2021

Anh Trung Nguyen

N10042377

9/19/2021

Place2C!



CAB432 Assignment 1

Table of Contents

[Mashup Purpose & description 2](#_Toc82970277)

[Services used 2](#_Toc82970278)

[Flickr REST API (v.1.1) 2](#_Toc82970279)

[Bing Map V8 Web Control 2](#_Toc82970280)

[Mashup Use Cases and Services 2](#_Toc82970281)

[Location Images Search for a nearby Place 2](#_Toc82970282)

[Tourist Attraction Location Search on Map 2](#_Toc82970283)

[Technical breakdown 2](#_Toc82970284)

[Architecture and Data Flow 2](#_Toc82970285)

[Deployment and the Use of Docker 4](#_Toc82970286)

[Test plan 4](#_Toc82970287)

[Difficulties / Exclusions / unresolved & persistent errors 4](#_Toc82970288)

[Extensions 4](#_Toc82970289)

[User guide 5](#_Toc82970290)

[References 7](#_Toc82970291)

[Appendix A – Webpage Screenshots 7](#_Toc82970292)

[Appendix B – Dockerfile 10](#_Toc82970293)

Introduction

### Mashup Purpose & description

The Place2C! is a web mashup application, which is a combination of two APIs FlickrAPI and BingMapsAPI, allows user to find any locations around the world by showing it on the Bing map along with reference images of that location. This application would be ideal for travelling lovers, who also like to explore more places from anywhere just by a single search.

### Services used

#### Flickr REST API (v.1.1)

Returns a collection of images from Flickr matching a query based on a specific location.

Endpoint: http://api.flickr.com/services/rest

Docs: <https://www.flickr.com/services/api/request.rest.html>

#### Bing Map V8 Web Control

Returns a static map from Bing and display locations on the map matching a searched query.

Endpoint: http://www.bing.com/api/maps/mapcontrol

Docs: <https://docs.microsoft.com/en-us/bingmaps/v8-web-control/>

## Mashup Use Cases and Services

#### Location Images Search for a nearby Place

|  |  |
| --- | --- |
| As an | International tourist |
| I want | To know what interesting things that this certain location got |
| So that | I can prioritize where to go in my short trip |

#### Tourist Attraction Location Search on Map

|  |  |
| --- | --- |
| As a | Tourist |
| I want | To know where places for tourist in the city are |
| So that | I can go there to visit |

## Technical breakdown

### Architecture and Data Flow

Place2C! application is a webapp with a single webpage consists of two main components: client side and server side.

The client side is responsible for building webpages by using Pug, which is a modern engine to write HTML that provides many useful features as well as organizes HTML file to be much simpler and easier to read. There are two main Pug file of the application, “layout.pug” is overall structure for the webpage and “index.pug” is the main body content of the page. Besides the design, Javascript files help to process the submit form which is the only search form of the webpage and request to hit the APIs from the server side. When the search form is submitted, the event handler in “submitsearch.js” will use function from “bingmapapi.js” to check if the location target is valid first, then if only it is valid, there will be requests to get data from both APIs: Flickr images and search the location on Bing Map. In “flickrapi.js”, Flickr images data is requested from the server side and will be processed to be displayed on the webpage. In “bingmapapi.js”, there are functions to search all matching location results to pin them on the map. Also, if there are many locations with the same name, they will be labeled by number and the map will be displayed in a view that contains all the results (Appendix A2).

The server side has two routes: “index.js” for rendering “index.pug” – the homepage and “search.js” for hitting Flickr API to retrieve image results based the searched query. In route “search.js”, I process the Flickr endpoint with the target location then use the endpoint to hit Flickr API and with response data, I compressed it into a JSON object so in “flickrapi.js” of client side, I can call the route to fetch the JSON data.

Graphical user interface, text, application

Description automatically generated with medium confidence

*Figure1: Place2C! application structure*

### Deployment and the Use of Docker

Docker was used to containerize the application by using a Docker file to assemble an image before executing it in Docker. In the Dockerfile, Docker will grab the node version then copy the whole content of the application folder to the current directory which contains the Dockerfile. In the working directory, the Docker will install nodemon globally and any other necessary app dependencies. Finally, Docker exposes the webapp at port 3000 then start the app. (Appendix B)

### Test plan

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Expected Outcome | Result | Screenshot Guide  (Appendix ) |
| Search for location | Display related images results from Flickr | PASS | A1 |
| Search for location | Display correct location on Bing Map | PASS | A1 |
| Show what location target is searched | Display correct searched location right above image result container | PASS | A1 |
| Search for invalid location | Prevent submission and alert the user | PASS | A3 |
| Refresh the submit form | Search bar and search value is reset after each search | PASS | A1 |
| Search for new location after the previous search | Display results correctly without any conflicts with previous search | PASS | A2 |
| Control map view | Display map correctly in preferred view mode (Road, Aerial, Bird’s eye), zoom in or out and self-locate | PASS | A5 |
| Search for location with same name but many spots | Pin all results on the map correctly and view the map in a way that can show all possible location | PASS | A2 |

Difficulties / Exclusions / unresolved & persistent errors

My major roadblock was how to use the retrieved data from FlickrAPI of server side in client side. Then I found out the “fetch" function in Wikipedia example that showed me the way to put the data into JSON response so I could call it in client side.

## Extensions

There is one function that I would like to extend my webapp, it is display Flickr image results on the Bing map itself. When user clicks on the location pin, there will be an info box appear on top of the pin and show a group of Flickr images about that location. I tried to implement the feature, but it seemed to be difficult to me to make it look good enough and it began becoming messier, so I have to put the image results on the side of the map.

## User guide

My mashup application is a single page web page which is also my homepage like this screen:

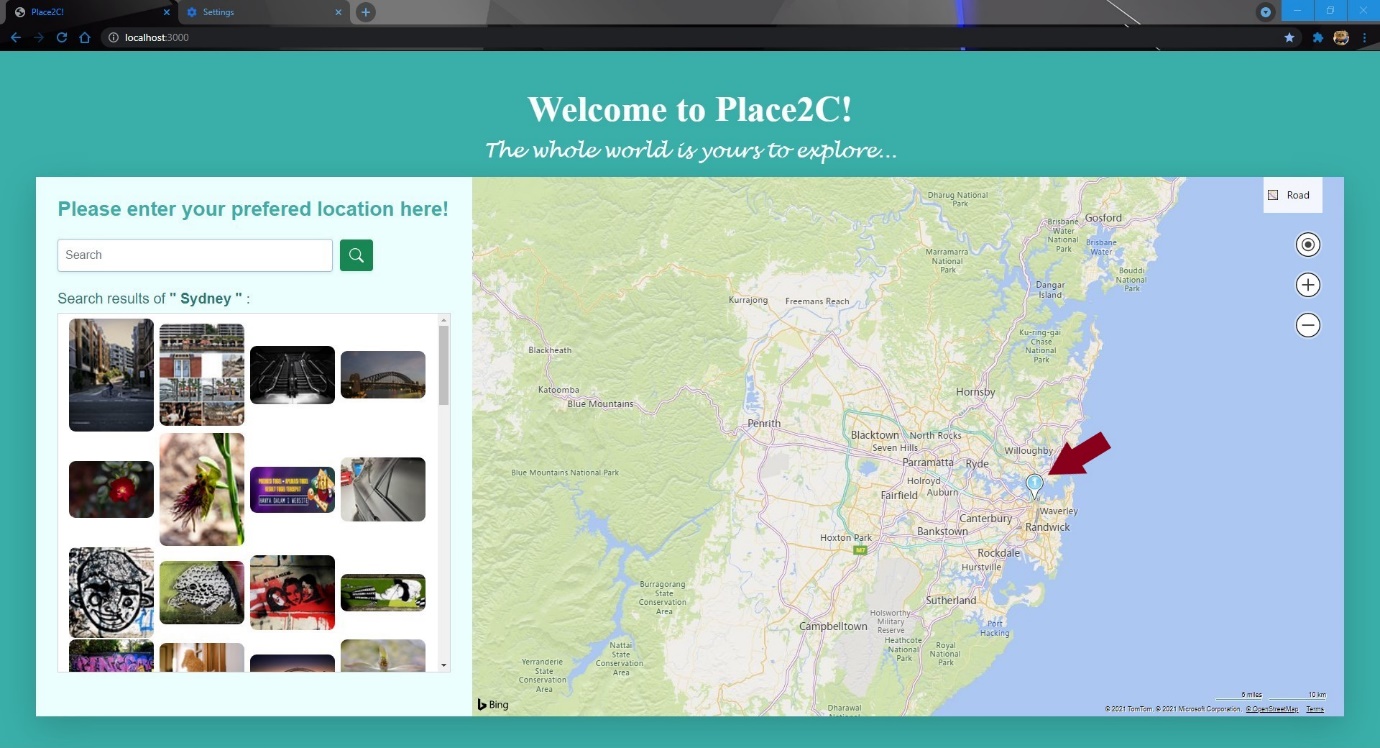
Website, map

Description automatically generated

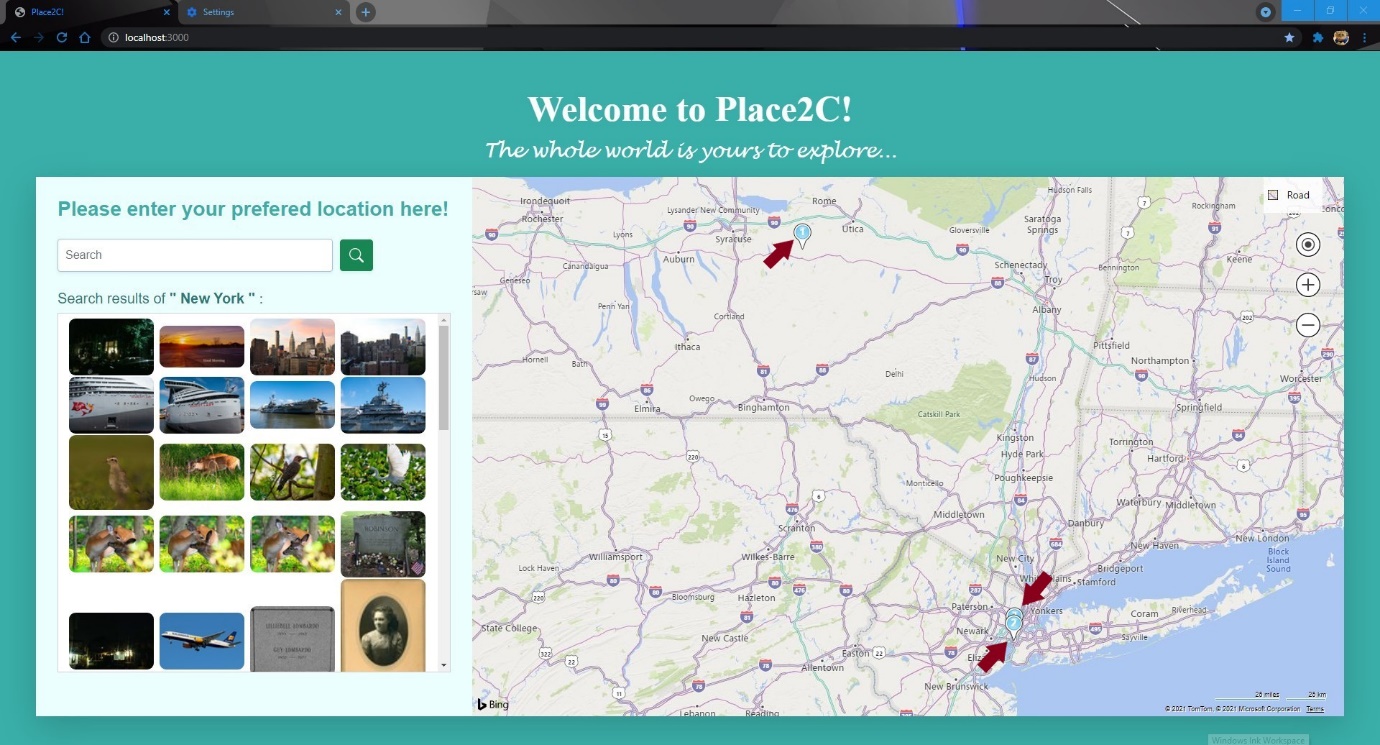
The Bing Map default state is always set to be user’s location.

To search for location, we just need to enter into the search bar. And there will be image results of the location on the left side and along with it, the Bing map will show the location results.

- Searched location with only 1 matched result on the map:



- Searched location with multiple matched results on the map:



Besides, there are also some functions for controlling the view of the map

- Set map view modes:

Graphical user interface, website

Description automatically generated

- Locate user and Zoom in/out:

Graphical user interface, map

Description automatically generated

## References

None.

## Appendix A – Webpage Screenshots

1. Search location with only one matched result:

Graphical user interface, website, map

Description automatically generated

2. Search location with multiple matched results:

Graphical user interface, map

Description automatically generated

3. Search invalid location:

Graphical user interface, website

Description automatically generated

4. Change map view mode:

Graphical user interface, website

Description automatically generated

5. Map after zooming in:

Graphical user interface, map

Description automatically generated

## Appendix B – Dockerfile

Text

Description automatically generated