



2017

Discover Restaurants

Discover food and restaurants from all around the world

CAB432 Assignment 1 Report
YeeChen Fong - 09602976

Table of Content

Introduction	2
Mashup Use Cases & Dependent Services	3
Use Case 1	3
Use Case 2	7
Use Case 3	9
Technical Description	11
Server-side	11
Client-side	12
Difficulties	13
Docker	14
Testing & Limitation	15
Test Plan & Results	15
Limitations	16
Possible Extensions	16
Reference	17
Appendix	18
Brief User Guide	18
Index Page	18
Result Page	21
Info Page	25
Dockerfile	31

Introduction

Discover Restaurants aims to provide a resource for users to find cuisine and restaurants from all over the world. The service provides a general information of the restaurant entity the user has lookup for based on the the city, type of cuisine, restaurant category and restaurant establishment. This includes the restaurant name itself, the location, the type of cuisine, the average spending, ratings and much more.

The service allows the user to locate the restaurant on Google Maps, with the options to show the direction to the restaurant if the user if currently located in the same country. Moreover, the user can look up at the weather conditions of the area around the restaurant.

Zomato Restaurant API is used to make restaurant searches and display general information. In the home page, the API is called to display list of city names where there are restaurants listed in Zomato API. The API is also used to display the restaurant category, cuisine, and restaurant establishment available in a city. Once the restaurant category, cuisine, and restaurant establishment options have been selected, the API query a search and display up till a maximum of 100 restaurants in the city based on the selected categories. The Zomato Restaurant API url: <https://developers.zomato.com/api>.

OpenWeatherMap API is used to query the weather conditions based on the geo-coordinates of the selected city. The API will return current weather information which includes weather description, temperature, humidity, wind speed, and the location of the weather queried. The OpenWeatherMap API url: <https://openweathermap.org/api>.

Google Maps Javascript API is used to display the restaurant locations. The API gets the location of the restaurants in Latitude and Longitude, and plot them in as tags in the map. The API also provides the option to get the current location of the browser. The API can display the direction between the user and the restaurant chosen with 3 types of transport mode: walking, transit, drive. The Google Maps Javascript API url: <https://developers.google.com/maps/documentation/javascript/>.

Mashup Use Cases & Dependent Services

Use Case 1

1. **Case:** As a **food enthusiast/ human that needs to eat**, I want the system to **suggest restaurants, or food stalls according to the type of cuisine I have chosen** so that **I could try different kinds of cuisine**.

2. **Screenshot:**

The screenshot shows the 'Discover Restaurants' web application. At the top, there is a dark header with the text 'Discover Restaurants!' and a hamburger menu icon. Below this is a green banner with the text 'Welcome to Discover Restaurants' and 'Discover Food and Restaurants from all around the world.' The main content area has a light gray background with the heading 'Search for a city.' Below this is a search bar with a magnifying glass icon and the placeholder text 'Search for a city'. Under the search bar, there is a section titled 'Find restaurants, cafés, bars and food stalls by selecting the options.' which contains three dropdown menus labeled 'Category', 'Cuisine', and 'Establishments', each with the text 'Options not selected' and a downward arrow. A blue 'Submit' button is located at the bottom right of this section.

Allan wants to eat some Japanese food. He would would like to find some Japanese restaurants around Melbourne. He first enters the city name.

This screenshot shows a close-up of the city search dropdown menu. The search bar contains the text 'mel' with a red underline. Below the search bar, a dropdown list is open, showing two options: 'Melbourne, VIC' and 'Melbourne, FL'. A mouse cursor is pointing at the 'Melbourne, FL' option. The dropdown list is bordered by a light gray background.

The city names are displayed. Allan can now choose the city. Once a city is selected, the restaurant category, cuisine and establishment options are displayed.

Welcome
Discover F

Q Melbourne, VIC

Find restaurants, ca

Category

Cuisine

Establishments

Submit

All Options
Delivery
Dine-out
Nightlife
Catching-up
Takeaway
Cafes
Daily Menus
Breakfast
Lunch
Dinner
Pubs & Bars
Pocket Friendly Delivery
Clubs & Lounges













All Options

All Options

All Options

Allan chooses the category and proceeds to click on the Submit button.

Result List:

 <p>Humble Rays Cafe, Coffee and Tea Carlton</p>	 <p>Calia Japanese, Australian Emporium Melbourne</p>	 <p>Kisumé Japanese Flinders Lane</p>	 <p>Chin Chin Thai, Asian, Fusion Flinders Lane</p>
 <p>Dinner by Heston Blumenthal European Crown Melbourne, Southbank</p>	 <p>Tito 00 Italian Little Bourke Street</p>	 <p>Mr. Miyagi Asian, Japanese Windsor</p>	 <p>Higher Ground Cafe, Australian, Coffee and Tea Little Bourke Street</p>
 <p>The Hardware Societe Cafe, French Hardware Street</p>	 <p>Lune Croissanterie Bakery, French Fitzroy</p>	 <p>Light Years Cafe, Coffee and Tea, Modern Australian Hawthorn</p>	 <p>Serotonin Eatery Cafe, Healthy Food Richmond</p>

A list of restaurant will be displayed. The list will have simple details of the restaurant's name, cuisine and location.



Calia

Emporium Melbourne
Emporium Melbourne, Level 3, 287 Lonsdale Street, CBD, Melbourne

CUISINES:	Japanese, Australian
COST FOR TWO:	\$ 55 (local currency)
RATING:	4.1 /5, <i>Very Good</i> [Based on 447 votes]
DELIVERY:	No
BOOKING:	No
PHOTOS:	Photos
MENU:	Menu
EVENTS:	Events

Allan can click on one of the restaurant links to get into another page which will now show greater restaurant details including price, ratings, location and cuisine.

3. API Endpoints Calls [1]:

Zomato API key is placed in the header instead of in the url when making the HTTPS GET request.

- **Get cities name:**

<https://developers.zomato.com/api/v2.1/cities?> + 'q=' + city name;

Zomato API returns a list of city with details from endpoints in JSON Format based on the query input given.

- **Get Latitude and Longitude of City:**

<https://developers.zomato.com/api/v2.1/locations?> + 'query=' + location name + '&count=' + max number of results to fetch;

Zomato API returns only one location details from endpoints in JSON Format based on the location name given as &count is always set to 1.

- **Get Categories of restaurant in a city:**

<https://developers.zomato.com/api/v2.1/categories?>;

<https://developers.zomato.com/api/v2.1/cuisines?> + 'city_id=' + city ID;

<https://developers.zomato.com/api/v2.1/establishments?> + 'city_id=' + city ID;

Zomato API returns a list of categories, cuisines and establishments from endpoints in JSON Format based on the city ID given.

■ **Search restaurants in a city:**

'<https://developers.zomato.com/api/v2.1/search?>' + 'entity_id=' + entity ID
'&entity_type=' + entity type + '&start=' + offset + '&radius=' + radius +
'&cuisines=' + cuisine + '&establishment_type=' + establishment +
'&category=' + category;

Zomato API returns a list of restaurants from endpoints in JSON Format based on the entity ID, cuisine type, establishment type, and category type given. &entity_type is always city, &start is set to 0, 20,40,60,80 respectively in 5 calls to get 100 restaurants and the &radius is set to 1000m.

■ **Get one restaurant:**

'<https://developers.zomato.com/api/v2.1/restaurant?>' + 'res_id=' +
restaurant ID;

Zomato API returns one restaurant details, from endpoints in JSON Format based on the restaurant ID given.

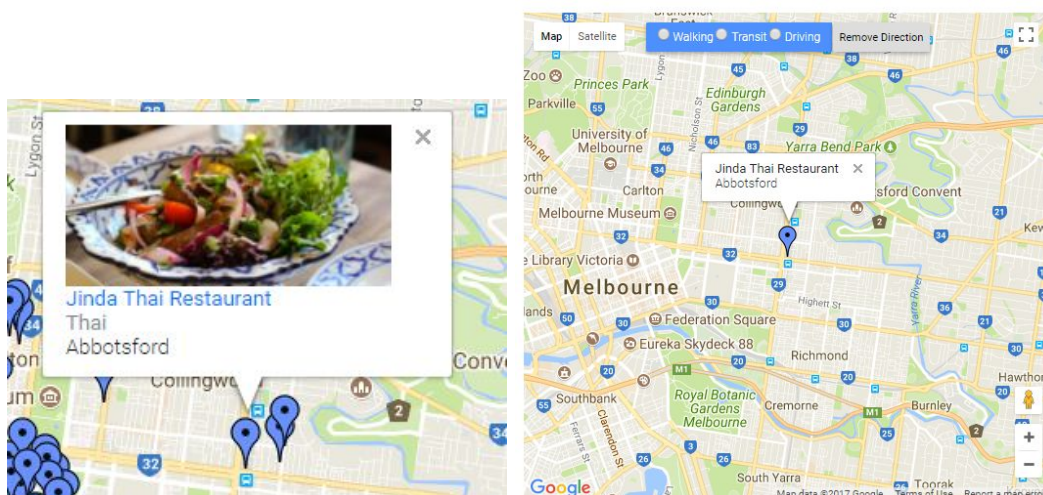
Use Case 2

1. **Case:** As a **food enthusiast/ human that needs to eat**, I would like to **check the location of the aforementioned restaurant/food stall on a navigable map** so that I could navigate to the location.

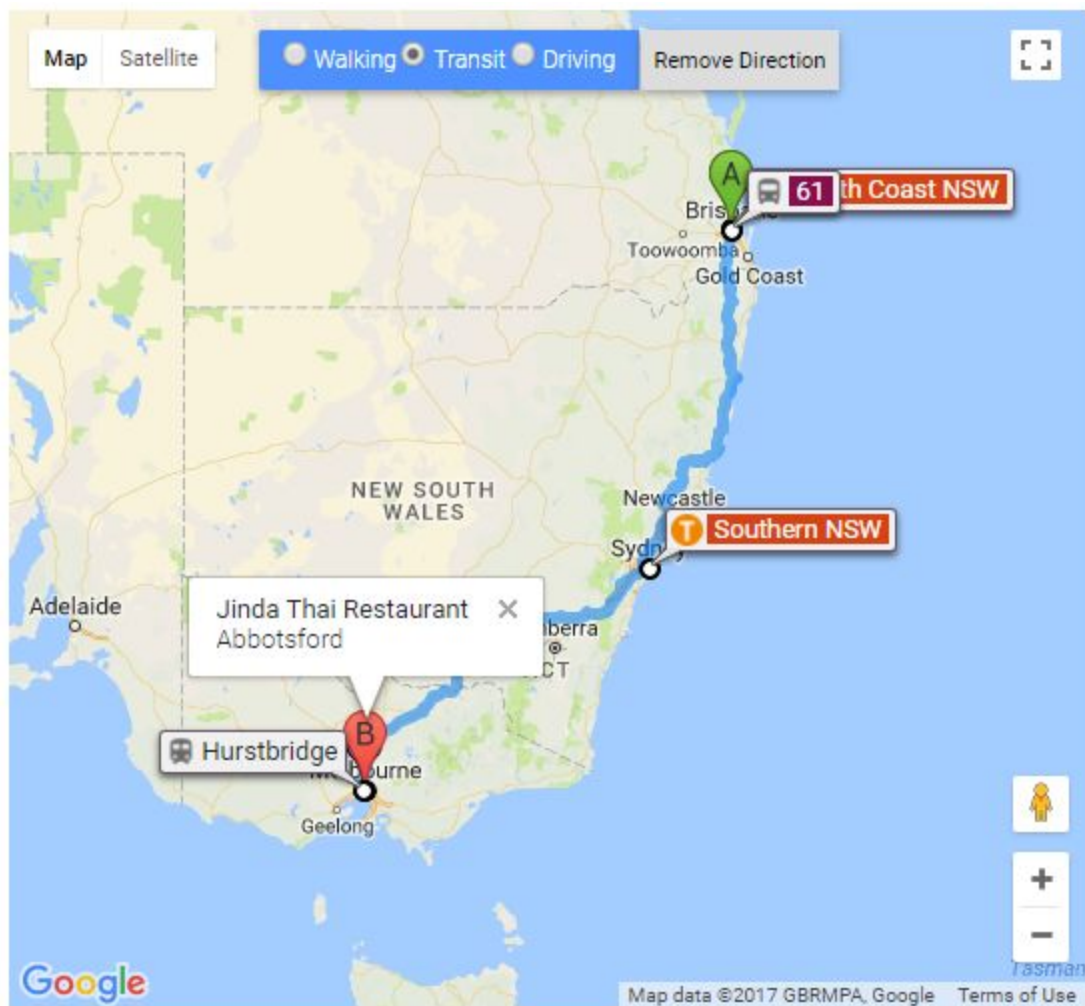
2. **Screenshot:**



Allan wants to go to a restaurant in Melbourne. He has searched and a list of restaurants and their locations are tagged on the Google Map.



He then clicked on one of the restaurant he has searched. The page redirects to the restaurant's detailed details and the map.



Allan can request the map to show the direction to the restaurant from his current location. He can choose 3 different modes which is Walking, Transit and Driving.

3. API Endpoints Calls [2]:

- **Get map:**

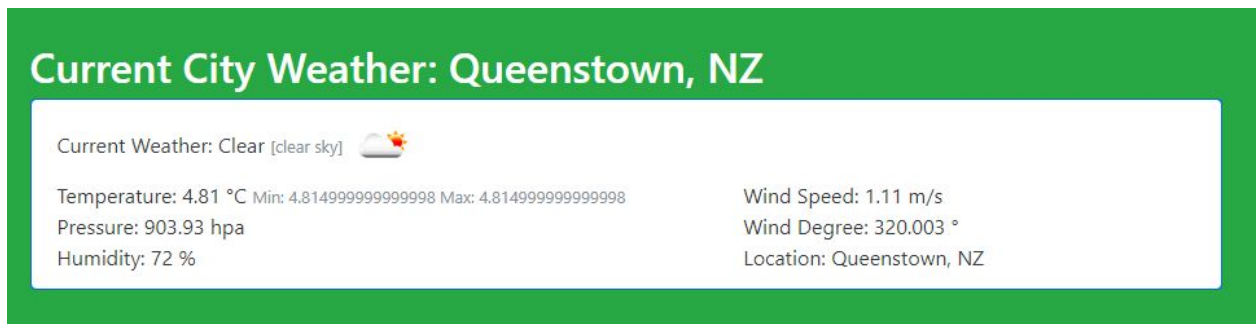
'<https://maps.googleapis.com/maps/api/js?>' + 'key=' + APIkey + 'libraries=' + places;

Google Map Javascript API returns a Google Map which can be used to place tags, set infowindow, get directions.

Use Case 3

1. **Case:** As a **food enthusiast/ human that needs to eat**, I want the system to **show me the current weather conditions nearby the restaurants** so that **I could avoid nasty weather conditions such as heavy rain or heavy snow**.

2. **Screenshot:**




Note: Image taken when it's Spring in Queenstown, NZ

Allan went to visit a mate in Queenstown, NZ. As a Queenslander, Allan could not withstand the cold New Zealand spring. He wants to check the current weather around Queenstown before deciding to go out from his hotel to eat.



Note: Image taken when it's Spring in Auckland, NZ

Alias, the weather is too cold for him. He decides to go to Auckland, NZ to eat, he checks the weather conditions of the restaurant around Auckland.

Current Weather: Clouds [scattered clouds] 

Temperature: 15.00 °C Min: 15 Max: 15	Wind Speed: 10.8 m/s
Pressure: 1013 hpa	Wind Degree: 250 °
Humidity: 72 %	Location: Takapuna, NZ

He has chosen a restaurant at Takapuna, north of Auckland. The current weather around Takapuna is displayed.

3. API Endpoints Calls [3]:

- **Get current weather:**

'<https://api.openweathermap.org/data/2.5/weather?>' + '&lat=' + entity Latitude + '&lon=' + entity Longitude + '&appid=' + APIkey;

OpenWeatherMap API returns the current weather condition details from endpoints in JSON Format based on the latitude and longitude given.

Technical Description

The service is built using markup language HTML5 and stylesheet language CSS3 in Embedded JavaScript templates and scripting language JavaScript which is served by a Node.js server contained in a Docker container hosted by Amazon Web Services EC2 on an Ubuntu 16.04 machine.

Server-side

The web server runs on the Express JS web framework, which sits atop of Node JS, which sits on a Docker container. A number of other Node modules packages are also used to fulfil the needs of the service. Here are all the server-side components used in greater detail:

- **Node JS [4]:**
Open-source, cross-platform, asynchronous javascript runtime framework for building event driven network applications. The node.js server is used to handle serving files as well as send requests and receive response from APIs.
- **Express JS [5]:**
A simple web framework that sets the routes and views easily, allowing the client and server-side to be integrated easily, which allows for rapid development. This is where API calls are made and parsed using the Async JS and Request JS library.
- **Async JS:**
A utility module that allows multiple HTTP/HTTPS calls to be made in a single callback. This modules is used in conjunction with Request JS.
- **Request JS:**
A simple HTTP/HTTPS request library that makes HTTP/HTTPS calls the simplest way possible. GET and POST request are made and data is returned back in JSON format which can be easily converted to Javascript objects and send to the client side.
- **Docker:**
Below in the Docker section.

Client-side

The web pages are presented using HTML5 and CSS3, and generated through the use of Embedded JavaScript templates. Twitter's Bootstrap CSS framework has also been leveraged to provide a simple and consistent UI. Here are all the client-side components used in greater detail:

- **HTML5 & CSS3:**

Markup and stylesheet language used to build web pages. HTML provides the structure of the page and CSS styles the pages, setting the colors, layout and fonts.

- **JavaScript::**

This client-side of Javascript enables interactive features and dynamic functions. It is used to render the Google Map, get user's browser location and plot restaurants locations.

- **Bootstrap:**

Most popular HTML, CSS and JavaScript framework used to develop simple and consistent UI for web page. It contains the HTML and CSS based design template for typography, forms, buttons, navigation and other interface components. All web pages in this services are styled using Bootstrap components.

- **JQuery::**

A Javascript library used for animating effect, event listening and event handling. It is also used for resetting query and sending query from the client-side via AJAX calls to the server-side Express JS routes which then returns Javascript objects to be handled.

- **EJS, Embedded JavaScript templates [6]:**

A template of Express framework. Javascript can be written directly with the HTML markup in EJS template. It also allows Javascript objects to be rendered from the server side (Node.js) to the client side(EJS) with ease.

Difficulties

1. Unstable API, HTTP Error 502 Bad gateway

The Zomato API gateway or proxy used to fulfil the API call from the service received an invalid response from an upstream server. Apparently, the upstream server was down for a moment and it happened during the services testing phase. Nothing can be done from my end, only hoping that the downtime is due to maintenance and not server down.

2. Rate Limiting

The Zomato API endpoints have rate limiting of 1000 requests / day [1], while the OpenWeatherMap API endpoints have rate limiting of 60 requests / minute [2]. The breach could occur if multiple users accessed and make too many API call at the site at the same time.

3. Multiple Request

It was a hassle to create a nested HTTP request call. Thus the Async Request library is used which first stores all the request options in an array. The Async function processed all the HTTP request and concat the response data in JSON format.

4. Geolocation API call from non secure-origin

Many web browsers such as Chrome, Opera, no longer supports obtaining the user's location using the HTML5 Geolocation API from pages delivered by non-secure connections [7]. Surprisingly, it still worked for Internet Explorer, Edge and Firefox. I have set the service to run on HTTPS which mean Geolocation can be used in any browser. The caveat is the SSL certificate is self signed [8], thus, the connection to the Docker container service will display a “Unsecure connection warning” (but don't worry, it's safe).

5. API Inaccuracy

The Zomato API for /locations is not working accurately for some location names. <https://developers.zomato.com/api/v2.1/locations?> + 'query=' + location name + '&count=' + max number of results to fetch;

For example, passing the city name “Sydney, NSW” into the query will return Canberra, ACT, with coordinates at Queanbeyan, NSW.

Docker

Docker allows containerized services to run the same regardless of the environment and operating system it is running on. Docker containers include the application and all of its configs and dependencies beforehand, allowing resource isolation and allocation benefits of running off a virtual machine [9]. It also removes the worries of environmental inconsistency in deployment. Docker images are used to run the container, which is built by following a set of commands laid out in a Dockerfile.

This mashup services is deployed using Docker and is hosted on Amazon EC2. The Docker image is built using Ubuntu 16.04 as a base and the commands written in a Dockerfile similar to the sample given in the Containerising Express App Practical. It first installs NodeJS Boron, and gets the software stack with pre-installed node dependencies copied into the image. The Dockerfile then sets the working directory, installs NPM, exposes port 3000, and sets the NPM start command for the server.

Now the mashup service is containerised in a Docker image. All it takes to run the image is to put it in a Docker container on the AWS machine, ensuring it ran daemonised and mapping port 443 to port 3000 used by the server.

Testing & Limitation

Test Plan & Results

ID	Purpose	Expected/Actual	Pass?	ScreenLink
1	Search for Restaurants by inputting letters	E: City names list shown if matched A: City names list shown if matched	Pass	1.0 - 2.0
2	Select a city	E: Form with restaurant categories unlocked, can submit A: Form with restaurant categories unlocked, can submit	Pass	3.0
3	Change city name	E: Form locked, cannot submit A: Form locked, cannot submit	Pass	4.0
4	Submit form	E: Restaurant List, Map, Current weather displayed A: Restaurant List, Map, Current weather displayed	Pass	5.0, 8.0, 9.0
5	Geolocation request approved	E: Show current location A: Show current location	Pass	N/A
6	Geolocation request not approved	E: Current location not shown A: Current location not shown	Pass	N/A
7	Click on restaurant tag	E: Restaurant name, location, picture displayed A: Restaurant name, location, picture displayed	Pass	6.0
8	Enter origin and destination	E: Direction between origin and destination shown in map A: Direction between origin and destination shown in map	Pass	7.0
9	Select transport mode (in Result page)	E: Direction shown in map, with relevant transport A: Direction shown in map, with relevant transport	Pass	7.0 - 7.2
10	Click on restaurant picture/ link	E: Restaurant Details displayed, location shown A: Restaurant Details displayed, location shown	Pass	10.0 - 12.0
11	Select transport mode (in Info page)	E: Direction shown in map, with relevant transport A: Direction shown in map, with relevant transport	Pass	13.0 - 13.2
12	Click on button to remove direction	E: Direction removed from map A: Direction removed from map	Pass	13.3
13	Click on back button	E: Return to Result page A: Return to Result page	Pass	N/A

Limitations

1. Inaccurate API Calls:

Zomato's /cities API calls to get city details do not return the Latitude and Longitude of the city. This is a problem and the only workaround is to use to /locations to get the Latitude and Longitude of the city. As stated in the Difficulties section, some cities like Sydney, NSW returned coordinates of a different location altogether which is impossible to resolve from my end.

2. Limited Restaurant objects API Calls:

Zomato's /search API returns all the restaurant found based on the API query, but it can only return a maximum of 100 restaurants. To make matters worse, only 20 restaurants can be get per HTTP request, which can be been solved by calling 5 HTTP request with different offset parameters of 0,20,40,60,80. Unfortunately, that is all that is possible from my end.

Possible Extensions

A feature that could be added is to display instagram pictures along with the restaurant details. This is especially useful for users who wants to try different kinds of cuisine in different restaurants, and the easiest way to attract a user is to present a good impression of the restaurant, in this case, beautiful photos!

This Instagram integration can be done by querying the restaurant name as a tag, for example, the restaurant name is Gordon Ramsay's Burger Las Vegas, then the tag would be **#Gordon Ramsay's Burger Las Vegas**. The Instagram API would then return Instagram pictures links related to the tagged restaurant name.

Reference

- [1] Zomato, "Zomato API", *Zomato*, 2017. [Online]. Available: <https://developers.zomato.com/documentation>. [Accessed: 16- Sep- 2017].
- [2] Google Developers, "Google Maps JavaScript API | Google Developers", *Google Developers*, 2017. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/>. [Accessed: 16- Sep- 2017].
- [3] OpenWeatherMap, "Current weather data - OpenWeatherMap", *Openweathermap.org*, 2017. [Online]. Available: <https://openweathermap.org/current>. [Accessed: 16- Sep- 2017].
- [4] N. Foundation, "Node.js", *Node.js*, 2017. [Online]. Available: <https://nodejs.org/en/>. [Accessed: 16- Sep- 2017].
- [5] Express, "Express - Node.js web application framework", *Expressjs.com*, 2017. [Online]. Available: <https://expressjs.com>. [Accessed: 16- Sep- 2017].
- [6] EJS, "EJS -- Embedded JavaScript templates", *Ejs.co*, 2017. [Online]. Available: <http://ejs.co/>. [Accessed: 16- Sep- 2017].
- [7] P. Kinlan, "Geolocation API Removed from Unsecured Origins in Chrome 50 | Web | Google Developers", *Google Developers*, 2017. [Online]. Available: <https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only>. [Accessed: 16- Sep- 2017].
- [8] F. Rappl, "How to Use SSL/TLS with Node.js — SitePoint", *SitePoint*, 2017. [Online]. Available: <https://www.sitepoint.com/how-to-use-ssl-tls-with-node-js/>. [Accessed: 16- Sep- 2017].
- [9] Docker, "What is Docker", *Docker*, 2017. [Online]. Available: <https://www.docker.com/what-docker>. [Accessed: 16- Sep- 2017].

Appendix

Brief User Guide

Index Page

1.0

Discover Restaurants! Search

Welcome to Discover Restaurants

Discover Food and Restaurants from all around the world.

Search for a city.

Search for a city

Find restaurants, cafés, bars and food stalls by selecting the options.

Category Options not selected ▼

Cuisine Options not selected ▼

Establishments Options not selected ▼

Submit

1. This is the Index page. Enter some letters to start searching for a city.

2.0

Search for a city.

Melbourne, VIC

Melbourne, FL

Melita, MB

Melbourne, AR

Melrose, MN

Melvern, KS

Mellen, WI

Melissa, TX

2. The search bar will automatically display the list of cities available to choose.

3.0

Search for a city.

Find restaurants, cafés, bars and food stalls by selecting the options.

Category	<input type="text" value="Dinner"/>
Cuisine	<input type="text" value="Asian"/>
Establishments	<input type="text" value="All Options"/>

Submit

3. Select the restaurant category, cuisine and establishments. Click on the Submit button to submit the form.

4.0

Search for a city.

Q

Melbourne, VI

Find restaurants, cafés, bars and food stalls by selecting the options.

Category

Takeaway

Cuisine

Bar Food

Establishments

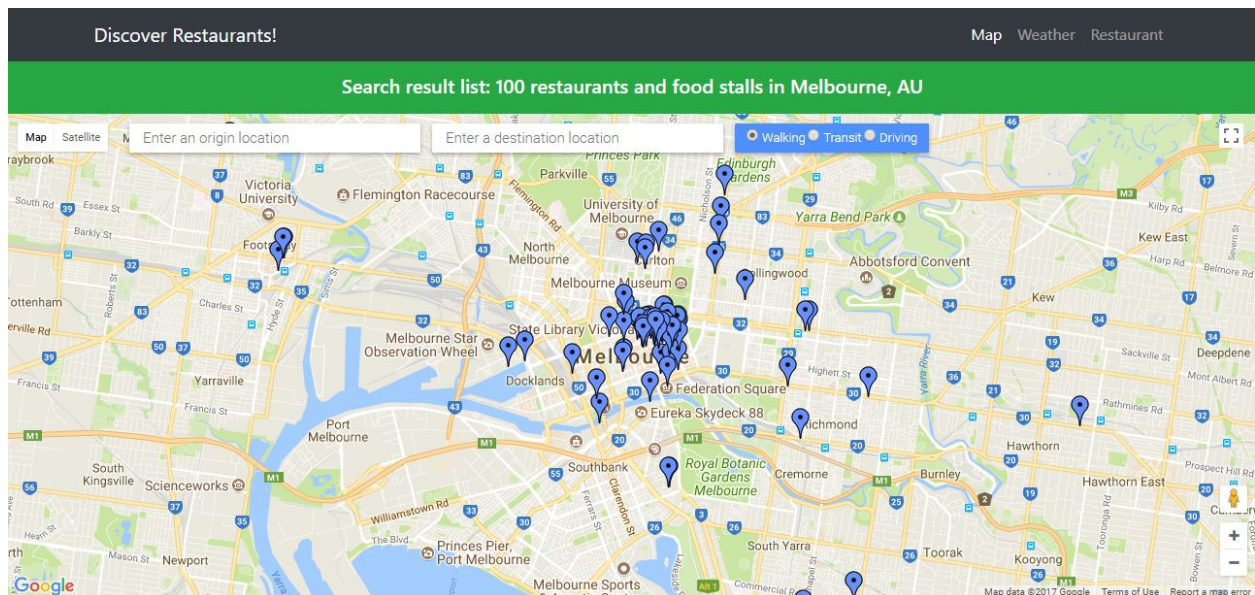
All Options

Submit

4. If you decided to change the city, the form will lock itself and you cannot submit the form until you have chosen a city.

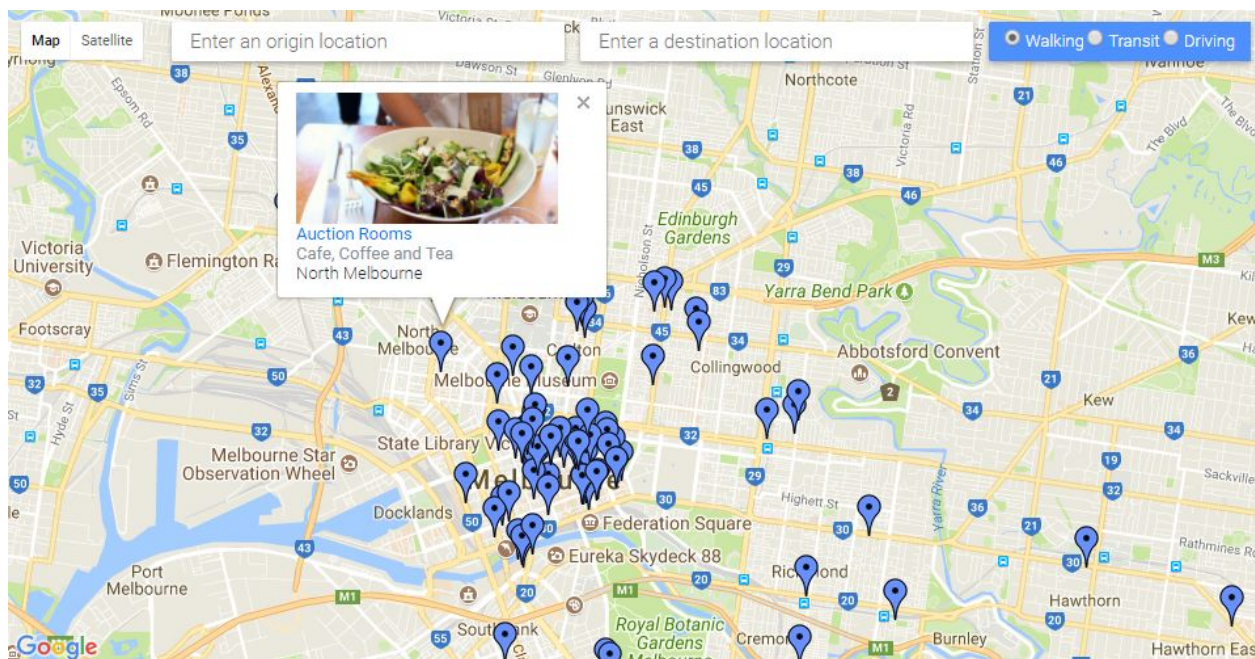
Result Page

5.0



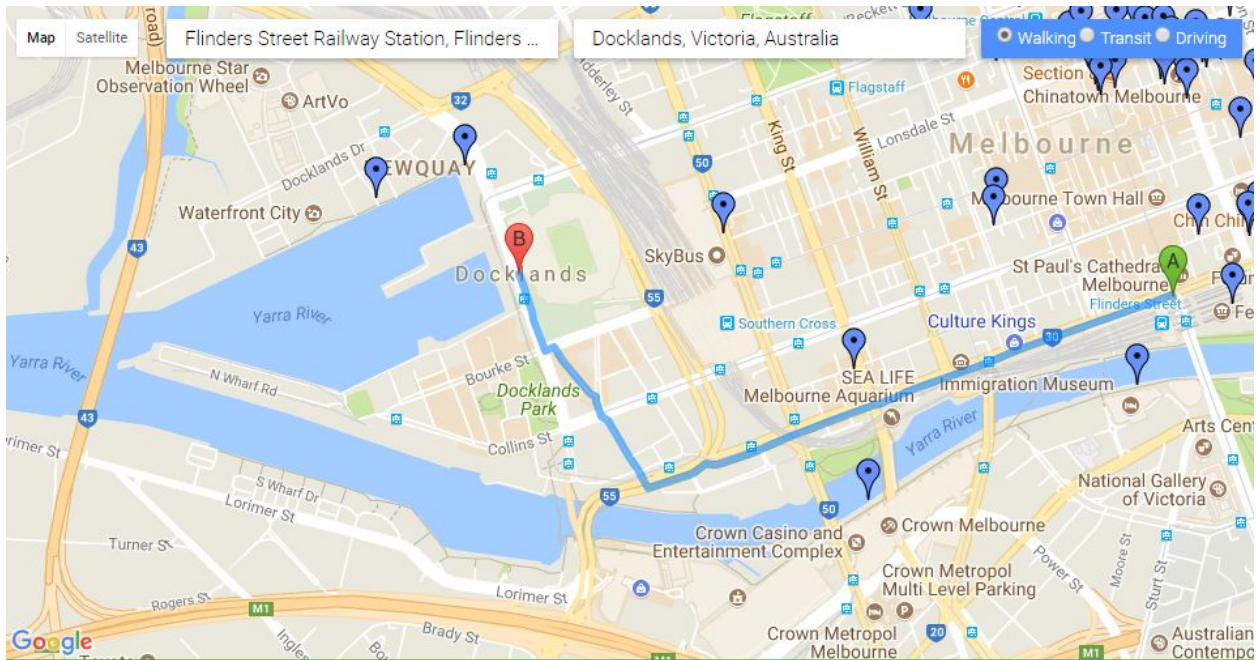
5. The queried restaurants that fulfill the criteria is displayed on the map.

6.0



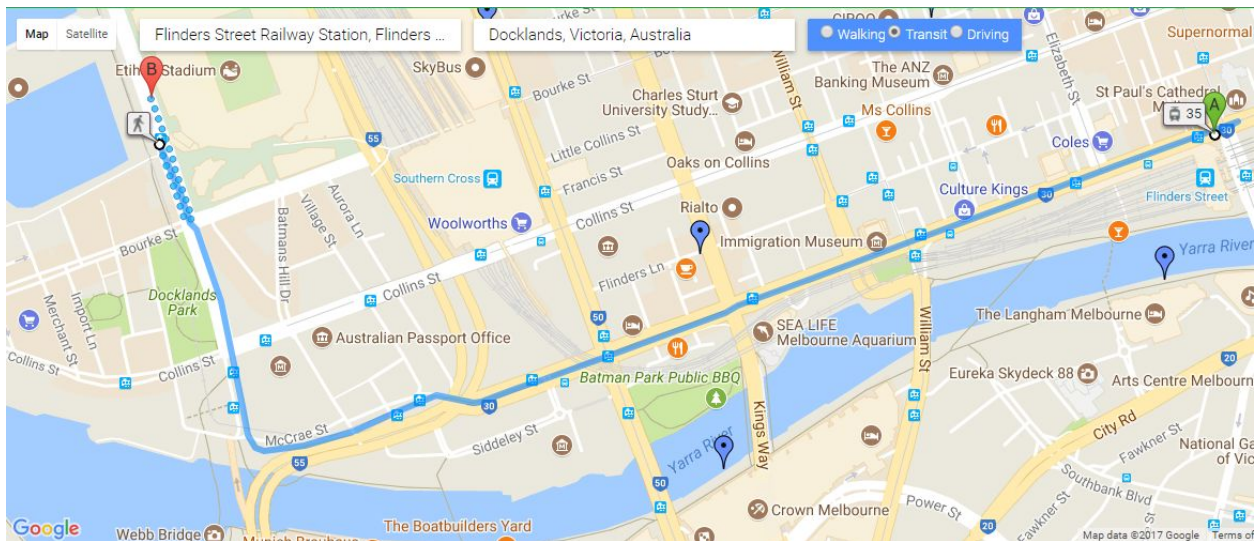
6. Click on one of the tag. The restaurant's picture, name, cuisine and location is displayed.

7.0



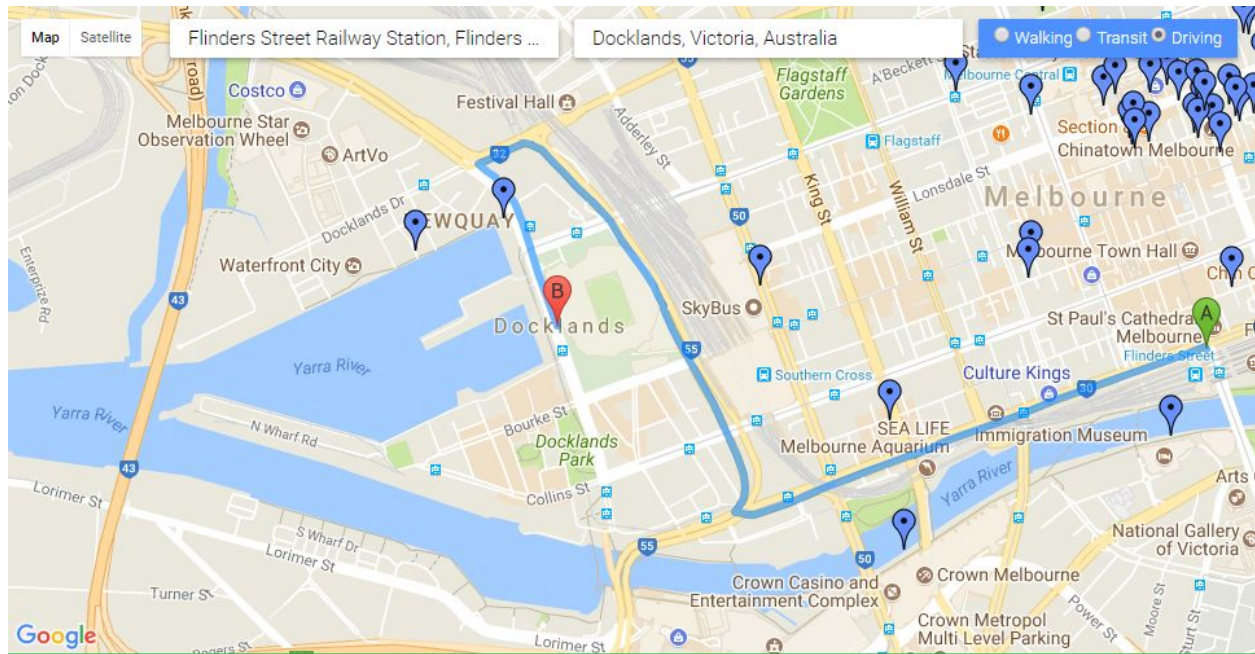
7. Enter the origin and destination address to search for directions. You are now looking at the recommended walking direction.

7.1



8. You can change the transport mode by clicking on one of the radio buttons. You are now looking at the transit direction.


7.2



9. You are now looking at the recommended driving direction.

8.0

Current City Weather: Melbourne, AU

Current Weather: Clouds [few clouds] 

Temperature: 6.38 °C Min: 5 Max: 8

Pressure: 1027 hpa

Humidity: 86 %

Wind Speed: 3.6 m/s

Wind Degree: 270 °

Location: Melbourne, AU

10. Scroll down and the current city weather is displayed.

9.0

Result List:



Chin Chin

Thai, Asian, Fusion
Flinders Lane



Mr. Miyagi

Asian, Japanese
Windsor



Rice Paper Scissors

Asian, Tapas
Liverpool Street



Supernormal

Asian, Fusion
Flinders Lane



Longrain Melbourne

Asian, Thai
Little Bourke Street



Darac Grill & Bar

Asian, Japanese, Korean
A'Beckett Street



Hocht Mama

Asian, Vietnamese
Liverpool Street



Peko Peko

Taiwanese, Japanese, Asian
South Melbourne



Don Don

Japanese, Asian
Little Lonsdale Street



Tom Phat

Asian, Fusion
Brunswick



Laksa Bar

Asian
Little Lonsdale Street



Gami

Korean, Asian
Little Lonsdale Street

11. Scroll further down and the restaurants are displayed.

10.0



Mr. Miyagi

Asian, Japanese
Windsor

12. Interested? Click on one of the restaurants to view for more detail.


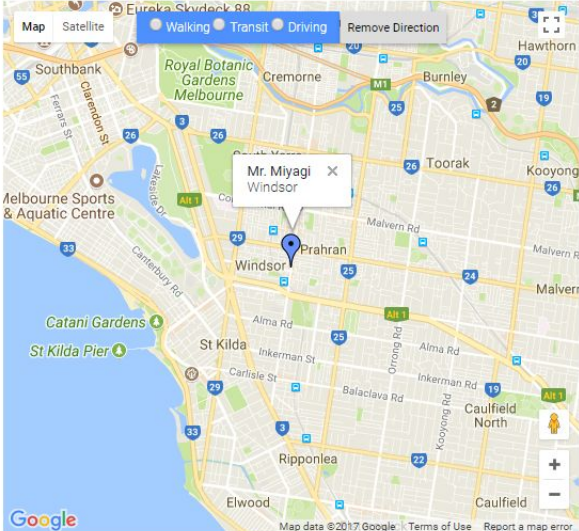
Info Page

11.0

Discover Restaurants!

Restaurant Details Back

Mr. Miyagi




Mr. Miyagi

Windsor

99 Chapel Street, Windsor, Melbourne

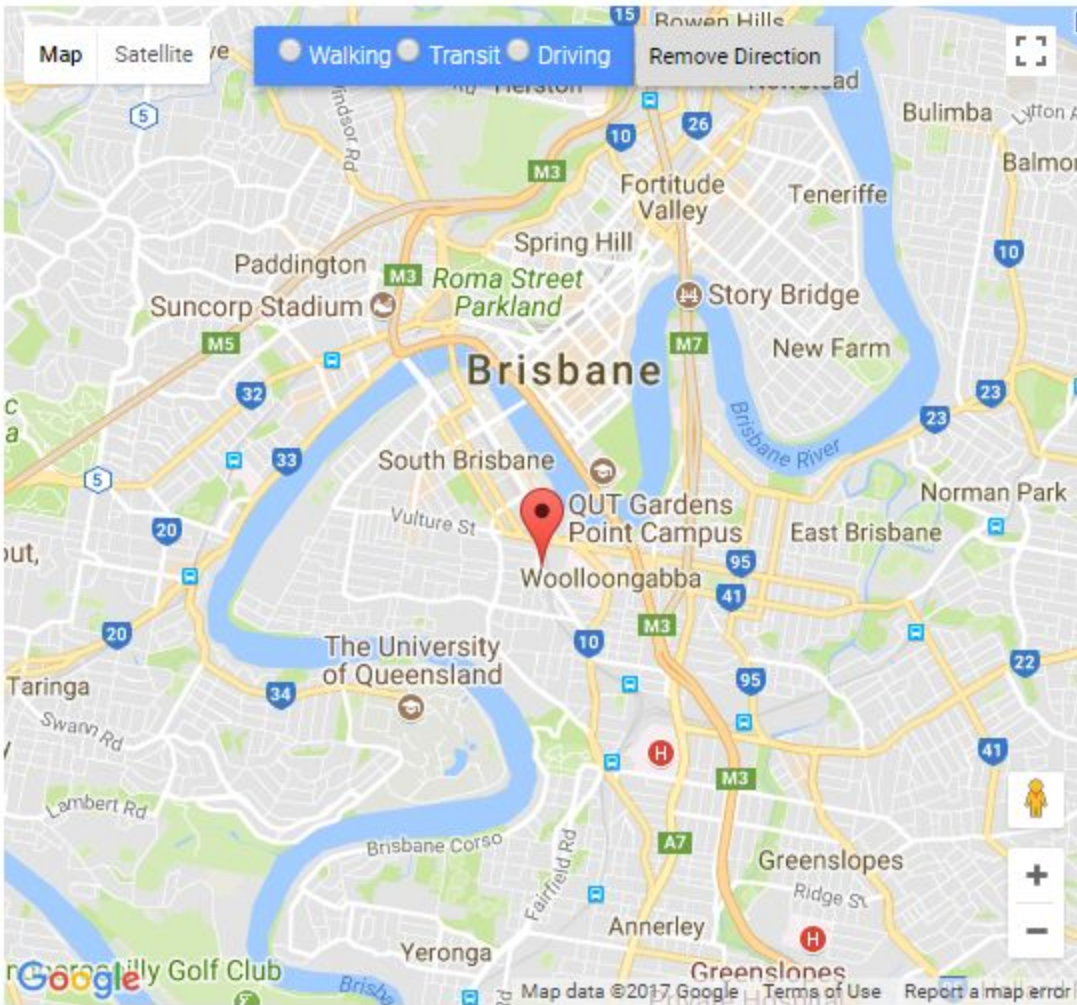
CUISINES:	Asian, Japanese
COST FOR TWO:	\$ 80 (local currency)
RATING:	4.4 /5, <i>Very Good</i> [Based on 1423 votes]
DELIVERY:	No
BOOKING:	No
PHOTOS:	Photos
MENU:	Menu
EVENTS:	Events

Current Weather: Clouds [few clouds] 

Temperature: 6.39 °C Min: 5 Max: 8	Wind Speed: 3.6 m/s
Pressure: 1027 hpa	Wind Degree: 270 °
Humidity: 86 %	Location: Prahran, AU

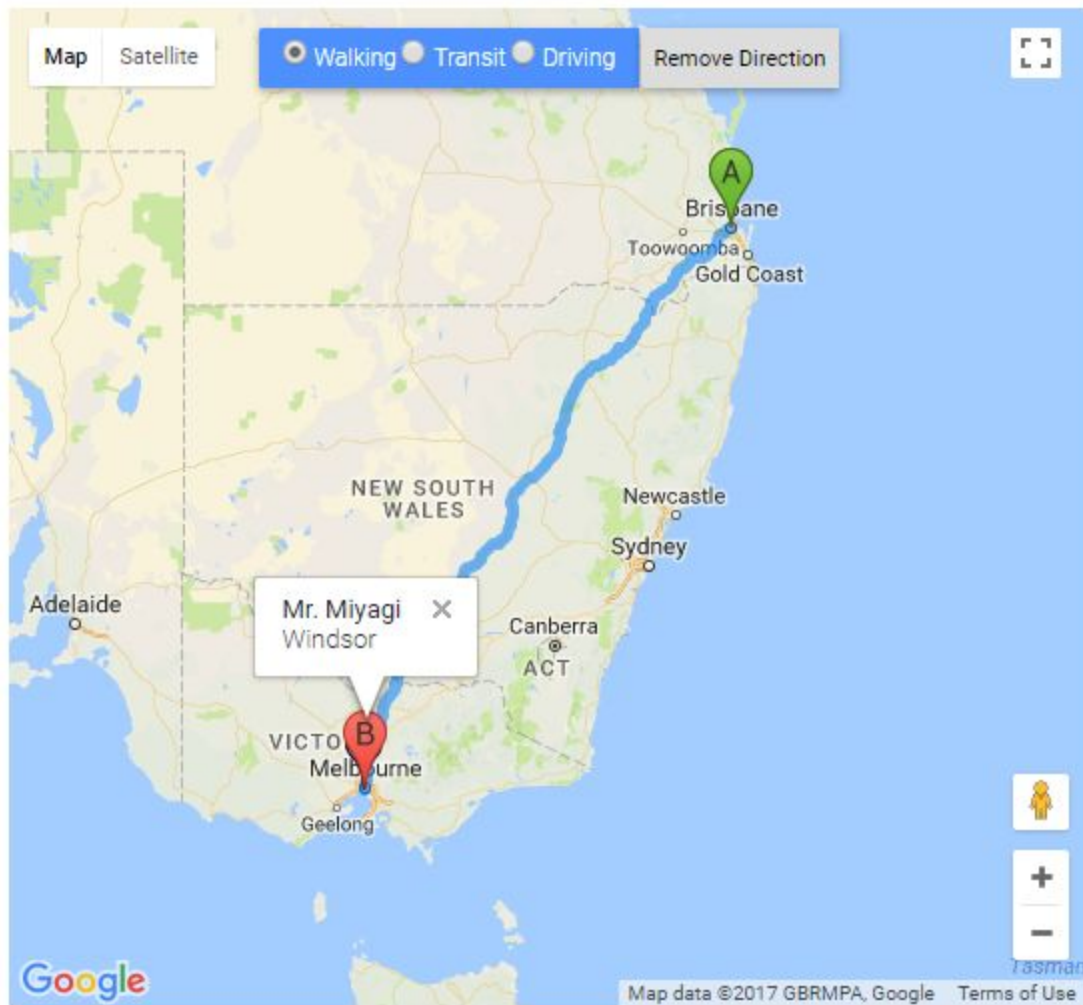
13. The restaurant details, current weather conditions and its location is displayed.

12.0



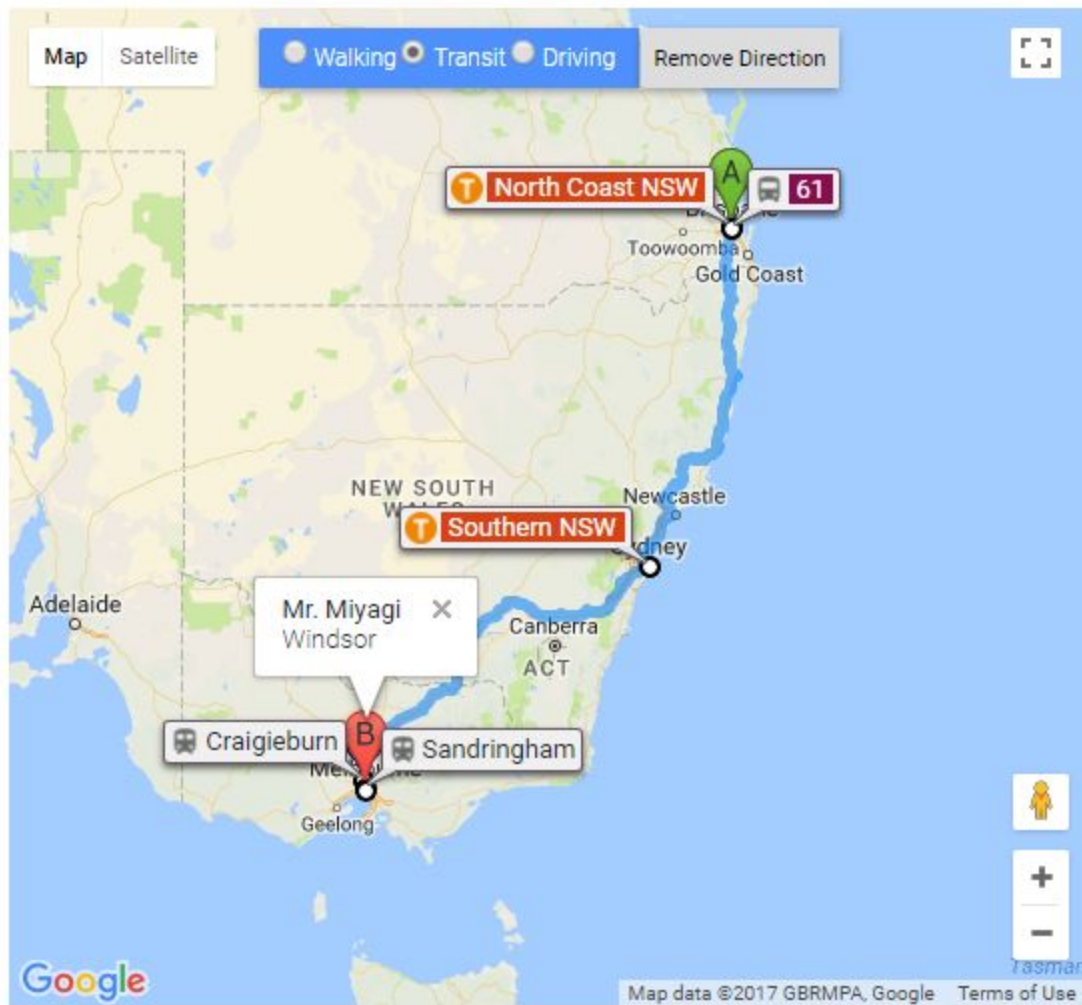
14. If you have enabled Geolocation, you can check where you are currently at.

13.0



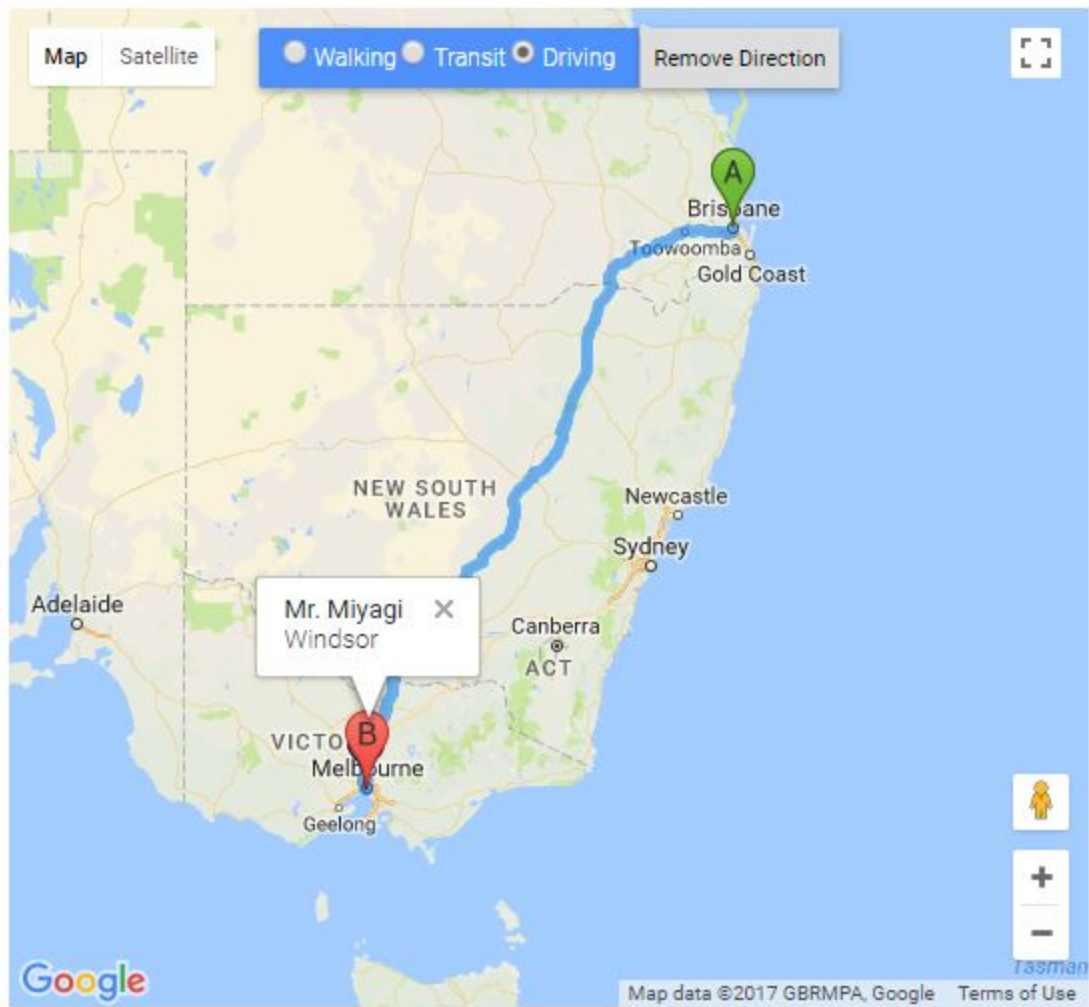
15. Select the direction mode to display the direction to the restaurant. Geez, that is a long distance.

13.1



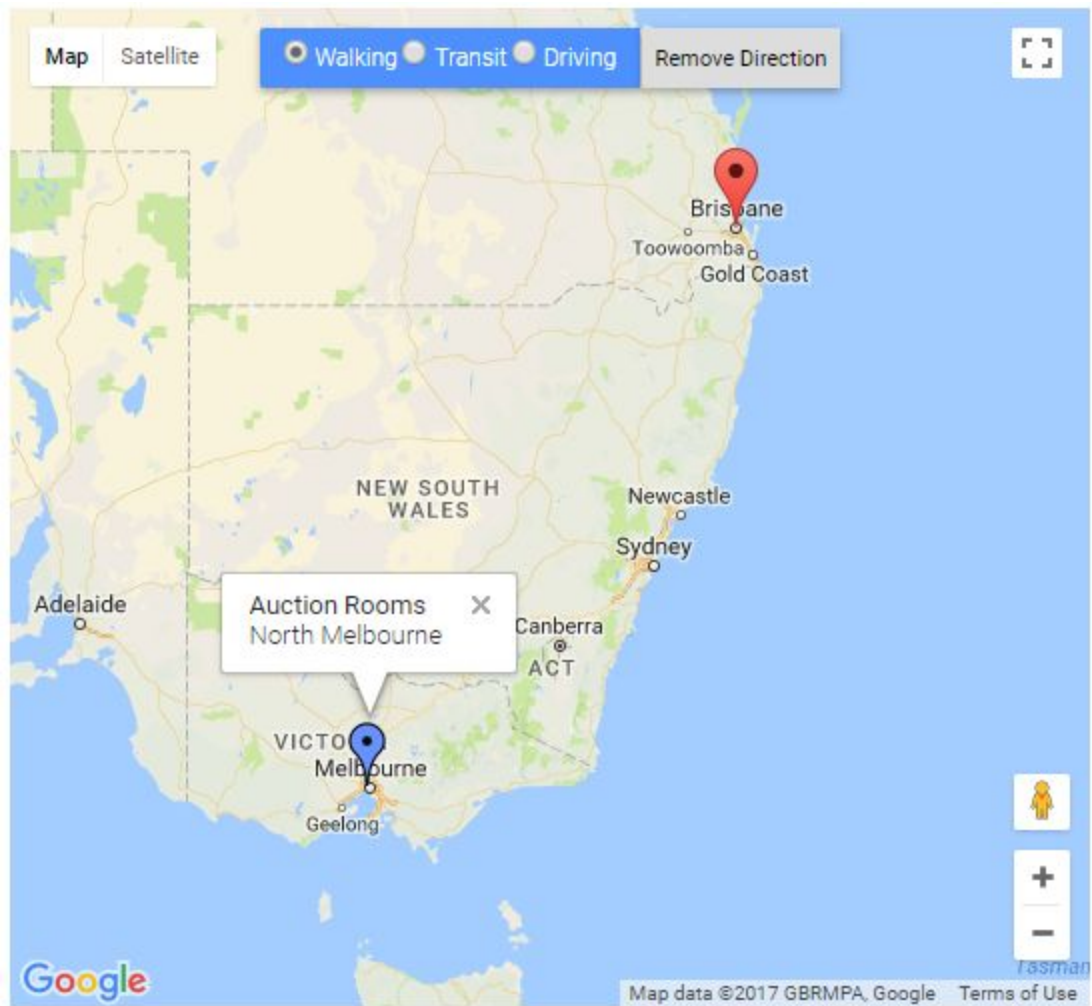
16. You can change the direction mode to Transit.

13.2



17. Here is the direction mode by driving.

13.3



18. Click on the Remove Direction button to remove the direction from the map.

Dockerfile

FROM node:boron

Copy app source
COPY . /src

Set work directory to /src
WORKDIR /src

Install app dependencies
RUN npm install

Expose port to outside world
EXPOSE 3000

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