CAB432 – Cloud Computing

Assignment 1 Report

Student Name: Jason Zhong

Student Number: 9674985

Introduction

The product mashup for this project is named WorldScope. WorldScope aims to empower users to browse cities around the world to plan their future travels, based on the use of Leaflet, Flickr API and Clarifai API. In WorldScope, users are able to search for cities from a map and have a look at some photos of the cities. In addition, a photo tagging service will show the users some tags of the photos in order that the users can find similar photos that they are interested in.

At the beginning, there will be a map marked with the location of some famous cities as tourism recommendations for users to browse. A search bar is also provided for users who want to search for some other cities. When users click a marker or focus on a particular city from the search, they will be able to see some photos in a penal. After they choose a photo, there will be some tags provided by the owner and analysed by Clarifai. Provided that they click a particular tag, they will be able to see similar photos taken nearby based the tag. The detailed services will be descripted later and a user guide will be provided in Appendix A.

APIs

Leaflet

https://leafletjs.com/

Leaflet is an open-source and lightweight map API used to present the geographic content of cites around the world and their photos found by Flickr. The users can be aware of where exactly the city is and the photos were taken, through the map created by Leaflet. It is a client-side API that can be accessed and utilized in JavaScript after including its SDK.

Flickr API

https://www.flickr.com/services/api/

Flickr API is another open-source API used to get some photos of particular cities. It is also used to find similar photos within a region and searched with tags. In WorldScope, flickr.photos.search will be used as the endpoint to search photos, including the location and tags data of the photos. HTTP request is used to fetch the data from Flickr.

Clarifai API

https://clarifai.com/

Clarifai API is used here as an image classifier to detect some tags of photos obtained from Flickr so that the users can be aware of what the photos are about and focus on

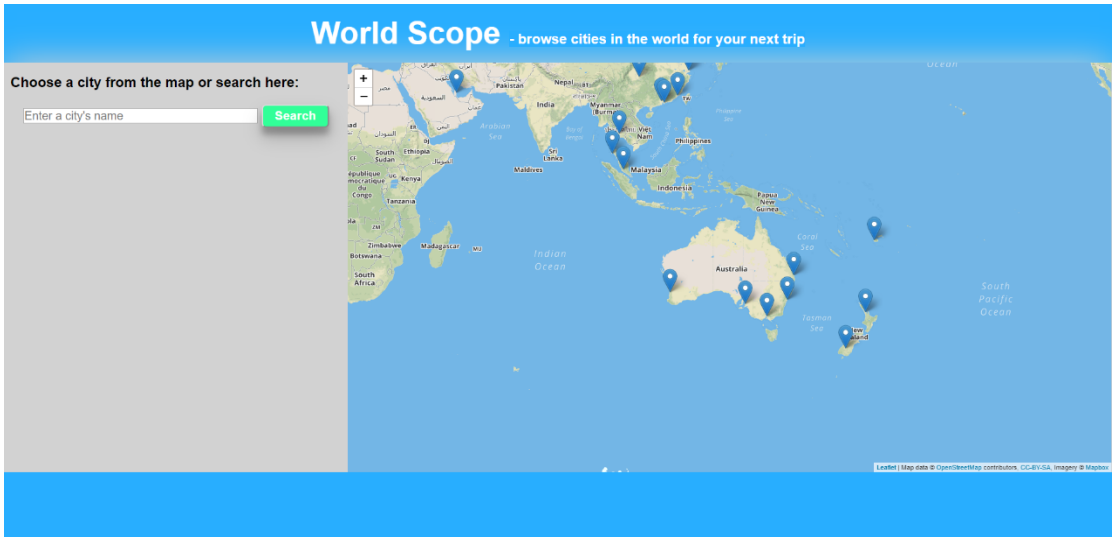
a specific type of photos. Since it has been wrapped as a library, the API can be utilized through the installation from NPM.

User cases

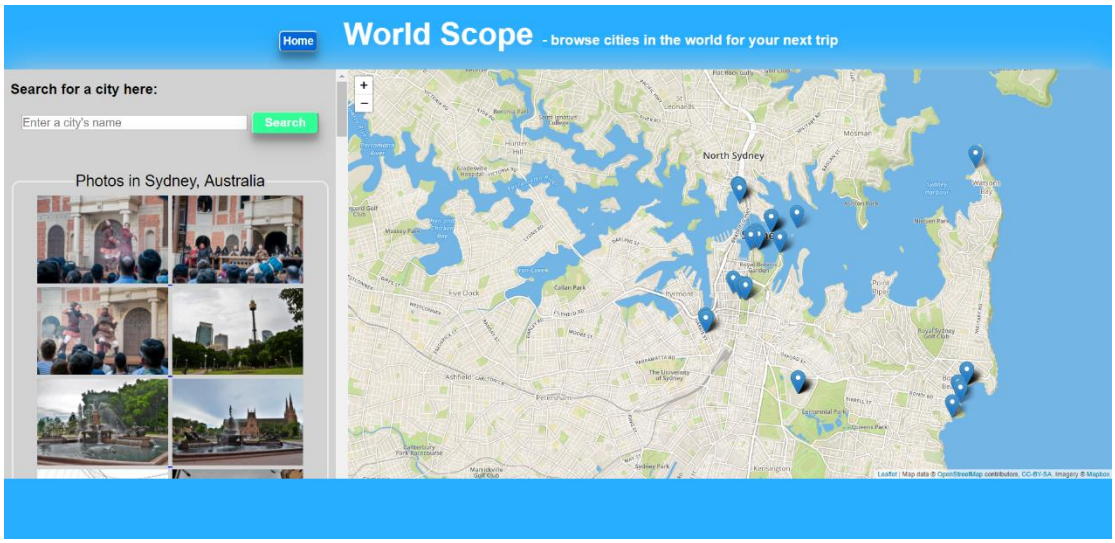
Case 1

As a user, I want to browse some cities around the world with their photos, so that I can find a city to plan my next travel.

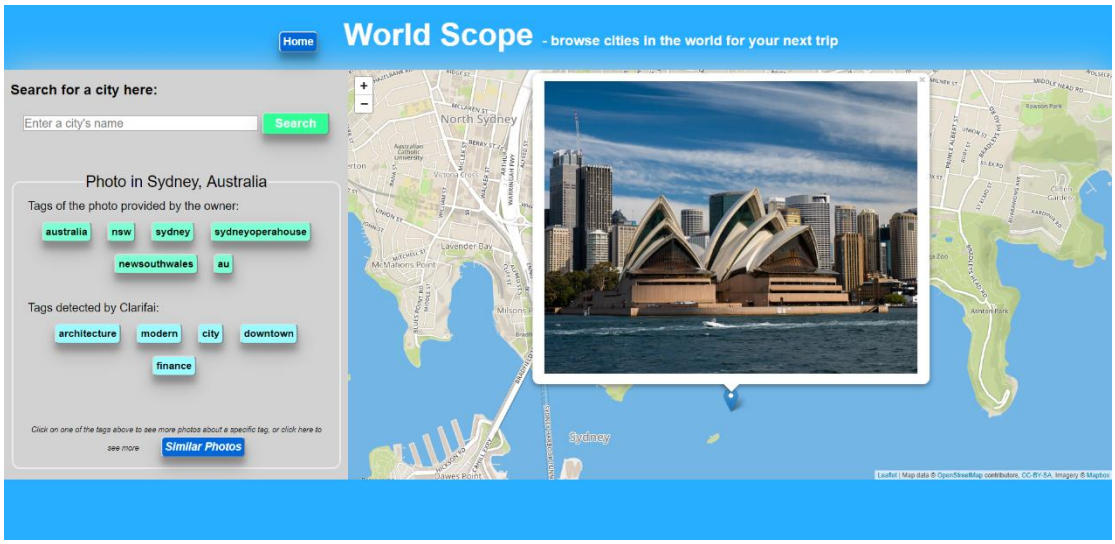
The users are provided with a map and some famous cities in the world as suggestions for future travels. There is also a search bar which enables the users to search for some other cities as they want to see.



When users click a marker or focus on a particular city from the search, they will be able to see some photos in a penal.



After they choose a photo, there will be some description and tags provided by the owner and analysed by Clarifai.

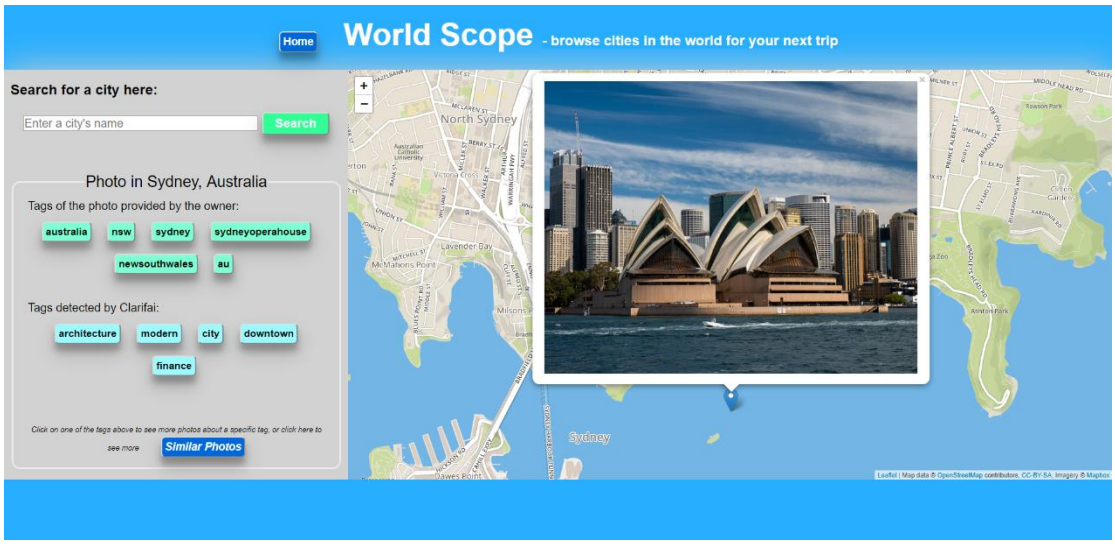


APIs used for this case: Leaflet, Flickr API, Clarifai API

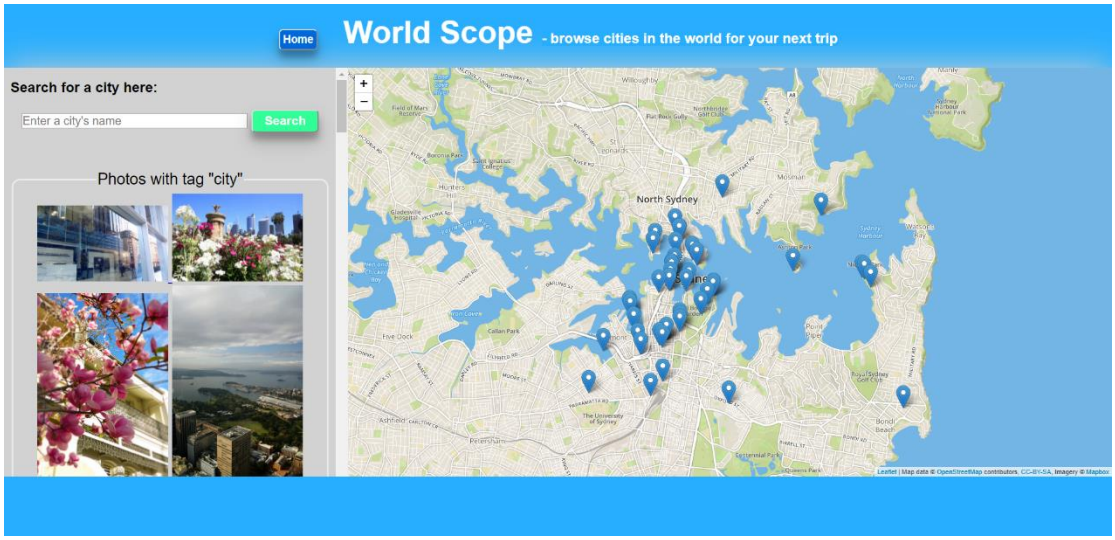
Case 2

As a user, I want to search for a particular category of photos in a city, so that I can focus on enjoying photos about a specific place or landmark that attracts me.

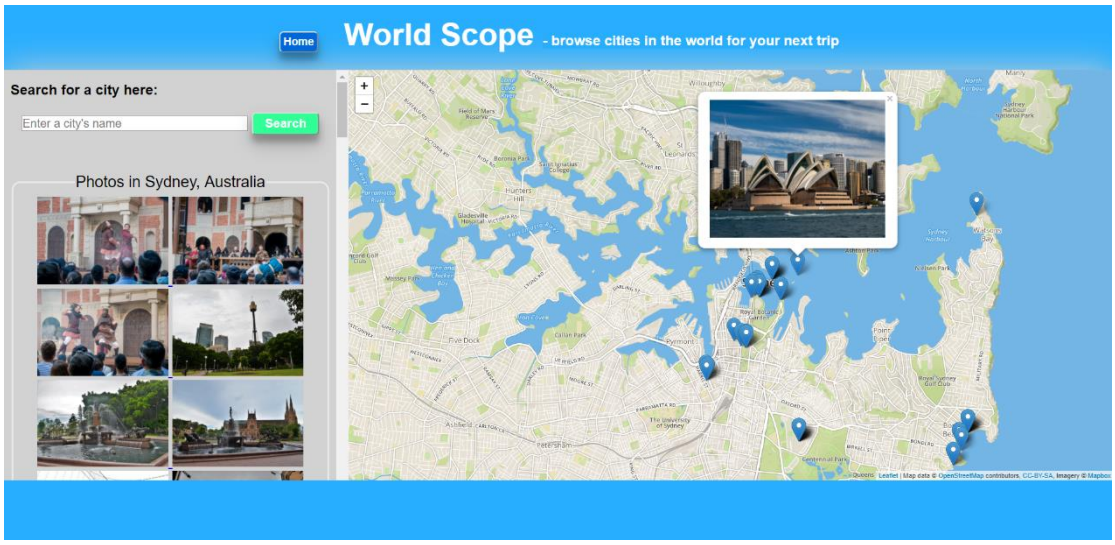
The tags presented for each photo can be clicked and directed to more photos with a specific tag, in order that the users can focus on browsing a view spot or landmark in a city.



The tags to be chosen are divided into two parts: provided by the owner as direct and precise tags and detected by Clarifai's classification model. This offers a wide range of choices to the users, and even when a photo has no tags from its owner, the users can still focus on a specific type.



The users can also preview the photos from the map markers so that they are able to focus on a particular place or view spot.

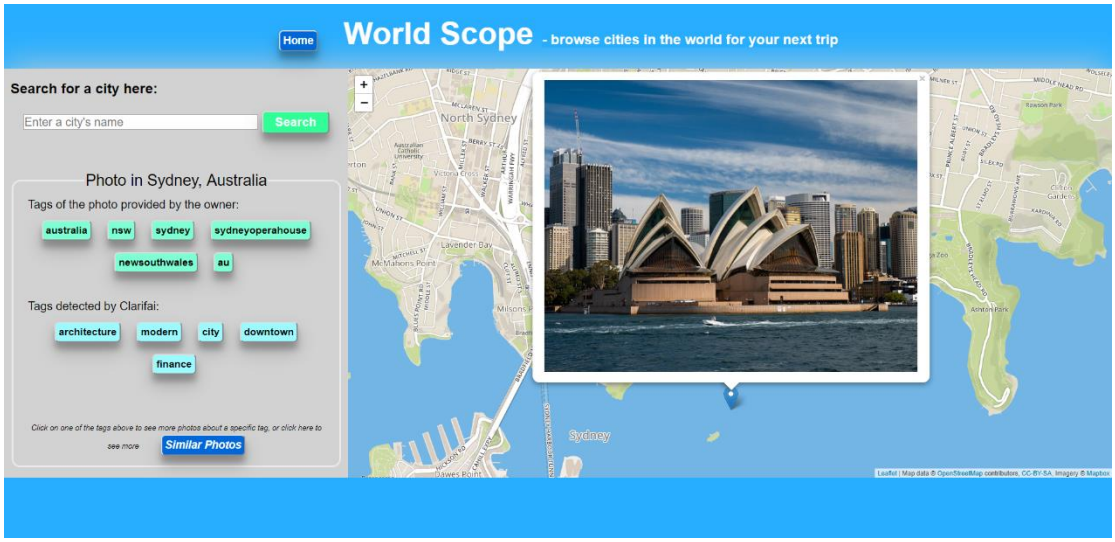


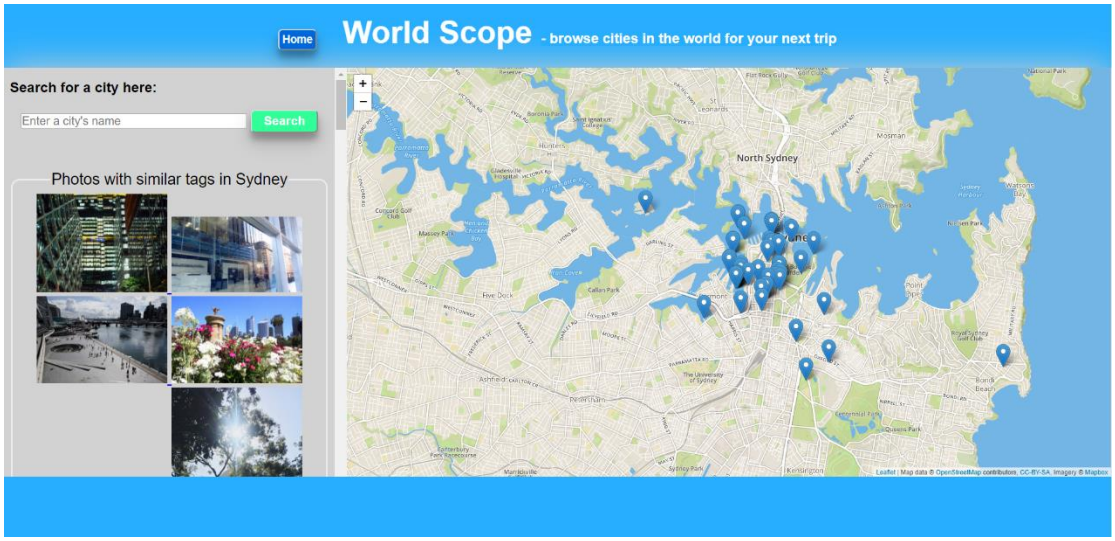
APIs used for this case: Leaflet, Flickr API, Clarifai API

Case 3

As a user, I want to see more similar photos after I see an interesting one, so that I can know more about the city with similar views.

On the bottom of the penal (under the tags), there is a button which after clicked will show more similar photos with almost the same tags.



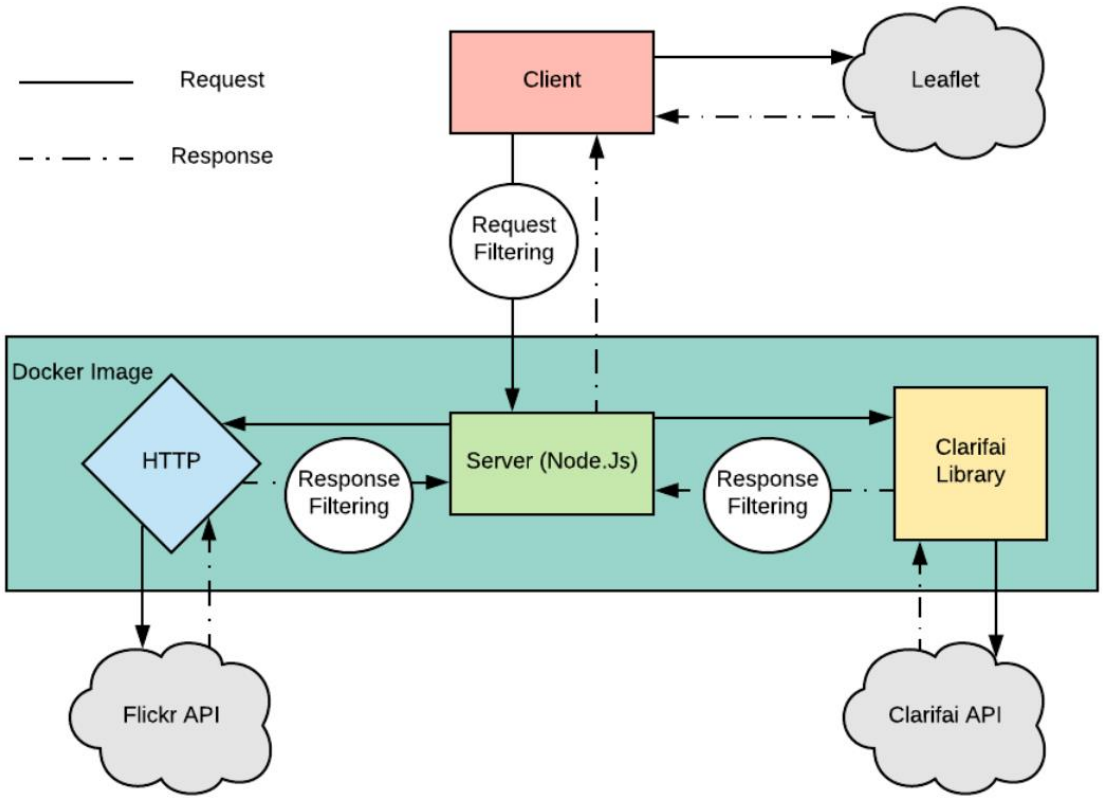


APIs used for this case: Leaflet, Flickr API, Clarifai API

Technologies

Architecture

As can be seen from the figure below, most of the services are provided from the server side, where is contained by Docker. At the beginning, the client request to the server using GET method, afterwards the server will request some data from Flickr API and/or Clarifai API based on the filtered client request content. In the next stage, the server will response the client with the data by writing it in the HTML page. Finally, the client script will read a small amount of the data in HTML and request Leaflet to display a map using the data.



Client

The technologies used in client side includes HTML template (the dynamic content is actually generated by Handlebars from the server), CSS for styling, and JavaScript. The client-side JavaScript is used to extract some important data from the HTML and display a map with markers. In addition, it writes redirect links to buttons, and handles the validation check of the input text from the search bar in order to ensure that the users are not searching with empty string.

Leaflet map data is requested from the client because it is defined as a client-side API and shows a map using some of the response data of Flickr API which has been secured from the server side and sent to the client.

Server

Node.js is adopted as the backend technology in the server side. It utilizes the following libraries to implement the functionality:

- **Express**: used to construct a RESTful server which provides the routing and interact with the client.
- **Body-parser**: used as a middleware to parse incoming requests.
- **Handlebars (hbs)**: used as the view engine to render dynamic HTMLs (will be describe in detailed later)
- **Fs-sync**: used to read the data file (csv format) synchronously

- **Clarifai**: the node-version library of Clarifai to utilize its API to detect image tags

Data of photos is requested through HTTP to the Flickr API. When a photo is selected, its URL is then sent to the encapsulated Clarifai library to access the API and return a series of tags.

Handlebars (hbs)

Handlebars is one of the most popular view engines in Node.js, which empowers the server to render HTML templates by inserting server-side data to an HTML in order to produce dynamic webpages. For example, the names of cities in the search bar drop list, and the URLs of photos, are both produced by hbs using the data in server.

Filtering

Request Filtering from Client:

The request sent from client to server uses GET method through the URL. The parameter values in the request will be pre-processed into some particular formats before they are used to request APIs in the server. For example, the city's value for search will be split by a comma to extract the city's name and country's name (if entered) so that they can be used to query of a specific city.

Response Filtering in Server:

The responses from Flickr API and Clarifai API are both in JSON format. As a result, there could be a range of data for each JSON response including some properties which are useless. If all the data were passed to the client, it would be a very heavy load and reduce the response speed to the client significantly. Therefore, the filters extract only required data from the JSONs and reformat it into useful strings to response to the client.

For instance, the figure below presents part of a response JSON from Flickr when searching photos in Brisbane. There are a lot of properties returned, however, the server only needs the data indicated in the next figure.

{"photos":{"page":1,"pages":745,"perpage":250,"total":"186121","photo":
[{"id":"29790775537","owner":"60988113@N06","secret":"28a9f0137e","server":"1842","farm":2,"title":"yellow","ispublic":1,"isfriend":0,"isfamily":0,"tags":"macromondays hmm cogwheel yellow
fun","latitude":"-27.500392","longitude":"152.972319","accuracy":"16","context":0,"place_id":"UmFYXqFTWr_NFvTzXw","woeid":"28589321","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"43817605125","owner":"60455048@N02","secret":"ea531452fc","server":"1862","farm":2,"title":"Adelaide St Below George St","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane adelaidestreet
georgestreet jholt painter glazier signwriters
jholtpaintersglazierssignwriters","latitude":"-27.469249","longitude":"153.023500","accuracy":"14","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"geo_is_friend":0,"geo_is_cont
act":0,"geo_is_public":1},{"id":"29788546367","owner":"75704644@N02","secret":"f99d85d090","server":"1897","farm":2,"title":"Mobile
Monday","ispublic":1,"isfriend":0,"isfamily":0,"tags":"","latitude":"-27.454217","longitude":"153.038697","accuracy":"16","context":0,"place_id":"kmO84UBWULircL8U","woeid":"7225514","geo_is_family":0,"geo_is_fr
iend":0,"geo_is_contact":0,"geo_is_public":1},{"id":"44716208281","owner":"133320179@N02","secret":"e6cf0bbc81","server":"1884","farm":2,"title":"Brisbane: View from the rooftop pool of the Spire
Residences","ispublic":1,"isfriend":0,"isfamily":0,"tags":"","latitude":"-27.467764","longitude":"153.023071","accuracy":"11","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"ge
o_is_friend":0,"geo_is_contact":0,"geo_is_public":1},{"id":"29779698917","owner":"133320179@N02","secret":"d3f3e7f657","server":"1867","farm":2,"title":"Brisbane: Infinity Pool of the Spire
Residences","ispublic":1,"isfriend":0,"isfamily":0,"tags":"","latitude":"-27.467764","longitude":"153.023071","accuracy":"11","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"ge
o_is_friend":0,"geo_is_contact":0,"geo_is_public":1},{"id":"44716198271","owner":"133320179@N02","secret":"ed8e24384d","server":"1865","farm":2,"title":"Spire Residences, Brisbane: Infinity rooftop
pool","ispublic":1,"isfriend":0,"isfamily":0,"tags":"","latitude":"-27.467764","longitude":"153.023071","accuracy":"11","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"geo_is_f
riend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"44667091802","owner":"55630656@N04","secret":"ae8af8ae81","server":"1859","farm":2,"title":"Sentinels","ispublic":1,"isfriend":0,"isfamily":0,"tags":"cityscape street milkfactory australia people
sentinels southbrisbane milkfactorybar candid tree industrial streetscape streetphotography urban vertical wall building line
aligned","latitude":"-27.471015","longitude":"153.015274","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1}
,{"id":"43996726844","owner":"32536320@N05","secret":"ec483f4e4b","server":"1846","farm":2,"title":"View from my room at the Adina apartments, Brisbane
city","ispublic":1,"isfriend":0,"isfamily":0,"tags":"australia queensland brisbane park southbank view river
brisbaneriver","latitude":"-27.468453","longitude":"153.026900","accuracy":"16","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_publ
ic":1},{"id":"43803718295","owner":"94735786@N00","secret":"fd023d03d4","server":"1886","farm":2,"title":"Brisbane River at dusk, Bulimba, Queensland","ispublic":1,"isfriend":0,"isfamily":0,"tags":"2018
australia queensland brisbane bulimba river
iphone6plus","latitude":"-27.455534","longitude":"153.053405","accuracy":"16","context":0,"place_id":"kmO84UBWULircL8U","woeid":"7225514","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1
},{"id":"42902606480","owner":"94735786@N00","secret":"61dab4e8ae","server":"1888","farm":2,"title":"Ferry wharf, Brisbane River, Bulimba, Queensland","ispublic":1,"isfriend":0,"isfamily":0,"tags":"iphone6plus
river bulimba brisbane queensland australia
2018","latitude":"-27.450659","longitude":"153.052550","accuracy":"16","context":0,"place_id":"kmO84UBWULircL8U","woeid":"7225514","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"43992206954","owner":"82155102@N07","secret":"10cae83492","server":"1868","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.469062","longitude":"153.026072","accuracy":"16","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"42900193880","owner":"82155102@N07","secret":"6916182bc0","server":"1865","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.470545","longitude":"153.024113","accuracy":"16","context":0,"place_id":"I_iCgvNTWr.9IuNS0A","woeid":"28589319","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"43992202454","owner":"82155102@N07","secret":"c3b2ccaf07","server":"1889","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.474356","longitude":"153.020594","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"42900187410","owner":"82155102@N07","secret":"ffb3cc4fa8","server":"1862","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"queensland australia
au","latitude":"-27.474939","longitude":"153.021144","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"44661030302","owner":"82155102@N07","secret":"a6f0c562b1","server":"1843","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"queensland australia
au","latitude":"-27.475031","longitude":"153.021352","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"44709864791","owner":"82155102@N07","secret":"5e285700dc","server":"1884","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.478125","longitude":"153.022561","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"44709861751","owner":"82155102@N07","secret":"138f61552c","server":"1841","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.478703","longitude":"153.022880","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"43992181544","owner":"82155102@N07","secret":"dc5dfee91d","server":"1884","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.478548","longitude":"153.022838","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"30839413828","owner":"82155102@N07","secret":"981b9a9491","server":"1848","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.478131","longitude":"153.022530","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},
{"id":"42900158850","owner":"82155102@N07","secret":"701bf81f7f","server":"1879","farm":2,"title":"Sunday 16-09-18","ispublic":1,"isfriend":0,"isfamily":0,"tags":"brisbane queensland australia
au","latitude":"-27.478214","longitude":"153.022555","accuracy":"16","context":0,"place_id":"TCoi5k5TWry6Ph5ehg","woeid":"28676620","geo_is_family":0,"geo_is_friend":0,"geo_is_contact":0,"geo_is_public":1},

```
"farm": photo.farm,
"server": photo.server,
"id": photo.id,
"secret": photo.secret,
"lat": photo.latitude,
"lon": photo.longitude,
"tags": photo.tags,
```

Issues Encountered

Use of the Clarifai API

Clarifai API is the API that I spent the most time compared to the other two. This is because in Clarifai API's official website only examples using Python are provided. The whole site, including its documentation, barely mentions that the Node.js library can be installed and used from NPM and how to start an instance using an API key. However, the semantics of the API in Node.js is much different from that in Python, which initially caused that I do not know how to call a model to predict an image and could not have it running successfully. It was not tackled until I searched for plenty of demonstrations of Node.js version from Google and learnt from them.

Passing Data from Server to Client

This is also a difficulty that I encounter at the beginning. Some important data, for example, the coordinates of the cities and photos that the client needs to display in Leaflet map, should be passed from the server. Eventually, I decided to use handlebars in the server side to write the important data as an attribute value in some HTML tags, such that the client-side JavaScript can read.

Implementation

In the development process, some user stories and a plan regarding the APIs to be utilized as well as the architecture of the codes were firstly made. Next, the coding stage and test stage took place with an IDE and a particular browser respectively (the IDE and browser will be described later). Each functionality was implemented in sequence according to the plan. Finally, the product was containerized using a Docker image in AWS.

IDE

The source codes of this project were all written using JetBrains WebStorm as the integrated development environment (IDE). This is because the use of IDE can immediately highlight any syntax error and provide useful debugger, which remarkably improves the efficiency of development.

Browser

Google Chrome was used as the browser during the development, as it is one of the most popular browsers at the moment and has enough support to the client side. Other browsers, for example, Firefox, may not support well the “datalist” tag of HTML.

Docker

Docker was used as the container tool for WorldScope. The content of the Docker file can be seen in Appendix B. the Docker file was used to build a Docker image such that normal machines can serve WorldScope easily as a container.

The latest LTS version of “boron” was used as the image base as it has all the dependencies that WorldScope required. Port 3000 was chosen to expose to the Node server as it was defined in the server code too. The final line (CMD npm start) just simply start the service so that users can access WorldScope.

Testing

Robustness

There are two primary solutions for the robustness, to handle the errors occurring in requesting Flickr API and Clarifai API respectively:

```
flickrReq.on('error', (e) => {
    console.error(e);
    res.render("index", {
        cities: cities,
        failureStatus: 'hidden',
        errorStatus: '',
        error: 'Error occurred in the request to Flickr API'
    });
});
```

```
}).catch(function(clarifaiError) {
    console.log(clarifaiError.toString());
    res.render("index", {
        cities: cities,
        failureStatus: 'hidden',
        errorStatus: '',
        error: 'Error occurred in the request to Clarifai API'
    });
});
```

The first one deals with the error happening in the Flickr API request, which redirects to the index page and sends an error notification. The second one handles the error in the Promise object returned by Clarifai API, which similarly redirects to the index page and sends an error statement to the users.

Test Cases

Plan	Expected Outcome	Result
1. search for a city by only city's name	Photos of the city are displayed	Pass
2. search for a city by city and country	Photos of the city are displayed	Pass
3. search with an empty string	Notify by the client	Pass
4. search with incorrect names	Notify by a search failure sentence	Pass
5. hover around city markers on the map	City's and country's names are displayed	Pass
6. click a city's marker on the map	Photos of the city are displayed	Pass
7. hover around the photo markers on the map	The photo can be previewed	Pass
8. click a photo marker on the map	Tags of the photo are displayed	Pass
9. click a photo in the penal	Tags of the photo are displayed	Pass
10. click a tag provided by the photo owner from the penal	Photos of the tag are searched and displayed	Pass

11. click a tag detected by Clarifai from the penal	Photos of the tag are searched and displayed	Pass
12. click the “similar photos” on the penal	Photos with similar tags are searched and displayed	Pass
13. search for a city in the page listing photos	Photos of the city are displayed	Pass
14. search for a city in the page listing tags	Photos of the city are displayed	Pass
15. click the home button	The index (home page) is returned	Pass
16. Flickr error occurs	Error is displayed in the page	Pass
17. Clarifai error occurs	Error is displayed in the page	Pass

The result screenshots are presented in Appendix C.

Compromise

My initial plan for displaying the photo is to acquire a large HD photo so that the users can enjoy some views better. Nevertheless, in a slow network (for example in my place) the response from Flickr API could take such long time that contributes to the request time out issue to Leaflet and fails to load the map. As a result, I can only request medium-size photos to ensure that the users can at least see a photo clearly and the map can be load in a normal network.

Extensions

Front-end Optimization

Although front-end at the moment has been managed to make as clear as possible, it looks still very plain and not attractive. Therefore, after the development of primary functionality the front-end could be optimized to make the pages more beautiful.

User and Comment System

Another extension is to allow users to create their account and comment some cities that they have travelled to. In this case, the other users can know more about a city as suggestions before their next trip. This could include registration, login and comment functionalities in WorldScope.

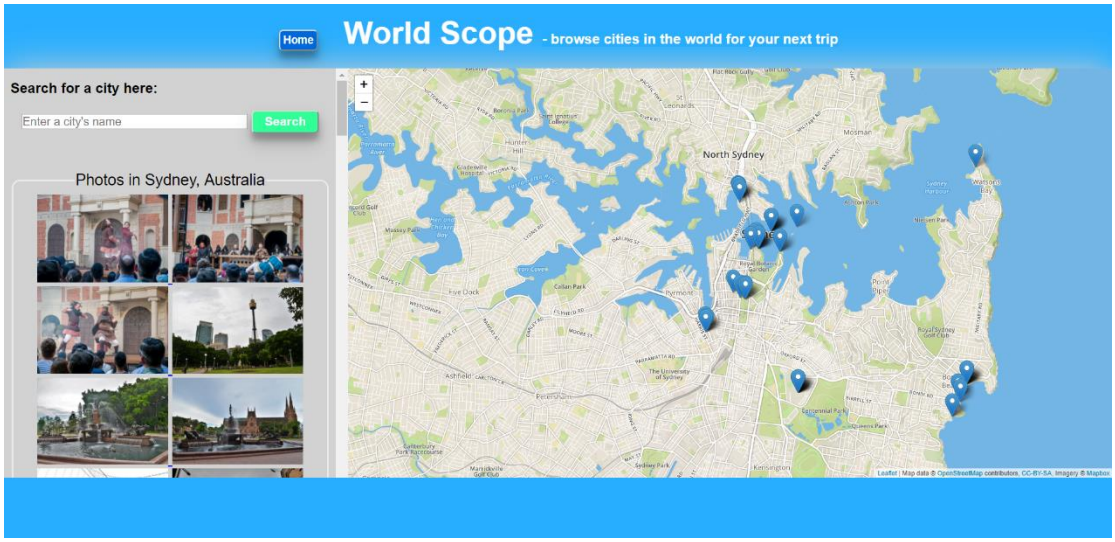
Appendix A – User Guide

Before you start using WorldScope you have to firstly ensure your network is smooth without obvious interruptions. This is because a very slow network can result in the delay of loading tens of photos and then contribute to time out issue of loading Leaflet since Leaflet needs to wait for the photos to be loaded. The following is the user guide to use WorldScope, which is very simple and straightforward.

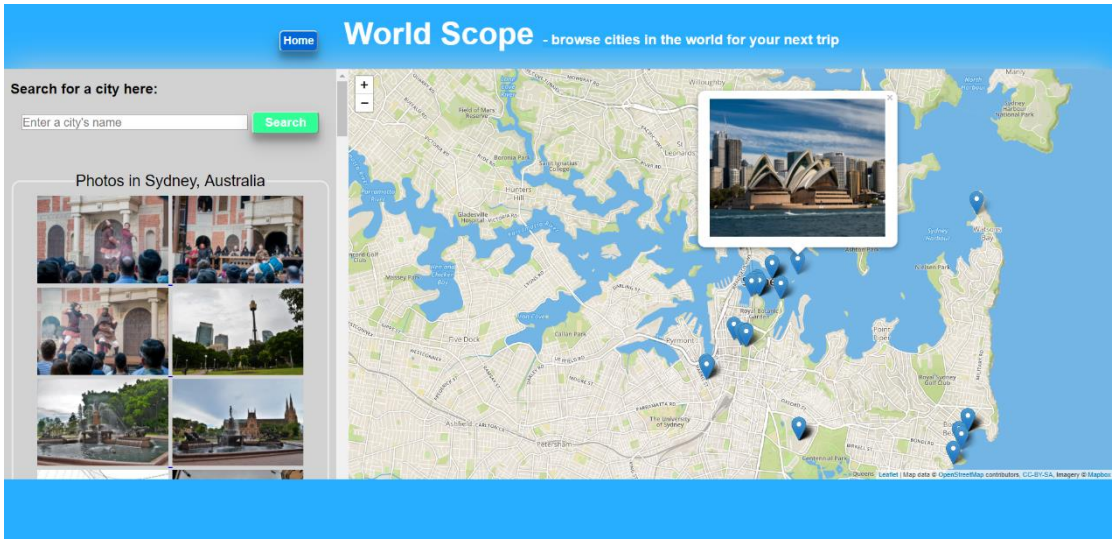
1. when you open the home page of WorldScope, you can either enter a city's name in the search bar which locates in the top left of the page, or click one marker of the recommended cities on the map, to search photos in that city.

Note that if you choose the first way to search for a city, when you are typing there will be some possible cities suggested for you. The input format could be either "cityname" or "cityname, countryname".

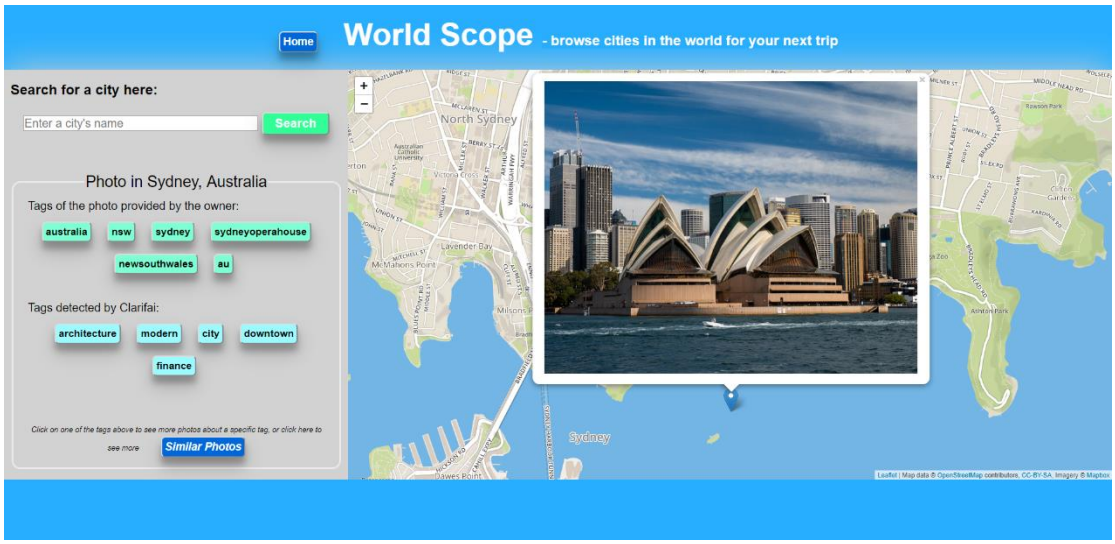
2. After you have chosen a particular city, 50 photos taken within that city's area will be displayed for you to browse on the left-hand side penal.



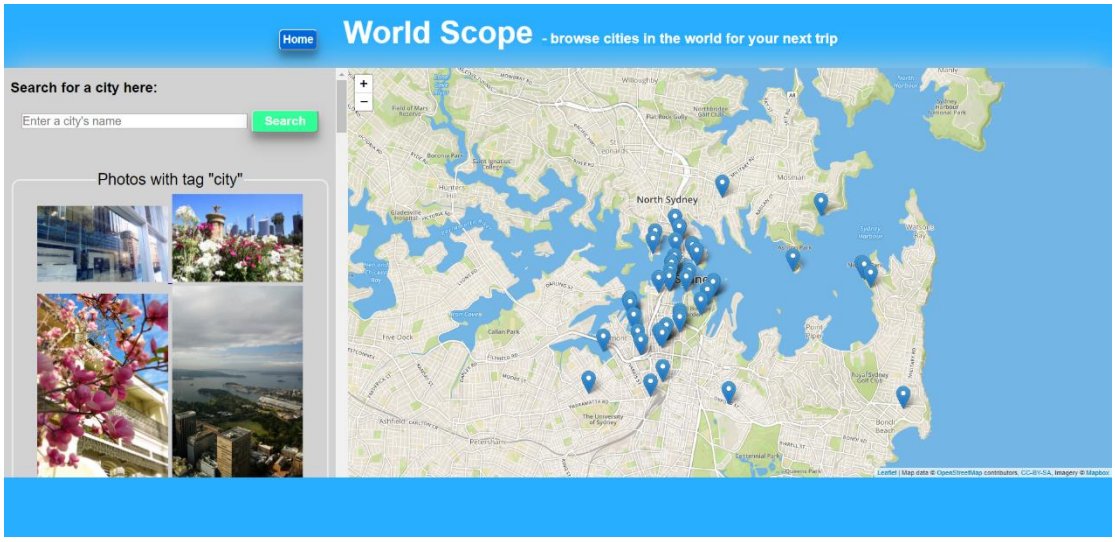
You can also preview the photos from the map by hovering around the markers.



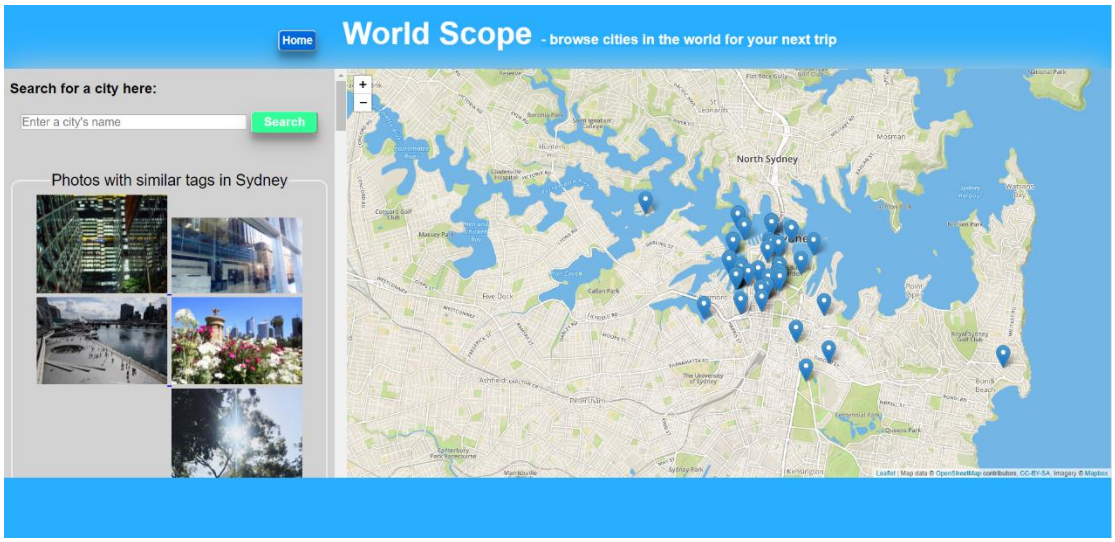
3. After click on one of the photos from the penal or marker, you will see the large version of the photo and there will be some tags provided by the photo owner and detected by Clarifai.



4. You can click one of the tags to see more photos with that particular tag. For example, if you clicked the “city” tag from the penal, you would see something like the following.



5. You can also click the “Similar photos” button on the bottom of the penal to see more photos similar to one you have chosen to know more about the city.



6. If you wanted to search for the photos of another city, you could always find the search bar in the penal to do that. Alternatively, you can also click the “Home” button on the top of the page to go back to the home page and browse again.

Appendix B – Docker File

The content of the Docker file is shown below:

```
FROM node:boron

COPY . /src

WORKDIR /src

RUN npm install

EXPOSE 3000

CMD ["npm", "start"]
```

Before building image, ensure the docker file is in the first-level directory of the WorldScope project. To build this image, just simply use the command:

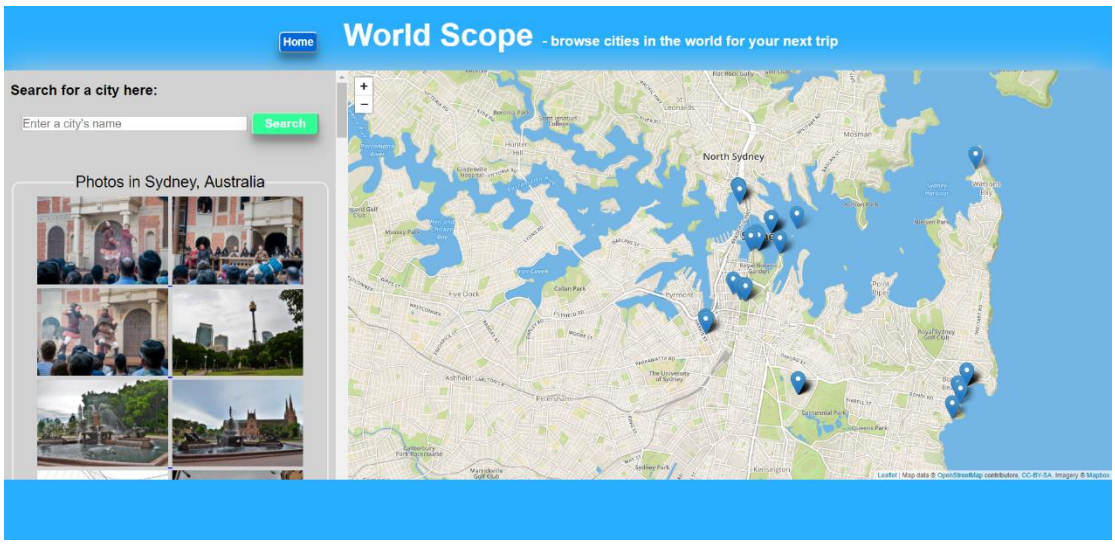
sudo docker build -t worldscope .

And to run this image, port 3000 is required to be opened for Node:

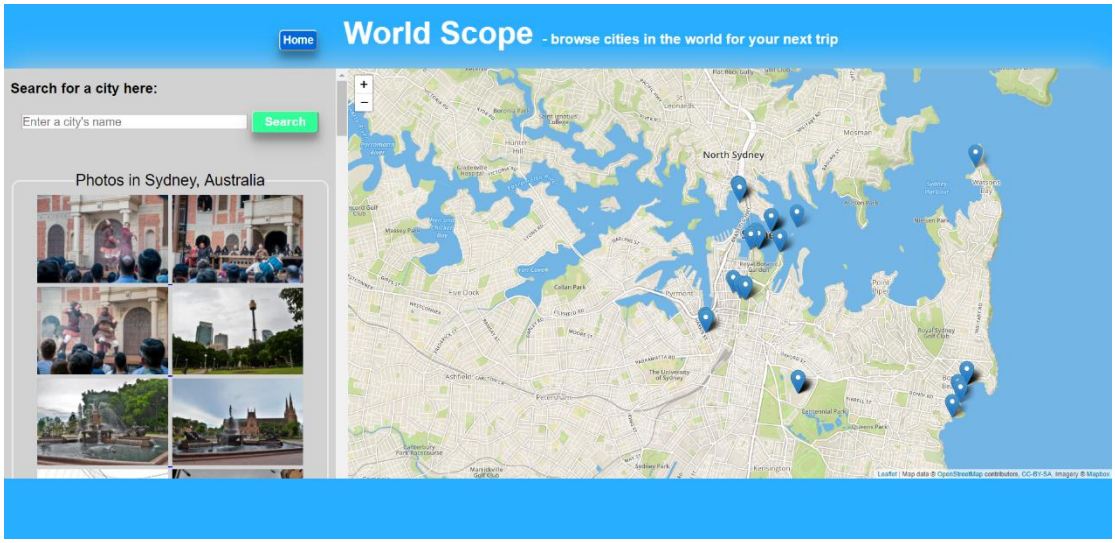
sudo docker run -p x:3000 -itd worldscope

Appendix C – Test Screenshots

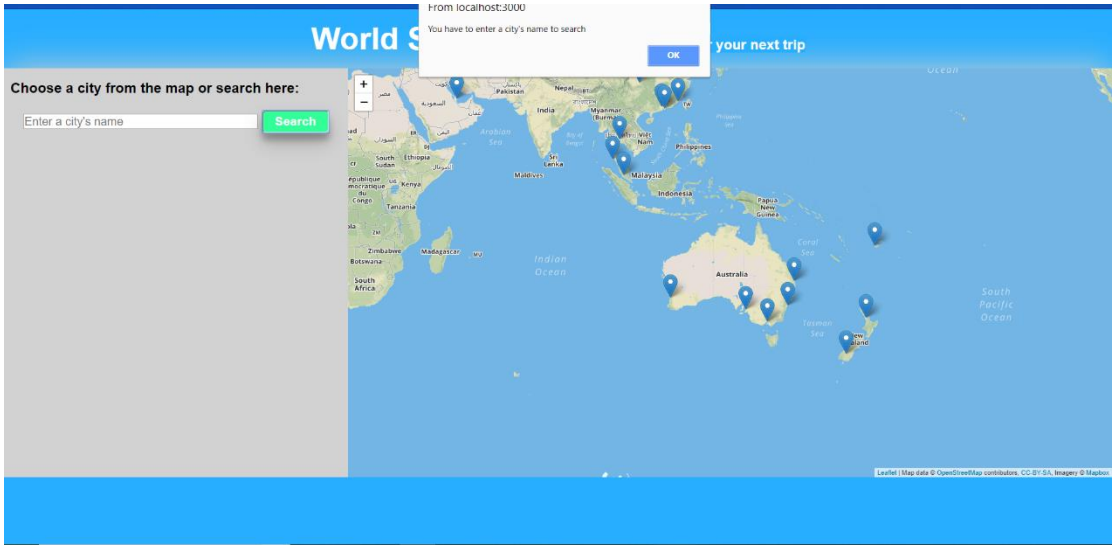
1.



2.



3.


From localhost:3000
You have to enter a city's name to search
OK
World S
your next trip
Choose a city from the map or search here:
Enter a city's name
Search

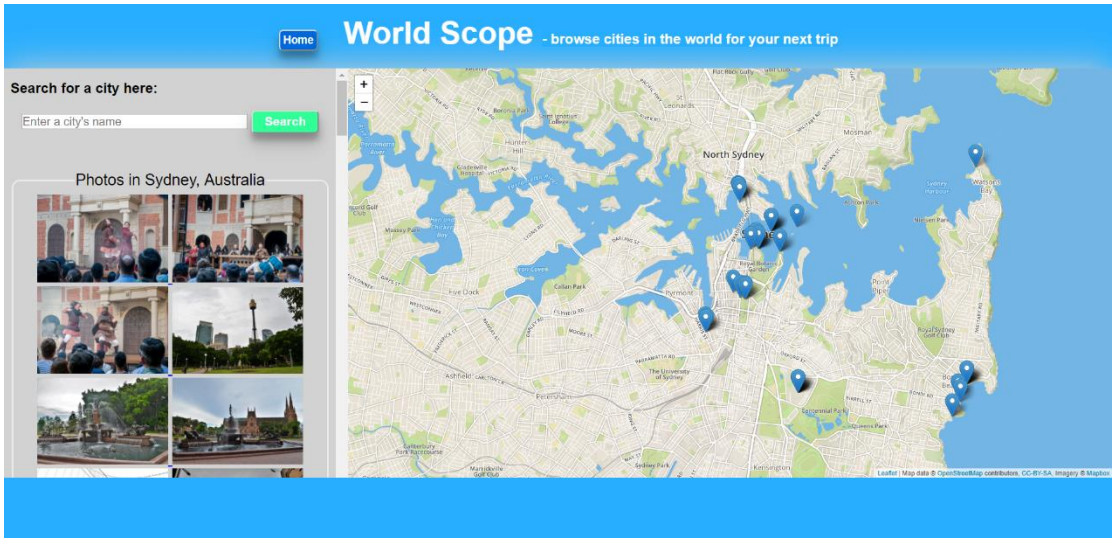
4.


World Scope - browse cities in the world for your next trip
Choose a city from the map or search here:
Enter a city's name
Search
Sorry, we can't find that city!

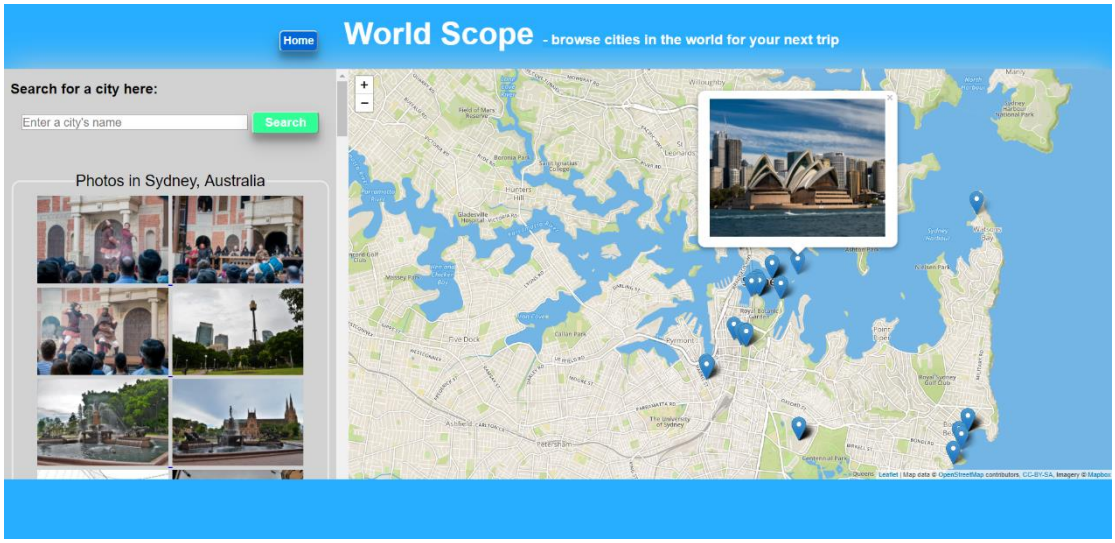
5.


World Scope - browse cities in the world for your next trip
Choose a city from the map or search here:
Enter a city's name
Search
Perth, Australia

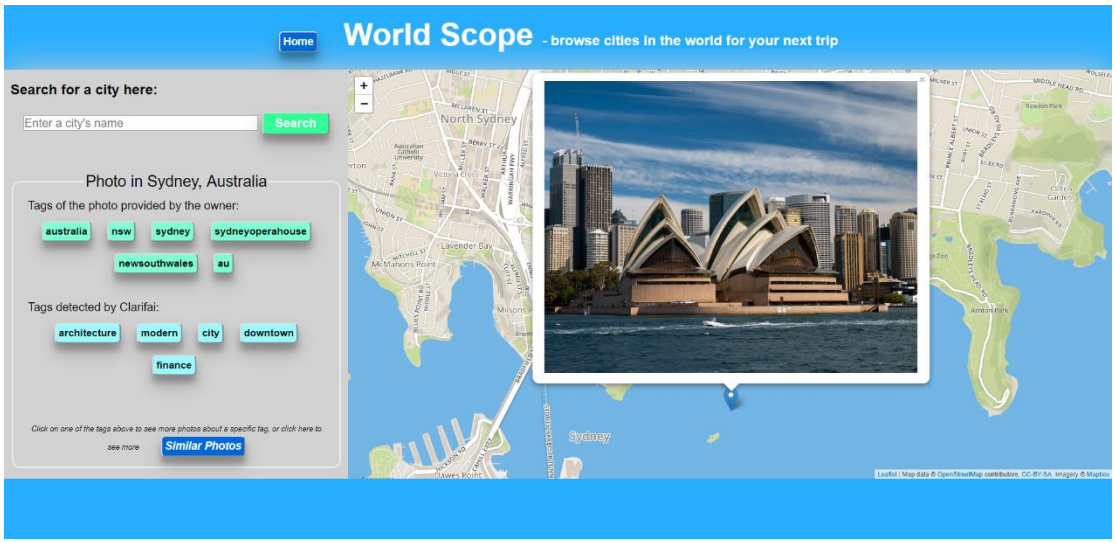
6.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos in Sydney, Australia
North Sydney

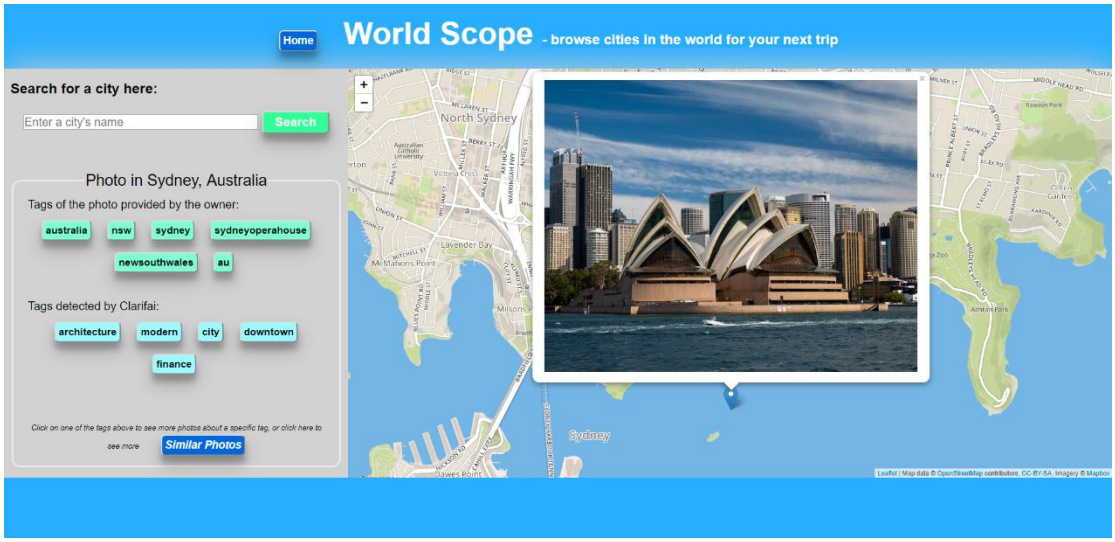
7.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos in Sydney, Australia

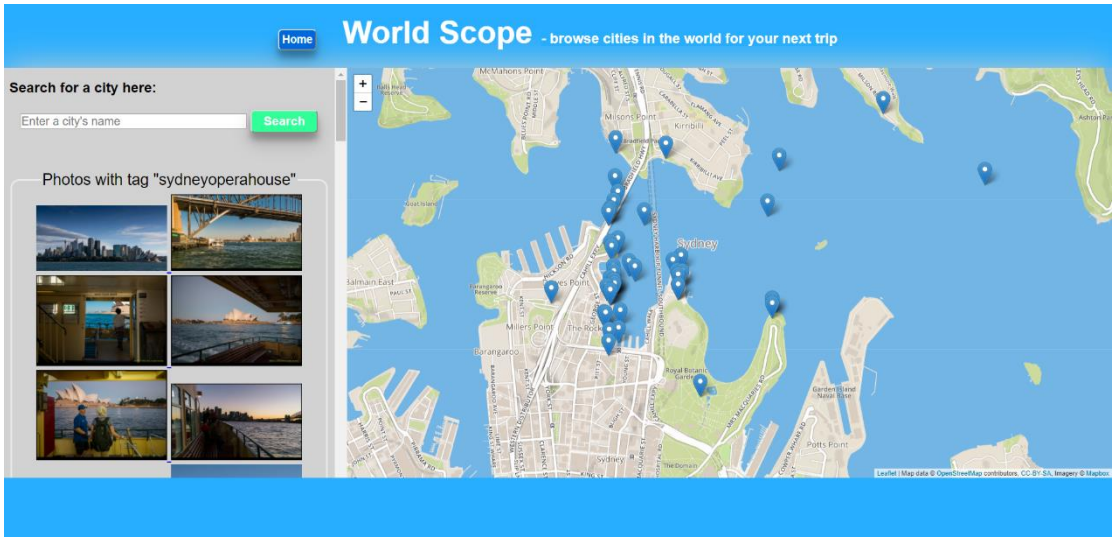
8.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photo in Sydney, Australia
Tags of the photo provided by the owner:
australia
nsw
sydney
sydneyoperahouse
newsouthwales
au
Tags detected by Clarifai:
architecture
modern
city
downtown
finance
Click on one of the tags above to see more photos about a specific tag, or click here to see more
Similar Photos
North Sydney
Sydney

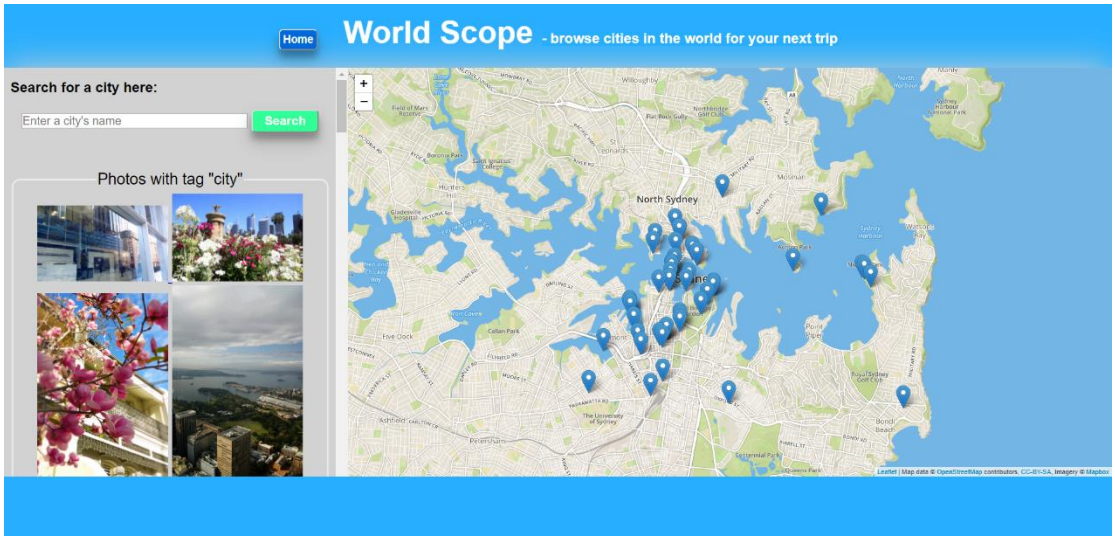
9.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photo in Sydney, Australia
Tags of the photo provided by the owner:
australia
nsw
sydney
sydneyoperahouse
newsouthwales
au
Tags detected by Clarifai:
architecture
modern
city
downtown
finance
Click on one of the tags above to see more photos about a specific tag, or click here to see more
Similar Photos
North Sydney
Lavender Bay
McMahons Point
Sydney

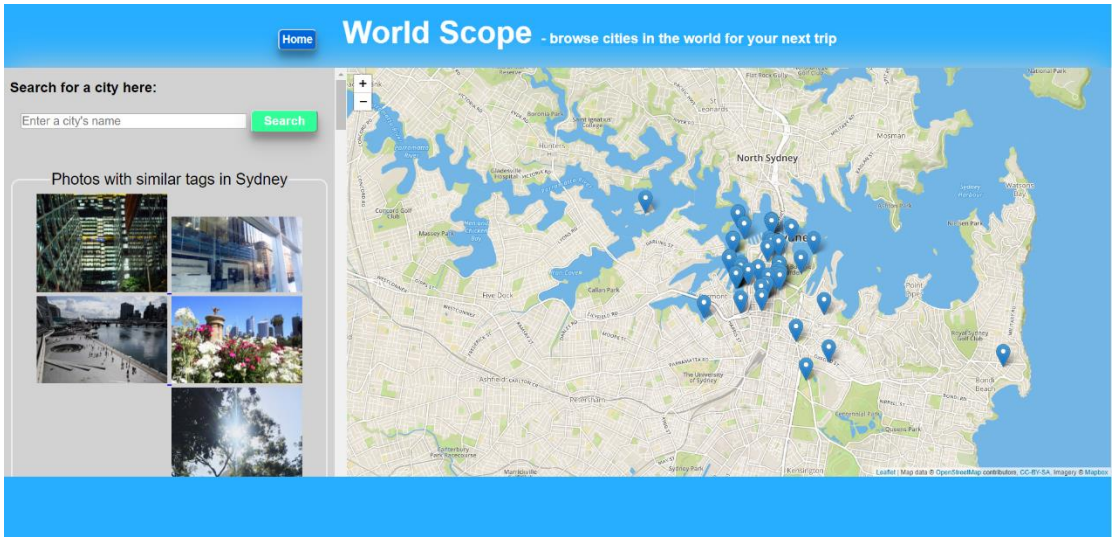
10.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos with tag "sydneyoperahouse"
McMahons Point
Sydney
Millers Point
Potts Point

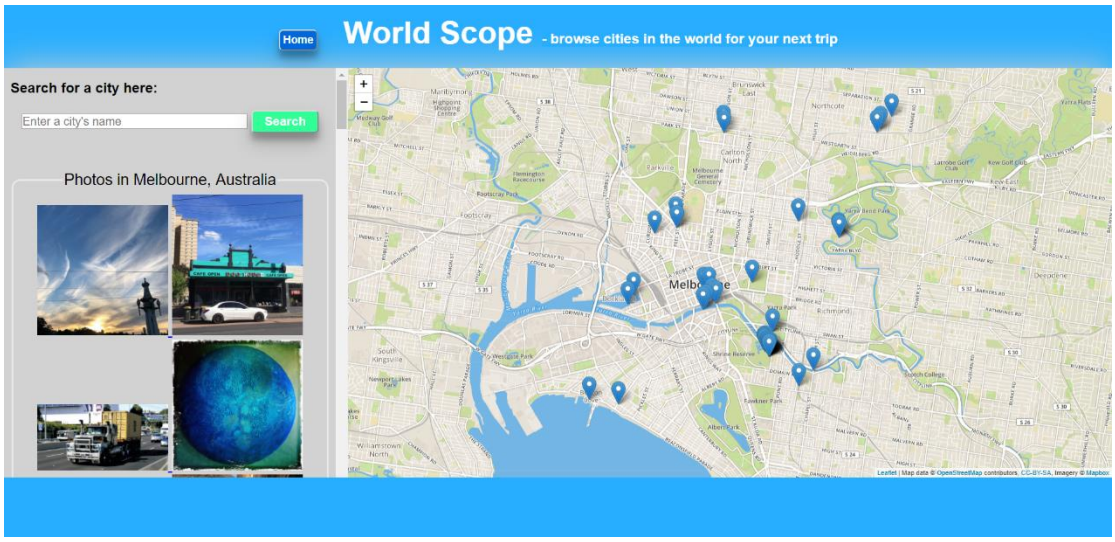
11.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos with tag "city"
North Sydney

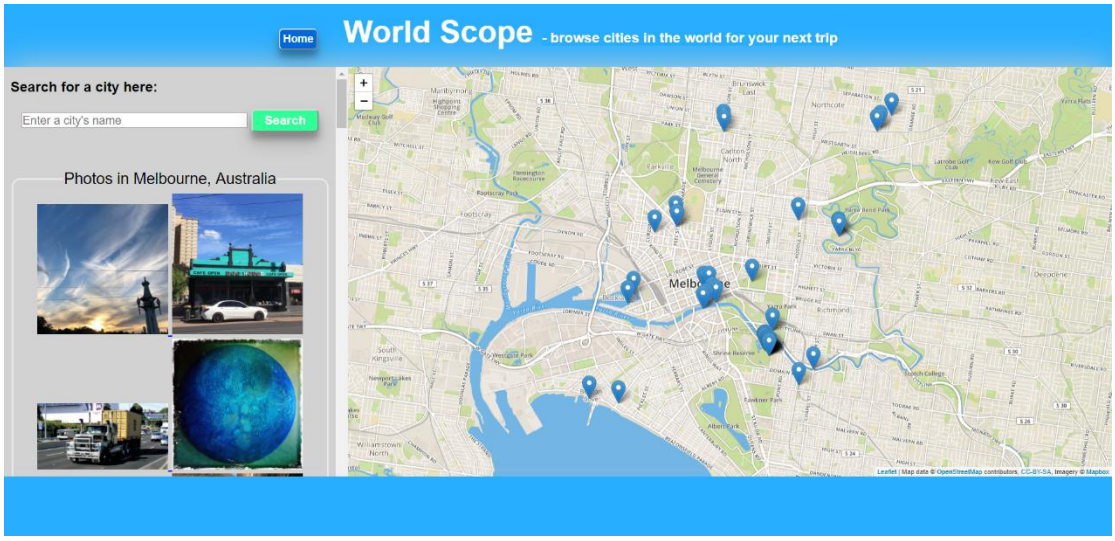
12.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos with similar tags in Sydney
North Sydney

13.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos in Melbourne, Australia

14.


Home
World Scope - browse cities in the world for your next trip
Search for a city here:
Enter a city's name
Search
Photos in Melbourne, Australia

15.


World Scope - browse cities in the world for your next trip
Choose a city from the map or search here:
Enter a city's name
Search

16.


World Scope - browse cities in the world for your next trip
Choose a city from the map or search here:
Enter a city's name
Search
Sorry, Error occurred in the request to Flickr API

17.


World Scope - browse cities in the world for your next trip
Choose a city from the map or search here:
Enter a city's name
Search
Sorry, Error occurred in the request to Clarifai API