2022

Scenic Spot

CAB432 Assignment 1

Ning Wang

N11148870

9/21/2022

Contents

Introduction 2

Mashup Purpose & description 2

Services used 2

Twitter Standard Search API (v.1.1) 2

API 2 2

API n 2

Persistence Service 3

Mashup Use Cases and Services 3

Restaurant Search in a Foreign City 3

US 2 4

US n 3

Technical breakdown 4

Architecture and Data Flow 5

Deployment and the Use of Docker 5

Test plan 5

Difficulties / Exclusions / unresolved & persistent errors 6

Extensions (Optional) 7

User guide 8

Analysis 8

Question 1: Vendor Lock-in 8

Question 2: Container Deployment 9

References 10

Appendices 10

This is a template for your assignment report. It is not compulsory to use it, but it will save a lot of effort if you do. Some of these sections may not be very long, but you should make sure that you cover the key sections describing the application functionality and implementation.

*Throughout this document, you should assume that black text in italics is there as guidance and you should read it, follow the instructions and then **delete it** when you have entered your own text. Some examples are not italicized, but should obviously be replaced by your own material. It doesn't look good when we mark it if the guide text is still there.*

There is a mix here between the sections of an ordinary software development report – descriptions of the use cases, technologies used, architecture and testing – and a section in which we ask you to analyse your application based on some prompts that we give you. These two components attract the same number of possible marks.

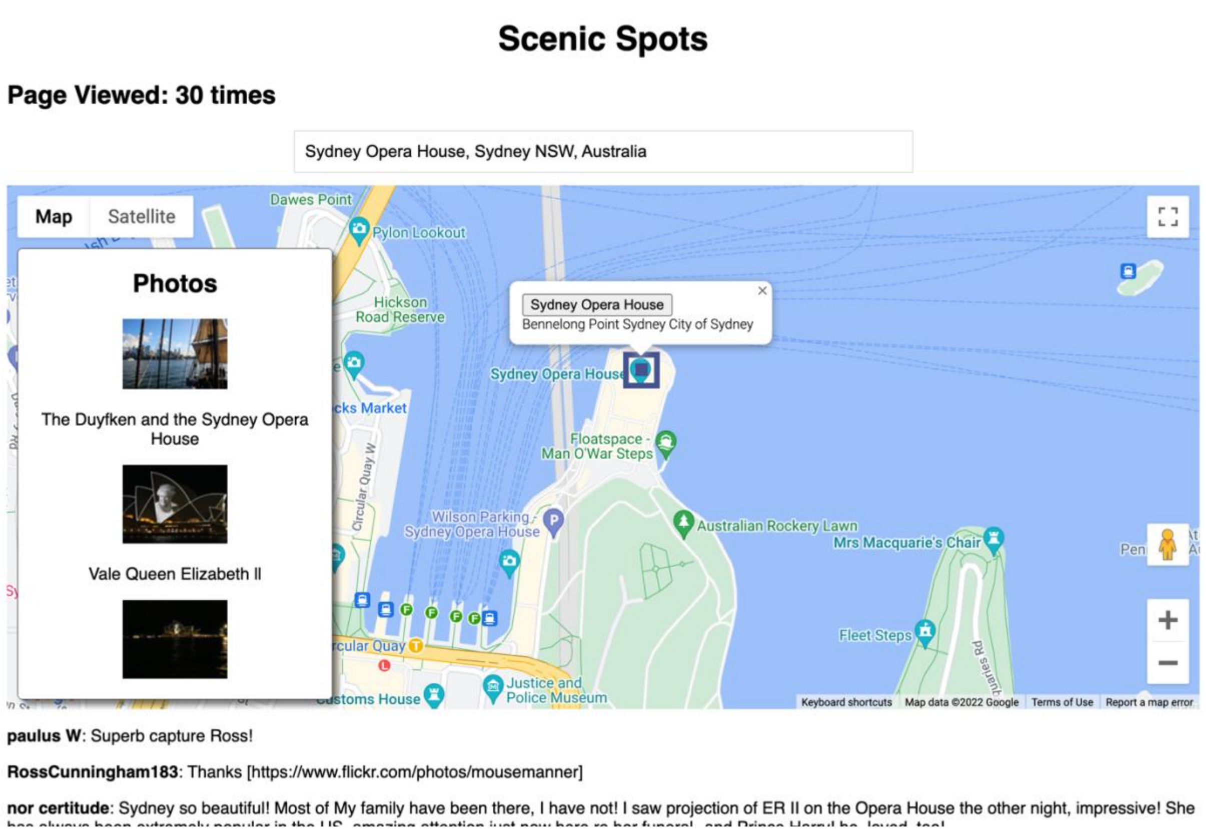
This report should be around 10-15 pages including screenshots, but this is a guide only – we will not be enforcing a page limit or marking you down for submitting something with 16 pages instead. But be sensible, we really don't want something that is 25 pages or more.

[Our thanks to the students who allowed us to use images from earlier reports as examples here and in the template for Assignment 2.]

Introduction

Mashup Purpose & description

This application aims to provide a platform that allows visitors to search locations and find out photos that have been taken at the location. For potential travelers, this website works as a preview of their target destinations to help their decision making. By simply entering the location, a bunch of images related to that location will be displayed. Moreover, the users can also read the comments for a particular photo by clicking on it.



Scenic Spots

Page Viewed: 30 times

Sydney Opera House, Sydney NSW, Australia

Photos

The Duyfken and the Sydney Opera House

Vale Queen Elizabeth II

paulus W: Superb capture Ross!

RossCunningham183: Thanks [https://www.flickr.com/photos/mousemanner]

nor certitude: Sydney so beautiful! Most of My family have been there, I have not! I saw projection of ER II on the Opera House the other night, impressive! She has always been extremely popular in the US, amazing attention just now here re her funeral, and Prince Harry! he loved, too!

Services used

1. Google Map api

This api works as the core function of the application. It uses a callback function to render the map, marker for destination. With pre-developed functions, this api allows to create features by initializing new google elements.

Endpoint:

```
https://maps.googleapis.com/maps/api/js?key=AIzaSyCn4HhwD9z99kXEfPv6nHYTYHqsEkxMk38&libraries=places&callback=initMap&v=weekly
```

Docs:

https://developers.google.com/maps/documentation/javascript/adding-a-google-map?hl=en_GB

2. Flickr api

First flickr api is used to extract photo data by a searching method. When this api gets called, it takes the value searched as tags then fetch associated data from flickr database.

The Second api is used to fetch comments for a particular photo. With a photo id extracted from the first api, the second api is able to acquire comments for that specific photo.

Endpoint 1:
https://api.flickr.com/services/rest/?&method=flickr.photos.search&api_key=XXX&tags=XXX

Endpoint 2:

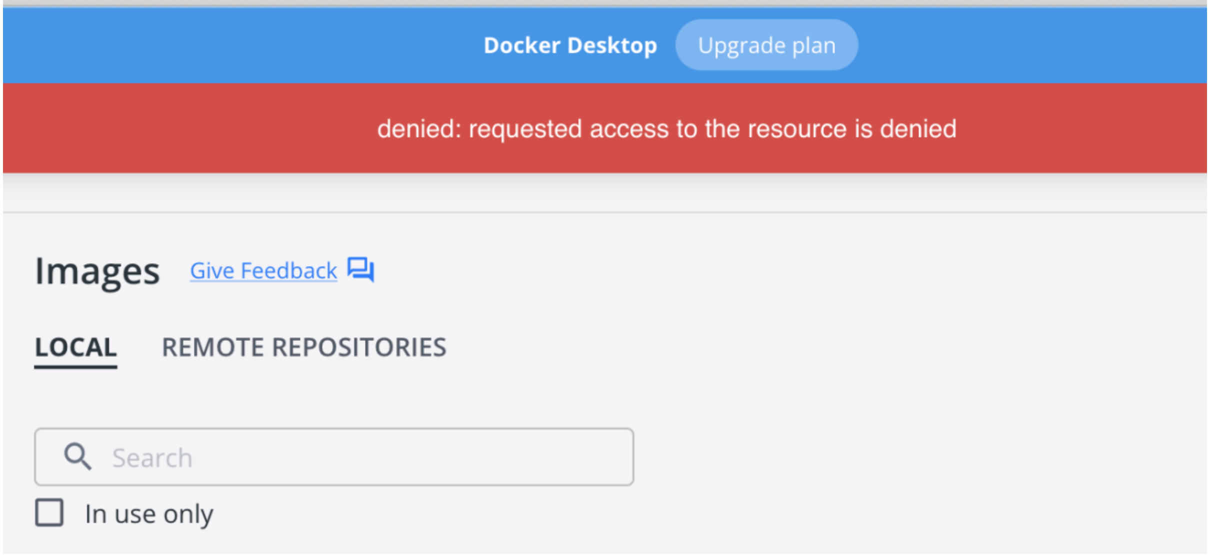
Docs:

https://www.flickr.com/services/api/flickr.photos.search.html

https://www.flickr.com/services/api/flickr.photos.comments.getList.html

Persistence Service

1. For page view counter
   Dynamodb
2. Docker deployment
   I wasn't able to properly set up aws ENV and failed to push image to docker hub



denied: requested access to the resource is denied

Mashup Use Cases and Services

Photo Search for a Location

As an	International tourist

I want	To find photos shot for the location that I want to find
So that	I know what the place looks like

Entering an address in the input box, then hit enter or click on the auto filled address. The map will mark for the location. A small window will popup stating the name of the address. Click on the address name, relevant photos with titles will be displayed on the left white panel. Scroll to view more photos.

Check comments for a photo

As a	Potential traveler
I want	To find photos of a location I want to travel and see other's perspective of the location
So that	I can decide if I want to travel there

Following the same procedure, the user can then click on the photo and comments for that photo will be displayed underneath the map.

Technical breakdown

Frontend

The google map api uses a callback function to initialize a map. Inside the function, instances such as map, marker, infowindow, autocomplete will be constructed using google in-built functions. The autocomplete instance takes the input of users, then binds to the map variable. Then uses "addListener" function to execute the action. Upon the location change, zoom the view port and place a marker for the specified location. Finally open an infowindow that contains the map and marker variables with a content of a html string. In this html string, include an onclick function which will then call the flickr search api to fetch photo data. The onclick function fetch a backend url and convert the response into json format then parse the json structured elements into usable data to construct a html string. Get comment function is implemented in the same method by fetching customized backend url.

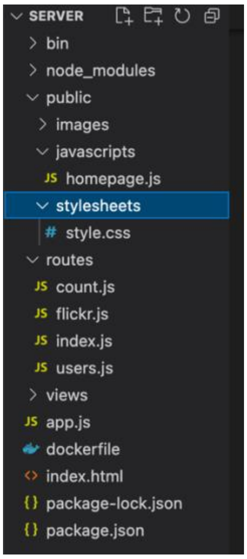
Backend

The backend for this application mainly sends https request to the flickr database. Firstly, it sets up a flickr request option which includes all parameters of the endpoint. When the backend router is triggered, it sends a https request with the options to flickr database to get a string of buffers, then parse the buffer into readable json format. Upon receiving the calls from the frontend, the backend will write the data and store data in a customized url.

Page view counter

Firstly, the backend connects with the dynamodb in a router named "/count". When this router is called, it will query the backend for an attribute value named "count" and assign the value to an integer, it will then update the count in the database with the integer + 1. So whenever, this query gets called, the count value is fetched, then update it to previous value+1. In the frontend, whenever the page is loaded, it will automatically call a fetch(/count) function to get the count value, then display it on the website.

Architecture and Data Flow

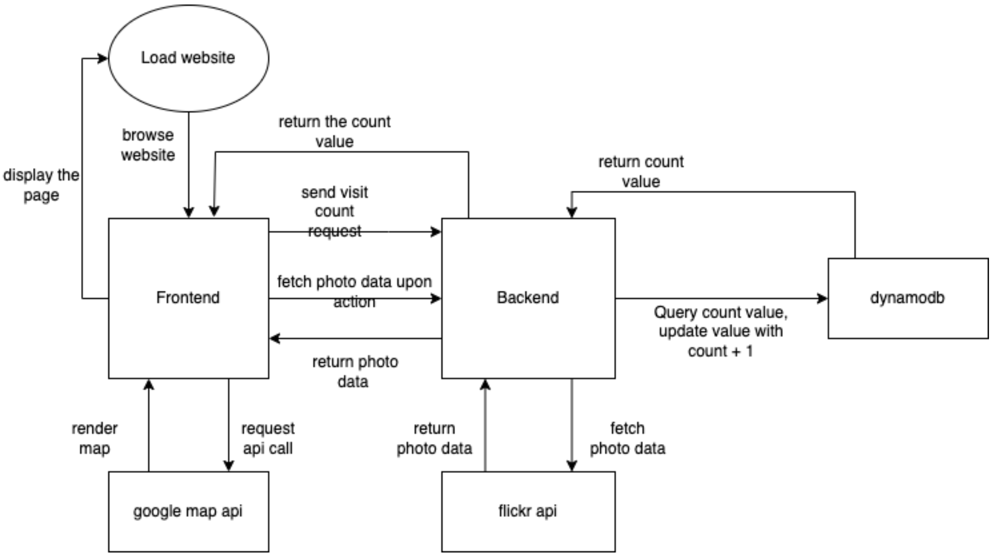


```
SERVER
  > bin
  > node_modules
  v public
    > images
    v javascripts
      homepage.js
    v stylesheets
      style.css
  v routes
    count.js
    flickr.js
    index.js
    users.js
  > views
  app.js
  dockerfile
  index.html
  package-lock.json
  package.json
```

Application Architecture

For this application, I have created an express generator as the backend, packed a static html page in the same folder. The frontend is mostly implemented based on a public javascript named "homepage.js", and fetching backend urls are called inside this js file. The backend code is mainly distributed in 3 js files named "count.js", "flickr.js" and "index.js" where index.js reads the static html file and the other two routers run as customized url for frontend fetches.

Data flow



Deployment and the Use of Docker

Test plan

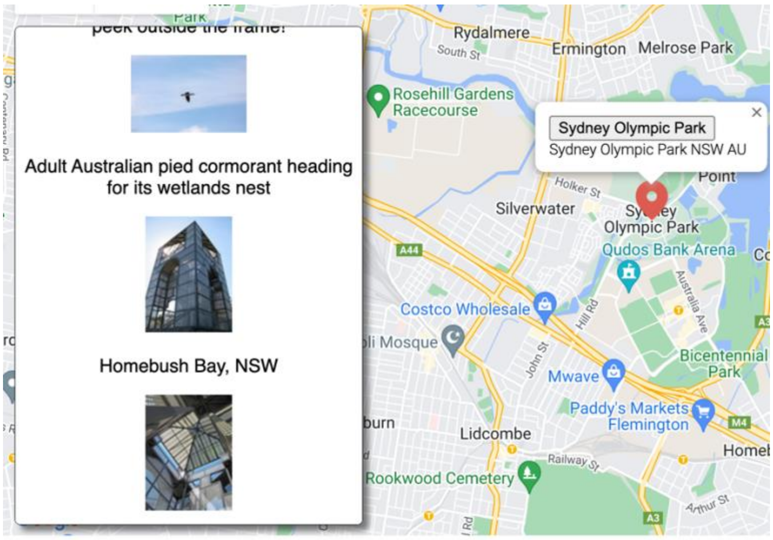
Tasks	*Expected Outcome*	*Results*	*Screenshot/s (Appendix)*
Zoom in/out map, drag map using cursor	*Map zoomed in/out, draggable*	*Pass*	
Auto fill the remaining of location	*A drop-down list of potential address*	*Pass*	01

Click on the location, hit enter on location complete	*Location is searched and View port zoom to the searched location, pin and info of the location displayed*	*Pass*	02
Click on the button of location info window	*Images displayed at the left panel*	*Pass*	03
Limited number of photos	*Result count limited as per selection*	*Pass*	
Click on the Photo	*Comments will be displayed beneath the map*	*Pass*	04
No Comments for image	*"Image has no comments" displayed beneath the map*	*Pass*	05
Handle Flickr response error	*Alert popup, app continues*	*Pass*	06
View count	*View count increases when reload the page*	*Pass*	*07*

Difficulties / Exclusions / unresolved & persistent errors /

First challenge during the development is that I cannot move all frontend javascripts to backend. Because of most of the source code of google map api was from documentation, I was not able to modify it and implement the functions in the backend server app.

Second challenge is api issues when I tried to use flickr.geo.getLocationForImage. To get more specific images for a scenic spot, it is best to search with geographical data (latitude & longitude) instead of searching names of the location. Also, the theme of the photo is not limited to only locational as the results will display any photos that are related to what is searched.



Adult Australian pied cormorant heading for its wetlands nest

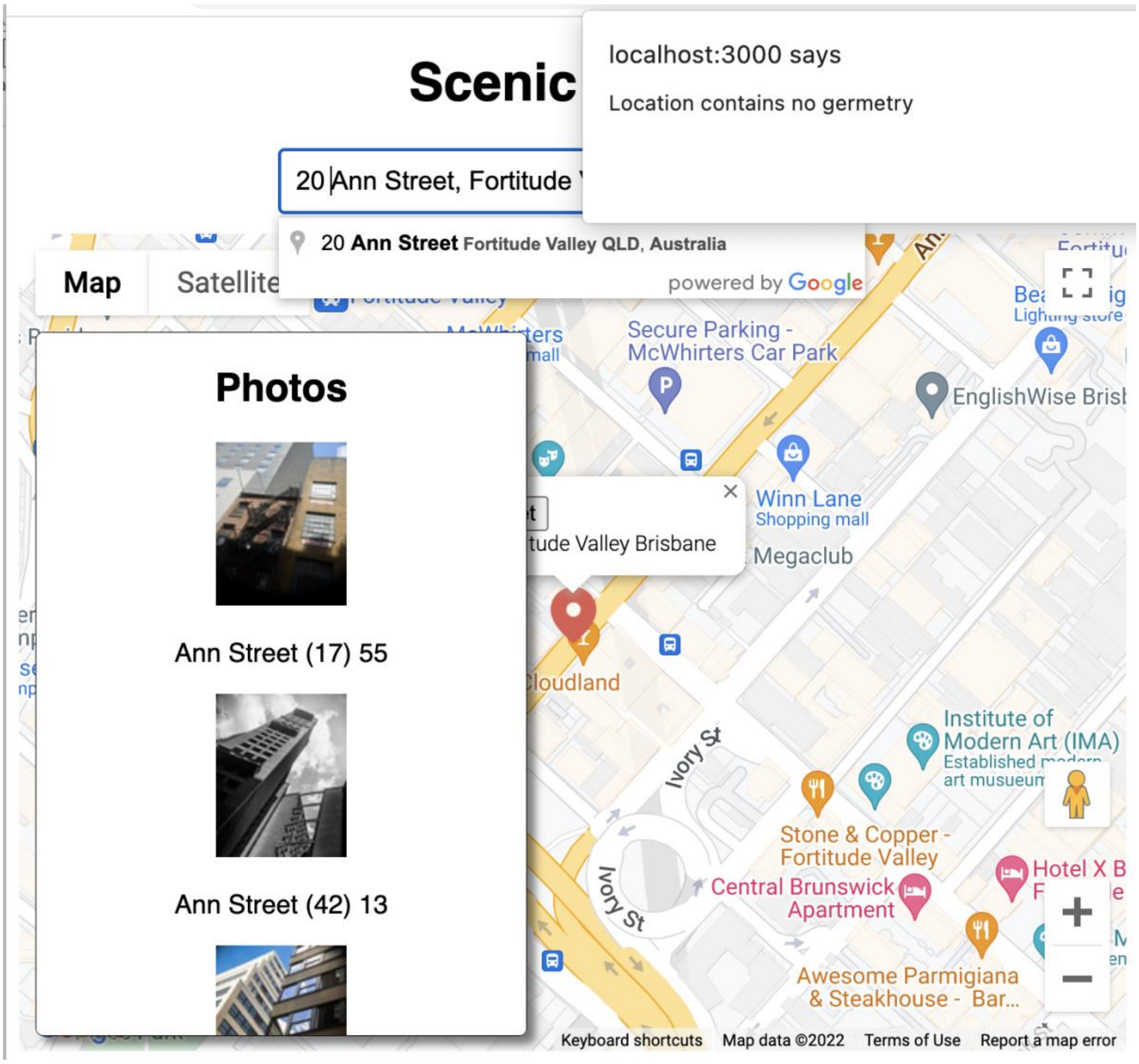
Homebush Bay, NSW

Sydney Olympic Park

Sydney Olympic Park NSW AU

A solution to this is that the endpoint of the api should also include the content of the photo, for instance "&content=location" or "&content=portrait", etc.

Another issue is that the results is not interactive. When searching a new location while the location has no data, the photos from previous search will still be displayed in the panel. This is due to the render of photos only responsive to “button onclick” function.



localhost:3000 says

Location contains no germetry

Scenic

20 Ann Street, Fortitude

20 **Ann Street** Fortitude Valley QLD, Australia

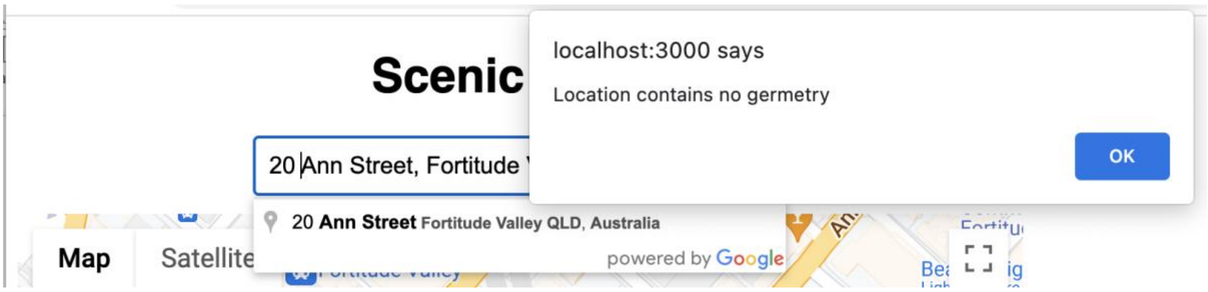
Photos

Ann Street (17) 55

Ann Street (42) 13

powered by Google
Map
Satellite
Secure Parking - McWhirters Car Park
EnglishWise Brisb
Winn Lane Shopping mall
Megaclub
tude Valley Brisbane
Cloudland
Ivory St
Institute of Modern Art (IMA)
Established modern art museum
Stone & Copper - Fortitude Valley
Central Brunswick Apartment
Hotel X B
Awesome Parmigiana & Steakhouse - Bar...
Keyboard shortcuts
Map data ©2022
Terms of Use
Report a map error

Overall, there is no obvious bugs, but will throw an error when trying to search street names instead of City (i.e., search photos “Ann Street Brisbane city” will result an alert since no photos can be found).



localhost:3000 says

Location contains no germetry

Scenic

20 Ann Street, Fortitude

20 **Ann Street** Fortitude Valley QLD, Australia

Extensions (Optional)

As stated in the *Difficulty* section, the images are not found by the geographical location but the name of the place, which leads to getting unrelated or inaccurate images. Due to time consumptions, I was not able to fully understand all Flickr apis and authorities. For future improvements, the first thing to look at is the choice of apis as this occurs to be the most challenging difficulty during the development. Also, using React framework will be a better choice than having a static html page saved in a backend application folder.

User guide

Step 1: Browse the webpage

Step 2: Enter address in the input bar

Step 3: Either click on the autocomplete address or fill in the full address then hit enter

Step 4: click on the button of the infowindow on the map above the location marker

Step 5: scroll to view more photos on the left panel

Step 6: click on the image to view comments beneath the map

Analysis

In this section we ask you look at your application and to analyse it in response to a couple of prompts we supply below. The marking is based on the quality of the analysis and not on the length of your response. There are two questions. In each case, there is an overall question, and then a series of bullet points that help you respond. A good answer can be no more than a couple of sentences in response to each of them. Say more if you have more to say, but don't waffle. Say it quickly and get on with the next one. This exercise is comparatively straightforward but there is a corresponding task in Assignment 2 which will be far more difficult.

Question 1: Vendor Lock-in

Looking at your mashup as it stands, how dependent are you on the people who provide the services that you use? In a commercial context, the APIs, the data and the cloud services all matter to you. How hard would it be to change?

Answer:

For this mashup assignment, I have selected 2 main api providers, *google map api & flickr api*. Google map api was mainly used to display the map and allow users to search locations, whereas flickr apis were used to fetch photo and comment data. The application is heavily depended on both apis since the website does not have any content itself, rather it works as a data demonstration. The core search function of this application is based on google built in services, which require payments to use some of the services. Although flickr apis were free to use, I did not get familiar with every single api endpoint and were not able to implement functions that I intended to use. It is relatively easy to deal with flickr api as it only needs more studies of the api structure and documentation, whereas finding an alternative solution for map api is not realizable at this stage.

In your response, you should consider the following prompts:

- *How hard would it be to replace the APIs that you are working with? Could you easily replace them if they were shut down suddenly or their terms of service changed? Consider each of them in turn and explore the domain of the API and tell us about obvious competitors or their absence.*

(4 marks)

Answer:

The change of service can be easily fixed by changing endpoints of the apis and re-structured the data to adapt the changed json format. If both api providers are shut down, the flickr api can be easily replaced by different providers since this website only fetches photo data from flickr endpoints in a json format, whereas the search location function needs to be completely re-written.

- *How hard would it be to change the persistence service you have used to one supplied by another vendor? Identify equivalent alternatives and discuss briefly how that might affect your approach.*

(2 marks)

Answer:

For a simple data attribute like page counter, it can be easily replaced by another type of database since the size of it is quite small. But for a large project at scale, nosql outperforms other sql data queries in terms of flexibility and runtime efficiency.

Question 2: Container Deployment

The mashup you have created as part of this assignment is a very limited application by commercial standards. Without getting too carried away, I want you now to think of a more substantial application which has similar characteristics – drawing mainly on external services, perhaps extending to include some user accounts and security, adding in some additional persistence services like those discussed in the weeks leading up to the assignment submission.

Working with this more substantial application, what are the advantages and disadvantages of the container-based deployment? This new context will in practice involve scaling and load balancing. You don't yet have direct experience of these services, so I want you to focus on the deployment of the application through software containers. You should refer to earlier notes about the trade-offs between containers and 'full' VMs and consider which might apply here, and which might not.

In your response, you should consider the following prompts:

- *Assuming that we have access to a service to manage container deployment and communication, are there any disadvantages to a container-based deployment for this application? Would we consider deployment of the application directly to a virtual machine i.e. one instance of the application for each instance of the EC2 or Azure Linux VM?*

(2 marks)

Answer:

Containers virtualize the host OS and run packaged application on top of it. Beside many advantages of docker deployment such as scalability and portability, one of the disadvantages is the lacking security measures as containers provides lightweight isolation from the host OS and containers within the same system. The security boundary is weaker compared to VMs. So if considering the deployment at a security level, it's better to deploy to VMs as they provides more isolation between neighbouring systems.

- *Drawing upon the discussions of cloud architectures in the early lectures and the material on persistence from week 5 onwards, what persistence options would you consider if the application were to be deployed at scale using a collection of software containers? You may consider more than one level.*

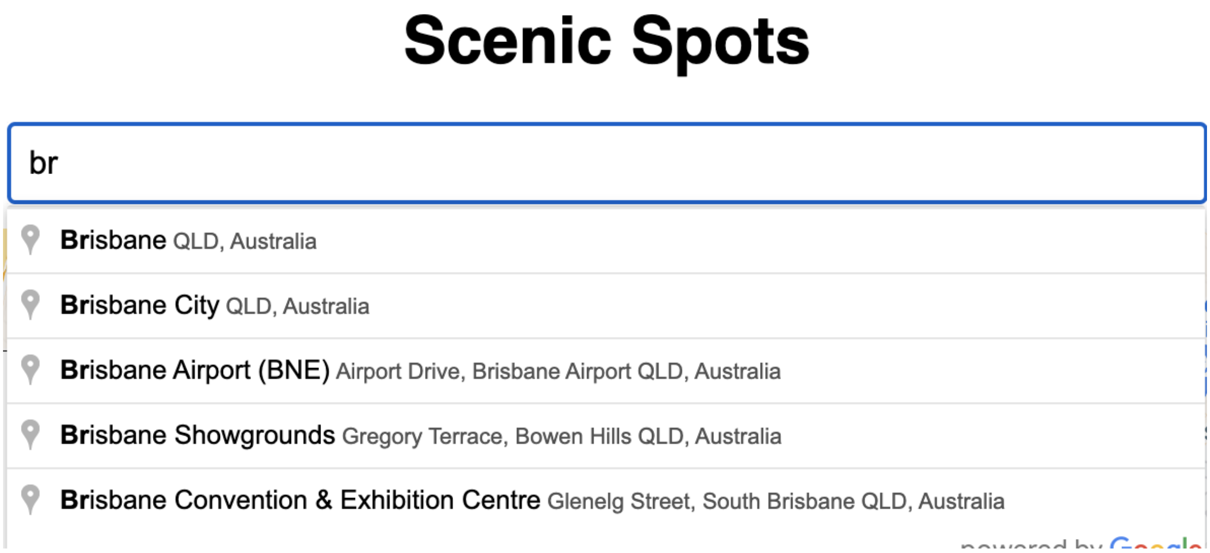
(2 marks)

Answer:

References

Appendices

01 –



Scenic Spots

br

Brisbane QLD, Australia

Brisbane City QLD, Australia

Brisbane Airport (BNE) Airport Drive, Brisbane Airport QLD, Australia

Brisbane Showgrounds Gregory Terrace, Bowen Hills QLD, Australia

Brisbane Convention & Exhibition Centre Glenelg Street, South Brisbane QLD, Australia

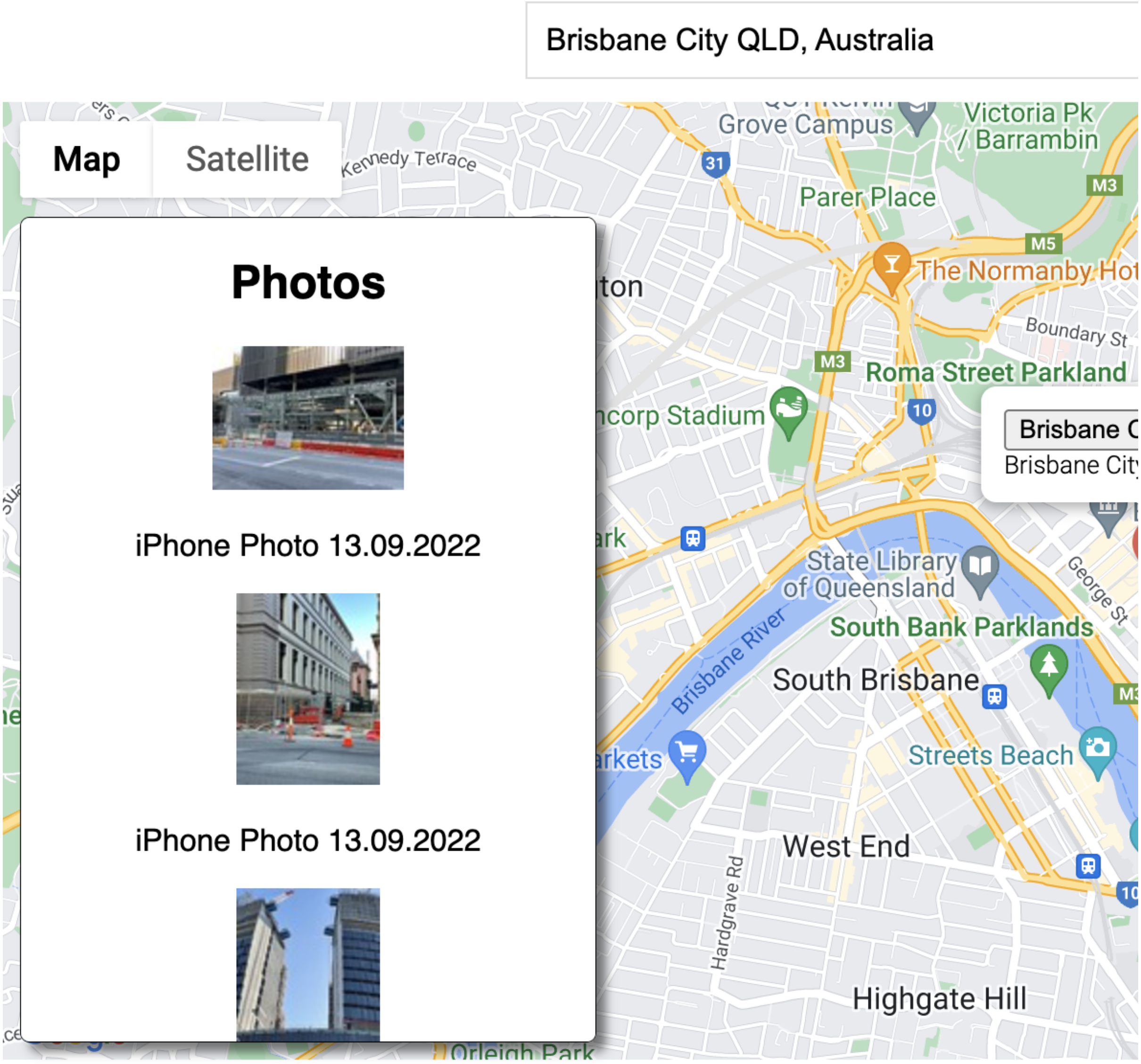
02 –

Scenic Spots

Brisbane QLD, Australia

Caloundra
Kilcoy
Woodford
Bribie Island
Coral Sea
Moreton Island
Caboolture
Moreton Bay
North Lakes
Esk
Brisbane
Brisbane QLD AU
Chermside
Brisbane
North Stradbroke Island
Plainland
Sunnybank
Ipswich
Laidley
M1
A17
A2
A5
M5
M7
M2
M6
6
85
58
31
37
47
59
93

03 –



Brisbane City QLD, Australia

Photos

iPhone Photo 13.09.2022

iPhone Photo 13.09.2022

04 –

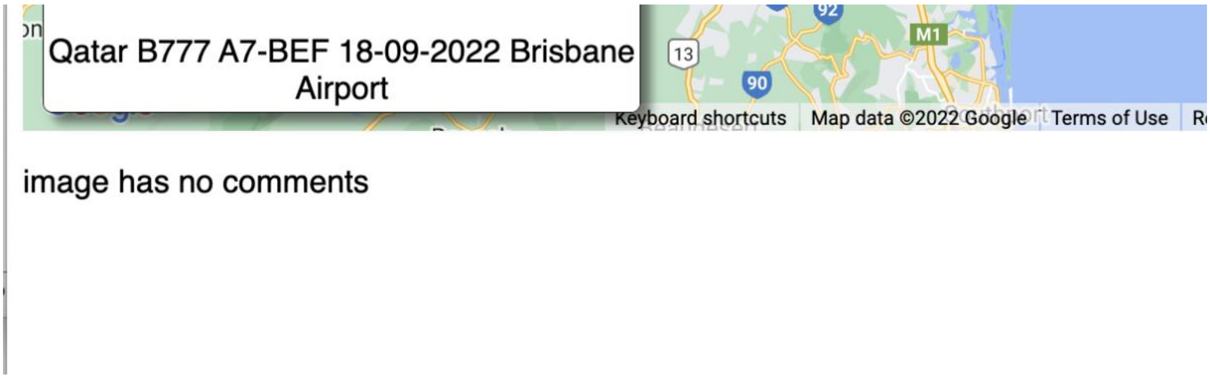
Qatar B777 A7-BEF 18-09-2022 Brisbane Airport

Visual Images1: Wonderful metal sculpture! HCT!

Kris Olin: [https://www.flickr.com/photos/irenesbest] Thank you for your comment!

Christa Annarumma: Sehr schön 😄HaPpY CrAzY Tuesday 😄

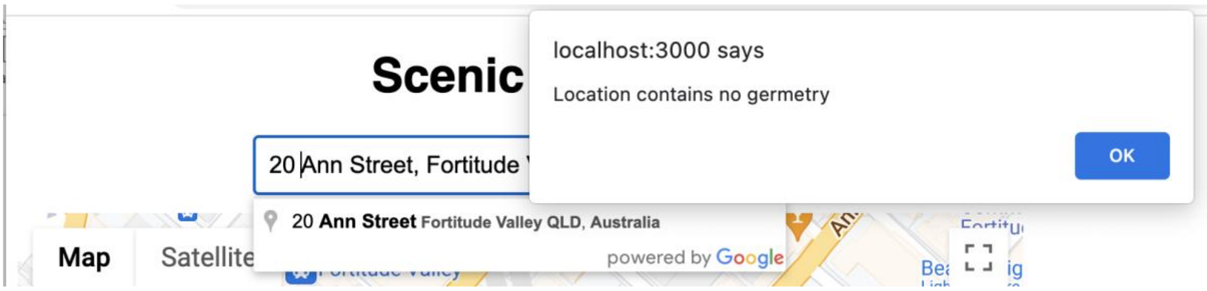
05 –



Qatar B777 A7-BEF 18-09-2022 Brisbane Airport

image has no comments

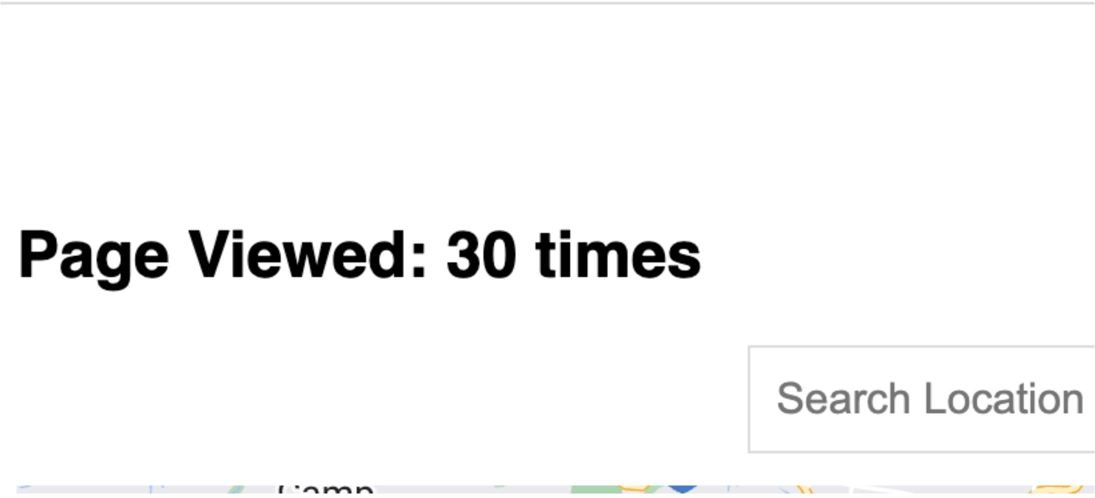
06 –



localhost:3000 says

Location contains no germetry

07 –



Page Viewed: 30 times