FIT3164 Data Project Final Report

Word Count:9445

https://group14testing.azurewebsites.net/

Group members:

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

Content

1.3 Acknowledgements

First and foremost, our team would like to send our utmost gratitude and appreciation to the project directors Daniel Jitnah, Abishek Sriramulu and Uday Kapur for their help, guidance and support throughout this semester. Ever since the start of the project, up to the project construction, and until the final week, our team has received positive and constructive comments and feedback which contributes to the completion of this project.

Also, we would like to express our thankfulness to each of the group members of the DS_Group 14 which are Yao, Lam, Cipta and Gavin. By working each part assigned whole heartedly for two whole semesters, each member was able to encourage and support each other throughout the process of making and developing the website. Without the unity and cooperation of the group, this project would never have been finalised. Once again, much gratitude is given to every person in this group's data science project for sacrificing enough time to pour out every attention and effort for the sake of the completion of this project.

2.Introduction

Ever since the Covid-19 pandemic started, Australia has never been the same. With multiple lockdowns happening domestically and internationally, the number of international students is decreasing in the country since the majority prefers to go back rather than pursuing education online by staying at home. Fortunately, some of the restrictions along the year starting from 2021 are being eased and lifted, which causes students to return back to Australia. Knowing this phenomenon was currently happening, it could be concluded that the returned students could have a possibility to have a sense of uncertainty in terms of the financial condition for accommodation prices meaning that the data or knowledge the students comprehend before this phenomenon might be invalid or inaccurate. In the meantime, the sudden rise in the number of international students arriving to Australia led to a significant increase in accommodation's demand as well. This also affects the housing market as it is directly proportional towards the demand of the place. Hence, proving the assumption that both international students that have been and haven't been to Australia for the purpose of education have somewhat inadequate insights about the housing market since there have been a few changes before the pandemic phenomenon.

By acknowledging such a problem, there is yet another issue that could be noticed. Selecting the most efficient way to reside temporarily in Australia during study period obviously could take some time to figure out since it is required for students to compare and research some factors themselves that could lead to a good place to stay such as the cost for their living expenses. There could be multiple sources that will reveal different data about a certain topic that could play an important role to determine the option of the location where the student might reside. In other words, this demonstrates that it will consume a slight amount of energy and time as to not only checking factors that could affect a decent housing to stay but also discerning and distinguishing whether the data source could be trusted or a valid one, assuming that the person will commence research by themselves with no hints of insights.

Like it was previously mentioned before, it is required to be able to recognize and understand the factors that might be taken into consideration for choosing the most appropriate location to stay. Usually, these are the general common essential questions that could determine an efficient selection of where to live:

a) How much will the rent of the accommodation cost?

b) How long will it take to travel to campus?

c) Checking the location of the accommodations (is it convenient? Is there a place to buy groceries? close to public transport?)

Looking at these questions, it is quite a nuisance to investigate each of the important questions and research thoroughly for the solution which continually supports the initial claim that reveals that the effort of finding out the perfect location individually or without any leads and information is slightly overwhelming. By acknowledging such issues, this project is intended to provide a suitable solution that will be broadly discussed throughout this final report. Throughout this report, there will be a detailed discussion that explains about the website program such as its features, products and deliverables, also discussing each and every one of its functionalities. This report will also discuss the outcomes of this project which consists of the number of tasks that's been finished, the method of achieving the planned goal such as tools, software, and the algorithms, and also the justification of the implementation of our method. Lastly, there will be a critical discussion on the software project as a whole.

3.Project Background

Project Background Aim

Like it was mentioned before, ever since the Covid-19pandemic phenomenon started to ease up, Australia has been slowly lifting the restrictions especially for international borders which allows most of the countries all around the world to visit. This also opens up opportunities for international students to once again return to Australia (for those students which went back to their country during the Covid-19pandemic) or even new students which have never been to Australia before. Addressed before, the issue is that the newcomer students to Australia have no knowledge of the suburbs and the housing market while the students who just returned to Australia will also possess lack of information about the newly changed accommodation price around Melbourne after the lockdown has been lifted. This project proposes a solution to the presented issues in the form of a website application which will select a certain location of suburb that best suits the user by following their personal preferences of the factors that would determine the most preferable place to reside temporarily in Australia.

Project Rationale

In order to maximize working performance on this project, the team will need reasonable and strong motivation throughout the working process. Explanations about the rationale for the project will also be mentioned and elaborate more in this section.

The motivation for the project is based on the team's aim which targets international students as the main target object. By clearly stating and identifying the issue, the main motivation for this project is that all of the team members could work together on the completion of the initial discussed project goals which is to create a suburb selector in the form of a website product. With the suburb selector program, the team hopes to achieve the requirement of helping not only some of the target users

(which in this case international students) in recommending the suitable area but also other users that are not included on the focused scope of the project's user target.

Furthermore, in this last year, the team members expect to put into practice all of the knowledge that has been instructed throughout the learning process of this whole major. This project will be an opportunity for all team members to grasp a new experience of the real theory implementation that has been learnt. Not solely applying the technical skills to practice, however this unit also teaches us to develop soft skills in terms of managing workload methods and communicating with other team members on carrying out each task efficiently. Such practice is applied so that every team member that desires to continue their careers in the IT or Computer Science area of expertise could adjust and adapt effortlessly to the structure of how actual software projects are operated. Hence, this particular motivation plays a significant role to drive every team member to perform to their maximum capacity.

By the time that the website product is finalized, the team members carry out each task that was given by the tutors in the purpose of getting the marks that are required to pass this unit. This motive is also essential for the whole group since this subject is one of the units that is required to pass in order to graduate in time. As most of the motives are shared amongst the group, each team member is encouraging and pushing one another to give maximum effort on working on this data science project altogether.

Literature Review

To gain understanding about a particular topic or area of studies, a literature review must be conducted. This section will helps us to discover the theoretical context of our topic about this project. Hence, we uncovered and discussed two main points.

a. Website Design

Ngadiman, Sulaiman and Kadir (2015) stated that user interface design is a significant factor and an essential element to develop a web application. Garett, Chiu, Zhang and Young (2016) also explained that website as well as mobile application engagement is critically determined by the quality of the design.

After searching related articles, the review also uncovered that there are listed 20 distinct design elements commonly discussed that affect the user's engagement. With 30% threshold level, there are seven website design elements most often discussed that is mentioned in the reviewed studies which are navigation (62.86%), graphical representation (60%), organisation (42.86%), content utility (37.14%), purpose (31.43%), simplicity (31.43%), and readability (31.43%). It is also stated in the review that website designers must be mindful of different platforms and browsers to minimise the risk of losing users due to compatibility issues. Therefore, it is imperative to be aware of cross-platform compatibility and social media integration since users are beginning to diversify their web browser use. Also added, by integrating social media into website design, user engagement by facilitating participation and interactivity might increase in number (Garett, Chiu, Zhang & Young, 2016).

Kelly believed that the ability to create a good educational website entails more than just knowing how to code. The information to be delivered must be structured to optimize its conveyance. Therefore, he stated that usability, which by definition is how easy a website is to be used, is a significant factor to distinguish between good and bad websites (Kelly, n.d.). One important factor in web designing is how to design the contents within the website itself. In the article, a decent website content design

approach requires a focus on what the users want in a website. Lynch and Horton (1999) summarized five steps for organizing information. Here those steps:

1. Divide your content into logical units
2. Establish a hierarchy of importance among the units
3. Use the hierarchy to structure the relations among units
4. Build a site that closely follows your information structure
5. Analyse the functional and aesthetic success of your system

In the first point of the step it is said to divide contents into logical units. This actually means chunking information for screen space usage, cognitive and hyperlink reasons. The chunking process is necessary because users tend to scan rather than read (Nielsen, 2000). A study found by Nielsen and Morkes stated that 79% of new users scanned pages rather than read them word-by-word probably due to the click-oriented nature of the web and the difficulty of reading from screens.

b. Google Map API

Pansambal, Iyer, Meherkar, and Sharma (2016) believed that google maps has been a great application for travelers since it provides a lot of information such as routes about a certain source to another destination. This application's API will be important for this project since it gives clear direction, distance, and time for users from one place to another.

The google API gives access to map usage which could be added and implemented in a website using JavaScript. Within the API, there are numerous features and functionality that could give more information for users. By adding maps from the Google Map API, a good website could be developed (Ericho, 2021).

To actually implement this in our project, Ahmad, Zia and Khalid (2013) elaborates that google maps can be embedded into a website and also shows areas of business, office, restaurant, hospital etc. Not only does it provide sufficient information, google

map is also simple to customize, including zooming and dragging features and other options as well. The only thing that the website developer is required to do is ask Google for a secret key. Google map only requires a user's registration to register the secret key of Google map API (Ahmad, Zia & Khalid, 2013).

4.Outcomes

a. What has been implemented

Filtering and weighted scoring model has been implemented as the combination of Filtering and weighted scoring model has been implemented as a combination of recommendation algorithms. The overall goal is to strictly generate top-5 suburb recommendations using user inputs.

Users were required to fill in all fields to form a complete preference. Then, in the recommendation process, such combination was applied in two stages:

Stage 1: Stubborn filtering

Predominantly based on the user's manual input of preference, stubborn filtering will strictly satisfy all preferences indicated by the users. Then, it would pass on the list of five selections to stage 2 algorithms to perform ranking.

Stage 2: Ranking scoring

After receiving the filtered list of selections, ranking weighted scoring will proceed in ranking them from highest to lowest using hardcoded multiples. The multiples for each preference attribute were referenced from interviewing local real estate agents in determining the weighting contributing to the overall recommendation. After ranking, the algorithm will return the list in descending order, attached with computed scores.

Optional stage: Follow-up scoring

This stage will operate as a filler in cases when stage 1 couldn't generate a full list (less than five selections). This secondary scoring will be applied to all remaining suburbs to find top-n highest score options to fulfil the list.

b. Results achieved/product delivered

A deployed web application dedicated to suburb recommendations for international and regional students in Australia has been produced. Users could manually enter their preference in 6 criteria and the website would generate top-5 recommendations with less than one minute of loading time.

Apart from the primary function of suburb recommendations, the website also incorporates other features to provide a higher level of details for suburbs, such as:

1. Suburb information: This section provides in-depth information about the suburbs beyond information used in recommendation criteria. This advocates for further understanding and evaluation of a specific suburb, including several data visualisations related to crime.

2. Distance travel time prediction: This section provides an interactive distance and travel time calculator, departing from a specific suburb (this depends on which suburb's information page the user was landing on) to a selection of universities. It further elaborated on the number of changes required as well as detailed step-by-step public transport instructions based on the user's selected departure time.

3. Region suburb listing: This section displays lists of suburbs belonging to specific regions upon selection. It benefits the users who express a pre-existing interest in a specific region, in learning about neighbouring suburbs that support their decision-making.

c. How are requirements met

Evaluating the requirements from the RTM (Appendix A):

Search functions: this function supported finding suburb information, and was successfully integrated into Home Search Page, Information for Region Page and Selection Search Page (generated to display recommended results). Search options can be manual input (typing), or dropdown selection (based on regions).

Preference setting (primary):

This function recommended suburbs according to user preferences, and was successfully integrated into the Suburb Selection Page. All criteria worked to assist the user in inputting within accepted ranges to prevent unexpected errors.

Information page:

This function showed the information of selected suburbs and was successfully integrated into the Suburb Information Page, which could be directed from any search page. Beyond basic information, this function was further extended with the Distance & Travel time interactive section for in-depth exploration.

Maps:

This function showed the map section of specific suburbs, including icons representing relevant landmarks such as transport stations and shopping centres, and was successfully integrated into each suburb's information page.

Visualization:

Static visualizations with interactive tooltips of the statistical analysis of criminal history for each suburb (on the suburb information page) as well as analysis of average rent history for common accommodation types in Victoria (on the main page).

d. Justification of decisions made

Preference setting functions:

Selecting relevant criteria with weightings was a crucial contribution to the website's primary feature: the recommendation algorithm. The criteria chosen were based on data extracted from highly reliable sources such as government databases with long-tracked records. Therefore, data credibility would be consistently maintained with the least manipulation involved.

Distance travel function:

We decided to include this feature in the suburb information page as it provides further usage for the site apart from purely suburb recommender. As suggested by our beta testers (Monash students), interactive public transports instruction provide them with more realistic expectations of travel time beyond distance measurements.

Maps functions:

Apart from hard coded data used for suburb recommendations, we decided to use the map as an optimised geographical visualisation for the relevant landmarks such as transport stations and shopping centres. This greatly supports the user's understanding of the suburb's degree of convenience rather than presenting it as either a figure or a list of addresses (assuming the user wasn't familiar with Victoria's geography).

Visualisation:

Choosing rent and criminal records comparison as the data is extensive with long-period annual comparison (tracking from 2000 to 2022), showcasing a better vision for users to anticipate the suburb's affordability and safety.

e. Discussion of all results

As discussed from previous sections, each feature implemented in our final deliverables had been through thorough internal testing until no bugs were identified. It allowed us to fully deploy this fully-functioning web application as the Minimum Viable Product (MVP) ready to be used at a wider scale immediately. The algorithm has been seamlessly integrated to reduce the memory and storage required to run.

In terms of business stakeholder's requirements, every proposed technical feature has been developed, implemented, with thorough back-and-forth discussions to make sure the final output met the expectations and requirements. Additional outreach to relevant professionals has been conducted to strengthen the accuracy

In terms of non-technical aspects, due to lack of time and expertise resources between team members, little effort is spent on enhancing the security of the website. Most effort was dedicated to smoothen the user interface for them to utilise all provided applications. The choice we made was to have this website as an open source for public access without saving user's input data, therefore, our focus shifts to enhance the quality and user experience.

In terms of user experience evaluation, every proposed technical feature has been developed, implemented, and offered to potential beta users (Monash students) to reflect on whether it matches with the actual users' needs. We received the feedback from our beta testers to which we use to further beautify the interface of the website, with additions of interactive features such as Distance Travel Calculator. We found great advocacy from our users' responses gathered from feedback and direct trials.

Overall, the product delivered by our team members greatly aligned with both the business stakeholders' requirements and the direct users' realistic demand. The choices

our team made in adding more technical features beyond proposed ones to improve quality and maintainability for a faster and more efficient website.

f. Limitations of project outcomes

Databases are hardcoded manually: the central database is manually inputted from CSV and JSON static files, without the ability for incremental updates. Therefore, available datasets to be encoded were limited in terms of quantity and quality, with the need for self-update within a specific time interval to maintain recency and accuracy. This deteriorated the potential of enhancing and refining the recommendation algorithm due to the limitedness of data diversity and recency.

Limit in the number of API calls per day: extensive calls of API required high cost, as most of the integrated APIs have boundaries for free and paid options (not unlimited). Therefore, the backend was coded to only handle limited calls of APIs per day, which restrained the website from serving a large volume of requests in the long run.

Algorithms can be outdated quickly: as the weighted scoring model was purely based on regression coefficients, these coefficients were manually implemented through primary research rather than machine learning and regression training. Therefore, technical changes in database structure and dataset volume impacted the accuracy of currently implemented algorithms. Non-technical changes such as user demand and priority also deteriorated the quality of the algorithm.

g. Discussion of possible improvements and future works

Considering previously discussed limitations, there are certain suggestions for improvements to be made in future operating versions of this project. Providing this project to be continued with support from Monash University, we would look forward to:

1. Enhancing our data warehousing structure: giving our database the capacity to receive, process, and store data more effectively so it can leverage less manual operations. This leverages the time required for data processing and the flexibility for us to incorporate and manage more datasets in the future.

2. Implementing real-time regression training for the recommendation algorithm: with the frequent extension of the existing database, the regression coefficients used in the weighted scoring model could be incrementally improved (in a set interval) by seasonal re-training. Further consideration of changing the algorithms to random forest could also be considered providing a high volume of qualified data stored.

3. Partnerships with governments for higher accuracy data: apart from extracting multiple sources and depending on public APIs (such as Google Maps API, Google Distance Matrix API, etc.) which retrieve high cost, we also consider partnering with local and state-wise governmental organisations for most updated information and data as it serves international students' need effectively as a non-for-profit project.

h. Project outcome quality evaluation from test report

Blackbox, Whitebox (Integration) and Usability testing had been completed throughout the website deployment and bug-fixing period. As we encoded the website to limit its acceptance for user input, most backend testing was smoothly processed with little to no internal server error, handling extreme inputs with warnings and system halt rather than running the server into errors. For frontend testing, website navigation and user interactivity had been thoroughly refined to ensure a seamless experience for a wide range of users regardless of their technical or non-technical backgrounds.

The limitation of this testing process is that it is purely done through manual encoding, with no usage of PyTest or automatic testing function. Simplicity in the website's available functions made complicated testing unnecessary to implement. Dependencies on external Google APIs as a highly recognised resource for its accuracy and timeliness also prevented us from conducting unnecessarily complicated tests. Furthermore, the focus lies on integration testing of the website as most information is formatted to display as interactive visualisations. Therefore, more involvement in manual testing internally and with external beta testers was considered more crucial in incrementally improving the platform from back to front, as our time and labour resources were scarce amongst team members.

i. Critical discussion on outcome.

Mostly the actual technical implementation was commenced and completed predominantly within semester two, to which it managed to balance between the business stakeholders' requirements and the direct users' interest in delivering a highly satisfied product.

Our focus shifted from over-investing into algorithm testing and development to user interface enhancement. Therefore, our team settled on hardcoded weighted scoring model for consistency and avoidance of unexpected predictions due to limitedness of high-quality and extensive datasets available for extraction publicly (such as rent information, crime information, up-to-date listing of entertainment including restaurants, cinemas, transports stations, etc.).

These were then transferred into the frontend infrastructure in forms of interactive functions and visualisations for our broad range of users (including non-technical users). Due to lack of expertise and time available to either execute complex feature testing, our deliverables inevitably still had spaces for innovation before serving wider ranges of purposes. We also consider a feedback and a suburb rating page for the users to be able to provide direct and constant feedback for the web application.

Lack of internal resources and external offerings restraint our team from optimising the performance of our project deliverables from the technical perspectives, but we have managed to provide a viable solution accommodating our target users' primary needs. If we were to be provided with further time, monetary, and partnership support, we strongly believe the potential of our deliverables can excel exponentially.

5.Methodology

1. Design:

Version control system

The version control system our team will be using is GitHub. We chose GitHub because it's an excellent source hosting platform for collaborative work. For long-term projects, this platform can track the progress of the project, help us track every addition and modification of the code, and keep all versions of the project code for all project members to view.

GitHub not only serves as a repository for code and has basic web administration pages, but also provides discussion groups, an online file editor, and snippet sharing. GitHub is a shared platform that supports team operations, so each team member can submit their own code and make changes to it. In the design scheme of 3163, the control version we choose is also GitHub. Although we tried to use GitLab as the control version system during the project, our Monash GitLab could not connect to azure application services. Besides, the Monash GitLab server also has limitations. It can no longer be used after our graduation. Therefore, although GitLab has a perfect management interface and permission control, GitHub has everything that the team needs.

Management of data

For the database, we use the SQLite database, embedded in the project file. Because SQLite is a single-user architecture, each database is targeted to one client, which is exactly what we need for our application. Each variable determining suburban scores has its own pre-processed data and is kept in the set. Most of the preprocessing is done in python. Multiple queries are made to extract the suburbs based on the user filter, and a score for each suburb is calculated and sorted.

In the initial design proposal of 3163, we plan to use MongoDB in the NoSQL database. NoSQL databases can store hierarchical data well, but it is not good at complex query and has limited processing ability, while SQLite has strong query ability and stable processing ability. Therefore, SQLite is a better fit for our project.

Collection of data

Most of our data comes from official sources, which ensures the reliability of the data. This includes criminal history, public transport information and a list of hospitals. In addition, data on rentals, housing listings and shopping malls will be collected from other sources, including GitHub for suburbs and the data of universities from the web of University Reviews. For the collection of Map data, three API data sources are used, namely Google's Directions API, Places API and Google Map API.

In order to determine the best suburb, five recommended results can be given, so each variable will be given its own weight and the score of each variable will be calculated based on the data of each suburb. The final score will be calculated based on the score of each variable and its weight. This was used to create a final score to select the top five suburbs for users to choose from. For data collection, basically follow the initial design proposal of 3163. And it uses more sources of data collection.

Preprocessing of data

We preprocessed five datasets, namely crime, cinema, hospital, rent, and the distance from the suburb to the CBD. First, we cleaned the data, deleted missing values and dealt with outliers. For the crime dataset, this dataset will contain the history of crimes committed in Australia. The data should include the time the crime occurred and the suburb. The number of crimes committed in the suburbs each month will be calculated for visualisation and overall use to calculate the score.

For the cinema dataset, the dataset has a list of cinema names, as well as their locations. The dataset also includes the latitude and longitude of the location, the

suburbs they are located in, the number of hospitals in each suburb, and the calculation of distances. For the rent data set, the data set includes the number of rooms counted in each suburb, and the median value of these rents is used to represent the rent of the suburb. For the hospital dataset, the key data to be retained are the list of hospital names and their suburbs and locations, which are mainly used to count the number of hospitals in each suburb and calculate the distance. For the distance dataset from suburbs to CBD, there should be a list of suburbs and their distance data to CBD.

Secondly, data integration is carried out. For example, distance data in the distance data set from hospitals, cinemas and suburbs to CBD are integrated to integrate a new data set, which includes the nearest distance from suburbs to hospitals, cinemas and CBD respectively, and the number of hospitals and cinemas in each suburb is given.

Compared with the initial design of data preprocessing in 3163, the data sets to be processed are different. First of all, in the initial design we planned to use the PTV API to get directions, now we use Google's directions API and public transport vehicle types to get directions, because PTV API is too complex for us, and Google's directions API is enough to get the information we want. In addition, the added data sets preprocess the hospital and distance related data sets.

Algorithm used

Two algorithms are used in our project. One is the filtering algorithm, which aims to achieve the suburban recommendation function. It can recommend the most suitable suburban options according to the user's input preferences about universities, rent, distance, cinemas and hospitals. Filtering algorithm is an iterative algorithm. The filtering algorithm filters out parts of the text that are not a match, while finding possible matches.

The other algorithm is the scoring algorithm. In order to list the best five results in all the search results, it is necessary to score each suburb according to the scoring algorithm, arrange according to the scores, and select the five suburbs with the highest

scores and recommend them to users. Fisher score is one of the most widely used supervised feature selection methods. The scoring algorithm we will use returns the rank of the variable in descending order based on the fisher score.

High level of pseudocode

After the user enters their preferences, firstly it will calculate the scores for each of the individual attributes. There will be scores for rent, distance to hospital, distance to cinema, criminal incidents, distance from the suburb to the uni, etc. Each score is calculated depending on the value while also considering how much it should affect the overall scores. A type of weighted scoring system is used to calculate the overall score based on the previous individual score. Some variables such as distance to university as well as distance to hospital would have much higher impact than distance to cinema for the overall score. The score then will be used to sort the results.

After all the scores for each suburb were obtained, the five highest-scoring suburbs that meet all of the user's preferences will be recommended to the user. If some of the relevant conditions in these high-scoring suburbs do not match the preferences entered by the user, the content that does not match the preferences entered by the user will be marked in red.

The idea from the original design in FIT3163 does not change. It has some minor differences on the scoring system as well as the variables, but it is made as what we have designed. Initially there should be much more variables than just hospitals, cinemas, and rents. However, as we have a limited amount of time to gather more data, our team could only manage with the current number of attributes. Furthermore, initially we planned to save the score in the database for each suburb except the distance to university, however it is unnecessary as the algorithm could just calculate it directly.

Software and tools

The editor we have chosen to use is Visual Studio Code. Because it's very quick and easy to use, and it's also an editor that most of our panellists are familiar with. The primary programming language of choice for project development was Python. The reason is mainly because the members of our team are most familiar with Python and it works so well for data manipulation. Flask is used to implement the functions of the back end.

Use libraries like pandas and NumPy for data preprocessing, manipulating tables, and analysis. These two libraries work well with raw data. matplotlib was also used to help with statistical analysis and visualisation, and some related diagrams were drawn. These operations are done on the Jupyter Notebook, using python3. The database type we are using is SQLite database. UI design optimization continues on Figma, which was used in 3163. Finally, using HTML and CSS as our front-end implementation, it will connect to Flask, the Python-based Web framework that we'll use to build our website.

6.Software Deliverables

1. Summary of software deliverables

a. What is delivered

There are 2 types of deliverables that we are delivering to the client, which are the technical aspect of the deliverables (product-related deliverables) and the non-technical aspect (project-management deliverables).

The product related deliverables will contain most of the product made on this project, which mainly is the website itself and the source code of the website. The website is already fully functional where it could recommend suburbs to users based on their chosen preferences and based on our algorithm built inside the source code. It also contains basic information for every suburb (only those exists in our database) such as criminal incidents and median rent price for one bedroom. It also has one major feature, which is that the website could show directions, travel time from a suburb to a particular university with Google's Directions API. The user could choose the university as well as the departure time.

Furthermore, our team has also delivered User Guide, Testing Report, and Code Demonstration to the client, where it has more detailed information on how end-users can use our website, how to install and run the source code locally, as well a report to show the extensive testing we have done to the website's and the source code. In addition, the codes regarding data cleaning as well as the raw and cleaned datasets for the database and visualisations are included as well. Lastly, the User Interface initial design and python codes for visualisations are included inside the deliverables archive.

Secondly, the project-management deliverables will contain the report of the management aspect of our team. We deliver the updated Requirements Traceability

Matrix (Appendix X) to show the technical and non-technical requirements of the project. Our team builds the website while making sure that all of the requirements are fulfilled. Furthermore, we included Risk register (Appendix X) where we made our proposal to identify potential setbacks during the project and how our team managed to solve the problems listed. Lastly we Include the Project Management Report and Team Management Report where it has detailed information regarding how our team operates, the project management methodology, and our team detailed meeting minutes, where we discuss our weekly problems, goals, what have been or have not been made and implemented, as well as new ideas that would help the website even better.

b. Sample screenshots and description of usage

In this section, it has some of the main pages of our website, which satisfy both the main project goal as well as some notable out-of-scope features.

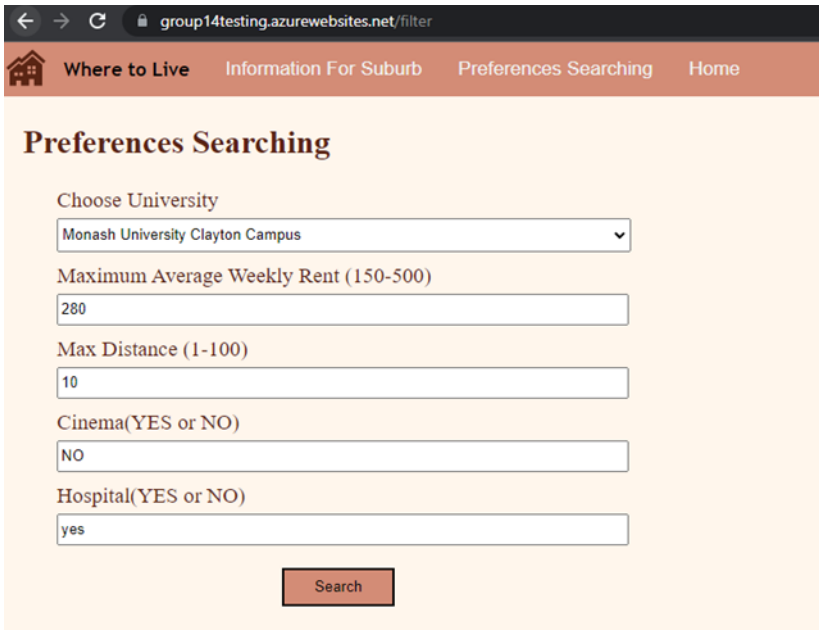


Figure 6.1, Preference Page

In the preference page (Figure 6.1), Users can input their preference so we could choose the best suburbs based on the input. Our Algorithm will then consider the input then show the users the results in Figure 6.2.

Figure 6.2, Preference Result Page

Five best suburbs based on user input will be shown to the user, together with some of the basic information of the suburb itself. Some information is highlighted red where it means that Suburb has some Category where it is not within based on the user preferences.

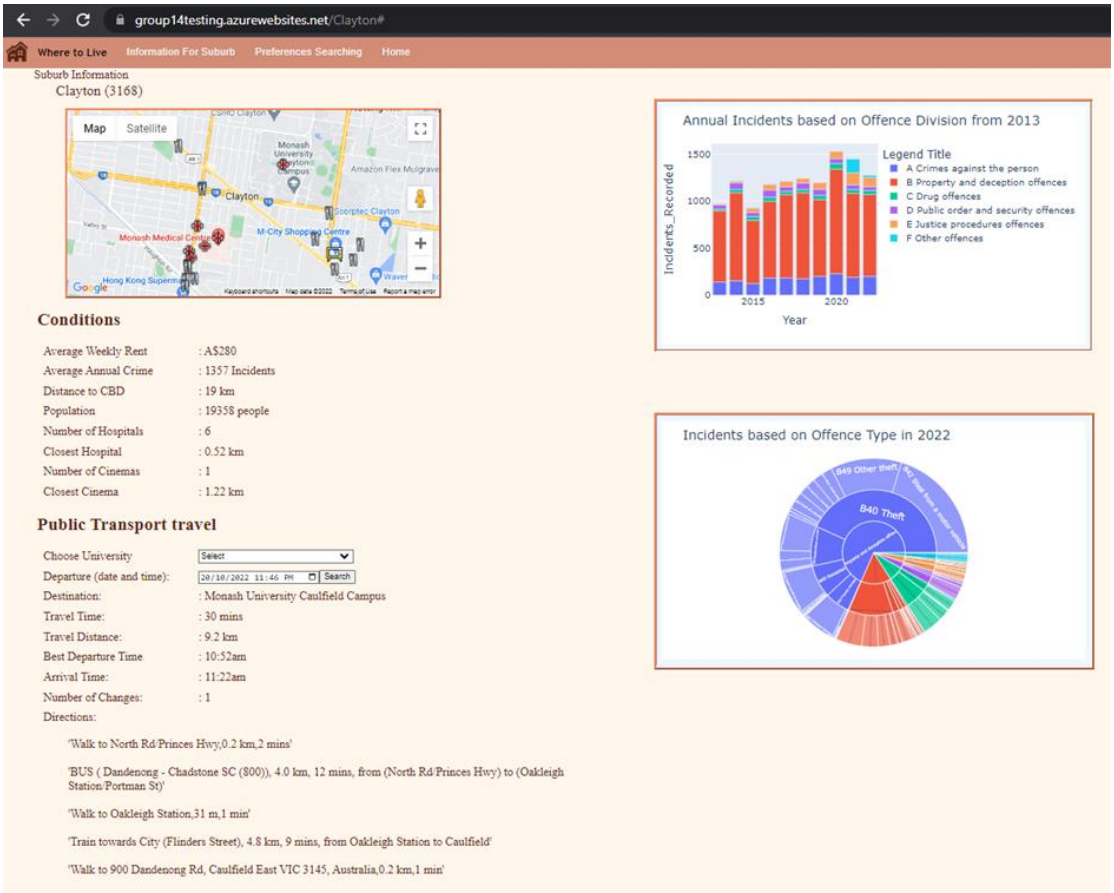


Figure 6.3, Suburb Page

In Figure 6.3, it is the Suburb Page, where it contains some information for the suburb, journey planner to universities, as well as a map with landmarks and visualisations. We

try to show as much information as we can so it will be easier for the users to determine which suburb they will choose.

More detailed descriptions for our project are written in the reports as a part of the deliverables.

2. Software Qualities

a. Robustness

Our team has implemented input validations and extensive testing on the code and website to maintain the robustness of the website. Every user input in all pages is checked and validated, where it will direct the user or show an error message if wrong inputs are written after clicking the submit button. Furthermore, our team has tested the APIs we have used inside the website and made sure that all the data shown is as correct and as expected. Our code has exception handling and extensive conditions to accept multiple kinds of data sent by the APIs. In conclusion, all wrong inputs will be handled by the website and error messages will show and information shown should all be as expected.

b. Security

Unfortunately, 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 cyber-attacks. However, the product doesn't necessarily need a significant amount of security aspects since there are no user's credentials or important information stored in the database. Our database only has information regarding Suburbs, which can be found publicly on the Internet. Moreover, we made the website using Flask and deployed it using Microsoft Azure, which has its

inbuilt guard. Furthermore, we have set a limit for API Requests per day so it will stop on the limit and save our API costs.

c. Usability

The user interface of the website is intentionally made to be simple. The website only has a few pages and users can easily navigate to other pages using the navigation bar on top of the website. Furthermore, the home page has a search bar for users to search a particular suburb. It also has an Information for Suburb page where it has a list of all suburbs where users can choose from. On the labels before the input box, we inform the user what and how to input so the users know what to enter. In addition, all of the input boxes in the website have their own input validation where it informs the user and pops up an error message if they input the unexpected value. Lastly, our team also paid attention to minor details such as placement, mouse hovering, choosing the colour theme, etc to make sure it is easy for users to use the website.

d. Scalability

There is a limited number of users that can interact with the web application in terms of scalability. 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. Furthermore, because of the Azure Free Plan, the website will need 2-3 minutes of loading when it is not used for 20 minutes. Nevertheless, the ability of the web application to cope with users interacting concurrently is still very fast.

Furthermore, to reduce cost, we are limiting the number of API requests to 250 requests for each API daily. Therefore, since every time a user go to a Suburb Page, since it uses one Maps API request and one Places API request, as well as Directions API request if the user uses the journey planner feature, the suburb page will not be accessible after the suburb page has been entered 250 times until the day after. In conclusion, at the

current state of our website, it cannot handle large numbers of users as our budget is extremely limited.

e. Documentation and Maintainability

The source code of our website is written tidily and easily readable. Some parts of the code are written and documented in a certain way so people can easily identify the functionality of parts of the code. However, we acknowledge that some of the naming conventions used for the variables in the code are slightly confusing. Furthermore, as we also delivered data cleaning codes on archive, it is difficult for other users to use as we did not document it and did not manage to tidy the code in time.

3. Sample Source Code

Parts of the source code for our website are mentioned in Appendix C. It shows some of the main functions of the website where it does represent the main functionality of the project's goal itself. Appendix C.1-C.2 shows the main algorithm of our code, how we retrieve data from the database as well as how to calculate scores to choose the best suburbs. Appendix C.4-C.6 shows how we retrieve data from API and show the results to the user, shown in the Suburb Page. Appendix C.5 shows how we are using Maps API with markers to show the location of the suburb as well as landmarks marked on the map.

7.Critical discussion on the Software Project

a. Project execution

Throughout the project timeline, our team undoubtedly went through multiple milestones from product research and ideation towards actual development and refining. Overall, our team's progression was timid at first during the few weeks and was fast-forwarded during the technical implementation from the fourth week.

One of the major reasons behind the initial seizure of progress was slow kickstart, lack of communication and crashes in meeting timelines from changes in allocated sessions from the semester break, resulting in us being one week behind schedule set by Gantt Chart (Appendix D). However, prior thorough high-level technical research with feature demands validated with business stakeholders from the previous semester allowed our team to get back on track quickly, with immediate communications between team members, and support from the relevant faculty's supervisor. We soon realised the importance of conducting more frequent meetings as one of the strategies for keeping things on track. Therefore, we conducted more micro-meetings beyond the main ones as catch ups to ensure everyone was on the same page, as well as focusing on setting the outcome expectations more realistic and relevant to our team members. This resulted in significantly higher involvement and commitment expressed by our team members' effectiveness and timeliness in completing tasks.

Another problem arose when we underestimated the level of expertise required to conduct our deliverables. Despite time delegated for prior self-training in Flask and HTML/CSS, the complexity demanded from our initial proposal surpassed our team members' current level. Through our team members' immense effort to tightly support each other's tasks, we learned to flexibly adapt to new ways of allocating tasks depending on the level of learning required for members to handle alongside their other

academic commitments. It also encouraged us to rethink and reduce unnecessary mini-tasks broken down from the Gantt Chart, allowing higher agility for us.

Our team had the opportunity to grow in both technical and managerial aspects in sustaining the entire project from start to end, through intensive discussions and incremental innovations in project management styles and technical methodologies. This enhanced our confidence in delivering a highly satisfying output to our stakeholders.

b. Deviation from project proposal

Referring to the product user acceptance criteria lodged in our team's project proposal, the majority of requirements relating to suburb selection/recommendation were prioritised to be completed with little to no bugs interrupted. It entirely fulfilled the proposed criteria in serving the user's primary needs when it comes to our deliverables. Extended features such as map display and suburb information were also refined and strictly tested as they were follow-up features complementary to the user's journey on the website after and/or before exploring the main feature. The selection of certain data resources for the main database had been finely verified with high recency from the government's open-source databases, which enhanced the credibility of the information displayed. With website loading time, suburb recommendation system operating time, and website navigation time being less than one minute, the website is highly maintainable with considerably light operations with minimal cloud space required. As a result, it satisfied most, but not all, acceptance criteria from the team's initial proposal.

What deviated from the proposal was predominantly out-of-scope features such as login and storing users' credentials and recent preferences. The elimination of these features resulted from our team's decision to optimise the web application's quality - in terms of speed, effectiveness and ease of usage. From surveying our target users, we realised the importance of the website's ease of access and navigation compared to having an account with tracked data. Considering the limited timeframe and necessity of each proposed attribute, our team had to decide which one to prioritise based on the users. Therefore, the security requirements weren't incorporated as extensively as we initially anticipated. However, we do think it's a rightful decision as incorporating these features will involve higher expectations for cybersecurity, bringing our website at a higher stake of cyber attack if we don't keep frequent checks and extensive protection layers, requiring more time invested than any of our team members can afford. Consequently, we've eliminated the risk of handling unnecessary data storage and processing which further delayed the website's overall performance while being distracted from serving the user's central needs.

c. Limitations and things to consider

Despite having the web application deployed successfully and ready for immediate usage, we also suffered from short-term convenience to long-term sustainability trade-offs, choosing simple-to-implement software to produce our MVP within a restricted time frame. These resulted in architecture-scale disruptions for future architectural innovation in order to handle more complexity and volume of requirements.

Considering our backend infrastructure, Flask is only suitable for light-weighted and simplistic web page architecture. Having a higher potential for security risks and maintenance costs for more complex systems (Stempniak et al., 2022), Flask restrained the capacity of handling production for this webpage. If this project were to progress

further, switching to Django would be considered due to its higher capacity for developing large and complex architectures.

Considering our extended features integration, over-usage of publicly accessible APIs such as Google Maps API, Distance Matrix APIs, and Directions APIs restrained the website from potential scalability, specifically in the volume of users for a specific time interval. Even though publicly used APIs provide the highest accuracy at any instant of calling, overdependence can either lead to restriction limits or additional costs to pursue. However, the convenience of APIs compensated for the time invested in self-developing an individual feature with similar attributes, which we might consider retaining it.

Considering our data warehousing process, due to the involvement of multiple separate datasets manually encoded at different timings, two problems inevitably arose that we were yet to develop a concrete pipeline to resolve: manually unstructured data processing protocols and a lack of automatic updates. If we were given more time, unifying and standardising the data cleansing and updating system would also be our priority. We still consider the feasible implementation of automatic updates due to the demand for high-quality data which weren't publicly accessible with the lack of data quality control for incoming data.

8.Conclusion

The increase in the number of international students arriving in Australia has led to soaring demand for accommodation, while the accommodation options available to educational institutions remain limited. Ignorance of local geographical living conditions can make international students confused when choosing a suitable place to live. In this context, our project provides a digital platform for international students to study in the University of Virginia to recommend suitable suburbs according to user preferences.

The overall project process is divided into two major phases, each of which is three months long. First of all, from March to June, the formation and running-in of new project team members will be completed first, and the preliminary planning and design of the project will be carried out. During this process, the project management method is determined, the project objectives are determined, the preliminary design of the website functions is completed, and the initial version of the UI and UX design prototype is drawn from it. During this period, detailed optimization was carried out step by step to give the final project proposal.

The second phase is from July to October. This stage is based on the project proposal, the implementation of the project. Compared to the first stage, this stage contains more technical requirements. The first is the collection and processing of the required data, and the multi-channel collection is carried out according to the relevant field data planned in the first stage. Process the collected data set and store it for later use. And then for the realisation of the front end, according to the initial prototype design, the optimization of multiple change, choose the design of the final version, and then use the HTML and CSS for the realisation of the static multiple page, and then for some of the backend work, for example with the algorithm implementation of recommendations for computing distance, search and the function and the realisation

of the function of information display, etc. After the front-end page is completed, the front-end and back-end links are made, including API docking.

During this period, I also made two progress presentations and reports on the website, and based on the feedback I got, I optimised and improved the project to ensure customer satisfaction. Before the project was delivered, the site was carefully tested, including algorithm testing, API testing, database retrieval functionality testing, functionality testing in each page, integration testing, system testing, usability testing, performance testing, scalability and security testing. In addition, a user guide has been written so that users can understand how to operate the web page.

Conclusion Overall, the goals of the project have been achieved with the website product, and the website has passed all the test cases. This product provides users with an effective way to find a place to live in Melbourne.

9.Appendix

A. Requirement Traceability Matrix

ID	Requirements	Description/Assumptions/Customer Needs	Category	Source	Status
A.1	Search Function	Search information about each of the suburb's status (crime, public transport, average rent price, etc).	Functional	Client	PASS
A.2	Databases	Database to save the information regarding the suburbs and user's information.	Functional	Client	PASS
A.3	Preference Settings	Set the user preference regarding their needs so the software could find the few best suburbs for the user.	Functional	Client	PASS
A.4	Information Page	Showing the information for the suburbs after the user searches for it or when it's suggested based on their preferences.	Functional	Client	PASS
A.5	Map	The software should be able to show the suburbs on a map with all the public transport stations highlighted.	Functional	Client	PASS
A.6	Visualisation	Create a statistical analysis for crime history in every suburb	Functional	Client	PASS

B.1	Security	The software should be safe and secure.	Non-Functional	Client	PASS
B.2	Quality	The software must be fast, effective, and easy to use.	Non-Functional	Client	PASS
B.3	Maintainable	The software must be easy to maintain and manage.	Non-Functional	Client	PASS

B. Risk Register

The screenshots are to show how the risk register is structured as a table, since the table is too big to be handled on the written documents. Writings of the content will be extracted.

No.	Rank	Risk	Description	Category	Triggers	Root Cause	Potential Responses	Risk Owner	Probability	Impact	Status	Score
1	6	Low adoption rate from customers	Customer finds it more difficult to use the product than expected.	External - Business (customer)	The product might be hard to navigate.	Lack of usage instruction and website design.	Risk avoidance: prepare and present navigation and headlines to visually guide the audience.	Front-end developer	5	8	Open	40
2	12	Low adoption rate from customers	Customer finds it more difficult to use the product than expected.	External - Business (customer)	The selectors might be hard to use and filter.	Lack of usage instruction and/or flexibility with the chosen selectors.	Risk avoidance: research on customer experience journey and make the selector easier to use and filter effectively.	Front-end developer	3	5	Open	15
3	1	Insufficient technical skills	Team members haven't worked on a project like this before and lacks the necessary technical skills	Internal - Technical	Product Owner detects that team is taking too long to begin executing technical tasks	Not providing proper training or guidelines to get the relevant skillsets	Risk mitigation: Provide close support and training (intensively if needed) via expert.	Back-end developer	8	9	Open	72
4	2	Insufficient resources	Not being able to negotiate enough resources from business clients and top management	Internal - Project Management (Resources)	Delay in execution timeline. Data output is incomplete	Lack of timely planning or foreseeing and negotiation	Risk Avoidance: Ensure proper planning (resource breakdown structure) and getting top management support	Product Owner	6	9	Open	54
5	13	Conflict among team members	Team members not being able to reach an agreement for specific execution or tasks	Internal - Project Management (Team)	Inconsistency in deliverables provided. Delay in execution timeline.	Lack of proper communication and involvement of supervisors in providing perspectives	Risk mitigation: Provide immediate communication procedures to resolve conflicts with supervisors.	Product Owner	2	5	Open	10
6	7	Incorrect information from business clients	Business clients provide unclear demands following with vague initiatives requirement	External - Business (Client)	Delay in execution due to confusion. Internal conflict due to different interpretation.	Lack of proper communications and documentation of project flow, leading to confusion for both sides.	Risk avoidance: Prepare and structure clearly the concerns versus the expected delivery from the clients, given them sufficient context for consideration and confirmation.	Product Owner	5	7	Open	35
7	14	Underestimation of budget	Underestimation of expense for project activities and resources.	External - Business (Client)	Unexpected arising costs to cover expenditures or upgrade resources	Originated from business client's abrupt change in requirements.	Risk acceptance: Cut cost by leveraging free resources during this period.	Product Owner	1	3	Open	3
8	4	Delay in approval by business clients of design proposal	Business clients can respond to approval report later than anticipated response deadline.	External - Business (Client)	Business client being not responsive in communicating about the project as planned	Lack of communication and agreed planning.	Risk avoidance: Ensure proper reminder and/or advanced notice. Use calendar invites to block in the time for business clients.	Product Owner	6	8	Open	48
9	10	Expertise dependency	Only one team member has the required skill to execute a certain task, but is having conflict with other team members and potentially leave the team.	Internal - Project Management (HR)	Slow work delivery, passive-aggression during communications	Different perspectives	Risk transference: If the task doesn't involve confidential data, potential outsourcing to third-party in handling tasks to leverage workload.	Product Owner	4	7	Open	28
10	7	Web component unmatched with back-end functions	Web component of the application can't handle/cooperate with back-end functionalitites	Internal - Technical	Delays in delivering full testing versions	Front-end and back-end team working independently, not co-working in agile.	Risk mitigation: Increase frequency of meetings and communication between these team to enhance corporation.	Product Owner	7	5	Open	35
11	5	Functions elimination	Specific functions might be removed from the main program.	Internal - Technical	Functions being dropped in the middle of execution, moving onto new functions.	Due to change in requirement or difficulty to execute in real life.	Risk mitigation: Consistent communication with business clients and supervisors throughout the making process to limit the chance of it being made redundant.	Product Team	9	5	Open	45
12	2	Fast-forwarded workpace	Work timeline might be fast-forwarded compared to expected planning.	Internal - Project Management	Tasks completed earlier than due date with acceptable quality.	Pre-built technical expertise	Risk exploitation: Provide time for team members to upskill whenever possible to fast forward the process.	Product Owner	6	9	Open	54
13	11	External partnership	Property/Real-estate companies express their interest of partnering to promote the product for their customers.	External - Business (Partners)	Receiving emails and partnering proposals from them	They see potentials and values delivered from our product.	Risk enhancement: Prepare potential training and documents in case of emerging partnership opportunities.	Product Owner	3	7	Open	21
14	7	Technology advancement	Arrival of new technology that can minimise friction and increase in adaptive implementation.	External - Technical	New/Existing companies releasing products fitting with the project's need.	External companies develop the product to suit with their target customers' demand, which accidentally fits with the project.	Risk acceptance: Be prepared to always welcome and try new things, because we don't know what benefit such advancement can bring.	Product Owner	7	5	Open	35

C. Parts of Main Source Code

```python
@app.route('/filter',methods =["GET", "POST"])
def your_view():

    your_list = []
    your_list5=[]
    con = sql.connect("site.db")
    con.row_factory = sql.Row
    cur = con.cursor()


    if request.method == "POST":
        counter = 0
        uni = request.form.get("uni")
        max_distance = request.form.get("max_distance")
        cinema = request.form.get("cinema")
        hospital = request.form.get("hospital")
        rent = request.form.get("rent")

        print(max_distance,flush=True)

        if cinema.lower() == 'yes':
            cinema = '>= 1'
        elif cinema.lower() == 'no':
            cinema = '<= 99'

        if hospital.lower() == 'yes':
            hospital = '>= 1'

        elif hospital.lower() == 'no':
            hospital = '<= 99'
```

Appendix C1. Filter page, Retrieving user inputs and some input handling

```
        for i in range (len(for_temp)):
            wsm_rent = 'select "Median Rent" from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
            cur.execute(wsm_rent)
            rows = cur.fetchall();
            for j in rows:
                wsm_rent = j[0]

            wsm_numberhospital = 'select "Number of Hospitals" from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
            cur.execute(wsm_numberhospital)
            rows = cur.fetchall();
            for j in rows:
                wsm_numberhospital = j[0]

            wsm_numbercinemas = 'select "Number of Cinemas" from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
            cur.execute(wsm_numbercinemas)
            rows = cur.fetchall();
            for j in rows:
                wsm_numbercinemas = j[0]

            wsm_closesthospital = 'select "Closest Hospital" from suburb_distance_updated where "Suburb/Town Name" = "'+for_temp[i]+'"'
            cur.execute(wsm_closesthospital)
            rows = cur.fetchall();
            for j in rows:
                wsm_closesthospital = j[0]

            wsm_closestcinema = 'select "Closest Cinema" from suburb_distance_updated where "Suburb/Town Name" = "'+for_temp[i]+'"'
            cur.execute(wsm_closestcinema)
            rows = cur.fetchall();
            for j in rows:
                wsm_closestcinema = j[0]
```

```
                wsm_distancetouni = 'select "'+uni+'" from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
                cur.execute(wsm_distancetouni)
                rows = cur.fetchall();
                for j in rows:
                    wsm_distancetouni = j[0]


                wsm_crime = 'select "Incidents Recorded" from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
                cur.execute(wsm_crime)
                rows = cur.fetchall();
                for j in rows:
                    wsm_crime = j[0]

                wsm_population = 'select population from v3 where "Suburb/Town Name" = "'+for_temp[i]+'"'
                cur.execute(wsm_population)
                rows = cur.fetchall();
                for j in rows:
                    wsm_population = int(     str(j[0]).replace(",","")     )
```

```
calculate = -1.3*float(wsm_distancetouni)-
0.1*float(wsm_closestcinema)-0.1*float(wsm_closesthospital)-
0.1*float(wsm_distancetocbd)-5*(int(wsm_crime)/int(wsm_population))-float(wsm_rent)/10
```

Appendix C2. Codes to retrieve individual suburbs data and parts of score calculation

```
    checker = 'select "Suburb/Town Name" from v3 where "Suburb/Town Name" = "'+suburb+'"'
    cur.execute(checker)
    rows = cur.fetchall();
    for i in rows:
        check_exist = i
    if check_exist == None:
        return render_template('notavailable.html')


    show = 'select "Median Rent","Incidents Recorded", Postcode, population, "Number of Hospitals","Number of Cinemas" from v3 where "Suburb/Town Name" = "'+suburb+'"'
    cur.execute(show)
    rows = cur.fetchall();
    for j in rows:
        show = [j[0],math.ceil(j[1]/9),j[2],j[3],j[4],j[5]]

    url_uni = 'select University from uni_loc'
    cur.execute(url_uni)
    rows = cur.fetchall();
    for j in rows:
        list_of_uni.append(j[0])


    url_cinema = 'select lat, long, name, rating, user_ratings_total, Suburb from cinema where Suburb = "'+suburb+'"'
    cur.execute(url_cinema)
    rows = cur.fetchall();
    for i in rows:
        t = (i[0],i[1],i[2])
        your_list.append(t)

    url_suburb = 'select lat,lng from v1 where "Suburb/Town Name" =  "'+suburb+'"'
    cur.execute(url_suburb)
    rows = cur.fetchall();
    for i in rows:
        suburb_coor = (i[0],i[1])

    url_hospital = 'select "Formal Name", Suburb, lat, lng from hospital where Suburb = "'+suburb+'"'
    cur.execute(url_hospital)
    rows = cur.fetchall();
```

Appendix C3. Individual Suburb page, retrieving information for the suburb

```
restaurant_list=[]
url33="https://maps.googleapis.com/maps/api/place/nearbysearch/json?keyword=restaurant&location="+latlng[0]+"%2C"+latlng[1]+"&radius=1500&type=restaurant&key=AIzaSyBYkIz-pq
url33 = requests.get(url33).json()
for i in range(len(url33['results'])):
    restaurant_list.append([
        url33['results'][i]['geometry']['location']['lat'],
        url33['results'][i]['geometry']['location']['lng']
        ]
        )
```

Appendix C4. Places API to retrieve nearby 20 restaurants data

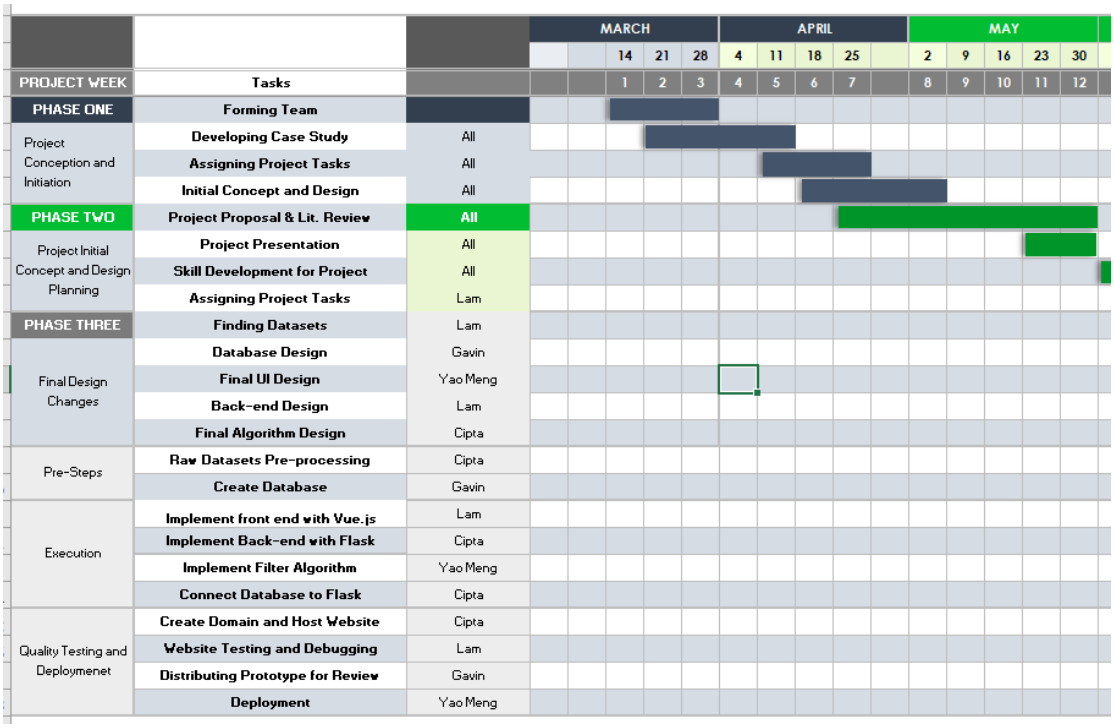
```
<script type="text/javascript">
    function myFunc(vars) {
        return vars
    }

    function initMap() {
        var iconBase = 'https://maps.google.com/mapfiles/kml/shapes/'
        myCinema = myFunc({{ your_list | tojson }});
        myHospital = myFunc({{ your_list2 | tojson }});
        myRestaurant = myFunc({{ restaurant_list | tojson }});
        const uluru = { lat: parseFloat('{{suburb_coor[0]}}'), lng: parseFloat('{{suburb_coor[1]}}') };
        const map = new google.maps.Map(document.getElementById("map"), {
            zoom: 14,
            center: uluru,
        });
        for (i = 0; i < myCinema.length; i++) {
            const marker = new google.maps.Marker({
                position: { lat: parseFloat(myCinema[i][0]), lng: parseFloat(myCinema[i][1]) },
                map: map,
                icon: { url: iconBase + 'electronics.png',scaledSize: new google.maps.Size(25, 25)}
            });
        }
        for (i = 0; i < myHospital.length; i++) {
            const marker = new google.maps.Marker({
                position: { lat: parseFloat(myHospital[i][2]), lng: parseFloat(myHospital[i][3]) },
                map: map,
                icon: { url: 'http://maps.google.com/mapfiles/kml/pal3/icon46.png', scaledSize: new google.maps.Size(25, 25) }
            });
        }
        for (i = 0; i < myRestaurant.length; i++) {
            const marker = new google.maps.Marker({
                position: { lat: parseFloat(myRestaurant[i][0]), lng: parseFloat(myRestaurant[i][1]) },
                map: map,
                icon: { url: iconBase + 'dining.png',scaledSize: new google.maps.Size(22, 22)}
            });
        }
    }


</script>
```

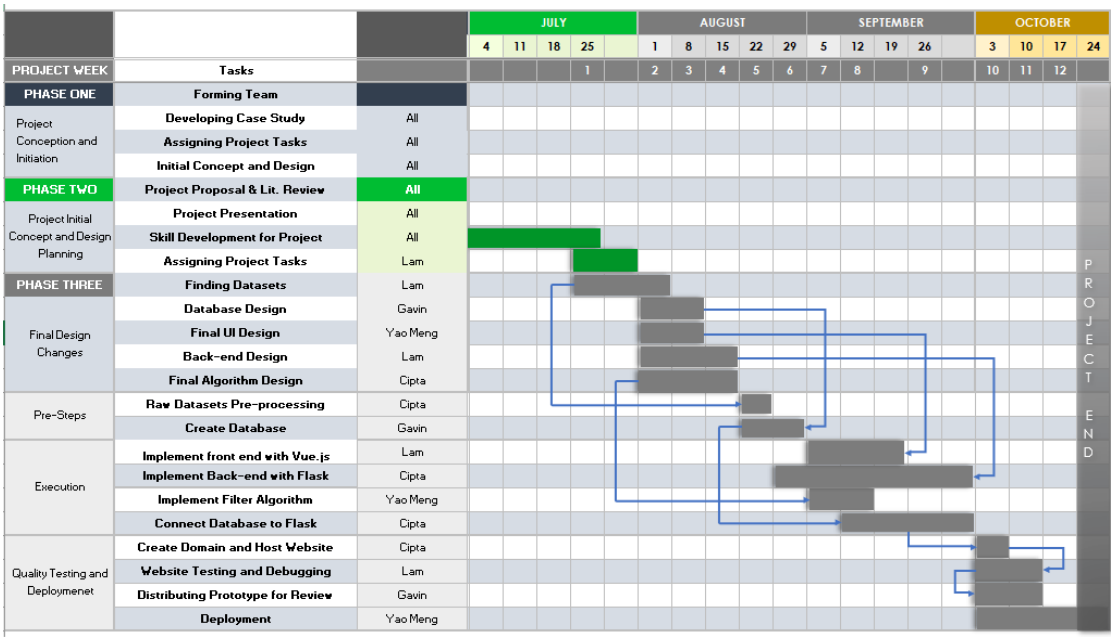
Appendix C5. Maps API to show maps of the suburbs with markers of landmarks

D. Gantt Chart



			MARCH			APRIL				MAY				
			14	21	28	4	11	18	25	2	9	16	23	30
PROJECT WEEK	Tasks		1	2	3	4	5	6	7	8	9	10	11	12
PHASE ONE	Forming Team													
Project Conception and Initiation	Developing Case Study	All												
	Assigning Project Tasks	All												
	Initial Concept and Design	All												
PHASE TWO	Project Proposal & Lit. Review	All												
Project Initial Concept and Design Planning	Project Presentation	All												
	Skill Development for Project	All												
	Assigning Project Tasks	Lam												
PHASE THREE	Finding Datasets	Lam												
Final Design Changes	Database Design	Gavin												
	Final UI Design	Yao Meng												
	Back-end Design	Lam												
	Final Algorithm Design	Cipta												
Pre-Steps	Raw Datasets Pre-processing	Cipta												
	Create Database	Gavin												
Execution	Implement front end with Vue.js	Lam												
	Implement Back-end with Flask	Cipta												
	Implement Filter Algorithm	Yao Meng												
	Connect Database to Flask	Cipta												
Quality Testing and Deploymenet	Create Domain and Host Website	Cipta												
	Website Testing and Debugging	Lam												
	Distributing Prototype for Review	Gavin												
	Deployment	Yao Meng												

Appendix D1. Schedule of the first part of the project



Appendix D2. Schedule of the second part of the project

E. Final Report Contribution Annexes

Proposal part	% total	Lam	Cipta	Yao	Gavin
Cover sheet	2				2
Introduction	8				8
Background	10				10
Outcomes	20	20			
Methodology	15			15	
Software deliverables	20		20		
Critical discussion	10	10			

Conclusion	5				5
Code Listing in Appendix	5		5		
References	5			5	
Style and presentation	5			5	
Total	100	30	25	25	25

10.References

John. (2020, October 8). *Australian-Suburbs*. GitHub. https://github.com/michalsn/australian-suburbs/tree/master/data

Latest Victorian Crime Data. (2022, June 30). Crimestatistics. https://www.crimestatistics.vic.gov.au/crime-statistics/latest-victorian-crime-data

Victorian Hospital Lists. (2022). Healthcollect. https://www.healthcollect.vic.gov.au/HospitalLists/MainHospitalList.aspx

Rental Report - Quarterly: Moving Annual Rents by Suburb. (2022, June). Discover. https://discover.data.vic.gov.au/dataset/rental-report-quarterly-moving-annual-rents-by-suburb

Directions API Overview. (2022, May 13). Google. https://developers.google.com/maps/documentation/directions/overview

Places API Overview. (2022, May 13). Google. https://developers.google.com/maps/documentation/places/web-service/overview

Maps API Overview. (2022, May 13). Google. https://developers.google.com/maps/documentation/javascript/overview

Stempniak, A., & Semik, W. (2021). *Flask vs. Django: Which Python Framework Is Better for Your Web Development?* Stxnext. https://www.stxnext.com/blog/flask-vs-django-comparison/

Mccue, C. (2007). Process Models for Data Mining and Analysis. *ScienceDirect*, 45–66. https://www.sciencedirect.com/science/article/pii/B978075067796750026X

Martha , V. S. (2015). Big Data Processing Algorithms. *Big Data*, *11*, 61–91. https://link.springer.com/chapter/10.1007/978-81-322-2494-5_3

Metcalfe, L. (2015, August 5). *Get the Facts the Property Ad Left Out*. Microburbs. https://www.microburbs.com.au/

Partida, D. (2020, December 4). *5 Data Storage Concepts Serious Website Owners Should Understand*. Torque. https://torquemag.io/2020/12/5-data-storage-concepts-serious-website-owners-should-understand/

Troiola, W. (2021). *Crime Statistics by Suburb*. Findmyrates. https://www.findmyrates.com.au/crime-statistics-by-suburb/

Simosa, F. (2018). *Risk Response Strategies for Positive Risks*. Projectcubicle. https://www.projectcubicle.com/risk-response-strategies-positive-risks/

Usmani F. (2021) *Stakeholder Mapping: A Complete Guide with Examples*. PM Study Circle. https://pmstudycircle.com/stakeholder-mapping/

Gibson, C. (2021, August 30). *How Do You Know If a Suburb Is Booming? It Comes down to These 5 Indicators*. Openagent. https://www.openagent.com.au/blog/how-data-indicators-can-help-you-identify-boom-suburbs

Glossop, A. (2021). *What Is a Risk Management Strategy?* Ideagen. https://www.ideagen.com/thought-leadership/blog/what-is-a-risk-management-strategy

Harrin, E. (2019, December 12). *Response Strategies for Positive Risk*. Thebalancecareers. https://www.thebalancecareers.com/positive-risk-response-strategies-2779621#:~:text=There%20are%20also%20formal%20management,share%2C%20enhance%2C%20and%20accept

Bansal, S. (2014). *How To Handle Negative Risks-Learn Negative Risk Response Strategies*. Izenbridge. https://www.izenbridge.com/blog/how-to-handle-negative-risk-learn-negative-risk-response-strategies/#:~:text=The%20five%20basic%20strategies%20to,basis%20of%20the%20risk%20exposure

Bitrix 24 Team (2021). *Project Management Communication Plan*. Bitrix 24. https://www.bitrix24.com/about/blogs/tasks-projects/how-to-create-a-communication-plan-for-project-management1.php

Baldwin, N. (2020). *Learning System for PMP® Exam Preparation*. Certwise. https://www.certwise.com/wp-content/uploads/2017/03/CW_PMP_Reading-Sample.pdf

Contal, E. (2020, May 10). *What Are Today's Top Recommendation Engine Algorithms?* Itnext. https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639

Literature Review:
Ahmad, W., Zia, A., & Khalid, U. (2013, June 10). Scientific Research. Retrieved from https://pdfs.semanticscholar.org/0bdf/934ce6d16b17b6b81d6f670bf44fc3e81356.pdf

Pansambal, S., Iyer, N., Meherkar, V., & Sharma, P. (2016, March). International Journal of Advanced Research in Computer and Communication Engineering. Retrieved from Ijarcce: https://www.ijarcce.com/upload/2016/march-16/IJARCCE%20120.pdf

Ericho, Rolan. (2021). Paper Literature Review Implementasi Google Maps API. https://www.researchgate.net/publication/354716890_Paper_Literature_Review_Implementasi_Google_Maps_API/citation/download

Nielsen, J. (2000). Designing Web usability: The practice of simplicity. Indianapolis: New Riders.

Higashino, K,. Kelly, C., Ishitani, H., and Nakamura, K. (2001). Usability of e-learning in educational Web sites. The 26th Zenkoku Taikai Ronbunshu (26). pp. 199-200. Osaka, Japan: Japanese Society of Information and Systems in Education

N. Ngadiman, S. Sulaiman and W. Mohd Nasir Wan Kadir, "A systematic literature review on attractiveness and learnability factors in Web applications," 2015 IEEE Conference on Open Systems (ICOS), 2015, pp. 22-27, doi: 10.1109/ICOS.2015.7377272.

Garett, R., Chiu, J., Zhang, L., & Young, S. (2016). A Literature Review: Website Design and User Engagement. Online journal of communication and media technologies, 6(3), 1-14. http://dx.doi.org/10.29333/ojcmt/2556 Retrieved from

https://escholarship.org/uc/item/64b019qh

Lynch, P., and Horton, S. (1999). Web style guide. New Haven: Yale University Press