.

# Proposed System

The main objective of the proposed system is to notify the user about the weather conditions of a place. The user can not only view the weather of just one place but add many places to the city list. So, the user will be able to view the weather conditions of many places simultaneously.

# System Components

1. **Frontend**: HTML, Bulma CSS

**2.Backend** :Python,Django framework

## HTML:

HTML is HyperText Mark-up Language used to documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. User interface is designed using HTML CSS and Bootstrap.

## CSS:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

## Bulma CSS:

The Bulma framework is a free CSS solution based on the Flexbox layout. With Bulma, the extensive range of built-in features means faster turnaround and less CSS code writing.

## 2.1Django(2.0.3):

Django is a free and open-source web framework, written in Python, which follows the model-view-template architectural pattern.In our project we have used this framework to host our leave management system in which users will login to apply for leave that will be approved or not approved by admin.

Django is a widely-used Python web application framework with a "batteries-included" philosophy. The principle behind batteries-included is that the common functionality for building web applications should come with the framework instead of as separate libraries.

Authentication, URL routing, a template engine, an object-relational mapper (ORM), and database schema migrations are all included with the Django framework. Compare that included functionality to the Flask framework which requires a separate library such as Flask-Login to perform user authentication.

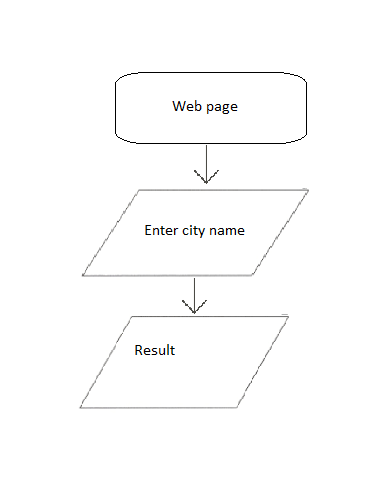
Companies, organizations and governments have used Django to build all sorts of things — from content management systems to social networks to scientific computing platforms.

## 2.3 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a

scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. The fast edit-test-debug cycle makes this simple approach very effective.

# Architecture



Weather Prediction

**WORKING OF SYSTEM**

The Weather Prediction website consists of an entry box in which a person has to enter the name of the desired city. After clicking enter, it will find out the weather conditions of that particular city. It shows the temperature as well as atmospheric conditions of that place.

The user can add as many cities as he wants. The names of all cities along with weather conditions shall be stored in a list.

# Django Source Code

## Views.py

import requests

from django.shortcuts import render

from . models import City

from .forms import CityForm

def home(request):

url = 'http://api.openweathermap.org/data/2.5/weather?q={}&units=imperial&appid=1f0c363da4ab5e493b3055eeb5aca54b'

err\_msg = ''

if request.method == 'POST':

form = CityForm(request.POST)

if form.is\_valid():

new\_city = form.cleaned\_data['name']

existing\_city\_count = City.objects.filter(name=new\_city).count()

if existing\_city\_count == 0:

form.save()

else:

err\_msg = 'city exists'

form = CityForm()

cities = City.objects.all()

weather\_data = []

for city in cities:

r= requests.get(url.format(city)).json()

city\_weather= {

'city' :city,

'temperature' : r['main']['temp'],

'description' : r['weather'][0]['description'],

'icon' : r['weather'][0]['icon'],

}

weather\_data.append(city\_weather)

context = {'weather\_data' : weather\_data, 'form': form}

return render(request, 'home.html', context)

## Models.py

from django.db import models

class City(models.Model):

name = models.CharField(max\_length=25)

def \_\_str\_\_(self):

return self.name

class Meta:

verbose\_name\_plural = 'cities

1. Urls.py

from django.urls import path

from . import views

urlpatterns = [

path('', views.home, name="home"),

]

1. Admin.py

from django.contrib import admin

from . models import City

admin.site.register(City)

1. Apps.py

from django.apps import AppConfig

class WeatherConfig(AppConfig):

name = 'weather'

1. Forms.py

from django.forms import ModelForm, TextInput

from .models import City

class CityForm(ModelForm):

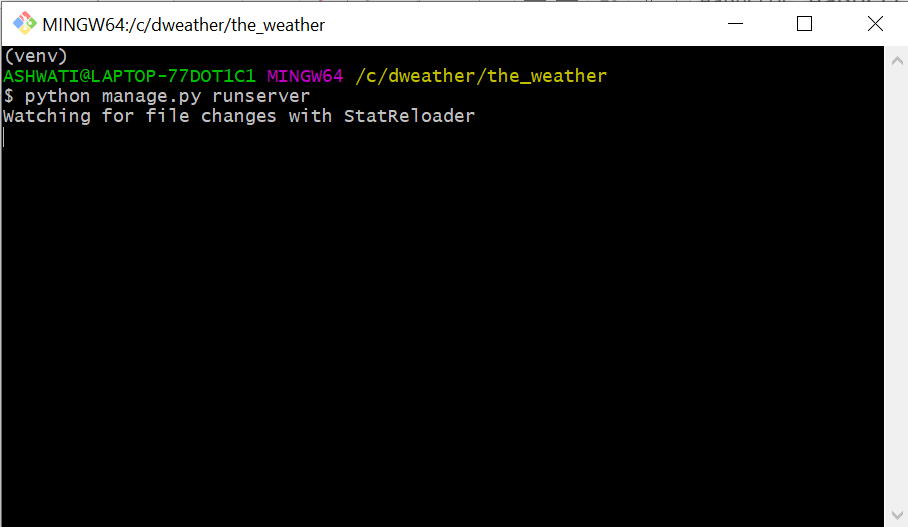
class Meta:

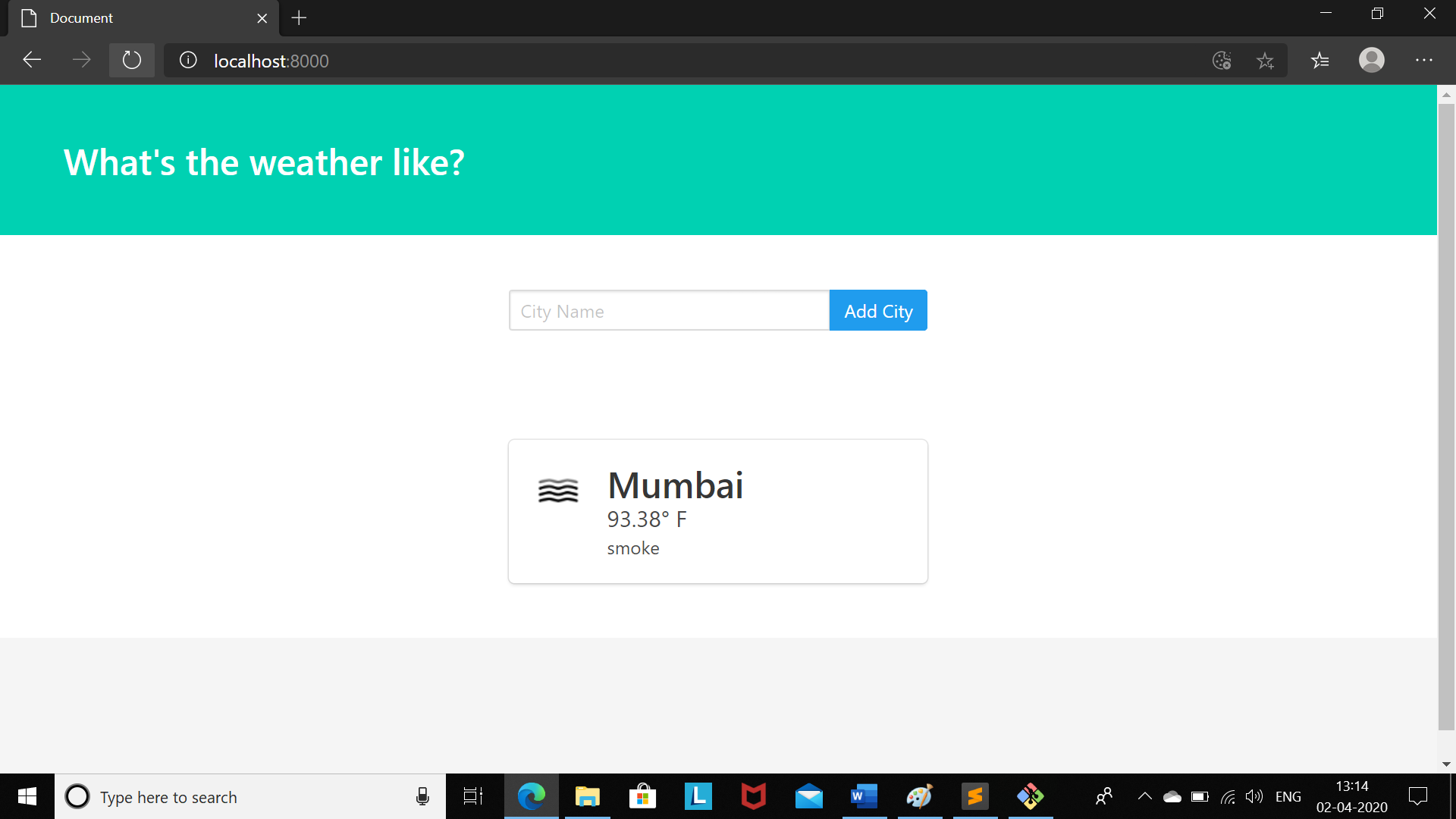
model=City

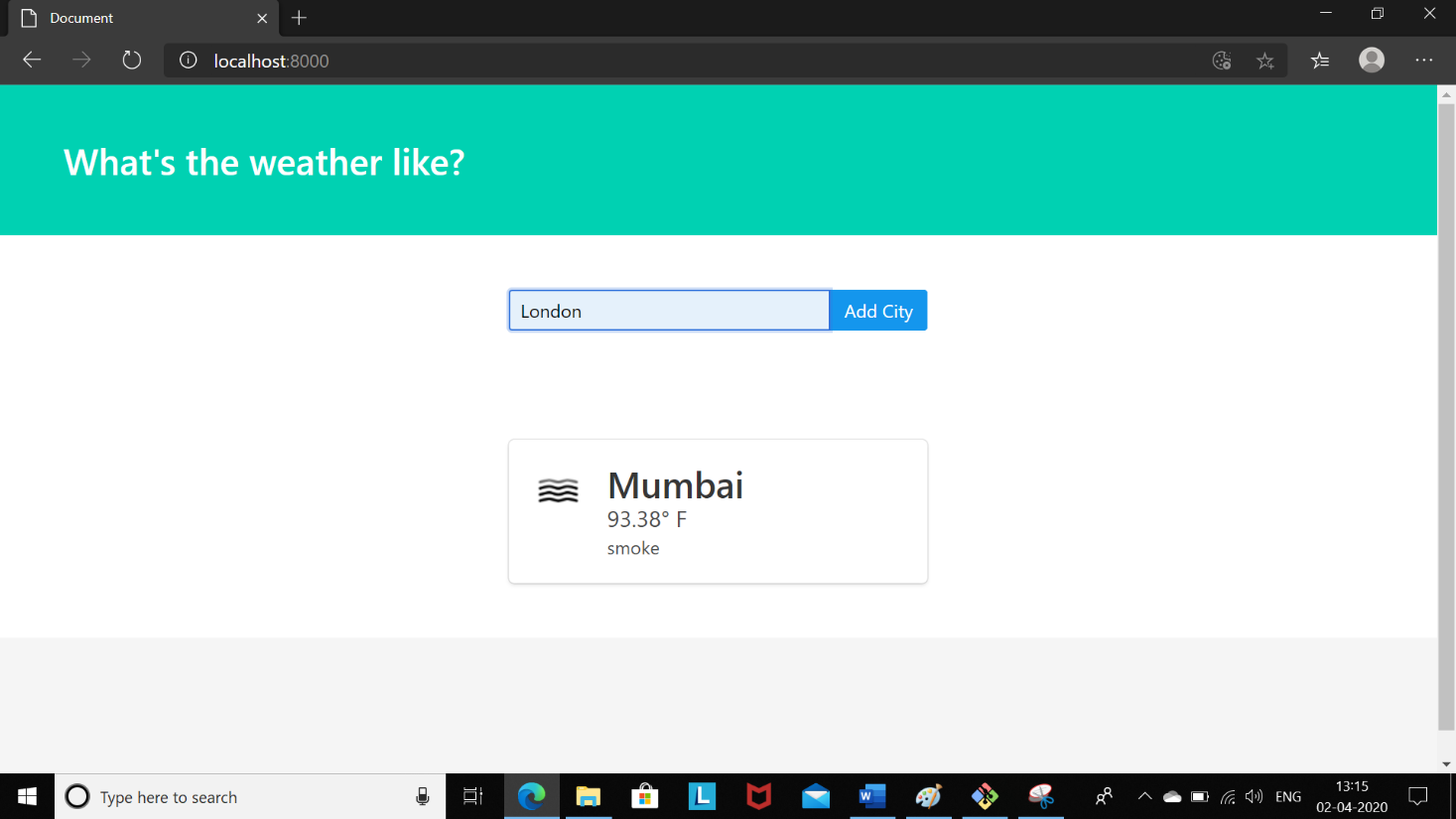
fields= ['name']

widgets = {'name' : TextInput(attrs={'class' : 'input','placeholder' : 'City Name'})}

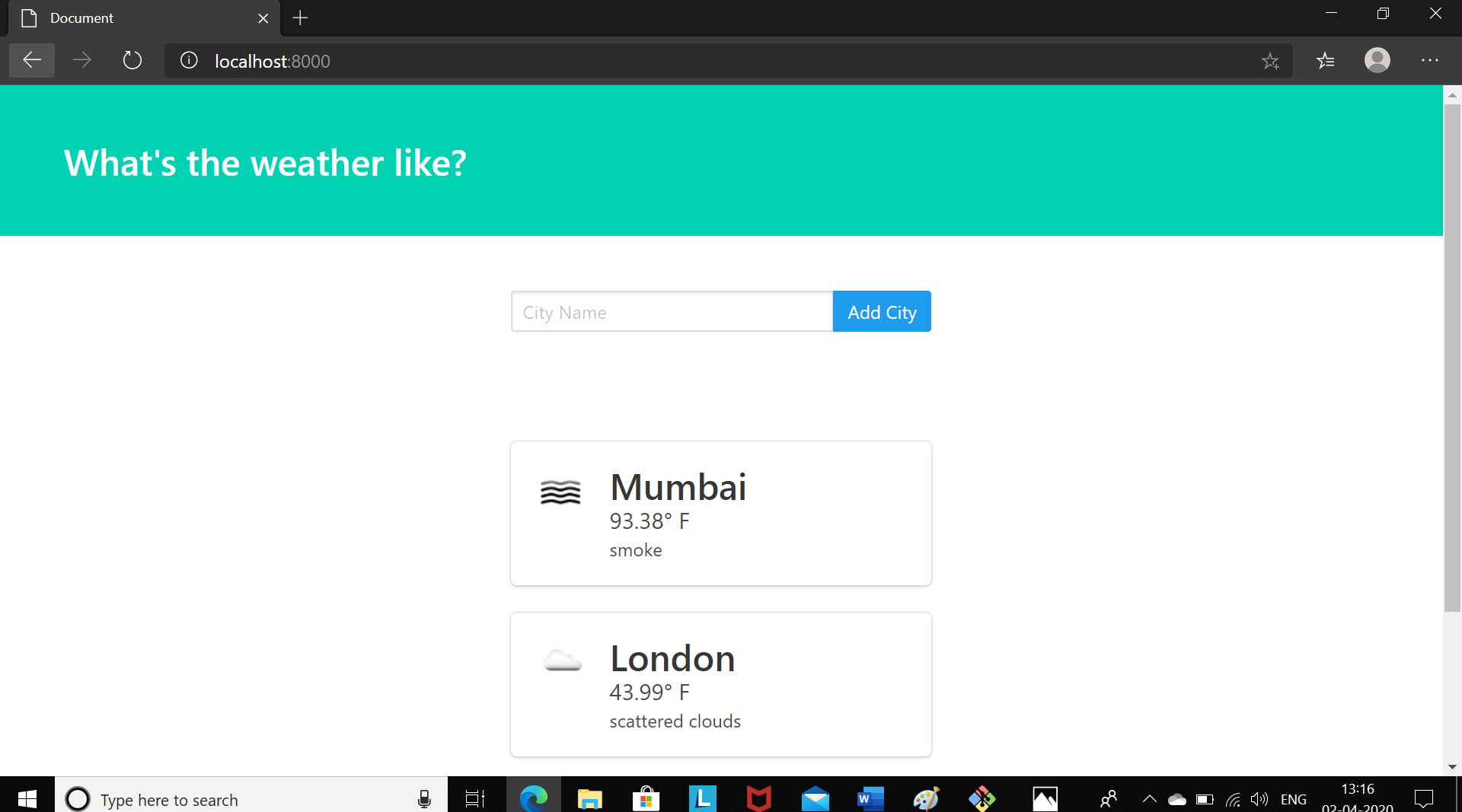
# Implementation of Web Application

* 1. Start the Django Web Server. 
  2. To check the server is running or not type localhost:8000 in browser(using Google chrome here) .If the server is running the staring page of website will be displayed.



* 1. Then type the name of the desired city.

4.Then click on ‘Add City’.



**Conclusion**

Weather Prediction is very useful for people to keep a track of weather conditions of different places.

# References

* Django documentation (2.0.3)
* The Definitive Guide to Django Web Development Done Right by Adrian Kaplan-Moss, Jacob
* Telusko Youtube
* freeCodeCamp.org Youtube