Backend API Using Python Flask

Documentation

- [Github url click here] - Public code repository url
- API url click here - This is a sample api url for london city
- Open weather app - This is the free public api we get weather data
- Road goat api for images - This is the free public api for images
- The api is exposed as a get method using Python Flask Restful
- This one calls two apis that is open weather map and api road goat
- weather information is returned by open weather map and photo urls by road goat
- The api takes weather and images entites and aggregtes and return that as a json response

API GET endpoint /cityName/{name of city}

example /cityName/london

Aggregates the response from two public apis and return the json response which contains the weather data and the images for a city

Handling Bad input

The code checks for the backend apis to see if any one returns a status code other than 200 and sends a response immediately as below

```
{"response": "either api is down or bad input"}
```

Sample success response

The below response is for the city london

```
{
   "weather":[
      {
         "id":804,
         "main":"Clouds",
         "description":"overcast clouds",
         "icon":"04d"
      }
   ],
   "image":{
      "full":"https://cdn.roadgoat.com/uploads/photo/image/431/travel-guide-of-london-united-kingdom-original.jpg",
      "large":"https://cdn.roadgoat.com/uploads/photo/image/431/large_travel-guide-of-london-united-kingdom-original.jpg",
      "medium":"https://cdn.roadgoat.com/uploads/photo/image/431/medium_travel-guide-of-london-united-kingdom-original.jpg",
      "thumb":"https://cdn.roadgoat.com/uploads/photo/image/431/thumb_travel-guide-of-london-united-kingdom-original.jpg",
      "avatar":"https://cdn.roadgoat.com/uploads/photo/image/431/avatar_travel-guide-of-london-united-kingdom-original.jpg"
```

```
    }
}
```

Google Cloud

The code is deployed to app engine standard in google cloud platform using gcloud app deploy command.

Improvements

- The error handling needs to be improved as currently the images are taken from the response without checking if the images are returned null. Some cities the images can be null which the code is not checking.
- Using Flask Response we can set the status code to the correct status code during error. As of now it returns 200 for error
- The images in the response can be downloaded and saved to a directory as part of enhancement
- The code can be deployed as a cloud run container which saves cost as it charges only for the data used unlike app engine instance