

EXPLORECAN

Final Report

Rahulnikki Yerrapothu

Anusha Adem

Sai Prathyusha

Sahithi

Vikranth

Group 7

Contents

[1. Overview 2](#_Toc141398508)

[2. Vision 2](#_Toc141398509)

[3. Mission 2](#_Toc141398510)

[4. Project Goals and Objectives 2](#_Toc141398511)

[5. Challenges 2](#_Toc141398512)

[6. Business Model 4](#_Toc141398513)

[7. Cost Estimation 4](#_Toc141398514)

[7.1 Cost distribution and quick break down 4](#_Toc141398515)

[8. RACI Matrix 5](#_Toc141398516)

[9. Stakeholders Identification and analysis 5](#_Toc141398517)

[10. Work Break Down Structure 6](#_Toc141398518)

[11. Individual Technical Approach 14](#_Toc141398519)

[11.1 System Architecture 14](#_Toc141398520)

[11.2 Data Collection 15](#_Toc141398521)

[11.3 Data Pre-Processing 17](#_Toc141398522)

[11.4 Feature Extraction 17](#_Toc141398523)

[11.5 Recommendation Generation 17](#_Toc141398524)

[11.6 User Interface 18](#_Toc141398525)

[11.7 Evaluation metrics 18](#_Toc141398526)

[11.8 Conclusion 19](#_Toc141398527)

[12. Project Timeline 19](#_Toc141398528)

[13. Technologies and Tools 20](#_Toc141398529)

[14. Current State of Project: 21](#_Toc141398530)

[15. Sample Questionnaires for chatbot 26](#_Toc141398531)

[16. LUIS for creating recommendation systems. 28](#_Toc141398532)

[17. Designing the conversation workflow 28](#_Toc141398533)

[18. Challenges 28](#_Toc141398534)

[20. References 29](#_Toc141398535)

# Overview

"In Canada, restaurants and tourist attractions face a competitive market, with a large number of options available to customers. As a result, it can be challenging for these businesses to differentiate themselves and attract customers. This is further complicated because customers have different preferences and seek personalized recommendations. There is a need for a solution that can provide customers with relevant and personalized recommendations, while also helping restaurants and tourist attractions to understand their customers better and make informed business decisions."

The Canadian restaurant and tourist attraction industry face several challenges that need to be addressed. A recommendation system can provide a solution that benefits customers and businesses by improving customer experience, increasing sales and revenue, and offering a competitive advantage. By addressing these challenges, the industry can overcome its limitations and provide better services to its customers.

# Vision

Our vision is to create a cutting-edge recommendation system.

That revolutionizes the way people discover Canadian restaurants and tourist attractions, providing them with a personalized and delightful experience.

# Mission

1. Our mission is to develop a comprehensive and user-friendly recommendation system.
2. That leverages the latest technologies, including data analysis and machine learning.
3. To help users find the best options for their needs, leading to increased satisfaction and engagement.

# Project Goals and Objectives

1. Improving the user experience
2. Increasing engagement and retention
3. Driving revenue and business growth
4. Generating data insights

The goal of this recommendation system for Canadian restaurants and tourist attractions is to create a

more personalized and enjoyable experience for users while also driving business growth and generating valuable data insights.

# Challenges

1. Data collection and quality
   1. One of the biggest challenges we have faced is collecting and maintaining high-quality data on restaurants and tourist attractions.
2. Cold-start problem
   1. Occurs when the recommendation system has insufficient data on new users or items.
3. Scalability
   1. As the number of users and items in the system grows, we may face scalability challenges.
4. User engagement
   1. If the user interface is not intuitive or visually appealing, users may be less likely to interact with the system and provide feedback on the recommendations.

# Business Model

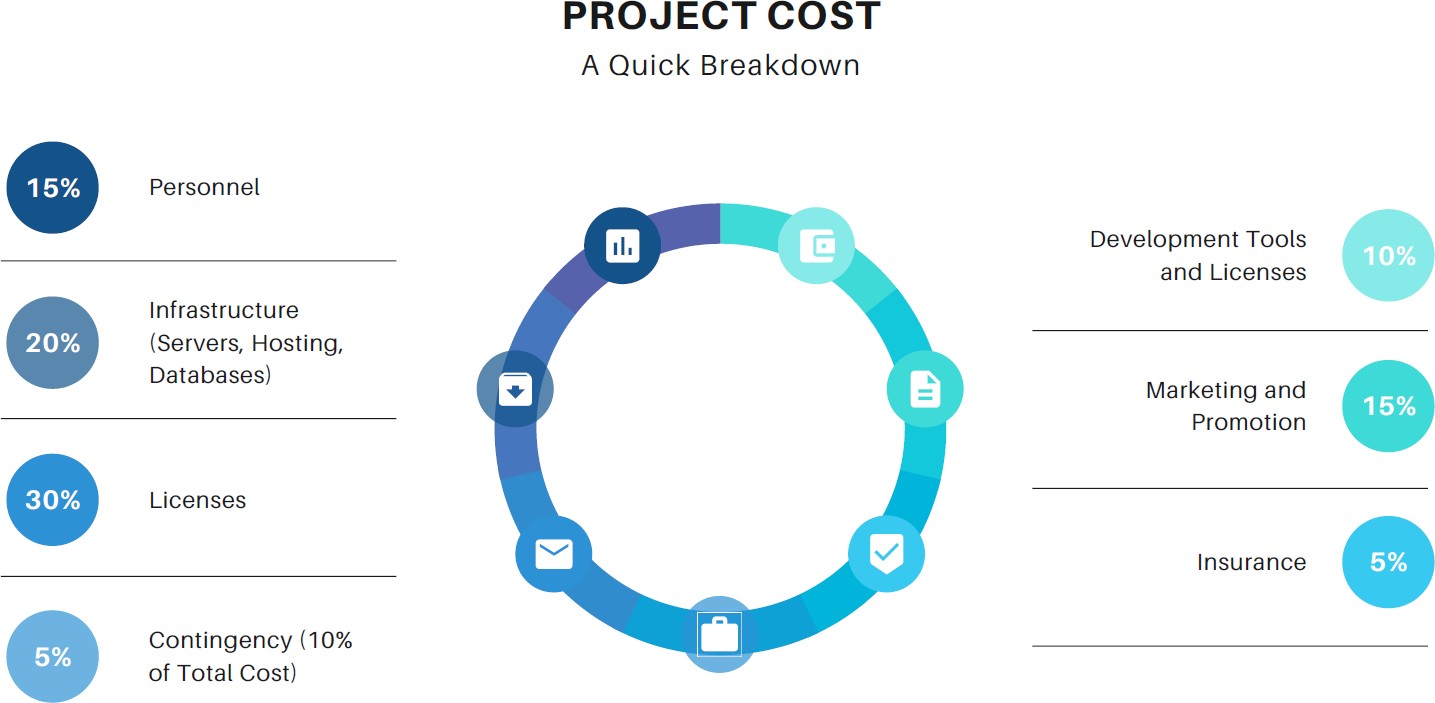
Pathways to generate our revenue.

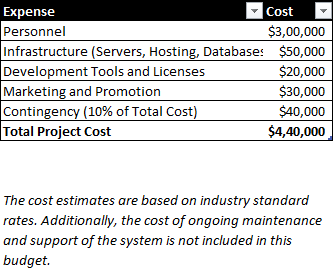
* 1. Advertising: You could generate revenue by selling advertising space to restaurants or tourist attractions on your platform. For example, you could allow restaurants to promote their business to users through sponsored recommendations or featured listings.
  2. Commission-based sales: You could earn a commission on sales made through your platform, such as booking a tour or making a restaurant reservation.
  3. Subscription model: You could charge a subscription fee to users for access to premium features or personalized recommendations.
  4. Data licensing: You could sell access to the data and insights generated by your recommendation system to other businesses or organizations, such as travel companies or food and beverage brands.
  5. Partnerships: You could partner with restaurants, tourist attractions, or other businesses to offer exclusive discounts or promotions to users of your platform.

# Cost Estimation

* 1. Data Collection and Preparation: This includes the cost of acquiring and cleaning the data required for building the recommendation system, which can include data from customer surveys, restaurant sales, and tourist attraction visitor numbers.
  2. Development and Implementation: This includes the cost of hiring software engineers, data scientists, and other specialists to build and implement the recommendation system. This can range from a few thousand to hundreds of thousands of dollars, depending on the system's complexity.
  3. Infrastructure and Hosting: This includes the cost of hosting the recommendation system on cloud-based servers, as well as any additional hardware and software required for running the system.
  4. Maintenance and Upgrades: This includes the cost of maintaining and updating the system over time, including fixing bugs, adding new features, and updating the data used for recommendations.
  5. The total cost of creating a recommendation system for Canadian restaurants or tourist attractions can range from tens of thousands of dollars to hundreds of thousands of dollars, depending on the scale and complexity of the system.

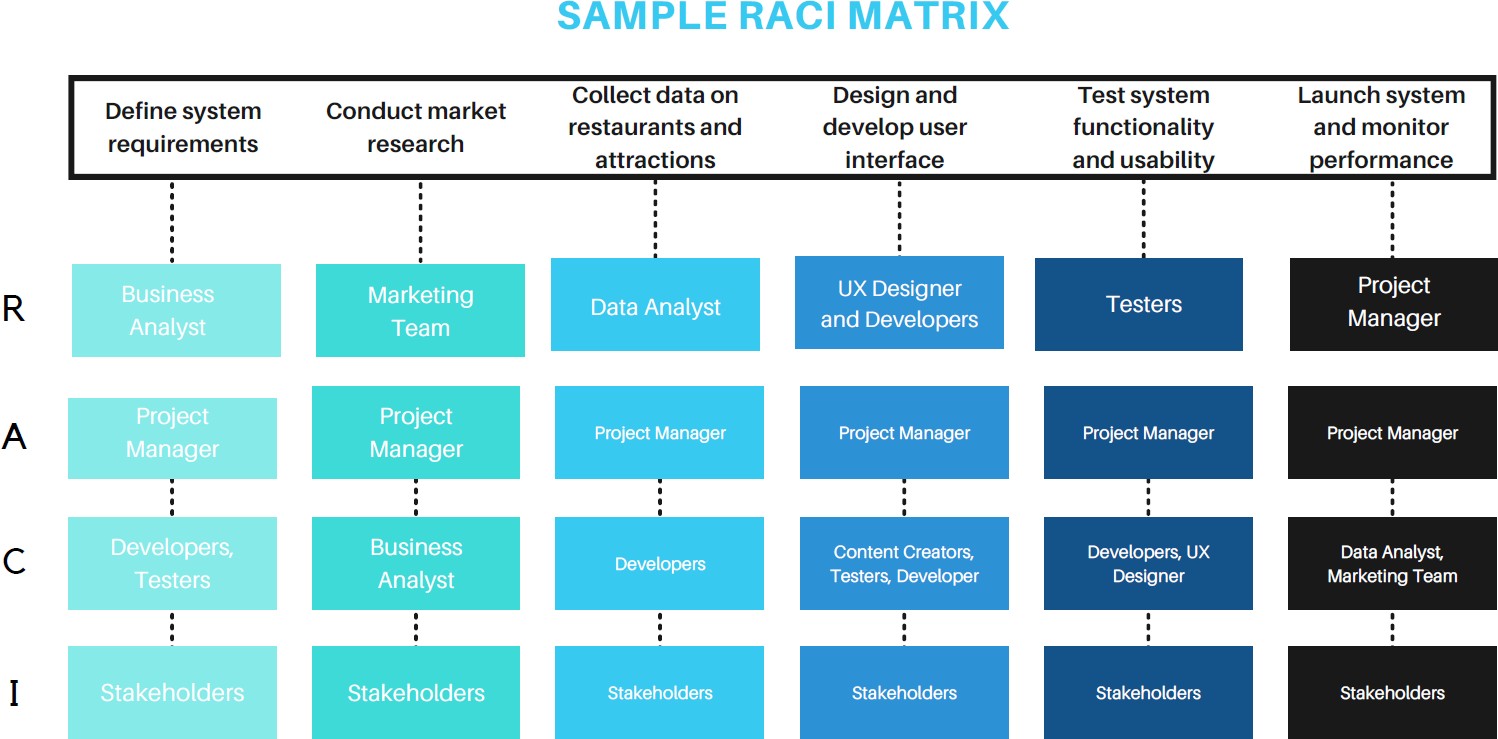
## 7.1 Cost distribution and quick break down





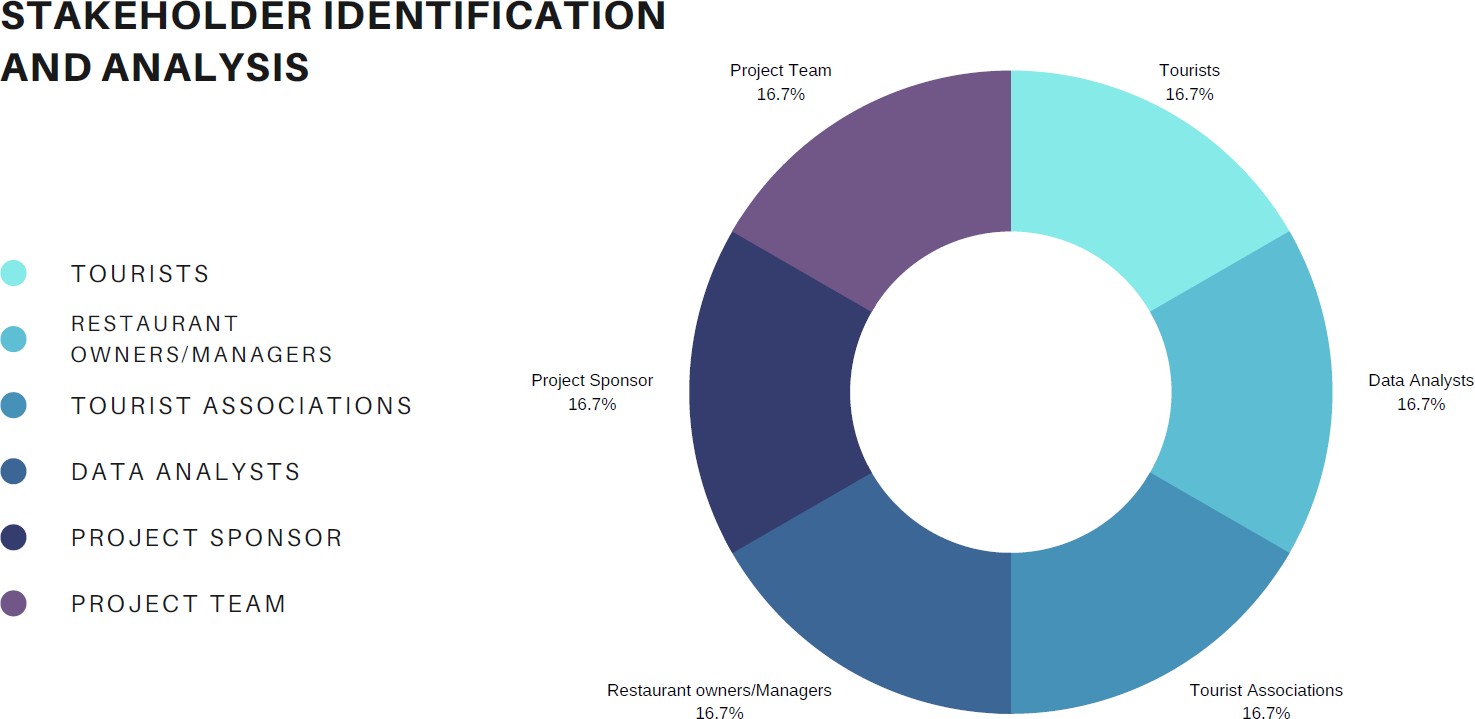
# RACI Matrix

A RACI matrix can be a useful tool for creating a recommendation system for Canadian restaurants and tourist attractions by ensuring that everyone on the project team understands their roles and responsibilities, improving communication, increasing accountability, identifying potential bottlenecks, and facilitating project management.

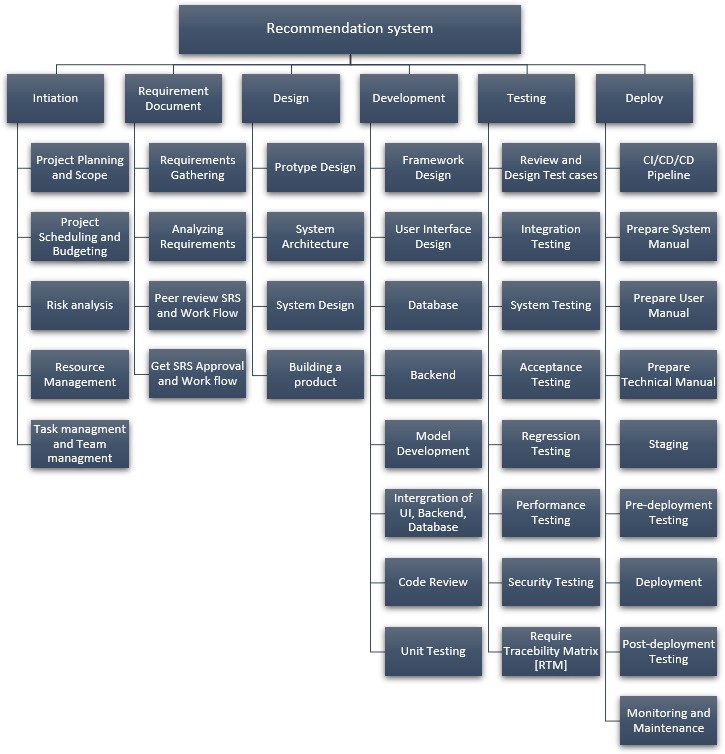


# Stakeholders Identification and analysis

* 1. Restaurants and Tourist Attractions: The primary stakeholders, as they will be the ones using the recommendation system and benefiting from increased sales and improved customer experience.
  2. Customers: Another vital stakeholder, as they will be the ones receiving the recommendations and providing feedback to improve the system.
  3. Technology Vendors: These companies provide the technology and resources required to build and maintain the recommendation system.
  4. Data Scientists and Engineers: These professionals are responsible for building and implementing the recommendation system and ensuring that it provides accurate and relevant recommendations.
  5. Investors: If the recommendation system is being developed as a commercial product, investors may be involved to provide funding for development and growth.
  6. Regulators: Depending on the data being used for recommendations, regulators such as privacy protection agencies may be involved to ensure that the data is being used responsibly and ethically.



# Work Break Down Structure



WBS: Project Initiation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Activity Name** | **Description** | **Mapping Activity** | **Estimated Cost** | **Person Responsible** | **Risk** | **Success Criteria** |
| **1.** | **Initiation** | Project Initiation  communication starts with  Project  Manager and Stakeholder. Will do the Feasibility  Study | Business Case | - | Project Manager |  | Project Goal should be  clear |
| **1.1.** | **Project Planning and Scope** | Prepared the Project Scope and Charter | Project Charter/Review  the Scope | - | Project  Manager and Business  Analyst |  | Scope should be clear |
| **1.2.** | **Project Scheduling/Budgeting** | Based on the Project Scope, Project  Manager will decide the Cost  and Time | Project Scope | - | Project Manager, Business Analyst |  | Scope Finalized |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1.3.** | **Risk analysis** | Identify all the Internal and External Risks and keep track of those Risks  into Documents | Project Charter/Project  Scope | - | Project Manager, Business Analyst,  Stakeholders |  | Risks should be  documented |
| **1.4.** | **Resource Management** | Identify the Resources headcounts to  this project | Finalizing Scope | - | Project, Development and Testing  Manager |  | Utilizing the available resources  efficiently |
| **1.5.** | **Task and Team Management** | A solid communication plan is setup to resolve  bottlenecks  quickly. | All templates ready and Team members onboarded | - | Project  Manager and Project Charter |  | Task  Assessment and Task Division |

WBS: Requirement Document

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Activity Name** | **Description** | **Mapping Activity** | **Estimated Cost** | **Person Responsible** | **Risk** | **Success Criteria** |
| **2.** | **Requirement** | This is the | All templates | - | Project |  | Final Version of |
|  | **Document** | phase where | ready & |  | Manager, | End Product will |
|  |  | we document | Scope |  | Business | be good |
|  |  | all high-level | Finalized |  | Analyst |  |
|  |  | and low-level |  |  |  |  |
|  |  | Requirement |  |  |  |  |
|  |  | in detail |  |  |  |  |
| **2.1.** | **Requirements** | Note down all | All | - | Business |  | List assumptions |
|  | **Gathering** | the | Requirement |  | Analyst | and requirements |
|  |  | Stakeholders | templates |  |  | are ready |
|  |  | Requirement | ready & |  |  |  |
|  |  |  | Scope |  |  |  |
|  |  |  | Finalized |  |  |  |
| **2.2.** | **Analysing** | Team will | All | - | Project |  | List assumptions |
|  | **Requirements** | analyze all | Requirement |  | Manager and | and requirements |
|  |  | Requirement | templates |  | Business | are ready |
|  |  | and discussed | ready & |  | Analyst |  |
|  |  | with all | Identify the |  |  |  |
|  |  | Stakeholders | Requirement |  |  |  |
|  |  | to add or |  |  |  |  |
|  |  | update |  |  |  |  |
|  |  | anything in |  |  |  |  |
|  |  | the |  |  |  |  |
|  |  | requirement |  |  |  |  |
|  |  | document |  |  |  |  |
| **2.3.** | **Draft SRS and** | Prepare the | All | - | Business |  | Keep Track of all |
|  | **Workflow** | Draft version | Requirement |  | Analyst | Requirement and |
|  |  | of the | templates |  |  | understand the |
|  |  | Specification | ready & |  |  | end-to-end flow |
|  |  | Requirement | Identify the |  |  | to get more |
|  |  | document and | Requirement |  |  | understanding for |
|  |  | draw the flow |  |  |  | development |
|  |  | diagram |  |  |  |  |
| **2.4.** | **Peer review SRS** | Based on the | Requirement | - | Project |  | Avoid |
|  | **(Software** | Project Scope, | Draft version |  | Manager, | Misunderstanding |
|  | **Requirement** | Project | needs to be |  | Business | in requirement |
|  | **Specifications)** | Manager will | ready |  | Analyst | beginning |
|  | **and Workflow** | decide the |  |  |  |  |
|  |  | Cost and Time |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **2.5.** | **Get SRS Approval and Workflow Approval** | Identify all the Internal and External Risks and keep  track of those Risks into  Documents | All Requirement  ready & approved | - | Project Manager, Business Analyst,  Stakeholders |  | Will not deviate from the project |

WBS: PROJECT DESIGN

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.n o** | **Activity Name** | **Description** | **Mapping Activity** | **Estimate d Cost** | **Person Responsibl**  **e** | **Ris k** | **Success Criteria** |
| **3.** | **Design** | The Design phase models the way | Functional | - | Software |  | Developmen |
|  |  | a software application will work | requiremen |  | Architect | t phase will |
|  |  |  | t |  |  | be smooth |
|  |  |  | documents |  |  | and Final |
|  |  |  |  |  |  | Version of |
|  |  |  |  |  |  | End Product |
|  |  |  |  |  |  | will be good |
| **3.1.** | **Protype** | Users | Functional | - | Software |  | Developmen |
|  | **Design** | evaluate the developer's proposal | requiremen |  | Architect | t phase will |
|  |  | s and test them before | t |  |  | be smooth |
|  |  | implementation. This helps the | documents |  |  | and Final |
|  |  | user understand specific needs |  |  |  | Version of |
|  |  | that the developer may not have |  |  |  | End Product |
|  |  | considered when designing the |  |  |  | will be good |
|  |  | product |  |  |  |  |
| **3.2.** | **System** | System architecture describes | End to End | - | Software |  | Developmen |
|  | **Architecture** | its main components, connections | Flow |  | Architect | t phase will |
|  |  | (structures) and how they interact | Diagram in |  |  | be smooth |
|  |  | with each other. | SRS |  |  | and Final |
|  |  |  |  |  |  | Version of |
|  |  |  |  |  |  | End Product |
|  |  |  |  |  |  | will be good |
| **3.3.** | **System** | It includes the design of | End to End | - | Software |  | Developmen |
|  | **Design** | application, network, database, | Flow |  | Architect | t phase will |
|  |  | user interface and | Diagram in |  |  | be smooth |
|  |  | computer interfaces | SRS |  |  | and Final |
|  |  |  |  |  |  | Version of |
|  |  |  |  |  |  | End Product |
|  |  |  |  |  |  | will be good |
| **3.4.** | **Building a** | Final approval from the | All Design | - | Stakeholders |  | Developmen |
|  | **product and** | stakeholders about design of | documents |  |  | t phase will |
|  | **Getting** | product before development | completed |  |  | be smooth |
|  | **Stakeholder** | phase starts |  |  |  | and Final |
|  | **s approval** |  |  |  |  | Version of |
|  |  |  |  |  |  | End Product |
|  |  |  |  |  |  | will be good |

WBS: PROJECT DEVELOPMENT

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.n o** | **Activity Name** | **Description** | **Mapping Activity** | **Estimate d Cost** | **Person Responsibl**  **e** | **Ris k** | **Success Criteria** |
| **4.** | **Developmen** | Development of the application | Prototype, | - | Developmen |  | All |
|  | **t** | by keeping in mind with system  design & architecture | System  Design and |  | t Manager | functionalitie  s should be |
|  |  |  | Functional |  |  | implemented |
|  |  |  | Document |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| **4.1.** | **Framework Design** | A framework used by development teams to create high-quality software in an  efficient and cost-effective manner. | End to End Flow Diagram in SRS,  Identify Reusable Functions and  Applicatio  n pages | - | Developmen t Manager |  | Easy to Use and Less Complicated while debugging |
| **4.2.** | **User Interface Design** | System architecture describes.  its main components, connection s (structures) and how they  interact with each other. | End to End Flow Diagram in SRS | - | Developmen t Team |  | Development phase will be smooth and Final Version of End  Product will be good |
| **4.3.** | **Database** | It includes the design of application, network, database,  user interface and computer interfaces | End to End Flow Diagram in SRS | - | Developmen t Team |  | Development phase will be smooth and Final Version of End  Product will be good |
| **4.4.** | **Backend** | Final approval from the stakeholders about design of product before development phase starts | End to End Flow Diagram in SRS,  Identify Reusable Functions and  Applicatio  n pages | - | Developmen t Team |  | Development phase will be smooth and Final Version of End  Product will be good |
| **4.6.** | **Model Developmen t** | Developing the model which is accurate for our business case | End to End Flow Diagram in SRS,  Identify Reusable Functions and  Applicatio  n pages | - | Developmen t Team |  | Fault  Tolerance and Response time will be  good |
| **4.7.** | **Integration** | Integrating the frontend & backend code to make the application work | End to End Flow Diagram in SRS,  Identify Reusable Functions and  Applicatio  n pages | - | Developmen t Team |  | Fault  Tolerance and Response time will be  good |
| **4.8.** | **Code Review** | Review the code with the team lead to make sure following coding principles | Code needs to be  completed | - | Developmen t Manager |  | Less Defects and cover all the  functionality |
| **4.9.** | **Unit Testing** | Write classes for all components for testing the functionality | Code needs to | - | Developmen t Team |  | Cover all the functionality,  requirement, |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | be reviewed |  |  |  | testing end to end from development  side |

WBS: Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Activity Name** | **Description** | **Mapping Activity** | **Estimated Cost** | **Person Responsible** | **Risk** | **Success Criteria** |
| **5.** | **Review and** | Cover all the | Code should | - | Testing |  | Cover all the |
|  | **Design Test** | Positive and | be Ready |  | Team, | Requirement |
|  | **cases** | Negative |  |  | Testing | and Maximum |
|  |  | Scenarios of all |  |  | Manager | Coverage so |
|  |  | the Functional |  |  |  | that testing |
|  |  | and End to End |  |  |  | team should |
|  |  | Requirements |  |  |  | not miss any |
|  |  |  |  |  |  | requirement |
|  |  |  |  |  |  | while doing |
|  |  |  |  |  |  | testing |
| **5.1.** | **Integration** | In this phase, the | All the | - | Testing Team |  | Cover all the |
|  | **Testing** | different units or  modules of code are tested together to ensure they  function correctly | Functional components should be  tested |  |  | Requirement |
|  |  | as a whole. This |  |  |  |  |
|  |  | phase helps detect |  |  |  |  |
|  |  | any issues or |  |  |  |  |
|  |  | conflicts between |  |  |  |  |
|  |  | the various |  |  |  |  |
|  |  | modules of the |  |  |  |  |
|  |  | software. |  |  |  |  |
| **5.2.** | **System Testing** | This phase involves | Functional | - | Testing Team |  | Cover all the |
|  |  | testing the entire  system to ensure it meets the  requirements and specifications | components should be  tested  independently |  |  | Requirement |
|  |  | outlined in the |  |  |  |  |
|  |  | project plan. |  |  |  |  |
|  |  | System testing |  |  |  |  |
|  |  | typically involves |  |  |  |  |
|  |  | using both manual |  |  |  |  |
|  |  | and automated |  |  |  |  |
|  |  | testing tools and |  |  |  |  |
|  |  | techniques to |  |  |  |  |
|  |  | identify any |  |  |  |  |
|  |  | potential issues or |  |  |  |  |
|  |  | bugs. |  |  |  |  |
| **5.3.** | **Acceptance** | In this phase, the | All the testcases should be executed  based on the acceptance  criteria | - | Testing Team |  | Cover all the |
|  | **Testing** | software is tested  by the client or |  |  | Requirement |
|  |  | end-user to ensure |  |  |  |
|  |  | that it meets their |  |  |  |
|  |  | expectations and |  |  |  |
|  |  | requirements. This |  |  |  |
|  |  | phase involves |  |  |  |
|  |  | testing the |  |  |  |
|  |  | software in a real- |  |  |  |
|  |  | world |  |  |  |

.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | environment, and it helps ensure that  the software is ready for  deployment. | |  |  |  |  |  |
| **5.4.** | **Regression Testing** | This phase involves retesting the system or  application after any changes or modifications have been made to the code to ensure that there are no negative impacts  on existing | | All Functional Testing should be performed | - | Testing Team |  | Existing functionalities should stable |
| functionality |  |
| **5.6.** | **Performance Testing** | This phase involves testing the software's  performance under different workloads and conditions to ensure that it can handle the  expected load and  traffic. | | All Functional Testing should be performed | - | Testing Team |  | Cover all the Requirement |
| **5.7.** | **Security Testing** | This phase involves testing the  software's security features to ensure that it is secure from unauthorized access and potential cyber-  attacks. | | All Functional Testing should be performed | - | Testing Team |  | Cover all the Requirement |
| **5.8.** | **Require Traceability Matrix** | Mapp all the testcases with the  requirements. | | Testing should be done | - | Testing Team |  | Maximum Coverage so that testing team should not miss any requirement  while testing |
| **5.9.** | **Final**  **Approval** | Final approval on all the testing Environment and ready for  deployment | | Testing should be done with minimum defect leakage or no critical and major  defects |  | Testing Manager, Project Manager |  | All the functionalities are tested and ready for Live |

WBS: Project DEPLOY

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.n o** | **Activity Name** | **Description** | **Mapping Activity** | **Estimate d Cost** | **Person Responsibl e** | **Ris k** | **Success Criteria** |
| **6.** | **Deploy** | It covers the work required to deploy the final | Testing should be  completed | - | Dev-ops Team |  | Application deploys  without any |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | solution to target production environments. |  |  |  |  | hindrance and user use the application without having any  issues |
| **6.1.** | **CI/CD/CD**  **Pipeline** | A series of steps to follow to deliver a new version of the software. CI/CD/CD introduces monitoring and automation to  improve processes | Production Environment should be  ready | - | Dev-ops Team |  | Successfully running the build |
| **6.2.** | **Prepare System Manual** | It is a system-specific hybrid document that includes  an  operating manual, a maintena nce manual, and  additional information. | Production Environment should be  ready and clear  understandin g of all the Requirement  s | - | Testing Team |  | Easy to use for users |
| **6.3.** | **Prepare User Manual** | In this phase, the software is tested by the client or end-user to ensure that it meets their  expectations and requirements. This phase involves testing the software in a real-world  environment, and it helps ensure that the software is ready for deployment. | Production Environment should be  ready and clear  understandin g of all the Requirement s | - | Testing Team |  | Easy to use for users |
| **6.4.** | **Prepare Technical Manual** | This phase involves retesting the system or application after any changes or modifications have  been made to the code to ensure that there are no negative  impacts on existing functionality. | Production Environment should be  ready and clear  understandin g of all the Requirement s | - | Testing Team |  | Easy to use for users |
| **6.5.** | **Staging** | In this phase, the project is deployed to a staging  environment that closely mimics the production environment. This environment is used to test the deployment process and ensure  that everything is working as expected. | Production Environment should be  ready and clear  understandin g of all the Requirement  s | - | Testing Team |  | Cover all the  Requireme nt |
| **6.6.** | **Pre- deployment Testing** | Once the staging environment has been set up, the project is tested to ensure that it is fully functional and meets the requirements and expectations of the client or end- user. | Production Environment should be  ready and clear  understandin g of all the Requirement  s | - | Testing Team |  | Cover all the  Requireme nt |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **6.7.** | **Deploymen t** | In this phase, the project is deployed to the production environment. This typically involves using automated  deployment tools or scripts to ensure that the deployment  process is efficient and error-free. | Testing should be done | - | Testing Team |  | Successfully Deploy the build in all platforms |
| **6.8.** | **Post- deployment Testing** | Once the project has been deployed to the production  environment, it is tested again to ensure that everything is working correctly. This testing is typically done in a production-like  environment to ensure that the project is ready for use by end-  users. | Testing all the functionalities |  | Testing Manager, Project Manager |  | Successfully Deploy the build in all platforms |
| **6.9.** | **Monitoring and Maintenanc e** | Once the project has been deployed, it is important to monitor its performance and ensure that it is running  smoothly. Any issues or problems that arise should be addressed  immediately to minimize downtime and ensure that the  project is available to end-users. | Application should be on Production |  | Support Team |  | Application running smoothly |

# Individual Technical Approach

1. The purpose is to provide a detailed overview of the approach that will be taken to create a recommendation system for Canadian restaurants and tourist attractions.
2. The system aims to provide personalized recommendations to users based on their preferences, past behaviour, and other relevant factors.
3. The system will utilize machine learning algorithms and data analysis techniques to make accurate predictions and improve over time based on user feedback.
4. This document outlines the technical approach for developing, testing, and deploying the recommendation system, including the tools and technologies to be used, the development process, and the testing methodologies.

## System Architecture

* + 1. Data Ingestion and Pre-processing

This component will be responsible for collecting data from various sources such as restaurant and tourist attraction websites, social media platforms, and user feedback. The data will be pre-processed to remove any duplicates, missing values, or inconsistent data.

* + 1. Data Storage and Management

The pre-processed data will be stored in a cloud-based database such as Amazon Web Services (AWS) or Microsoft Azure. The database will be designed for high scalability and performance, with data partitions and replication to ensure high availability.

* + 1. Machine Learning Models

The recommendation system will use various machine learning models such as collaborative filtering, content-based filtering, and hybrid models to generate recommendations. These models will be trained on the pre-processed data and will utilize algorithms such as K-Nearest Neighbors (KNN), Singular Value

Decomposition (SVD), and Neural Networks.

* + 1. Recommendation Engine

The recommendation engine will be responsible for generating personalized recommendations for each user based on their preferences and past behavior. The engine will consider factors such as location, cuisine, price, ratings, and popularity.

* + 1. User Interface

The user interface will be designed to be user-friendly and intuitive, with features such as search, filters, and recommendations. The interface will be developed using modern web technologies such as ReactJS and NodeJS.

* + 1. Deployment and Monitoring

The system will be deployed on a cloud-based platform such as AWS or Microsoft Azure. The system will be monitored using tools such as CloudWatch and Azure Monitor, and any issues or errors will be addressed promptly.

## Data Collection

We will describe the process of collecting data from various sources, including tourist attraction websites, restaurant booking websites, social media platforms, and other relevant sources. It will also explain how we plan to clean and pre-process the data to make it suitable for recommendation generation.

* + 1. Data collected from various sources using the following:
       1. Web scraping: You can use web scraping tools, such as Beautiful Soup or Scrapy, to extract data from websites that list Canadian restaurants or tourist attractions, such as TripAdvisor, Yelp, or Google Maps.
       2. APIs: You can use APIs provided by websites that list Canadian restaurants or tourist attractions, such as TripAdvisor or Yelp, to access their data directly. You will need to sign up for an API key and follow their API usage guidelines.
       3. User-generated data: You can collect user-generated data by allowing users to rate and review Canadian restaurants or tourist attractions through your website or mobile app.
       4. Public datasets: You may also find public datasets that contain information on Canadian restaurants or tourist attractions, such as open data portals or data-sharing platforms.
       5. Manually collected data: You can also manually collect data by researching Canadian restaurants or tourist attractions and recording their information, such as their name, location, and rating.
    2. Entities for creating the chatbot.
       1. Name of the restaurant or attraction
       2. Location (address, city, province)
       3. Type of cuisine or attraction (e.g., Italian restaurant, outdoor attraction)
       4. User ratings and reviews
       5. Price range
       6. Opening and closing times
       7. Amenities offered (e.g., parking, wheelchair accessibility)
       8. Images or photos
       9. Historical visitor data (e.g., number of visitors, peak season)
       10. Popularity or trending information
    3. Various types of datasets
       1. Yelp Open Dataset: A large dataset containing information on businesses, including Canadian restaurants and tourist attractions, as well as user reviews and ratings. The data can be accessed for free through the Yelp website.
       2. TripAdvisor Dataset: A dataset containing information on tourist attractions and restaurants around the world, including Canada. The dataset includes information on location, ratings, and reviews.
       3. Canadian Tourism Commission Data: The Canadian Tourism Commission provides data on Canadian tourist attractions and destinations, including information on visitor numbers and popular attractions.
       4. Statistics Canada: Statistics Canada provides a range of data on Canadian businesses and tourism, including data on restaurants and tourist attractions.
       5. OpenData.gc.ca: The Government of Canada's open data portal contains a range of datasets related to tourism and travel in Canada, including data on restaurants and tourist attractions.
       6. It's important to note that while these datasets can provide valuable information for your recommendation system, you may need to combine or supplement the data with other sources to ensure that you have a comprehensive and up-to-date dataset.
    4. Finalized Datasets
       1. Candian\_resturant.xlsx
          1. Has the following entities:

Restaurant Name

Cuisine Type

Address

City

Province/State

Country

Postal/Zip Code

Phone Number

Price Range (per person)

Average Rating (out of 5)

Number of Reviews

* + - 1. Candian\_places.xlsx

1. Longitude
2. Latitude
3. Name
4. Place\_type
5. Phone
6. dates\_open
7. amenities
8. state
9. State Name
10. City
11. Province/State
    * + 1. Candian\_Provinces.xlsx
12. Longitude
13. Latitude
14. state
15. City
16. Province/State

## Data Pre-Processing

This section will describe the techniques that we plan to use for pre-processing the data, including data normalization, outlier removal, and feature selection. It will also explain the tools and technologies that we plan to use for data pre-processing.

* + 1. Data cleaning: This involves removing or correcting any inaccurate, incomplete, or irrelevant data. For example, removing duplicates or filling in missing values.
    2. Data normalization: This technique involves scaling the data to a common range to ensure that each attribute is given equal weight. For example, scaling the ratings of different restaurants on a scale of 1 to 10.
    3. Data transformation: This technique involves converting the data into a suitable format for analysis. For example, converting categorical data such as cuisine type or location into numerical data using one-hot encoding.
    4. Data reduction: This technique involves reducing the dimensionality of the data by selecting only the most relevant features or attributes. For example, selecting only the most frequently visited tourist attractions.
    5. Outlier detection: This technique involves identifying and removing any data points that are significantly different from the others. For example, removing a restaurant with an unusually high rating compared to other restaurants in the same category.

## Feature Extraction

This section will describe the features that we plan to extract from the pre-processed data. It will explain the techniques that we plan to use for feature extraction, including text mining, sentiment analysis, and image processing.

* + 1. Bag of Words (BoW): This technique involves representing the text data as a collection of words without considering the order in which they appear. It can be used to extract features from customer reviews or descriptions of restaurants and tourist attractions.
    2. Term Frequency-Inverse Document Frequency (TF-IDF): This technique is used to weigh the importance of words in a document based on their frequency and how frequently they appear in other documents. It can be used to identify the most important features in a corpus of text data, such as customer reviews or descriptions of restaurants and tourist attractions.
    3. Collaborative Filtering: This technique is based on the idea that users who have similar preferences in the past are likely to have similar preferences in the future. It can be used to identify patterns in customer behavior and preferences based on their past interactions with the system, such as their ratings or reviews of restaurants and tourist attractions.

## Recommendation Generation

This section will describe the techniques that we plan to use for recommendation generation. It will explain the algorithms that we plan to use for recommendation generation, including collaborative filtering, content-based filtering, and hybrid approaches.

* + 1. Collaborative Filtering: This technique involves analyzing user behavior and preferences to identify patterns and similarities between users. It can be used to recommend restaurants and attractions that similar users have enjoyed. For example, if User A and User B both enjoyed a particular restaurant, the system might recommend that restaurant to User C who has similar preferences.
    2. Content-Based Filtering: This technique involves analyzing the attributes and characteristics of restaurants and attractions to make recommendations based on user preferences. For example, if a user has indicated that they prefer vegetarian restaurants, the system might recommend vegetarian restaurants in the area.
    3. Hybrid Techniques: These techniques combine collaborative and content-based filtering to improve the accuracy of recommendations. For example, the system might use collaborative filtering to identify similar users, and then use content-based filtering to recommend restaurants and attractions that are likely to appeal to those users.
    4. Matrix Factorization: This technique involves breaking down large data sets into smaller, more manageable components to make recommendations. For example, the system might break down

the data into user-item matrices, and then use matrix factorization to identify patterns and make recommendations based on those patterns.

* + 1. Deep Learning: This technique involves using neural networks to make recommendations based on large amounts of data. For example, the system might use a deep learning algorithm to analyse user behavior and preferences and make personalized recommendations based on that analysis.

## User Interface

This section will describe the user interface that we plan to develop for the recommendation system. It will explain the design principles that we plan to follow and the tools and technologies that we plan to use for developing the user interface.

* + 1. Responsive design: This technique allows the user interface to adapt to different screen sizes and devices, making it easy to use on desktops, tablets, and smartphones.
    2. Interactive design: Interactive elements such as sliders, buttons, and animations can make the user interface more engaging and intuitive.
    3. Personalization: Personalization techniques can be used to customize the user interface for each user based on their preferences, location, and other factors.
    4. Visual design: A visually appealing and well-designed interface can help attract and retain users and make the system more user-friendly. This includes the use of colors, typography, and imagery.
    5. Navigation: The user interface should have clear and intuitive navigation to make it easy for users to find what they're looking for. This can include menus, search bars, and filters.
    6. Feedback: The user interface should provide feedback to the user on their actions and provide clear error messages when needed.

## Evaluation metrics

This section will describe the evaluation metrics that we plan to use for evaluating the performance of the recommendation system. It will explain the different metrics that we plan to use, including precision, recall, F1 score, and mean average precision.

* + 1. Precision and Recall: Precision is the proportion of relevant items recommended, while recall is the proportion of relevant items that were recommended out of the total number of relevant items. For example, if the recommendation system recommends 10 items and 7 of them are relevant, the

precision is 0.7 and the recall is 0.58.

* + 1. Mean Absolute Error (MAE): MAE is the average difference between the predicted rating and the actual rating given by the user. For example, if the predicted rating for a restaurant is 4.5 and the actual rating given by the user is 4, the MAE is 0.5.
    2. Root Mean Squared Error (RMSE): RMSE is the square root of the average of the squared differences between the predicted rating and the actual rating given by the user. For example, if the predicted rating for a restaurant is 4.5 and the actual rating given by the user is 4, the squared difference is 0.25, and the RMSE is 0.5.
    3. Mean Average Precision (MAP): MAP is the average precision across all users. For example, if the precision for User A is 0.8 and the precision for User B is 0.6, the MAP is (0.8+0.6)/2 = 0.7.
    4. Normalized Discounted Cumulative Gain (NDCG): NDCG measures the quality of the recommendation list by assigning higher scores to relevant items that are ranked higher in the list. For example, if a user is interested in Italian restaurants and the recommendation system recommends 5 Italian restaurants in the following order: A, B, C, D, E, the NDCG score would be higher if A and B were ranked higher than C, D, and E.
    5. Mean Reciprocal Rank (MRR): MRR is the average of the reciprocal ranks of the first relevant item in the recommendation list. For example, if the first relevant item in the recommendation list for a user is ranked 3rd, the reciprocal rank is 1/3, and the MRR is the average of all reciprocal ranks for all users.

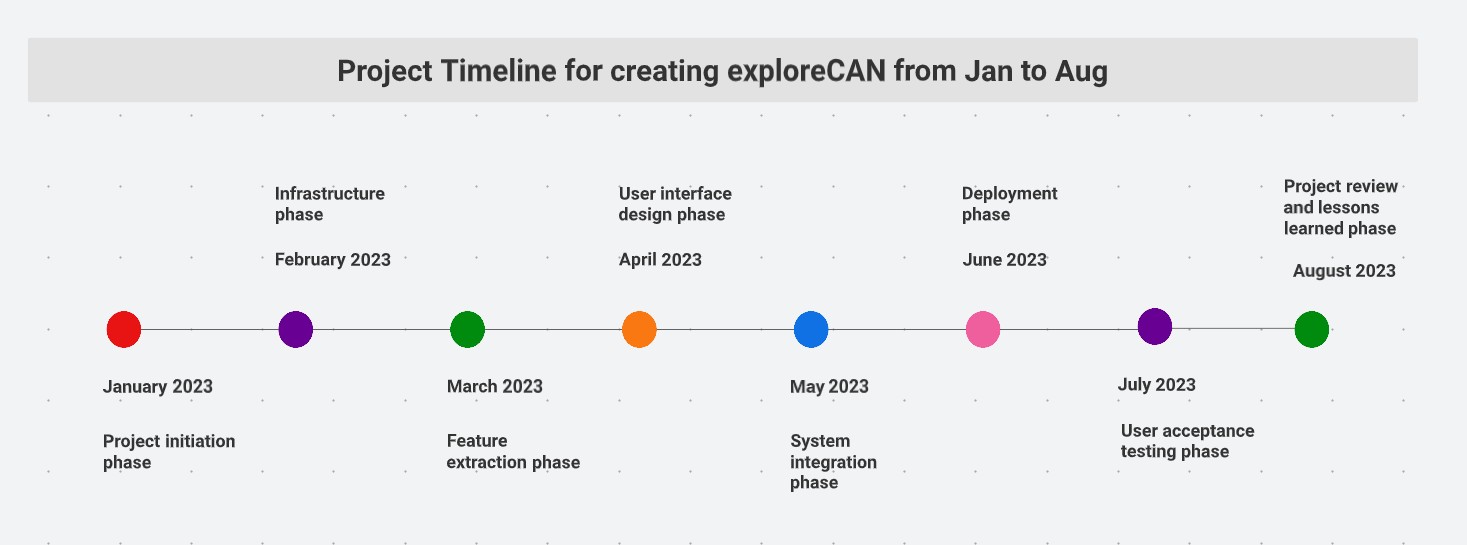
## Conclusion

The conclusion section will summarize the technical approach that we plan to follow for developing the recommendation system for Canadian restaurants and tourist attractions. It will also mention the challenges that we anticipate and the future work that we plan to undertake to improve the system.

## Project Timeline

A screenshot of a computer screen

Description automatically generated



Jan 2023

* + 1. Project initiation phase: Develop project charter, define scope and objectives, identify stakeholders.
    2. Requirements gathering phase: conduct user interviews, identify data sources, define user requirements.

Feb 2023

1. Infrastructure phase: select and set up hardware and software environment, implement security measures.
2. Data pre-processing phase: collect and clean data, perform exploratory data analysis.

Mar 2023

1. Feature extraction phase: extract relevant features from the pre-processed data.
2. Recommendation generation phase: develop recommendation algorithms and test their performance.

April 2023:

1. User interface design phase: develop wireframes and prototypes, design user interface.
2. Evaluation metrics selection phase: select appropriate evaluation metrics, design testing procedures.

May 2023:

1. System integration phase: integrate recommendation algorithms and user interface into a single system.
2. Testing and quality assurance phase: conduct thorough testing, ensure system meets user requirements and quality standards.

June 2023:

1. Deployment phase: deploy the system in a production environment, monitor its performance and user feedback.
2. Training and documentation phase: develop user training materials and system documentation. July 2023:
3. User acceptance testing phase: conduct user acceptance testing, gather user feedback, and make necessary improvements.
4. Finalize project deliverables and close out phase: finalize all project documentation and deliverables, obtain sign-off from stakeholders.

August 2023:

i. Project review and lessons learned phase: conduct a project review, identify areas for improvement, and document lessons learned.

# Technologies and Tools

1. Programming Languages: Python

Usage: These programming languages are commonly used for developing machine learning models and implementing data processing pipelines.

1. Machine Learning Libraries: TensorFlow, PyTorch, Scikit-learn.

Usage: These libraries can be used for developing and training machine learning models for recommendation generation.

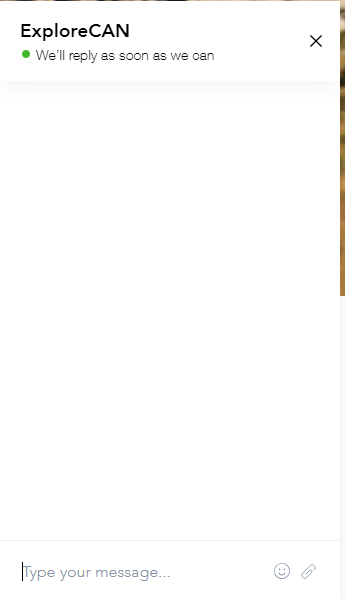
1. Big Data Processing Frameworks: Apache Spark, Hadoop

Usage: These frameworks can be used for processing and analysing large datasets of user and item interactions.

1. Database Systems: MySQL

Usage: These systems can be used for storing and managing the recommendation system's data, such as user profiles, item metadata, and interaction logs.

1. Web Development Frameworks: Flask, Django, React

Usage: These frameworks can be used for developing the recommendation system's user interface and deploying it as a web application.

1. Version Control Systems: Git, SVN

Usage: These systems can be used for managing the recommendation system's source code and collaborating with team members.

1. Cloud Platforms: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP)

Usage: These platforms can be used for deploying the recommendation system in a scalable and cost- effective manner, as well as for managing and monitoring its infrastructure.

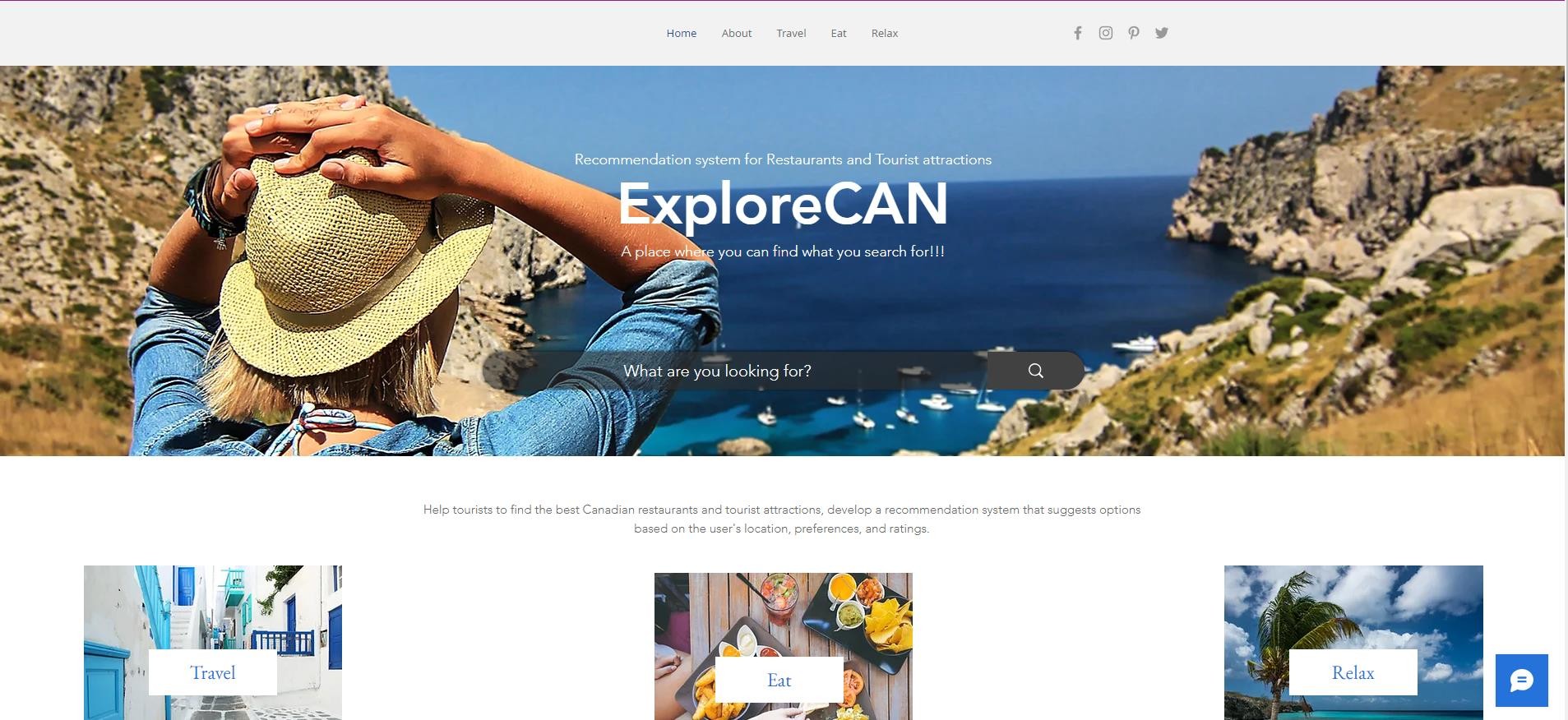
A close-up of words

Description automatically generated

# Current State of Project:

1. User interface:

Front-end landing page designed using HTML/CSS/Bootstrap



User interaction chat box

1. Database has been designed using Ms-SQL



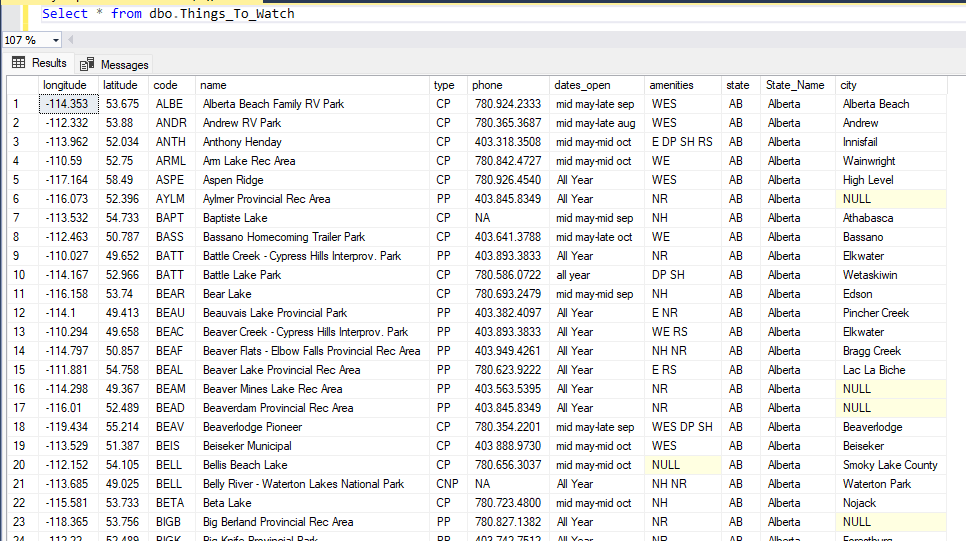
1. Tableau Dashboard Analysis

A screenshot of a graph

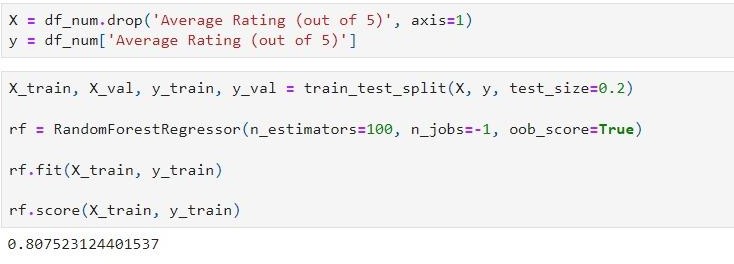
Description automatically generated

A chart of different colored rectangles

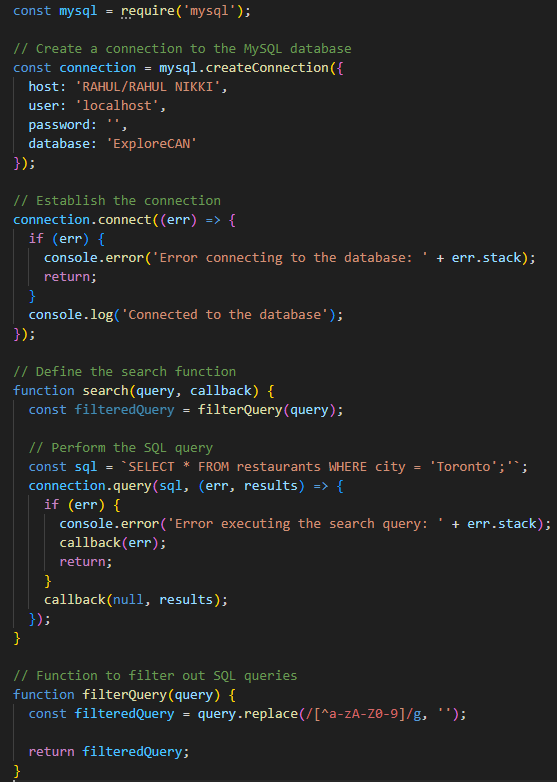
Description automatically generated



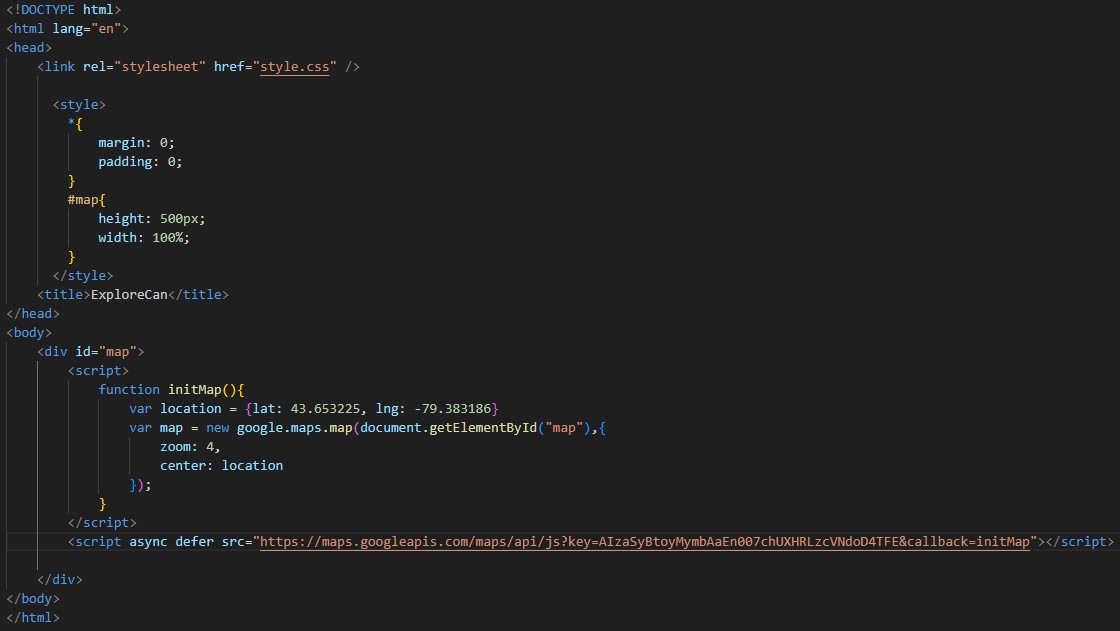
1. Accuracy has been tested for the restaurants data.

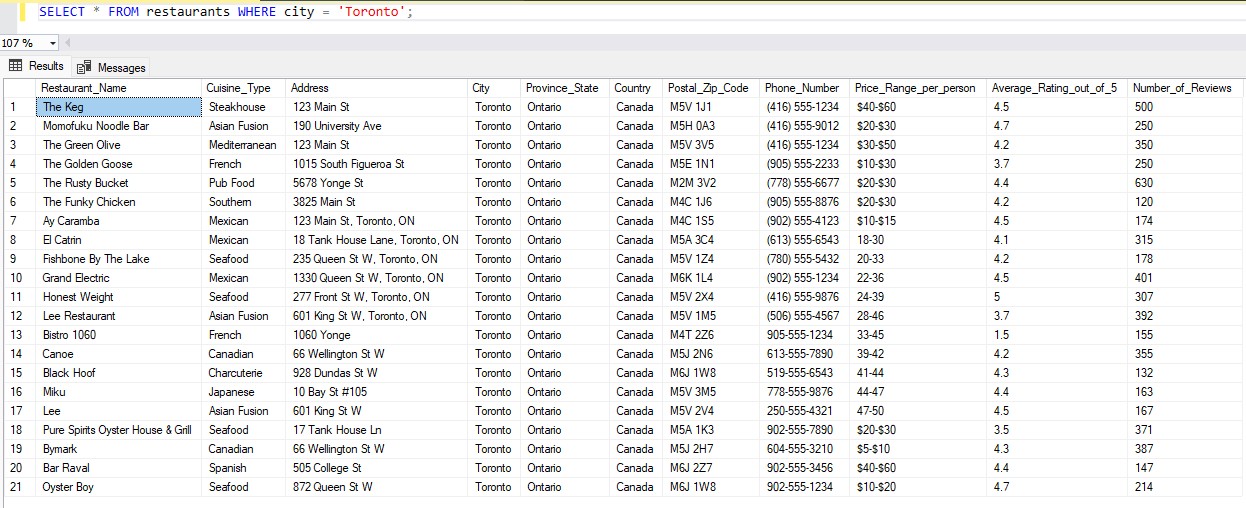
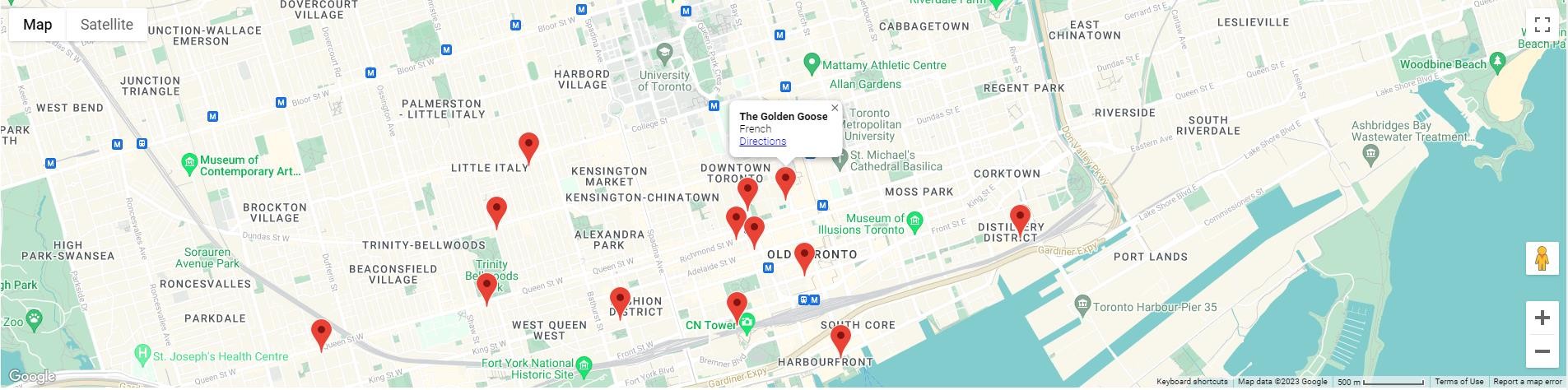


1. Backend Code for UI and Database connection to filter restaurants.



1. Google Maps API code





All the restaurants of Canada are displayed on this map.

# Sample Questionnaires for chatbot

* + - User: "Can you suggest a good restaurant in Canada?"
    - Chatbot: "Certainly! What type of cuisine are you looking for?"
    - User: "I need a recommendation for a restaurant in Canada."
    - Chatbot: "Of course! What kind of food do you prefer?"
    - User: "Can you help me find a nice place to eat in Canada?"
    - Chatbot: "Absolutely! Are you in the mood for a specific cuisine?"
    - User: "I'm visiting Canada. Any suggestions for a great restaurant?"
    - Chatbot: "Certainly! Any preferences on the type of cuisine you'd like?"
    - User: "I want to explore some tourist attractions in Canada."
    - Chatbot: "Great! What kind of attractions are you interested in?"
    - User: "I'm looking for a unique dining experience in Canada."
    - Chatbot: "Sure! Are you open to any specific cuisine or ambiance?"
    - User: "Recommend a restaurant in Canada with a beautiful view."
    - Chatbot: "Certainly! Any cuisine you'd like to enjoy while taking in the view?"
    - User: "Can you suggest a popular restaurant in Canada?"
    - Chatbot: "Absolutely! Any specific cuisine or location preferences?"
    - User: "I need a recommendation for a family-friendly restaurant in Canada."
    - Chatbot: "Of course! Any cuisine your family enjoys?"
    - User: "What are some must-visit attractions in Canada?"
    - Chatbot: "Great! Are you interested in historical, cultural, or natural attractions?"
    - User: "I'm searching for an off-the-beaten-path restaurant in Canada."
    - Chatbot: "Certainly! Any specific cuisine or location preferences for the restaurant?"
    - User: "Recommend a restaurant with live music in Canada."
    - Chatbot: "Absolutely! What type of music and cuisine are you interested in?"
    - User: "I want to try something unique and local in Canada. Any suggestions?"
    - Chatbot: "Sure! Are you open to trying different types of cuisines?"
    - User: "Any hidden gem restaurants in Canada?"
    - Chatbot: "Certainly! Do you have any specific cuisine preferences or dietary restrictions?"
    - User: "What are some iconic tourist attractions in Canada?"
    - Chatbot: "Great! Are you interested in specific provinces or types of attractions?"
    - User: "I'm in Canada and craving seafood. Any recommendations?"
    - Chatbot: "Absolutely! Are you nearing a specific city or looking for a particular type of seafood?"
    - User: "I want to explore outdoor activities in Canada. Any suggestions?"
    - Chatbot: "Sure! Are you interested in hiking, skiing, wildlife, or something else?"
    - User: "I'm looking for a romantic restaurant in Canada."
    - Chatbot: "Certainly! Any cuisine or ambiance preferences for the romantic dinner?"
    - User: "What are some famous landmarks to visit in Canada?"
    - Chatbot: "Great! Are you looking for landmarks in specific cities or provinces?"
    - User: "Recommend a restaurant in Canada that offers vegetarian options."
    - Chatbot: "Absolutely! Are you looking for a specific cuisine or location?"
      * Conversation 1:

1. User: "Hi, I'm looking for a nice restaurant in Canada."
2. Chatbot: "Of course! What type of cuisine are you in the mood for?"
3. User: "I would love some Mexican food."
4. Chatbot: "Great! Based on your location, I recommend trying 'Cantina Mexicana'. It's a popular Mexican restaurant known for its authentic flavors and vibrant atmosphere."
   * + - Conversation 2:
5. User: "Hey, I need some suggestions for tourist attractions in Canada."
6. Chatbot: "Absolutely! What kind of attractions are you interested in? Historical, cultural, natural?"
7. User: "I'm more into historical sites."
8. Chatbot: "Got it! I suggest visiting 'The Citadel' in Halifax, Nova Scotia. It's a historic fort that offers stunning views of the city and provides insights into Canada's military history."
   * + - Conversation 3:
9. User: "Hello! Can you recommend a restaurant or attraction in Canada?"
10. Chatbot: "Of course! Are you looking for a restaurant or a tourist attraction?"
11. User: "I'm interested in exploring tourist attractions."
12. Chatbot: "That's great! What kind of experience are you seeking? Something adventurous, cultural, or natural?"
13. User: "I'd love a natural experience."
14. Chatbot: "In that case, I recommend visiting 'Banff National Park' in Alberta. It's known for its breathtaking landscapes, stunning mountains, and beautiful hiking trails."

# LUIS for creating recommendation systems.

* Define Intents:
  + We have Identified the user intents related to the recommendation system. For example, "FindRestaurants", "FindAttractions", "GetRecommendations", etc. These intents represent what users want to do with the chatbot.
* Created Utterances:
  + Created a list of example utterances for each intent. These are sample user inputs that the chatbot should be able to recognize and classify correctly. For instance:
  + "Find restaurants in Toronto."
  + "Suggest me some Italian restaurants."
  + "What are the top attractions in Vancouver?"
* Training and testing the utterances and intents:
  + We trained the LUIS model using the defined intents, entities, and utterances. LUIS will learn to recognize patterns and map user input to the appropriate intents and entities.
  + Test the model to ensure that it accurately understands user input and classifies it correctly.

# Designing the conversation workflow

* Entity Extraction:
  + Extract important entities from the user input using LUIS. Entities like "Location," "Cuisine," or "Activity Type" are essential for providing accurate recommendations.
* Get User Preferences:
  + If the user query includes entities, ask follow-up questions to gather additional preferences. For example, "What type of cuisine are you interested in?" or "Do you have a specific location in mind?"
* Location-Based Recommendations:
  + If the user provides a location or if the chatbot knows the user's location (based on integration with geolocation services), provide recommendations specific to that location.
* Recommendation Generation:
  + Use the recommendation algorithms and external APIs to generate a list of recommended restaurants or tourist attractions based on the user's preferences and location.
* Present Recommendations:
  + Display the recommendations to the user in a user-friendly format. You can show details like the name, description, ratings, and contact information for each recommendation.
* Handling User Feedback:
  + Provide options for the user to give feedback on the recommendations, such as liking or disliking specific suggestions.
* Additional Actions
  + Allow users to perform other actions like refining their preferences, asking for more recommendations, or saving their favorite places for future reference.

# Challenges

1. Data availability and quality:
   1. The quality and quantity of data available can greatly affect the accuracy and effectiveness of the recommendation system.
   2. There may also be challenges in acquiring relevant data, especially for new or small businesses that may not have a significant online presence.
2. Privacy concerns:
   1. Collecting and using personal data to make recommendations can raise concerns around privacy and data security.
   2. It will be important to ensure that the system is designed in compliance with relevant laws and regulations and that users are fully informed about the use of their data.
3. User diversity:
   1. Tourists and locals may have different preferences and requirements when it comes to

restaurants and attractions, making it difficult to create a one-size-fits-all recommendation system.

* 1. Additionally, cultural differences and language barriers may also pose a challenge.

1. Scalability:
   1. As the number of businesses and users increases, it can become challenging to maintain the speed and efficiency of the recommendation system.
   2. This may require the adoption of more robust technologies and infrastructure.
2. Integration with existing systems: Integrating the recommendation system with existing booking or reservation systems used by businesses may be a challenge, especially if the systems are not compatible.
3. Conclusion

In conclusion, the creation of a recommendation system for Canadian restaurants and tourist attractions has been progressing well. The project charter was created, and the project team is following a well- defined timeline. The data cleaning, pre-processing, and feature extraction phases have been completed, and we are currently in the recommendation generation and user interface design stages.

We have faced some technical and logistical challenges, such as the collection of reliable and diverse data, ensuring the scalability of the system, and dealing with potential privacy concerns. However, we have

implemented appropriate techniques and strategies to address these challenges, such as adopting cloud- based technologies, using advanced machine learning algorithms, and following ethical guidelines.

Our team is continually striving to improve the system's accuracy, performance, and usability. We will conduct thorough testing and evaluation to ensure that the system meets the project goals and objectives. Additionally, we will maintain effective communication with stakeholders and users to gather feedback and incorporate their suggestions into the system.

Overall, we are optimistic about the project's progress and believe that the final system will provide valuable recommendations for Canadian restaurants and tourist attractions, benefiting both locals and tourists alike.

# References

1. Comprehensive Guide to build a Recommendation Engine from scratch (in Python) ***Available Online:*** [*https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/*](https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/) *Accessed on: 22nd February 2023*
2. Places API ***Available Online:*** [***https://developers.google.com/maps/documentation/places/web-service***](https://developers.google.com/maps/documentation/places/web-service)

*Accessed on: 12th Jan 2023*

Create A Travel/Tourism Website Using HTML and CSS ***Available Online:***

[***https://www.codewithrandom.com/2022/12/03/travel-website-using-html-css/***](https://www.codewithrandom.com/2022/12/03/travel-website-using-html-css/) *Accessed on: 24th Mar 2023*