

# **FIT3164 - Data Science Project 2**

## **Testing Report**

### **Project 8A: Suburb Selector**

Group 14

1. Cipta Pratama 30685125
2. Lam Thy Ngo 30510287
3. Yao Meng 31109160
4. Gavin Toby Darmawangsa 30533236

<b>FIT3164 - Data Science Project 2</b>	<b>1</b>
<b>Introduction</b>	<b>4</b>
<b>Test Approach</b>	<b>4</b>
<b>White Box Testing</b>	<b>5</b>
Path Transversal (Unit Testing)	5
Test Number 1 ( Only-Filtering Algorithm )	7
Test Number 2 ( Only-Scoring Algorithm )	8
Test Number 3 (Result from both Filtered List and Scoring-only List)	9
Result:	10
API Test	11
Test Number 5:	11
Test Number 6:	12
Result:	13
Database Retrieval Test:	14
Test Number 7: Suburb Query	14
Test Number 8: Rent Query	14
Test Number 9: Hospital Query	15
Result:	16
<b>Black Box Testing</b>	<b>17</b>
Test Number 1 [Navigation Bar]	17
Test 1.1 Home Redirection	17
Test 1.2 Information For Suburb Redirection	18
Test 1.3 Preferences Searching Redirection	19
Test Number 2 [Home Page]	21

Test 2.1 Home Search Button Redirection	21
Test 2.2 Suburb Substring Match Search	22
Test 2.3 Preferences Searching Redirection	24
Test Number 3 [Suburb Information Page]	25
Test 3.1 Testing Drop Down List	25
Test 3.2 Testing Suburb Redirection	26
Test Number 4 [Preferences Searching Page]	28
Test 4.1 Input Validation	28
Test 4.2 Path Redirection to Top Suburb Search Result Page	30
Test 4.3 Path Redirection to Suburb Information	31
Test Number 5 [Travel Direction Calculation Feature]	32
Test 5.1 User Input Handling	33
Test 5.2 Direction Accuracy Testing	34
<b>Integration Testing</b>	<b>35</b>
<b>System Testing</b>	<b>38</b>
<b>Usability Testing</b>	<b>39</b>
<b>Performance</b>	<b>40</b>
<b>Scalability</b>	<b>40</b>
<b>Security</b>	<b>40</b>
<b>Limitations of Testing</b>	<b>41</b>
<b>Conclusion</b>	<b>41</b>

## Introduction

This project's main goal is to deliver a website for users, especially international students, which brings insights in terms of efficiently living in Melbourne during their study period abroad. In order to further reach the initial goal of the website product, a test report is required to be made that aims to assure that the product will be in its best form to fulfil the right quality requirement for users.

This report will cover a variety of types of testing for the website products. The testing includes unit testing, integration testing, usability testing and system testing. The unit testing and integration ensures all user inputs are handled properly and every functionality between the main interface's code and external libraries are working accordingly while the usability testing offers the product to be user-friendly and easy to use. Furthermore, this report will discuss more about the performance and security of this website.

## Test Approach

For website testing, manual testing is the approach that is decided to be used as for the test approach. Manual testing for each and every user input box to check validation checking and page redirection for black box test. The black box tests, combined with integration tests, will make sure that all user inputs, not considering the back-end code, combined all together, will work as expected.

Regarding white box testing, manual tests will be done to check if the functions inside the code run correctly. We will be testing the API and database query directly, as well as unit testing for the whole preference-result page, making sure that all steps on the code runs as expected. For all tests, any errors found will be mentioned and fixed accordingly, if possible.

# White Box Testing

## Path Transversal (Unit Testing)

For unit testing, we are mainly testing for the preference-result page, which contains the main algorithm to select the best suburbs which act as the main goal of the project. These are the step-by-step way on how the preference-result page runs.

1. Retrieve the user inputs (Line 81-85)
  - a. Uni ( Selected from list of university)
  - b. Max\_distance ( integer input 1-100)
  - c. Cinema (YES or NO)
  - d. Hospital (YES or NO)
  - e. Rent (Integer 150-500)
2. Case sensitivity and replace hospital and cinema variable for query (Line 89-98)
3. Filter suburbs from database based on 5 previous variables (Line 101-107)
4. Retrieve all suburbs from database (Line 110-116)
5. Score calculation from both filtered results and all suburb results (Line 121-199)
6. Find top 5 suburbs based on the result (Line 208-233) and sort
  - a. If Filtered results has more than or equal to 5 results (Line 208-212)
    - i. Proceed to show 5 results as the output
  - b. If Filtered results has less than 5 results (Line 213-233)
    - i. Retrieve x results from filtered results
    - ii. Retrieve 5-x results from all suburbs results.
7. Replacing hospital and cinema variable to showcase user input (Line 333-340)

We are testing so that the function correctly runs all of the steps above, while also to check if it could take on user input where it will trigger all kinds of scenarios. We wrote print statements on each part of the code for the algorithm above. There are 3 user inputs that will trigger 3 scenarios:

1. To trigger the only-filtering algorithm, we input the very maximum value allowed for rent and max distance, and input NO for both cinema and hospital availability.
2. To trigger the only-scoring algorithm, where the filtering algorithm will not even be used. We input the very minimum rent, max distance, and YES for cinema and hospital availability. This way, there is a very minimal or even none suburb will have the qualifications.
3. To trigger where the result will use both filtering and weighted scoring algorithms.

The user input will be a reasonable input to check that the code uses both algorithm and pathing through the code as we wanted to.

## Test Number 1 ( Only-Filtering Algorithm )

Input:

Choose University

Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)

500

Max Distance (1-100)

100

Cinema(YES or NO)

NO

Hospital(YES or NO)

NO

**Search**

Print Output statements:

```
Step 2.a.2 (Cinema is NO)
Step 2.b.2 (Hospital is NO)
Step 3, Retrieving filtered suburbs
Step 4, Retrieving all suburbs
Step 5, Score Calculation for Filtered Results
Step 5, Score Calculation for All Results
Step 6a (Filtered results >= 5)
['Clayton', 'Ashburton', 'Burwood', 'Hughesdale', 'Springvale']
```

Result output:

Campus:Monash University Clayton Campus	Max Distance: 100 km	Cinema: exclude	Hospital: exclude	Rent:500
Clayton	Estimated Distance: 0.538091383 km Median Rent: A\$280 Number of Hospital: 6 Number of Cinema: 1 Closest Hospital: 0.52 km Closest Cinema: 1.22 km Score: 66.11	Ashburton	Burwood	Hughesdale
	Estimated Distance: 7.290105785 km Median Rent: A\$240 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 1.49 km Closest Cinema: 2.04 km Score: 62.44		Estimated Distance: 7.540468984 km Median Rent: A\$240 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.94 km Closest Cinema: 1.71 km Score: 62.28	Estimated Distance: 5.049328688 km Median Rent: A\$280 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 2.95 km Closest Cinema: 1.54 km Score: 61.77
				Springvale Estimated Distance: 4.05940792 km Median Rent: A\$270 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.23 km Closest Cinema: 3.13 km Score: 61.16

The output is the same as we expected, as it triggers Step 6a, where it retrieves 5 suburbs from the list that contains only-filtered suburbs. Furthermore, in the result output, there are no red highlights on the result, which means that all the output are all within the user preference. The print statements also show that all parts of the code used are based on what we expect. However, even though all the results are from the filtered suburbs, it is still ranked by the scoring system.

## Test Number 2 ( Only-Scoring Algorithm )

Input:

Choose University

Maximum Average Weekly Rent (150-500)

Max Distance (1-100)

Cinema(YES or NO)

Hospital(YES or NO)

Print Output Statement:

```
Step 2.a.1 (Cinema is YES)
Step 2.b.1 (Hospital is YES)
Step 3, Retrieving filtered suburbs
Step 4, Retrieving all suburbs
Step 5, Score Calculation for Filtered Results
Step 5, Score Calculation for All Results
Step 6b (Filtered results < 5)
['Clayton', 'Ashburton', 'Burwood', 'Hughesdale', 'Springvale']
```

Output:

Campus:Monash University Clayton Campus Max Distance: 1 km Cinema: include Hospital: include Rent:150	Clayton Estimated Distance: 0.538091383 km Median Rent: A\$280 Number of Hospital: 6 Number of Cinema: 1 Closest Hospital: 0.52 km Closest Cinema: 1.22 km Score: 66.11	Ashburton Estimated Distance: 7.290105785 km Median Rent: A\$240 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 1.49 km Closest Cinema: 2.04 km Score: 62.44	Burwood Estimated Distance: 7.540468984 km Median Rent: A\$240 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.94 km Closest Cinema: 1.71 km Score: 62.28	Hughesdale Estimated Distance: 5.049328688 km Median Rent: A\$280 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 2.95 km Closest Cinema: 1.54 km Score: 61.77	Springvale Estimated Distance: 4.05940792 km Median Rent: A\$270 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.23 km Closest Cinema: 3.13 km Score: 61.16
---	--	--	--	---	--

The output is **the same as we expected**, as it triggers Step 6b, where, in this particular case, retrieves 0 suburbs from the only-filtered suburbs, and takes 5 suburbs with the highest score from the whole database without considering the user filter. The print statements show that all parts of the code used are based on what we expect, and the result outputs have some red highlights in the figure, as it is not within what the user input wants.

If we pay attention more closely, Test 1 and Test 2 have the same exact results for all 5 suburbs. The reason this happens is because for test 1 the user input the maximum allowed for the variables (rent, distance, etc) which would filter a lot of results, while in the second test the user input the bare minimum, where it would not have any suburb in the filtered-list. Therefore, since both test 1 and test 2 still sort the result based on scores, it shows the same result.

### Test Number 3 (Result from both Filtered List and Scoring-only List)

Input:

**Choose University**

**Maximum Average Weekly Rent (150-500)**

**Max Distance (1-100)**

**Cinema(YES or NO)**

**Hospital(YES or NO)**

**Search**

Print Output Statements:

```
Step 2.a.2 (Cinema is NO)
Step 2.b.1 (Hospital is YES)
Step 3, Retrieving filtered suburbs
Step 4, Retrieving all suburbs
Step 5, Score Calculation for Filtered Results
Step 5, Score Calculation for All Results
Step 6b (Filtered results < 5)
['Burwood', 'Noble Park', 'Clayton', 'Ashburton', 'Hughesdale']
```

Output:

<b>Campus:</b> Monash University Clayton Campus <b>Max Distance:</b> 10 km <b>Cinema:</b> exclude <b>Hospital:</b> include <b>Rent:</b> 260				
Burwood	Noble Park	Clayton	Ashburton	Hughesdale
Estimated Distance: 7.540468984 km Median Rent: A\$240 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.94 km Closest Cinema: 1.71 km Score: 62.28	Estimated Distance: 6.847565633 km Median Rent: A\$250 Number of Hospital: 3 Number of Cinema: 0 Closest Hospital: 0.68 km Closest Cinema: 4.2 km Score: 59.92	Estimated Distance: 0.538091383 km Median Rent: A\$280 Number of Hospital: 6 Number of Cinema: 1 Closest Hospital: 0.52 km Closest Cinema: 1.22 km Score: 66.11	Estimated Distance: 7.290105785 km Median Rent: A\$240 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 1.49 km Closest Cinema: 2.04 km Score: 62.44	Estimated Distance: 5.049328688 km Median Rent: A\$280 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 2.95 km Closest Cinema: 1.54 km Score: 61.77

The output is **the same as we expected**, as it triggers Step 6b, where, in this particular case, retrieves 2 suburbs from the only-filtered suburbs, and takes 3 suburbs with the highest score from the whole database without considering the user filter. The print statements show that all parts of the code used are based on what we expect, and the result outputs have some red highlights on the last 3 results. The reason is because the first 2 results ( Burwood and Noble Park ) are taken from the filtered-only list, meanwhile the last 3 results ( Clayton, Ashburton, and Hughesdale) are taken from the all-suburb list. Although Clayton has a higher score than Burwood and Noble Park, it would still be third on the result as the Median Rent is still more than the user input and was taken from the all-suburb list, not the filtered-only list.

Result:

Based on the tests above, it shows that all the conditions and scenarios run normally, as well as all the steps are being followed, based on the print statements. Therefore, the result of the tests is PASS, since it correctly transverses the code and shows expected results.

## API Test

There are 3 APIs we are using on the website. There are Google's Directions API, Places API, and Google Map API. As the Map and Places API results are able to be clearly seen in the website, I am testing the Directions API that we are using to make sure the information is given correctly.

The way we are using the API is by using the URL to call the API while changing the origin and destination of the Direction's API. The departure time is also changing, however because it is dynamically changing, we are unable to do a testing for the departure time. The way we test the API is by directly comparing the results from API requests to Google Map at the same time.

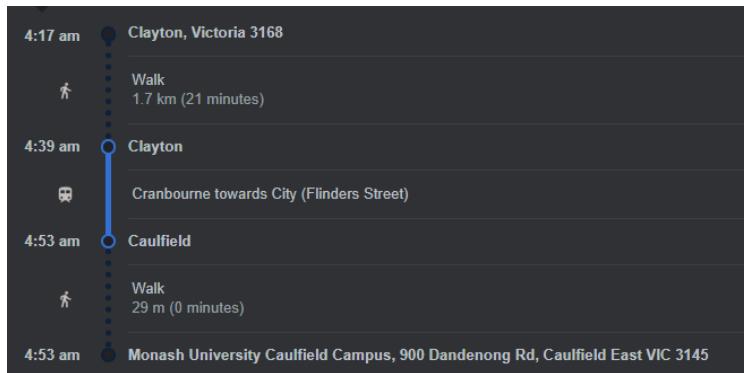
Test Number 5:

Input:

Origin: Clayton, VIC

Destination: Monash University Caulfield Campus

Expected Output:



Output:

```
{"departure_time": {  
    "text": "04:17",  
    "time_zone": "Australia/Melbourne",  
    "value": 1665940676  
},  
  
"arrival_time": {  
    "text": "04:53",  
    "time_zone": "Australia/Melbourne",  
    "value": 1665942803  
  
},  
"distance": {  
    "text": "10.7 km",  
    "value": 10651  
},  
"duration": {  
    "text": "35 mins",  
    "value": 2127  
}.
```

Test Number 6:

Input:

Origin: Fitzroy, VIC

Destination: Monash University Clayton Campus

Expected Output:



Output:

```
"arrival_time" : {
    "text" : "06:07",
    "time_zone" : "Australia/Melbourne",
    "value" : 1665947279
},
"departure_time" : {
    "text" : "04:49",
    "time_zone" : "Australia/Melbourne",
    "value" : 1665942560
},
"distance" : {
    "text" : "23.1 km",
    "value" : 23076
},
"duration" : {
    "text" : "1 hour 19 mins",
    "value" : 4719
},
```

Result:

PASS. The expected output is the same with the output. The API took the fastest way from origin to destination from a few options to travel with public transport. Furthermore, as expected, if public transport is unavailable at the chosen time of departure, it will change the departure time to the closest time when the public transport is available.

## Database Retrieval Test:

We are using SQLite database, embedded inside the project file, where we retrieve information to show to users on our website. Multiple queries are done to extract the suburbs based on user filters, as well as to calculate scores for each suburb and to rank them.

To make sure the database query is working, tests will be conducted by querying the database based on name, rent, and number of hospitals.

### Test Number 7: Suburb Query

Input: Clayton or Fitzroy or Melbourne or South Yarra

```
'select * from v3 where "Suburb/Town Name" in ("Clayton", "Fitzroy", "Melbourne", "South Yarra")'
```

### Expected Output:

It will print out information on 4 suburbs above.

### Output:

```
('Clayton', 3168, 12212, 19358, -37.91526, 145.12899)
('Fitzroy', 3065, 20256, 10445, -37.80092, 144.97919)
('Melbourne', 3000, 155759, 47285, -37.82744, 144.97142)
('South Yarra', 3141, 20185, 25147, -37.83822, 144.99138)
```

### Test Number 8: Rent Query

Input: Suburbs that has median rent less than A\$225

```
'select * from v3 where "Median Rent" < 225'
```

### Expected Output:

It will print out information about suburbs that has median rent less than A\$225

### Output:

```
('Alfredton', 3350, 3514, 9220, -37.55517, 143.80089, 0, 0, 190)
('Bairnsdale', 3875, 12488, 7580, -37.85346, 147.58099, 1, 1, 220)
('Benalla', 3672, 8214, 10330, -36.57111, 146.00681, 1, 0, 220)
('Corio', 3214, 18272, 15296, -38.06752, 144.37728, 0, 0, 223)
('Echuca', 3564, 14216, 14043, -36.15495, 144.75711, 1, 1, 200)
('Flora Hill', 3558, 1716, 3955, -36.78252, 144.296, 0, 0, 200)
('Golden Square', 3555, 5198, 8820, -36.775, 144.25141, 0, 0, 195)
('Hamilton', 3300, 7273, 9974, -37.74637, 142.01476, 1, 1, 180)
('Kangaroo Flat', 3555, 7550, 10394, -36.80627, 144.243, 0, 0, 195)
('Maffra', 3860, 2527, 5280, -37.9639, 146.99692, 0, 0, 200)
('Mildura', 3500, 39397, 32738, -34.21084, 142.1337, 1, 0, 210)
('Moe', 3825, 16103, 8778, -38.1751, 146.24948, 0, 0, 180)
('Morwell', 3840, 30423, 13771, -38.23174, 146.42003, 1, 1, 188)
('Newborough', 3825, 5208, 6763, -38.17746, 146.29389, 0, 0, 180)
('North Bendigo', 3550, 2434, 4059, -36.73733, 144.27886, 0, 0, 223)
('Sale', 3850, 15960, 13673, -38.10695, 147.07765, 1, 1, 200)
('Seymour', 3660, 8636, 6327, -37.03117, 145.16129, 1, 0, 200)
('Shepparton', 3630, 40284, 31197, -36.37339, 145.39239, 2, 1, 215)
('Swan Hill', 3585, 11691, 10905, -35.36213, 143.52343, 1, 1, 170)
('Traralgon', 3844, 27955, 24933, -38.20169, 146.51447, 0, 1, 210)
('Wendouree', 3355, 11897, 10445, -37.5389, 143.82592, 0, 0, 190)
('Wodonga', 3690, 17099, 18948, -36.12578, 146.88243, 2, 0, 200)
```

We could see from the last output of every suburb, which represents the median rent, where all the values are less than A\$225.

### Test Number 9: Hospital Query

Input: Suburbs that have more than 2 hospitals in it.

```
'select * from v3 where "Number of Hospitals" >2'
```

Expected Output:

It will show information for suburbs that have more than 2 hospitals inside the suburb.

Output:

```
('Bendigo', 4)
('Berwick', 4)
('Box Hill', 4)
('Brighton', 3)
('Bundoora', 4)
('Caulfield', 3)
('Clayton', 6)
('Dandenong', 5)
('East Melbourne', 7)
('Fitzroy', 3)
('Footscray', 6)
('Frankston', 8)
('Geelong', 5)
('Heidelberg', 5)
('Kew', 6)
('Malvern', 4)
('Melbourne', 6)
('Noble Park', 3)
('Parkville', 7)
('Richmond', 4)
('St Albans', 3)
('Warrnambool', 3)
('Werribee', 3)
```

It shows the list of suburbs that have more than 2 hospitals. Only the suburb name and its number of hospital information are shown for simplicity of the output.

Result:

PASS. Based on the tests above, it shows that the database and the query are working perfectly. All of the outputs are the same with the expected results. Therefore, we could conclude that all query results done in the code from the SQLite database for the website are working perfectly.

# Black Box Testing

In this testing part, all the features and functionalities in the website will be tested without having to know much about how the code runs. In this section, we will ensure all the basic functions in a page are tested separately (Black Box Testing) and furthermore testing This test will ensure all the basic functionality/features are working as expected by focusing more on whether the inputs required pre-processing and the outputs which need to produce the correct results.

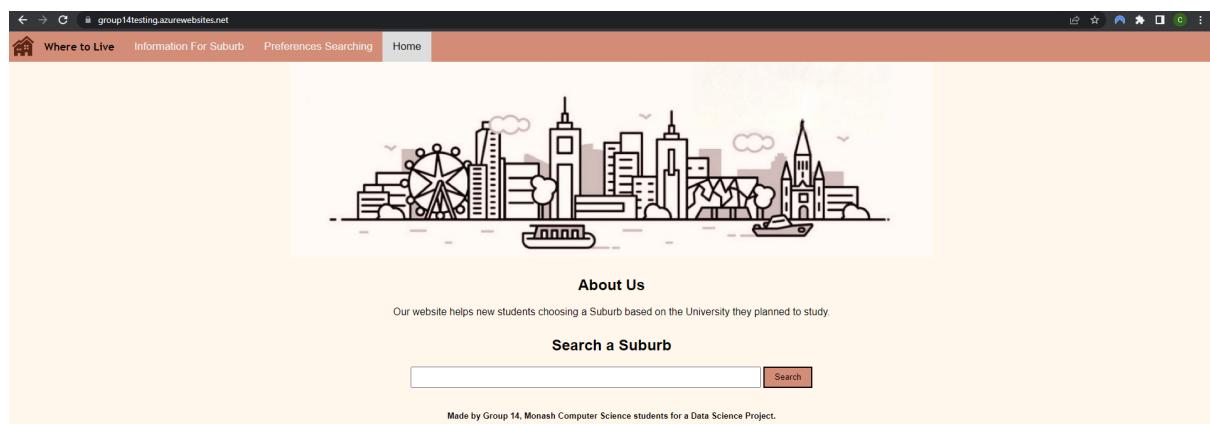
## Test Number 1 [Navigation Bar]

The first test is checking that the navigation bar works correctly which is required to redirect into different features in the website. The user will have to click the options and should successfully navigate to the corresponding page. The website will register the user's click on the certain text as the input and display the correct page for the output. This feature mainly works as a redirection tool and therefore, there will be four subtests that ensure redirection to each of the existing pages.

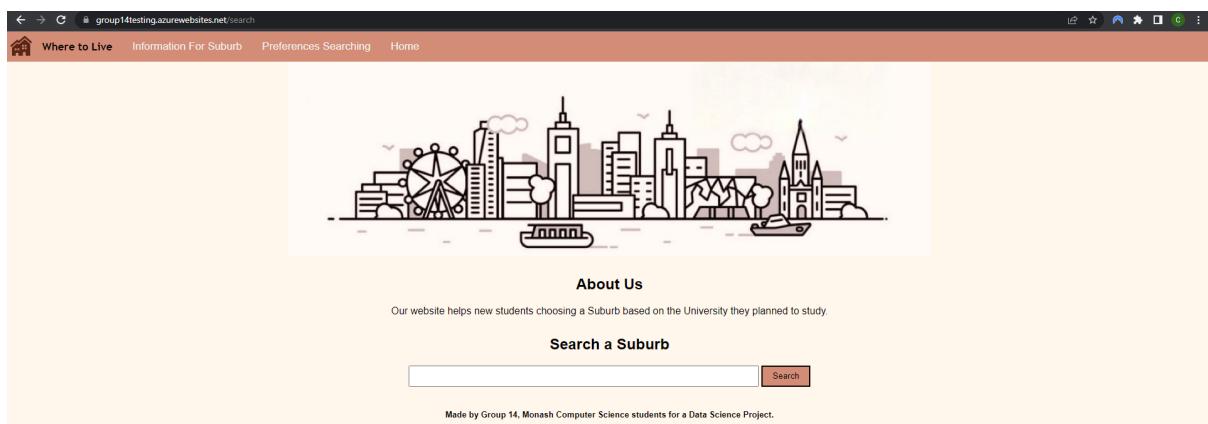
### Test 1.1 Home Redirection

This test aims that the redirection to the home page could happen by pressing the home button in the navigation bar.

Input:



## Output:



## Expected Output:

The expected output is directed to the home page by the '/search' path name (Home page) which is shown in the actual output.

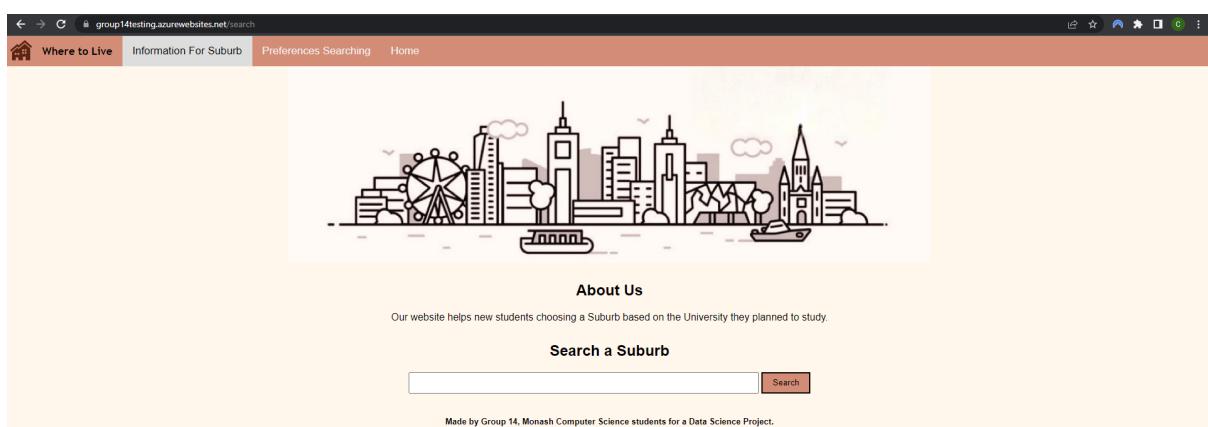
## Test Status:

PASS

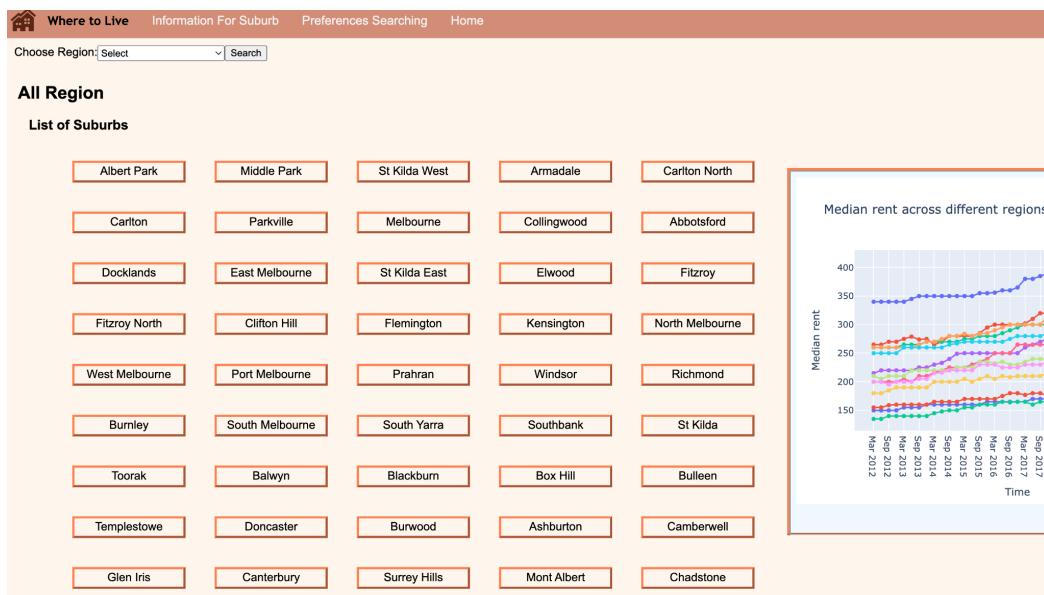
## Test 1.2 Information For Suburb Redirection

This test aims that the redirection back to the 'Information For Suburb' page could happen by pressing the 'Information For Suburb' button in the navigation bar.

## Input:



## Output:



### Expected Output:

The expected output is directed back to the information page by the '/search\_all' path name (Information For Suburb) which is shown in the actual output.

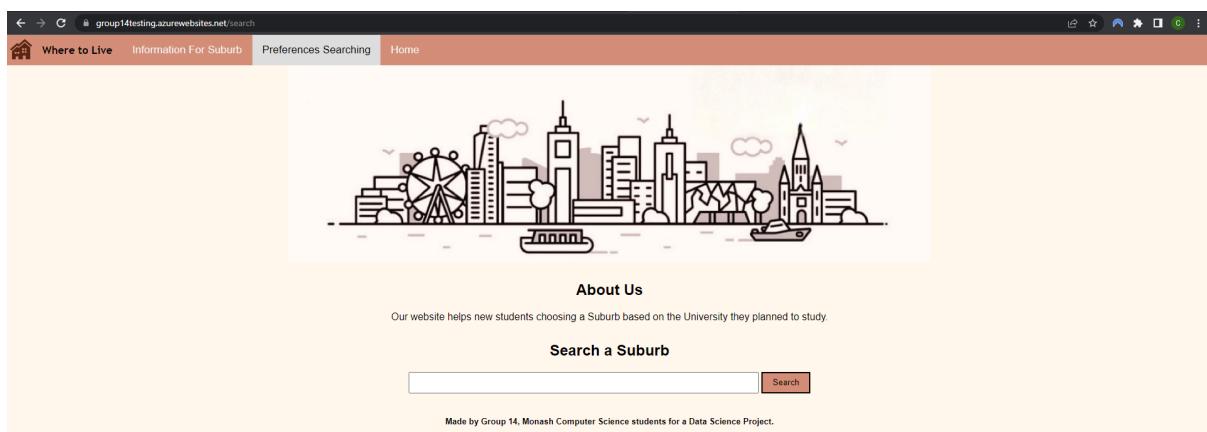
### Test Status:

PASS

### Test 1.3 Preferences Searching Redirection

This test aims that the redirection to the 'Preferences Searching' page could happen by pressing the 'Preferences Searching' button in the navigation bar.

### Input:



### Output:

Where to Live Information For Suburb Preferences Searching Home

Preferences Searching

Choose University

Select

Maximum Average Weekly Rent (150-500)

Maximum Rent Price

Max Distance (1-100)

Maximum Distance

Cinema(YES or NO)

Include Cinema?

Hospital(YES or NO)

Include Hospital?

Search

Expected Output:

The expected output is directed back to the preference searching page which is the '/filter' path name (Preferences Searching) which is shown in the actual output.

Test Status:

PASS

## Test Number 2 [Home Page]

This test is checking that the search feature works correctly in the home page which is required to display lists of suburbs according to the user's input. By entering valid and invalid suburbs names, the page should be able to show names of the suburbs or empty result page (if input doesn't match with any suburb in the database). The website will register the user's text written in the field as input and display the suburbs that match the substring of the user's input. In this section, we will also ensure that the redirection of the path after a certain action is accurate. There will be two main tests that will be checked in the home page testing.

### Test 2.1 Home Search Button Redirection

This test will ensure that the redirection to the 'List of Suburbs' page will happen by pressing the 'Search' button after the user's input is filled.

Input:

The screenshot shows a web page titled 'About Us'. Below the title, a sub-section titled 'Search a Suburb' contains a search bar with the word 'Clayton' typed into it. To the right of the search bar is a brown 'Search' button. At the bottom of the page, there is a footer note: 'Made by Group 14, Monash Computer Science students for a Data Science Project.'

Output:

The screenshot shows a 'Search Result' page for 'Clayton (3168)'. The page displays the following information in a box:

- Median Rent: 280
- Number of Hospital: 6
- Number of Cinema: 1

Expected Output:

The expected output is the redirection to a new page which is the search result page of the list of suburbs.

Test Status:

PASS

## Test 2.2 Suburb Substring Match Search

This test will ensure that the list of suburbs that is displayed will match with the substring that is inputted by the user.

Input 1:

The screenshot shows a web page with a light beige background. At the top center, the text "About Us" is displayed in bold black font. Below it, a paragraph of text reads: "Our website helps new students choosing a Suburb based on the University they planned to study." In the center, there is a section titled "Search a Suburb" in bold black font. Below this title is a search input field containing the text "Python". To the right of the input field is a brown rectangular button with the word "Search" in white. At the bottom of the page, a footer line of text states: "Made by Group 14, Monash Computer Science students for a Data Science Project."

Output 1:

The screenshot shows a web page with a light beige background. At the top, there is a navigation bar with a brown header. From left to right, the menu items are: a house icon, "Where to Live", "Information For Suburb", "Preferences Searching", and "Home". Below the navigation bar, the text "Search Result" is centered. The rest of the page is blank, indicating no search results have been displayed.

Expected Output:

The expected output is showing no suburbs since there are no suburbs with the name ‘Python’. Here, it is shown that the result of the search is an empty search result which reveals that this case also passed the test.

Input 2:

About Us

Our website helps new students choosing a Suburb based on the University they planned to study.

Search a Suburb

Melbourne

Search

Made by Group 14, Monash Computer Science students for a Data Science Project.

Output 2:

Document

Where to Live Information For Suburb Preferences Searching

Search Result

East Melbourne (3002)

Median Rent: 360  
Number of Hospital: 7  
Number of Cinema: 0

Melbourne (3000)

Median Rent: 300  
Number of Hospital: 6  
Number of Cinema: 8

North Melbourne (3051)

Median Rent: 310  
Number of Hospital: 0  
Number of Cinema: 0

Port Melbourne (3207)

Median Rent: 400  
Number of Hospital: 0  
Number of Cinema: 0

Expected Output:

The expected output is showing suburbs that have the name ‘Melbourne’. Here, it is shown that the result of the search is various suburbs which are ‘East Melbourne’, ‘Melbourne’, ‘North Melbourne’, etc. This case also passes the test since most of the results are suburbs that contain the substring ‘Melbourne’.

Test Status:

PASS

## Test 2.3 Preferences Searching Redirection

This test will ensure that the redirection to the ‘Preferences Searching’ page will happen by clicking on one of the suburb box buttons from the list of suburbs page.

Input:

Search Result

**Carnegie (3163)**

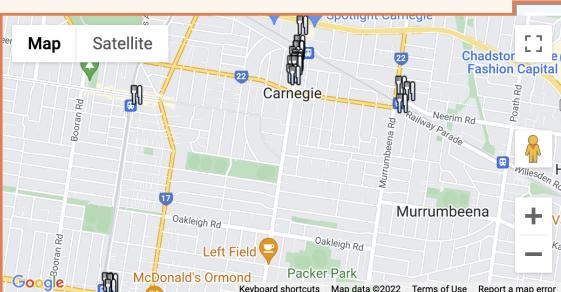
Median Rent: 290

Number of Hospital: 0

Number of Cinema: 0

Output:

Suburb Information  
Carnegie (3163)



**Conditions**

Average Weekly Rent	: A\$290
Average Annual Crime	: 700 Incidents
Distance to CBD	: 13 km

Expected Output:

The expected output is the redirection to a new page which is the suburb information page of the one of the selected suburbs on the list of suburbs page. The actual output follows the expected output which means that this case also passes the test.

Test Status:

PASS

## Test Number 3 [Suburb Information Page]

This test will ensure all functionality in the Information Suburb Page works according to the requirements. For this page, showing a list of suburbs to select based on region and path redirection after clicking on certain suburbs names have to be checked. Therefore, there will be two types of testing on this page.

### Test 3.1 Testing Drop Down List

This test aims that the page will be able to show all suburbs that are in a particular region which is chosen according to the user's selection within the drop down list .

Input:

The image contains two side-by-side screenshots of a web application interface. Both screenshots feature a header with a house icon, the text 'Where to Live', and 'Information For Suburb'. Below the header is a search bar labeled 'Choose Region' with a dropdown menu. In the top screenshot, the dropdown is set to 'Bendigo' and there is a 'Search' button next to it. In the bottom screenshot, the dropdown is set to 'Inner Eastern Melbourne' and there is also a 'Search' button next to it.

Output:

The image contains three screenshots of the 'Where to Live' website under the 'Information For Suburb' section. Each screenshot has a 'Choose Region' dropdown at the top set to 'Select' with a 'Search' button. The first screenshot is for 'Bendigo', showing suburbs like Bendigo, Flora Hill, Golden Square, Kangaroo Flat, and North Bendigo. The second screenshot is for 'All Region', showing a larger grid of suburbs including Abbotsford, Albert Park, Alfreton, Alphington, Altona, Armadale, Ascot Vale, Ashburton, Aspendale, Avondale Heights, Bairnsdale, Balwyn, Beyswater, Beaumaris, Belmont, Benalla, Bendigo, Bentleigh, Berwick, Blackburn, Boronia, Box Hill, Brighton, Broadmeadows, Brunswick, Bulleen, Bundoora, Burnley, Burwood, Burwood East, Camberwell, Canterbury, Carlton, Carlton North, Carnegie, Clayton, Doncaster, Doncaster East, Donvale, Forest Hill, Glen Iris, Glen Waverley, Hawthorn, Kew, Mitcham, Mont Albert, Mount Waverley, Mulgrave, Nunawading, Oakleigh, and Swan Hill. The third screenshot is for 'Other Regional Centres', showing suburbs like Bairnsdale, Benalla, Echuca, Hamilton, Horsham, Maffra, Mildura, Moe, Morwell, Newborough, Portland, Sale, Seymour, Shepparton, Swan Hill, Torquay, Traralgon, Warrnambool, and Wodonga.

## Expected Output:

The expected output is displaying a group of suburbs that matches with the current selected region. With these four outputs, the drop down list functions accordingly by showing all suburbs according to their regions. In conclusion, the drop down functionality passed the test.

## Test Status:

PASS

## Test 3.2 Testing Suburb Redirection

This test aims that the page will be able to show a particular suburb's information according to the user's selection by clicking one of the displayed suburbs.

## Input:

A screenshot of a web application interface. At the top left is a dropdown menu labeled "Choose Region: Select" with a search bar next to it. Below this, the title "Inner Eastern Melbourne" is displayed. Underneath the title is a heading "List of Suburbs". A grid of 20 suburb names is shown in orange-bordered boxes:

Ashburton	Balwyn	Blackburn	Box Hill	Bulleen
Burwood	Burwood East	Camberwell	Canterbury	Chadstone
Clayton	Doncaster	Doncaster East	Donvale	Forest Hill
Glen Iris	Glen Waverley	Hawthorn	Kew	Mitcham
Mont Albert	Mount Waverley	Mulgrave	Nunawading	Oakleigh

## Output:

A screenshot of the "Where to Live" website showing the suburb information for Balwyn (3103). The top navigation bar includes links for "Where to Live", "Information For Suburb", "Preferences Searching", and "Home". The main content area has a header "Suburb Information" and "Balwyn (3103)". It features two maps: a "Map" view showing local landmarks like Boroondara Sports Complex and Maranoa Botanic Gardens, and a "Satellite" view showing the same area from above. Below the maps is a "Conditions" section with tables for "Average Weekly Rent", "Average Annual Crime", "Distance to CBD", "Population", "Number of Hospitals", "Closest Hospital", "Number of Cinemas", and "Closest Cinema". To the right is a "Suburb Information" section for Mulgrave (3170), which includes a map of the suburb, its location relative to the CBD, and various demographic and facility statistics.

Condition	Value
Average Weekly Rent	: A\$340
Average Annual Crime	: 422 Incidents
Distance to CBD	: 11 km
Population	: 13312 people
Number of Hospitals	: 0
Closest Hospital	: 2.02 km
Number of Cinemas	: 1
Closest Cinema	: 0.7 km

Condition	Value
Average Weekly Rent	: A\$355
Average Annual Crime	: 1069 Incidents
Distance to CBD	: 23 km
Population	: 19368 people
Number of Hospitals	: 1
Closest Hospital	: 2.61 km
Number of Cinemas	: 0
Closest Cinema	: 2.84 km

The screenshot shows the 'Where to Live' website interface. At the top, there is a navigation bar with links for 'Where to Live', 'Information For Suburb', 'Preferences Searching', and 'Home'. Below the navigation bar, the page title is 'Suburb Information' followed by 'Clayton (3168)'. The main content area features a map of Clayton, Victoria, showing the location of Monash University Clayton Campus, M-City Shopping Centre, and Monash Medical Centre. The map also includes major roads like the Monash Freeway (M1) and local streets like Valley St and Houghton Rd. Below the map, a section titled 'Conditions' displays the 'Average Weekly Rent' as 'A\$280'. At the bottom of the page, there are links for 'Keyboard shortcuts', 'Map data ©2022', 'Terms of Use', and 'Report a map'.

### Expected Output:

The expected output is a redirection to the suburb information page where it displays a suburb's information and another feature to show directions to a certain university from that particular suburb. In this case, the actual output showed exactly each suburb information page correctly and also passed the test.

### Test Status:

PASS

## Test Number 4 [Preferences Searching Page]

This test will ensure all functionality in the Preferences Searching Page works according to the requirements. For this page, user's input validation and path redirection after searching on certain user's selection. Therefore, there will be two types of testing on this page.

### Test 4.1 Input Validation

This test aims that the page will be able to limit the user's option of entering invalid inputs (such as an out of range value and not whole numbers) in all different possible scenarios.

#### Input 1:

The image contains two side-by-side screenshots of a web application titled "Preferences Searching". Both screenshots show the same set of input fields: "Choose University" (dropdown menu showing "Monash University Clayton Campus"), "Maximum Average Weekly Rent (150-500)" (input field containing "600" with a dropdown arrow), "Max Distance (1-100)" (input field containing "45"), "Cinema(YES or NO)" (input field containing "YES"), and "Hospital(YES or NO)" (input field containing "YES"). A "Search" button is located at the bottom of each form.

#### Output 1:

The image contains two side-by-side screenshots of the "Preferences Searching" page after the user has submitted the form. The first screenshot shows an error message for the "Max Distance" field: "Value must be less than or equal to 500." The second screenshot shows an error message for the "Maximum Average Weekly Rent" field: "Value must be greater than or equal to 150."

#### Expected Output:

A warning will pop up if the user's input value does not meet the criteria or is out of range and is expected to be output in the Average Rent input field.

Input 2:

**Preferences Searching**

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
300.1

Max Distance (1-100)  
45

Output 2:

**Preferences Searching**

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
300.1

 Please enter a valid value. The two nearest valid values are 300 and 301.

Expected output:

A warning will pop up reminding the user to re-enter a value that is a whole number and give suggestions of the two nearest valid values between the ‘invalid’ number in the Average Rent input field.

Input 3:

**Preferences Searching**

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
400

Max Distance (1-100)  
101

**Preferences Searching**

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
400

Max Distance (1-100)  
-20

Output 3:

**Preferences Searching**

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
400

Max Distance (1-100)  
101

Cinema(YES)  Value must be less than or equal to 100.  
YES

Maximum Average Weekly Rent (150-500)  
400

Max Distance (1-100)  
-20

Cinema(YES)  Value must be greater than or equal to 1.  
YES

Expected Output:

A warning will pop up if the user’s input value does not meet the criteria or is out of range and is expected to be output in the Max Distance input field.

#### Input 4:

The screenshot shows the 'Preferences Searching' page with the following input fields:

- Choose University: Monash University Clayton Campus
- Maximum Average Weekly Rent (150-500): 350
- Max Distance (1-100): 8
- Cinema(YES or NO): abc
- Hospital(YES or NO): YES

A red 'Search' button is at the bottom.

The screenshot shows the 'Preferences Searching' page with the following input fields:

- Choose University: Monash University Clayton Campus
- Maximum Average Weekly Rent (150-500): 350
- Max Distance (1-100): 8
- Cinema(YES or NO): YES
- Hospital(YES or NO): ekd

A red 'Search' button is at the bottom.

#### Output 4:

The screenshot shows a browser window with the address bar: group14testing.azurewebsites.net/group14testing. The page title is 'Where to Live'. A modal dialog box is displayed with the message: "group14testing.azurewebsites.net says Please Input either YES or NO for Cinema". An 'OK' button is at the bottom right of the dialog.

The main content area shows the 'Preferences Searching' form with the same inputs as the previous screenshot, except for the Cinema field which now contains 'abc'. A red 'Search' button is at the bottom.

#### Expected Output:

There will be an alert pop-out from the browser that tells the user to re-enter the input which is only restricted to 'yes' and 'no' (case insensitive).

#### Test 4.2 Path Redirection to Top Suburb Search Result Page

This test aims that the page will be able to redirect to the page that reveals the top suburbs according to the valid user input.

Input:

### Preferences Searching

Choose University  
Monash University Clayton Campus

Maximum Average Weekly Rent (150-500)  
350

Max Distance (1-100)  
8

Cinema(YES or NO)  
YES

Hospital(YES or NO)  
YES

**Search**

Output:

Where to Live Information For Suburb Preferences Searching Home

### Search Result

Campus:Monash University Clayton Campus  
Max Distance: 8 km  
Cinema: include  
Hospital: include  
Rent:350

Clayton	Mount Waverley	Ashburton	Burwood
Estimated Distance: 0.538091383 km Median Rent: A\$280 Number of Hospital: 6 Number of Cinema: 1 Closest Hospital: 0.52 km Closest Cinema: 1.22 km Score: 66.11	Estimated Distance: 4.046872088 km Median Rent: A\$338 Number of Hospital: 2 Number of Cinema: 2 Closest Hospital: 1.6 km Closest Cinema: 1.95 km Score: 57.4	Estimated Distance: 7.290105785 km Median Rent: A\$240 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 1.49 km Closest Cinema: 2.04 km Score: 62.44	Estimated Distance: 7.540468984 km Median Rent: A\$240 Number of Hospital: 1 Number of Cinema: 0 Closest Hospital: 1.94 km Closest Cinema: 1.71 km Score: 62.28

Expected Output:

Redirects to the top suburbs search results page where it contains suburb recommendations based on the factors that were prompted and chosen by the users.

Test Status:

PASS

### Test 4.3 Path Redirection to Suburb Information

This test aims that the page will be able to show the particular suburb's information according to the previous user's preferences' choices by clicking one of the displayed suburbs on the top result.

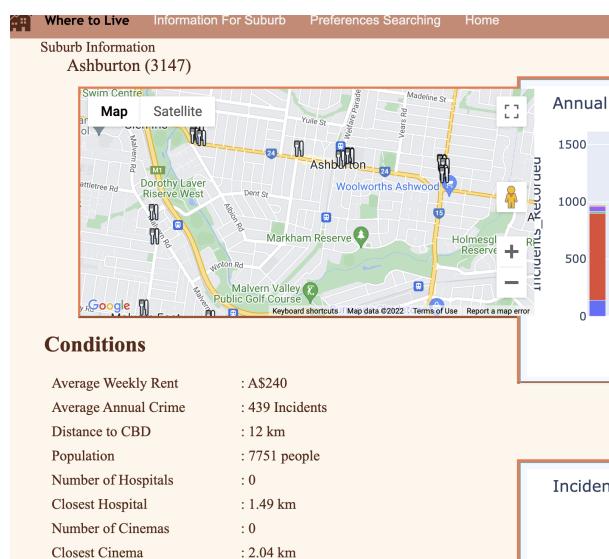
Input:

## Search Result

**Campus:**Monash University Caulfield Campus  
**Max Distance:** 5 km  
**Cinema:** exclude  
**Hospital:** exclude  
**Rent:**270

Ashburton	Carnegie	Murrumbeena	Caulfield
Estimated Distance: 3.081860492 km Median Rent: A\$240 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 1.49 km Closest Cinema: 2.04 km Score: 67.91	Estimated Distance: 2.078248608 km Median Rent: A\$280 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 2.35 km Closest Cinema: 2.66 km Score: 64.76	Estimated Distance: 3.010661337 km Median Rent: A\$280 Number of Hospital: 0 Number of Cinema: 0 Closest Hospital: 2.77 km Closest Cinema: 1.84 km Score: 64.67	Estimated Distance: 1.783184381 km Median Rent: A\$300 Number of Hospital: 3 Number of Cinema: 0 Closest Hospital: 0.55 km Closest Cinema: 2.27 km Score: 63.99

## Output:



## Expected Output:

The expected output is that there is a page redirection to the suburb information page where it shows the information of the clicked suburb which in this case is Ashburton. Matching with the expected output, the actual output proves that this case passes the test.

## Test Status:

PASS

## Test Number 5 [Travel Direction Calculation Feature]

This test will ensure that the feature shows clear directions from the selected suburb to a certain university depending on the user's selection. For this page, we will be testing whether the functionality of showing direction works accurately or not according to google maps.

Also, we will be testing whether in all possible cases that erroneous user inputs will be handled. There will be two types of tests on this feature.

## Test 5.1 User Input Handling

Input:

**Public Transport travel**

Choose University	<input type="text" value="University of Melbourne Southbank Ca"/>
Departure (date and time):	<input type="text" value="15/12/2021, 11:11 am"/> <input type="button" value="Search"/>
Destination:	:
Travel Time:	:
Travel Distance:	:
Best Departure Time	:
Arrival Time:	:
Number of Changes:	:
Directions:	

Output:

**Public Transport travel**

Choose University	<input type="text" value="University of Melbourne Southbank Ca"/>
Departure (date and time):	<input type="text" value="15/12/2021, 11:11 am"/> <input type="button" value="Search"/>
Destination:	 Value must be 15/10/2022, 11:11 am or later.
Travel Time:	:
Travel Distance:	:
Best Departure Time	:
Arrival Time:	:
Number of Changes:	:
...	

Expected Output:

There will be a warning pop out close to the field that suggests the user to choose a date after or the same as current time. Users also cannot choose the time before the current time.

## Test 5.2 Direction Accuracy Testing

Input:

**Public Transport travel**

Choose University	Monash University Caulfield Campus
Departure (date and time):	16/10/2022, 11:53 am
Destination:	:
Travel Time:	:
Travel Distance:	:
Best Departure Time	:
Arrival Time:	:
Number of Changes:	:
Directions:	

Output:

**Public Transport travel**

Choose University	Select
Departure (date and time):	15/10/2022, 11:54 am
Destination:	: Monash University Caulfield Campus
Travel Time:	: 35 mins
Travel Distance:	: 10.7 km
Best Departure Time	: 11:04pm
Arrival Time:	: 11:40pm
Number of Changes:	: 0
Directions:	

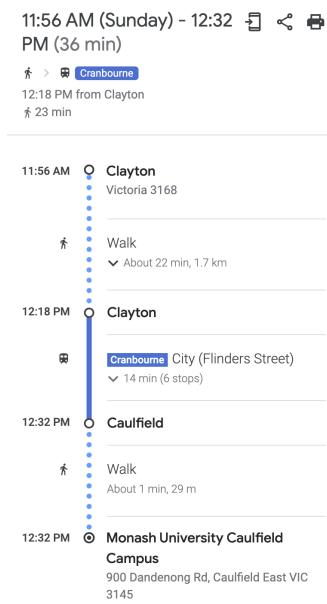
'Walk to Clayton,1.7 km,21 mins'

'Train towards City (Flinders Street), 8.7 km, 14 mins, from Clayton to Caulfield'

'Walk to 900 Dandenong Rd, Caulfield East VIC 3145, Australia,0.2 km,1 min'

Expected Output:

All the empty fields will be filled correctly according to the google API (google maps).



## Integration Testing

In this particular part of the testing, we will be commencing a group of black box tests to show that the functionality of this website service does not only work separately but it will appropriately function as a whole. For the table on this testing, the column ‘test’ shows the various black box tests that are run to ensure that the particular integration test is working. Here are the test scenarios.

No.	Test Description	Input	Test	Expected Output	Actual Output	Test Status
1.1	Suburb Selector Features Testing from Home Page using Search	Valid Input	Test 1.1 -> Test 2.1 -> Test 2.3	Redirection to the Suburb Selector page and showing top suburbs.	End Redirection: Suburb Information Page  Before Redirection: Top 5 Suburbs Page	PASS
		Invalid Input [Invalid Suburb Name for Searching]	Test 2.2	Redirection to the Suburb Selector however showing empty result	End Redirection: Empty Suburb Information Page Similar to Black Box Test 2.2	PASS
1.2	Suburb Selector Features Testing from Information For Suburb	Valid Input	Test 1.2 -> Test 3.1 -> Test 3.2	Redirection to the Suburb Selector page and showing top suburbs.	End Redirection: Suburb Information Page  Before Redirection: Top 5 Suburbs Page	PASS
1.3	Recommendation Suburb	Valid Input	Test 1.3 -> Test 4.1 ->	Redirection to the Suburb	End Redirection:	PASS

	Selector Features From Preferences Searching		Test 4.2 -> Test 4.3	Selector page and showing top suburbs.	Suburb Information Page  Before Redirection: Top 5 Suburbs Page	
		Invalid Input [Invalid Suburb Name for Searching]	Test 4.1	There will be a warning pop-up in the fields where the invalid input is entered.	Similar to Integration Test 1.1	PASS
2.1	Direction Features Testing from Home Page using Search	Valid Input	Continuation from Integration Test 1.1 -> Test 5.2	Showing steps of direction from the searched suburbs to a particular university	Steps from selected suburb to the selected university which should match google maps direction Similar to Black Box Test 5.2	PASS
		Invalid Input [Invalid Depart Time]	Test 5.1	There will be a warning pop-up in the fields of the departure time.	Stay on the same page. It doesn't redirect after user input is valid. Similar to Black Box Test 5.1	PASS
2.2	Direction Features From Information For Suburb	Valid Input	Continuation from Integration Test 1.2 -> Test 5.2	Showing steps of direction from the clicked (selected)	Steps from selected suburb to the selected university which	PASS

				suburbs to a particular university	should match google maps direction Similar to Black Box Test 5.2	
		Invalid Input [Invalid Depart Time]	Test 5.1	There will be a warning pop-up in the fields of the departure time.	Similar to Integration Test 2.1 Actual Output	PASS
2.3	Direction Features From Preferences Searching	Valid Input	Continuation from Integration Test 1.3 -> Test 5.2	Showing steps of direction from the selected suburbs from the top recommended one to a particular university	Steps from selected suburb to the selected university which should match google maps direction Similar to Black Box Test 5.2	PASS
		Invalid Input [Invalid Depart Time]	Test 5.1	There will be a warning pop-up in the fields of the departure time.	Similar to Integration Test 2.1 Actual Output	PASS

## System Testing

This testing ensures that all the product user acceptance criteria are accepted properly to show that the website product is ready to deploy and user-ready.

No.	Product User Acceptance Criteria	Test Status	Additional Notes
1.	Website suggests the suburbs accurately according to users	PASS	The algorithm shows accurate selection to user's based on their preferences and also displays extra information which are the factors to a good suburb selection.
2.	Each suburb informations are accurate by using datasets that are reliable	PASS	The suburb's information is retrieved from various different data sources which are verified and trusted.
3.	Map services display map location precisely and show few relevant landmarks	PASS	With Google API, features using maps to show location are implemented accordingly. Also with other APIs, landmarks such as hospitals and cinemas are also shown within the map. [Referring to Black Box Testing]
4.	Correct information successfully retrieved from database	PASS	Referring to White Box Testing 7-9, it is shown that the database retrieves all the information accurately.
5.	UI is friendly, easy to use, attractive and appealing	PASS	Referring to the usability testing, the overall score for the UI rating page is very significant.
6.	Fast performance in terms of speed and resource used	PASS	The performance for this website is sufficient for users to access. Not too long loading time since resources are managed efficiently in the backend.

## Usability Testing

This testing ensures the quality of this product from the front-end design perspective. This test's result will be mainly from the feedback of the users. We will be giving tasks in order to gain insights about how well is our UI and in return, the feedback will be observed thoroughly as the result of the usability testing.

Question/Prompt	Average Rating (of 5 users) / Combined Responses
How convenient is it to redirect pages?	24/25 Comments: 1. Redirecting Pages is easy because of the tool bar 2. Most pages directly redirects to another one
How readable is the text on each page? [Is the font readable with any device?]	21/25 Comments: 1. Most pages are easy to read and use. 2. The UI is very clean and simple.
How does the colour selection apply on the website? [Is the colour being used friendly? (Colour-blind user)]	23/25 Comments: 1. The colour theme is suitable for the website.
Is the visualisation placement in an efficient position?	21/25 Comments: 1. The graph is informative and nicely placed 2. Map shows relevant places 3. Some visualisation are overlapped (Mac Users)
Overall Rating	89/100

## Performance

In terms of performance, the redirection of pages for our website product has a stable speed. Since this website is deployed with the help of Microsoft Azure, we are only able to subscribe to the free plan of the subscription. This only gives us limited access to the network service of Azure. Therefore, although our website performance is good, there is a problem when entering the website. If there is a user accessing our website for the first time in a twenty minutes time interval, they have to wait 2-3 minutes and reload the URL, as Azure Free Plan is made for testing only. However, after the initial waiting time, any user can access with a decent amount of speed in the website.

## Scalability

There is a limited number of users that can interact with the web application in terms of scalability. Like it is mentioned before on the performance, at this moment, the website can't handle a large number of users at once since we are still using the free plan of Microsoft Azure. Nevertheless, the ability of the web application to cope with users interacting concurrently is still sufficient and competent.

## Security

For this website project, the product doesn't necessarily need a significant amount of security aspects since there are no important user's credentials that are stored in the database. Our database only has information regarding Suburbs, which can be found publicly on the Internet. Moreover, none of our team members have expertise on Software Security. Therefore, although there is no important information saved in the website or database, we are, unfortunately, vulnerable to DDOS attack, as we have set a limit for API Requests per day as well as we are using Azure Free Plan, which has limited amount of resources.

## Limitations of Testing

All of our testing is done by Manual Test. We do not use PyTest or automatic testing functions to test the code. The reason is because our application is a flask web application, not a simple python code. Although we potentially could test our code with a test function, it would be unnecessary as some simple manual black box, white box, and integration testing would prove that our program is running perfectly. It would also take too much time to convert our current code to accept a test function. Lastly, as we are using Google's API to collect some information, we couldn't do much unnecessary testing on some pages, as it would increase the number of API requests, which would increase the cost.

## Conclusion

Generally, this project's goal have been met through this website product because through this product it is able to provide insights to users about searching places to live in Melbourne in a efficient way.

In conclusion, most functions on our website work as expected. It passes almost all of the test cases and both the back-end code and the UI functions as expected. Although there are limitations both to the testing process, It is enough to make sure that errors are handled and fixed.