

Assignment 2

NAME: KRISHNA GADAM

ROLL NO: 33124

BATCH: L9

CONTACT NO: 9834303132

EMAIL: krishnagadam3501@gmail.com



Assignment - 2

Roll No.: 33124

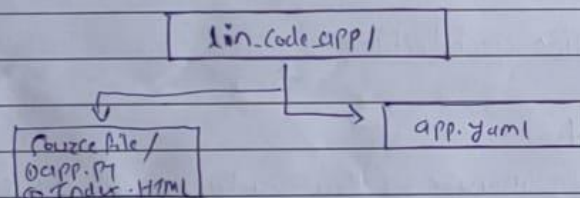
Aim:- Use GAE launcher to launch the web applications.

Theory:-

Q.1) Mention the structure of GAE Applications with their file structure and description?

⇒ Directory structure

- Each version of your App Engine Service is defined in an app.yaml file.
- For simple apps, the minimum requirement for development is to define the app.yaml file.
- The app.yaml acts as a deployment descriptor and defines the scaling type and runtime, handler and other resource settings for a specific version of service.
- If several versions of a service are to be deployed multiple yaml files can be created in the same directory to represent the configuration for each of the versions.



Directory Diagram

- My lincode app consists of a single lincode app ^{service} and hence only one app.yaml is used as a deployment descriptor. In case it runs multiple services other files can be included too.

Q2) What are the main components of web application?

⇒

View layer :- It gives an interface to the application. It is a bridge for getting the data in and out of the application. It does not contain business logic.

Business layer :- It accepts users requests from the browser, process them and determine the routes through which the data will be accessed.

Data Access layer :- It keeps the code that is used to pull data from data store like database, flat files or web services separate from business logic and presentation code thus avoiding rewriting.

Error handling, security, logging :-
When a web application is built end goal, building, and testing are only focused on for situations when things go right.

Website application architecture :-

Serverless Architecture :-

Don't need to manage infrastructure. All the resource management is done by third party service provider like AWS.

Single page Applications :-

It interfaces with the user by dynamically rewriting the current page rather than loading entire new pages from a server.

Microservices :-

- Responsible for executing a single functionality
- Allows developers working with it to not only enhance productivity but also speed up the entire deployment process.

Q3) 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 & receive validation.

④ Choose the tech stack to be used in your app & start building your application based on design.

⑤ Test application after regular interval of time.

⑥ Host & deploy your web application.

Conclusion :-

In this assignment, I learnt about launching web applications using GAE launcher.

Code:

```
index.html - Assignment 2 - Visual Studio Code

main.py
1 import os
2 import json
3 import urllib
4 import webapp2
5 from google.appengine.ext.webapp import template
6
7 class MainPage(webapp2.RequestHandler):
8     def get(self):
9         template_values = {}
10        path = os.path.join(os.path.dirname(__file__), 'index.html')
11        self.response.out.write(template.render(path, template_values))
12
13    def post(self):
14        pincode = self.request.get('zipCode')
15        url = "https://api.postalpincode.in/pincode/" + pincode
16        data = urllib.urlopen(url).read()
17        data = json.loads(data)
18        post_office = data[0]['PostOffice'][0]['State']
19        name = data[0]['PostOffice'][0]['Name']
20        block = data[0]['PostOffice'][0]['Block']
21        district = data[0]['PostOffice'][0]['District']
22        branchType = data[0]['PostOffice'][0]['BranchType']
23        division = data[0]['PostOffice'][0]['Division']
24        country = data[0]['PostOffice'][0]['Country']
25        template_values = {
26            "post_office": post_office,
27            "name": name,
28            "block": block,
29            "district": district,
30            "division": division,
31            "country": country,
32            "branchType": branchType
33        }
34
35    def render(self, template_values):
36        self.response.out.write(template.render('index.html', template_values))
37
38    def __init__(self, *args, **kwargs):
39        super(MainPage, self).__init__(*args, **kwargs)
40
41    def __call__(self, *args, **kwargs):
42        self.get(*args, **kwargs)
43
44    def __repr__(self):
45        return 'MainPage'
46
47    def __str__(self):
48        return 'MainPage'
49
50    def __del__(self):
51        pass
52
53    def __enter__(self):
54        pass
55
56    def __exit__(self, *args, **kwargs):
57        pass
58
59    def __getitem__(self, key):
60        return getattr(self, key)
61
62    def __setitem__(self, key, value):
63        setattr(self, key, value)
64
65    def __contains__(self, key):
66        return key in self.__dict__
67
68    def __len__(self):
69        return len(self.__dict__)
70
71    def __iter__(self):
72        return iter(self.__dict__)
73
74    def __reversed__(self):
75        return reversed(self.__dict__)
76
77    def __hash__(self):
78        return hash(self.__dict__)
79
80    def __eq__(self, other):
81        return self.__dict__ == other.__dict__
82
83    def __neq__(self, other):
84        return not self.__eq__(other)
85
86    def __lt__(self, other):
87        return self.__dict__ < other.__dict__
88
89    def __le__(self, other):
90        return self.__dict__ <= other.__dict__
91
92    def __gt__(self, other):
93        return self.__dict__ > other.__dict__
94
95    def __ge__(self, other):
96        return self.__dict__ >= other.__dict__
97
98    def __add__(self, other):
99        return self.__dict__ + other.__dict__
100
101    def __sub__(self, other):
102        return self.__dict__ - other.__dict__
103
104    def __mul__(self, other):
105        return self.__dict__ * other.__dict__
106
107    def __div__(self, other):
108        return self.__dict__ / other.__dict__
109
110    def __mod__(self, other):
111        return self.__dict__ % other.__dict__
112
113    def __pow__(self, other):
114        return self.__dict__ ** other.__dict__
115
116    def __radd__(self, other):
117        return other.__dict__ + self.__dict__
118
119    def __rsub__(self, other):
120        return other.__dict__ - self.__dict__
121
122    def __rmul__(self, other):
123        return other.__dict__ * self.__dict__
124
125    def __rdiv__(self, other):
126        return other.__dict__ / self.__dict__
127
128    def __rmod__(self, other):
129        return other.__dict__ % self.__dict__
130
131    def __rpow__(self, other):
132        return other.__dict__ ** self.__dict__
133
134    def __iadd__(self, other):
135        self.__dict__ += other.__dict__
136
137    def __isub__(self, other):
138        self.__dict__ -= other.__dict__
139
140    def __imul__(self, other):
141        self.__dict__ *= other.__dict__
142
143    def __idiv__(self, other):
144        self.__dict__ /= other.__dict__
145
146    def __imod__(self, other):
147        self.__dict__ %= other.__dict__
148
149    def __ipow__(self, other):
150        self.__dict__ **= other.__dict__
151
152    def __iand__(self, other):
153        self.__dict__ &= other.__dict__
154
155    def __ior__(self, other):
156        self.__dict__ |= other.__dict__
157
158    def __ixor__(self, other):
159        self.__dict__ ^= other.__dict__
160
151
```

```
Google Cloud SDK Shell - py google-cloud-sdk\bin\dev_appserver.py "C:\Users\KRISHNA\Desktop\CCL\Assignment 2"
Welcome to the Google Cloud SDK! Run "gcloud -h" to get the list of available commands.
---
C:\Program Files (x86)\Google\Cloud SDK>py google-cloud-sdk\bin\dev_appserver.py "C:\Users\KRISHNA\Desktop\CCL\Assignment 2"
INFO 2022-02-01 16:25:51,019 devappserver2.py:316] Skipping SDK update check.
INFO 2022-02-01 16:25:53,683 <string>:383] Starting API server at: http://localhost:57265
INFO 2022-02-01 16:25:53,701 dispatcher.py:281] Starting module "default" running at: http://localhost:8080
INFO 2022-02-01 16:25:53,714 admin_server.py:150] Starting admin server at: http://localhost:8000
INFO 2022-02-01 16:25:57,305 instance.py:294] Instance PID: 16840
```

Output:

