Practice Module for Graduate Certificate in Intelligent Software Agents (ISA)

Project Title:

Rental Buddy

A sophisticated solution for all your rental worries

Project Report

Group Members:

Jonathan Lim Ching Loong (A0261707E)

Pradeep Kumar Arumugam (A0261606J)

Table of Contents

1 Executive Summary

Singapore is a popular destination for foreigners and expatriates who come here for work and study. However, searching for rental house can be difficult if you have minimal knowledge about the place. As such, we developed Rental Buddy, a one-stop platform that assists tenants-to-be in finding their rental place whether they're new to Singapore or have already been here for years. Rental Buddy starts by collecting basic user preferences such as budget, comfortable travel time, frequent destination(s), and the choice of renting a room or a house, all through a form in the user interface. From there, a task bot will search for the nearest MRT station to the frequent destination(s) and shortlist the best location using a knowledge base to begin house-search given the user's preferences. From there, the task bot scrapes through Property Guru and finally returns a list of ten best properties to the user, each with links that lead them to the sites. All these are done in real time so the user can rest assure that the house is still available at the time of the search.

2 Background and Objectives

2.1 Business Case

Singapore is a vibrant city-state in Southeast Asia, known for its economic success and exceptional education system. As such, it has become a popular destination for foreigners and expatriates who come here for work or study. While Singapore is known for its efficient infrastructure and modern amenities, finding suitable long-term accommodation can be a challenge, especially for those who are new to the city-state.

Many existing websites and applications for housing in Singapore assume that the user has prior knowledge of the different neighbourhoods and amenities in the city-state. This can be daunting for newcomers who may not be familiar with the geography and culture of Singapore. They may not know where to begin their search or which areas are suitable for their needs.

To address this issue, we developed Rental Buddy for our project, a one-stop platform that aims to provide helpful advice and recommendations for those seeking long-term accommodation in Singapore, whether they're new to Singapore or have already been here for years. The project assumes that the user has no prior knowledge of Singapore and has not done any research on locations, which can save time and reduce stress for those embarking on their housing search. By providing insights on housing information in Singapore, this project hopes to kick-start the house-search process and potentially bypass the costs of engaging housing agents.

2.2 Market Research Survey

A survey was conducted over WhatsApp with 30 foreigners that consisted of Chinese, Indian, and Malaysian nationals at the start of this project to ascertain some sentiment and common house-search criteria. The questions posed were as follow:

1) When you first came to Singapore, how did you start your house-search process?
2) What criteria do you look out for when you're house-searching in Singapore?
3) How many Frequent Destinations do you consider when you're house-searching?

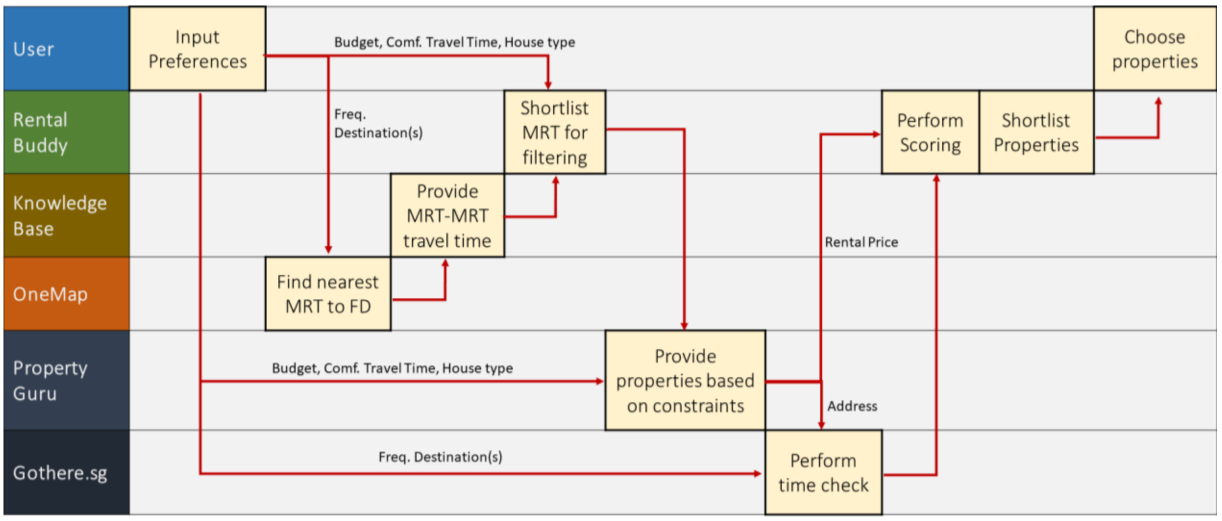
We used the term "Frequent Destination" to describe a place that they go to often every week, e.g. workplace, school, places of worship, etc. From the survey, we learnt from the respondents that their house-search process either start by information from local connections (e.g. relatives, friends, employer, etc.) or via intensive research online. Criteria that they looked out for can also be summarized in the table below:

Criteria	**Num of Responses (out of 30 respondents)**
Rental Cost	30
Distance to nearest MRT/Bus stop	30
Distance to Frequent Destination	30
Amenities nearby	13
Frequency of nearest bus	10

When asked about how many Frequent Destinations they considered, 24 out of 30 respondents indicated only 1 destination. 6 have indicated more than 1 destination. An observation made was that respondents who indicated more than 1 destination are usually those who rent as a group (e.g., renting with friends, spouse, etc) and have to consider the Frequent Destination of their renting partners.

The results of this survey were helpful in identifying what criteria the project can focus on. In order to help with the initial research, we designed an information page for users to have a one-stop location for their understanding of Singapore and its towns. As distance to nearest MRT/Bus stop and distance to Frequent Destination can be covered under the total travel time between house to Frequent Destination, we decided to focus on rental costs and travel time to Frequent Destination when designing the house-search process.

3 Higher Level Diagram



The overall process flow can be described in the diagram above. It begins with the user providing their preference on housing budget, comfortable travel time, house type, and their frequent destinations. We require minimal input from the user, and they include:

1) What is their rental budget (S$/mo)?
2) What is their Comfortable Travel Time in minutes?
3) What house type are they searching for?
4) Where are their Frequent Destinations?

As Singapore assigns a unique postcode to every building block, we request that users enter their Frequent Destinations as the postcode in order to be specific and avoid ambiguity in building names or streets.

This information is used to filter the housing website Property Guru for relevant properties. To obtain an MRT station for filtering, we also utilize the services of OneMap to find the nearest MRT to the frequent destination before performing a calculation to obtain a shortlisted MRT.

From there, Property Guru is scraped, and a time-check is performed to obtain the actual travel time from scraped address to the Frequent Destination. A scoring then performed to further shortlist the properties for the user to choose from. The final shortlist will contain information such as the Property address, rental price, estimated commute time, score, and the link to Property Guru for any further details or transactions.

More details about each process will be described in the sections to come.

4 Knowledge Base

4.1 Travel Time and Travel Fare Database

We designed an RPA process to scrape travel time and travel fare information from the website https://mrt.sg/fare. We tried using TagUI and UI Path to identify the most efficient process since it involves scraping more than 29,000 datapoints. The outcome was that UI Path was much faster but less accurate than TagUI as it depended on visual guides and can potentially scrape the incorrect information if the target visual element has a variation or even if screen resolution changes. However, it is less prone to crashing and can run for hours without crashing, with the longest scrape run being 15 hours straight. Internet issues and website glitches doesn't break the run or cause it to crash in any way as well, making UI path much more reliable than TagUI. As such, UI Path was used.

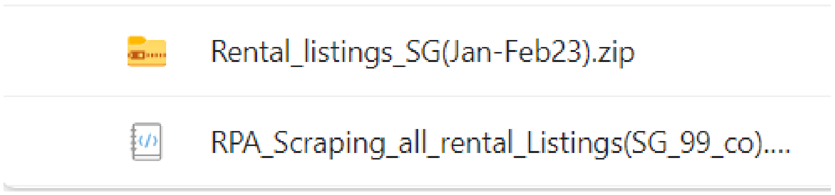
We considered the possibility of scraping only half the dataset since Jurong East to Bishan will have the same travelling time as Bishan to Jurong East, however, upon a preliminary analysis of scraping errors by UI Path, we figured it would've been safer to scrape both directions so that if one fails, the other could still be a backup. The end product showed that UI Path has a 98% accuracy in scraping but despite that, due to the high volume of data scraped, it produced roughly 500 errors.

Excel VBA Macro was used to clean the data by overwriting errors with their correct counterpart data. While that has been effective in reducing 70% of the errors, we still have roughly 150 erroneous data after the macro-cleaning. These errors were identified and TagUI was used to re-scrape the data for those errors to complete the Travel Time and Travel Fare database. Finally, we assigned indices to all the stations and applied this indices to the Knowledge Base for easier retrieval of station information.

4.2 Average Rental Prices Database

We have used opensource TagUI library and constructed an RPA flow for the famous rental website 99.co by all 171 available MRT stations and scraped all the properties listed and their respective rental prices (S$/mo) for export into a csv file. From this csv file, we tried various measures of central tendency and identified mean as the best measure for this set of data. We performed this task separately for room rentals and house rentals for better results.

The codes and the KB we generated from the process can be found for your reference under the folder path "App Pipeline and KB Prep (RPA)/"



Rental_listings_SG(Jan-Feb23).zip

RPA_Scraping_all_rental_Listings(SG_99_co)....

5 Shortlisting MRT Stations

5.1 Considerations

A property website like Property Guru has approximately 22,000 properties listed at any one point. Filters are usually applied in order narrow the scope of search and one of the most common filters used is the location filter, in which you can choose either to filter by various ways such as District, MRT Station, or Town. We considered the various methods as our filter criteria and chose MRT Stations as our filter criteria due to the following reasons:

1) The current MRT network in Singapore covers the island extensively and can be a good representation for the different housing locations.
2) Foreigners are quite likely to be able to relate and identify with the MRT stations more than with districts or towns.
3) There is a high chance that foreigners will use an MRT for travel in the end, which made it relevant to use MRT stations as our yardstick for calculation and filtering for suitable houses.
4) MRT-MRT travel time is also easily quantifiable compared to district-district travel time.

5.2 RPA Process

There are a few websites we can tap on for finding the closest MRT station to a location and among those we tried were Google Maps, Street Directory, Gothere.sg, YourSingaporeMap (YSM) and OneMap. We considered 5 criteria when testing these websites:

1) The loading speed of the website when we enter via the URL
2) The presence of popup ads throughout the process or captcha, which can be detrimental to our scraping process
3) A clear XPath to scrape as this can cause our code to potentially scrape wrong data if unclear
4) A reliable result, which we used a scale of 1 to 5 where 1 is very unreliable and 5 is very reliable. We ascertain this by searching for a variety of potential Frequent Destinations and comparing the scrape result with our manual identification of the nearest station. The more irrelevant the scraped stations (i.e. stations scraped that were way off), the lower the score reliability scale
5) Number of steps required to obtain the result. This is mostly the clicking and typing steps but excluding the first step of typing the location name.

The results of our testing on each website can be summarised in the table below:

Criteria	**Loading Speed**	**Popup Ads**	**Clear XPath**	**Reliability Scale**	**No. of Steps**
Google Maps	Moderate	No	No	2	4
Street Directory	Fast	Yes	Yes	5	2 or 4
Gothere.sg	Fast	No	No	1	2
YSM	Fast	No	No	4	1
OneMap	Fast	No	Yes	5	1

From our results, we identified OneMap as the most efficient and reliable source for getting the nearest MRT to a Frequent Destination. As such, we coded an RPA process using TagUI to scrape OneMap. TagUI was chosen over UI PAth as it can be easily incorporated with the rest of the functions in our Python code.

5.3 Calculation and Scoring Function

The scoring formula for shortlisting MRT stations was based on the following considerations:

1) The average rental for the MRT Station doesn't have to be less than the user's budget as the actual rental properties may include those below the budget.
2) Users are more concerned about travelling time when looking for search filters, hence more weight should be applied to this.
3) A user with a high budget is likely to seek properties within a range of his budget and not settle for the lowest possible price.
4) While travel time may exceed the user's comfortable travel time, if the property excels in every other feature, the user may still consider it. Hence travel time should not be a hard constraint either.

Based on those considerations, we crafted the following formula:

$$MRT\ Score = \frac{10}{0.2(Ave\ Rental) + 0.65(Ave\ Travel\ Time) + 1} + budget\ bonus + time\ penalty$$

Due to the large numbers of rental prices and small numbers in travel time, we applied normalization to these variables. Travel time is known as average travel time as we take the average the travel time to the different Frequent Destinations, i.e. if 15 mins to Destination 1 and 5 mins to Destination 2, the average time will be calculated as 10 mins and input into the MRT Score formula.

0.2 and 0.65 in the formula refers to the weights applied to the average rental and average travel time components. These numbers are obtained via trial and error. +1 was added to the denominator to prevent any circumstances where the Ave Rental and Travel Time is 0.

10 was used in the numerator to give the score a better number to look at. E.g. 8.02 vs 0.802. The fraction 1/score was used so that we can use the idea that the larger the score, the more suitable the shortlisted MRT station.

Budget bonus is applied to push up properties within the user's budget range and discourage the MRT stations that exceed the budget by too much. The budget bonus follow the following rules:

Condition *(budget = b, average rental = x)*	**Bonus Score Applied**
0.5b < x < 0.7b	1
0.7b < x < 1.1b	1.5
1.5b < x < 2.0b	-1
2.0b < x < 2.5b	-2.1
x > 2.5b	-5
Else	0

*** budget and average rental used here are the original values. i.e., not normalized.*

Time penalty is applied to add penalty to the scoring for every minute that it exceeds the comfortable travel time. The rules for the time penalty are as follow:

Condition *(comfortable travel time = $t_{comf,}$* *average travel time = $t_{ave)}$*	**Penalty Applied**
$t_{ave} > t_{comf}$	$0.5(t_{ave} - t_{comf})$
Else	0

*** comfortable travel time and average travel time used here are the original values. i.e. not normalized.*

5.4 Returning a Shortlist

Our system applies a For Loop to loop through all 171 stations to generate a list of scores for each station given the user's preferences as parameters and generates a list that is sorted by the scores. The best 3 scores are taken to shortlist 3 MRT stations. The number of shortlisted stations can be adjusted by applying a keyword argument to the function.

6 Shortlisting Properties

6.1 Websites

We initially wanted to compile a list of properties from multiple websites and be a one-stop platform for house-searchers. Among the websites considered were Property Guru, 99.co, Nestia, Facebook, Rent in Singapore, and Cove. However as some of these sites had either very limited capacity in filtering for user's preferences or have very limited listings available, we narrowed down the websites to just Property Guru and 99.co. The table below summarizes the websites vs the criteria we were looking for:

Websites	**Approximated Total number of listings**	**Budget filter**	**Search by MRT**	**House/Room Filter**	**Accessible without account**	**Allows RPA Scraping**
Property Guru	22,000	0 – 50,000	Yes	Yes	Yes	Yes
99.co	14,000	0 – 20,000	Yes	Yes	Yes	Yes
Rent in Singapore	4200	1 – 9,999	No	Yes	Yes	Yes
Cove	700	800 – 4,200	No	No	Yes	Yes
Nestia	500	200 – 5,000	Yes	Yes	Yes	Yes
Facebook	Unknown	Any value	No	Yes	No	Yes

99.co was later discovered to be very slow in loading and difficult to be efficient when automating, causing a large increase in scrape time. As such, to maintain a reasonable scrape time for users, we decided to only work with Property Guru for this minimum viable product.

6.2 Applying Filters and Sorting Method

Property Guru has a convenient URL system that incorporates the filter options. As such we developed a URL generator that takes the user's constraints as input to generate a URL that leads us to the Property Guru website in a filtered state. The filters we apply include: the shortlisted MRT station, the user's max budget, and the house-type that the user is looking for. The results returned by the website will be sorted according to the ascending order of rental prices. We obtained this procedure via a Genetic Algorithm optimization process which we will describe later in this report.

6.3 Scraping and Scoring

With the filters and sorting method applied to the website, our algorithm uses TagUI to begin reading the list of properties to obtain the property address, rental price (S$/mo), travel time from the address to Frequent Destination, and finally compute a score to help us shortlist a list a 10 best houses for a user.

We applied a similar scoring method to the MRT Shortlisting formula:

$$Property\ Score = \frac{10}{0.65(Rental\ Price) + 0.2(Ave\ Travel\ Time) + 1} + budget\ bonus + time\ penalty$$

The only difference between this formula and the MRT Shortlisting formula is the exchange of weights for rental and travel time as we assumed the users are more concerned about travelling time when looking for search filters but when it comes to actually looking through the rental properties, the rental price plays more weight.

As our goal is to let the user obtain the list of properties in the shortest time possible, we designed a heuristic method to scrape the properties which we will describe further under the Genetic Algorithm section later in this report.

6.4 Returning a Shortlist

Upon the completion of the scraping and scoring, the best 10 results are shortlisted and displayed to the user.

7 Genetic Algorithm and Optimization Processes

Our system aims to return a list of the 10 best properties in the shortest amount of time to maximise user satisfaction. As there as many as 114 permutations on how we can search the property listings in Property Guru, we employed Genetic Algorithm as our optimization process to find the best combination that will help us achieve our aim. The optimization process looks to optimize only the Property Guru segment of the system and as such uses a set of constants pre-generated by the earlier segments to prevent unwanted scoring variations caused by the earlier segments. For all generations, we apply the following constants as input:

Frequent Destination	Marina Square, Singapore 039594
Budget	S$1500/mo
House type	Room
Comfortable Travel Time	47 minutes
Shortlisted MRT	Potong Pasir MRT

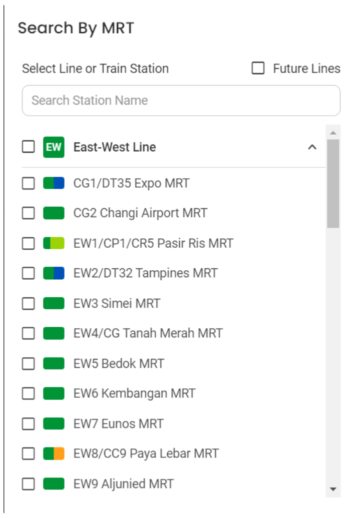
7.1 Chromosomes and Genes

7.1.1 Gene 1: Filtering by MRT Option

We identified three methods for filtering our housing website to our shortlisted MRT and all three methods return different sets of results.

7.1.1.1 Method 1: via "Search by MRT" function

This method utilizes Property Guru's built-in function of searching by MRT. The function lists all the current and future MRT stations in Singapore and allows you to check as many MRT boxes as you desire for the search.



7.1.1.2 Method 2: typing MRT name + "MRT" in search box

This method involves directly typing the MRT name into the Search Box along with the word "MRT" (e.g. Bugis MRT, City Hall MRT, etc).

7.1.1.3 Method 3: typing MRT name directly into search box

This method is similar to Method 2, just that it only types the MRT name into the Search Box (e.g Bugis, City Hall, etc).

7.1.2 Gene 2: Sort Method

The website offers 8 ways to sort your search results after filtering to your preferences. These 8 methods are listed as follow:
Method 1: Recommended
Method 2: Newest
Method 3: Lowest Price
Method 4: Highest Price
Method 5: PSF (low-high)
Method 6: PSD (high-low)
Method 7: Size (low-high)
Method 8: Size (high-low)

7.1.3 Gene 3: Scrape Method

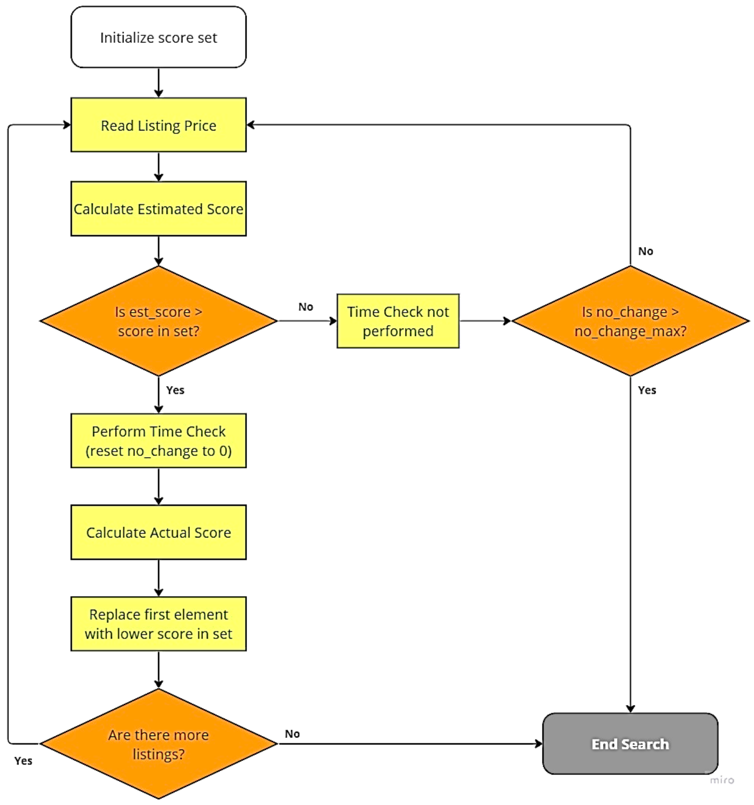
We came up with two methods in scraping the listings for a shortlist. For both methods, we perform scraping to the list after filtering and constraints.

7.1.3.1 Method 1: Brute-Scrape Method

In this method, all the listings are scraped and a time-check is performed on all to generate a score for each property. The list is then rearranged in descending order of the score and the 10 best properties are returned.

7.1.3.2 Method 2: Heuristic Scrape Method

In Method 1, all listings are scraped and time-checked before assessing their scores. If the search results by Property Guru is very long, much time is used to shortlist the scores as some properties with obvious bad scoring (e.g. rental price at a high extreme) is also scraped and time-checked. We felt that if we could come up with a way to find out which properties obviously won't make the cut, we can skip the time-check process and save time on that. As such, Method 2 was developed. Below is the flowchart describing the method:



Initialize score set
Read Listing Price
Calculate Estimated Score
Is est_score > score in set?
No
Time Check not performed
Is no_change > no_change_max?
No
Yes
Yes
Perform Time Check (reset no_change to 0)
Calculate Actual Score
Replace first element with lower score in set
Are there more listings?
Yes
No
End Search
miro

We begin the scrape by initializing a set of scores with the value of -1000. The first listing in the website is then read and a scoring is performed using the price listed and the MRT-MRT travel time indicated in the knowledge base. We assumed that the MRT-MRT travel time would be the shortest possible travel time from this property to the Frequent Destination since it does not include walking time and any bus rides, hence the score estimated from this is supposedly the best score this property can achieve given the rental price since our scoring only consider rental price and travel time.

From there if the estimated score (indicated as est_score in the diagram) is better than any of the current scores in the set, a real time-check will be performed and the actual score will be obtained using the actual travel time. The first element in the score set that is lower than the estimated score is then replaced with the actual score and if more listings exists, the next listing will go through this loop again.

If the estimated score is lower than the scores in the current set, then time-check will be skipped and the property is ignored. We perform a no_change count on this system to count the number of times a property is skipped and if the no_change count exceeds the maximum limit that we set (denoted as no_change_max), the search will end. The no_change count is reset every time a Time-check is performed as that means there is a change to the score set. We assume that when no_change count hits the no_change_max, we have arrived at a maximum score and can end the search process. From trial and error, we identified that a no_change_max of 7 produces reasonable data quality within a reasonable set of time and hence set the default value at 7. We believe that coupled with the right sorting method, this heuristic method will be able to be optimized to produce the best results.

7.1.3.3 Memory Technique

For both methods mentioned above, we apply a memory technique to the scraping process. As the system scrapes and performs the time-check, a copy of the address and time is stored into a temporary list. If in the next time-check iteration the targeted address is already in the memory, the time-check is skipped and the travel time is obtained from the memory. This saves us a considerable amount of time by preventing redundancy in checking.

7.1.4 Gene 4: Time-Check

In reality, travelling time isn't purely MRT-MRT and involves other factors like bus rides and walking time. As such our knowledge base only store MRT-MRT travel times and cannot accommodate to the endless possibilities of travel routes, we identified 3 websites to carry out this commute time-checking function. For all three websites, we applied the same search criteria:

- Start from the House address
- End at the Frequent Destination postcode/name
- If a list of routes is provided by the site, we take the shortest travel time and convert it to minutes.
- If more than one Frequent Destination is provided, we sum up the time taken for each Frequent Destination and divide it over the number of Frequent Destinations to get the average travel time.
- All journeys start at 12.30pm (to prevent errors that may arise if a user searches at 2am).
- Travel time reflects travel time using public transport only.

7.1.4.1 Method 1: Google Maps

Google Maps is quite straightforward in producing our required results. It reads postcodes more efficiently when we input it in the format of "Singapore xxxxxx". From there the results are scraped and calculated.

7.1.4.2 Method 2: Street Directory

Street Directory not read postcodes well when it comes to searching for the shortest travel time and even when entering the name of a place, it still requires you to choose the name from a dropdown list. As such, our code has to perform extra steps to obtain travel time. These steps also include closing all the pop-up ads that appear endlessly in the site.

7.1.4.3 Method 3: Gothere.Sg

Gothere.sg has a very simple user interface and includes the search queries into the URL of the site. As such we obtain the travel time directly after entering the site via a URL generated with the query incorporated.

7.1.5 Chromosomes: Pipeline Function

For the purposes of performing this Genetic Algorithm, we coded a Pipeline Function that holds 4 elements in a list (e.g. [1,3,2,1]), each element corresponding to Genes 1-4 in their respective order. The population is built by generating different pipelines (chromosomes) and evaluating the fitness of each chromosome as part of the selection process. The pipeline function carries out the whole scraping and calculation process according to the specified methods (genes) in the pipeline.

7.2 Fitness Function

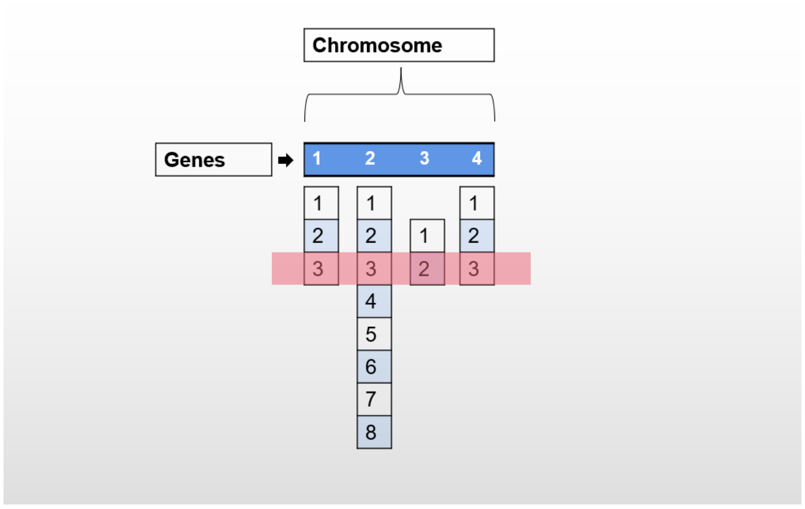
In our algorithm, we used 2 level of fitness to introduce a degree of fairness to all the different pipelines. As we are now aware of that each pipeline generated within a random population of 10 will get us a slightly different list of outputs. For example, Gene 3 has 2 methods, and the 1st method will scrape and calculate the total score of all the available properties while 2nd method will scrape and eliminate members with lower scoring (please refer to 7.3 for scoring details) during the scraping process until it reaches 10 good scores. So, we introduced a constraint for the function to return only the top 10 properties from all the properties it scraped. The fitness score of each pipeline was calculated with the below formula.

Fitness score = $\sum_1^n Listing\ Score/Time\ taken$

n = no of properties shortlisted by RPA task-bot

7.3 Constraints

The constraints we introduced for our algorithm closely follows the Fitness function as both are focused to produce a fair assessment of different pipelines. As we mentioned in the fitness section, first constraint is for the function to return only the top 10 properties from all the properties it scraped. Second constraint is the available (limited) number of methods for each element of a pipeline. The below picture will give an overview of the constraints of individual gene members.



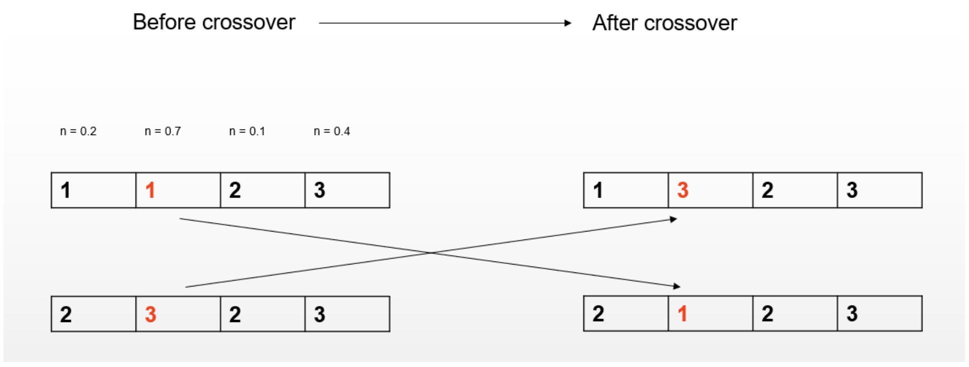
7.4 Methodology

7.4.1 Selection and Evaluation

Once a random population of 10 chromosomes (pipelines) were generated, the selection and evaluation of fitness happen amongst them. The algorithm randomly selects 2 from the 10 and compare their fitness score and only keep the pipeline with higher score. These steps continue for 5 turns to select 5 pipelines. These members are the selected members of each generation.

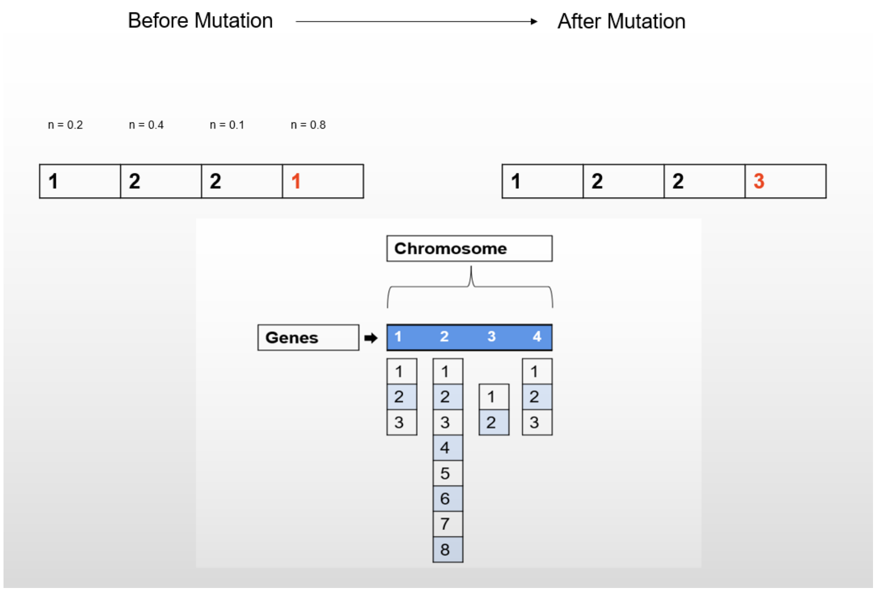
7.4.2 Crossover

Following this for we perform crossover operation on the selected members to get new members. For each iteration for 5 turns, a random number **n** in the range of (0,1) along with 2 random members which are picked from the selected pool and individual gene components of member 1 will get replaced with that of the gene from member 2 corresponding to the same position given n is greater than 0.5. Here the **crossover rate** is set to **0.5** for introducing diversity of population. This will generate 5 new child members from the selected members.



7.4.3 Mutation

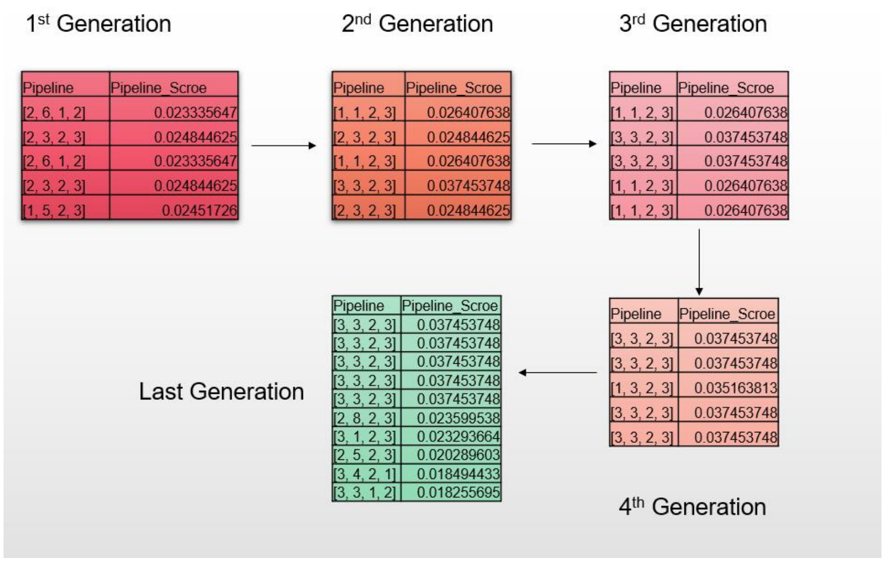
Following crossover, we introduced mutation. For each iteration within each child members generated from crossover, and for 5 turns, a random number **n** in the range of (0,1) and if n is greater than 0.5, that for current gene (in loop) of current chromosome will get mutated to a different number withing the method constraints we talked about in section 8.3. Here also the **Mutation rate** is set to **0.5** for introducing diversity of population. This will generate 5 new child members from the selected members.



We repeated the steps for 5 generation to find the best possible pipeline which fetches properties with highest score.

7.5 Results

As we can see below the algorithm start converging from the 4th generation and picks the pipeline [3,3,2,3] as the best amongst the entirety of population which we have introduced.



1st Generation

Pipeline	Pipeline_Scroe
[2, 6, 1, 2]	0.023335647
[2, 3, 2, 3]	0.024844625
[2, 6, 1, 2]	0.023335647
[2, 3, 2, 3]	0.024844625
[1, 5, 2, 3]	0.02451726

2nd Generation

Pipeline	Pipeline_Scroe
[1, 1, 2, 3]	0.026407638
[2, 3, 2, 3]	0.024844625
[1, 1, 2, 3]	0.026407638
[3, 3, 2, 3]	0.037453748
[2, 3, 2, 3]	0.024844625

3rd Generation

Pipeline	Pipeline_Scroe
[1, 1, 2, 3]	0.026407638
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[1, 1, 2, 3]	0.026407638
[1, 1, 2, 3]	0.026407638

4th Generation

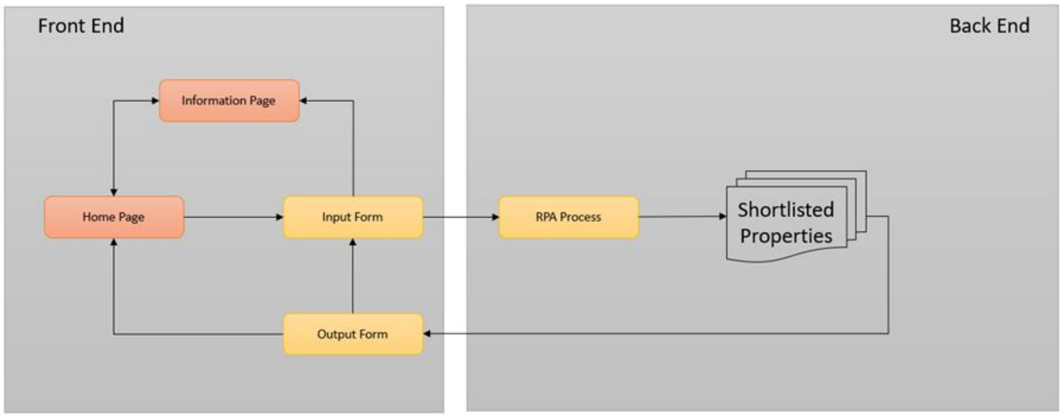
Pipeline	Pipeline_Scroe
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[1, 3, 2, 3]	0.035163813
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748

Last Generation

Pipeline	Pipeline_Scroe
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[3, 3, 2, 3]	0.037453748
[2, 8, 2, 3]	0.023599538
[3, 1, 2, 3]	0.023293664
[2, 5, 2, 3]	0.020289603
[3, 4, 2, 1]	0.018494433
[3, 3, 1, 2]	0.018255695

Here the last population has 5 selected members from the final generation along with 5 child members from the selected ones which were generated by crossover and mutation. They were also evaluated at the final stage with the selected members and overall best pipeline was identified with best score which is **0.037453748**. Hence the optimization was successfully completed and identified the best pipeline which can produce the higher scored properties.

8 UI Design

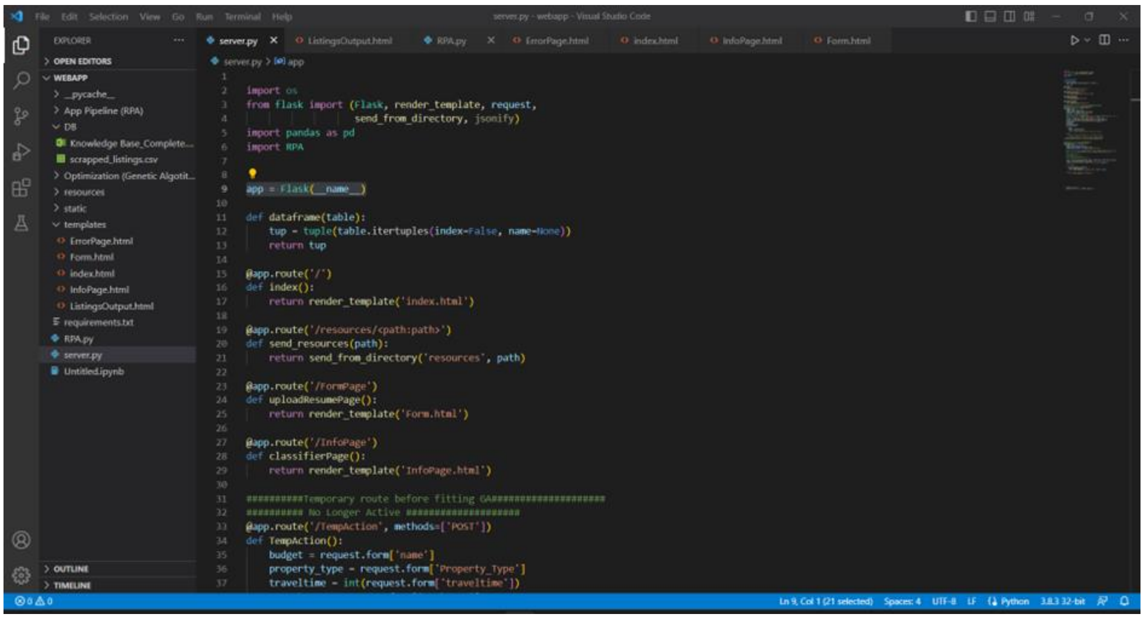


Please note that no transactions will be performed on our system. Users will be directed straight to the housing website, who will provide the methods to contact the landlord or agent.

8.1 Technical explanation of the webpage

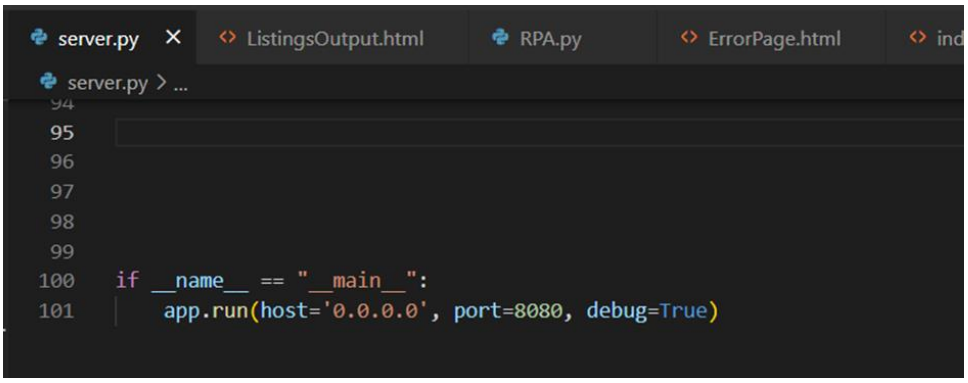
We have used python and flask for routing our web app. The flask web framework (written in python) is used to initialize our web server in the local and navigation to other pages. It is a list of routes (web pages) and their functions in that specific route. Routes will be defined as @app.routes() and functions below such route is the function which needs to be done with that page.

We have built the flask here. The path variables are defined and configured as shown below, which will be used later for specific tasks.



```python

import os
from flask import (Flask, render_template, request,
                   send_from_directory, jsonify)
import pandas as pd
import RPA


app = Flask(__name__)

def dataframe(table):
    tup = tuple(table.itertuples(index=False, name=None))
    return tup

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/resources/<path:path>')
def send_resources(path):
    return send_from_directory('resources', path)

@app.route('/FormPage')
def uploadResumePage():
    return render_template('Form.html')

@app.route('/InfoPage')
def classifierPage():
    return render_template('InfoPage.html')

##########Temporary route before fitting GA######################
########## No Longer Active #####################
@app.route('/TempAction', methods=['POST'])
def TempAction():
    budget = request.form['name']
    property_type = request.form['Property_Type']
    traveltime = int(request.form['traveltime'])
```

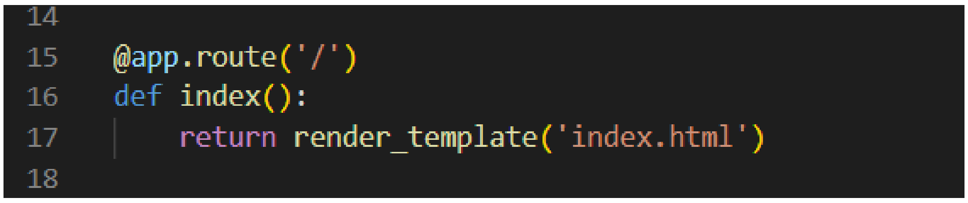
As seen in the below screenshot, we have defined a case statement, so whenever a user run this python file directly, python sets this variable with the value __main__, means we want this code to run only when this file is run directly, not when called from another python file. So, it will start the server in that specified port when executed and the file will be waiting for the user to work with the web app, which will be active in our local server.



```python
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8080, debug=True)
```

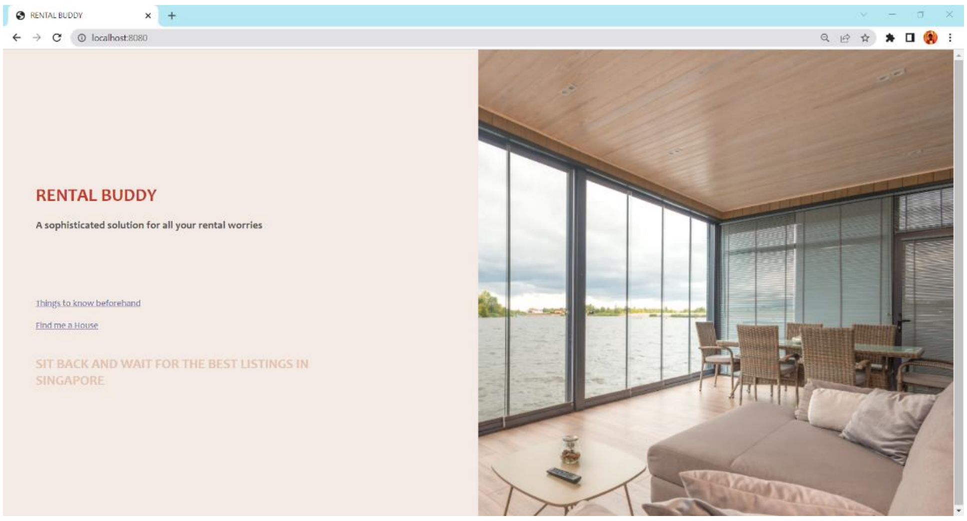
After running the file, the server will be initialized. You can just either type the localhost web url in your chrome and will be able to access the web page initialized. When you initialize the server in 8080, the web page url will be http://localhost:8080/ . The Active page (which is the home page) will be the default "/" (route to root path) and we have rendered our home page of our webapp under that flow. This will be default in the http://localhost:8080**/**.

Hence, as soon "/" was found, flask will launch our main home page (our main page is named as index.html) will be called by our server.py script using **render_template()** function, which was defined in the route and the function as seen below



```python
@app.route('/')
def index():
    return render_template('index.html')
```

The route is " / " and the function **render_template** is a Flask function which is used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder. Hence, we created the template folder in the same path as file, as it defaults to the search location in the templates folder. And our template will be visible in the homepage below



RENTAL BUDDY

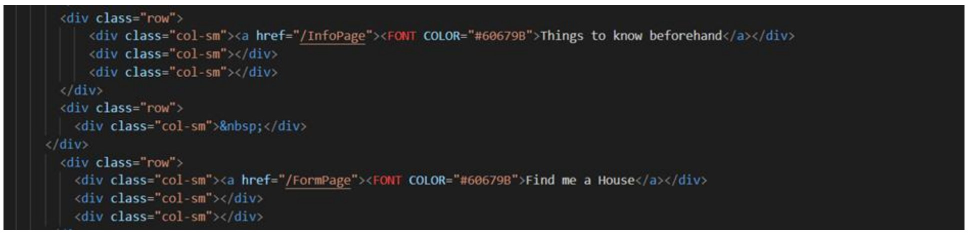
A sophisticated solution for all your rental worries

Things to know beforehand

Find me a House

SIT BACK AND WAIT FOR THE BEST LISTINGS IN SINGAPORE

Here there are two links defined under the index page which can be seen below.

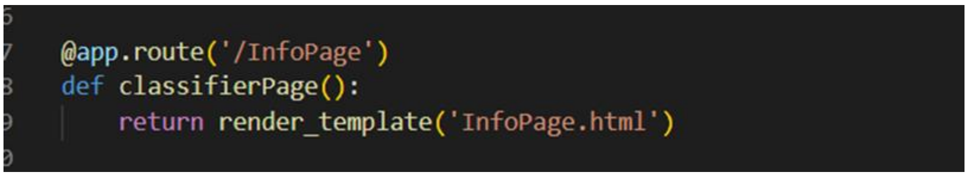


```html
    <div class="row">
        <div class="col-sm"><a href="/InfoPage"><FONT COLOR="#60679B">Things to know beforehand</a></div>
        <div class="col-sm"></div>
        <div class="col-sm"></div>
    </div>
    <div class="row">
      <div class="col-sm">&nbsp;</div>
  </div>
    <div class="row">
      <div class="col-sm"><a href="/FormPage"><FONT COLOR="#60679B">Find me a House</a></div>
      <div class="col-sm"></div>
      <div class="col-sm"></div>
```

In HTML href, the inline (anchor) element denotes a hyperlink from one web address to another and we used flask route definition here (/**InfoPage** and /**FormPage**). The former has a list of few considerations and some basic instructions for using this webapp and the latter, is for page is for users to fill in their basic preferences for getting a list of properties which might be their best options. So, when we click those, the respective route will be called from our server.py file and the function along with it.

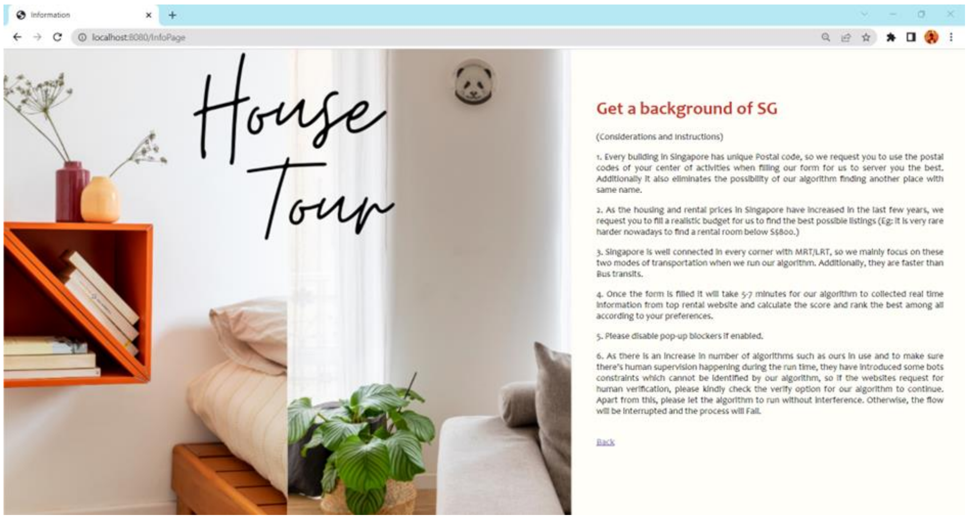
8.1.1 InfoPage:

Once the user clicked **Things to know beforehand**, the below codes will be called from our server script. As we are using flask, the href reference can be used for routing through different pages in the same manner.



```python
@app.route('/InfoPage')
def classifierPage():
    return render_template('InfoPage.html')
```

And our page will be displayed as shown below.



Get a background of SG

(Considerations and instructions)

1. Every building in Singapore has unique Postal code, so we request you to use the postal codes of your center of activities when filling our form for us to server you the best. Additionally it also eliminates the possibility of our algorithm finding another place with same name.

2. As the housing and rental prices in Singapore have increased in the last few years, we request you to fill a realistic budget for us to find the best possible listings (Eg: It is very rare harder nowadays to find a rental room below S$800.)

3. Singapore is well connected in every corner with MRT/LRT, so we mainly focus on these two modes of transportation when we run our algorithm. Additionally, they are faster than Bus transits.

4. Once the form is filled it will take 5-7 minutes for our algorithm to collected real time information from top rental website and calculate the score and rank the best among all according to your preferences.

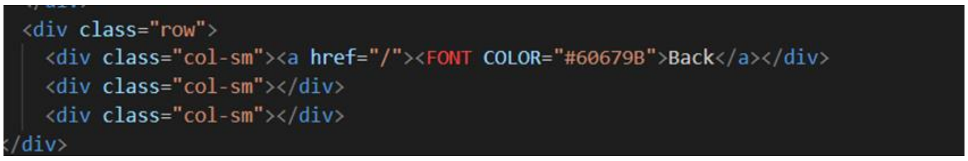
5. Please disable pop-up blockers if enabled.

6. As there is an increase in number of algorithms such as ours in use and to make sure there's human supervision happening during the run time, they have introduced some bots constraints which cannot be identified by our algorithm, so if the websites request for human verification, please kindly check the verify option for our algorithm to continue. Apart from this, please let the algorithm to run without interference. Otherwise, the flow will be interrupted and the process will Fail.

Back

This page, is for the user to get a background of Singapore and our considerations for building the app.

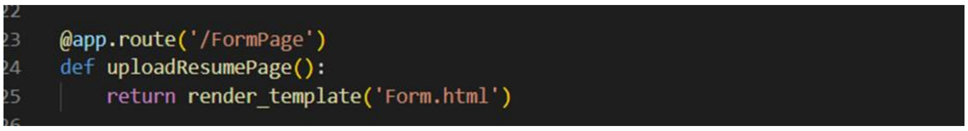
To Go back the user can access the **Back** button to home/index page and the flow is defined as shown below



```html
  <div class="row">
    <div class="col-sm"><a href="/"><FONT COLOR="#60679B">Back</a></div>
    <div class="col-sm"></div>
    <div class="col-sm"></div>
</div>
```

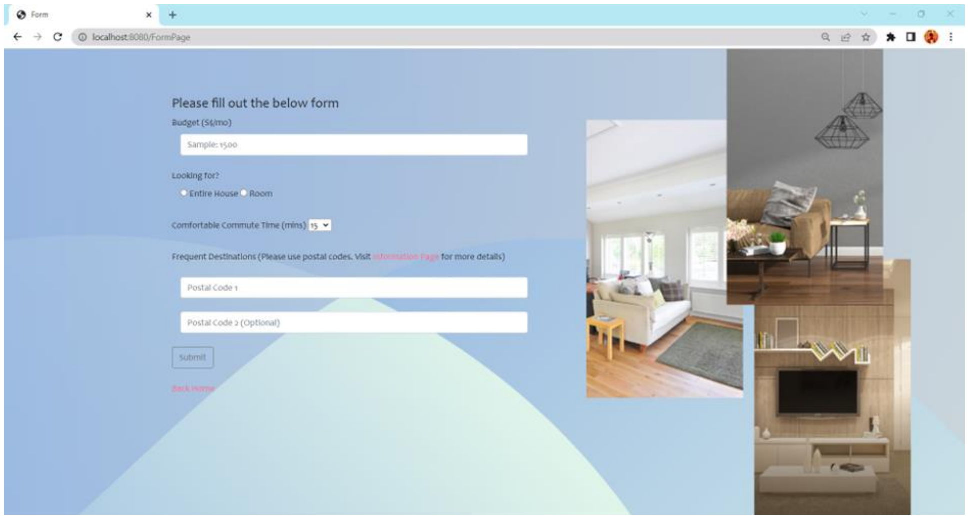
8.1.2 FormPage:

Once the user clicked **Find Me a House** from home, the below codes will be called from our server script.



```python
@app.route('/FormPage')
def uploadResumePage():
    return render_template('Form.html')
```

It calls the Form page which can be seen as shown below.



Form

localhost:8080/FormPage

Please fill out the below form

Budget (S$/mo)

Sample: 1500

Looking for?

Entire House Room

Comfortable Commute Time (mins) 15

Frequent Destinations (Please use postal codes. Visit Information Page for more details)

Postal Code 1

Postal Code 2 (Optional)

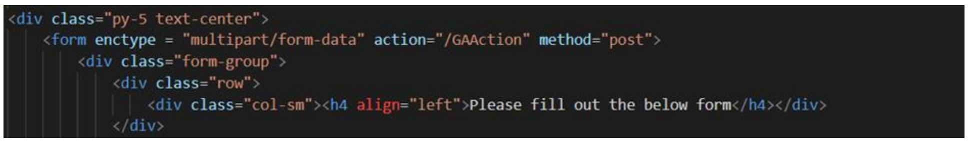
Submit

Back Home

This Page has the following components for the users to fill in.

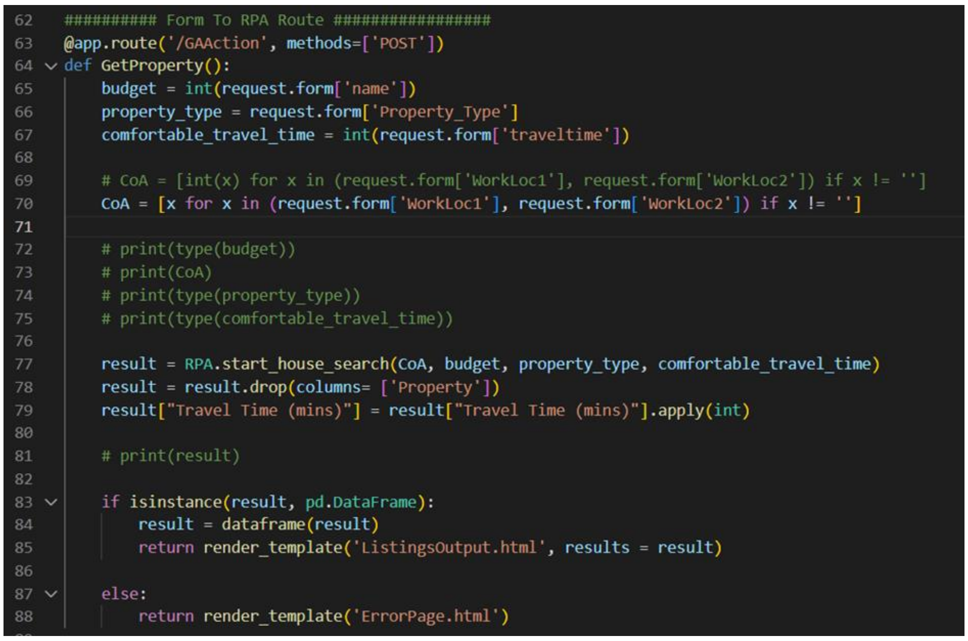
- **Budget (S$/mo.)**: An integer, e.g.: 1500 which will be considered as SGD 1500/month as your budget
- **Looking for**: The type of property you're looking for (either just a room / entire house)
- **Comfortable Commute Time (mins)**: Your accepted commute time in minutes
- **Frequent Destinations**: Postal codes of 1 or 2 of your Frequent Destinations in Singapore

The action once the submit button is highlighted below.



```html
<div class="py-5 text-center">
    <form enctype = "multipart/form-data" action="/GAAction" method="post">
        <div class="form-group">
            <div class="row">
                <div class="col-sm"><h4 align="left">Please fill out the below form</h4></div>
            </div>
```

This calls a different route and there will be an associated RPA function/operation to work on the details which the user has filled. When it was clicked the below block of codes in our server script will be called vial the route calls "/ **GAAction**" and executed.

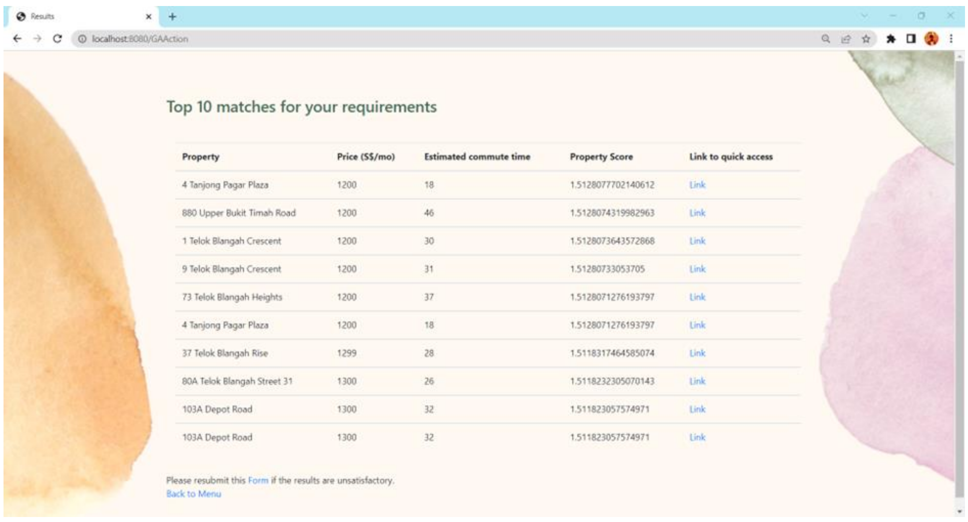


```python
########## Form To RPA Route #################
@app.route('/GAAction', methods=['POST'])
def GetProperty():
    budget = int(request.form['name'])
    property_type = request.form['Property_Type']
    comfortable_travel_time = int(request.form['traveltime'])

    # CoA = [int(x) for x in (request.form['WorkLoc1'], request.form['WorkLoc2']) if x != '']
    CoA = [x for x in (request.form['WorkLoc1'], request.form['WorkLoc2']) if x != '']

    # print(type(budget))
    # print(CoA)
    # print(type(property_type))
    # print(type(comfortable_travel_time))

    result = RPA.start_house_search(CoA, budget, property_type, comfortable_travel_time)
    result = result.drop(columns= ['Property'])
    result["Travel Time (mins)"] = result["Travel Time (mins)"].apply(int)

    # print(result)

    if isinstance(result, pd.DataFrame):
        result = dataframe(result)
        return render_template('ListingsOutput.html', results = result)

    else:
        return render_template('ErrorPage.html')
```

Please go through the RPA section in our report for detailed information on how the task-Bot operation is being executed and the scores for the properties are calculated

8.1.3 Results Page:

Once the task-Bot completes its process of identifying and scoring the top 10 houses and gets the results, it passes the results into the **Results page** (**ListingsOutput.html**) as seen below.



Top 10 matches for your requirements

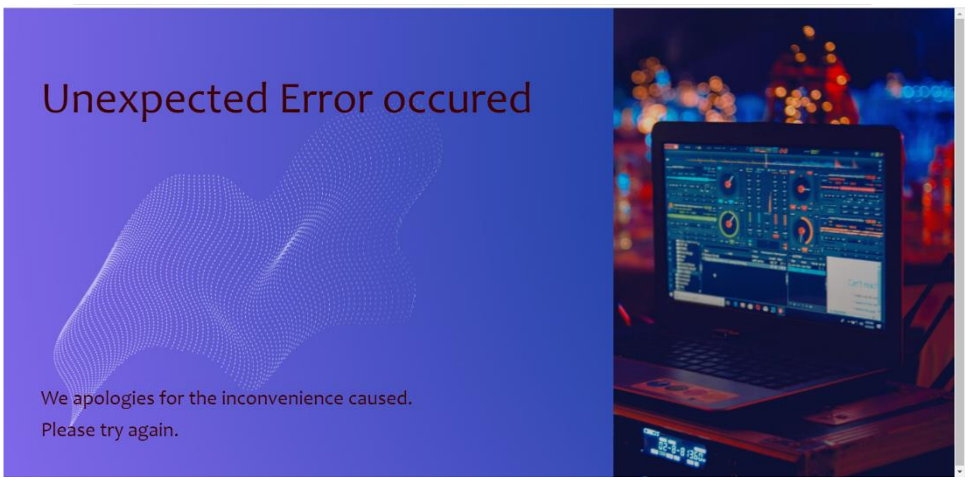
Property	Price (S$/mo)	Estimated commute time	Property Score	Link to quick access
4 Tanjong Pagar Plaza	1200	18	1.5128077702140612	Link
880 Upper Bukit Timah Road	1200	46	1.5128074319982963	Link
1 Telok Blangah Crescent	1200	30	1.5128073643572868	Link
9 Telok Blangah Crescent	1200	31	1.51280733053705	Link
73 Telok Blangah Heights	1200	37	1.5128071276193797	Link
4 Tanjong Pagar Plaza	1200	18	1.5128071276193797	Link
37 Telok Blangah Rise	1299	28	1.5118317464585074	Link
80A Telok Blangah Street 31	1300	26	1.5118232305070143	Link
103A Depot Road	1300	32	1.511823057574971	Link
103A Depot Road	1300	32	1.511823057574971	Link

Please resubmit this Form if the results are unsatisfactory.
Back to Menu

We can directly access the listings by clicking any of the listed links here.

8.1.4 Error Page:

In any case if the RPA could not produce a list of properties, we have an error page to show the operation failed.



Unexpected Error occured

We apologies for the inconvenience caused.
Please try again.

9 Discussion and Further Improvements

Our project is still an MVP (Minimum Viable Product) as it still has much to improve. The heuristic search method proves to be promising but the process and parameters can still be refined to find the best properties in a shorter time. With the current method it is quite likely to obtain a local optimum instead of a global one. Hopefully by improving this method, we can search for more than one shortlisted MRT at a time. Perhaps we can model it after other heuristic models in the future. Another idea is to research more on APIs provided by the map websites and how to use it to generate our travel time (if that were possible).

Processing time aside, the project can consider consolidating listings from more websites to achieve our goal of being a one-stop platform for users to search for their home away from home. A chatbot can also be designed to provide useful advice to the user regarding locality and amenities, just as how a local connection would do.

Lastly, our project focuses on travel time and budget in searching for a house. While these two were the biggest criteria in searching for a house, there is still much that we can apply AI to, particular in analysing the individual descriptions posted on each listing to rule out any listings that may not be applicable to the user. E.g., applying Natural Language Processing to the description can help us to identify if a listing is air-conditioned, inclusive of utilities, or only available for a certain nationality or gender, as is usually posted on the list. Filtering out all the non-applicable listings will help our shortlists to be more relevant to the user and bring up the quality of the shortlists. We can also consider using more reliable Python libraries for the scraping purpose as TagUI is prone to crashing when it encounters an error, and it encounters installation issues with some OS.

10 Conclusion

Searching for a rental accommodation in Singapore can be difficult, but through an optimized RPA process, much time can be saved. For the rental market, this is a business opportunity that focuses on Singapore's potential in attracting foreigners and expatriates, offering them a fast way of finding a rental space with minimal knowledge of the country. While the process is efficient, much can still be improved in the process in terms of accuracy, reliability, and size of search space.

11 References

1) https://www.propertyguru.com.sg/
2) https://www.99.co/
3) https://mrt.sg/fare
4) https://yoursingaporemap.com/
5) https://www.onemap.gov.sg/main/v2/
6) https://www.facebook.com/
7) https://www.nestia.com/
8) https://www.streetdirectory.com/
9) https://gothere.sg/maps
10) https://www.google.com/maps

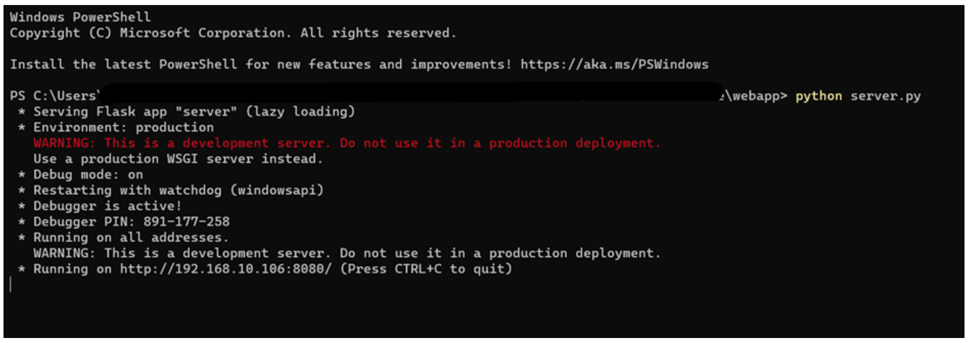
12 Appendices

12.1 Rental Buddy User and Installation Guide

Below are the steps to run the website after downloading the folder structure as it is and placing it in any path on your local.

Step 0: Go through the requirements.txt file and set up the environment with the list of libraries and packages mentioned. Type pip install -r requirements.txt in your command prompt for the setup. If you encounter any error during the installation of Tagui, please refer to 'https://www.yuchenkuang.com/category/TagUI/TagUI-Python-Setup/' for correct installation steps and to set up path variables (if needed).

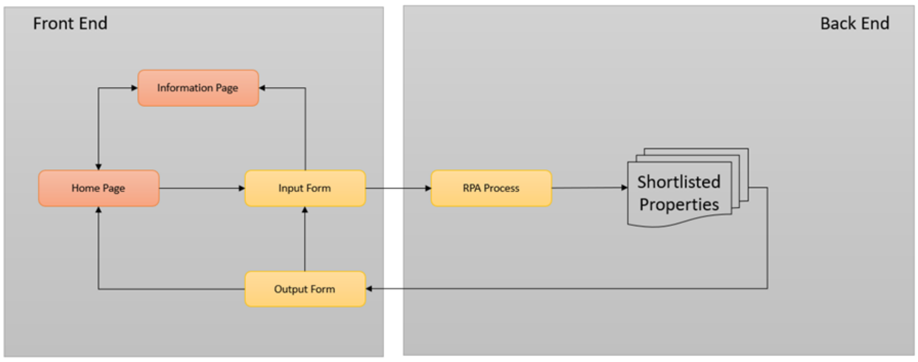
Step 1: open command prompt, navigate to the path (cd to the "server.py" file location) and run the command py server.py or python server.py or py3 server.py if you have both python 2 and python 3 installed. Use the method you usually use to trigger a python file.



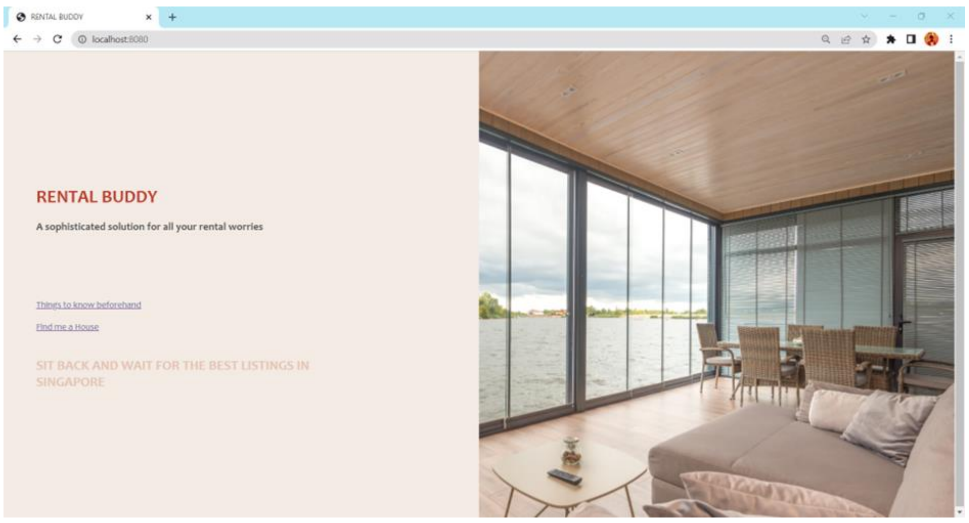
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\[redacted]\webapp> python server.py
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 891-177-258
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://192.168.10.106:8080/ (Press CTRL+C to quit)
```

Step2: After running you'll see something like the above screenshot. Either copy paste the url displayed above or just type localhost:8080 in your browser. Which will open our home page.

The Structural flow of our webapp is like below:



1. **Index - Home page**: - Used to access two types of services – From filling page (Find me a House) & Considerations and Instructions (Things to know beforehand).



RENTAL BUDDY

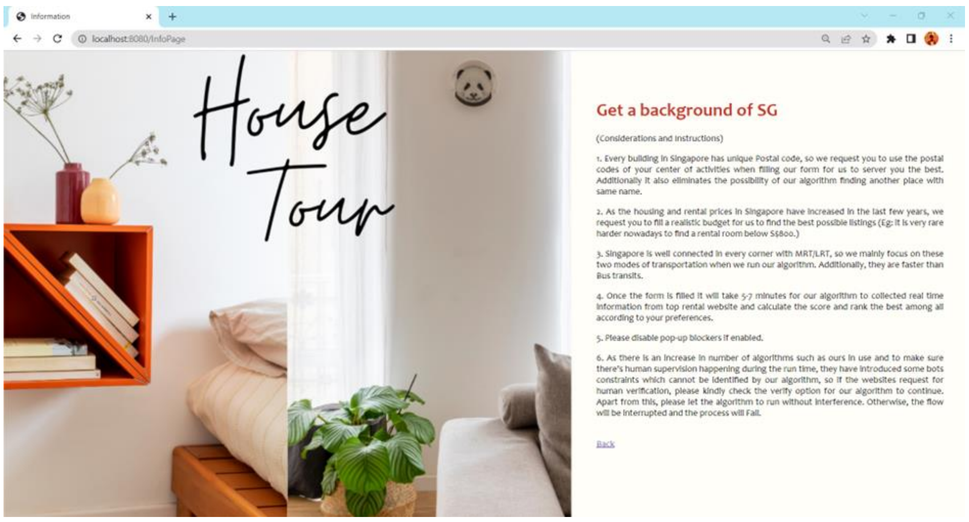
A sophisticated solution for all your rental worries

Things to know beforehand

Find me a House

SIT BACK AND WAIT FOR THE BEST LISTINGS IN SINGAPORE

2. **Considerations and Instructions page (Things to know beforehand)** – Has a list of few considerations and some basic instructions for using this webapp



Get a background of SG

(Considerations and instructions)

1. Every building in Singapore has unique Postal code, so we request you to use the postal codes of your center of activities when filling our form for us to server you the best. Additionally it also eliminates the possibility of our algorithm finding another place with same name.

2. As the housing and rental prices in Singapore have increased in the last few years, we request you to fill a realistic budget for us to find the best possible listings (Eg: It is very rare harder nowadays to find a rental room below S$800.)

3. Singapore is well connected in every corner with MRT/LRT, so we mainly focus on these two modes of transportation when we run our algorithm. Additionally, they are faster than Bus transits.

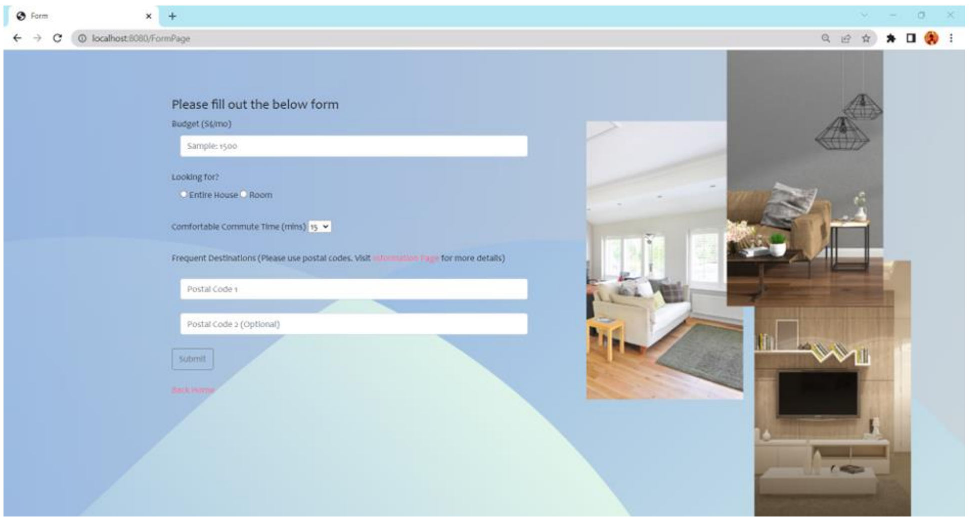
4. Once the form is filled it will take 5-7 minutes for our algorithm to collected real time information from top rental website and calculate the score and rank the best among all according to your preferences.

5. Please disable pop-up blockers if enabled.

6. As there is an increase in number of algorithms such as ours in use and to make sure there's human supervision happening during the run time, they have introduced some bots constraints which cannot be identified by our algorithm, so if the websites request for human verification, please kindly check the verify option for our algorithm to continue. Apart from this, please let the algorithm to run without interference. Otherwise, the flow will be interrupted and the process will Fail.

Back

3. **From filling page (Find me a House)** – This page is for users to fill in their basic preferences.



Please fill out the below form

Budget (S$/mo)

Sample: 1500

Looking for?

Entire House Room

Comfortable Commute Time (mins) 15

Frequent Destinations (Please use postal codes. Visit Information Page for more details)

Postal Code 1

Postal Code 2 (Optional)

Submit

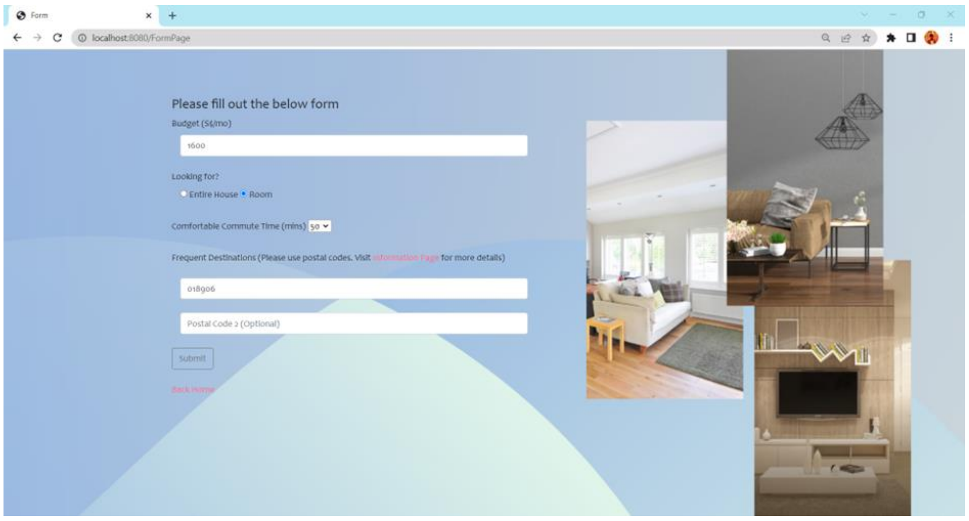
Back Home

Components of this page:

- Budget (S$/mo): An integer, e.g.: 1500 which will be considered as SGD 1500/month as your budget
- Looking for: The type of property you're looking for (either just a room / entire house)
- Comfortable Commute Time (mins): Your accepted commute time in minutes
- Frequent Destinations: Postal codes of 1 or 2 of your Centre of Activities (CoA) in Singapore

A sample Filled form will look like below:

(We also have options to navigate to Information page and home page from here.)



Please fill out the below form

Budget (S$/mo)

1600

Looking for?

Entire House [✓ Room]

Comfortable Commute Time (mins) 50

Frequent Destinations (Please use postal codes. Visit Information Page for more details)

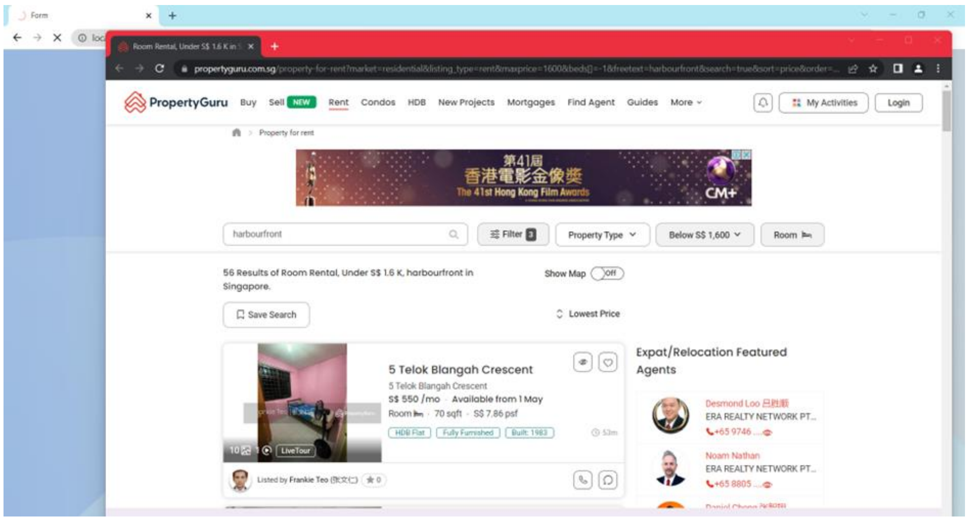
018906

Postal Code 2 (Optional)

Submit

Back Home

Once your press Submit Our Automation scripts will be triggered, and a new additional chrome session will pop up and start the Task-bot's processing steps. Which you can view as shown in the screenshot below.



Room Rental, Under S$ 1.6 K in
propertyguru.com.sg/property-for-rent?market=residential&listing_type=rent&maxprice=1600&beds[]=-1&freetext=harbourfront&search=true&sort=price&order=...
PropertyGuru
Buy
Sell NEW
Rent
Condos
HDB
New Projects
Mortgages
Find Agent
Guides
More
My Activities
Login
Property for rent
第41屆 香港電影金像獎
The 41st Hong Kong Film Awards
CM+
Filter 3
Property Type
Room
Show Map Off
Save Search
Lowest Price
LiveTour

harbourfront

Below S$ 1,600

56 Results of Room Rental, Under S$ 1.6 K, harbourfront in Singapore.

5 Telok Blangah Crescent

5 Telok Blangah Crescent

S$ 550 /mo · Available from 1 May

Room · 70 sqft · S$ 7.86 psf

HDB Flat | Fully Furnished | Built: 1983

Listed by Frankie Teo (张文仁)

Expat/Relocation Featured Agents

Desmond Loo 吕胜顺

ERA REALTY NETWORK PT...

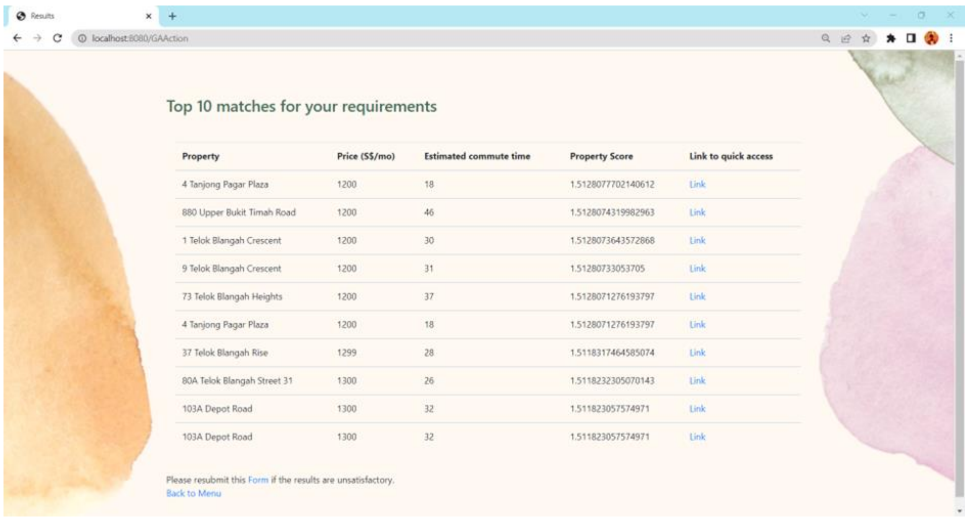
+65 9746 ...

Noam Nathan

ERA REALTY NETWORK PT...

+65 8805 ...

4. **Results Page (Top 10 matches for your requirements).** Once Submitting the form and the task-bot starts working on finding you the best listings, it may take up to 3-5 minutes (depending on your network speed) for the bot to score the top 10 and take you to results page, which can be seen below.



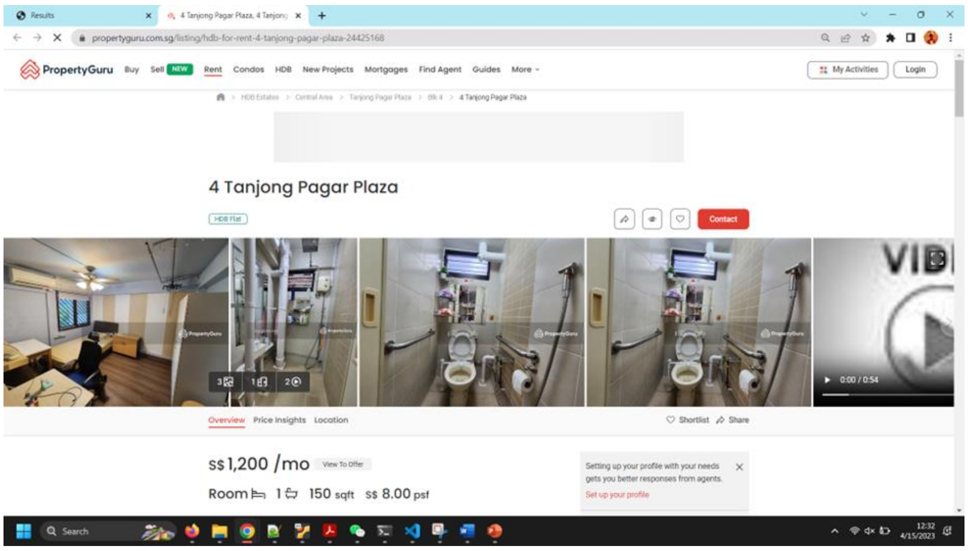
Top 10 matches for your requirements

Property	Price (S$/mo)	Estimated commute time	Property Score	Link to quick access
4 Tanjong Pagar Plaza	1200	18	1.5128077702140612	Link
880 Upper Bukit Timah Road	1200	46	1.5128074319982963	Link
1 Telok Blangah Crescent	1200	30	1.5128073643572868	Link
9 Telok Blangah Crescent	1200	31	1.51280733053705	Link
73 Telok Blangah Heights	1200	37	1.5128071276193797	Link
4 Tanjong Pagar Plaza	1200	18	1.5128071276193797	Link
37 Telok Blangah Rise	1299	28	1.5118317464585074	Link
80A Telok Blangah Street 31	1300	26	1.5118232305070143	Link
103A Depot Road	1300	32	1.511823057574971	Link
103A Depot Road	1300	32	1.511823057574971	Link

Please resubmit this Form if the results are unsatisfactory.
Back to Menu

For Scoring details, please check the Scoring section above for detailed explanations.

You can directly click any of the listed links here to open that listing for further check.



Incase if the listings are unsatisfactory, please try and resubmit the form with slight changes and you can get different listings.