

ChatHDB: An Advanced HDB Valuation & Analytics Platform

Executive Summary

ChatHDB is an intelligent platform designed for the Singapore housing market, providing advanced valuation and analytics for HDB properties. This report details the project's business case, highlighting the strong and growing demand for data-driven solutions in the Singapore resale HDB market.

The system architecture integrates a modern technology stack, leveraging machine learning, real-time market data, and an AI-powered chat assistant to offer users comprehensive insights. Key features include ML-powered valuation, adjusted market value calculations incorporating real-time trends, an interactive map view, and a suite of analytical tools.

While the platform demonstrates significant potential, areas for future development include refining the valuation logic and enhancing data integration. The analysis concludes that ChatHDB aligns well with market needs and presents a valuable tool for homeowners, property agents, and institutions in Singapore.

1. Introduction

1.1 Project Overview: ChatHDB - Advanced HDB Valuation & Analytics Platform

ChatHDB represents a sophisticated approach to understanding the Singaporean Housing and Development Board (HDB) resale market. It is an intelligent tool that furnishes users with advanced HDB property valuations and in-depth analytics, capitalizing on the synthesis of machine learning methodologies and up-to-the-minute market data to deliver precise insights.

At its core, the platform aims to empower users with the knowledge necessary to navigate the complexities of the HDB resale market. Key features underpinning this objective include a machine learning-driven valuation model, an adjusted market value that reflects contemporary market dynamics, an interactive map interface for spatial analysis, a comprehensive array of analytical tools, and an AI-powered chat assistant designed to address user queries efficiently.

1.2 Report Objectives and Scope

This report serves as a comprehensive documentation and analysis of the ChatHDB project. Its primary objectives are to articulate the project's underlying business case, to elucidate its technical design and architecture, and to present a discussion of the findings and potential avenues for future development. This includes an examination of the market landscape, the system's technical underpinnings, its implementation details, and a mapping of its functionalities to relevant academic disciplines.

2. Business Case and Market Analysis

2.1 Market Overview of Singapore's HDB Resale Market

Singapore's HDB resale market demonstrates resilient fundamentals, underpinned by consistent demand, limited housing supply, and a growing appetite for data-driven decision-making. In 2024, resale transaction volumes rose by 8.4%, reaching 28,986 units—a signal of increased market liquidity and growing confidence among both buyers and sellers.

Over the past two decades, resale prices have maintained a steady compound annual growth rate (CAGR) of 4.7%. Looking ahead, a moderate CAGR of 4% is projected through 2029, with the market value expected to reach SGD 21 billion. The recent launch of HDB's official resale portal in 2024 has further democratized access to listings, catalyzing demand for tools that deliver actionable insights beyond static historical data.

This shift marks a critical inflection point—signaling strong alignment with ChatHDB's core value proposition: empowering stakeholders through predictive analytics, sentiment tracking, and contextual market intelligence. As the market becomes more complex and competitive, the demand for solutions like ChatHDB is poised to expand.

2.2 Target Market Segmentation

ChatHDB is strategically designed to serve three primary market segments:

1. **Individual Buyers and Sellers** – Digitally savvy consumers leveraging platforms such as HDB's resale portal, 99.co, and PropertyGuru seek clarity in pricing and insights into neighborhood dynamics. ChatHDB enhances their decision-making by offering intelligent recommendations and valuation transparency.
2. **Property Agents and Agencies** – Real estate professionals require

differentiated tools to elevate client servicing and negotiation capabilities. ChatHDB equips agents with automated valuation models, macroeconomic context, and market sentiment analyses—boosting their advisory edge.

3. **Institutions and Government Bodies** – Entities including the HDB, financial institutions, and urban planning agencies benefit from ChatHDB’s data infrastructure, APIs, and predictive analytics. These capabilities support planning, policy development, and investment strategies grounded in robust, real-time insights.

2.3 Market Sizing and Potential

A top-down market sizing approach reveals significant opportunity. Singapore’s resale transaction value is projected to reach SGD 21 billion by 2029 (see Appendix D Table 1 for assumptions). Applying a conservative estimate of 10% of agent commission value for tech adoption yields a Serviceable Addressable Market (SAM) of approximately SGD 41 million.

Based on projected adoption rates among agents, integration partnerships, and platform traction, ChatHDB targets a 5% market share of this SAM within five years—equating to SGD 2 million in annual recurring revenue (see Appendix D Table 2 for projections). This trajectory supports a compelling investment and operational case for scaling

2.4 Market Strategy

ChatHDB’s go-to-market strategy prioritizes user acquisition through value-added partnerships, integration with established listing portals, and tailored B2B offerings. By embedding its tools into existing digital ecosystems, ChatHDB augments rather than competes with incumbent platforms.

The platform’s edge lies in its holistic and forward-looking insights—integrating macroeconomic indicators, sentiment analysis, and amenity data—to provide real-time, contextual valuation. This positions ChatHDB as a strategic advisor rather than a passive listing tool, fostering long-term user engagement and institutional relevance.

For government and enterprise clients, ChatHDB offers structured APIs and analytics dashboards that facilitate urban policy planning, macro-financial modeling, and real estate portfolio optimization.

2.5 Industry Analysis: Porter's Five Forces

Threat of New Entrants (Moderate): While simple property search tools are easily replicable, ChatHDB's proprietary valuation framework, multi-source data integration, and domain-specific intelligence present substantial technical and data-related barriers to entry.

Bargaining Power of Buyers (Moderate to High): Individual users have access to competing platforms; however, ChatHDB's B2B integration strategy and focus on complementing rather than replacing existing tools help mitigate direct user substitution.

Bargaining Power of Suppliers (Moderate to High): Reliance on external data is a risk factor. To address this, ChatHDB is actively investing in proprietary pipelines, public data partnerships, and in-house analytics capabilities.

Threat of Substitutes (Moderate): Existing calculators and trend reports lack ChatHDB's predictive depth and personalization. Its explainable AI models and visualization tools create clear differentiation.

Industry Rivalry (High): The sector is crowded, but few platforms offer ChatHDB's level of analytical depth. Its data-first positioning enables it to carve out a niche focused on value-added intelligence rather than simple property listings.

2.6 Business Analysis: Strength, Weakness, Opportunities, Threats (SWOT)

Strengths: ChatHDB's integration of macroeconomic, sentiment, and geospatial data elevates its valuation precision. Its visual and interactive interfaces enhance user trust and engagement. API-based data delivery expands B2B utility.

Weaknesses: As a new entrant, building brand equity and data robustness are critical challenges. High integration complexity requires sustained technical investment.

Opportunities: Rising demand for smart property tools and governmental interest in market transparency offer expansion avenues. AI adoption trends further validate the platform's relevance.

Threats: Incumbents may replicate analytical features, and data regulations may evolve. ChatHDB's adaptability and investment in proprietary data mitigate these risks.

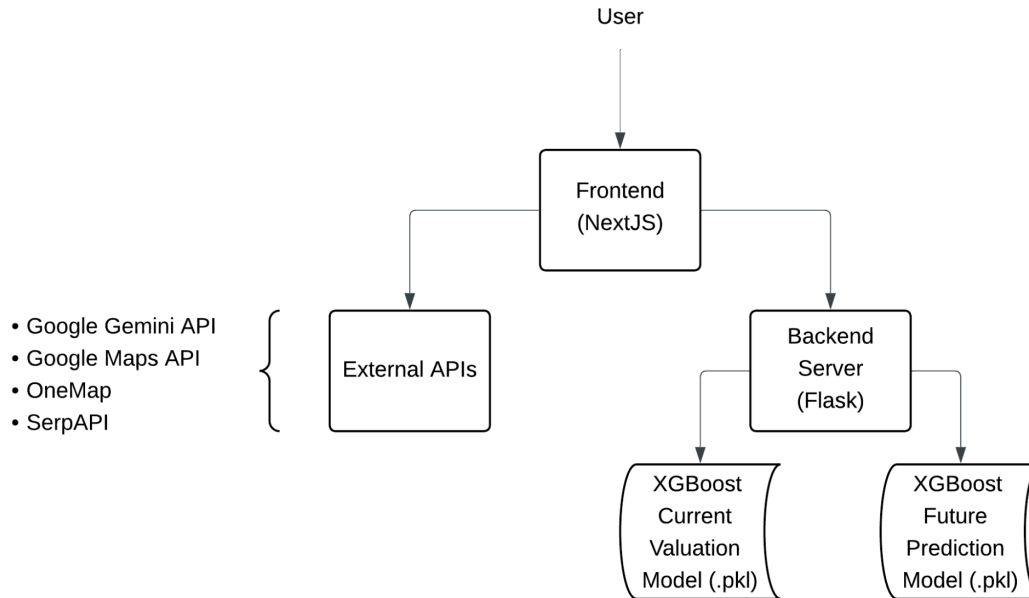
2.7 Viability and Strategic Outlook

The confluence of rising demand for data-driven decision-making, increasing digital adoption in property transactions, and the availability of real-time data sources presents a timely and compelling case for the viability of ChatHDB as a business. By bridging the gap between raw public data and actionable insights, ChatHDB addresses a critical market need that is currently underserved by conventional listing platforms. Its value proposition extends beyond individual buyers and agents to institutions seeking structured, predictive analytics for strategic planning.

The integration of scalable, real-time data pipelines and cloud infrastructure, as outlined in the platform's future roadmap, positions ChatHDB for long-term growth and operational resilience. Moreover, its modular architecture enables seamless B2B integrations, allowing for monetization through licensing, and white-label partnerships.

In light of the expanding HDB resale market and the clear trend toward analytics-enabled property services, ChatHDB is strategically placed to emerge as a differentiated insights platform. With a sound technical foundation, a well-defined go-to-market strategy, and strong alignment with evolving market demands, ChatHDB demonstrates both commercial viability and long-term strategic relevance.

3. System Design and Model



3.1 Price Prediction Using XGBoost

We use XGBoost regression to predict both current and future resale prices, chosen after evaluating multiple regression models for accuracy and performance.

3.2 Resale Price Adjustment via External Data

To enhance prediction, we apply a multiplier to the base price using alternative data sources like economic indicators, Google Trends, and news sentiment. These were aggregated separately due to their differing granularity.

3.3 Chatbot & Text Analysis

The system features a chatbot built on Google's Gemini 2.0 Flash model, offering contextual answers to HDB property queries. It personalizes responses by processing uploaded PDF valuation reports and interacts in real-time via streaming, using markdown and dynamic suggestions. Text analysis also employs Gemini 2.0 Flash to classify sentiment (positive, negative, neutral) from real-time property news fetched via SerpAPI, generating multipliers that directly impact property valuations.

3.4 User Interaction via Website

After using Python libraries (sklearn, pandas) to train the regression models, we save

the models into two .pkl files and store them as static files in the Flask backend server. The frontend hosted on <https://chathtb.vercel.app/> will then call the Flask server to retrieve the price prediction values given some user inputs.

4. Key Use Cases

4.1 Explainable AI for Current & Future Valuation

ChatHDB enables users to request a property valuation by entering a postal code and key features such as floor level, unit size, and lease start year. A pre-trained XGBoost model predicts the base resale price for both current and future timeframes. This prediction is then refined using a multiplier informed by external data sources like economic trends and sentiment analysis.

To foster user confidence, we provide transparent explanations of the model's workings, how each input contributes to the final price, and how the multiplier is derived.

4.2 Interactive Map View

ChatHDB utilizes Google Maps Platform for an interactive map. After a user enters a postal code, the OneMap API provides coordinates. The map then displays the target property alongside nearby HDB transactions (filtered from local data) and local amenities such as schools, MRT stations, and malls (sourced from Google Places API), offering a visual overview of the neighborhood and recent sales.

4.3 Comprehensive Analytics

ChatHDB provides in-depth analytics, visualizing filtered transaction data with Nivo charts. Users can explore historical price trends (line chart), price variation by storey (bar chart), transaction volume distribution (pie chart), lease decay effects (line chart), and comparative area price performance (animated race chart) for nuanced market insights.

4.4 AI Chat Assistant

To enhance user experience and provide immediate answers to HDB-related queries, ChatHDB incorporates an AI Chat Assistant powered by Google Gemini. This chat assistant allows users to ask questions in natural language about various aspects of

the HDB market, property valuation, regulations, and more.

A key feature of this AI assistant is its ability to provide context-aware follow-up suggestions. After answering an initial query, Gemini can suggest related questions that the user might find helpful, encouraging further exploration and providing a more comprehensive understanding of the topic. The integration of a sophisticated AI chat assistant like Gemini makes ChatHDB a more interactive and informative platform, offering users on-demand access to a wealth of HDB-related knowledge.

4.5 Market Pulse

ChatHDB includes a "Market Pulse" feature designed to provide users with a real-time snapshot of the current market sentiment and trends. This feature integrates two key elements: Google Trends visualization and Top News Stories with sentiment analysis.

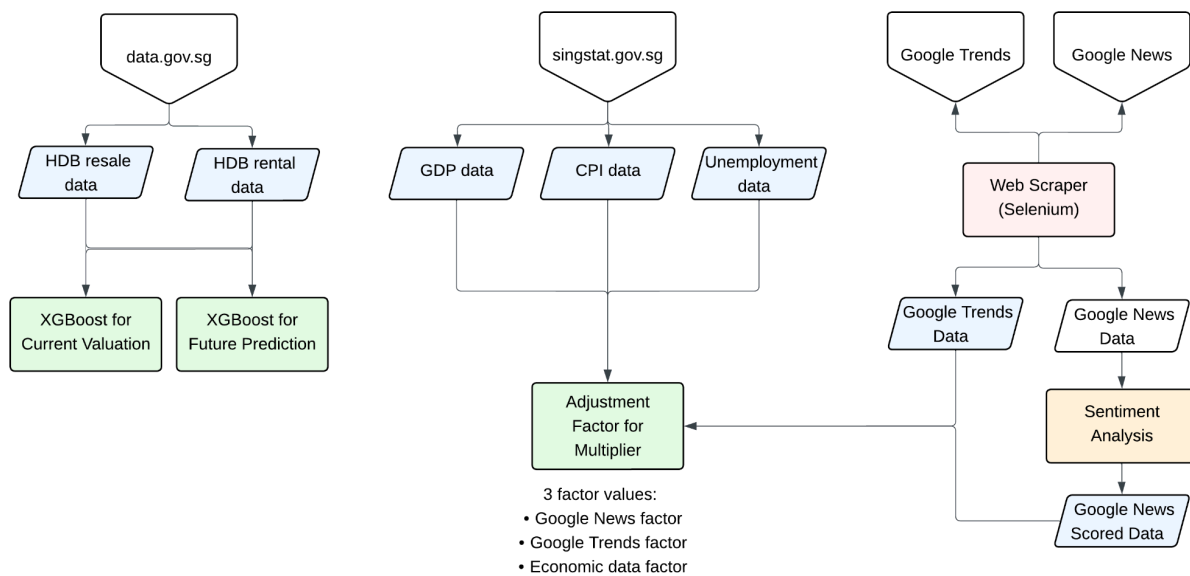
The visualization of Google Trends data allows users to see the current level of search interest in HDB-related topics in Singapore. This can provide an indication of market activity and buyer/seller sentiment.

Additionally, the platform fetches the top news stories related to the Singapore property market. These news articles are then analyzed using Google Gemini to determine the overall sentiment (positive, negative, or neutral) prevailing in the news. This sentiment analysis provides users with a qualitative understanding of the current market mood, complementing the quantitative data from the valuation models and analytics.

By integrating these real-time indicators, the "Market Pulse" feature helps users stay informed about the latest developments and sentiment shifts in the Singapore HDB resale market.

5. System Development and Implementation

5.1 Overall Application System Design



The ChatHDB system architecture is structured around four primary components that interact to deliver its functionalities: the frontend, backend server, the machine learning models, and the data scraping and cleaning.

The frontend is built using NextJS and ReactJS and serves as the user interface, enabling users to interact with the platform and visualize the results. It also interacts with external APIs to fetch live information, interact with third-party LLMs, and our backend server using Next.js API Routes and Node.js, providing a robust and scalable foundation.

In addition, the data scraping and cleaning component integrates various data sources, fetching data from various sources and performing data cleaning to better integrate with our use case.

The next step is the model training, which is performed on the data stored in the backend codebase as static files. For the current stage of the project, the model output is saved in a Pickle format and called subsequently by the APIs.

Lastly, the backend server is built using a Flask server and is responsible for handling

the interactions between the frontend and the models we built. The AI/ML module encompasses the intelligent features of ChatHDB, including property valuation and sentiment analysis. This component utilizes Python with Scikit-learn and XGBoost and uses the Flask library to handle the API interactions.

Table 1: Main Components and Their Responsibilities

Component Name	Description/Responsibility
Frontend	<p>Handles user interface, input, and output, built with Next.js, React, Tailwind CSS, etc.</p> <p>The frontend interacts with external APIs such as Google APIs to fetch live updates on news and interaction with the map, as well as other sources like Google Gemini, OneMap API and SerpAPI.</p>
Data Scraping and Cleaning	<p>Python web scraping and data cleaning scripts were also used to consolidate data required for model training. These provide the necessary data for the system to function, including HDB resale data, economic indicators, Google News and Google Trends data.</p>
Backend Server	<p>Flask server is used for integration with static models built by the team.</p>
Model Training	<p>Provides intelligent features like price valuation, prediction, and sentiment analysis, utilising Python with XGBoost for the current and future pricing prediction models, as well as calculation of the Pearson Correlation Coefficient for the multiplier value.</p>

Table 2: Technology Stack

Category	Technology
Frontend	<p>Next.js (App Router), React, TypeScript, Tailwind CSS, shadcn/ui, Framer Motion, Nivo, ReactFlow, Google Maps Platform (JavaScript</p>

	API, Places API)
Backend/API	Flask Python, Next.js API Routes, Node.js
AI/ML	Python (Scikit-learn, XGBoost), Google Gemini
Data Sources & APIs	HDB Resale Flat Prices CSV, Economic Indicators CSV, OneMap API, SerpAPI

5.2 Web Scraping and Data Cleaning

Data Sources and APIs

ChatHDB relies on a variety of data sources and external APIs to provide its comprehensive valuation and analytics services.

Data sourcing strategies varied across project phases to suit evolving development needs. During the initial development phase, data from Google Trends and online news articles were scraped and stored locally to support preprocessing efforts and facilitate the identification of factors influencing HDB resale price fluctuations. As the project progressed toward the development of the minimum viable product (MVP), the data sourcing process was refined through the use of SerpAPI, enabling more consistent and structured retrieval of search trends and news data.

Data cleaning processes were also tailored based on the nature of the data—specifically distinguishing between scraped and readily available datasets. For scraped data, such as Google Trends and news articles, the raw responses were initially saved as static files. These files were then parsed into tabular formats, after which preprocessing steps were applied to ensure consistency in key labels, including time periods and town classifications. This step was crucial for aligning the auxiliary data with the main HDB resale dataset. The cleaned data were then consolidated based on shared attributes to prepare them for downstream integration.

In contrast, for publicly available datasets—such as resale and rental prices from *data.gov.sg* and macroeconomic indicators from *singstat.gov.sg*—a structured multi-stage cleaning approach was employed. Static data files were first downloaded and stored locally. In the initial preprocessing stage, the resale dataset was enhanced by computing a `remaining_lease` variable, defined as the difference between the transaction year and lease commencement year, assuming a 99-year lease duration. The macroeconomic data underwent temporal disaggregation (e.g., converting

quarterly to monthly formats) and were backfilled to address missing values and maintain continuity. In the second stage, the cleaned resale and macroeconomic datasets were merged. Additional backfilling was applied to macroeconomic variables during integration to ensure completeness for subsequent modeling and analysis tasks.

In addition to these datasets, several APIs were integrated to enrich data quality and enhance application functionality. The OneMap API, provided by the Singapore Land Authority, was used for geocoding, enabling the conversion of postal codes and street names into geographic coordinates. SerpAPI facilitated the real-time retrieval of data from Google Trends and top news stories relevant to the property market, offering timely insights into search interest and current events that could influence resale valuations. The Google Maps Platform, including the JavaScript API, was employed to render interactive maps and visualize property locations alongside nearby resale transactions. Furthermore, the Google Places API provided contextual data on nearby amenities such as schools, MRT stations, shopping malls, and parks—adding depth to property profiles and enhancing user experience.

The integration of these diverse data sources and APIs enables ChatHDB to present a holistic, data-driven perspective of the Singapore HDB resale market. However, it is important to note that the current proof-of-concept relies heavily on locally stored CSV files, which may limit data freshness compared to real-time API integrations with official sources. A more robust and scalable data integration framework is proposed in Part Seven: *Future Works*.

5.3 Model Building

5.3.1 Price Prediction with XGBoost

In the model development phase, we are following the standard procedures of data preprocessing, model training, fine-tuning, and evaluation.

During data preprocessing, we performed the following steps:

- Transformed categorical location data using one-hot encoding. From our dataset, we had the option to use either town or street name as the location reference. We opted for the more detailed street name level, as the dataset provides sufficient density—averaging over 1,500 transactions per street—allowing for more granular modeling.

- Encoded the HDB flat types (e.g., 4-room, 5-room, maisonette) using one-hot encoding.
- Applied standard scaling to numerical features such as the lease commencement year and floor area.
- Categorized floor levels into three bands, low, medium, and high, based on an approximately equal distribution, followed by standard scaling. This approach reflects our belief that floor level impact does not require individual floor precision.

Model training, we have

- [Extreme Gradient Boosting](#) (XGBoost), relatively new regression technique that has garnered a lot of attention. The algorithm is based on decision trees, optimized for speed and performance. It works by combining many weak learners (simple models) to form a strong predictive model, making it highly effective for structured data and regression tasks like price prediction.
- [Random Forest](#) (RF), an ensemble learning method that builds multiple decision trees and combines their outputs to improve prediction accuracy and reduce overfitting. Each tree is trained on a random subset of the data and features, making the overall model more robust and generalizable.
- Neural Network (NN), a classic model that can uncover complex patterns. Models such as Long Short-Term Memory (LSTM) can be used to predict future housing pricing. It captures long-term contextual information in sequential data such as time-series data.

From the training, we find that XGBoost has slightly higher R^2 (0.86) vs NN (0.68) & RF (0.63). The R^2 scores for RF and NN can be higher if we increase the training time. However, as we have over 1 million rows and 600 columns, the training time required by NN & RF to reach the R^2 is already around 8 minutes, longer than XGBoost's 3 minutes training time.

With that, we decide to go with fine-tuning XGBoost.

The fine-tuning process leverages a Bayesian optimization approach using the [Optuna](#) library. Optuna is an open-source hyperparameter optimization framework designed to efficiently search for the best model parameters. By applying Bayesian optimization, the tuning process intelligently explores the parameter space, balancing exploration and exploitation to converge on optimal settings faster than traditional grid or random search methods. This helps improve model performance while

reducing the computational cost of experimentation.

As this is a regression problem, the evaluation is done using R^2 and root mean squared error (RMSE). We had R^2 value of 0.98 and RMSE of 23,433.

Lastly, for the training of future price prediction, we re-train the model and hide the last 2 years data from the test data. We run through the same process and ends up with R^2 of 0.80

Explanation for XGBoost Performance

After performing testing with multiple models, we have come to the conclusion that XGBoost has the best performance. We have then further evaluated the strengths of the XGBoost model to review and assess the performance of the model in our application.

- **Scalability:** As mentioned above, we have a dataset of over a million rows and a few hundred columns. XGBoost is known to run ten times faster than existing popular models on a memory-limited setting due to multiple factors, including using a sparsity aware algorithm for handling sparse data, as well as using out-of-core computation to allow us to process millions of rows of data on a single machine.
- **Accuracy:** XGBoost also boasts a high accuracy score compared to other models because it learns the full tree. While other models such as Random Forest also rely on decision trees, it has lower performance relative to XGBoost when dealing with the minority classes in datasets. This is because Random Forest only takes into account a random subset of features while constructing each separate decision tree, and this results in a lower representation for minority classes such as Singapore streets with less HDB resale transactions. On the other hand, XGBoost uses a greedy algorithm that enumerates over all the possible splitting points.

5.3.2 Multiplier Estimation with Adjustment Factor

The base valuation generated by the machine learning model is further refined to reflect current market conditions by incorporating real-time multipliers. These multipliers are calculated based on three key factors: Google Trends, News Sentiment, and Economic Indicators. News sentiment is analyzed on the collected Google News data for each area over the past years to determine the overall sentiment (positive,

negative, or neutral).

These three multipliers are then applied to the base valuation to produce the final Adjusted Market Value, which provides a more dynamic and up-to-date estimate of the property's worth, taking into account real-time market dynamics and broader economic factors.

In the Google News dataset, we utilized both the aggregated score and mean score to capture the overall sentiment strength and consistency at the town level. To synthesize these metrics into a unified sentiment indicator, we summed the two values and then applied a time decay function, giving greater weight to more recent data points and gradually diminishing the impact of older observations. This temporal weighting ensures that our model reflects current sentiment trends more accurately. After weighting, the resulting score was scaled using min-max normalization, constraining the output within a custom range of -0.03 to 0.03, ultimately producing the `adj_factor_gnews` used in our final adjustment model.

For the economic component, we derived a simplified macroeconomic signal by summing the Consumer Price Index (CPI) and Gross Domestic Product (GDP), then subtracting the unemployment rate, representing a proxy for general economic health. Since this data is not town-specific but varies by month, we selected the most recent month's data to represent the national economic climate. This value was also normalized to fall within the same range, yielding the `adj_factor_econ`.

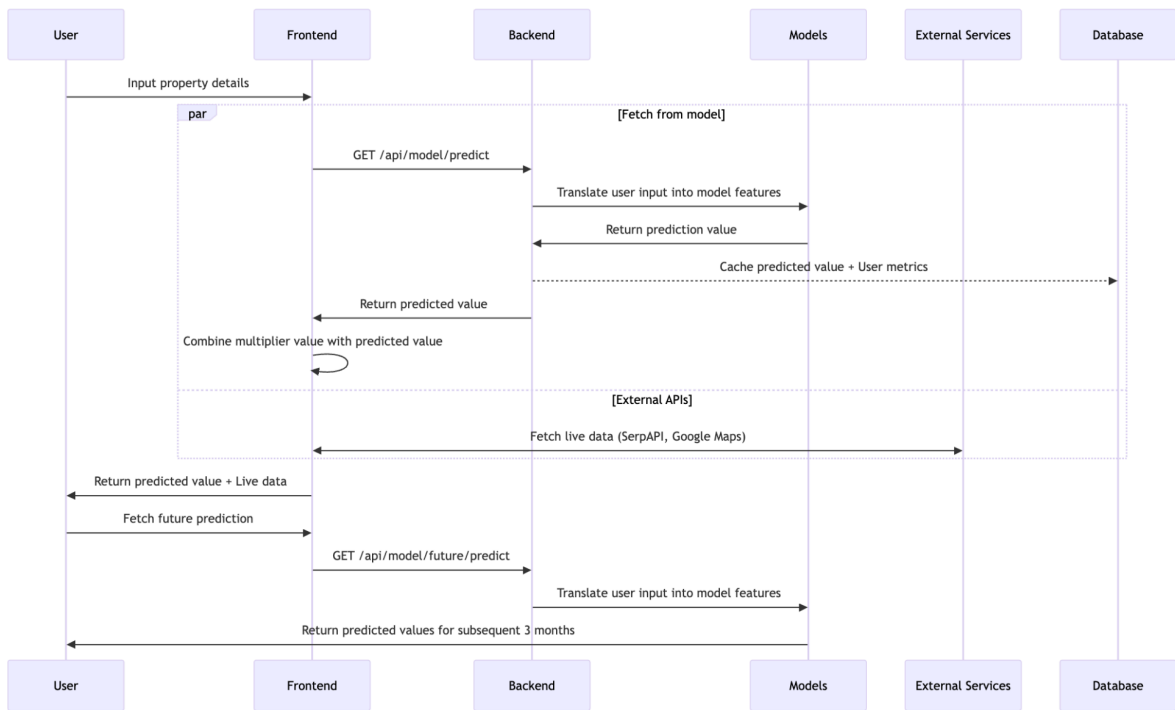
The third adjustment factor, `adj_factor_gtrend`, was based on Google Trends data, capturing the search volume and interest levels at the town level over time. Similar to the GNews data, we applied the same time decay logic to prioritize recent user interest while reducing the influence of older data. The decayed trend scores were then normalized into the defined range to ensure consistency across towns.

In the current version of our model, we assume equal weighting among these three adjustment factors. Each factor reflects a unique and complementary dimension—public sentiment, macroeconomic signals, and collective attention—and when used together, they help modulate the final prediction results in our downstream analysis, with recency-aware adjustments improving responsiveness to real-world dynamics.

To ensure consistency and control across our adjustment model, we applied min-max normalization to scale each factor—sentiment, search trends, and economic indicators—into a uniform range of -0.03 to 0.03. This approach allowed us to

harmonize data from different domains while maintaining interpretability and balance in their impact on the final prediction. By bounding the adjustment values, we prevented any single source from dominating the outcome, allowing each factor to contribute incrementally. Looking back, this method not only provided stability to our model but also helped preserve meaningful relative differences between towns, ensuring our adjustment framework remains both practical and transparent for real-world application.

5.4 Web Application



5.4.1 System Architecture

The user interface of ChatHDB is constructed using a modern technology stack, primarily based on Next.js with its App Router

5.4.1 Frontend Application

The user interface is built with Next.js (App Router) and React, utilizing TypeScript for type safety and improved maintainability. For our styling & UI, we used component

libraries like shadcn/ui and Framer Motion adds animations for a smoother user experience. We have an extensive use of client directive to focus on client-side rendering for rich interactivity and dynamic updates.

5.4.2 Backend Services

The backend logic is split into two main parts: API routes co-located with the Next.js frontend for orchestration and data fetching, and a separate Python Flask server dedicated to machine learning model predictions.

A. Next.js API Routes (Orchestration & Data Integration)

We have serverless functions, running on Node.js via Next.js API Routes, to handle user requests, interact with external services, and communicate with the Flask ML backend.

Key Functions:

1. Communication with Google Gemini for AI chat features.
2. Fetching real-time data (Google Trends, news) via SerpAPI, potentially followed by Gemini-based sentiment analysis.
3. Executing ML predictions, likely by calling the dedicated Flask ML backend API (/api/valuation route).
4. Potential Implementation: While the primary ML execution seems delegated to Flask, the architecture allows for Node.js to execute other Python scripts (e.g., using `child_process`) if needed for specific tasks.

B. Flask ML Backend Server (Modeling Service)

We also have a dedicated Python Flask application that serves the machine learning models for property valuation.

The backend server is built using Flask Python, and deployed on Google Cloud Run for the frontend to interact with.

It uses Flask backend infrastructure to coordinate and define API routes, a static

folder to store any static files such as the Pickle files for the models, and logic for data scraping, cleaning, as well as interactions with the frontend. The details for the various Flask APIs can be seen under Appendix E.

For future expansions, we would like to integrate some new features in the backend server to better serve the users with the latest information.

- We would like to periodically re-train the model based on the latest information. In order to do so, the Flask server can include a Cron scheduler which runs a daily task to fetch the latest information on data such as resale price, Google news and trends, etc. With the latest data, we can re-train the model and serve it via our APIs.
- To determine and improve the performance of our website, we can also integrate user metrics tracking. This includes tracking user footprint through our various APIs, to determine which functions are more popular - the current price valuation or future price prediction.

6. Challenges

During development, we encountered several key challenges:

- **Data Availability:** We initially considered incorporating rental data as an input, but due to its sparse availability—reducing our usable dataset by up to 90%—we ultimately excluded it.
- **Data Leakage:** In early experiments, we observed unrealistically high R^2 scores (up to 0.998) even without fine-tuning, which we traced back to data leakage when attempting to compute missing rental values using another model.
- **Training Time:** Limited computational resources meant that a single fine-tuning run could take over three hours, restricting our ability to explore multiple algorithms or tuning strategies.
- **Multiplier Calibration:** While economic indicators, sentiment, and trend data are valuable, deriving a reliable coefficient to adjust base prices proved challenging due to their varied nature, sparsity and granularity.
- **API Costs:** Our reliance on Google APIs for alternative data introduces significant cost considerations, especially when scaling the system.
- **Policy Sensitivity:** HDB prices are heavily influenced by government regulations, such as changes in ABSD or COVID-related measures. Since our model is trained on historical transactions, it can lag in adapting to sudden policy shifts, particularly right after new regulations are introduced.

7. Future Works

Looking ahead, there are several promising directions to further enhance this project:

- **Automated Data Engineering Stack:** To support scalability, improve data freshness, and enable continuous integration, future iterations of the data pipeline will transition from local storage to a robust, cloud-based data engineering stack. Given that HDB resale price data from *data.gov.sg* is updated daily on working days, while macroeconomic indicators from *singstat.gov.sg* are updated quarterly, a hybrid update cadence through automated API pulls will be implemented to efficiently handle the differing data frequencies. This transition will enable near-real-time data availability and allow ChatHDB to deliver up-to-date insights and predictive analytics with greater reliability and reduced manual overhead.
- **Policy Impact Modeling:** A deeper analysis of how past government

regulations—such as ABSD adjustments or cooling measures—have influenced resale price trends could improve the model’s ability to anticipate the effects of future policy shifts. This not only enhances model responsiveness but may also uncover valuable insights into the effectiveness of specific regulatory interventions.

- **Extension to Private Property Market:** Expanding the scope to include private residential transactions would allow us to build a parallel pricing model for private properties, enabling broader market coverage and comparative insights between public and private housing trends.
- **Refining Future Price Prediction:** Although current results show strong performance (with high R^2 scores), further research could focus on boosting future price prediction accuracy—possibly by incorporating macroeconomic indicators, long-term sentiment trends, or more advanced temporal modeling techniques.
- **Real-Time Adaptive Modeling:** Developing a dynamic model that updates continuously as new data (e.g., transactions, policies, or economic signals) becomes available, minimizing lag in prediction accuracy.
- **User Personalization:** Tailoring predictions and explanations based on user profiles or intent (e.g., buying vs. investing) to improve relevance.

8. Conclusion

ChatHDB presents a practical and forward-looking approach to HDB resale valuation, integrating machine learning with both structured and alternative data sources to provide accurate, transparent predictions. The use of XGBoost for base price modeling, enhanced by external multipliers from economic indicators and sentiment trends, allows the system to reflect real-world dynamics more effectively.

Throughout development, we encountered key challenges including data sparsity (especially in rental data), computational constraints during fine-tuning, risks of data leakage, and the difficulty of calibrating external multipliers. Additionally, the model currently lags in responding to sudden regulatory changes due to its reliance on historical transactions.

Looking ahead, future work can focus on modeling the impact of policy changes, extending the framework to private properties, and improving the future pricing model through better temporal and economic features. With these enhancements, ChatHDB has strong potential to become a robust and trusted valuation tool for both everyday

users and institutional stakeholders in Singapore's evolving property market.

Appendices

Note: as our project proposal is very long, we moved it to be the last appendix

Appendix A: Mapped System Functionalities against Modular Courses

ChatHDB Functionality	Related technique
Price Prediction	Data preprocessing Linear Regression
Multiplier Calculation	Sentiment Analysis with Hugging Face
Chatbot	Natural Language Cognition

Appendix B: Installation and User Guide

Installation Guide

1. **Clone the repository:** `git clone https://github.com/your-username/ChatHDB.git`
2. **Navigate to the project directory:** `cd ChatHDB`
3. **Install dependencies:** `npm install` or `yarn install`
4. **Set up Environment Variables:** Create a `.env.local` file in the root directory and add the following variables:
Google Maps API Key (Frontend usage)
NEXT_PUBLIC_GOOGLE_MAPS_API_KEY=YOUR_GOOGLE_MAPS_API_KEY
Google Gemini API Key (Backend API usage)
GEMINI_API_KEY=YOUR_GEMINI_API_KEY
SerpAPI Key (Backend API usage for Trends/News)
SERPAPI_API_KEY=YOUR_SERPAPI_API_KEY
OneMap API Token (Optional - If hardcoded token expires)
ONEMAP_TOKEN=YOUR_ONEMAP_TOKEN
Obtain the necessary API keys from the respective platforms:
 - **Google Maps API Key:** Google Cloud Console
 - **Gemini API Key:** Google AI Studio
 - **SerpAPI Key:** SerpApi Website
 - **OneMap Token:** OneMap API
5. **Run the development server:** `npm run dev` or `yarn dev`
6. **Open in browser:** Open `http://localhost:3000` in your browser.

User Guide

1. **Input Postal Code and Flat Type:** On the homepage, enter a Singapore postal code and select the desired flat type (e.g., 3 Room, 4 Room).
2. **View Map and Transactions:** The system will display the location on an interactive Google Map, along with markers for recent HDB transactions in the vicinity. Nearby amenities will also be shown.
3. **Request Valuation:** To get a valuation, provide additional details such as the storey range, floor area, and lease commencement year in the valuation modal.
4. **Understand Adjusted Market Value:** The platform will display the Adjusted Market Value, which is the base valuation refined by real-time market multipliers.
5. **Access Comprehensive Analytics:** Click on the analytics button to open a modal with various charts visualizing historical price trends, price distribution, transaction volume, lease decay, and area comparisons.

6. **Interact with AI Chat Assistant:** Use the chat window to ask any HDB-related questions. The AI assistant will provide answers and suggest follow-up questions.
7. **View Market Pulse:** The Market Pulse section displays current Google Trends for property-related searches and top news stories with sentiment analysis.
8. **Explore Model Transparency:** Navigate to the Model Transparency section to visualize the steps and factors involved in the property valuation process.

Appendix C: Project Proposal

ChatHDB

Group Number: 8

Introduction

Project Background

ChatHDB is designed to assist both buyers and sellers by providing critical information to facilitate property pricing decisions, with a primary focus on **price estimation**. In addition to pricing insights, ChatHDB offers key market summaries to enhance user understanding.

To achieve this, we develop a sophisticated pricing model by integrating multiple data sources, including:

- **HDB resale transaction data**
- **Unit-specific attributes (location, size, floor level, etc.)**
- **Rental market trends**
- **Sentiment analysis of news articles**
- **Google Trends data**

Additionally, ChatHDB offers mid-term price increase projections to assist investors in evaluating potential appreciation.

Beyond pricing insights, ChatHDB enhances information accessibility by providing:

- **Concise market summaries** available on a web platform
- **An AI-powered chatbot** that assists users in retrieving key market information
- **Real-time web search capabilities**, enabling access to the latest updates on:
 - Building attributes and conditions
 - Current sentiment about the area
 - News on urban planning and development
 - Nearby amenities

Market Research

The Singapore HDB resale market faces several challenges that hinder informed decision-making for buyers and sellers:

- **Valuation Uncertainty:** Determining fair market values is difficult due to the complex interplay of various price-influencing factors.

- **Information Asymmetry:** Limited access to comprehensive market data results in knowledge gaps among participants.
- **Decision Complexity:** Beyond price, property decisions are influenced by location quality, amenities, and future developments.
- **Market Volatility:** Rapid changes in market conditions make it challenging to stay updated on trends.

These challenges contribute to suboptimal decision-making, extended transaction times, and potential financial losses.

Project Scope

ChatHDB integrates three or more AI techniques as required by the assignment:

1. **Decision Automation:**
 - Automated fair pricing estimation based on structured market data and sentiment analysis.
2. **Business Resource Optimization:**
 - AI-powered chatbot for user-friendly and efficient information retrieval.
3. **Knowledge Discovery & Data Mining:**
 - Analysis of past HDB transactions and property characteristics.
 - Sentiment analysis from news articles to assess market perception.
 - Google Trends analysis for consumer interest insights.
4. **Cognitive System Design:**
 - **Web-based platform** (hosted on Vercel) to present data in an intuitive manner.
 - **Chatbot powered by GPT** for interactive property-related queries.

Data Collection and Preparation

Data sources:

- **HDB resale transaction data**
 - Source: data.gov.sg (Housing & Development Board dataset)
 - Features: town, flat type, block, street name, storey range, floor area, flat model, lease commencement date
 - Preprocessing: calculation of flat age based on lease commencement date and time of sale; one-hot encoding of categorical data such as town, flat type, storey range, flat model
- **Rental transaction data**
 - Source: data.gov.sg (Housing & Development Board dataset)
 - Features: town, flat type, block, street name, monthly rental amount
 - Preprocessing: aggregation of monthly rental amount based on town, flat type, block, street name
- **Population income levels**
 - Source: data.gov.sg (Ministry of Manpower dataset)
 - Features: annual median monthly income
 - Preprocessing: temporal disaggregation of higher frequency income
- **Planning area data**
 - Source: API calling onemap.gov.sg
 - Features: classification of URA town planning area
 - Preprocessing: extraction of planning area name from geojson; reclassification of town planning area based on classes within resale transaction dataset
- **News articles**
 - Source: web crawling of Google News RSS xml
 - Features: article title
 - Preprocessing: sentiment analysis based on article title; aggregation of lower frequency dataset
- **Search trends**
 - Source: web crawling of Google trends
 - Features: relative search interest of the respective town over time
 - Preprocessing: reformatting of month year

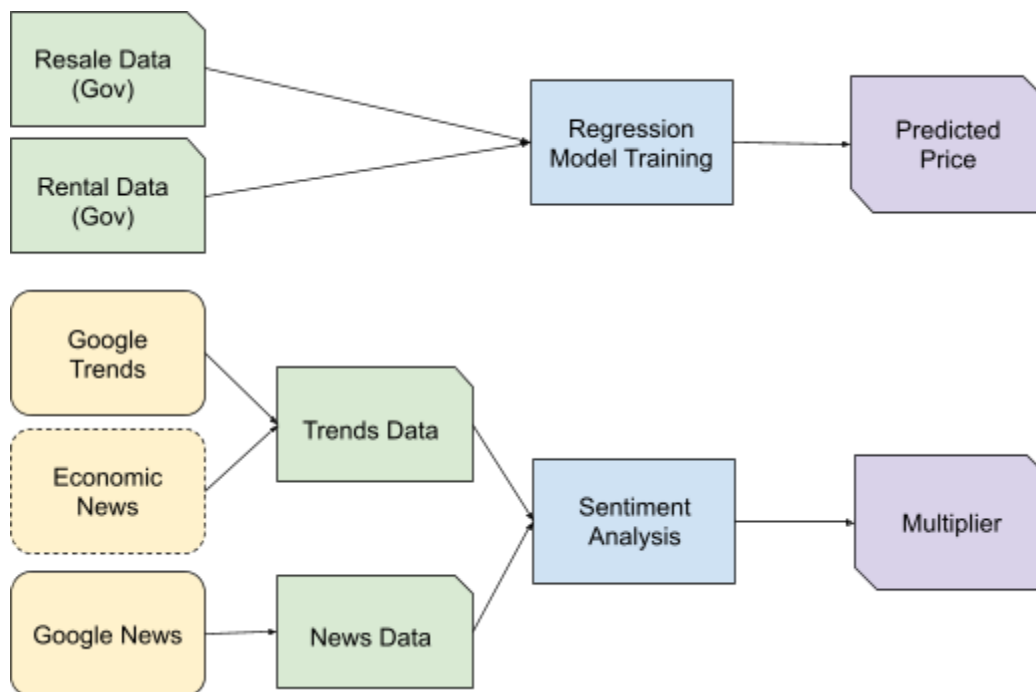
Challenges faced:

- Dataset like the news article and search trends are not readily available and require web scraping and parsing of the raw response into usable data.
- Alternative datasets are in varying levels of granularity which requires aggregation and disaggregation to map to resale transaction dataset.
- Alternative datasets are sparse and do not extend to the entire time frame of the resale transaction dataset.

System Design

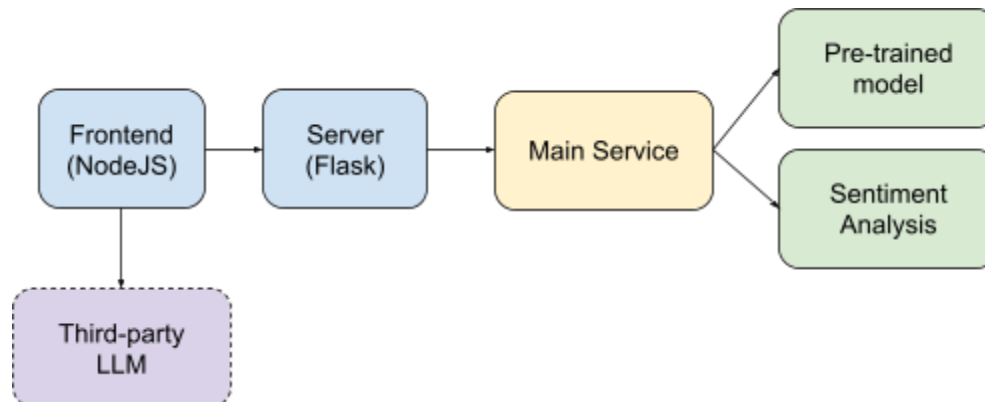
Considering the market requirements for the project, we want to focus on several aspects of the application: Price prediction, estimation of a price multiplier, as well as an LLM chatbot.

Price Prediction and Multiplier



Application Overall Architecture

We will run the frontend on Vercel using the NextJS framework, and the backend will be running on Google Cloud Run using Flask.



Implementation

We will begin by selecting suitable models and then proceed with practical steps to optimize their performance.

Since our objective is to predict both **current resale prices** and **future resale prices**, we are dealing with a **regression problem**. As a baseline model, we will use **linear regression** to establish a simple benchmark.

For our primary model selection, we have researched regression algorithms that perform well on large datasets. The three promising candidates are:

- **XGBoost** - known for its efficiency and accuracy in structured data,
- **Random Forest** - a robust ensemble method that handles non-linearity well, and
- **Neural Networks** - capable of capturing complex patterns in the data.

To determine the best model, we will conduct comparative testing across these algorithms, evaluating them based on key performance metrics: **R² (coefficient of determination)** and **MSRE (Mean Squared Root Error)**.

Beyond model selection, we will implement additional steps to enhance performance:

1. **Data Preprocessing** – Handling missing values, scaling numerical features, and applying one-hot encoding for categorical variables.
2. **Model Fine-Tuning** – Optimizing hyperparameters by testing multiple configurations to achieve the best performance.

This structured approach ensures that we select and refine the most effective model for resale price prediction.

Prediction Models

Model	Description
XGBoost	<p>XGBoost, short for extreme gradient boosting, is an optimised distributed gradient boosting library designed to be highly efficient, flexible and portable. It is designed for efficiency and uses decision trees for its base model.</p> <p>Each new tree is trained to correct errors made by the previous tree, and it has built-in parallel processing to train models on large datasets quickly.</p>
Random Forest	Random Forest algorithm also uses decision trees to make predictions

	based on ensemble learning . It uses many decision trees to do a consolidated voting to make a prediction, where each tree is trained by taking a random different part of the training dataset.
Neural Networks	Recurrent neural networks (RNNs) such as Long Short-Term Memory LSTM can be used to predict future housing pricing. It captures long-term contextual information in sequential data such as time-series data, which would be helpful in our project.

Multiplier Estimation

On top of using related variable data to predict pricing, we are also going to scrape and consolidate Google News for each area of housing (e.g., Woodlands, Tampines) and split these news by month and year. We will then transform this data into a time-series data and perform **sentiment analysis** on each area for a given month and year. Correlating this with the results from the price prediction model, we can estimate and check for any correlation between the Google News sentiments and the price fluctuations.

Based on the correlation between the news sentiment results and resale price fluctuations, we expect to predict a multiplier value which estimates the impact on the changes in resale price based on overall societal sentiments. We would like to test this hypothesis and see if any accurate relation can be drawn here.

However, a challenge posed here would be the scraping and obtaining of the news. Currently, we are using Google News to obtain raw data and clean them up. However, some areas in Singapore could be largely underrepresented in the news data, as they could be too new or for other reasons. We might have to extrapolate based on nearby areas if necessary.

Conclusion

ChatHDB leverages AI-driven decision automation, data mining, and cognitive user interaction techniques to address challenges in the HDB resale market. By integrating real-time and historical data, it empowers users with actionable insights, ultimately improving decision-making efficiency in Singapore's property market.

AI Disclaimer

This project proposal has been refined using AI for grammar correction and coherence. We also plan to utilize AI tools to assist with code debugging and brainstorming model approaches.

Appendix D: Business Valuation and Strategy Breakdown





Table 1
Projected HDB Resale Market and ChatHDB Addressable Market (2024–2029)

Metric	2024 (Actual)	2025	2026	2027	2028	2029
Total resale value (billion SGD)	17	18	18	19	20	21
Resale value YoY growth	16%	4%	4%	4%	4%	4%
Commission rate	2%	2%	2%	2%	2%	2%
Total commission value (million SGD)	341	355	369	384	399	415
Serviceable addressable market (SAM) share	10%	10%	10%	10%	10%	10%
SAM (million SGD)	34	35	37	38	40	41
Serviceable obtainable market (SOM) share	5%	5%	5%	5%	5%	5%
SOM (SGD)	1,705,100	1,773,304	1,844,236	1,918,006	1,994,726	2,074,515

Resale HDB transactions YoY growth	—	3.50%	3.50%	3.50%	3.50%	3.50%
Total resale HDB transactions	28,986	30,001	31,051	32,137	33,262	34,426
Property agents YoY growth	—	0%	0%	0%	0%	0%
Total property agents	35,251	35,251	35,251	35,251	35,251	35,251

Note. YoY = year-over-year. SAM = Serviceable Addressable Market; SOM = Serviceable Obtainable Market. All monetary values in SGD.

Table 2

ChatHDB			
Monetization Strategy and Revenue Projections			
Segment	Monetization Approach	Pricing Model	Estimated Annual Revenue (SGD)
 Platforms	Platform fee	50,000 per platform license	$50,000 \times 6 = 300,000$
 Banks & Mortgage Providers	AVM licensing	75,000 per platform license	$75,000 \times 3 = 225,000$
 Government & Research Institutions	Data insights & forecasting	50,000 per platform license	$50,000 \times 2 = 100,000$
 Property Agents	Lite version subscription	240 per user	$240 \times 7,050$ (20% of agents) = 1,692,048
TOTAL			SGD 2,317,048

Segment	Monetization Approach	Pricing Model	Estimated Annual Revenue (SGD)
Platforms	Platform fee	50,000 per platform license	$50,000 \times 6 = 300,000$
Property Agents	Lite version subscription	240 per user	$240 \times 7,050$ (20% of agents) = 1,692,048
Banks & Mortgage Providers	AVM licensing	75,000 per platform license	$75,000 \times 3 = 225,000$
Government & Research Institutions	Data insights & forecasting		

Total

Note. AVM = Automated Valuation Model. Revenue calculations assume base-case adoption rates.

Appendix E: Flask Server APIs

GET <code>/api/health</code>	Healthcheck to ensure the server is running.
GET <code>/api/model/predict</code>	Given parameters from the frontend, return base predicted price. Parameters: <ul style="list-style-type: none">- street_name- floor_area- storey_range- lease_start- flat_type
GET <code>/api/model/predict/test</code>	Get test future prediction data to ensure model calling is working as expected. No parameters required.
GET <code>/api/model/future/predict</code>	Given parameters from the frontend, return predicted price for the <i>next 3 months</i> . Parameters: <ul style="list-style-type: none">- street_name- floor_area- storey_range- lease_start- flat_type
GET <code>/api/model/future/predict/test</code>	Get test future prediction data to ensure model calling is working as expected. No parameters required.