ISY5001: PRACTICE MODULE for Certificate in Intelligent Reasoning Systems (IRS)

January 2025 Semester 2

BizBeacon SG

Intelligent Singapore Business Location Advisor System

Final Report

Group 3

Group **3**	Name	Student ID
Member #1	CHUA HIENG WEIH	A0315386Y
Member #2	GU HAIXIANG	A0131920U
Member #3	LIZABETH ANNABEL TUKIMAN	A0315378X
Member #4	SRITAM PATNAIK	A0115530W
Member #5	MUHAMMAD HARUN BIN ABDUL RASHID	A0164598L

Table of Contents

1.0 Executive Summary

Choosing the right location and type for a business is a critical decision that impacts its success, influencing factors such as customer accessibility, brand visibility, and operational efficiency. However, many entrepreneurs and small businesses in Singapore face challenges in making informed decisions due to fragmented, time-consuming, and costly research methods. To address this, our team of 5 aim to develop an AI-driven business advisor tailored to the Singapore market that streamlines the decision-making process. Our tool offers a unified platform that not only suggests profitable business locations but also recommends viable business types and answers location-specific queries.

Our solution revolves around an intelligent chatbot integrated with retrieval-augmented generation (RAG), built upon a knowledge graph. The process begins with data collection and preparation, followed by the creation of the knowledge graph on Neo4j, which encodes the relationships between businesses, business types, Singapore's planning areas or subzones, and their respective attributes. To extract valuable insights from this data, we integrate a scoring engine that involves geospatial density analysis, fuzzy logic, and graph neural networks (GNNs). These techniques help compute scores that are eventually used to rank potential business locations based on factors like population density and competitor presence.

When a user interacts with our chatbot, it extracts their intent and relevant entities from their input, and selects the most appropriate Cypher query or queries from a predefined list to retrieve the relevant data from the knowledge graph. This retrieved data is then incorporated into the user's query to provide richer context, before being processed by a large language model (LLM), specifically OpenAI's GPT-4o, to generate user-friendly answers and recommendations.

Our tool is specifically designed to help first-time business owners in Singapore make informed, data-driven decisions on where to establish their businesses or what type of business to start. By integrating geospatial data, competitor analysis, and demographic insights, our system enables entrepreneurs to confidently navigate the complexities of starting a business.

2.0 Introduction

Starting a business in Singapore involves navigating complex decisions around location selection, competitor analysis, demographic suitability, and property availability. To support first-time entrepreneurs and small businesses across a range of industries, we built an intelligent Singapore business advisor that integrates reasoning with a LLM, knowledge retrieval from a knowledge graph, and data-driven optimisation.

Our system helps users in multiple ways, such as recommending optimal locations given a business type, suggesting suitable business types to open based on a location, as well as providing structured information about business types and locations, including competitor presence, demographic trends, and available properties.

Unlike traditional dashboard-based platforms or general advisory services, our solution offers an intuitive conversational AI interface, allowing users to naturally interact and access data-driven recommendations without needing technical expertise. By making location and business intelligence accessible and cost-effective, our intelligent chatbot aims to automate and simplify key early-stage business decisions, empowering users to make more informed strategic decisions.

2.1 Problem Background

Choosing the right location and business type is crucial for business success, influencing factors such as customer accessibility, brand visibility, and operational efficiency, all of which in turn impact business sustainability and growth (Singapore Company Formation, n.d.). For instance, poor location decisions can lead to significant financial losses and operational challenges, as demonstrated by Marks & Spencer's closure of its store at Parkway Parade, which was driven by rising rental costs and concerns about foot traffic (Lim & Sheo, 2025). This example highlights the importance of making informed, data-driven decisions when selecting a business location.

However, traditional methods of gathering relevant information, such as relying on real estate listings, SME consultations, or independent, manual research, are often time-consuming, fragmented, and costly. Existing tools either focus narrowly on property listings or provide generalised business advice without integrating competitor analysis, demographic suitability, and commercial availability into a unified decision-making platform. Additionally, most AI advisory tools available in the market are not customised for Singapore's unique urban and commercial landscapes, limiting their relevance and effectiveness.

Thus, our project addresses the need for an AI-driven advisor that not only recommends profitable business locations but also suggests viable business types and provides data-driven insights on Singapore's demographic landscape and competitive environment.

2.2 Market Research

Singapore's vibrant business environment offers significant opportunities for entrepreneurs across industries, but also presents unique challenges that require strategic decision-making. While support structures like Enterprise Singapore's SME Centres provide business advisory services, these services often require manual follow-up by entrepreneurs (Enterprise Singapore, n.d.).

Meanwhile, current digital solutions only partially address business site selection needs. On one hand, property listing platforms such as 99.co and CommercialGuru allow users to browse available commercial spaces but do not offer competitor analysis or demographic suitability assessments. On the other hand, crowd analytics platforms like BestTime.app provide foot traffic insights for certain locations but are not integrated with commercial property databases nor optimised for business advisory use.

More sophisticated platforms like xMap offer advanced geospatial analytics, combining demographic, competition, and real estate data to allow users to select sites with maximum returns and optimal operations. However, these tools primarily target enterprise users, with subscription costs and technical complexity that make them less accessible to small businesses and new entrepreneurs. For example, xMap's pro plan costs almost $500 per month, placing it out of reach for some startups (xMap, n.d.). Additionally, international competitors such as BestPlace provide AI-powered location intelligence, including insights into local customer behavioural patterns, but their coverage remains concentrated in the United States and Europe, with limited applicability to Singapore's commercial landscape (BestPlace, n.d.).

Furthermore, while AI-powered business advisors such as William Pena's AI business advisor GPT provide relatively comprehensive guidance for entrepreneurs, they typically do not link users to specific property listings, requiring additional manual research on real estate platforms, or place usage restrictions unless users subscribe to a paid plan (Pena, n.d.). Other solutions like Cody's business consultant chatbot require users to manually create and integrate their knowledge bases and perform custom prompt engineering, which can be time-consuming and technically demanding for entrepreneurs seeking quick, actionable advice (Cody, n.d.).

These observations reflect broader market trends, where there is a growing demand for AI-driven, data-integrated decision tools that are both accessible and locally relevant. According to IMDA's Singapore Digital Economy 2024 report, AI adoption among non-SME firms has increased significantly, with 44% of non-SME firms adopting AI in 2023, up from 16.7% in 2018. Despite the potential benefits, SMEs face challenges such as high investment costs and insufficient infrastructure, with only 4.2% of SMEs adopting AI in 2023 (IMDA, 2024, pp. 15-16). This suggests that while larger businesses are increasingly leveraging AI for productivity and cost reduction, smaller enterprises continue to grapple with the barriers to AI adoption. Nevertheless, according to DBS' 2025 Business Pulse Check Survey, 73% of local SMEs plan to invest in Gen AI-powered solutions, implying a growing recognition of AI's potential within the SME sector, with more businesses seeking to tap into the benefits of AI-driven tools (DBS, 2025).

Given this trend, there is a clear opportunity for a more accessible, cost-effective AI-powered business advisory tool that integrates competitor analysis, demographic insights, and property listings, specifically tailored to the needs of small businesses in Singapore.

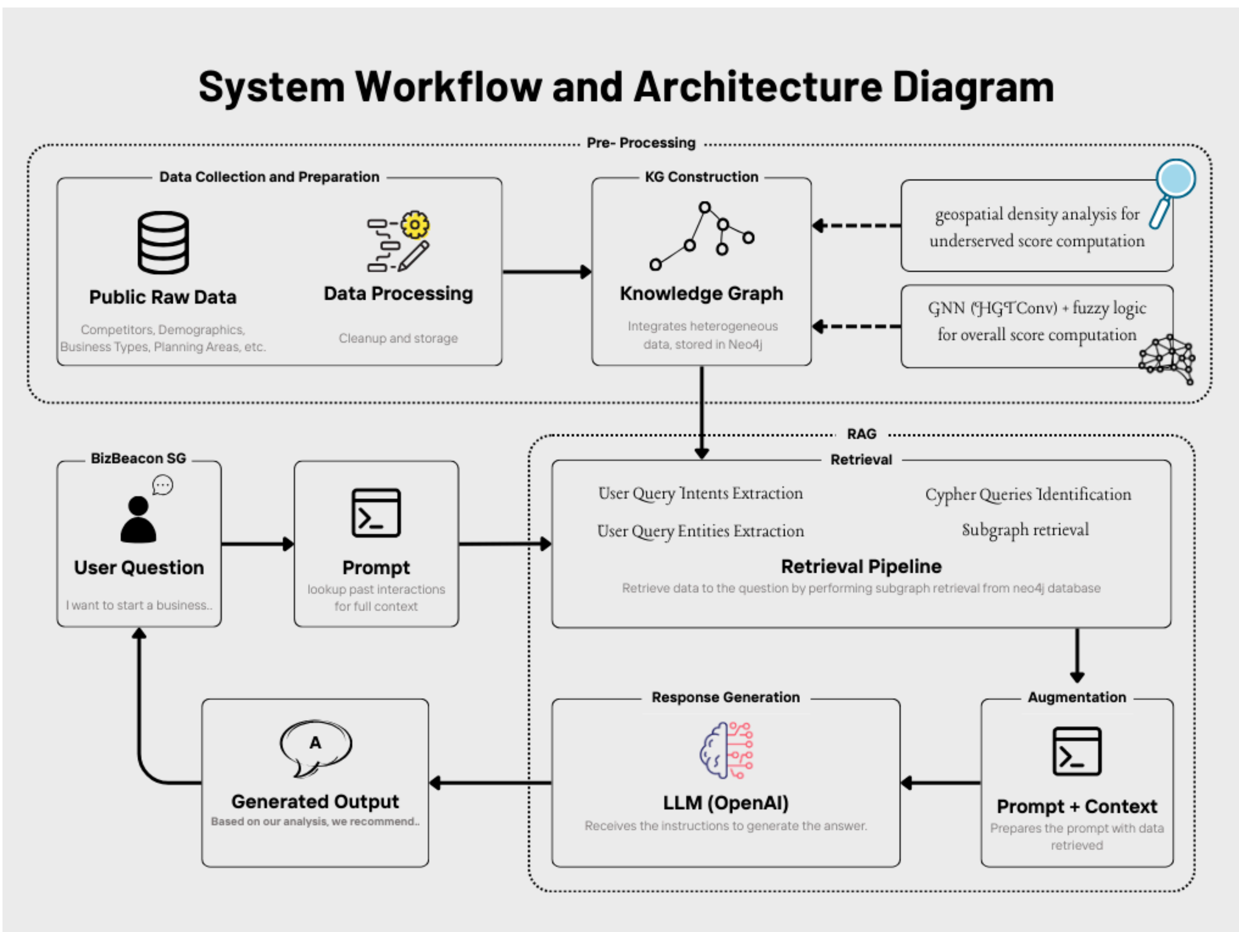
2.3 Project Objective

The main objective of this group project is to build a conversational AI chatbot using RAG with a knowledge graph, which would be capable of answering user's queries about starting businesses in Singapore, with a focus on location and business type suggestions.

3.0 Project Solution

This chapter describes the end-to-end solution for our Intelligent Singapore Business Location Advisor System. We begin with data collection and preprocessing, proceed to knowledge graph construction, then detail our hybrid scoring engine (geospatial density, fuzzy logic, and GNN), and finally explain our RAG pipeline and LLM-driven response module.

3.1 System Workflow and Architecture Diagram



This diagram illustrates the complete workflow of our system. The system is composed of four logical layers:

1. Pre-Processing:

   a. Data Collection and Preparation: Ingest raw public data (competitors, demographics, housing profiles, business types, planning areas).
   b. Data Processing: Clean, normalise and stage data for graph ingestion.
2. Knowledge Graph Construction:
   a. Knowledge Graph (Neo4j): Integrates heterogeneous entities and relationships.
3. Scoring Engine:
   a. Geospatial density analysis for computing an “underserved” score.
   b. Fuzzy Logic + GNN for overall “location suitability” scoring.
4. RAG-Enabled Chatbot:
   a. Retrieval Pipeline: Extracts user intent & entities, formulates Cypher subgraph queries, retrieves contextual data.
   b. Augmentation and LLM: Marshals retrieved subgraphs into prompts and uses an LLM (OpenAI) to perform reasoning and generate user-friendly recommendations.

By combining a flexible knowledge graph backbone, uncertainty-aware fuzzy logic, relational learning via HGTConv, and a RAG-powered LLM interface, our solution delivers both data-driven rigor and human-readable insights — equipping small businesses in Singapore with precise, explainable location recommendations.

3.2 Data Collection and Preprocessing

Sources: We collect competitor registries, census demographics, housing profiles, business-type catalogs, and planning area geodata from public sources.

Processing Steps:

1. All categorical labels are standardised. (e.g., “CAFE” vs “Coffee Shop”).
2. Venue coordinates are geocoded and age-bracket mappings are applied to determine likely patron demographics per venue type.
3. We compute population density by dividing each subzone’s total population by its land area in square kilometers.
4. We calculate competitor density for each venue type in a subzone by dividing its total competitor count by the subzone’s land area (km^2), then use the 20th, 40th, 60th, and 80th percentiles of the resulting densities as thresholds to classify each into five

levels—extremely low (≤20th), low (20th–40th), medium (40th–60th), high (60th–80th), and extremely high (>80th).

5. The cleaned and enriched data is stored in a structured database, Supabase for efficient ingestion into graph.

3.3 Knowledge Graph Construction

In Neo4j, we represent the domain using node types such as PlanningArea, VenueType, Competitor, and CompetitorStats, along with multi-typed relationships like :LOCATED_IN, :FOR_TYPE, and :HAS_COMPETITOR_STATS. The detailed implementation is explained in the "Project Implementation" section below.

3.3.1 Why a Knowledge Graph?

A knowledge graph is particularly well suited to our system for several reasons. It natively models the multi-relational structure of our domain, without flattening those relationships into rigid tables, and captures relationships between entities in a versatile and interpretable fashion. This makes it easy to traverse from any node (for example, a specific subzone) to all related entities (competitor counts, age-bracket populations) in a single query. By exposing rich, connected subgraphs, the knowledge graph directly supports both our scoring pipeline and the RAG interface. We can retrieve exactly the nodes and relationships an LLM needs to generate context-aware, explainable recommendations.

3.4 Hybrid Scoring Engine

3.4.1 Geospatial Density for Underserved Score

We derive an underserved metric by comparing the spatial distribution of competitors against local population clusters. This score highlights subzones where demand (population) significantly outweighs supply (competitor presence), signaling white-space opportunities for new entrants.

Rather than relying on raw counts alone, geospatial density analysis captures the real-world spatial context — identifying not just how many competitors exist, but where they are clustered relative to where people live. By normalising competitor locations against population density, we surface areas that are genuinely underserved rather than simply low on absolute competitor numbers.

3.4.2 Fuzzy Logic for Overall Score

We apply fuzzy logic to model the inherently ambiguous nature of real-world criteria like competitor density, local population, and spending power. Rather than imposing hard cutoffs, fuzzy logic maps each metric onto smooth “low,” “medium,” and “high” membership functions. This approach yields an overall suitability score that closely mimics expert reasoning under uncertainty and smooths abrupt threshold transitions, and provides clear explanations (e.g., “This area scores 0.7 on population and 0.4 on competitor density, leading to an overall suitability of 68”). Because the computations reduce to simple weighted combinations of fuzzified inputs, the method remains transparent and easy to adjust as domain knowledge evolves. The base weight for each input and the venue-type multiplier are defined based on our own common knowledge and understanding of business. With an overall score computed, the system orders and recommends the most suitable location to start a business.

3.4.3 Graph Neural Network Enhancement

We enhance our scoring pipeline with a heterogeneous GNN (HGTConv) that learns latent representations for each node by propagating information across linked entities — planning areas, demographics, competitors, and housing profiles. This approach fills gaps where raw data are sparse by borrowing context from related nodes. For example, there are certain subzones that lack sufficient competitor data for specific venue types. By integrating the GNN’s embeddings into our scoring mix, we achieve more robust and consistent rankings, especially in areas with incomplete or noisy inputs, yielding a more data-driven overall score.

3.5 Retrieval-Augmented Generation (RAG) Pipeline

Intent & Entity Extraction: We use prompt-engineered LLM calls to parse user queries (e.g., “I want to open a cafe in Bugis”) and extract the target venue type and subzone / planning area.

Cypher Query Identification & Subgraph Retrieval: We define a library of parameterised Cypher templates that retrieve the relevant subgraph, which are nodes, relationships, and key attributes (scores, densities) for the identified venue type and location. The LLM dynamically selects and populates the appropriate template at runtime.

Prompt Augmentation: We embed the retrieved graph insights into the prompt, ensuring the LLM has accurate, up-to-date information and reducing the risk of hallucinations.

LLM Invocation: The augmented prompt is supplied to OpenAI's model, which then produces a concise, actionable recommendation tailored to the user's query.

3.6 LLM-Based Response Generation

Role of the LLM: We leverage LLM (OpenAI) to transform numeric scores and graph facts into a coherent recommendation, for example:

"Based on our analysis, the best location to start a cafe business is Dover. Here is the reasoning:

1. Competitor Density and Underserved Score: …

2. Population and Housing: …

3. Overall Score: …"

Benefits:

1. Users can ask follow-up questions and clarification seamlessly in conversation without breaking context.
2. Each recommendation cites underlying metrics (e.g., underserved score of 80), fostering trust.
3. The LLM reduces cognitive load because complex multi-dimensional data is distilled into clear, prioritised bullet points or narratives. It also ensures uniform presentation of recommendations, minimising interpretation errors.
4. Non-technical stakeholders can understand and act on insights without reading raw data or Cypher output.
5. New data streams can be fed into the graph and surfaced automatically.

By leveraging an LLM in this way, we surface advanced analytics and reasoning results to users by transforming rich graph data into guidance that is both explainable and easy to understand.

4.0 Project Implementation

4.1 Knowledge Extraction

The group identified a list of raw data relevant for the system to consider before returning recommended locations for a particular type of business:

- Singapore map data for planning areas and subzones
- Demographics data for planning areas and subzones
- Industrial property listings

4.1.1 Singapore Planning Areas and Subzones Data

Under the Master Plan 2019 Planning Area Boundary by the Urban Redevelopment Authority (URA), Singapore is divided into 5 regions and 55 planning areas. Each planning area has several subzones, totalling to 332 subzones. The GeoJSON dataset was taken from data.gov.sg (https://data.gov.sg/datasets/d_4765db0e87b9c86336792efe8a1f7a66/view).

A Python script was used to process and extract fields such as subzone names, planning area names, regions, and coordinate boundaries. The data was then added into the table planning_areas inside Supabase.

Key steps performed by the script:

1. Parsing HTML fields:
    a. The GeoJSON file contains metadata fields (Description) encoded in HTML format. The script used BeautifulSoup to parse and extract subzone names (SUBZONE_N), planning area names (PLN_AREA_N), and regions (REGION_N).
2. Handling geometry types:
    a. The geometry field was processed to handle both Polygon and MultiPolygon structures, ensuring all boundary coordinates were correctly flattened and extracted.
3. Computing boundary extents. For each subzone, the script calculated:

    a. Minimum and maximum latitude
    b. Minimum and maximum longitude
4. Saving structured output:
    a. The processed data was stored in a CSV file (planning_areas_with_coords.csv) containing the following fields:
        i. Subzone
        ii. Planning Area
        iii. Region
        iv. Min/Max Latitude
        v. Min/Max Longitude
        vi. Geometry Type (Polygon/MultiPolygon)
        vii. Raw Coordinates (as JSON)

This structured CSV output was later imported into Supabase to support geospatial analysis in the system. This preprocessing step allowed the system to later perform spatial matching of businesses and properties based on their geographic location within Singapore. Below is the schema of the planning_areas table in Supabase:

Column Name	**Data Type**	**Description**
id	serial (Primary Key)	Unique identifier for each record
subzone	text	Name of the subzone
planning_area	text	Name of the planning area
region	text	Name of the region (e.g., Central, East)

min_latitude	double precision	Minimum latitude coordinate boundary
max_latitude	double precision	Maximum latitude coordinate boundary
min_longitude	double precision	Minimum longitude coordinate boundary
max_longitude	double precision	Maximum longitude coordinate boundary
created_at	timestamp	Record creation timestamp (default: current time)
updated_at	timestamp	Record update timestamp (default: current time)
geometry_type	text (nullable)	Type of geometry shape (Polygon or MultiPolygon)

coordinates	text (nullable)	Geometry coordinates in JSON string format

4.1.2 Demographics Data

The Demographics data is extracted from the Singapore's Department of Statistics census data (https://www.singstat.gov.sg/publications/reference/cop2020/cop2020-sr1/census20_stat_release1). While the population and subzone size are from raw data, the population density is derived data from the population and subzone size.

Column Name	**Data Type**	**Description**
planning_area	text	Name of the planning area
subzone	text	Name of the subzone, which is subsidiary of planning area
population_chinese_male, population_chinese_female, population_malays_male, population_malays_female, population_indians_male, population_indians_female, population_others_male, population_others_female	integer	Population of each race/gender in a given subzone
subzone_size	double precision	Subzone's area in km^2
population_density	double precision	All population in the subzone divide by subzone's area size, given in population/km^2

Note that this data is not current-to-date as it is only published once every 10-year-cycle, the last updated data is from 2020. Nonetheless, this demographics data still gives a good indication of population density of the planning areas and subzones.

4.1.3 Industrial Property Data

The group scraped industrial property listings from 99.co (https://www.99.co/singapore/commercial/rent/industrial) using its backend API. Before inserting into the industrial_properties table in Supabase, only a certain set of fields are extracted per listing. The schema for the industrial_properties table are listed below:

Column Name	Data Type	Description	Notes
property_id	serial (auto-increment)	Unique ID for the property	Primary Key
listing_id	character varying(50)	Unique listing identifier	Unique constraint
property_segment	character varying(50)	Segment category (e.g., industrial)	
listing_type	character varying(20)	Sale or rental type	
main_category	character varying(50)	Main property category	
sub_category	character varying(50)	Sub-category of the property	
status	character varying(20)	Listing status	Default: active
description	text	Description of the property	Nullable

price	numeric(15,2)	Price of the property	
area_size	numeric(10,2)	Size of the property (sqft or sqm)	
area_ppsf	numeric(10,2)	Price per square foot	Nullable
district_number	integer	District number	Nullable
postal_code	character varying(10)	Postal code	Nullable
latitude	double precision	Latitude coordinate	Nullable
longitude	double precision	Longitude coordinate	Nullable
address_name	text	Address or property name	Nullable
closest_mrt	character varying(100)	Nearest MRT station	Nullable
photo_url	text	URL to property image	Nullable
listing_url	text	URL to the property listing page	Nullable
location	geography	Spatial geography field	Nullable
created_at	timestamp	Record creation timestamp	Default: now()

updated_at	timestamp	Record last update timestamp	Default: now()
planning_area	text	Planning area name	Nullable
subzone	text	Subzone name	Nullable

The data inside planning_areas is used to link the planning area and subzone for each property listing by checking the listing's coordinates against each set of coordinates for all subzones.

4.1.4 Competitors and Establishments Data

We used a Python script to extract and populate business location data into the table in Supabase called "establishments". Below are the key steps of the script.

1. Data Retrieval from Geoapify API
    a. The script used the Geoapify Places API (https://apidocs.geoapify.com/playground/places) to search and retrieve different types of businesses across Singapore, categorised by specific types (e.g., supermarkets, cafes, clinics, shopping malls).
    b. Singapore was divided into four quadrants (North-West, North-East, South-West, South-East) using specific Geoapify place_id filters to ensure comprehensive geographical coverage.
    c. Categories were selected based on predefined business types such as "commercial.shopping_mall", "catering.restaurant", "healthcare.clinic_or_praxis", "education.school", etc. These categories are a preset list by Geoapify.
2. Data insertion into Supabase
    a. Each retrieved business record (venue) was parsed and inserted into the establishments table.
    b. Key fields stored include:
        i. venue_name
        ii. venue_address
        iii. latitude

        iv. longitude
        v. venue_type
        vi. region (temporarily blank, populated later)
        vii. planning_area (populated in the next step)

3. Populating Planning Area and Subzone details
    a. Some records initially did not have these information
    b. For those with such missing details
        i. The script called the OneMap Singapore API using latitude and longitude to retrieve the associated planning area.
    c. For missing subzones:
        i. The script used previously extracted polygon geometries (from the planning areas GeoJSON) to perform point-in-polygon matching with the Shapely library.
        ii. If a business location fell within a subzone polygon, it was assigned that subzone accordingly.

The final output is the establishments table which is a dataset complete with establishments' venue names, addresses, coordinates, planning areas and subzones. This data supports some of the downstream analyses such as geospatial density analysis. Below is the schema for the establishments table:

Column Name	Data Type	Description
id	serial (Primary Key)	Unique identifier for each establishment
subzone	text (nullable)	Subzone where the establishment is located
planning_area	text (nullable)	Planning area where the establishment is located

region	text (nullable)	Region of Singapore (e.g., North-East, Central)
latitude	double precision	Latitude coordinate of the establishment
longitude	double precision	Longitude coordinate of the establishment
original_type	text (nullable)	Original category type from external source (e.g., API)
rank	integer (nullable)	Rank based on external scoring (if available)
venue_name	text	Name of the establishment
venue_address	text (nullable)	Full address of the establishment
venue_id	text (nullable)	ID of the venue from external data source

avg_weekday_footfall	double precision (nullable)	Average weekday footfall (if available)
avg_weekend_footfall	double precision (nullable)	Average weekend footfall (if available)
created_at	timestamp	Record creation timestamp (default: current time)
updated_at	timestamp	Record update timestamp (default: current time)
venue_type	text (nullable)	Mapped internal venue type (e.g., RESTAURANT, CAFE, etc.)

4.1.5 Business Types Data

We classify each business or establishment under one of the following 11 categories to make it more constrained and not too granular:

type_name
ARTS
APPAREL
CAFE

CLUBS
DOCTOR
RESTAURANT
SHOPPING
PERSONAL_CARE
SCHOOL
VEHICLE
SPORTS_COMPLEX

These types were taken from BestTime (https://documentation.besttime.app/#response-attributes-query-venue) as we originally intended to use their APIs to extract footfall data, but eventually decided against it. This is further elaborated under the limitations in section 6.2.

4.2 Knowledge Representation

The group uses Supabase to store relational data and Neo4j to store graph data.

4.2.1 Knowledge Graph Construction

4.2.1.1 Detailed Explanation

In our system, the knowledge graph is built through a repeatable pipeline that begins by pulling in all of our domain tables from Supabase, which are planning areas, venue types, demographic profiles, competitor listings, precomputed competitor metrics, and industrial property listings. We handled pagination as needed to ensure a complete data pull. Once the data is loaded into memory, the script clears any existing graph in Neo4j with a single DELETE operation, guaranteeing that each run starts from a clean slate.

A new Neo4j session is then opened, and entities are inserted in a logically sequenced manner:

1. Spatial and categorical baselines

   a. Every subzone becomes a PlanningArea node (using a normalised uppercase identifier).
   b. Each business category or classification becomes a VenueType node.
2. Establishments
   a. Every valid establishment record is represented as a Competitor node, and linked to its corresponding planning area (:LOCATED_IN) and venue type (:OF_TYPE), modeling the fact that businesses both reside in a subzone and belong to a business type.
3. Competitor Stats
   a. For each (subzone, venue_type) pair, a CompetitorStats node captures overall score, underserved score, density, and competitor count, and is connected back to its planning area (:HAS_COMPETITOR_STATS) and its venue type (:FOR_TYPE). This dual linkage allows us to traverse from geography to performance data or from category to local intensity. The calculations of overall score, underserved score, and density are explained in the Knowledge Inference section below.
4. Demographics
   a. Demographic profiles are handled similarly: for each subzone, we create three distinct nodes, which are AgeDistribution, HousingProfile, and PopulationStats, and each of them carry the full set of original fields. They are attached to their planning area via dimension-specific relationships (:HAS_AGE_DISTRIBUTION, :HAS_HOUSING_PROFILE, :HAS_POPULATION_STATS), ensuring that demographic context is directly accessible from any location.
5. Industrial properties
   a. Industrial real-estate data is represented in two layers. A PropertiesAvailable node aggregates availability for each subzone, and individual IndustrialProperty nodes hold detailed listing metadata (price, size, type, status). Each listing is linked to its aggregator (:HAS_PROPERTY), and the aggregator in turn is linked to its planning area (:OFFERS_PROPERTIES).
   b. We then enrich each PropertiesAvailable node with average price attributes. For example, averagePrice_factory_rent is derived from our property listings table, so that pricing benchmarks are immediately available in the graph. To compute these averages, we group all listings by subzone, property subcategory, and

listing type (rent or sale), then calculate the mean price for each grouping. This yields a granular pricing profile for every category and listing type in each subzone.

6. Automated Validation
    a. Finally, a validation step iterates over every source record and programmatically confirms the presence of its corresponding node and relationships in Neo4j. Any missing element is flagged for inspection, closing the loop between our Supabase database and the graph model. By capturing geography, competition, demographics, and real-estate listings as interconnected nodes and edges, this knowledge graph provides a richly linked foundation for our RAG-enabled AI location advisor.

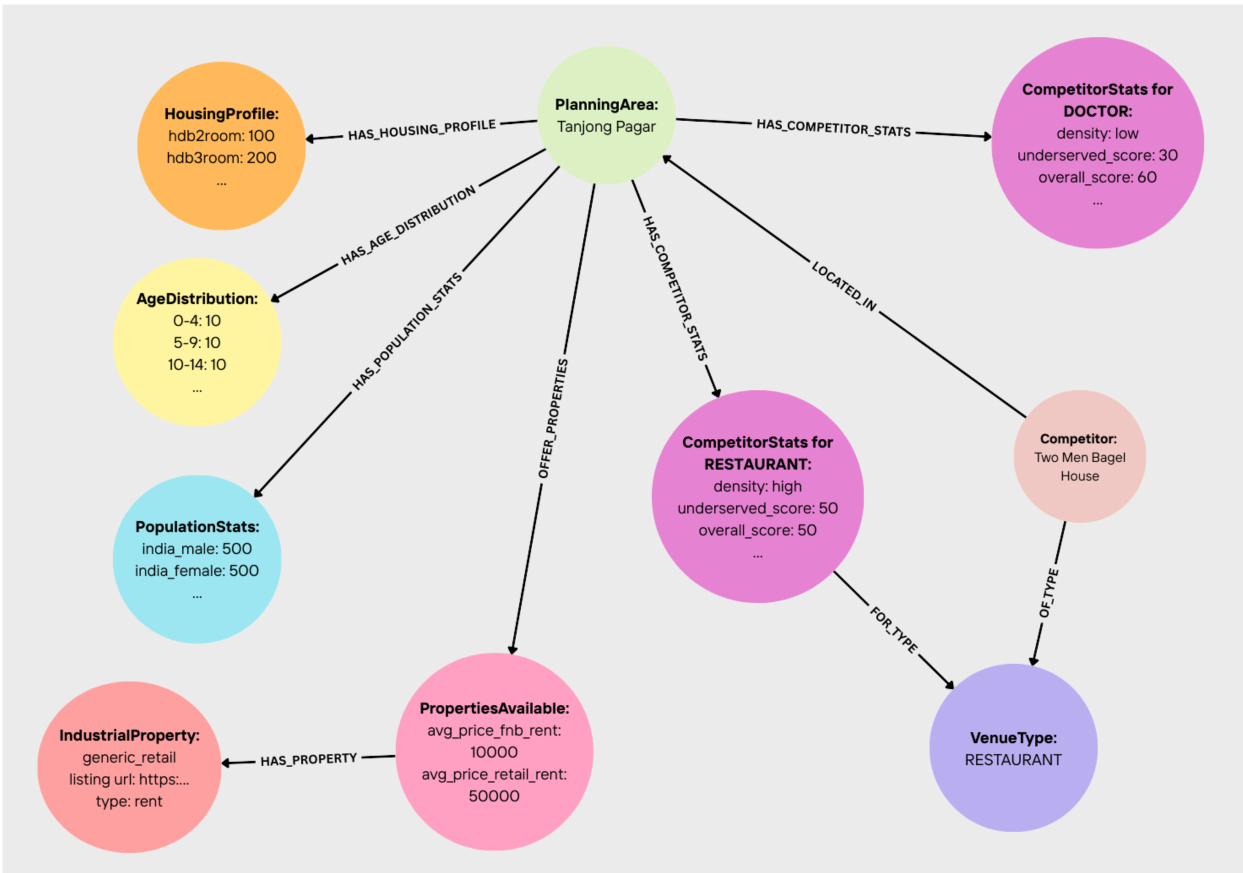
4.2.1.2 Graph Schema

Below is the graph schema in table format.

From (Label)	Relationship	To (Label)	Cardinality
PlanningArea	:HAS_COMPETITOR_STATS	CompetitorStats	1 planning area -> N stats (by type)
CompetitorStats	:FOR_TYPE	VenueType	1 stats -> 1 type
Competitor	:LOCATED_IN	PlanningArea	N competitors -> 1 area
Competitor	:OF_TYPE	VenueType	N competitors -> 1 type
PlanningArea	:HAS_AGE_DISTRIBUTION	AgeDistribution	1 area -> 1 age profile
PlanningArea	:HAS_HOUSING_PROFILE	HousingProfile	1 area -> 1 housing profile
PlanningArea	:HAS_POPULATION_STATS	PopulationStats	1 area -> 1 population profile
PlanningArea	:OFFER_PROPERTIES	PropertiesAvailable	1 area -> 1 properties aggregator
PropertiesAvailable	:HAS_PROPERTY	IndustrialProperty	1 aggregator -> N individual listings

4.2.1.3 Graph Diagram

Below is a graph diagram illustrating one of our PlanningArea nodes, its neighboring nodes, and the relationships between them.



4.3 Knowledge Inference

4.3.1 Geospatial Density Analysis

To identify underserved subzones for different business types (venue_type), we did some analysis using the data from our Neo4j knowledge graph (KG) and Supabase data.

For each establishment (Competitor nodes) in the KG, we tagged the longitude and latitude coordinates and linked it to its corresponding subzone.

We then calculated an “underserved_score” metric for each pair of (subzone, venue_type) using the formula below:

$$\text{underserved\_score} = \frac{\text{population density}}{\text{competitor count} + 1}$$

The result is then normalised so the final output is between a range of 0 to 100. A high underserved_score means there are very few competitors serving a large population size. In general, this could mean potential business opportunities. Below are some of the data that act as CompetitorStats nodes in the KG:

name	subzone	venue_type	underserved_score
"Stats for TIONG BAHRU STATION ARTS"	"TIONG BAHRU STATION"	"ARTS"	100.0
"Stats for TIONG BAHRU STATION CAFE"	"TIONG BAHRU STATION"	"CAFE"	100.0
"Stats for TIONG BAHRU STATION CLUBS"	"TIONG BAHRU STATION"	"CLUBS"	100.0

This could mean that Tiong Bahru is currently lacking in clubs, retail outlets and personal care establishments.

On the other hand for places such as below:

name	subzone	venue_type	underserved_score
"Stats for PHILLIP APPAREL"	"PHILLIP"	"APPAREL"	0.0
"Stats for RAFFLES PLACE APPAREL"	"RAFFLES PLACE"	"APPAREL"	0.0
"Stats for PHILLIP CAFE"	"PHILLIP"	"CAFE"	0.0

It could mean the business types for these subzones are already saturated so opening an apparel shop or cafe in these areas could be more challenging in terms of attracting customers.

But it is important to note that this metric is purely quantitative. It does not account for underlying real-world context. For example, while Raffles Place may seem underserved for tuition centres, it is primarily a central business and office district, so the actual student foot traffic is low. This shows that while the underserved score highlights interesting gaps, human judgment and contextual reasoning are still important when interpreting the results, though it does indicate on a general level in terms of level of demand and supply.

4.3.2 Graph Neural Network & Fuzzy Logic

Feature Engineering and Adaptive Thresholds

In this stage, the system combines three core attributes for each subzone–venue type pair:

1. Competitor count - Pulled directly from the CompetitorStats node. If the value is missing, the record is flagged and left out of the threshold setting. The reason for valid values at the 33rd and 67th percentiles (terciles) is to balance sensitivity and robustness. Terciles are less affected by extreme outliers than quartiles or deciles but still give enough detail for our fuzzy rules.
2. Relevant population - For each venue type, only the age groups most likely to visit are counted. This stops demand estimates from being skewed by residents outside a venue's main customer base, improving both rule-based and graph learning accuracy.

   The mapping used is as follows:

Venue Type	Age cohorts
SCHOOL	5 - 9, 10 - 14, 15 - 19
CAFE	20 - 24, 25 - 29, 30 - 34
RESTAURANT	25 - 29, 30 - 34, 35 - 39, 40 - 44
DOCTOR	60 - 64, 65 - 69, 70 - 74
APPAREL	25 - 29, 30 - 34, 35 - 39
ARTS	20 - 24, 25 - 29, 30 - 34
CLUBS	20 - 24, 25 - 29, 30 - 34
SHOPPING	25 - 29, 30 - 34, 35 - 39, 40 - 44
PERSONAL CARE	35 - 39, 40 - 44, 45 - 49
VEHICLE	25 - 29, 30 - 34, 35 - 39, 40 - 44
SPORTS COMPLEX	20 - 24, 25 - 29, 30 - 34

3. Spending power - Estimated from housing counts, with higher weights for larger or more expensive homes, and then summed across all housing types.

   The weights applied are as follows:

Housing Type	Weight Assigned
HDB 2 room	1
HDB 3 room	1.5
HDB 4 room	2
HDB 5 room ea	2.5
Condominiums and apartments	3
Landed properties	3
Others	1.0

After extraction, each list is cleaned of missing values and sorted. Dynamic thresholds are set at the first and second terciles, so “few” vs “many” competitors (and likewise for population and spending) match the real data. Since this step runs at the start of every pipeline, thresholds adjust automatically as the data changes.

Finally, linear fuzzy-membership functions convert raw values to a 0–1 scale:

- Score = 1 at or below the lower threshold
- Score = 0 at or above the upper threshold
- Linearly scaled in between.

In our dataset of 3652 subzone–venue records, 1048 have valid competitor counts between 1 and 144. Sorting those 1048 values yields first and second terciles at 2 and 5. A record with comp_count = 1 gets a “few” membership of 1.0, while one with comp_count = 4 maps to (5−4)/(5−2)=0.33.

Fuzzy Rule-Based Scoring

These scores feed 6 soft rules that add context beyond raw numbers. Each rule focuses on a different factor so no single metric outweighs the rest:

a. Few competitors in a high-population area
b. Many competitors
c. A standard baseline condition
d. High population density
e. High consumer spending
f. An area with high spending but few options

Each rule starts with a base weight that is then scaled by a venue-specific multiplier from RULE_IMPORTANCE (e.g., restaurant, school). This lets the model reflect different business goals. For instance, a restaurant may set a 3.5× multiplier on “many competitors” to avoid crowded markets, while a school may double the “high population” rule to seek dense areas.

For every rule we:

- Find its activation value (the fuzzy score or a combination of scores).
- Multiply by the rule’s base weight and the venue multiplier.

The sum of these weighted activations is the numerator. Dividing by the sum of all applied weights keeps the final rule score between 0 and 1. If competitor data is missing, the engine skips all rules except the baseline and returns a neutral score.

Example:

For a CAFE business at Dover:

Soft rule	Fuzzy membership μ	Base weight w	Multiplier m*	Term μ × w × m
few_comp_high_pop	0	1	2	0
many_competitors	1	2	1.5	3
medium_default	0	4	1	0
high_population	1	1.5	1	1.5

high_spending	1	1.5	1.2	1.8
underserved_high_spend	0	2.5	1.8	0
Totals numerator				6.300

The denominator is the sum of weights x multipliers: 16.8.

Final rule score = numerator/denominator = 6.3 / 16.8 ≈ 0.375.

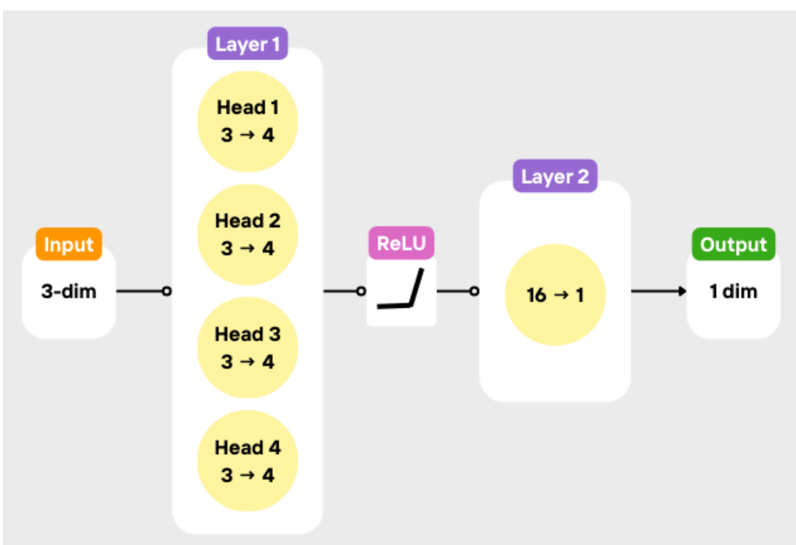
So the cafe in Dover receives a normalised soft-rule score of approximately 0.375 before any contrast stretch, GNN blending or underserved-score adjustments.

Heterogeneous Graph Transformer Model

To uncover hidden spatial and business links, we build a heterogeneous graph:

- Nodes: Each node is a subzone–venue-type record with the feature vector [competitor_count, relevant_population, spending_power].
- Edges:
  - same_subzone: connects all nodes in the same subzone.
  - same_type: connects all nodes of the same venue type.

This captures both geographic closeness and business similarity. The diagram below illustrates the graph structure.



A two-layer Heterogeneous Graph Transformer (HGTConv) processes the graph:

1. Layer 1 expands features from 3 to 16 dimensions, using 4 attention heads and the ReLU activation function.
2. Layer 2 squeezes the 16-dim vector down to a single value per node.

After inference, we run min-max normalisation on all node outputs to get GNN score values between 0 and 1, keeping them on the same scale as the fuzzy scores.

Example:

1. Original Features (CAFE at Dover):
   [comp_count=8, pop_relevant=2500, spending_power=27985]
2. Some of the neighbor snapshots (first HGT layer):

Neighbour	Feat (c, p, s)	conv1
RESTAURANT	[21, 0, 27985.0]	[3759.588, 0.0, 0.0, 3593.427, …]
PERSONAL CARE	[3, 0, 27985.0]	[3761.093, 0.0, 0.0, 3594.738, …]
DOCTOR	[2, 1860, 27985.0]	[4500.903, 0.0, 0.0, 3165.578, …]

3. Raw GNN Output (after second HGT Layer)

   Raw_out [CAFE]: 1438.711

4. Normalised Score (min-max)

   normalised_score [CAFE] = (1 438.711 – min) / (max – min) ≈ 0.688

Hybrid Score Blending

The final score blends the 3 aforementioned parts:

1. Rule vs GNN mix - A blend factor α (env-set, default 0.6) controls the weight on the GNN score: higher α represents more GNN influence.

2. Contrast tweak - In our preliminary execution, there are many records clustered near 0.5, and reduced actionable differentiation. We then multiplied deviations by 1.3 (midpoint 0.5). This spreads clustered mid-range values so they are easier to rank. Here is the formula where 1.3 is the tweak:

   *0.5 + (original_rule_score - 0.5) * 1.3*
3. Underserved boost - The blended score is finally merged with the underserved score from density analysis at a fixed 35% weight to recognise areas short of options.

Example: For a CAFE at Dover, let the contrast-adjusted rule score be 0.375, the GNN score be 0.688, and the existing underserved score be 2.18. First, apply contrast adjustment on the rule score: 0.5 + (0.375 - 0.5) * 1.3 = 0.3375. Then, blend the updated rule score and GNN: 0.6×0.688 + 0.4×0.3375 = 0.5478. The value is then scaled to a percentage, which is 54.78%. After that, incorporate the underserved score: 54.78 x 0.65 + 2.18×0.35 = 36.37. Hence, the overall score for a CAFE at Dover would be 36.37%.

4.4 Predefined Cypher Queries

To help with the retrieval of relevant information from our knowledge graph, we defined a list of 11 Cypher queries to call depending on the user intents we identify. We grouped queries into 11 possible intents, including queries related to business advice given a venue type and/or area, as well as queries involving specific information regarding a venue type or area, such as competitor information, population and demographics statistics, and properties information.

All our predefined queries first involve searching a fulltext index (a planning area index or venue type index or both) for the entity or entities identified and passed into the query from the user input. Each fulltext index search is then followed by logic involving match and return clauses to achieve the desired outputs of each query depending on their respective intents. The outputs of all queries are strings with lines formatted in a “graph-like” structure, as follows: “Node 1 name [Node 1 type] {Node 1 properties} - Relation type -> Node 2 name [Node 2 type] {Node 2 properties}”. For instance, the figure below shows a part of the output of the query called to get competitor information in Katong, a neighbourhood within the Marine Parade planning area.

KATONG [PlanningArea] {"subzone":"KATONG"} - HAS_COMPETITOR_STATS -> Stats for KATONG ARTS [CompetitorStats] {"density":"medium","venue_type":"ARTS","name":"Stats for KATONG ARTS","underserved_score":6.6,"subzone":"KATONG","overall_score":74.81,"competitor_count":2}

Stats for KATONG ARTS [CompetitorStats] {"density":"medium","venue_type":"ARTS","name":"Stats for KATONG ARTS","underserved_score":6.6,"subzone":"KATONG","overall_score":74.81,"competitor_count":2} - FOR_TYPE -> ARTS [VenueType] {"type_name":"ARTS"}

Juz Art Studio [Competitor] {"venue_name":"Juz Art Studio","venue_type":"ARTS","latitude":1.305329,"subzone":"KATONG","longitude":103.9038162} - LOCATED_IN -> KATONG [PlanningArea] {"subzone":"KATONG"}

Juz Art Studio [Competitor] {"venue_name":"Juz Art Studio","venue_type":"ARTS","latitude":1.305329,"subzone":"KATONG","longitude":103.9038162} - OF_TYPE -> ARTS [VenueType] {"type_name":"ARTS"}

4.5 Retrieval Augmented Generation (RAG)

We implemented a RAG system that enhances a LLM with structured knowledge retrieved from a Neo4j graph database. The RAG system was designed specifically for a Business Location Advisor application focused on Singapore, where users can ask complex questions like "*Where should I open a cafe in Bukit Timah?*" or "*What is the age distribution in Tampines?*".

The RAG chain works as follows:

- It classifies the user's question into one or more intents.
- It extracts entities (e.g., specific business types like "CAFE" or locations like "Bedok") from the question or prior chat history.
- It maps the recognised intent(s) to predefined Cypher queries that are dynamically parameterised using the extracted entities.
- It retrieves the most relevant data from the Neo4j knowledge graph.
- Finally, it generates a response using a customised prompt that ensures concise, well-structured business advice based on the retrieved information.

This RAG approach ensures that the LLM responses are both factually grounded and contextually relevant to real-world business planning needs.

4.5.1 Intent Extraction

Intent extraction is a critical first step in the RAG pipeline that ensures user questions are properly mapped to backend graph queries. This was implemented using a lightweight IntentClassifier module powered by an LLM with a custom-designed prompt.

The intent classifier:

- Presents the LLM with a fixed list of 11 allowable intents (e.g., "competitor information in given planning area", "business advice for given venue type at given planning area").
- Enforces strict matching rules where the LLM is not allowed to paraphrase or invent new intents.
- Applies important disambiguation rules:
    - If both business type and location are mentioned → classify as business advice.
    - If only location is mentioned → classify as business type suggestion.
    - If only business type is mentioned → classify as location suggestion.

Example:

- Input: *"Where should I open a cafe?"*
    - Intent: *location suggestion given a business type*
- Input: *"What business should I start in Bedok?"*
    - Intent: *business type suggestion given a planning area*

This controlled, deterministic intent extraction enables direct mapping to the correct predefined Cypher query templates in **`predefined_queries_graph_like.py`**.

4.5.2 Entity Extraction

Once the intent is determined, the next step is entity extraction, which identifies key structured fields that parameterise the Cypher queries. Specifically, two entities are extracted:

- Venue Type (e.g., "CAFE", "APPAREL", "SCHOOL")
- Planning Area (e.g., "Tampines", "Jurong West", "Bedok")

Entity extraction is implemented via an LLM chain with structured output validation using Pydantic (`Entities` class). The system prompt clearly defines:

- Allowed venue types, strictly from a hardcoded list of 11 options.
- Planning areas, limited to actual Singapore subzones.

If the user's question lacks any required entity (e.g., the location isn't specified), the system attempts to recover missing entities from prior chat history to maintain conversation continuity.

Example:

- Input: *"Tell me about opening a restaurant."*
- Chat History: *"I'm interested in places around Clementi."*
- Extraction: Venue Type = "RESTAURANT", Planning Area = "Clementi"

This structured entity extraction ensures that database queries are accurately parameterised, even across multi-turn conversations.

4.5.3 Prompt Engineering

Prompt engineering was a core technique used to make the entire RAG pipeline robust, safe, and performant. We designed and deployed multiple specialised prompts:

- Intent Classification Prompt
  - Highly restrictive: only allows returning one or more exact matches from the list of intents.
  - Contains system instructions enforcing no paraphrasing and no hallucination.
- Entity Extraction Prompt
  - Strongly constrained: only specific venue types and recognised Singapore planning areas.
  - Clear fallback behavior ("Return None if not found") prevents extraction of irrelevant or hallucinated values.
- Condense Follow-up Question Prompt
  - Rephrases follow-up questions into standalone queries.
  - Ensures preservation of any business types or planning areas mentioned in either the history or follow-up.

- Answer Generation Prompt
    - Instructs the LLM to reason carefully based on retrieved graph data.
    - Requires the model to explicitly note if any data was missing (e.g., “no results found, answering based on general knowledge”).
    - Mandates Markdown formatting with clear headings, bullet points, tables, and emphasised reasoning.

Overall, the prompt engineering strategy carefully shapes the model's behavior to maximise accuracy, transparency, and user trust in the generated business advice.

4.6 Frontend

The frontend of BizBeacon SG is a modern, highly responsive web application designed to deliver an intuitive user experience for business location analysis. It emphasises scalability, maintainability, and performance.

Key Technologies and Structure:

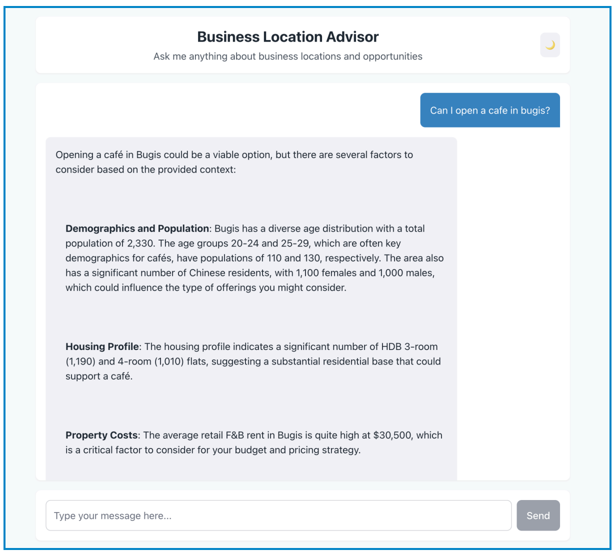
- React and Tailwind CSS form the backbone of the user interface (UI), promoting modularity and rapid development.
- Configuration files such as **`tailwind.config.js`** and **`postcss.config.js`** are present, allowing customised styling and post-processing setups.
- A clear separation of concerns is maintained:
    - src/: Contains all source code organised into modular components.
    - public/: Stores static assets for efficient delivery.

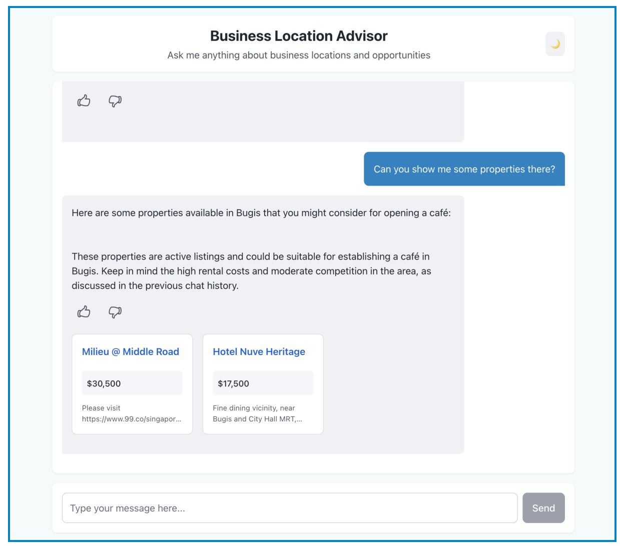
Development and Build Management:

- Dependency management is handled with **`package.json`** and **`package-lock.json`**, ensuring reproducible builds and smooth third-party library integrations.
- Modern JavaScript tooling supports streamlined build processes and enhances development efficiency.

Key Features:

- Responsive Design: Tailwind CSS ensures optimal performance across desktops, tablets, and mobile devices.
- Modular Codebase: Organised architecture under the **`src`** directory supports easy maintenance and future feature expansions.
- Efficient Asset Handling: Static assets are cleanly managed through the **`public`** folder, optimising load times.
- Scalable Development Practices: The setup encourages consistent coding standards and scalability for growing user needs.





5.0 Project Performance and Validation

5.1 Chatbot Response Evaluation

To evaluate our system, we tested it against 4 realistic user scenarios and compared its performance with that of ChatGPT. For reference and comparison, we also provide our chatbot's responses and ChatGPT's responses for Conversation 1 listed below.

Conversation 1: Opening a Cafe

User: "I'm interested in opening a cafe in Singapore. Where should I consider?"

User: "What is the competitor situation like in the top recommended area?"

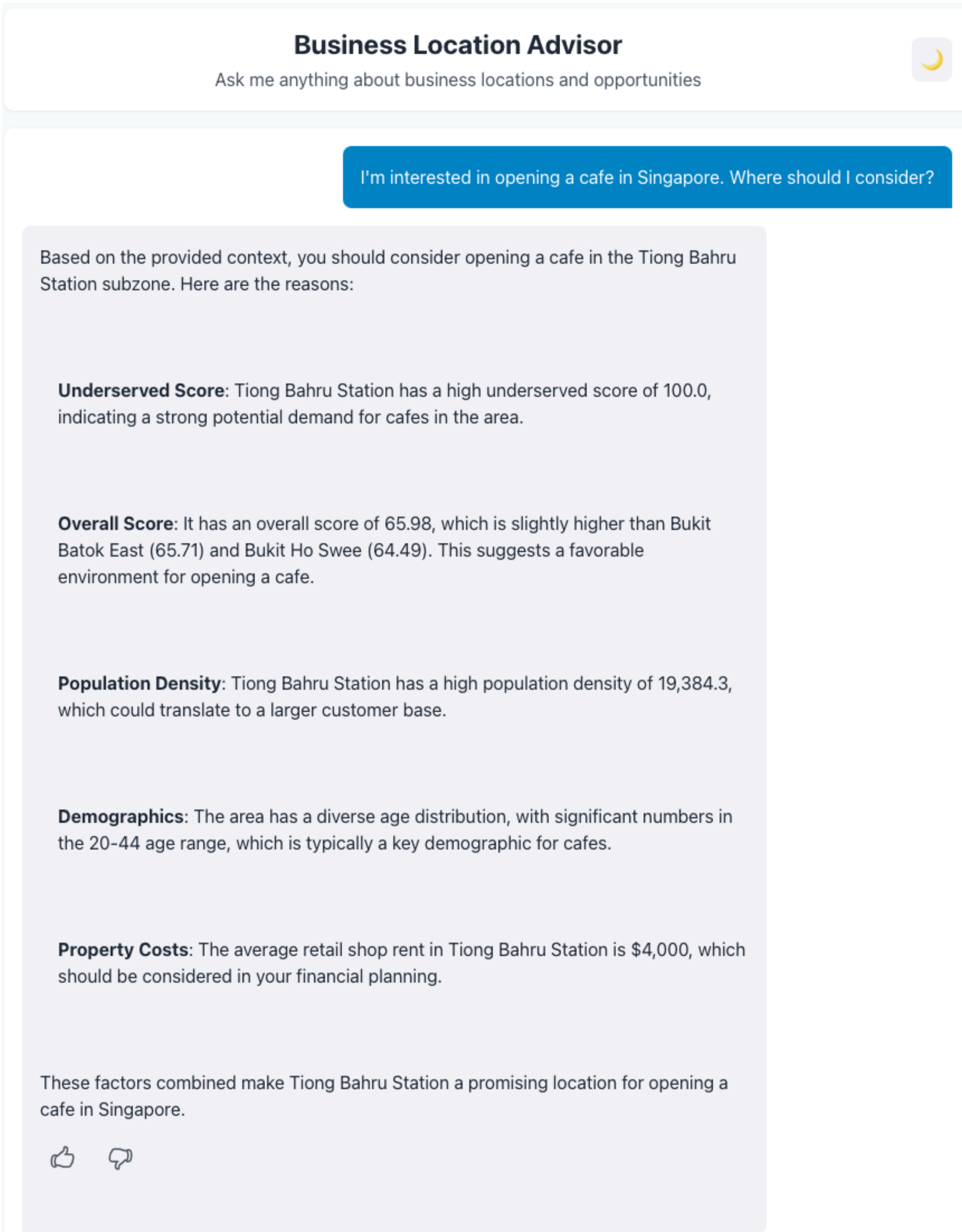
User: "Can you show me the age distribution and housing profile for that area?"

User: "What are the average property prices for commercial spaces there?"

User: "Are there any available properties I can look at?"

User: "Given all this, do you think it's a good idea to open a cafe there? Why or why not?"

Our Chatbot Response:



What is the competitor situation like in the top recommended area?

Based on the context provided, the competitor situation in the Tiong Bahru area for cafes is as follows:

Underserved Score: The Tiong Bahru area has an underserved score of 60.56. This suggests that there is a moderate level of unmet demand for cafes, indicating potential for new entrants to capture market share.

Overall Score: The overall score for cafes in Tiong Bahru is 64.34. This score reflects a combination of factors, including competition, demographics, and the underserved score, suggesting a reasonably favorable environment for opening a cafe.

Competitor Density: The density of competitors is marked as "unknown," which means specific data on the number of existing cafes is not provided. However, the overall and underserved scores imply that while there is some competition, there is still room for new cafes to thrive.

In summary, while there is some competition in Tiong Bahru, the moderate underserved score and overall score suggest that there is still potential for opening a successful cafe in the area.

Can you show me the age distribution and housing profile for that area

Based on the context provided, here is the housing profile for the Tiong Bahru area in the Bukit Merah planning area:

- **Landed Properties**: 10
- **HDB 2-Room Flats**: 320
- **HDB 3-Room Flats**: 2,870
- **HDB 4-Room Flats**: 6,100
- **HDB 5-Room and Executive Apartments**: 2,510
- **Condominiums and Apartments**: 280
- **Other Types of Properties**: 110

Regarding the age distribution, the chat history mentions that Tiong Bahru Station has a diverse age distribution, with significant numbers in the 20-44 age range. This demographic is typically a key target group for cafes, suggesting a potential customer base for your business.

If you need more specific age distribution data, it was not provided in the context. Therefore, this response is based on the general information available from the chat history.

What are the average property prices for commercial spaces there?

Based on the context provided, the average property prices for commercial spaces in the Bukit Merah area, specifically in the subzone of Bukit Merah, are as follows:

Average Retail Shop Sale Price: $2,680,000

Average Retail Shop Rent: $9,900

These figures indicate the general cost of acquiring or renting retail space in the Bukit Merah area. It's important to consider these costs in your financial planning if you're looking to open a cafe or any other business in this area.

Are there any available properties I can look at?

Based on the context provided, there are several available properties in the Bukit Merah area that you can consider for opening a cafe or other commercial ventures. Here are some options:

Retail Shop for Rent:

Listing ID: R4wVKPSkPpc2Amu7H5wLmg

Price: $6,800/month

Area Size: 750 sqft

Description: Rare HDB shop facing the road, located in Bukit Merah Central. Suitable for various trades such as massage, clinic, tuition center, etc.

Listing URL: View Listing

Retail Shop for Sale:

Listing ID: 96WDYDWS2qW5Aqd9muWvYE

Price: $2,680,000

Area Size: 1582 sqft

Description: 2-storey HDB shophouse near Redhill MRT, selling with tenancy, great for investors.

Listing URL: View Listing

General Industrial Space for Rent:

Listing ID: oWaxg4f4eM9EQ8UcFAyfj4

Price: $6,300/month

Area Size: 3154 sqft

Description: Bare condition factory space, near CBD, suitable for production, I.T., storage, and office use.

Listing URL: View Listing

These properties offer a range of options in terms of size, price, and type, allowing you to choose based on your specific needs and budget. If you are interested in any of these listings, you can follow the provided URLs for more details.

Bukit Merah Central

$6,800

** Rare HDB shop HDB facing Road ** * Crowded,...

Bukit Merah Central

$2,680,000

2-STOREY HDB SHOPHOUSE....

Henderson Industrial Park

$6,300

Bare condition factory Bright with lots of windows high...

Given all this, do you think it's a good idea to open a cafe there? Why or why not?

Based on the context provided, opening a cafe in the Tiong Bahru area appears to be a promising idea for several reasons:

Underserved Score: The Tiong Bahru area has an underserved score of 60.56, indicating a moderate level of unmet demand for cafes. This suggests potential for new cafes to capture market share.

Overall Score: The overall score for cafes in Tiong Bahru is 64.34, reflecting a favorable environment when considering factors such as competition, demographics, and the underserved score.

Population Density: The population density in Tiong Bahru is 11,738.5 people per square kilometer, which provides a substantial customer base.

Demographics: The age distribution in Tiong Bahru shows significant numbers in the 20-44 age range, a key demographic for cafes. Specifically, there are 440 people aged 20-24, 780 aged 25-29, 1,100 aged 30-34, 1,310 aged 35-39, and 1,020 aged 40-44.

Housing Profile: The area has a diverse housing profile, with a significant number of HDB 3-room (2,870) and 4-room flats (6,100), which may indicate a stable residential base.

Property Costs: While specific average property prices for commercial spaces in Tiong Bahru were not provided, the chat history mentioned that the average retail shop rent in Tiong Bahru Station is $4,000, which should be factored into financial planning.

In summary, the combination of a moderate underserved score, favorable overall score, high population density, and a key demographic presence makes Tiong Bahru a viable location for opening a cafe. However, it's important to conduct further market research and financial analysis to ensure alignment with your business goals.

ChatGPT's response can be found here:
https://chatgpt.com/share/6810eae4-ea64-800c-88bb-45728e617b19

Below are other conversation scenarios that we have tested for the purpose of evaluating our system. Users are welcome to test out the below conversations directly with our chatbot system.

Conversation 2: Exploring Business Opportunities in a Location

User: "I live in Jurong East. What kind of business should I consider starting here?"
User: "Why do you recommend that business type?"
User: "What is the competition like for that business in Jurong East?"
User: "How about the demographics—does the age distribution support this business?"
User: "What are the property prices and availability for this business type?"

Conversation 3: Comparing Two Locations

User: "I'm deciding between Bedok and Clementi for opening a restaurant. Which is better?"
User: "Show me the competitor scores for both areas."
User: "What about the population and age distribution in each?"
User: "Are property prices significantly different between the two?"
User: "Based on all this, which would you choose and why?"

Conversation 4: Edge Case and Data Absence

User: "Is it a good idea to open a sports complex in Bukit Panjang?"
User: "What if there is no data available for sports complexes in that area?"
User: "Can you suggest an alternative area with better data for sports complexes?"

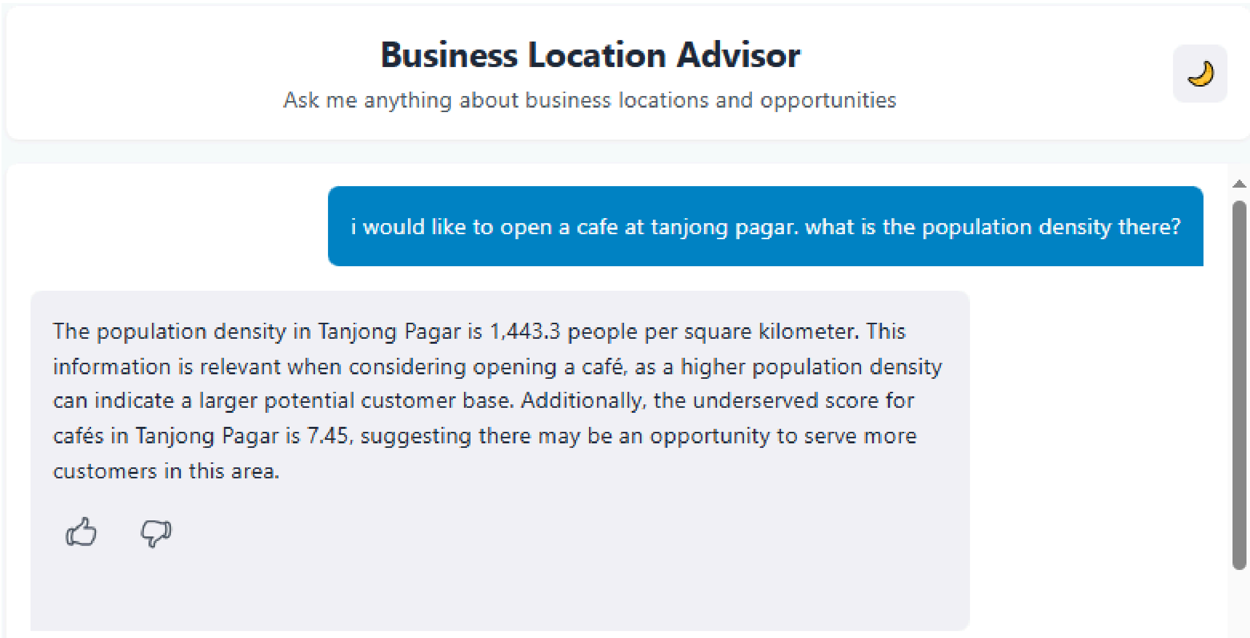
Evaluation

Aspect	Our Chatbot	ChatGPT
Focus	Very focused on business locations and pre-defined factors for analysis.	Generally relevant to business locations and the specific factor(s) prompted, but often ends with additional questions leading to further conversations.

Response Style	Provides only relevant replies focused on the user prompt.	Offers a more conversational experience.
Data Source	Answers are based strictly on the knowledge graph and backend database.	Sometimes pulls data from web sources or relies on LLM training data, leading to many inaccuracies or hallucinations.
Accuracy and Predictability	Provides predictable and accurate responses based on the trained model and data.	Provides responses with varied reply formats and recommendations.
Completeness	Gives full details of the data requested in the prompts.	Selectively replies with top 3 attributes to maintain conversational readability.
User Suitability	Preferred for professional and serious users who require complete and accurate information.	Better suited for general enquiries where readability and casual conversation are prioritised.
Error Handling	Effectively detects and handles exceptions in user prompts, guiding the conversation back to the intended topic.	Handles errors in a conversational manner, but allows irrelevant prompts to derail the discussion and lead it off-topic.

5.2 User Feedback Mechanism

To capture user satisfaction with minimal development overhead, the system's UI includes a thumbs-up/thumbs-down widget on each chatbot response, allowing users to indicate satisfaction in real time. The diagram below illustrates this widget. Due to time constraints, the backend logging and analytics pipeline has not yet been implemented. These components are planned for future work.



6.0 Conclusion

6.1 Summary

In conclusion, our AI-driven business advisor effectively addresses the challenges faced by first-time business owners in Singapore, offering a streamlined and data-driven decision-making process. By building a RAG chatbot upon a knowledge graph, we have created a unified platform that provides suggestions for profitable business locations and viable business types, as well as answers location-specific queries. Our solution not only simplifies the decision-making process but also empowers entrepreneurs with insights that would traditionally require multiple fragmented resources and costly consultations. Moreover, our chatbot demonstrated superior accuracy, completeness, and relevance, particularly for users seeking detailed and data-grounded responses, compared to ChatGPT which often responded conversationally without grounded data. This highlights the practical utility and dependability of our chatbot for serious decision-making contexts.

6.2 Limitations

Regardless, our analysis was constrained by several data-related challenges:

Lack of Footfall data

We originally planned to incorporate footfall data using the BestTime API (https://documentation.besttime.app/#api-reference). However, the free plan significantly restricted the number of requests we could make. Additionally, the BestTime API provides relative footfall percentages rather than absolute visitor counts. 100% represents the peak footfall for a location's own historical data, but percentages are not comparable across different areas. For example, a 90% footfall rating in Location A is not necessarily equal in volume to a 90% footfall in Location B. As such, we decided to exclude footfall in all analyses to prevent inconsistencies in our data.

Incomplete Establishment Data

Business venue data was extracted using the Geoapify Places API. The free tier of Geoapify imposes a limit of up to 500 results per query for each business category within a specified area (e.g., North-West, North-East). This restriction meant that only a subset of available businesses could be captured. For example, there could have been 1000 cafes in the North-Western quadrant of Singapore, but maybe we only extracted 500 cafes in that area. Therefore, the completeness and representativeness of competitor data were constrained, which could affect the accuracy of analyses and machine learning (ML) outputs relying on this dataset.

Since the ML models and analysis are trained and queried based on the available data in Supabase, any missing or incomplete data may cause effects such as skewed competitor density estimates and incorrect underserved area detection.

Lack of Domain Knowledge

The base weight and venue-type multipliers in our fuzzy-logic scoring were set using our common knowledge. Consequently, these parameters may not accurately reflect real market dynamics.

Lack of Outcome Metrics

Due to the lack of hard labels such as revenue, profit-margin or survival-rates, we cannot distinguish between heavily trafficked but unprofitable locations and truly succeeding businesses. The competitor counts and demographic density show only the exposure. In the absence of real-world customer counts, the model cannot easily determine features of a successful business.

Besides, we cannot systematically evaluate model depth or validate which configurations produce the best predictive performance due to lack of labeled outcomes.

6.3 Recommendations

Building on the limitations discussed above, we outline several directions for future development and system improvement.

Firstly, incorporating real-time data, such as live footfall, business openings and closures, and active or inactive property listings, could significantly improve the system’s relevance. Although our current approach excluded footfall due to API constraints and relative measurement issues, sourcing alternative footfall providers or leveraging anonymised telco data may help capture visitor volume more reliably across planning area subzones.

Next, to address the incompleteness of establishment data, we could consider integrating multiple sources beyond Geoapify or partnering with data providers that offer full-coverage commercial venue datasets. This would ensure a more representative baseline for competitor analysis, including the computation of our underserved and location suitability scores. Similarly, as our current system lacks labeled outcome metrics like revenue or profit, establishing partnerships with industry stakeholders may help us derive proxy success indicators and validate the model’s performance more rigorously.

Recognising the absence of domain-calibrated rules, we also recommend engaging with domain experts, such as location analysts or SME consultants, to fine-tune the base weights and fuzzy-logic multipliers in our scoring engine. Such collaborations could ensure that our scoring logic better reflects real-world business viability. In addition, we plan to incorporate regional rule-based logic that adjusts scoring thresholds based on characteristics unique to each planning area or subzone. For instance, optimal business types and acceptable competitor densities may differ significantly between high-density commercial areas and quieter residential neighbourhoods. Encoding such distinctions into our analytics engine would enable more context-sensitive and locally grounded suggestions.

Another crucial enhancement would be the integration of Singapore zoning regulations which govern land-use designations, such as residential, commercial, and open space. Factoring zoning constraints into our location filtering and scoring processes would prevent inappropriate suggestions, like recommending a retail outlet in a purely residential area, and thereby improve recommendation feasibility.

Lastly, we aim to implement a feedback analytics system that captures each thumbs-up/thumbs-down event alongside associated metadata, such as message ID, session and user identifiers, and intent classification. This feedback will then be sent to an analytics pipeline that continuously monitors overall satisfaction rates and per-intent distributions. If a rating falls below a predefined threshold, the corresponding interactions will be reviewed to identify gaps in context, reasoning, or relevance. Insights from this review will drive prompt-template tweaks, fuzzy-logic recalibrations, and other targeted updates, ensuring that every piece of feedback fuels ongoing improvements in recommendation accuracy, relevance, and clarity.

7.0 Appendix

7.1 Appendix A: Project Proposal

Intelligent Singapore Business Location Advisor System

ISY5001-2025 (15.03.25)

Group 3
CHUA HIENG WEIH (A0315386Y) - Group Leader
GU HAIXIANG (A0131920U)
LIZABETH ANNABEL TUKIMAN (A0315378X)
MUHAMMAD HARUN BIN ABDUL RASHID (A0164598L)
SRITAM PATNAIK (A0115530W)

Introduction

We propose to build an intelligent Singapore business location advisor that incorporates reasoning with a large language model (LLM) and data-driven optimization to recommend the most profitable locations for businesses, namely retail and food and beverage (F&B) businesses.

Finding the right business location is a complex decision that involves evaluating commercial property availability, conducting demographic research, and analysing competitor dynamics (Finding the Right Business Location in Singapore | IntraConnect).

Our system thus aims to simplify this process by integrating LLMs with structured knowledge to provide businesses with data-backed location recommendations, automating location scouting.

For our project, we aim to accomplish 3 main goals:

1. Build a conversational AI chatbot using retrieval-augmented generation (RAG)
2. Enhance RAG reasoning using a knowledge graph that encodes relationships between business types, locations, customer demographics, and competitor presence to improve recommendation accuracy

3. Develop an optimization module for location recommendations based on constraints such as foot traffic and competition

Background / Market Context

Choosing the right business location is crucial for success, influencing factors such as customer accessibility, brand visibility, and operational efficiency. Poor location decisions can lead to significant financial losses and operational challenges (Seven reasons why location is important | JLL and Why Is Location Important For A Business In Singapore? | Singapore Company Formation). However, selecting an optimal business location is an often complex and time-consuming process, requiring analysis of foot traffic, competitor presence, and consumer demographics.

Despite the availability of real estate platforms and open data sources, there is no unified platform that consolidates property listings, competitor insights, and demographic trends into a single decision-making tool. Business owners currently rely on government-backed small and medium enterprises (SME) consultations and third-party advisory services, which require significant manual effort, field research, or costly expert guidance. While solutions like BestTime.app provide crowd analytics and foot traffic predictions, and SME Centres offer general business advice, they do not provide a dedicated AI-driven approach to business site selection. Additionally, competitors like BestPlace seem to cater towards the United States and Europe, with limited relevance to Singapore.

To address this gap, our project leverages machine reasoning and cognitive systems to transform business location selection from a manual, search-based approach into an intelligent, data-driven recommendation process. By integrating foot traffic, competitor, and demographics analyses, we aim to provide business owners with an accessible, cost-effective, and efficient business location selection tool tailored to Singapore's unique landscape.

Market Research

In 2024, 6322 business entities were formed in the retail trade industry ,and 3793 were formed in the F&B services sector ((DOS) | SingStat Table Builder – Formation Of All Business Entities By Industry; total 71639 businesses formed in 2024). However, in the same period, 6147 retail businesses and 3047 F&B businesses ceased operations, suggesting significant challenges in sustainability and profitability ((DOS) | SingStat Table Builder – Cessation Of All Business Entities By Industry; total 55491 businesses ceased operations in 2024). One potential contributing factor is suboptimal business location decisions, as seen from how retail stores like Marks & Spencer close due to rising rental costs and declining foot traffic (Several Parkway Parade mall tenants leave or downsize amid rising rents | The Straits Times). As digital transformation reshapes consumer behaviour and as more businesses explore hybrid models combining physical storefronts with online operations, the need for data-driven location intelligence is increasing. Business owners can benefit from AI-powered decision tools that evaluate potential locations and recommend them a profitable one. The relevance of AI-powered tools in business site selection can also be seen from the existence of research in leveraging deep learning for site selection (Site Selection via Learning Graph Convolutional Neural Networks: A Case Study of Singapore).

The market for business site selection in Singapore is currently served by government-backed advisory services, independent research, and business consultants. For instance, SME Centres offer complimentary consultations to business owners, but their support is more focused on overall business growth rather than providing specialised, data-driven location selection insights. Property platforms like 99.co and CommercialGuru provide insights into available spaces but do not analyse competitor presence, foot traffic, or demographic suitability, while platforms like BestTime.app offer crowd-level insights but do not specialise in helping businesses choose optimal locations. Other available resources include interactive real estate maps that provide information on commercial spaces, such as this map published by the Singapore Economic Development Board (EDB), but they lack AI-driven recommendations for profitability and long-term success.

A competitor, xMap, provides advanced geospatial analytics that allows users to select sites with maximum returns and optimal operations. However, while its free plan includes relevant information such as demographics and competition data, traffic data is only accessible through its pro plan with a monthly cost of almost $500. Another global competitor, BestPlace, provides AI-powered location intelligence, including end-to-end visibility of local customer behavioural patterns, to businesses (Finding the BestPlace: Meet an AI Solution that Knows Your Next Business Location | by Kate Saenko | Toloka | Medium), but its services seem to be mainly available in the United States and Europe, with limited coverage in Singapore and the broader Asia-Pacific region. This presents a market gap for a cost-effective AI-powered business location advisor tailored to Singapore's commercial landscape that integrates multiple data sources into a single recommendation engine to guide business site selection.

By streamlining the site selection process, our solution helps reduce the risk of poor location choices, making it a valuable resource for first-time entrepreneurs and existing businesses looking to relocate. With the increasing adoption of AI in Singapore's business landscape, there is strong potential for our project to help small retail and F&B businesses make data-driven location decisions.

Project Scope

Our project focuses on the 3 aspects of intelligent reasoning systems listed below:

1. Decision automation using knowledge-based reasoning techniques over a knowledge graph
2. Knowledge discovery and data mining for recommendation
3. Incorporation of a cognitive tool through a chatbot

Decision Automation: Knowledge-based Reasoning

Our system involves reasoning over knowledge graphs using an LLM. Following the approach outlined in the paper GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning by Mavromatis and Karypis, a graph neural network (GNN) will be used to reason over the knowledge graph to retrieve answers to a prompt. More information about our knowledge graph and how we plan for GNN-RAG to be incorporated into our system can be found in the System Design section.

Knowledge Discovery and Data Mining: Recommendation

Our system leverages knowledge discovery and data mining techniques to analyse large-scale datasets on competitor density, and demographic trends, allowing it to identify key factors influencing profitability. Through the use of LLM embeddings and reasoning, it helps businesses select optimal locations that maximise profit while mitigating risks.

Cognitive Interface: Chatbot

Our solution involves the user prompting the chatbot for a recommendation for their business location, such as “Recommend me a place to open my cafe.” The bot will make a suitable district recommendation based on the available data related to foot traffic, demographics, competitors, etc., as outlined in the subsequent Data Collection and Preparation section. If a particular property is found, it will be recommended instead. More details on the system design can be found in the System Design section.

Data Collection and Preparation

This section describes and explains what data we collected, how we store and process them, and how they contribute to this project’s overall objective. We will mainly be using postman and python scripts to scrape as well as using publicly available datasets such as data.gov.sg.

We will use the district codes in Singapore to make it easier to query for footfall data, which will be further explained below, and to provide a form of categorisation or clustering in terms of location. Below are the district codes taken from the Urban Redevelopment Authority (URA):

List of Postal Districts

Postal District	Postal Sector (1st 2 digits of 6-digit postal codes)	General Location
01	01, 02, 03, 04, 05, 06	Raffles Place, Cecil, Marina, People's Park
02	07, 08	Anson, Tanjong Pagar
03	14, 15, 16	Queenstown, Tiong Bahru
04	09, 10	Telok Blangah, Harbourfront
05	11, 12, 13	Pasir Panjang, Hong Leong Garden, Clementi New Town
06	17	High Street, Beach Road (part)
07	18, 19	Middle Road, Golden Mile
08	20, 21	Little India
09	22, 23	Orchard, Cairnhill, River Valley
10	24, 25, 26, 27	Ardmore, Bukit Timah, Holland Road, Tanglin
11	28, 29, 30	Watten Estate, Novena, Thomson
12	31, 32, 33	Balestier, Toa Payoh, Serangoon
13	34, 35, 36, 37	Macpherson, Braddell
14	38, 39, 40, 41	Geylang, Eunos
15	42, 43, 44, 45	Katong, Joo Chiat, Amber Road
16	46, 47, 48	Bedok, Upper East Coast, Eastwood, Kew Drive
17	49, 50, 81	Loyang, Changi
18	51, 52	Tampines, Pasir Ris
19	53, 54, 55, 82	Serangoon Garden, Hougang, Punggol
20	56, 57	Bishan, Ang Mo Kio
21	58, 59	Upper Bukit Timah, Clementi Park, Ulu Pandan
22	60, 61, 62, 63, 64	Jurong
23	65, 66, 67, 68	Hillview, Dairy Farm, Bukit Panjang, Choa Chu Kang
24	69, 70, 71	Lim Chu Kang, Tengah
25	72, 73	Kranji, Woodgrove
26	77, 78	Upper Thomson, Springleaf
27	75, 76	Yishun, Sembawang
28	79, 80	Seletar

Source: SingPost

As far as possible, we will scrape and store data in our own database hosted on Supabase, rather than querying external APIs to retrieve such data. This will reduce the processing time for each user prompt. The following subsections below further elaborates on the data which we will be extracting.

Real Estate Listings

We will extract property listings from property sites such as 99.co and the Housing Development Board (HDB) portal and store them in Supabase. Our real estate data consists of two major categories:

- Residential Properties: These are HDB flats, condominiums, and landed properties.

- Industrial Properties: Commercial spaces like offices, factories, and F&B outlets.

High-density residential properties may make it suitable for businesses such as clinics, retail stores, and childcare centres to be set up while industrial areas are suitable for places such as warehouses, manufacturing and logistics.

MRT Information

Data on MRT exit locations as well as number of MRT and LRT stations can easily be found in data.gov.sg. One such dataset can be found in this link: http://data.gov.sg/collections/367/view.

Business locations that are near these stations should be more accessible to consumers and, therefore, be a significant factor to consider in setting up a new retail business.

Amenities

Other key amenities such as schools, preschools, hawker centres, parks, and community clubs. This information also contributes to how suitable a business location is and knowing potential competitors.

Footfall Data

We identified a gap in publicly available footfall data in Singapore, but we found a web application called besttime.app which provides such data globally, including Singapore and on different days and hours too. This data can be served through their chatbot interface. This can be integrated into our project to further enhance our processing of prompts. They have APIs to provide footfall for specific locations at specific times of day or week, which we can then extract per district and store it in our own database to speed up processing instead of calling their API each time a user prompts something.

Below is an example of how we can use the chatbot to return data in a format which we can easily use. The district list from URA was given to the bestTime chatbot and was instructed to sort the footfall from highest to lowest and return in a JSON format. Below is a part of the full response from it:

```
{
  "district": "01",
  "areas": ["Raffles Place", "Marina", "People's Park"],
  "description": "Central Business District, Marina Bay Sands, Chinatown"
},
{
  "district": "09",
  "areas": ["Orchard", "Cairnhill", "River Valley"],
  "description": "Orchard Road shopping belt"
},
{
  "district": "07",
  "areas": ["Middle Road", "Golden Mile"],
  "description": "Bugis, Rochor, Kampong Glam, Suntec"
},
{
  "district": "10",
  "areas": ["Holland Road", "Tanglin", "Bukit Timah"],
  "description": "Expat and nightlife hotspots"
},
{
```


In addition to general footfall data, we are also able to query it to give competitor data in a specific location such as address, opening hours and what its peak periods are. We prompted the chatbot to give a list of food and beverage outlets in District 1 of Singapore and below is a part of its response:

```json
{
  "status": "success",
  "venues": [
    {
      "name": "Jigger & Pony",
      "type": "BAR",
      "rating": 4.6,
      "reviews": 1200,
      "price_level": 4,
      "peak_hours": [
        { "hour": 21, "intensity": "🔥🔥🔥" }
      ]
    },
    {
      "name": "LAVO Italian Restaurant & Rooftop Bar",
      "type": "RESTAURANT",
      "rating": 4.5,
      "reviews": 1500,
      "price_level": 5,
      "peak_hours": [
        { "hour": 20, "intensity": "🔥🔥" }
      ]
    },
    {
      "name": "Maxwell Food Centre",
      "type": "FOOD_AND_DRINK",
      "rating": 4.4,
      "reviews": 3200,
      "price_level": 2,
      "peak_hours": [
        { "hour": 12, "intensity": "🔥🔥🔥" }
```


We can further tailor the response structure based on what we need.

Demographic Data

The Department of Statistics Singapore provides several datasets related to the Singapore population. Some examples includes the distribution of employed people working in different areas of Singapore, as shown below:

GENERAL HOUSEHOLD SURVEY 2015

Table 139 Resident Population Aged 15 Years and Over by Planning Area, Economic Status and Sex

- continued

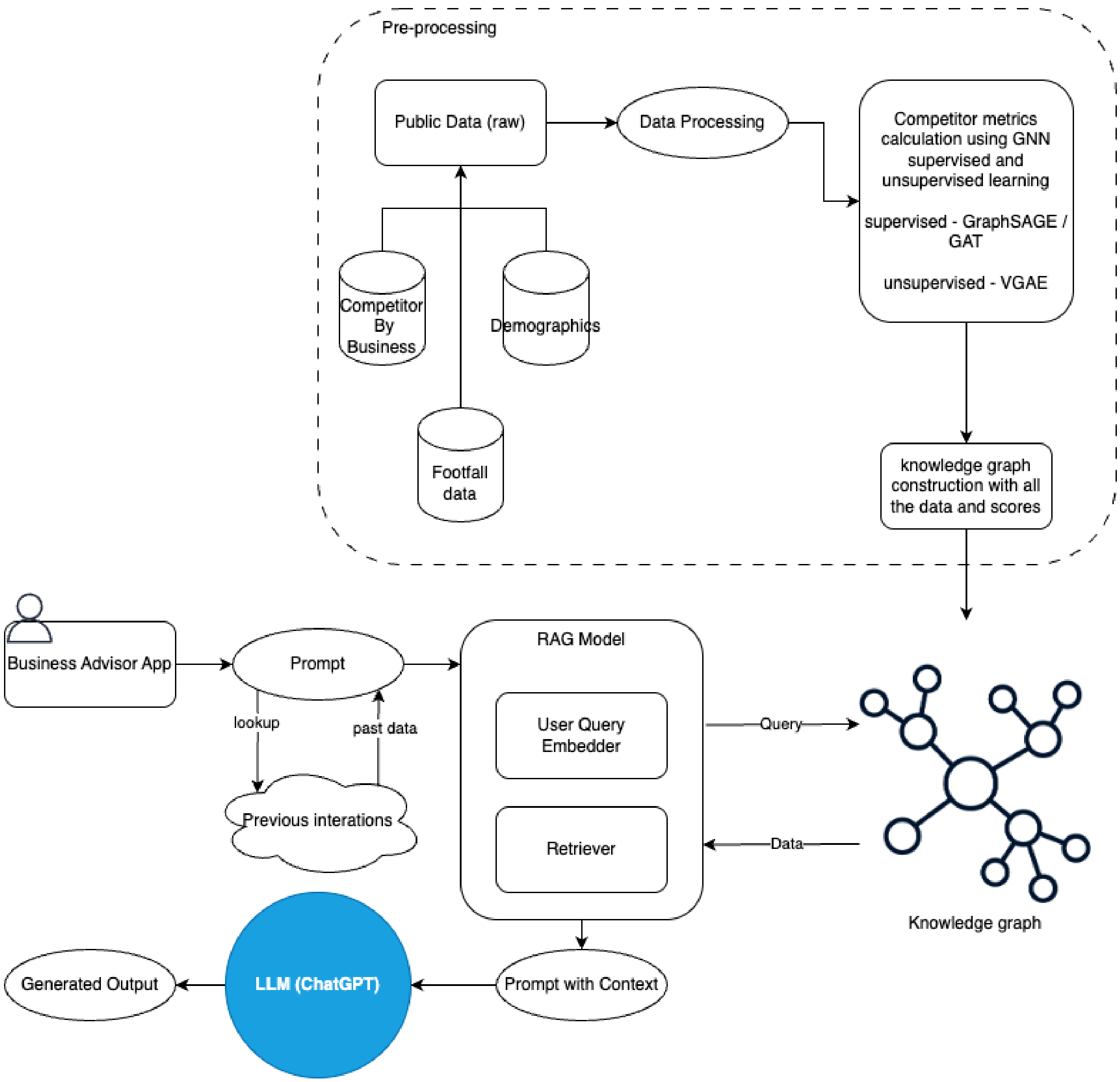
Thousands

Planning Area	Total			Economically Active									Economically Inactive		
				Total			Working			Unemployed					
	Total	Males	Females	Total	Males	Females	Total	Males	Females	Total	Males	Females	Total	Males	Females
Serangoon	109.4	51.4	58.0	70.7	37.1	33.6	68.2	35.7	32.6	2.5	1.5	1.0	38.7	14.2	24.4
Tampines	231.7	113.9	117.8	159.5	89.2	70.3	152.8	86.2	66.6	6.7	3.0	3.7	72.2	24.7	47.5
Tanglin	14.4	6.5	7.9	9.1	5.0	4.0	9.0	5.0	4.0	-	-	-	5.3	1.5	3.8
Toa Payoh	107.9	51.3	56.6	71.8	37.3	34.4	68.9	35.8	33.1	2.9	1.6	1.3	36.1	14.0	22.1
Woodlands	201.9	99.0	102.9	140.0	77.4	62.6	134.8	74.6	60.2	5.3	2.8	2.5	61.8	21.6	40.3
Yishun	172.3	83.9	88.3	123.1	65.8	57.4	119.3	63.6	55.7	3.8	2.2	1.7	49.1	18.2	31.0
Others	31.0	15.7	15.3	22.3	13.0	9.3	21.5	12.6	8.9	0.8	0.4	0.5	8.7	2.7	6.0

Note: Planning areas refer to areas demarcated in the Urban Redevelopment Authority's Master Plan 2014.

This can further enhance the accuracy of the output response as it could give a better picture to users as to which location is the more profitable one.

System Design



Proposed System Architecture

This is a **two-stage** architecture—(1) GNN-based learning and knowledge graph construction, and (2) RAG-based contextual retrieval and generation.

1. Data Ingestion & Preprocessing

   a. Public & Internal Data: We gather demographic information (e.g., population, income), foot traffic metrics, and competitor data (location, business type).
   b. Data Cleaning & Standardization: We convert raw inputs into consistent schemas, resolving missing or anomalous entries and ensuring data quality.

2. Competition Metrics Calculation (GNN)

   a. Supervised GNN (GraphSAGE/GAT):
      i. We propose using GraphSAGE (or GAT) to learn competitor intensity by encoding both active business types and location-based relationships. Each business (node) is initialized with feature vectors (such as one-hot–encoded business type). GraphSAGE iteratively aggregates neighbor features (e.g., competitor businesses within a given radius) to update each node's embedding, capturing the local competitive environment. This information guides the model to determine clusters with high competitor density for different business activities, ultimately yielding a supervised competitor intensity score.

   b. Unsupervised GNN (Variational Graph Autoencoder - VGAE):
      i. We propose VGAE to detect clusters or anomalies, especially useful in the absence of explicit success/failure data. This unsupervised approach helps uncover hidden risks or location-based patterns without relying on labeled outcomes. E.g. It could be a specific location that has a high rate of de-registration for certain business types.

   c. Output: A "competitor influence score" for each location and business type, reflecting competitive pressure and potential risks.

3. Knowledge Graph Construction
   a. Nodes: We represent each location (towns) as nodes.

   b. Edges: We model relationships such as demographic details, foot traffics, competitor scores for various business activities and other supporting data belonging to the locations.

   c. Justification: A graph-based structure is flexible for integrating heterogeneous data (demographics, foot traffic) rather than traditional table databases.

4. Retrieval-Augmented Generation (RAG) Model
   a. User Query Embedder & Retriever: Converts user queries into structured or embedded forms, then retrieves relevant data (e.g., locations, business types, demographics, competitor scores) from the knowledge graph.

   b. Contextual Prompt: Assembles retrieved information into a prompt for the LLM, ensuring targeted and accurate context. Combining a knowledge graph retriever with an LLM ensures responses are data-grounded rather than speculative.

5. LLM Integration (e.g., ChatGPT / DeepSeek)
   a. Response Generation: Uses domain-specific context to produce data-driven recommendations, minimizing hallucinations by grounding outputs in the knowledge graph.

   b. Explainable Output: Provides reasoned explanations (e.g., “Area X is recommended due to moderate foot traffic and low competitor density.”).

6. Business Advisor App
   a. User Chatbot Interface: Enables business owners to ask questions such as “Where should I open a new cafe?”

   b. Continuous Improvement: Logs interactions for query refinements, ensuring past interactions can be incorporated over time.

   c. UI: Presents location suggestions and features (demographic, foot traffics, and competitor insights) in an intuitive dashboard or chat interface.

Conclusion

In conclusion, our market research highlights significant challenges facing Singapore's retail and F&B sectors, notably the adverse impact of poor location decisions. By leveraging comprehensive data sources—including property listings, demographic profiles, footfall metrics, and competitor analysis—our project addresses these challenges through an intelligent, data-centric approach. Our solution integrates advanced cognitive systems, combining knowledge graph techniques, GNN-based competitor modeling, and retrieval-augmented generation (RAG), to deliver precise and contextually relevant recommendations. This approach not only meets the project objectives but also provides a practical, innovative tool to enhance business site selection strategies.

7.2 Appendix B: Mapped System Functionalities Against Courses

Course	System Functionality
Machine Reasoning	**Knowledge Extraction** • Access of government open data • Web scraping from property websites to get property listing data • Use of APIs (like Geoapify Places API) to get competitor data **Knowledge Representation** • Construction of knowledge graph using Neo4j, from table data in Supabase **Knowledge Reasoning and Inference** • Graph querying and reasoning using Cypher • Fuzzy logic and GNN to compute overall “location suitability” score • Geospatial density analysis to compute “underserved” score
Reasoning Systems	**Graph-Enabled RAG** • Knowledge graph construction and reasoning using Neo4j and Cypher • RAG incorporation using LangChain
Cognitive Systems	**Cognitive System** • Use of LLM to classify user intent, query the knowledge graph, and answer the user’s queries based on retrieved knowledge • Prompt engineering

7.3 Appendix C: Installation and User Guide

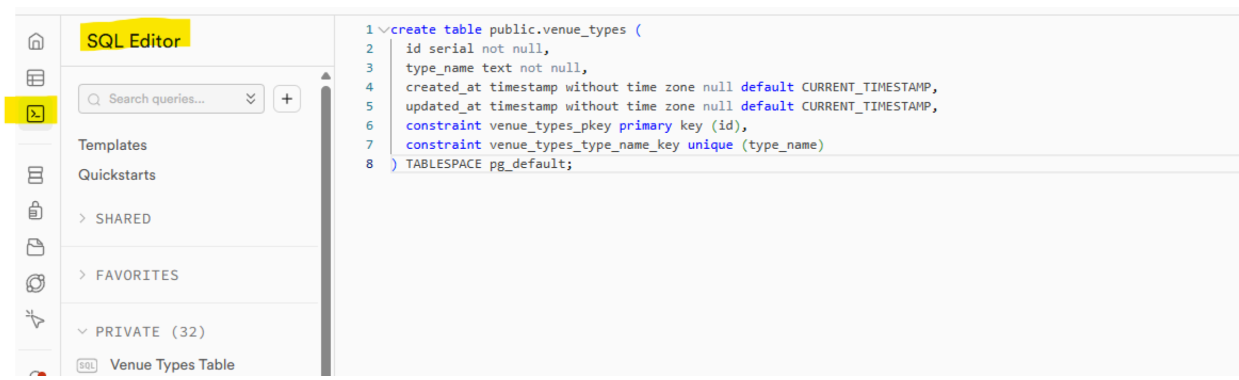
This section explains how to run this project locally. The first step is to clone from the GitHub repository: https://github.com/johnsonweih/biz-location-advisor-system.git. Subsequently, there are multiple services to set up and run and will be described in the subsections below.

7.3.1 Installation Guide

7.3.1.1 Supabase

Follow the steps below to set up a Supabase project with data.

1. Sign in or create a Supabase account at (https://supabase.com/dashboard/sign-in).
2. Create a new project in Supabase and head to the SQL Editor, which can be found on the leftside menu.



3. Run the following query to enable the PostGIS extension:

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

4. Inside the GitHub repository, navigate to the backend supabase_setup directory, which contains `table_creations` and `data` folders:

```
cd SystemCode/backend/supabase_setup
```

5. Run each script inside the `table_creations` folder in ascending order based on the prefixed number in their filenames. Based on the screenshot below:

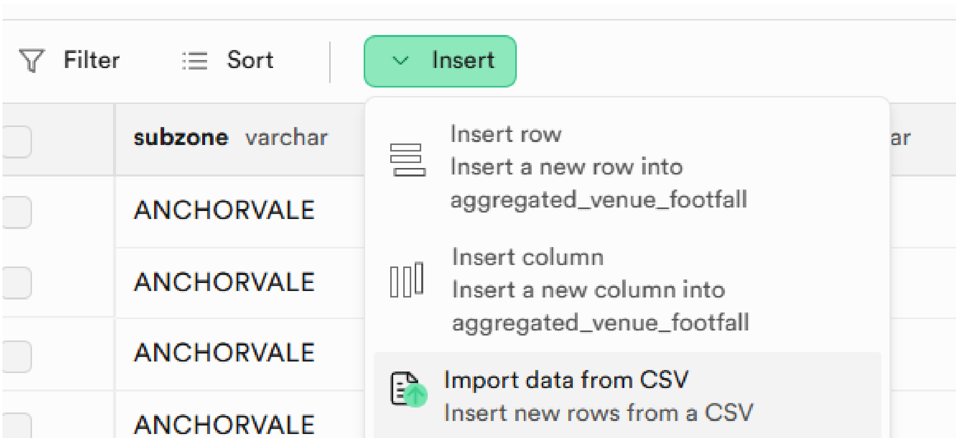
- 1_venue_types.sql
- 2_planning_areas.sql
- 3_industrial_properties.sql
- 4_establishments.sql
- 5_demographics_age_group.sql
- 6_demographics_housing_types.sql
- 7_demographics_population.sql
- 8_avg_industrial_prices.sql
- 9_competitor_stats.sql

Run `1_venue_types.sql`, followed by `2_planning_areas.sql`, and so on, by copying and pasting the queries in the SQL Editor and then clicking "Run". The corresponding table names will then be created in your own instance of Supabase, under the "public" schema.

6. In the `data` subfolder, we extracted the current state of data from each table in our Supabase instance. Below is a screenshot of all file names:

- avg_industrial_prices.csv
- competitor_stats.csv
- demographics_age_group.csv
- demographics_housing_types.csv
- demographics_population.csv
- establishments.csv
- industrial_properties.csv
- planning_areas.csv
- venue_types.csv

Head to Supabase's Table Editor, also found on the leftside menu, and in each table, click on insert, and select "Import data from CSV":

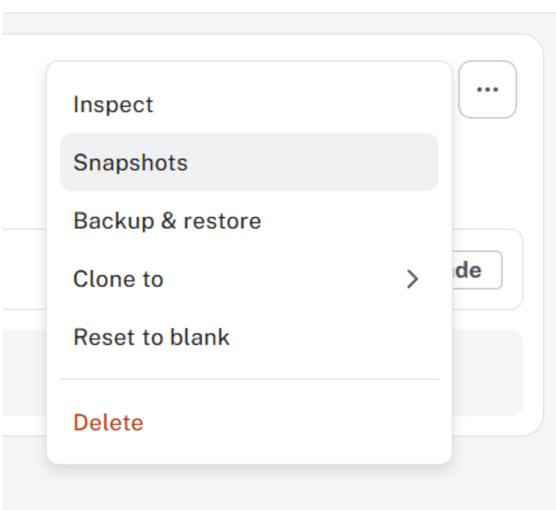


Import each csv file into their corresponding tables.

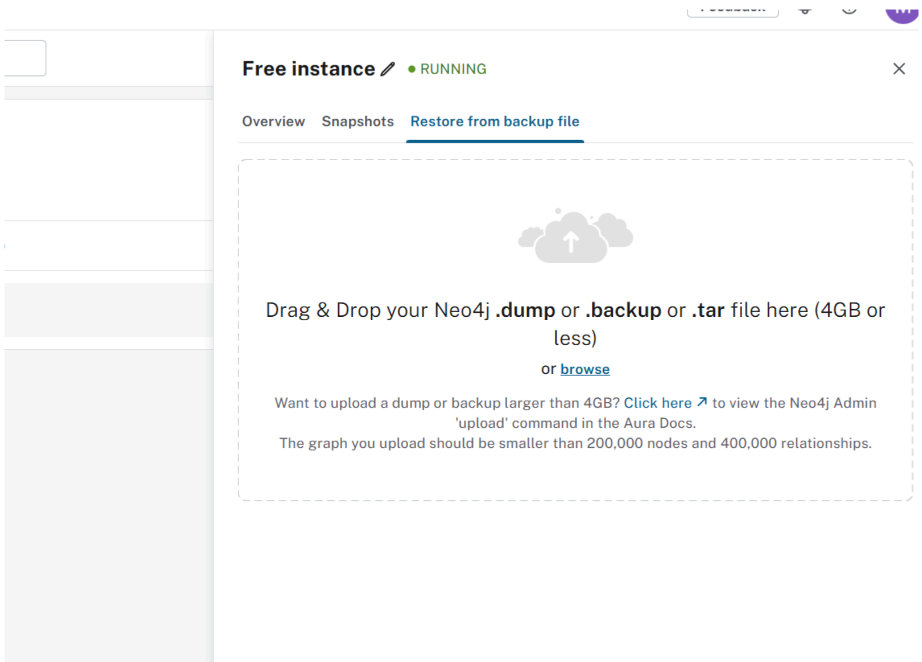
7.3.1.2 Neo4j

Neo4j powers the knowledge graphs, which allows structured and semantic querying of relationships between subzones , business types, demographic metrics, and more.

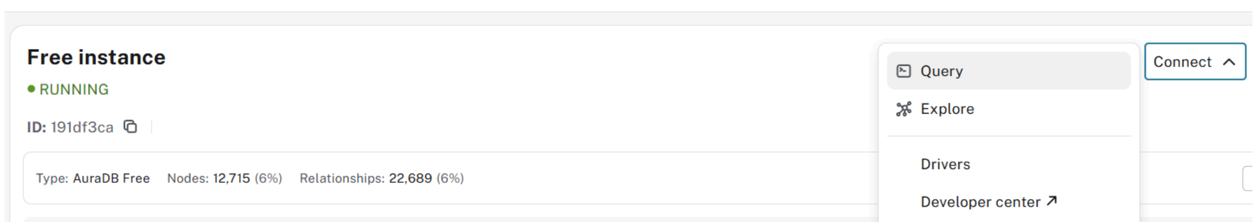
1. To set up, sign in or create an account at Neo4j Aura (https://console-preview.neo4j.io/).
2. Once logged in, create a new instance or use an existing instance.
3. Click the 3 dots by the instance and go to "Snapshots".



4. Upload the `neo4j_snapshot.backup` file which is in the `neo4j_setup` subfolder in the backend folder of the repository.



5. Once imported, the knowledge graph is ready to be viewed and queried. Click on "Connect" and then "Query" to connect to the uploaded knowledge graph snapshot.



7.3.1.3 Backend

The backend is a Python Flask server that integrates with OpenAI's GPT-4o API to provide intelligent responses.

Steps:

1. Open a terminal and navigate to the backend server directory: `cd SystemCode/backend/server`

2. Create and activate a virtual environment using the commands below:
    a. Run: `python -m venv venv`
    b. On Mac: `source venv/bin/activate`
    c. On Windows: `venv\Scripts\activate`
3. Install the required Python packages: `pip install -r requirements.txt`
4. Create a `.env` file in the same directory and add your OpenAI API key as well as the other credentials which we will need for the other platforms:
    a. OPENAI_API_KEY=<your_openai_api_key_here>
    b. NEO4J_URI=neo4j+s:<your_neo4j_uri_here>
    c. NEO4J_USERNAME=neo4j
    d. NEO4J_PASSWORD=<your_neo4j_pwd_here>
    e. SUPABASE_URL=<your_supabase_url_here>
    f. SUPABASE_KEY=<your_supabase_key_here>
5. Run the Flask server: `python app.py`
6. By default, the server will be available at http://localhost:4000.

7.3.1.4 Frontend

The frontend is a React-based web application built with Create React App.

Steps:

1. Open a terminal and navigate to the frontend directory: `cd SystemCode/frontend`

2. Install project dependencies: `npm install`

3. Start the development server: `npm start`

The application would then be available at http://localhost:3000

7.3.2 User Guide

After setting up the system, head to http://localhost:3000 to chat with our chatbot.

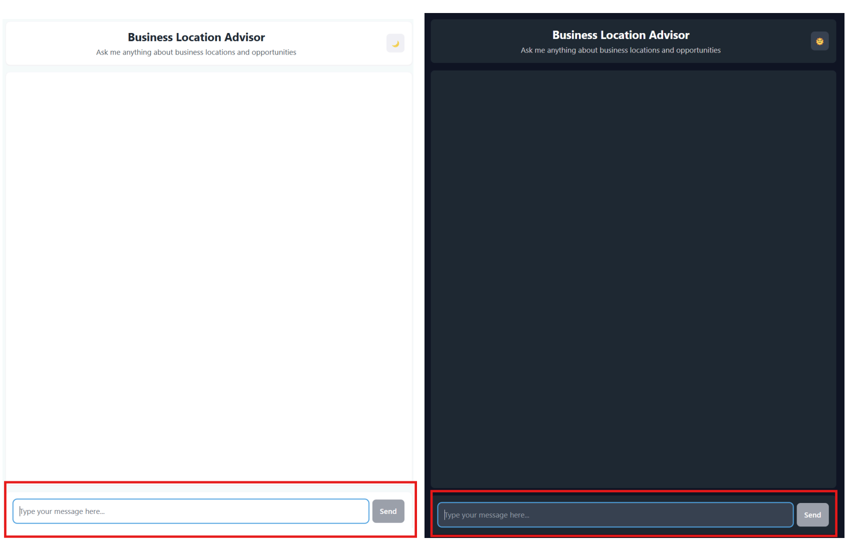
Our chatbot supports the following query types:

- general advice given a business type and planning area / subzone
    - e.g., “I would like to open a cafe at Katong. What do you think about it?”

- location suggestion given a business type
    - e.g., “Where should I open a cafe in Singapore?”
- business type suggestion given a planning area / subzone
    - e.g., “What type of business would you recommend me to start in Katong?”
- demographics information, such as population statistics, age distribution, and housing profile, in a given planning area / subzone
    - e.g., “How is the demographics in Katong?”, “Tell me more about Katong’s age distribution.”
- competitor information, such as overall score, underserved score, and competitor examples, for a given business type and/or planning area / subzone
    - e.g., “How is the competition for cafes like in Katong?”, “How is the overall competition like in Katong?”, “How is the competition for cafes like across the whole of Singapore?”, “I would like to open a cafe at Katong. Who are some competitors?”
- properties information, including average property prices for different venue types for both rent and sale, and available properties in a given planning area / subzone
    - e.g., “What is the average property prices for retail shops in Katong?”, “What are some properties I can consider if I want to open a cafe at Katong?”

Chatting with the Chatbot

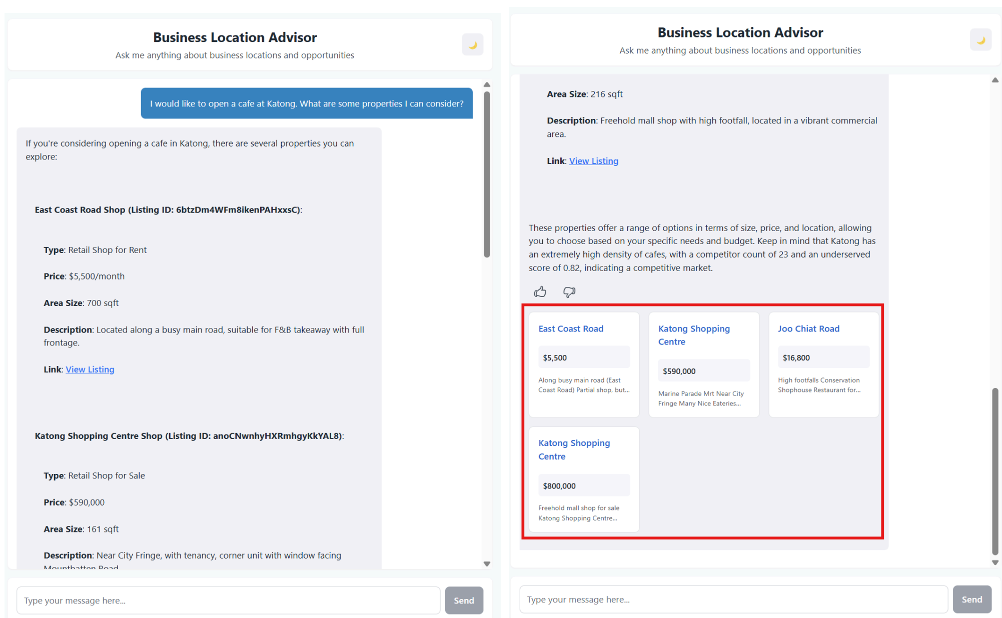
1. At http://localhost:3000, you should see the following UI. Both light and dark modes are available.



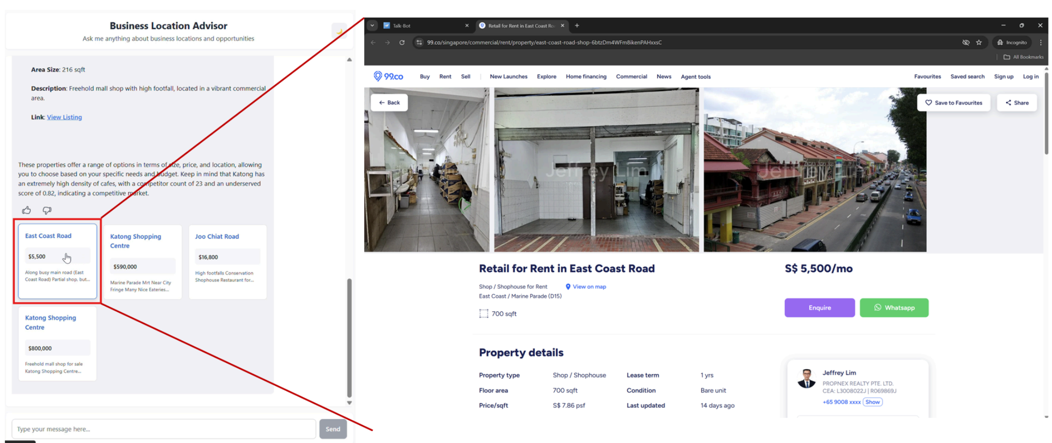
2. Type your question in the message box at the bottom of the chat, as outlined by the red box in the figures above, and press “Enter” or click the “Send” button.

Viewing Available Properties

1. For queries related to available properties in a given planning area, the property listings will additionally be embedded as cards in the generated response, as seen in the red box below.

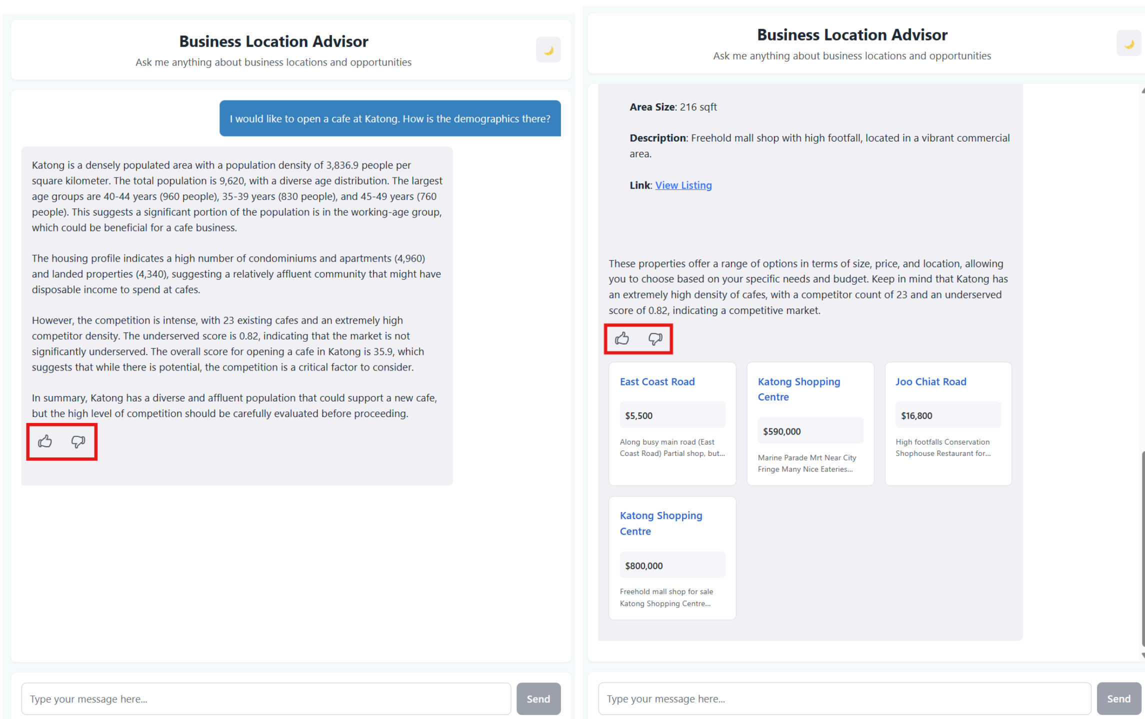


2. Click on the property card of your interest to learn more about the property and view its corresponding listing on 99.co.



Giving Feedback

To rate the chatbot responses, click on either the thumbs-up or thumbs-down icon at the end of the response.



7.4 Appendix D: References (All)

BestPlace. (n.d.). *Consumer Packaged Goods*. https://bestplace.ai/en/

Cody. (n.d.). *Business Consultant AI*. https://meetcody.ai/use-cases/business-consultant-ai/

DBS. (2025, February 3). *DBS Survey: SMEs embracing Gen AI, upskilling and global expansion to stay competitive*. https://www.dbs.com/newsroom/DBS_Survey_SMEs_embracing_Gen_AI_upskilling_and_global_expansion_to_stay_competitive

Enterprise Singapore. (n.d.). *SME Centres*. https://www.enterprisesg.gov.sg/about-us/contact-us/sme-centres

Infocomm Media Development Authority (IMDA). (2024). *Singapore Digital Economy Report*. https://www.imda.gov.sg/-/media/imda/files/infocomm-media-landscape/research-and-statistics/sgde-report/singapore-digital-economy-report-2024.pdf

Kau, A., He, X., Nambissan, A., Astudillo, A., Yin, H., & Aryani, A. (2024). Combining knowledge graphs and large language models. https://doi.org/10.48550/arXiv.2407.0 6564

Lim, J., & Sheo, C.T. (2025, February 13). Several Parkway Parade mall tenants leave or downsize amid rising rents. *The Straits Times*. https://www.straitstimes.com/singapor e/several-parkway-parade-mall-tenants-leave-or-downsize-amid-rising-rents

Pena, W. (n.d.). *AI Business Advisor*. https://chatgpt.com/g/g-e4m1QpiBe-ai-business-advisor

Singapore Company Formation. (n.d.). *Why is location important for a business in Singapore?* https://www.singaporecompanyformation.com.sg/why-is-location-important-for-a-busines s-in-singapore/

xMap. (n.d.). *Pricing Plans*. https://www.xmap.ai/pricing

7.5 Appendix E: Individual Reports

<table>
<tr><td>Name:</td><td>Chua Hieng Weih</td><td>ID:</td><td>A0315386Y</td></tr>
<tr><td colspan="4">Personal Contribution:</td></tr>
<tr><td colspan="4">
- Contributed to the end-to-end system architecture, including the knowledge graph schema, fuzzy logic scoring engine, and GNN integration.
- Evaluated and selected an online platform for data storage.
- Designed a knowledge graph structure optimised for RAG retrieval, defining nodes, relationships and attributes.
- Built and implemented the knowledge graph in Neo4j.
- Researched methods to embed spatial and categorical information and implemented a Heterogeneous Graph Transformer (HGTConv).
- Researched, developed appropriate rule sets, then delivered the fuzzy-logic layer.
- Contributed in some data preparation, which computed the average of industrial property prices across types, and calculated competitor density from counts and subzone area.
- Wrote Python scripts for all the above components.
- Contributed to the project report, writing sections on the system solution, knowledge graph, fuzzy logic, GNN, and related topics.
- Assisted with minor bug fixes in the RAG component.
- Contributed to the system’s video script and overall video production.
- Planned and managed the project timeline, tracked milestones, monitored progress allocated tasks across the team, and controlled scope.
</td></tr>
<tr><td colspan="4">Learning Outcome:</td></tr>
<tr><td colspan="4">
- Applied classroom concepts in machine reasoning to a real-world problem, turning theory into practice.
</td></tr>
</table>

- Integrated state-of-art techniques (GNNs, knowledge graph RAG, and LLMs) into a single cohesive system.
- Gained hands-on experience designing and building knowledge graph, then coupling it with RAG pipelines to feed an LLM.
- Discovered how prompt engineering directly shapes LLM output and refined prompts for clarity and context.
- Became proficient with key tools and libraries: Neo4j for knowledge graphs, PyTorch Geometric for GNNs, and some Python data-science libraries
- As a team leader, I learned to assess teammates' strengths and weaknesses, and allocated tasks accordingly, and encouraged collaboration to meet project deadlines.
- Strengthened communication and coordination skills I had previously lacked, translating high-level ideas into actionable tasks.

Knowledge and Skills Application:

The technical skills I gained translate directly to my work as a software engineer and to future AI roles. I can apply the knowledge-graph, RAG, and LLM integration pattern that many companies now use. The Python tools I learned, such as PyTorch Geometric and graph storage such as Neo4j, can be transferred and used in my work places. My experience designing the system architecture helps me create complete AI and software solutions. I believe the techniques we used such as KG, GNNs, and RAG can also solve problems in other domains.

The soft skills I gained through collaborating with the team, and the lessons I learned from my teammates, will benefit my future work as a tech leader and my day-to-day life.

Name:	Gu Haixiang	**ID:**	A0131920U

Personal Contribution:

Researched and identified relevant data sources suitable for training the business system model.
Extracted, mined, and transformed data, with a focus on demographic and subzone area information.
Collaborated on the development of the RAG and Cypher query scripts.
Conducted extensive testing of the models and system across a wide range of scenario cases.
Evaluated the performance of our model/system against generic LLMs using various metrics.
Contributed to the project proposal and report, especially in sections related to data handling and analysis.

Learning Outcome:

Learned to perform a proper ETL workflow, transforming incomplete source data into structured datasets suitable for model training. Practiced applying RAG on top of the LLM to produce more accurate and intent-aligned responses. Deepened my understanding of backend development, especially in managing Python applications that integrate multiple modules, dependencies, and functions. Gained hands-on experience with machine learning libraries and frameworks such as OpenAI, LangChain, and RAG etc. Familiarized myself with modern tools including Supabase (an open-source Firebase alternative), Neo4j (graph database), and Canva (for presentation and video content). Practiced writing technical documentation for the README, proposal, and project report. Effectively managed time and collaboration within the project team, balancing academic, professional, and personal commitments.
Knowledge and Skills Application:
I have already started applying proper ETL and RAG methods in other LLM-based mini AI tool development projects. As expected, this approach yields more accurate and controlled responses compared to using a pure LLM. This project also sparked deeper reflection on AIOps. In scenarios where the model requires continuous training and improvement, I began considering how to design and implement an automated pipeline to support that process.

Name:	Lizabeth Annabel Tukiman	**ID:**	A0315378X
Personal Contribution:			
For the project, I: - Helped with the first incorporation of RAG into the backend using LangChain, by referring to the Reasoning Systems Day 3 workshops. - Developed a list of predefined Cypher queries for retrieval of information from the knowledge graph. - Contributed to discussions of the system design, such as by sharing papers like Mavromatis and Karypis' "GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning" (https://doi.org/10.48550/arXiv.2405.20139) and giving inputs related to representing data in a knowledge graph. - Helped to test the system by trying various user inputs and conversations. - Contributed to the project proposal and report, including conducting online market research. - Recorded the voiceover used in the technical video and helped with the slides for the			

use case section.
Learning Outcome:
Through this project, I learnt: - More about version control using git and GitHub, including best practices like working on separate branches. - And explored how to extract and process data from open data sources like data.gov.sg, including working with GeoJSON file formats and calculating spatial centroids. While this ended up not being used in our final solution, it allowed me to learn more about data preprocessing. - How to write Cypher queries, by referring to documentation and relying on trial-and-error, as well as my groupmates and ChatGPT, to get the queries to output the desired information. - More about knowledge graphs, exploring nodes and relations in a Neo4j sandbox while working on the Cypher queries. - How to use APIs, like that of OpenAI, Neo4j, and Supabase, in a project. - How to build a RAG chain, by referring to the Reasoning Systems Day 3 workshops and looking at how my groupmates further adapted the code to our project. - The importance of prompt engineering in optimising the performance of LLM-based applications.
Knowledge and Skills Application:
I am currently unemployed, but I believe the knowledge and skills I gained from this project are highly transferable to both personal projects and future roles in data or AI-related fields. For instance, I am currently working on a music-related project and the experience of integrating RAG and prompt engineering makes me feel slightly more confident about incorporating some smart features, such as a chatbot that answers user queries about the artist and recommends songs based on user preferences. Additionally, I am now more familiar with version control and APIs, which would probably be useful in any technical team setting.

Name:	Sritam Patnaik	**ID:**	A0115530W
Personal Contribution:			
For the project, I: - Developed the frontend user interface using React and Tailwind CSS, ensuring a responsive and user-friendly experience for business location advisory tasks. - Added a thumbs up/thumbs down feedback system to each bot response, enabling users to indicate whether the advice was helpful or not. - Built a property preview feature that automatically detects property URLs in bot responses and displays rich previews. - Added support for streaming LLM responses for improved user experience and faster feedback in the frontend. - Designed and implemented the integration between the frontend and backend,			

enabling seamless data flow and dynamic display of business intelligence results.
- Developed to the backend API development using Flask, including the logic for dynamic query parameterization and efficient retrieval of graph-based data.
- Implemented a Retrieval-Augmented Generation (RAG) pipeline in the backend, leveraging LangChain and OpenAI's GPT-4o to provide context-aware, well-reasoned responses to user queries.
- Built an intent classification system using LLM prompts to map user questions to specific business intelligence intents, ensuring accurate query selection and data retrieval.
- Designed and coded entity extraction logic to identify business types and planning areas from user input, supporting both direct questions and follow-up queries using chat history.
- Developed robust error handling and logging throughout the backend (Flask app and RAG pipeline) to facilitate debugging and ensure reliability.

Learning Outcome:

Through this project, I gained hands-on experience in full-stack application development, integrating modern frontend frameworks with advanced backend systems and graph databases. I also learned how to design user-centric chat bot products that are similar to ChatGPT.

Most notably, I found the combination of Retrieval-Augmented Generation (RAG), Knowledge Graphs (KG), and Large Language Models (LLMs), especially using the LangChain framework extremely valuable. This approach allowed me to build a system that can reason over structured business data and provide context-aware, natural language advice. Working with LangChain to orchestrate LLMs and graph queries taught me how to bridge unstructured user input with structured knowledge, which I see as an important tool for future AI-driven applications.

Knowledge and Skills Application:

At my workplace, I'm already using LangChain and Retrieval-Augmented Generation (RAG), though instead of Knowledge Graphs, we rely on SQL for structured data access. Building on concepts from this course and project, I'm developing a chatbot integrated with Lark (our internal communication platform) that serves our sales team. The bot extracts intent and specific questions from natural language queries, dynamically generates SQL based on our database schema, retrieves relevant information, and passes the result, along with contextual business knowledge, to an LLM to generate a helpful, human-like response that the sales team can relay to customers.

I work at Glints, a leading job platform in Indonesia. Beyond this project, I've applied course learnings in two other initiatives: one is a fraud detection system using a gradient-boosted decision tree to identify suspicious employers, and the other is an item-based collaborative filtering model that recommends jobs to candidates based on their behavior.

Name:	Muhammad Harun bin Abdul Rashid	**ID:**	A0164598L

Personal Contribution:

- Contributed to the data collection, preprocessing, and geospatial density analysis.
- Wrote multiple Python scripts to automate data extraction, transformation, and loading (ETL) into Supabase. These scripts involved:
    - Scraping industrial property listings from 99.co
    - Querying the Geoapify Places API to gather competitor business data across various categories
    - Parsing and processing planning area boundaries from GeoJSON data provided by data.gov.sg, to extract subzones and associate them with establishments
- Scripts also handled data cleaning and enrichment, such as:
    - Populating missing planning areas and subzones using reverse geocoding and polygon containment logic
    - Matching coordinates to the correct region using shapely geometries and OneMap APIs
    - Transforming raw data into structured formats suitable for analysis and insertion into Supabase
- Did geospatial density analysis based on demographic data and competitor data in each subzone
    - Computed underserved scores for each pair of subzone and business types

Learning Outcome:

I further deepened my understanding of data engineering pipelines and knowledge graph integration. Learned how to:

- Work with real-world datasets from APIs and government open data sources
- Handle inconsistencies and missing data during preprocessing
- Leverage data science techniques (e.g., underserved score calculation, population-density blending) for decision-making use cases

Knowledge and Skills Application:

These skills are transferable to both software and data engineering roles which is applicable to my current role as a software engineer in a data engineering team. Some of these skills include using Python to setup ETL pipelines for data extraction and cleaning to be used by other downstream services.