Cloud Computing Laboratory

Assignment No. 2

Name : Satyajeet Prithviraj Chavan

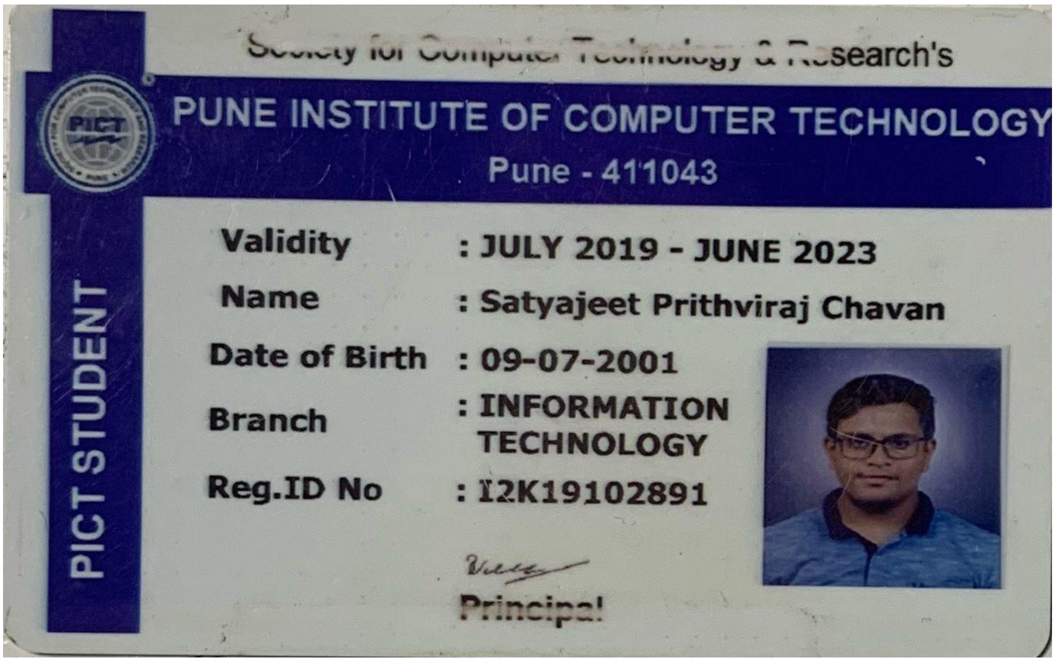
Roll no: 33308

Batch: L-11

Contact No: 8788496040

Email Id : satyajeetmanjusha@gmail.com

ID Card :



Society for Computer Technology & Research's

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Pune - 411043

PICT STUDENT

Validity	: JULY 2019 - JUNE 2023
Name	: Satyajeet Prithviraj Chavan
Date of Birth	: 09-07-2001
Branch	: INFORMATION TECHNOLOGY
Reg.ID No	: I2K19102891

Principal

Name - Satyajeet Chavan
Roll no - 33308

Cloud Computing Assignment No. 2

* Aim: Use GAE launcher to launch the web applicatio

* Theory:

① Mention the structure of GAE applications with their file structure and description?

⇒ ① Each version of your app Engine service is defined in an app.yaml file.

② For similar apps, the minimum requirement for deployment is to define the app.yaml file.

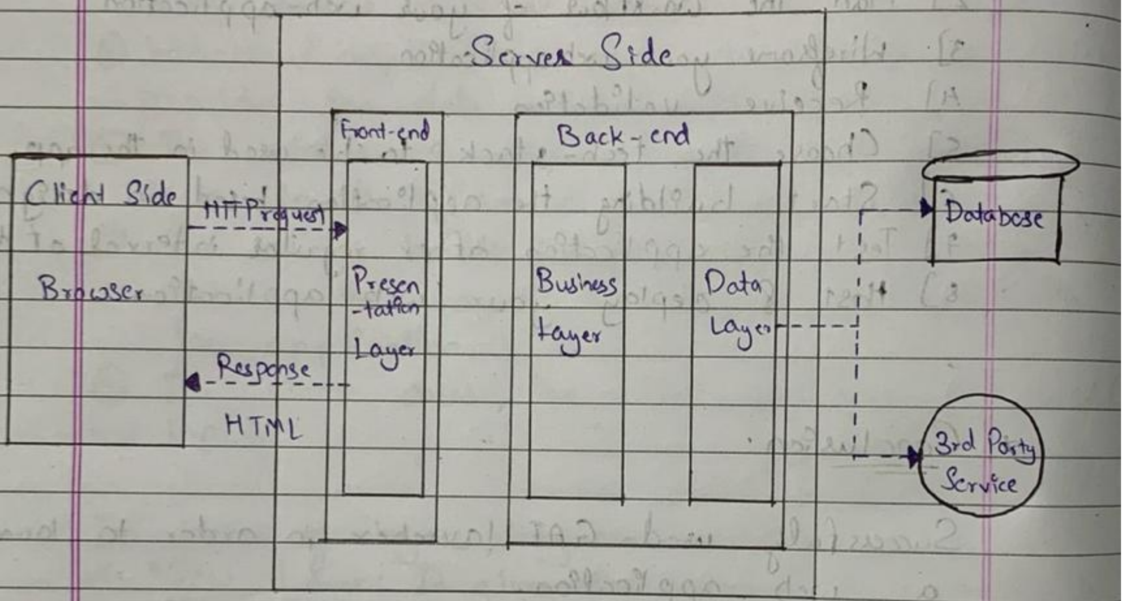
③. If you are deploying several versions of a service, you can create multiple yaml files in the same directory to represent configuration for each versions

④ For each service, you can create separate directories in the root of your app when you are developing locally.

⑤ Each directory or repository should represent a single service & contain that service's app.yaml file along with the associated source code.

⑥ The other optional configuration files should reside in the ~~rest~~ root directory or repository of the default service of your app.

⑦ In this assignment, our project directory includes app.yaml, index.html, main.py, results.html


Server Side
Front-end
Back-end
Client Side
Browser
HTTP request
Response
HTML
Presen-tation Layer
Business Layer
Data Layer
Database
3rd Party Service

(2) What are the main components of web application?

=) All web-based database applications have three primary components: A web browser, a web application server and a database server.

(1) Web-based applications rely on database server, which provides the data for the application.

(2) The database sometimes provide with business logic in the form of stored procedure.

(3) The clients handle the presentation logic, which controls the way in which users interact with the application.

(4) In some cases, the client validates user provided inputs.

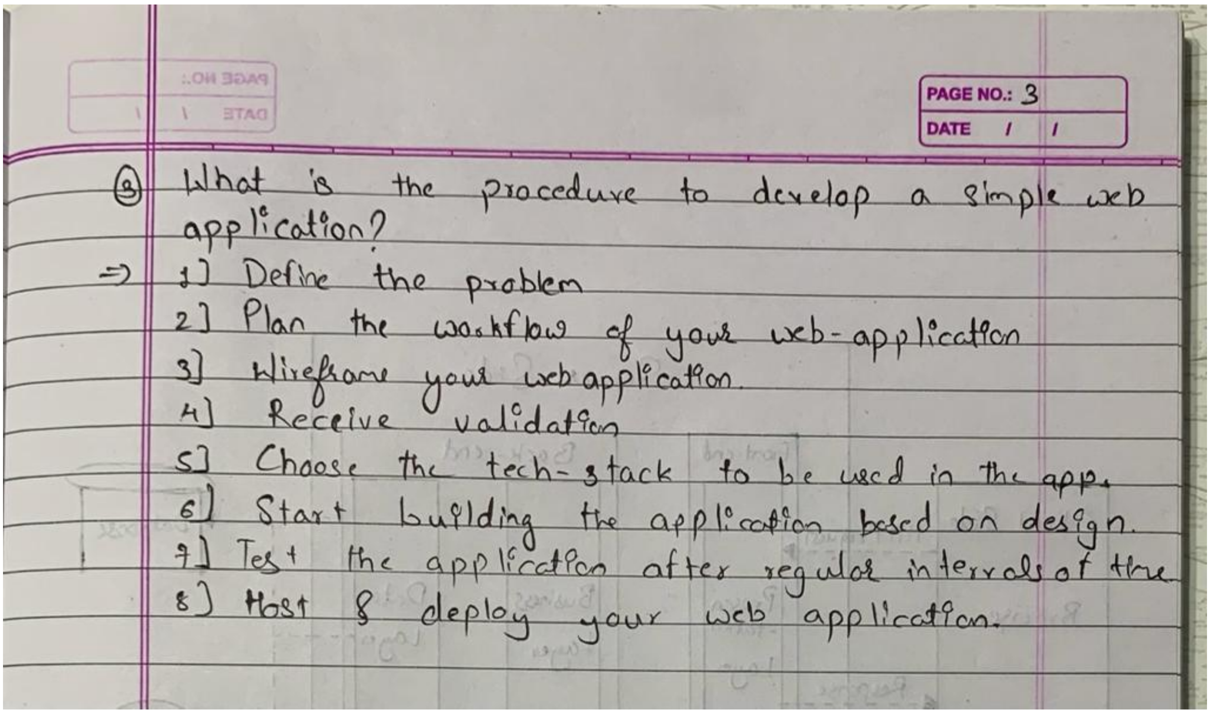
In terms of layers, they can be categorized as:

(i) <u>View Layer</u>: It gives the interface to application. It is the bridge of getting the data in and out of the application.

(ii) <u>Business layer</u>: It is known as Business logic or Application layer. The function is to accept user requests from the browser, process them & determine the router through which the data ~~[illegible]~~ will be accessed.

(iii) <u>Data Access Layer</u>: This layer is built to keep the code you use to pull data from your data store like database.

(iv) <u>Error Handling</u>, <u>Security</u>, <u>Logging</u>: It is the vital part in any application & user experience. It can leave your users feel informed and properly considered.



(3) What is the procedure to develop a simple web application?

=> 1] Define the problem

2] Plan the workflow of your web-application

3] Wireframe your web application.

4] Receive validation

5] Choose the tech-stack to be used in the app.

6] Start building the application based on design.

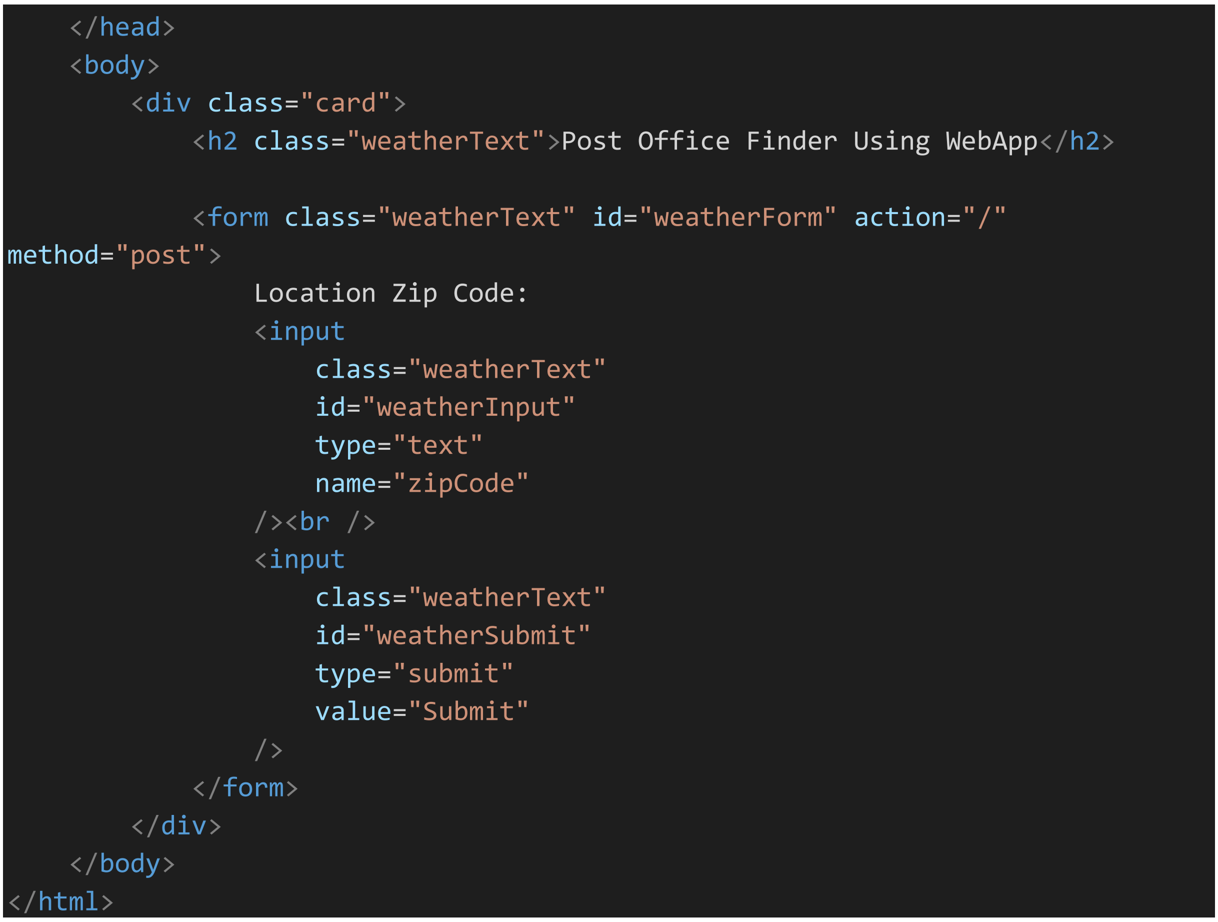
7] Test the application after regular intervals of time.

8] Host & deploy your web application.

OUTPUT :

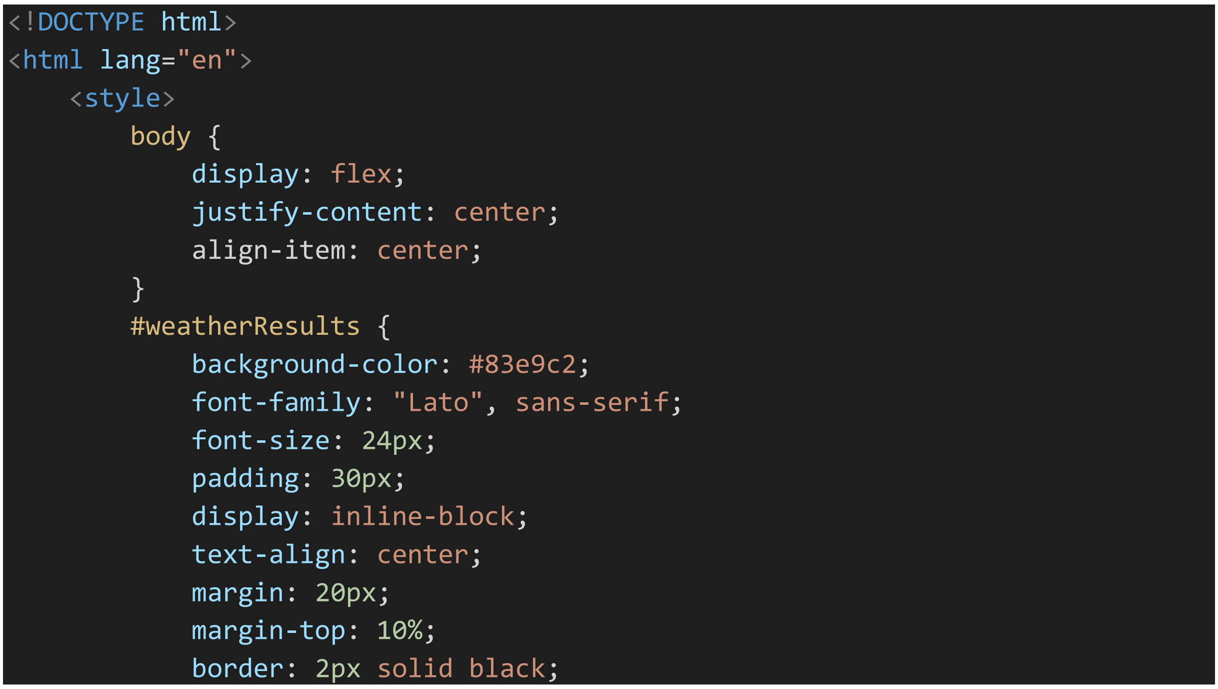
a) Index.html :

```
<html>
    <style>
        .weatherText {
            font-family: "Lato", "sans-serif";
            font-size: 24px;
            text-align: center;
        }

        #weatherForm {
            padding: 20px;
        }

        #weatherSubmit {
            color: white;
            background-color: #083375;
            padding: 5px 20px;
            border-radius: 5px;
            margin-top: 20px;
        }

        #weatherSubmit:hover {
            cursor: pointer;
        }
        body {
            display: flex;
            justify-content: center;
            align-items: center;
        }
        .card {
            border: 2px solid black;
            width: 50%;

            justify-content: center;
            align-items: center;
        }
    </style>
    <head>
        <title class="alignct">Post Office Finder</title>
        <link
            href="https://fonts.googleapis.com/css2?family=Lato:wght@400;700&display=swap"
            rel="stylesheet"
        />
```

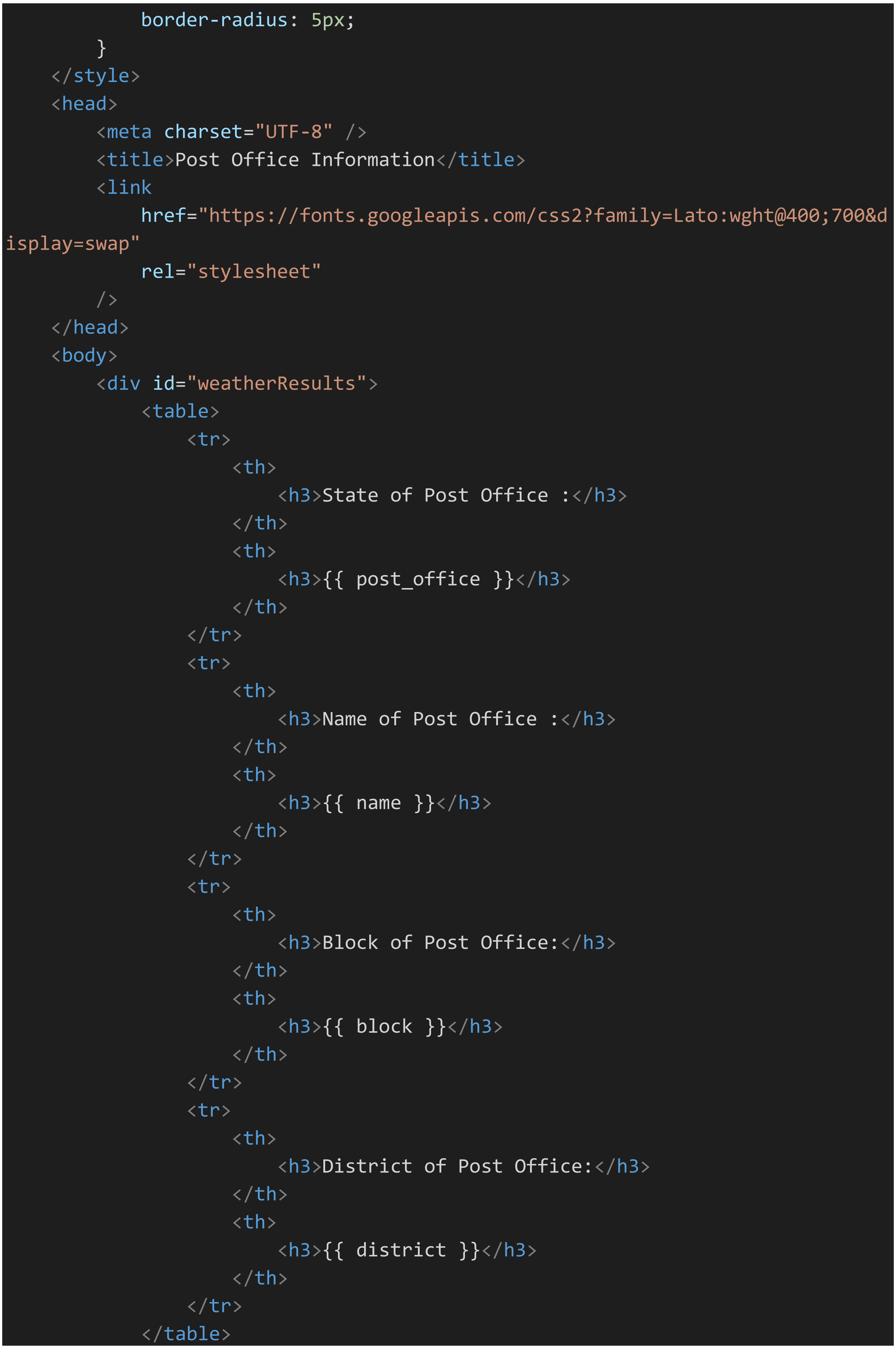


```html
    </head>
    <body>
        <div class="card">
            <h2 class="weatherText">Post Office Finder Using WebApp</h2>

            <form class="weatherText" id="weatherForm" action="/"
method="post">
                Location Zip Code:
                <input
                    class="weatherText"
                    id="weatherInput"
                    type="text"
                    name="zipCode"
                /><br />
                <input
                    class="weatherText"
                    id="weatherSubmit"
                    type="submit"
                    value="Submit"
                />
            </form>
        </div>
    </body>
</html>
```

b) Results.html:

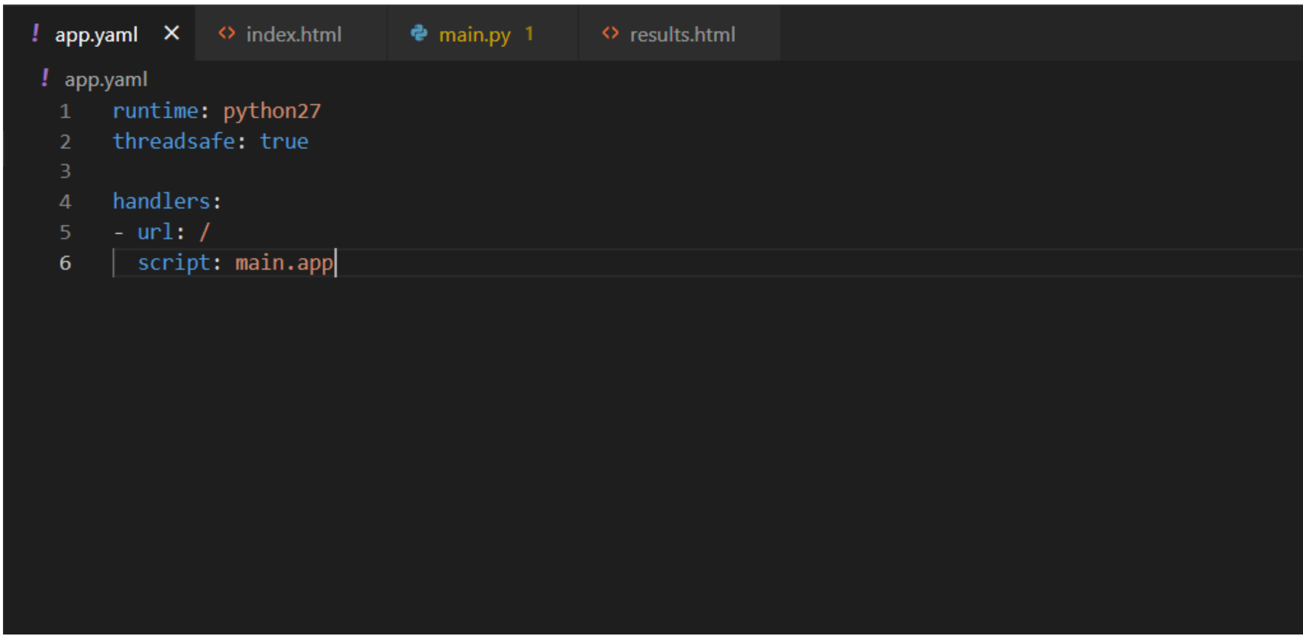


```html
<!DOCTYPE html>
<html lang="en">
    <style>
        body {
            display: flex;
            justify-content: center;
            align-item: center;
        }
        #weatherResults {
            background-color: #83e9c2;
            font-family: "Lato", sans-serif;
            font-size: 24px;
            padding: 30px;
            display: inline-block;
            text-align: center;
            margin: 20px;
            margin-top: 10%;
            border: 2px solid black;
```



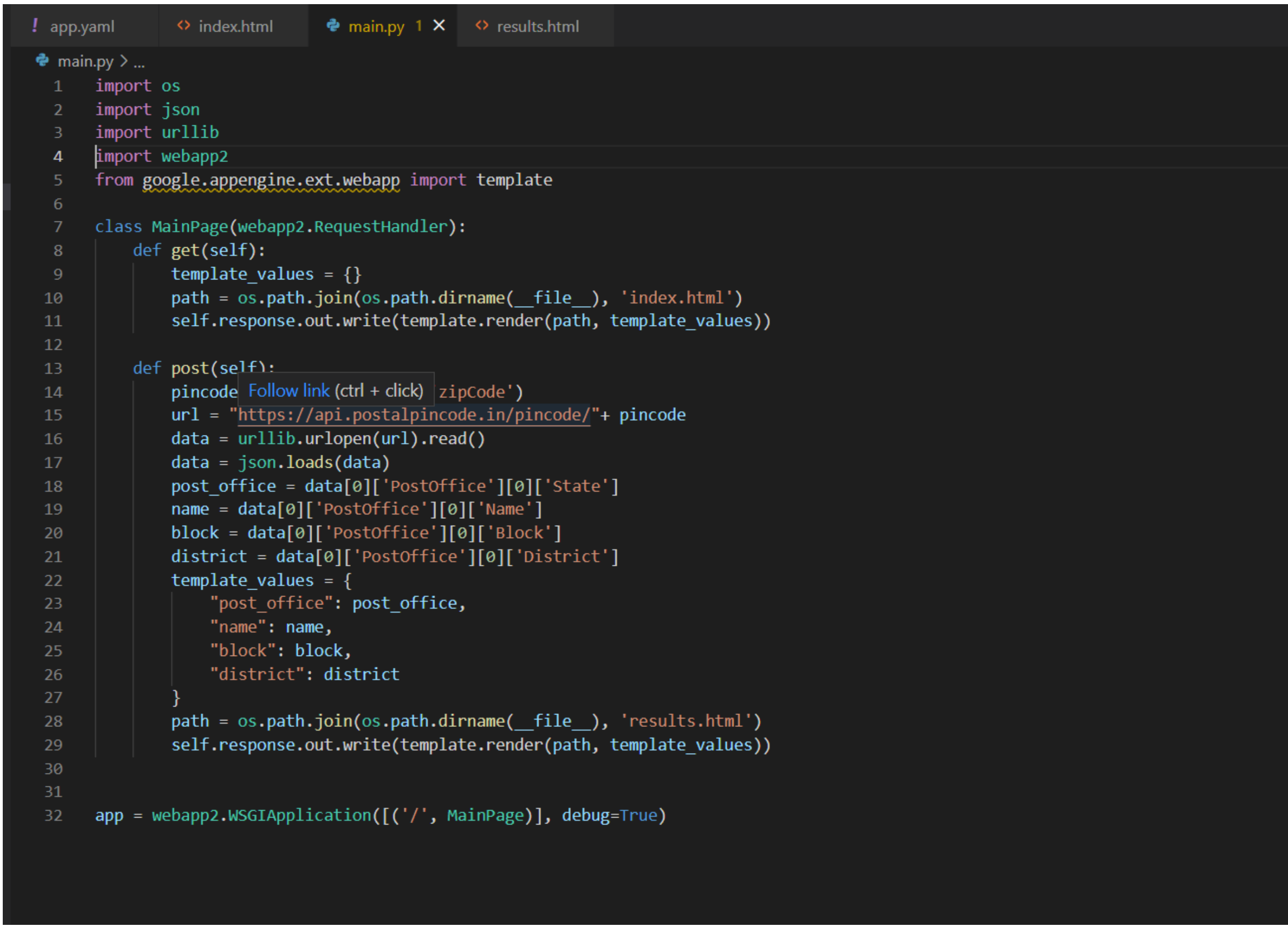
```
            border-radius: 5px;
        }
    </style>
    <head>
        <meta charset="UTF-8" />
        <title>Post Office Information</title>
        <link
            href="https://fonts.googleapis.com/css2?family=Lato:wght@400;700&display=swap"
            rel="stylesheet"
        />
    </head>
    <body>
        <div id="weatherResults">
            <table>
                <tr>
                    <th>
                        <h3>State of Post Office :</h3>
                    </th>
                    <th>
                        <h3>{{ post_office }}</h3>
                    </th>
                </tr>
                <tr>
                    <th>
                        <h3>Name of Post Office :</h3>
                    </th>
                    <th>
                        <h3>{{ name }}</h3>
                    </th>
                </tr>
                <tr>
                    <th>
                        <h3>Block of Post Office:</h3>
                    </th>
                    <th>
                        <h3>{{ block }}</h3>
                    </th>
                </tr>
                <tr>
                    <th>
                        <h3>District of Post Office:</h3>
                    </th>
                    <th>
                        <h3>{{ district }}</h3>
                    </th>
                </tr>
            </table>
        </div>
    </body>
</html>
```

c) app.yaml :

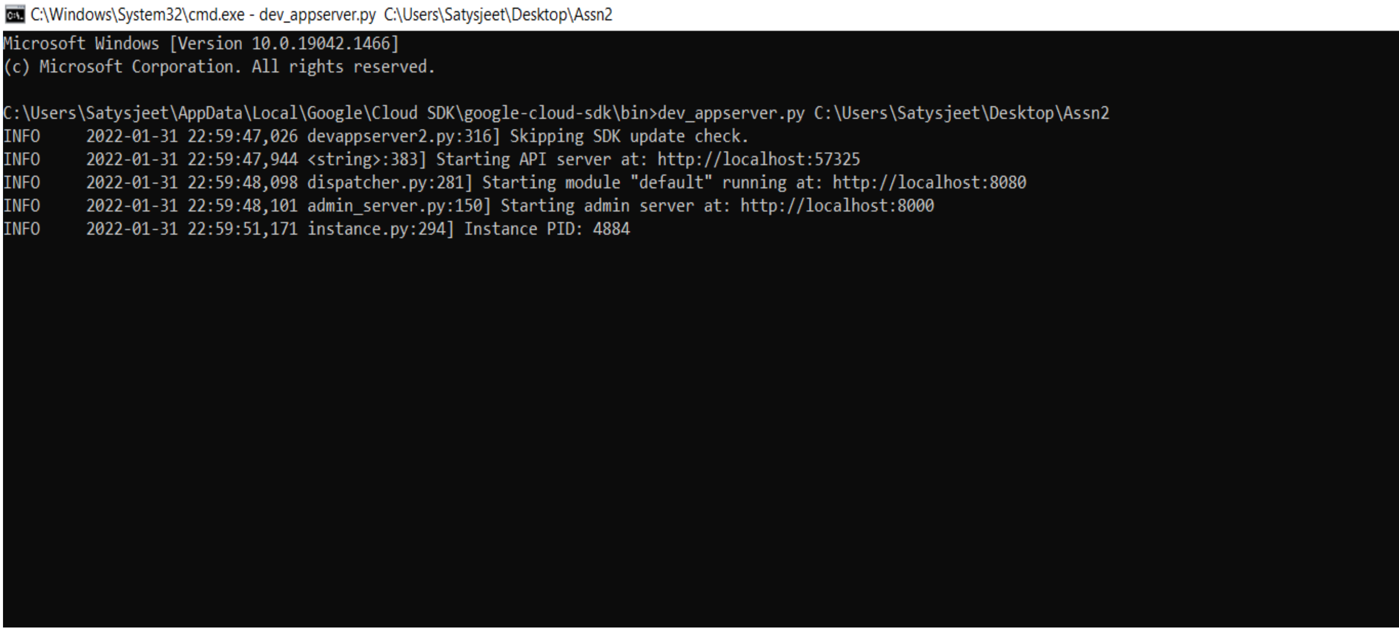


```yaml
runtime: python27
threadsafe: true

handlers:
- url: /
  script: main.app
```

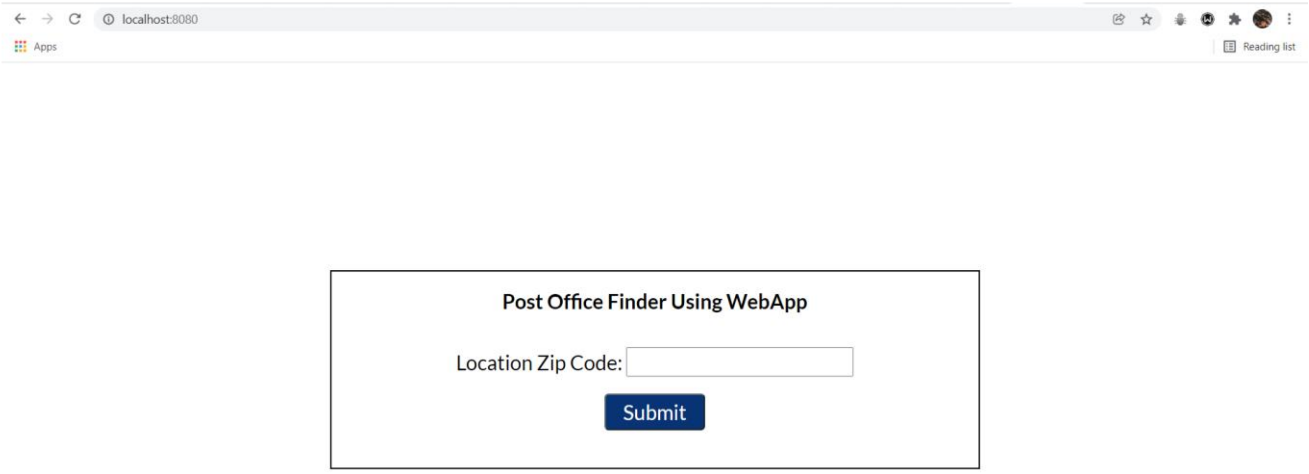
d) main.py :



```
import os
import json
import urllib
import webapp2
from google.appengine.ext.webapp import template

class MainPage(webapp2.RequestHandler):
    def get(self):
        template_values = {}
        path = os.path.join(os.path.dirname(__file__), 'index.html')
        self.response.out.write(template.render(path, template_values))

    def post(self):
        pincode                           zipCode')
        url = "https://api.postalpincode.in/pincode/"+ pincode
        data = urllib.urlopen(url).read()
        data = json.loads(data)
        post_office = data[0]['PostOffice'][0]['State']
        name = data[0]['PostOffice'][0]['Name']
        block = data[0]['PostOffice'][0]['Block']
        district = data[0]['PostOffice'][0]['District']
        template_values = {
            "post_office": post_office,
            "name": name,
            "block": block,
            "district": district
        }
        path = os.path.join(os.path.dirname(__file__), 'results.html')
        self.response.out.write(template.render(path, template_values))


app = webapp2.WSGIApplication([('/', MainPage)], debug=True)
```

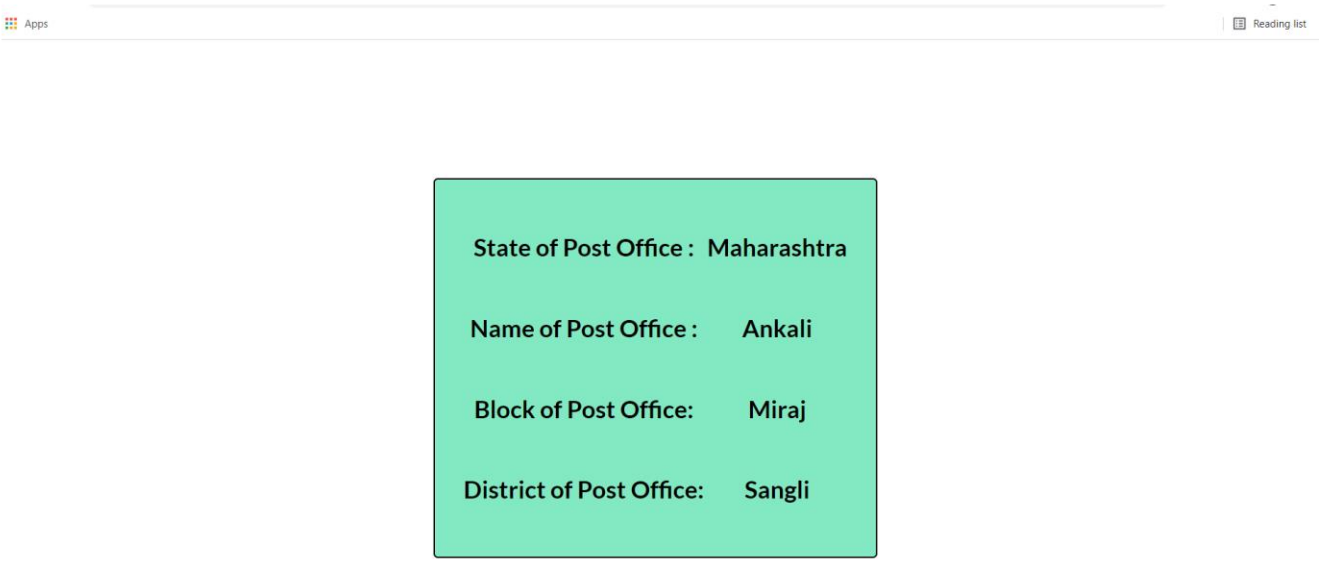

C:\Windows\System32\cmd.exe - dev_appserver.py C:\Users\Satysjeet\Desktop\Assn2

```
Microsoft Windows [Version 10.0.19042.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Satysjeet\AppData\Local\Google\Cloud SDK\google-cloud-sdk\bin>dev_appserver.py C:\Users\Satysjeet\Desktop\Assn2
INFO     2022-01-31 22:59:47,026 devappserver2.py:316] Skipping SDK update check.
INFO     2022-01-31 22:59:47,944 <string>:383] Starting API server at: http://localhost:57325
INFO     2022-01-31 22:59:48,098 dispatcher.py:281] Starting module "default" running at: http://localhost:8080
INFO     2022-01-31 22:59:48,101 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO     2022-01-31 22:59:51,171 instance.py:294] Instance PID: 4884
```



Post Office Finder Using WebApp

Location Zip Code:



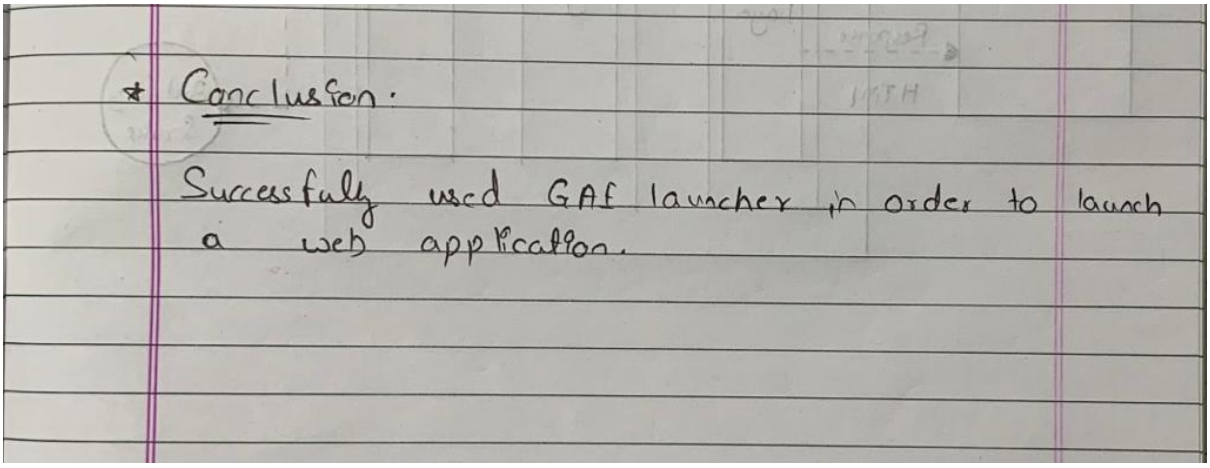
State of Post Office : Maharashtra

Name of Post Office : Ankali

Block of Post Office: Miraj

District of Post Office: Sangli

Conclusion :



★ Conclusion:

Successfuly used GAE launcher in order to launch a web application.