2019

CAB432 Assignment 1

MASHUP API

TUAN THANH DIEP – N9607234

Introduction

RestaurantMap helps users to decide on their best restaurant that fits their craves by providing all the necessary information of the restaurant as well as images about the food in an interactive environment. In addition, the application provides the current weather forecast details around the restaurant so that users can be prepared before going to the destination. Using Zomato API, OpenWeather API and Google Map API, users can search for and display restaurant images and details as well as reviews of a specific restaurant.

Upon launching application, users have the options to search for a specific restaurant name, a cuisine, even a specific suburb or they can use your current location to search for nearby restaurant. Once results come in, the map will move and set its centre to the most relevant result, but user can zoom out and choose the restaurant of their choice. Notice that all results as shown as markers on the map, when clicked an information window will opened with all the useful details inside as well as a slider of images of the food by the customers who eaten there. A number of reviews are also included at the bottom of the window.

Zomato API

https://developers.zomato.com/

Zomato provides various API endpoints that can be used to retrieve details of cuisines, restaurants, locations, daily menus, images of the restaurant or food as well as reviews from customers, and their profiles.

RestaurantMap only use one API endpoints from Zomato; */search*, which is used to search for restaurants by the user's query keywords which could be cuisines, locations, restaurant names, etc… This endpoints give a pretty 'dynamic' responses which including, details of restaurants, photos, reviews and reviewer profile details.

OpenWeather API

https://openweathermap.org/api

OpenWeather haves numerous API endpoints which give current, forecast weather along with statistical weather data and even air pollution. There are options to search these by city name, city ID, geographic coordinates or zipcode.

For this application, retrieving the current weather using latitudes and longitudes should satisfy the requirements.

Google Map API

https://developers.google.com/maps/documentation/javascript/tutorial

Google Maps JavaScript API is a popular, well-instructed JS library for creating maps. The library also provides customizable markers and pop-up information windows. For this application, the API is mostly used for visualising data and locations of the restaurant retrieved from Zomato API.

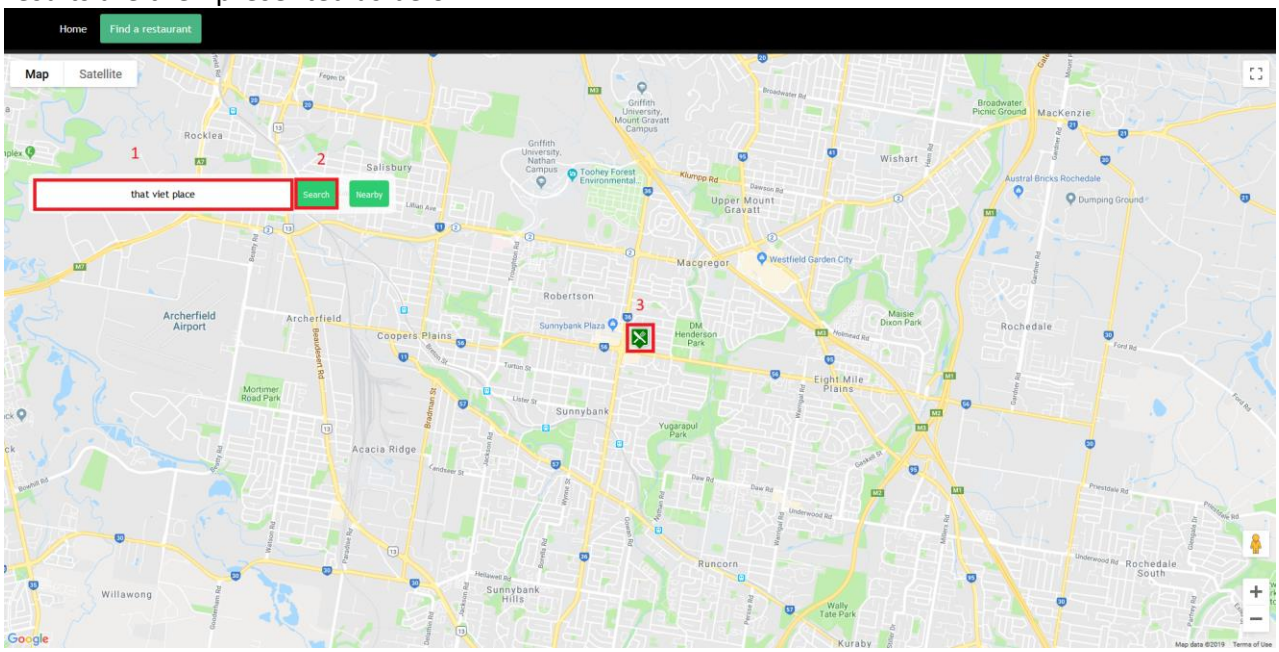
In addition, Geolocaiton is used for finding the user current location for one of the use case. The Google Map JavaScript library is included in the front-end html <script> tabs and can be obtained by CDN, the map is rendered and displayed once the browser is fully loaded.

Use Cases

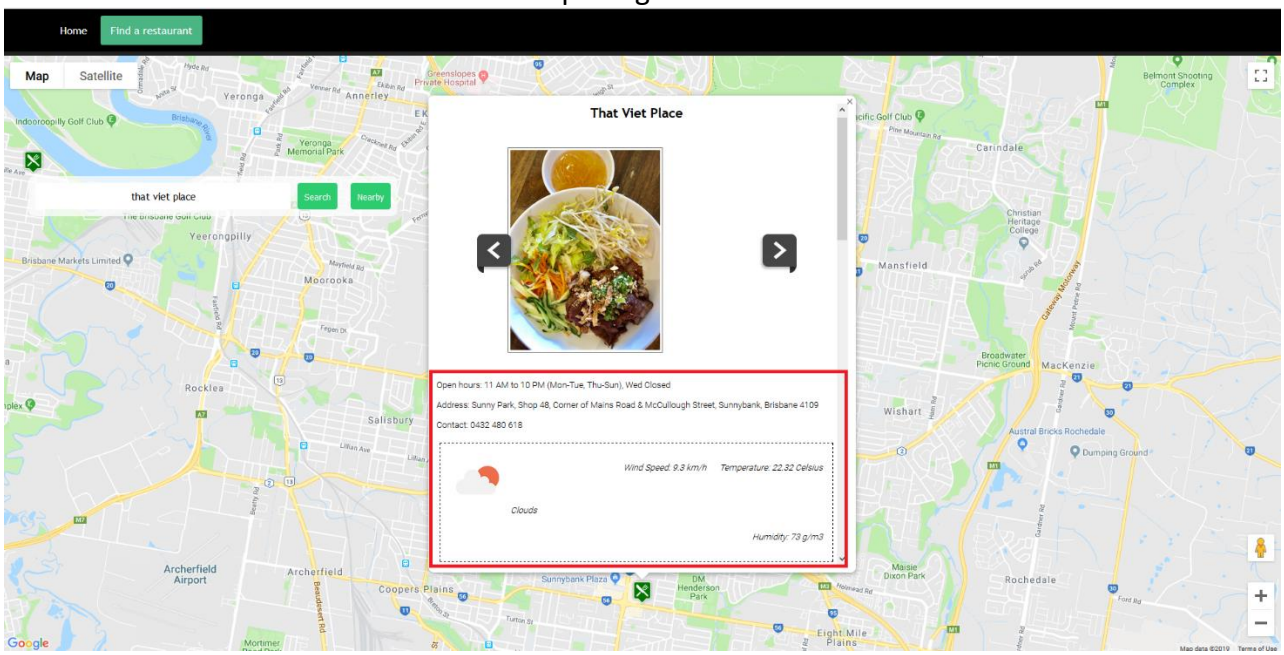
Use case 1

As a user, I want to search a specific restaurant by name, so that I can see what's their opening time is and the current weather at the location.

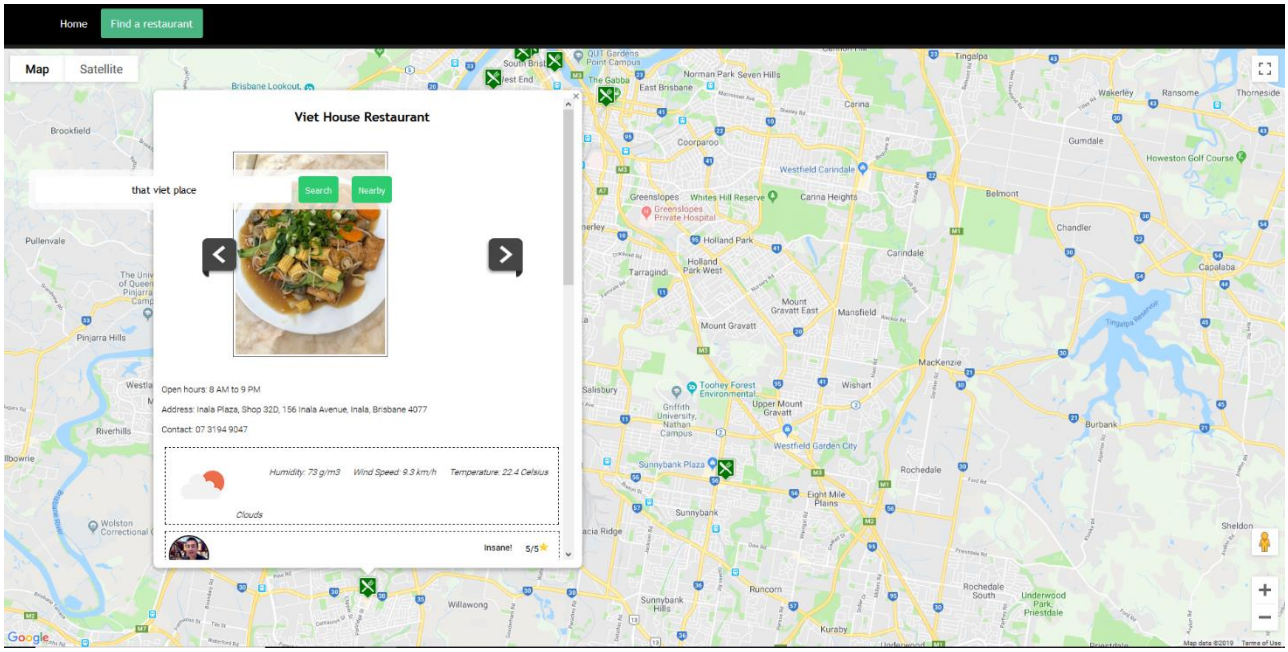
This user launches the search page of RestaurantMap, wanting to search for a restaurant named "That Viet Place" as they typed the name into the search and click *Search*. The results are then presented as below.



The map will zoom and centre at the first result that it finds. The user is then clicks on it and reads the weather forecast as well as the opening time of the restaurant.



After that, the user can zoom out and check out the other similar restaurants.

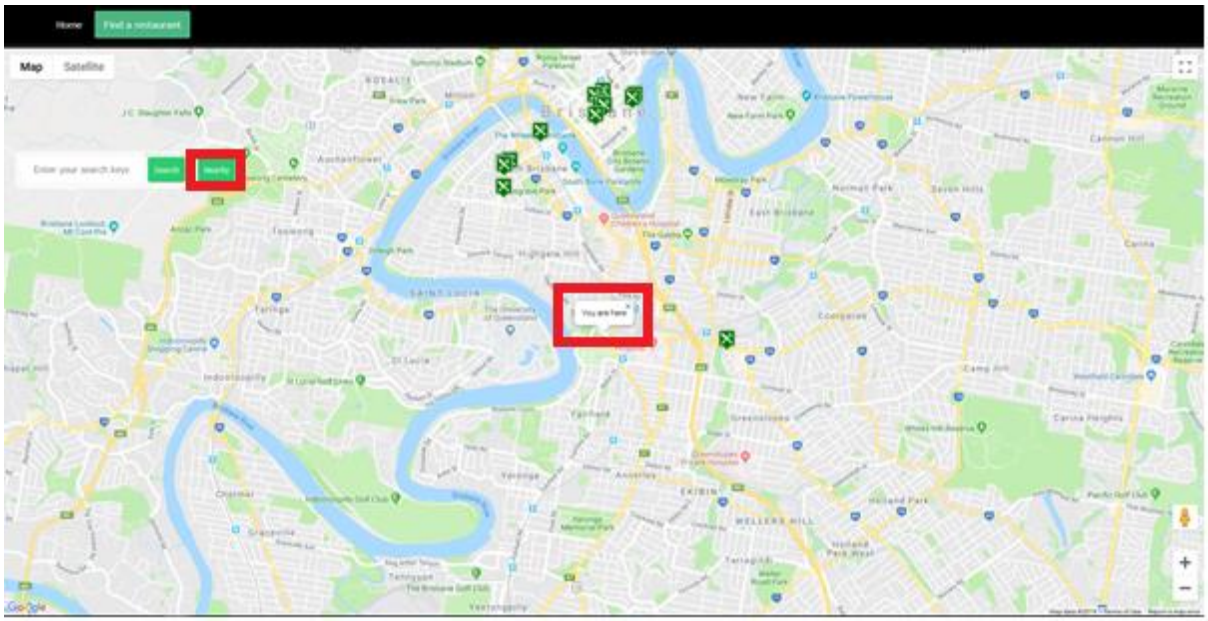


Usage of services: Zomato API, OpenWeather.

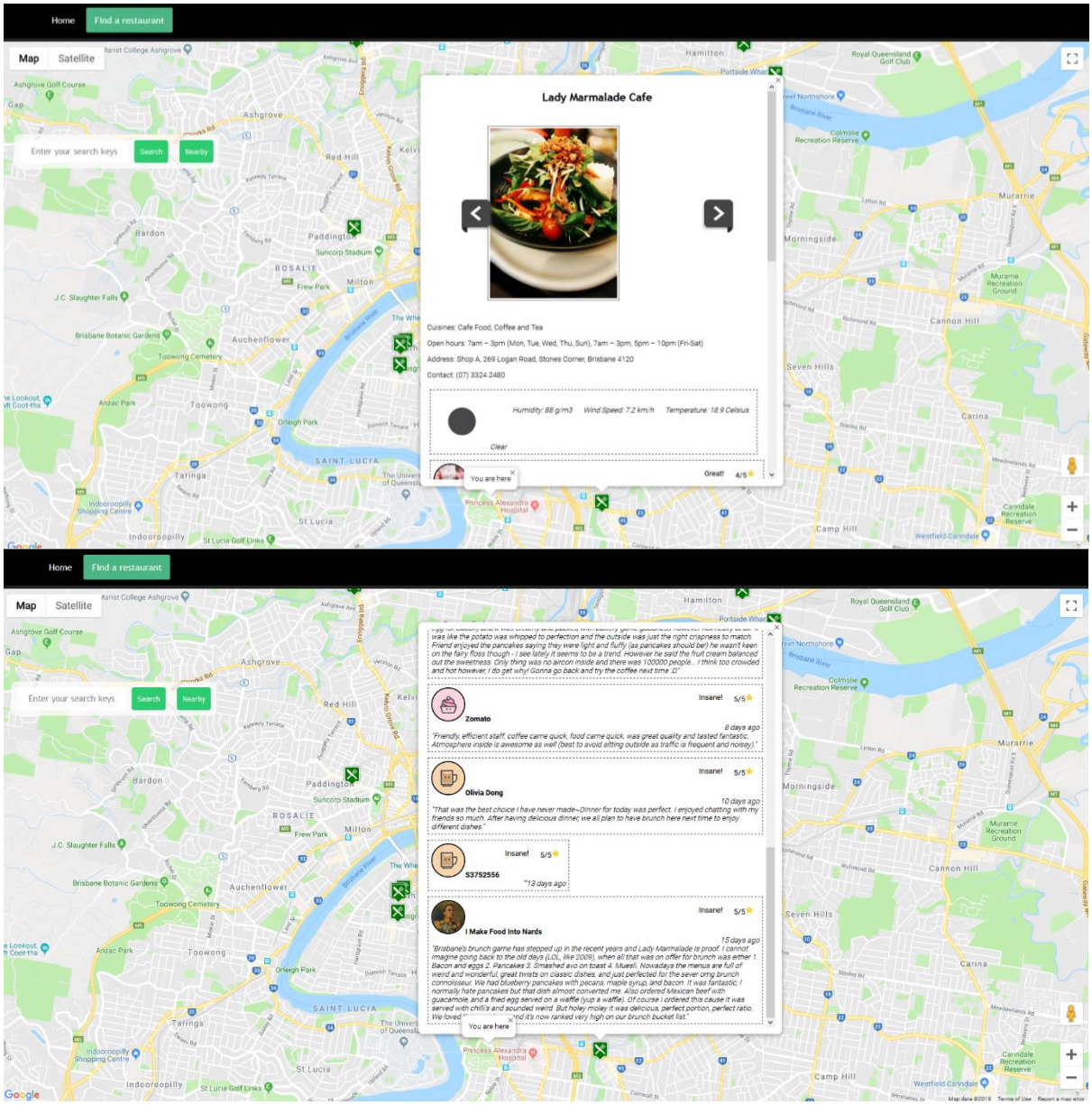
Use case 2

As a user, I want to search a restaurant that is near my location, so that I can find a decent restaurant that is open and have good ratings.

This user navigates to the search page of RestaurantMap and click the *Nearby* button in the search box. A popup box is then appear asking for permission to share your location for the browser, the user click *Allow* and is then presented with the results below.



The user is then can click around and choose whichever restaurant best to have dinner at by looking at the reviews.



Usage of services: Zomato API, Geolocation from Google Map API, OpenWeather.

Technical Description

As required in the assessment descriptions, most of the processing should be done in the server-side of the application. The application was developed with this idea in mind and all communication and responses details received from the APIs are filtered and optimised as much as possible in order to send sufficient results to the front-end with the smallest payload.

A basic flow chart of the data transfer process is shown below.

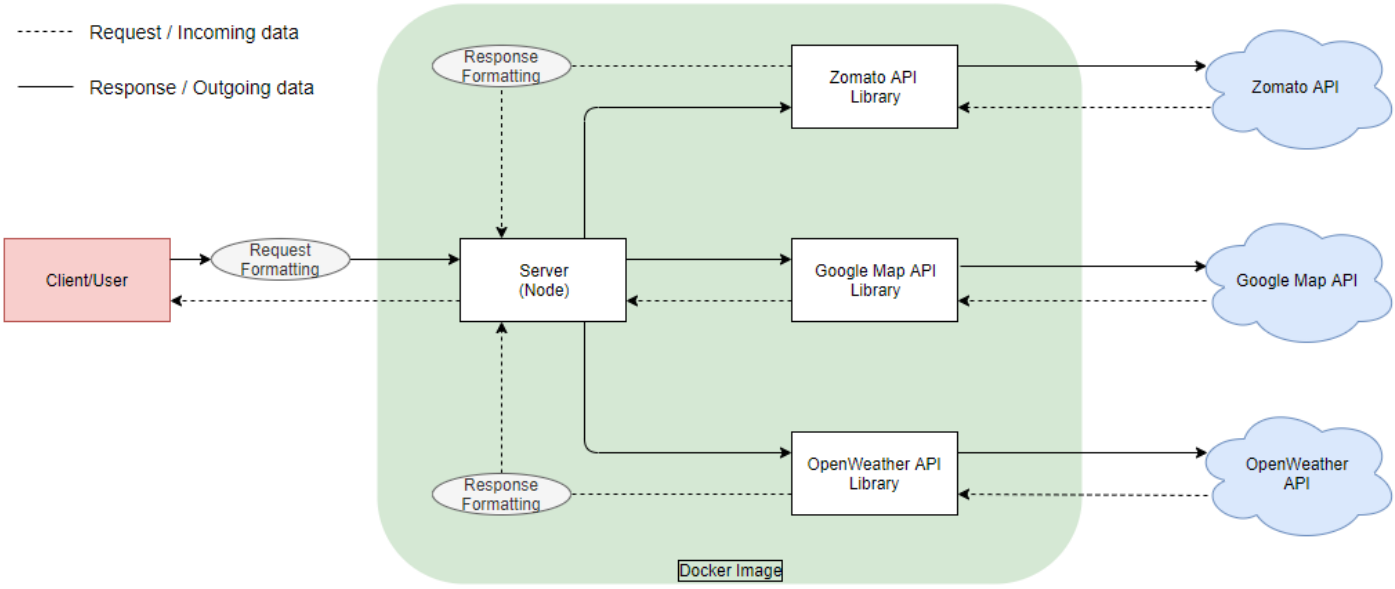


Figure 1 Flow chart of the back end process

Client

Upon the initial launch of the application, the client will make a request to the server, where the page is then served. After that, any interaction by the user to the client such as searching for restaurant, clicking a marker, viewing images. These request will be sent to the back-end using JQuery AJAX GETs.

After receiving the process payload from the server, the client-side then display it on Google Map.

Server

Upon receiving GET requests from the front-end, the server process match the relevant request and make GET requests to external APIs using middleware Axios. Once received the responses, the server filters and process the results then send back to the client-side, where the results will be displayed. Routes are created to handle appropriate and sufficient code to manage and process incoming AJAX requests and outgoing GET request to APIs.

Some middleware are used for the server-side including Axios, helmet and morgan for logging.

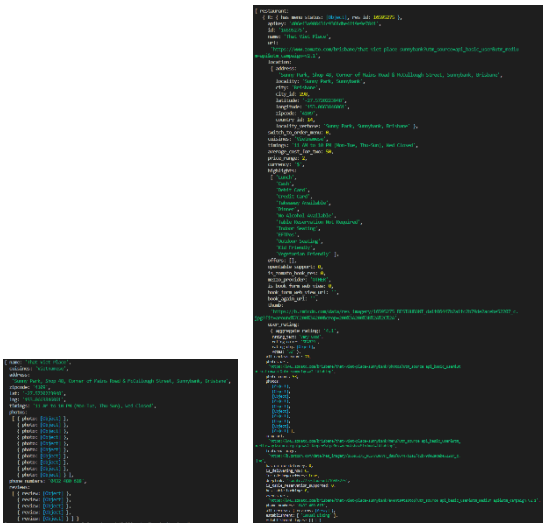
```
[Mon, 16 Sep 2019 07:35:54 GMT] "GET /" 200 0.267
[Mon, 16 Sep 2019 07:35:55 GMT] "GET /search" 200 0.354
[Mon, 16 Sep 2019 07:36:00 GMT] "GET /search/full?lat=-32.2602&lon=148.017" 200 1065.000
```

Figure 2 Logging format for Server-side

Results Filtering

The Zomato and OpenWeather API responses are not ready-to-use material and quite lengthy in term of depths. Besides removing the irrelevant information from the responses, the server also restructure the JSON responses, making it simpler and shorter in depths. This will make reading the information in the client-side a bit easier.

Figure shown below is a comparison between the raw responses from the APIs to the server versus the processed response from the server to the client-side.



Filtered | Raw
Zomato API

Filtered | Raw
OpenWeather API

Docker

As specified, Docker has to be used to contain RestaurantMap application as well as the Node server and the client html templates. The Dockerfile used to build the application image is shown below.

```
FROM node:boron

# Copy app source
COPY . /restaurantmap

# Set work directory to /src
WORKDIR /restaurantmap

# Install app dependencies
RUN npm install

# Expose port to outside world
EXPOSE 3000

# start command as per package.json
CMD ["npm", "start"]
```

The build uses the official Docker node image latest LTS version *boron*. The "*COPY . /restaurantmap"*copy the source code of RestaurantMap into the base directory of the image, and expose port 3000. The *RUN* command used to install the application dependencies included in the package JSON. The last command is for the Dockerfile to uses CMD "npm start" to start the application.

Testing and limitations

Error Handling

For the client-side, a few regulation has been created so that the user will not accidentally search with an empty query or appear/interact with user request when not allow to share their current location but want to search for nearby restaurant. Snippet A and B shows how those cases are handled in code.

Snippet A

```
if (content) {
    getResults(content, null, null, function(data) {
        // Clear old markers before put in new ones if any
        if (markers) deleteMarkers();
        displayResults(data);
        map.setCenter({ lat: parseFloat(data[0].lat), lng: parseFloat(data[0].lng) });
        map.setZoom(14);
    });
} else {
    alert("Please type in your search keys!");
}
```

Snippet B

```
if (navigator.geolocation){
    // Get user location and send GET request
    navigator.geolocation.getCurrentPosition(function(position){
        // Send GET request
    }, function(){
        // When user decline to share their location
        handleLocaitonError();
    });
} else {
    // For Broswer doesn't support Geolocation
    handleLocaitonError();
}
```

For the server-side, there is not much error response from both Zomato and OpenWeather API and since the client already has a few regulations for the unexpected error to happen less; thus, error handling has been taken slightly light for the server. Snippet C shows how the case are handled in code.

Snippet C

```
.catch((error) => {
    if (error.message) {
        // Request made and server responded
        console.log(error.response.data);
        console.log(error.response.status);
        console.log(error.response.headers);
    } else if (error.request) {
        // The request was made but no response received
        console.log("No repsonse for: " + error.request);
        res.sendStatus(204).json({ message: "No response from OpenWeather API" });
    } else {
        //  Something happenned in setting up the request that trigged an Error
        console.log('Error', error.message);
        res.sendStatus(400).json({ message: error.message });
    }
})
```

Test Cases

Task	Expected Outcome	Result	Screenshot/s (Appendix B)
Search for restaurants by name	Displays results on map	PASS	01
Empty search keys query	Alert box popup ask to fill keywords	PASS	02
Search for nearby restaurants (allow location)	Displays results on map	PASS	03
Search for nearby restaurants (not allow location)	Alert box pop up ask to allow to proceed	PASS	04
Click on makers	InfoWindow pop up with relevant details	PASS	05
Click on next button of the photo slider in InfoWindow	Next photo appear, current photo disappear	PASS	06
Click on previous button of the photo slider in InfoWindow	Previous photo appear, current photo disappear	PASS	07
Click on the profile picture of the reviewer	Redirect to the reviewer Zomato profile	PASS	08

Difficulties

Second Use Case

OpenWeather API is a last-minute decision to add in for the second use as it seems hard to find a second use case that is totally different to the first one.

CSS Styling

The CSS styling of the application is not perfect, but still it took a decent amount of time to figure out how to make the UI looks decent, yet simple; especially for the popup infoWindow. However, this could be improved be improved using Bootstrap or some other CSS Library.

Possible Extension

Second Use Case

The current second use case is different from the first one, but still have a few similar features with the first use case. This can be improved with a more sophisticate use case by adding tags for cuisines with each restaurant. For example, “That Viet Place” would have a Vietnamese cuisine tags. Clicking on the tags would search for other restaurants that are also Vietnamese. This would add more depth into the use case.

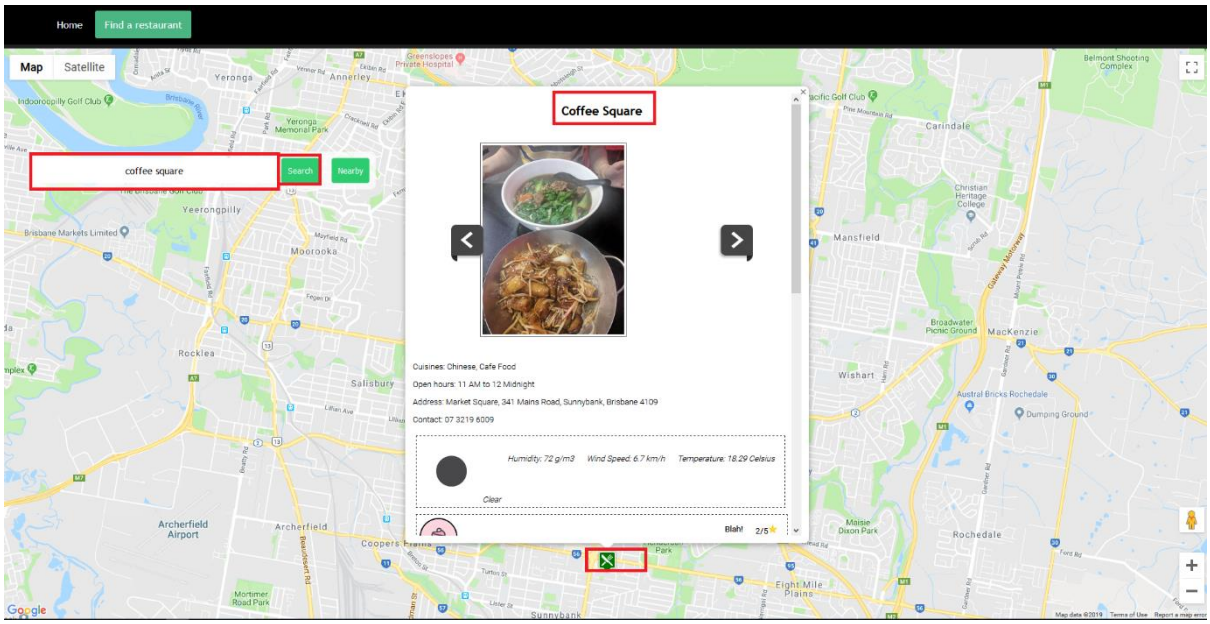
Search Bar Configuration

As it clear as it is, the current search bar does not have a parameter for the search query such as radius of the search, amount of results to display, cuisines, sorting, etc… This can be implemented by adding a few more dropdown list with some parameter in the search query.

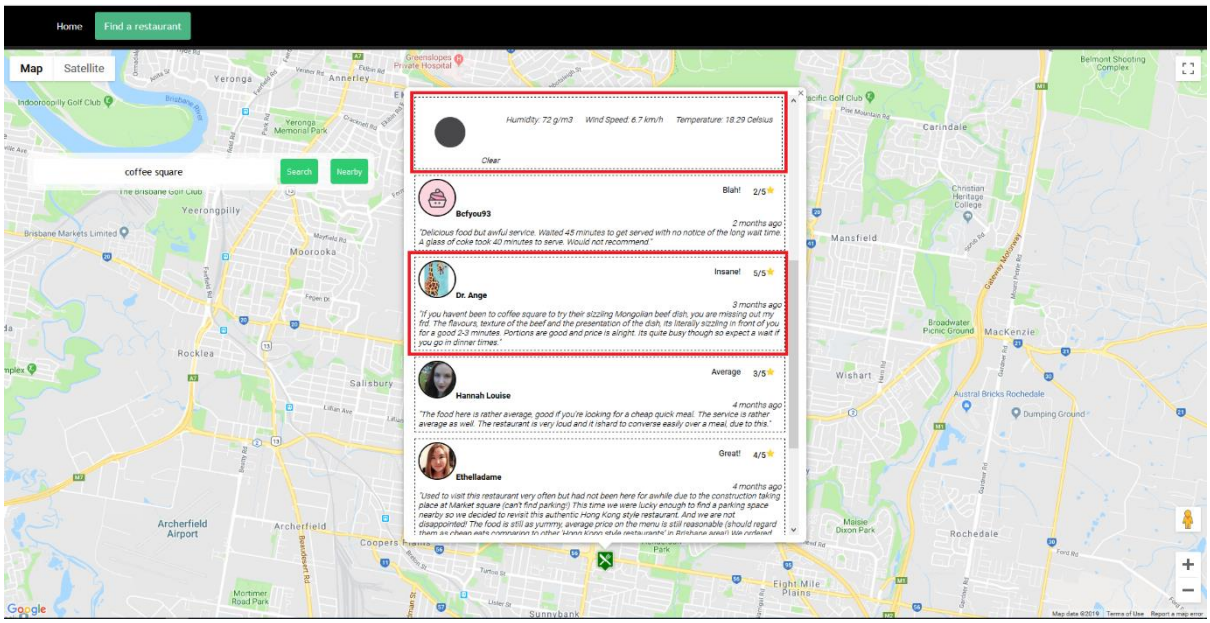
Appendix A: Brief user guide

When launching the index page, the user can see a short description of the application as well as which APIs were used. The user can click the button *Find* or *Find restaurant* in the navigation bar.

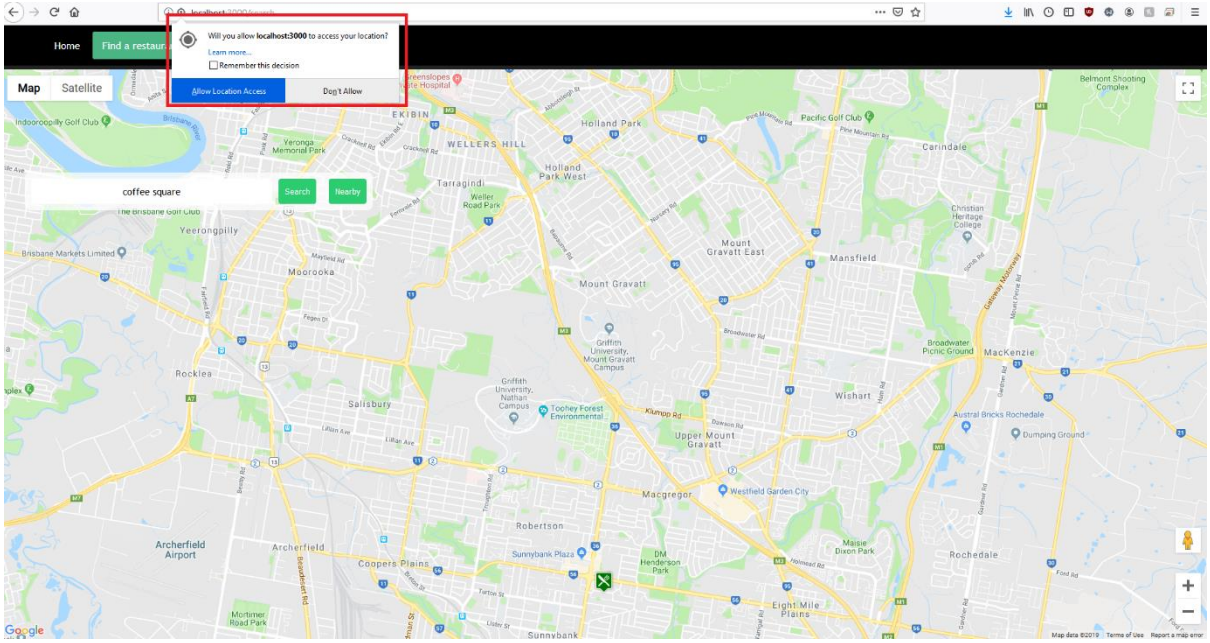
User is then redirect to a Google Map with a search bar and two buttons. The user can enter the name of the restaurant that they want to search for. For this case, its “Coffee Square”. The results will be shown as below.



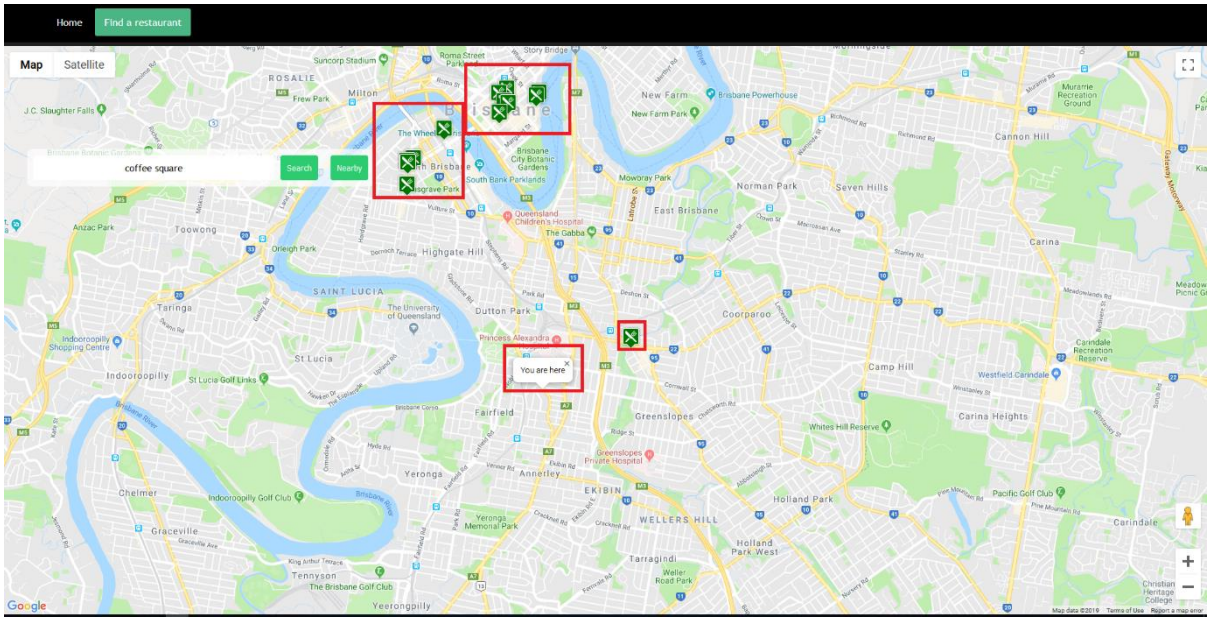
They can now scroll down the InfoWindow to check the current weather at the location as well as reading all most recent reviews of that restaurant.



Now if the user feels a bit venturous, they can click *Nearby* button to find out the location near the location they are at. A prompt will popup and ask for their permission to use their location for the search. Note that the user **have to** click *Allow* in order to proceed.



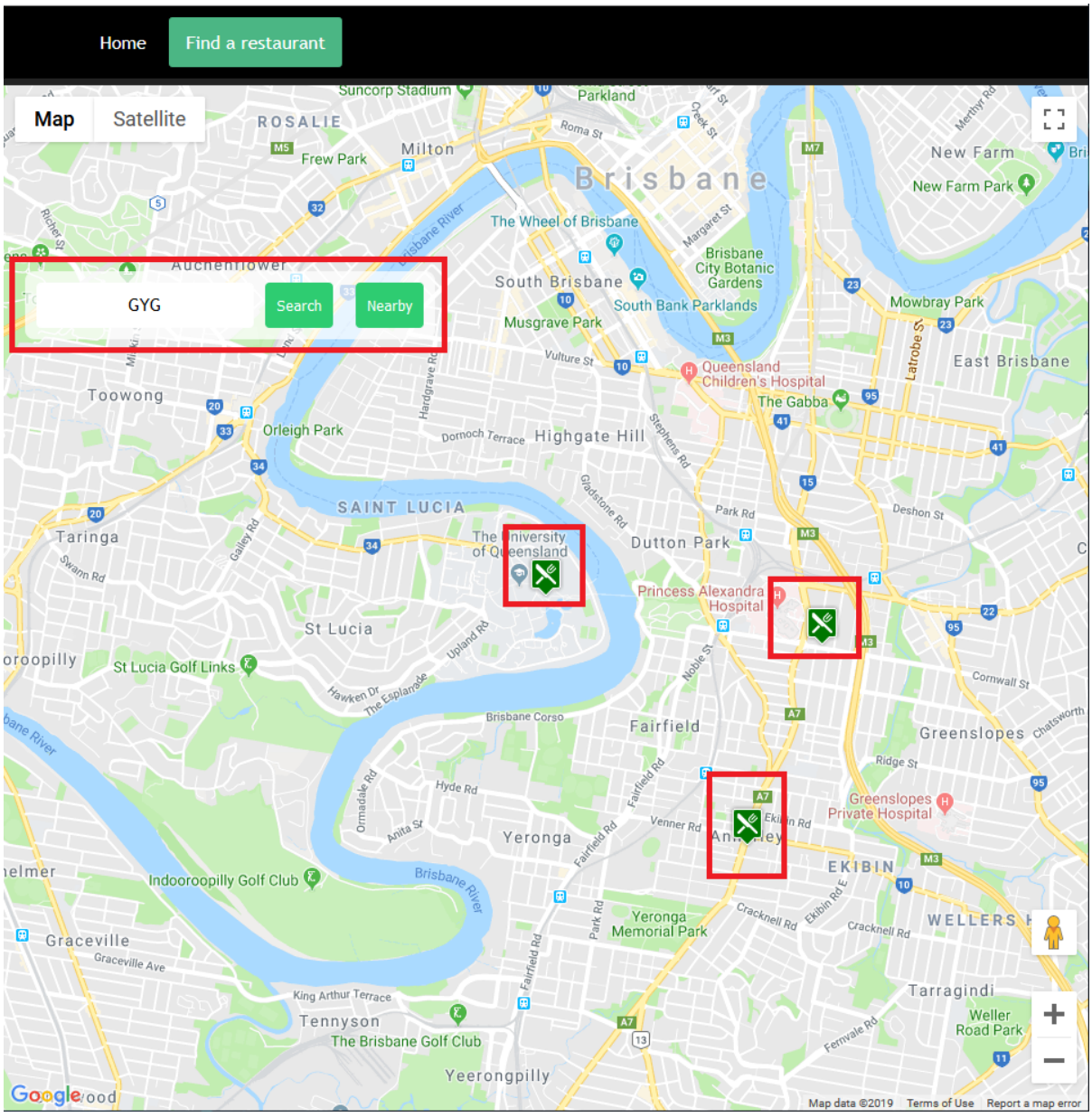
Once clicked *Allow,* the request is sent and might take a few seconds for it to come with results as shown below.



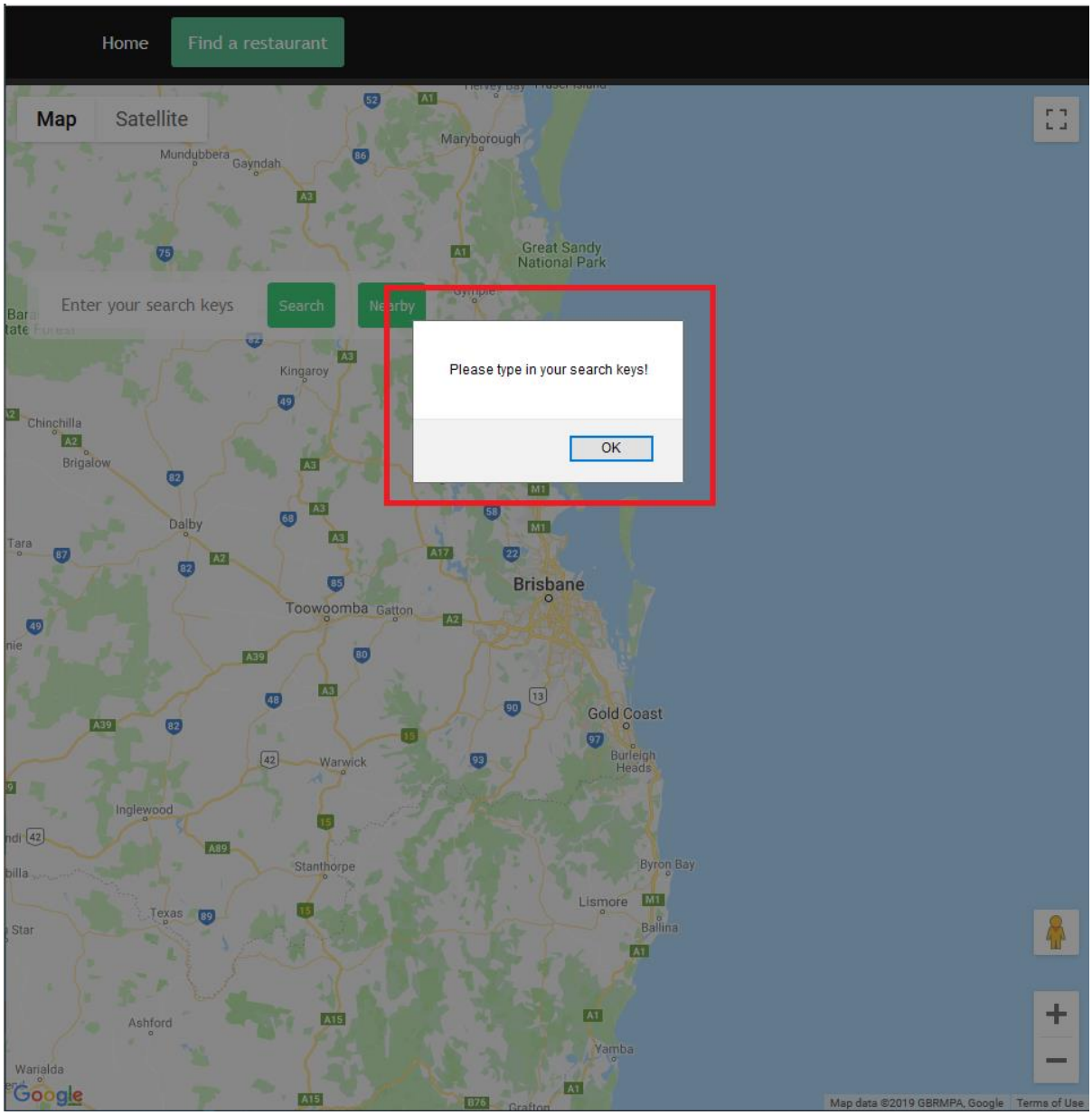
Once all the results shown, the user can continue check new places without moving an inch and might able to find a hidden gem!

Appendix B: Test Cases Screenshots

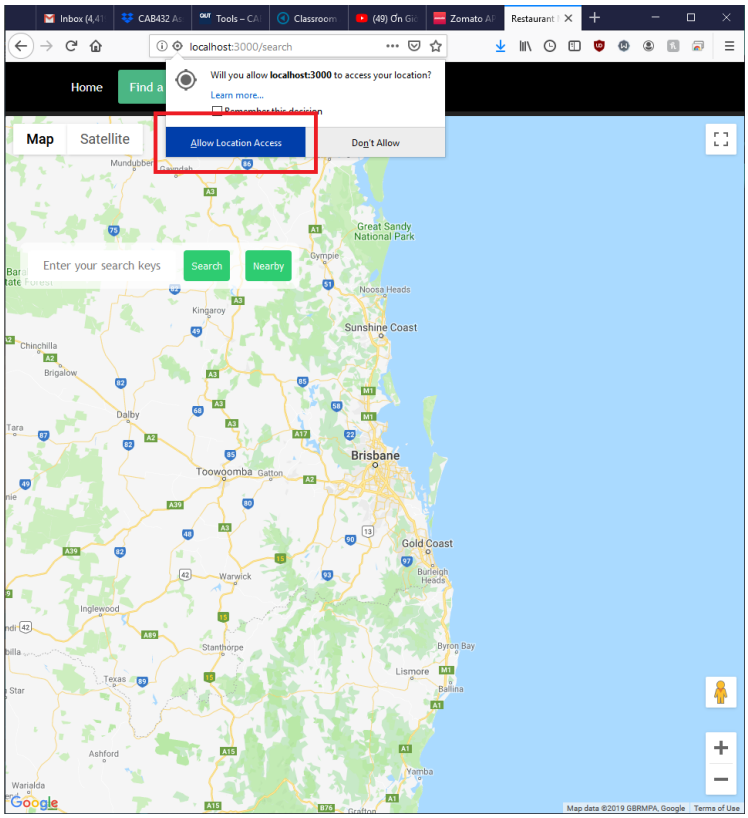
Case 01:

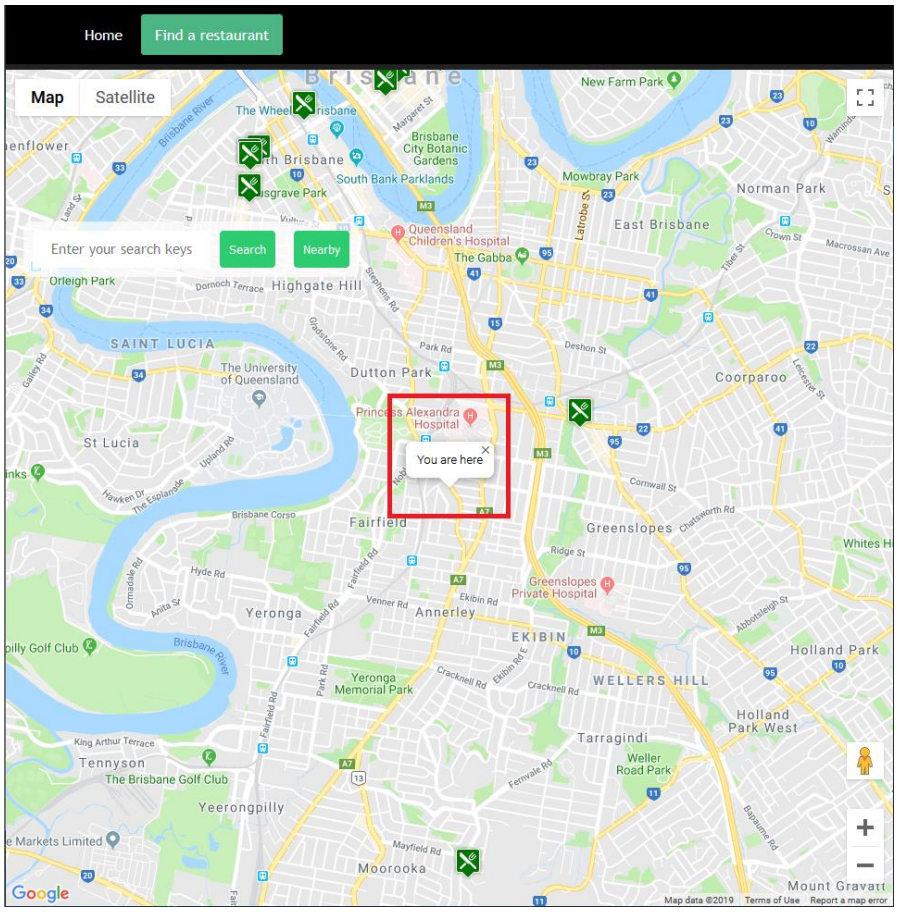


Case 02:

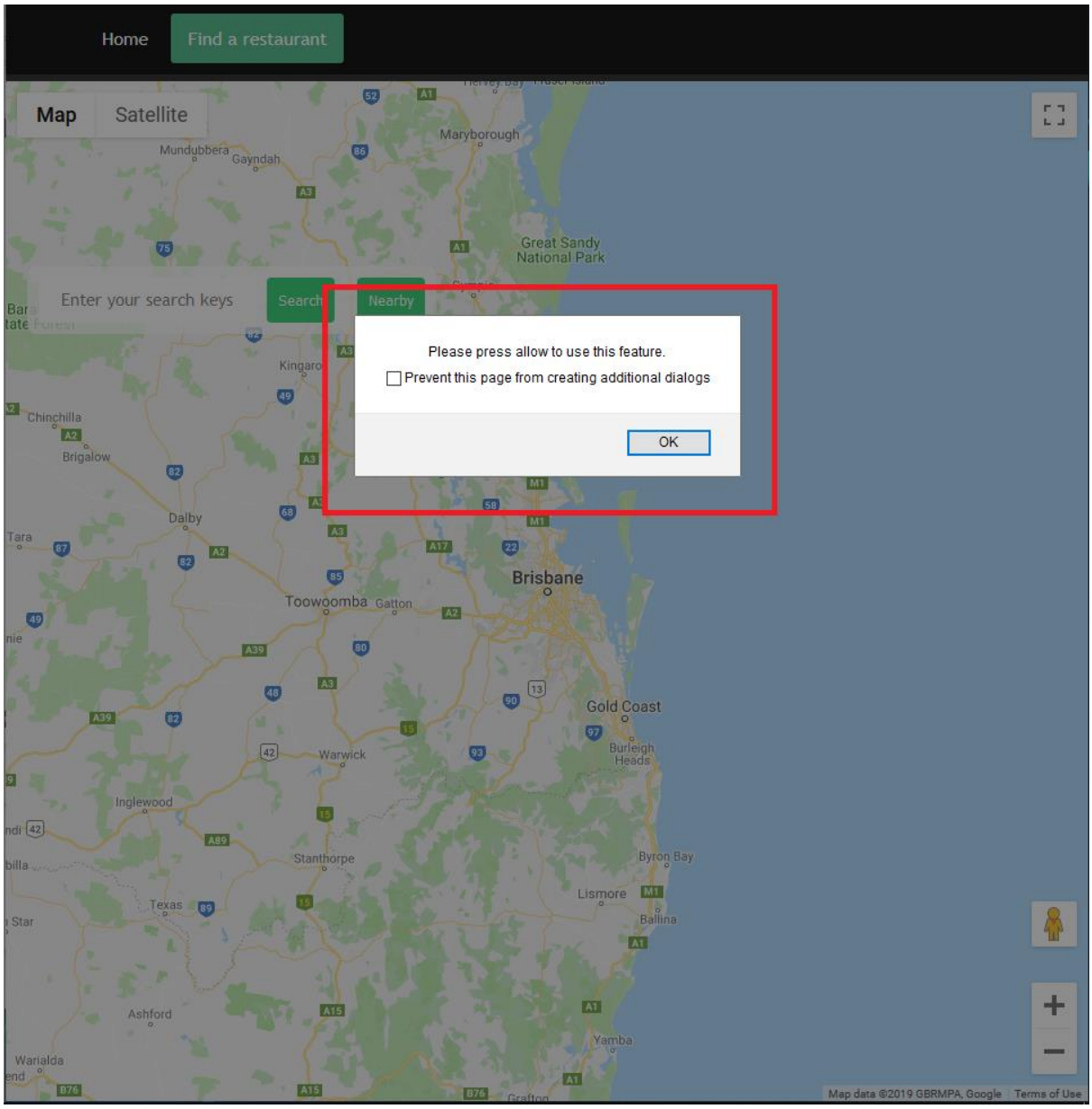

Home
Find a restaurant
Map
Satellite
Enter your search keys
Search
Please type in your search keys!
OK
Brisbane
Gold Coast
Toowoomba
Warwick
Dalby
Kingaroy
Maryborough
Great Sandy National Park
Lismore
Byron Bay
Ballina
Stanthorpe
Inglewood
Texas
Ashford
Yamba
Google
Map data ©2019 GBRMPA, Google
Terms of Use

Case 03:

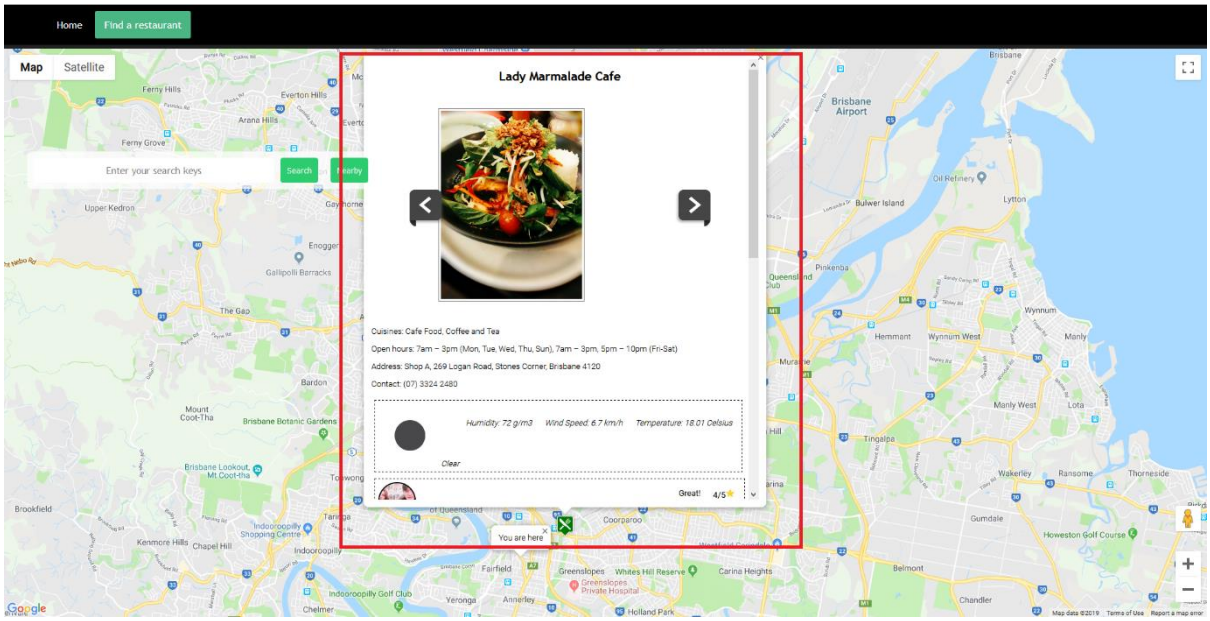

localhost:3000/search
Will you allow localhost:3000 to access your location?
Learn more...
Allow Location Access
Don't Allow
Home
Find a
Map
Satellite
Enter your search keys
Search
Nearby
Brisbane
Gold Coast
Sunshine Coast
Toowoomba
Great Sandy National Park
Google


Home
Find a restaurant
Map
Satellite
Enter your search keys
Search
Nearby
You are here
Brisbane
New Farm Park
Brisbane City Botanic Gardens
South Bank Parklands
Queensland Children's Hospital
The Gabba
Highgate Hill
Dutton Park
The University of Queensland
Princess Alexandra Hospital
Fairfield
Yeronga
Annerley
Greenslopes Private Hospital
EKIBIN
WELLERS HILL
Holland Park
Tarragindi
Holland Park West
Yeronga Memorial Park
Tennyson
The Brisbane Golf Club
Yeerongpilly
Moorooka
Mount Gravatt
Google

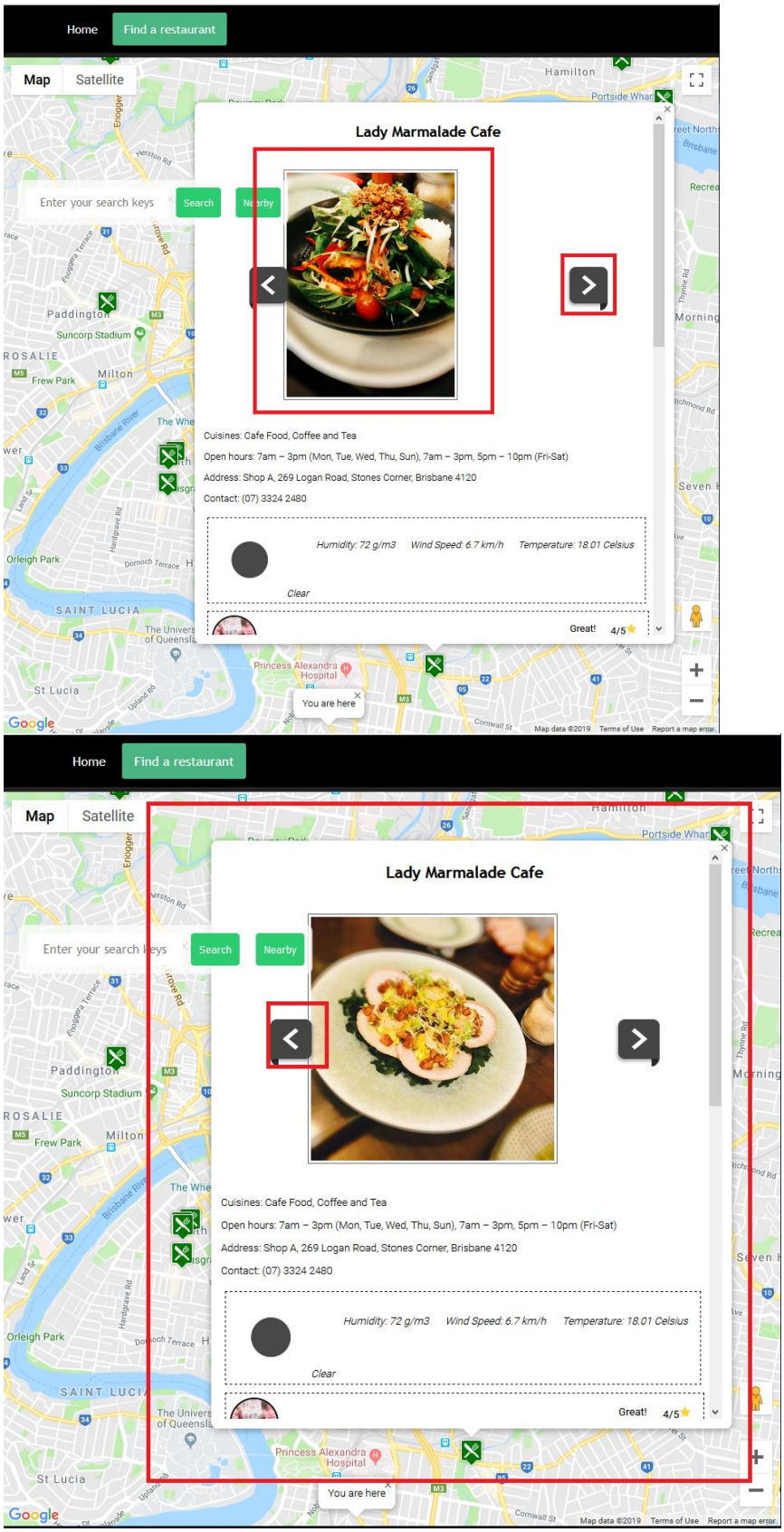
Case 04:


Home
Find a restaurant
Map
Satellite
Enter your search keys
Search
Nearby
Please press allow to use this feature.
Prevent this page from creating additional dialogs
OK
Brisbane
Gold Coast
Toowoomba
Great Sandy National Park
Map data ©2019 GBRMPA, Google
Terms of Use

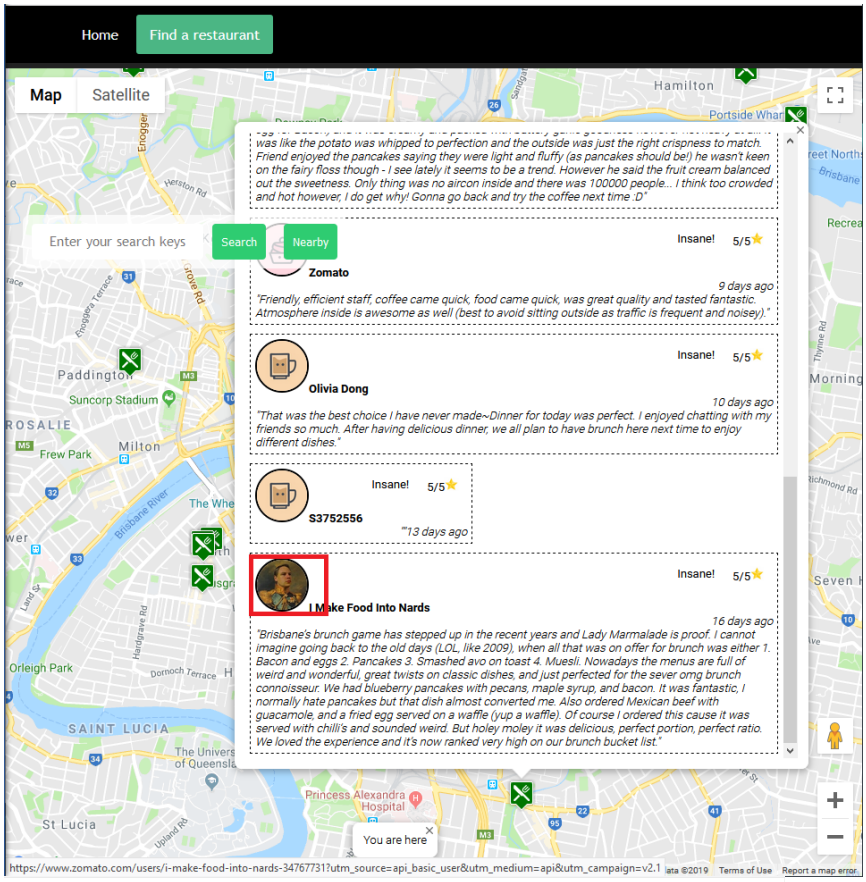
Case 05:

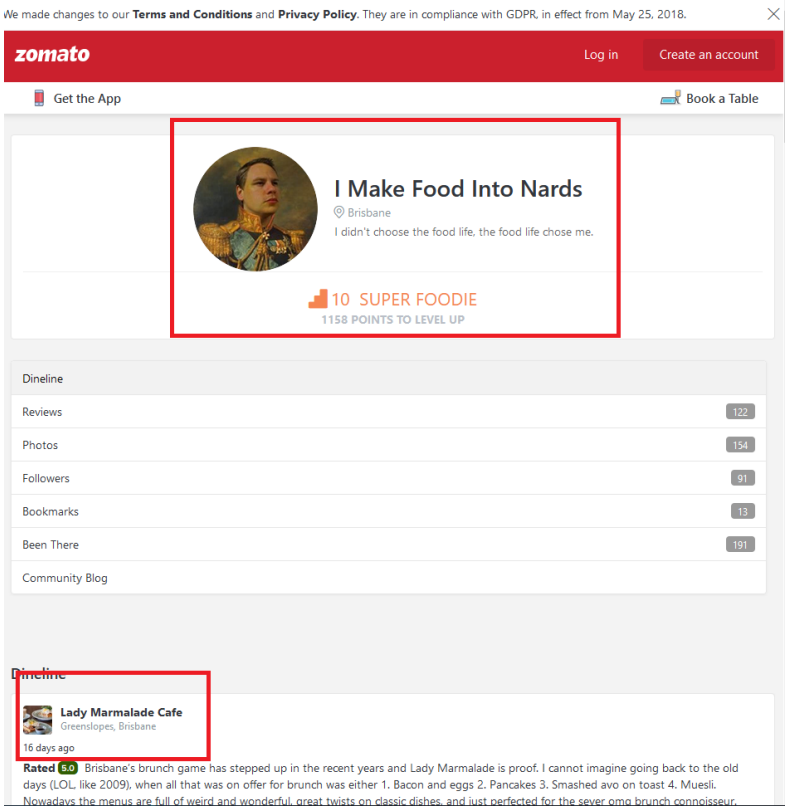

Home
Find a restaurant
Map
Satellite
Enter your search keys
Search
Nearby
Lady Marmalade Cafe
Cuisines: Cafe Food, Coffee and Tea
Open hours: 7am – 3pm (Mon, Tue, Wed, Thu, Sun), 7am – 8pm, 5pm – 10pm (Fri-Sat)
Address: Shop A, 269 Logan Road, Stones Corner, Brisbane 4120
Contact: (07) 3324 2480
Humidity: 72 g/m3
Wind Speed: 6.7 km/h
Temperature: 18.01 Celsius
Clear
Great!
4/5
You are here
Google

Case 06 & 07


Home
Find a restaurant
Map
Satellite
Lady Marmalade Cafe
Enter your search keys
Search
Nearby
Cuisines: Cafe Food, Coffee and Tea
Open hours: 7am – 3pm (Mon, Tue, Wed, Thu, Sun), 7am – 3pm, 5pm – 10pm (Fri-Sat)
Address: Shop A, 269 Logan Road, Stones Corner, Brisbane 4120
Contact: (07) 3324 2480
Humidity: 72 g/m3
Wind Speed: 6.7 km/h
Temperature: 18.01 Celsius
Clear
Great!
4/5
You are here
Google

Case 08:


Home
Find a restaurant
Map
Satellite
Enter your search keys
Search
Nearby
Insane! 5/5
Zomato
9 days ago
"Friendly, efficient staff, coffee came quick, food came quick, was great quality and tasted fantastic. Atmosphere inside is awesome as well (best to avoid sitting outside as traffic is frequent and noisey)."
Insane! 5/5
Olivia Dong
10 days ago
"That was the best choice I have never made~Dinner for today was perfect. I enjoyed chatting with my friends so much. After having delicious dinner, we all plan to have brunch here next time to enjoy different dishes."
Insane! 5/5
S3752556
"13 days ago
Insane! 5/5
I Make Food Into Nards
16 days ago
"Brisbane's brunch game has stepped up in the recent years and Lady Marmalade is proof. I cannot imagine going back to the old days (LOL, like 2009), when all that was on offer for brunch was either 1. Bacon and eggs 2. Pancakes 3. Smashed avo on toast 4. Muesli. Nowadays the menus are full of weird and wonderful, great twists on classic dishes, and just perfected for the sever omg brunch connoisseur. We had blueberry pancakes with pecans, maple syrup, and bacon. It was fantastic, I normally hate pancakes but that dish almost converted me. Also ordered Mexican beef with guacamole, and a fried egg served on a waffle (yup a waffle). Of course I ordered this cause it was served with chilli's and sounded weird. But holey moley it was delicious, perfect portion, perfect ratio. We loved the experience and it's now ranked very high on our brunch bucket list."
Hamilton
Paddington
Suncorp Stadium
Milton
Orleigh Park
SAINT LUCIA
St Lucia
Princess Alexandra Hospital
You are here
https://www.zomato.com/users/i-make-food-into-nards-34767731?utm_source=api_basic_user&utm_medium=api&utm_campaign=v2.1
Terms of Use
Report a map error


We made changes to our Terms and Conditions and Privacy Policy. They are in compliance with GDPR, in effect from May 25, 2018.
zomato
Log in
Create an account
Get the App
Book a Table
I Make Food Into Nards
Brisbane
I didn't choose the food life, the food life chose me.
10 SUPER FOODIE
1158 POINTS TO LEVEL UP
Dineline
Reviews 122
Photos 154
Followers 91
Bookmarks 13
Been There 191
Community Blog
Dineline
Lady Marmalade Cafe
Greenslopes, Brisbane
16 days ago
Rated 5.0 Brisbane's brunch game has stepped up in the recent years and Lady Marmalade is proof. I cannot imagine going back to the old days (LOL, like 2009), when all that was on offer for brunch was either 1. Bacon and eggs 2. Pancakes 3. Smashed avo on toast 4. Muesli. Nowadays the menus are full of weird and wonderful, great twists on classic dishes, and just perfected for the sever omg brunch connoisseur.