FIT3163 - Data Science Project 1

User Guide

Project 8A - Suburb Selecter

Group 14

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

Word Count: 1450 Words

Table of contents

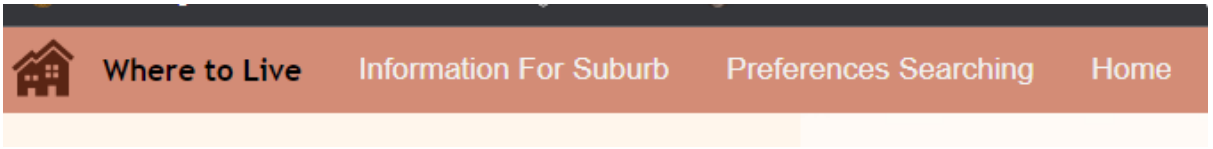
1. Introduction

The website helps people, students especially, who are new to Melbourne. It helps them to find and choose the best suburbs based on their preferences. Basic information for every suburb is also provided and people can use the journey planner features to find the distance and duration of travel from a suburb to a particular university.

The website can be accessed on the internet with this URL:
https://group14testing.azurewebsites.net/
Note that it will take 2 to 3 minutes to load and reload if no one accessed the website within 20 minutes.

2. End User Guide

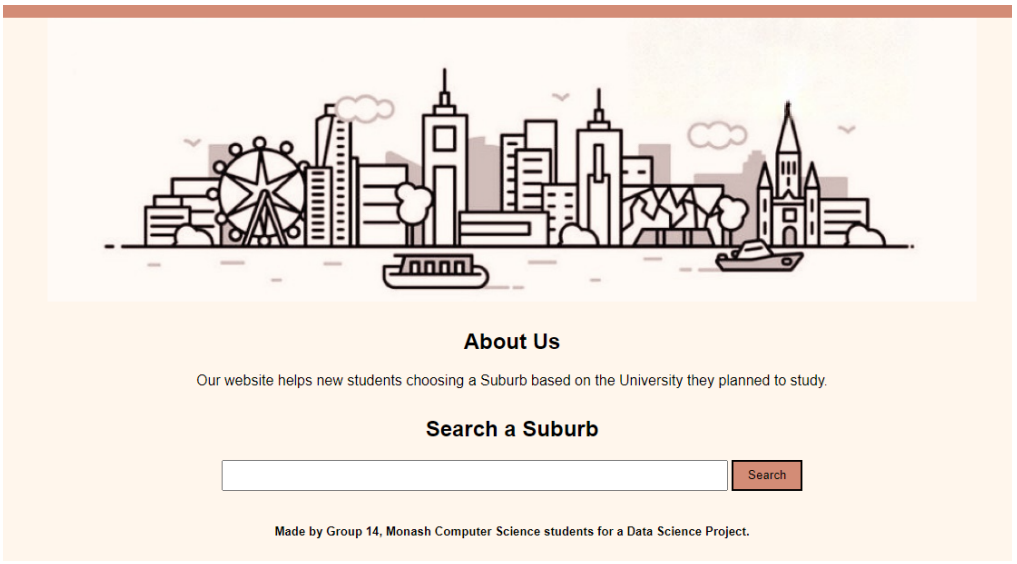
There are five main pages, which consists of the Home page, Home Search Result page, Preference page, List of Suburb page, Preference Search Result page, and Individual Suburb page.



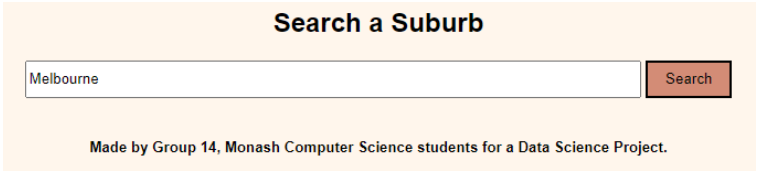
Users can navigate to Home page, Preference page, and List of Suburb by region page with the Navigation Bar. Users can also access each individual suburb page from clicking the suburb name on the suburb list or suburb result page. Users can also manually navigate to the suburb individual page by adding a path into the URL.
For example:
https://group14testing.azurewebsites.net/Clayton
Or if the suburb has a space between in the name:
https://group14testing.azurewebsites.net/Glen%20Waverley
Users need to add %20 that act as spaces in the URL.

2.1. Home Page and Home Search Result Page

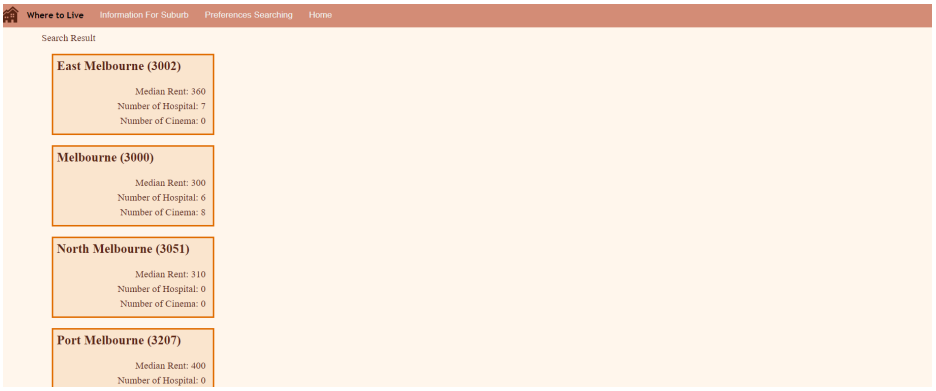
The home page can be accessed when the user enters the website or by clicking on the Home button in the bar. The user can find a suburb by typing the name of the suburb.



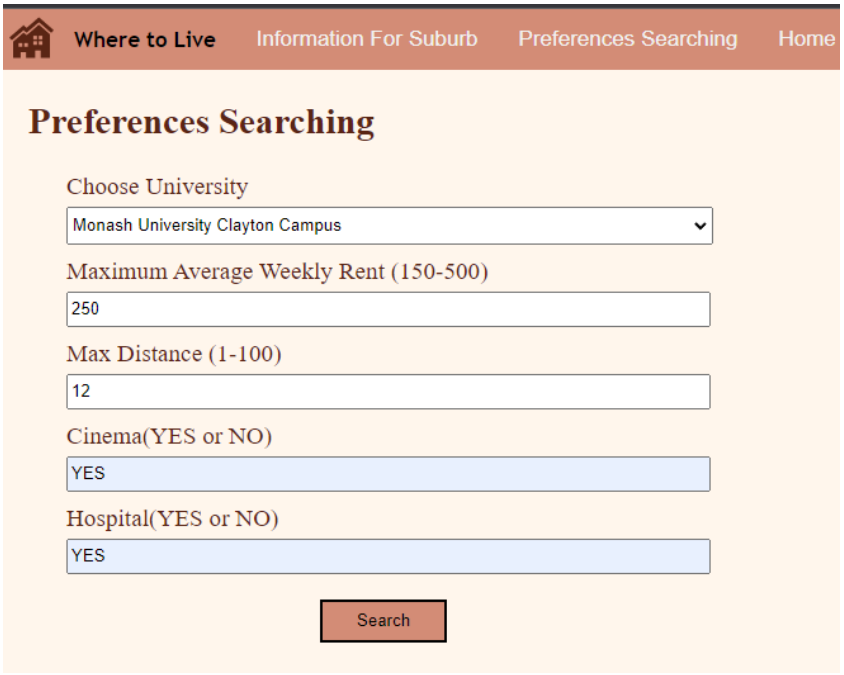
It will then take the user to a Home Search Result page, where the website will list the suburb based on what the user has searched. It will show suburbs that contain the input the user did. For example, when a user searches Melbourne, it will show all suburbs that have Melbourne in its name. The input type is text and is not case sensitive. The box cannot be left empty, but it accepts any input. However, if the input doesn't match with any suburb, it will show an empty result page.



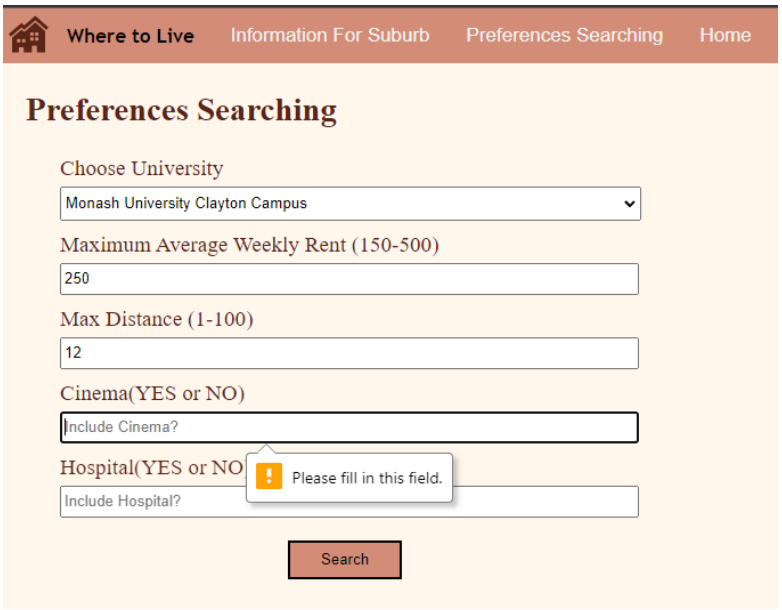
The user could also click on the suburb that will redirect them to a suburb individual page.



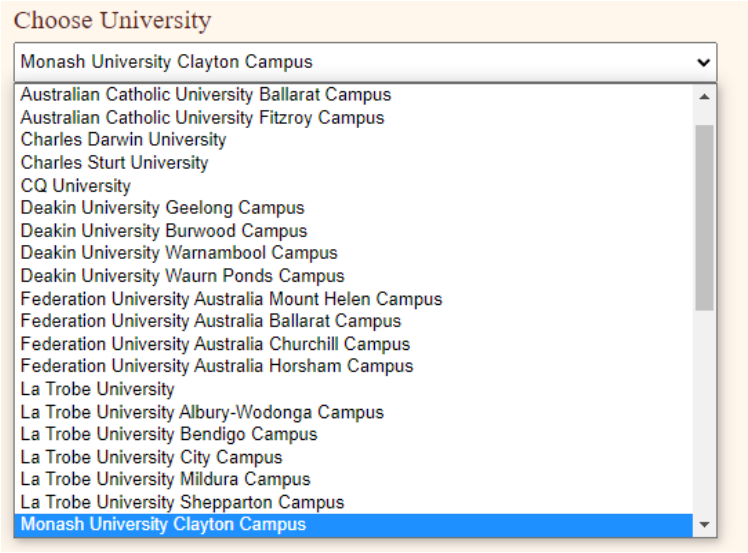
2.2. Preference Searching Page



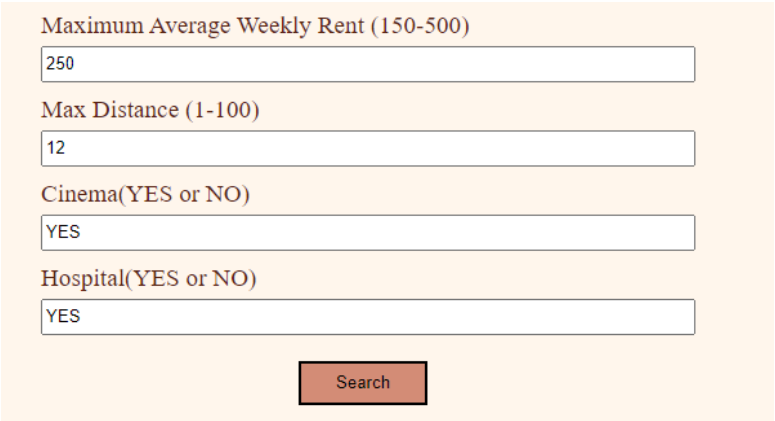
On this page, users can input some of the attributes based on their preferences. All fields should be filled in properly before it allows the user to click the “Search” button. Any missing field will display this warning message:



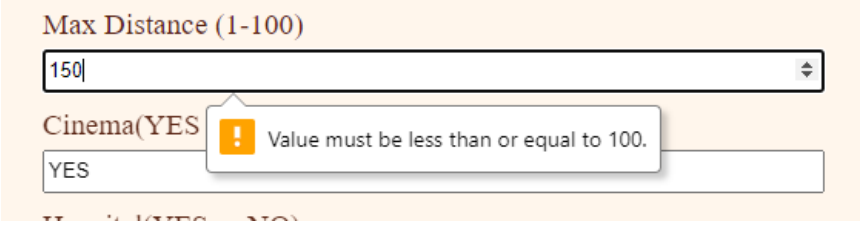
Firstly, users can choose which university they are interested in, or where they are currently or planning to study at. We have around 40 campuses registered in the database where the user can choose from.



Then, users can input the maximum weekly rent and distance.



If the user enters values beyond the indicated range in the title, the website will display a warning message, and the preference combination won't proceed. The renting and distance input can only receive a number. Users cannot input any other type except for numbers.

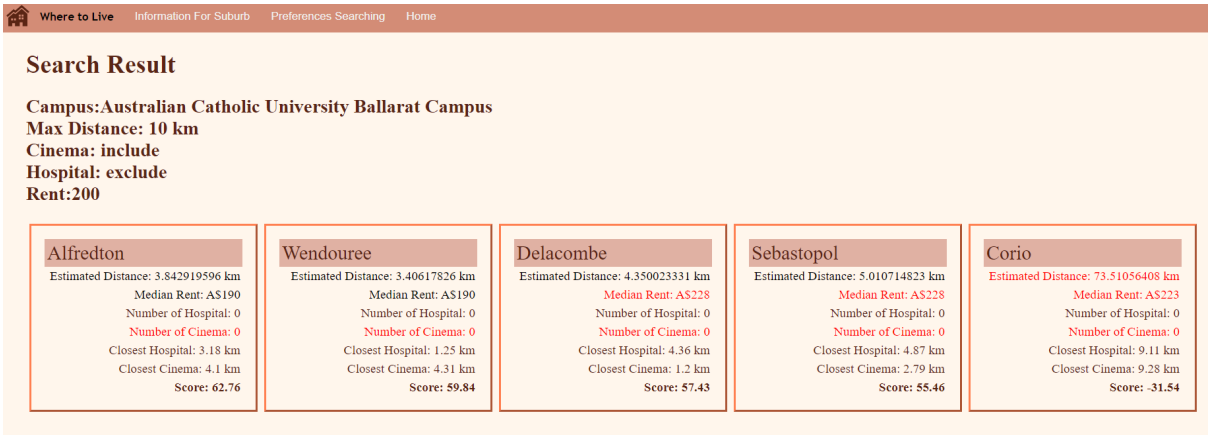


For cinema and hospital, NO and YES indicates whether the user prefers suburbs with cinemas and/or hospitals. If the user types YES, then the recommendation system will prioritise suburbs with the options chosen. If the user types NO, then the recommendation system will neutralise the decision, having all suburbs equally assessed. YES and NO input is not case sensitive.

After all filtering criteria have been filled in, the user can press the "Search" button to proceed to the Search Result page.

2.3. Search Result Page

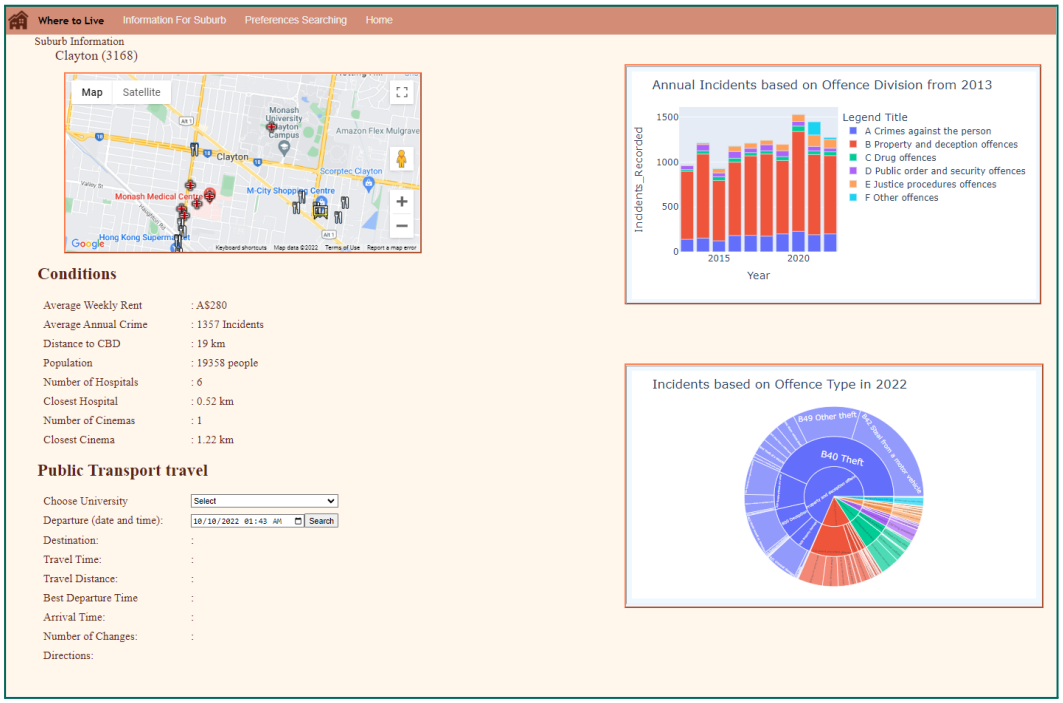
The user will be directed to the Search Result Page to display the top 5 recommended suburbs according to the preference combinations inputted by the user. The ranking is from left to right, having the one with the highest score at the leftest and the one with the lowest (in top 5) at the rightest of the page.



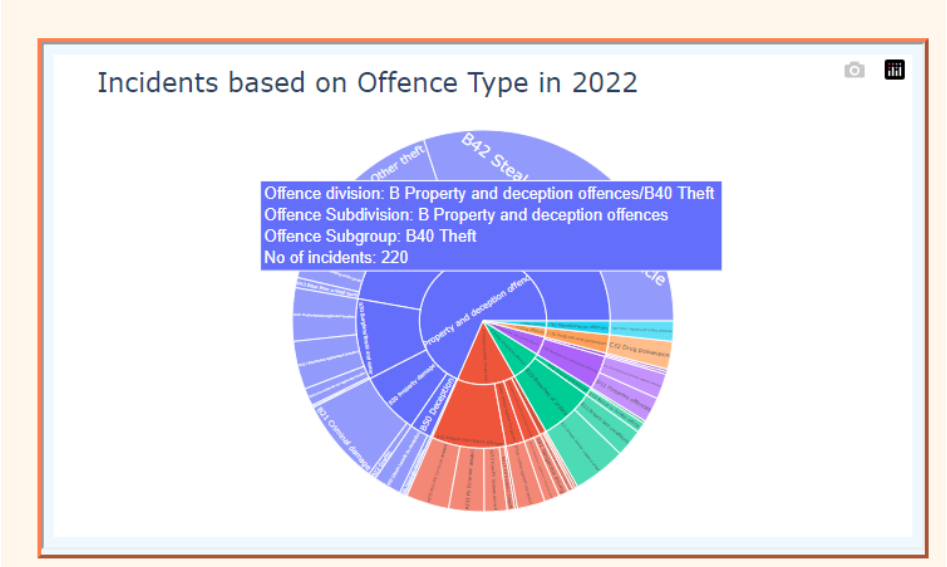
On this page, the user can click on any suburb's name from the recommendation list and will be directed to that specific suburb's information page.

2.4. Suburb information page

This page is used to display in-depth information about a particular suburb. The user can be directed to this page from Search Result Page, Home Result Page, and Information For Region page providing that the user clicks directly on any suburb displayed on three pages mentioned.

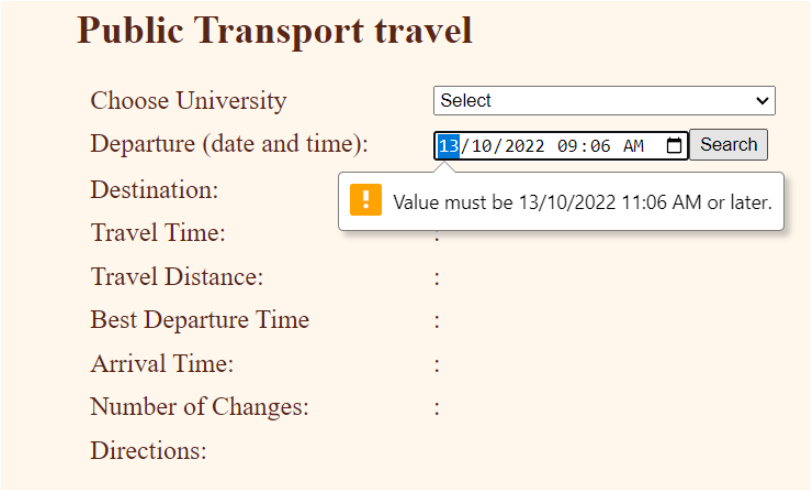


The user can hover to access information from the cropped map, with icons indicating restaurants, pharmacies, and other utilities. The user can also hover over the visualisations on the right side, including tooltips at each data section hovered to provide further information.

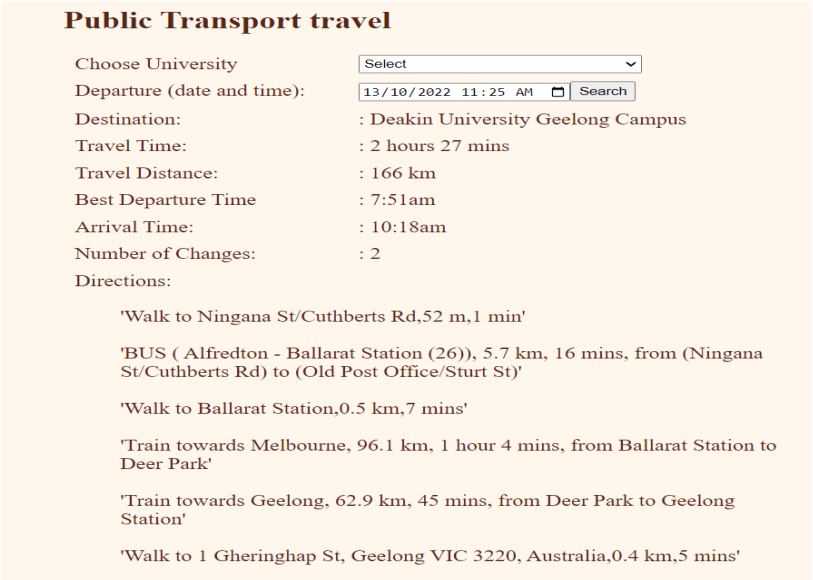


Another feature is called “Public Transport travel”, where the user can choose the university as a destination from the suburb. They can also indicate desired departure time for higher

timeliness of the recommendation. If the user ends up choosing departure time in the past (not current or in the future), a warning message will pop up. For example, the user uses the website at **9:06PM** 13/10/2022 but the user enters **9:06AM** 13/10/2022 in the website, it immediately displays the warning message.



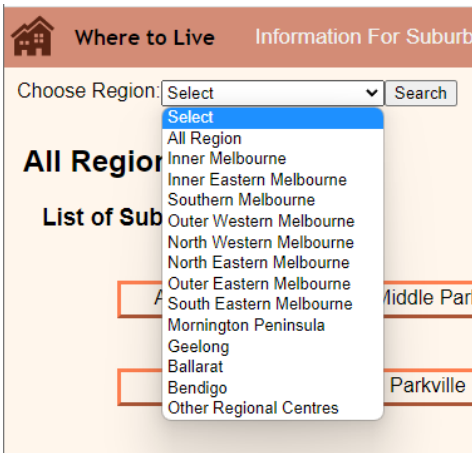
After selecting the university and the right departure time, the user will receive these informations as a response from the recommendation system, including instructions on how to arrive using available public transports:



2.5. Information for region Page

This page displays the list of suburbs according to the region selected. The user can access this page via clicking on the Menu bar located next to the website logo.

At this page, the user can use the dropdown menu at the left corner below the menu bar to select the region. This selection will dynamically change the list of suburbs displayed on the website. This is convenient for any users already interested in specific regions in Melbourne.



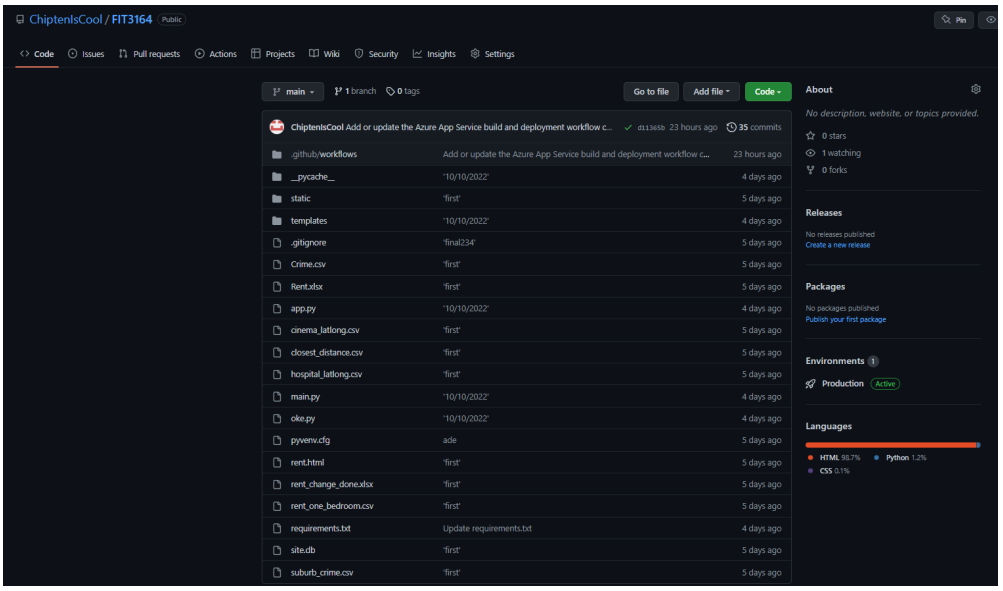
3. Technical Guide

Introduction

In this section, we will explain step by step instructions on how users can run and install the application inside their local machine.

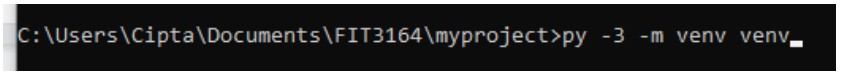
Step 1: Source Code Download

The source code for the website can be found in (https://github.com/ChiptenIsCool/FIT3164).
The user must also have Python (3.8) installed in their computer.

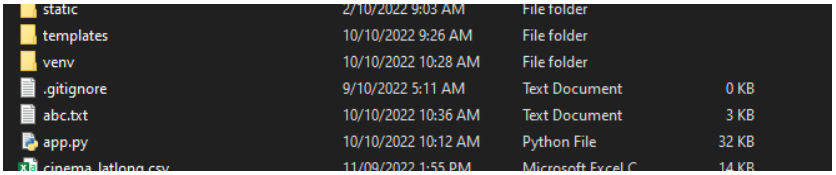


Step 2: Virtual Environment Set Up

After downloading the code, users could use a virtual environment to install the libraries later on, or they could just install it locally on the computer. To install the virtual environment:
Open Command Prompt in Windows
Navigate the path to the project folder
Type command : `py -3 -m venv venv`



This will create a venv folder inside the project file.



Then, the user needs to activate the virtual environment, so that every library installation will be installed inside the venv folder.
Type Command: venv\Scripts\activate

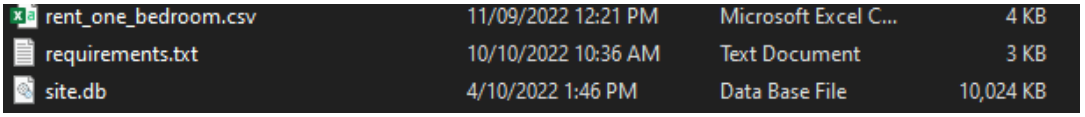
```
C:\Users\Cipta\Documents\FIT3164\myproject>venv\Scripts\activate

(venv) C:\Users\Cipta\Documents\FIT3164\myproject>
```

Your virtual environment is now active. This step needs to be done every time a user opens the command prompt.

Step 3: Libraries Configuration

Inside the file, we have made a text file called requirements.txt, where it has a list of libraries the user needs to install to run the app.



If the user is using a virtual environment, then after activating the virtual environment, run this command in the shell:
`pip install -r requirements.txt`
This will install all the libraries contained inside the text file into the virtual environment.
If the user is not using a virtual environment, then python will install the libraries locally inside the user's computer.

Step 4: Setting the python file

```
(venv) C:\Users\Cipta\Documents\FIT3164\myproject>set FLASK_APP=oke.py
```

To make sure that the computer knows which python file to run when we are running the file, we need to set the flask app. Type this command in the shell:
> set FLASK_APP=oke.py
Make sure that the user types the exact same thing as the above, as even spaces would cause an error on the command. oke.py is the main python file in the project.

Step 5: Running the app

After all the steps above have been followed, the only thing left to do is to run the flask app.
Type the command below:

"> flask run"

It will run the app locally, which should be on the default port (5000).

```
(venv) C:\Users\Cipta\Documents\FIT3164\myproject>flask run
 * Serving Flask app 'oke.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Users can then run the website on their computer.
Type localhost:5000 in the search bar of the browser.