

COMP4801 Final Year Project

SimpleTripAdvisor: An AI-Driven Travel Planning Platform for Personalized Itineraries

Wenlong Wu Student ID: 3035771220

Supervisor: Prof. **Hengshuang Zhao**

April 21, 2025

Abstract

Planning a trip can be a daunting and overwhelming task, often requiring extensive research across multiple platforms. This report introduces SimpleTripAdvisor AI, an intelligent travel planning platform that leverages natural language processing (NLP) to create personalized itineraries through conversational interactions. The platform aims to incorporate a web-based interface with a chatbot assistant to deliver a seamless and user-friendly experience. Currently, the platform utilizes the Gemini-2.0-Flash application programming interface (API) as its underlying large language model (LLM) for both dialogue management and recommendation generation. Looking ahead, the development team plans to explore training a dedicated neural network model to serve as a specialized recommendation engine for enhanced performance.

Acknowledgment

I would first like to extend my deepest gratitude to the supervisor, Professor Hengshuang Zhao, for his invaluable guidance and unwavering support throughout this project. His insightful advice and constructive feedback have been essential in overcoming critical challenges. This project would not have advanced so smoothly without his mentorship.

I would also like to express my heartfelt appreciation to my teammate, Yijiazhen Qi, for his significant contributions to this project. His dedication to designing and implementing the front-end interface has been crucial to the progress we have achieved.

Their support and collaboration have been vital to the success of this work, and I am truly thankful for their commitment and effort.

Table of Contents

Abstract	i
Acknowledgment	ii
Table of Contents	iii
List of Figures	v
List of Tables	vi
Abbreviations	vii
1 Introduction	1
1.1 Existing Solutions	1
1.2 Objective	2
1.3 Deliverables	2
1.4 Report Outline	3
2 Methodology	3
2.1 Adaptive Development Using Agile Methodology	3
2.2 Technology Stack and Environment Configuration	4
2.3 Overall System Architecture	5
2.4 Object-Oriented Data Model: Class Design and Entity Relationships	6
2.5 LLM Selection and Conversational AI Integration	7
2.6 Incorporation of External APIs	8
3 Application Workflow and System Showcase	9
3.1 Interactive Planning Flow	9

3.2	Interactive Use-Case Walk-through: A Typical User Journey in SimpleTripAdvisor AI	10
4	Future Enhancements and Expected Challenges	17
4.1	Toward a Specialized Recommendation Engine	17
4.2	Challenges in Data Collection and Feature Engineering	18
4.3	Enhancing Prompt Quality and Contextual Understanding in LLMs	19
5	Conclusion	20
	References	21

List of Figures

1	Detailed system architecture of the SimpleTripAdvisor AI platform.	5
2	Backend class diagram of the platform, showing the structure and associations between key entities.	6
3	Comparison of capabilities between Gemini-2.0-Flash-exp and GPT-4o-mini. . . .	7
4	Comparison of output quality and processing speed between Gemini-2.0-Flash-exp and GPT-4o-mini.	8
5	The user interaction flow within the platform.	10
6	Initial interface inviting users to begin a trip with natural-language input.	11
7	System-generated response to a user query for a solo 7-day Hong Kong itinerary, combining web-sourced inspiration and a structured travel plan.	12
8	Map and assistant-generated information for Ocean Park in Hong Kong, combining geolocation, a brief overview, and follow-up question suggestions.	13
9	Flight search and tracking interaction. Left: query results for flights between Hong Kong and Osaka. Right: real-time flight status for UO686, including departure/arrival details and gate information.	14
10	Proactive follow-up from the assistant, offering next steps such as alert setup, alternate travel dates, or booking assistance.	15
11	Responses to auxiliary user queries. Left: current date and time in the user's time zone. Right: 5-day weather forecast for Mongkok, Hong Kong, shown with temperature trends.	16

List of Tables

1	Comparison of Recommendation Techniques for Travel Planning	18
---	---	----

Abbreviations

NLP	Natural Language Processing
API	Application Programming Interface
LLM	Large Language Model
AI	Artificial Intelligence
HCI	Human-Computer Interaction
MVT	Model-View-Template
BFF	Backend-for-Frontend
DCD	Design Class Diagram
NN	Neural Network
PCA	Principal Component Analysis

1 Introduction

Artificial intelligence (AI) technology has become indispensable in daily life, offering solutions to various information-intensive tasks. Earlier researchers employed AI models to evaluate credit risk in finance [1] and to make medical diagnoses [2]. Planning a trip is similar to these tasks, as it involves gathering and analyzing large amounts of information to create optimized itineraries.

Today, the process of planning a trip remains complex and tedious. It is challenging for travelers to obtain all the necessary information from a single source or platform, which can leave them unaware of alternative routes or services [3]. Additionally, when planning a trip to a foreign destination with a different culture, cultural barriers and biases can arise during the information-gathering process. A study on the satisfaction of online trip planning for China [4] found that most American college students had a frustrating experience when organizing their trips due to cultural barriers. The dominant information portals in the U.S. can introduce biases into online travel resources, making them less effective for planning trips to China. To address these issues, AI's ability to manage large data sets and deliver comprehensive and unified information suggests that it may be well-suited for the task.

1.1 Existing Solutions

Current AI-powered travel planning tools such as Google Travel and Airbnb Experiences provide a wide range of services to travelers. However, they typically require users to manually review and assess numerous options, which can be both time-consuming and daunting. Google Travel offers in-depth planning capabilities but relies on user-provided information rather than offering predictive, personalized suggestions. Airbnb Experiences focuses on unique experiences but lacks an integrated approach to overall travel planning. These shortcomings indicate an unmet need for a travel planner that provides a more holistic and user-friendly experience.

1.2 Objective

This project is designed to close the gaps in existing solutions. The core motivation is to make trip planning an easy and enjoyable experience within a single, integrated platform by leveraging advanced technologies such as NLP. Instead of being a complex and time-consuming task that discourages people from traveling, trip planning should inspire the excitement of exploring.

This project also embraces the chatbot-based mode of human-computer interaction (HCI), which has become increasingly prevalent and widely accepted [5]. Many potential users are already accustomed to this form of interaction and are more inclined to engage with systems that support it. Additionally, new users face a lower barrier to entry, as the conversational interface eliminates the need to learn how to navigate multiple complex features. By consolidating functionalities into an intuitive chat interface, the platform becomes more accessible and attractive to a broader audience. Promoting this modern interaction model, which is still uncommon in traditional travel planning tools, is also a key contribution of this project, as it enhances user-friendliness and lowers the learning curve.

Ultimately, the objective of this project is to develop an AI-powered travel planning platform that stands out in providing users with unified information, personalized itineraries, and real-time updates, all through natural-language conversations with the AI assistant.

1.3 Deliverables

The deliverables for this project include the development of SimpleTripAdvisor AI, composed of two main components: a web-based user interface and an NLP-powered conversational agent. The website will offer all the necessary features to users to interact with the AI advisor, providing a smooth planning experience. The platform's chatbot interface will serve as the user's entry point, where inputs are processed and relayed to the underlying LLM to generate personalized travel plans. In the future, the development team plans to train a dedicated neural network model to serve as a more specialized and efficient recommendation engine. The team will also work

on developing mobile application versions for both iOS and Android devices to enhance cross-platform compatibility.

1.4 Report Outline

This report proceeds with a comprehensive methodology section (Section 2), discussing the system architecture and the technological foundation. Section 3 highlights the project outcomes, including a brief user interaction flow and a demonstration of the web-based application. Section 4 explores potential future enhancements and anticipated challenges, such as enhancing chatbot prompts and gathering training data for the recommendation model. Finally, Section 5 concludes the report, summarizing the key contributions and findings.

2 Methodology

This section outlines the structured approach adopted for the development of the project, presenting the overall system architecture. It also justifies the selected methodologies used for each component in terms of its functionality and design.

2.1 Adaptive Development Using Agile Methodology

To develop the SimpleTripAdvisor AI platform, the team employed the Agile methodology, chosen for its adaptability and iterative workflow [6], which aligns well with this project's dynamic and evolving nature. Agile facilitates continuous refinement through user and supervisor feedback, and allows for flexibility in incorporating new requirements, making it particularly well-suited for building a personalized, interactive AI-based system.

2.2 Technology Stack and Environment Configuration

This project was initially deployed as a web application. The web provides better accessibility since users can visit the website immediately without downloads. Moreover, web development is generally less complex than building mobile applications or desktop software, as it avoids platform-specific tools and extensive hardware optimization.

To ensure a consistent and reliable environment across development, testing, and deployment phases, Docker containers were used to set up the development environment. Python and Django were selected for the backend due to their efficiency in rapid development and robust support for RESTful APIs. For the frontend, React.js was adopted to create a responsive and interactive user interface.

In choosing a backend framework, the team evaluated both Django and Flask, two widely used Python web frameworks. While Flask offers simplicity and flexibility for small-scale applications, it requires additional configuration and third-party integrations as the system grows, which can lead to increased complexity and potential security risks.

Django, in contrast, follows a “batteries-included” philosophy, providing built-in tools such as an admin interface, authentication, and security features. These capabilities align well with the needs of SimpleTripAdvisor AI, which requires scalability, maintainability, and efficient development workflows. Although Django may incur some performance overhead compared to Flask, its caching features and database optimization options were sufficient to meet the project’s expected performance demands.

React.js was chosen for the frontend due to its robust ecosystem and component-based architecture. Libraries like Shadcn and NextUI accelerated the development process by offering pre-built components, allowing the team to focus on delivering the platform’s core interactive features efficiently. Additionally, React.js is well-established and has strong community support. Many popular websites, including Netflix and Facebook, are built using it, demonstrating its scalability and robustness.

2.3 Overall System Architecture

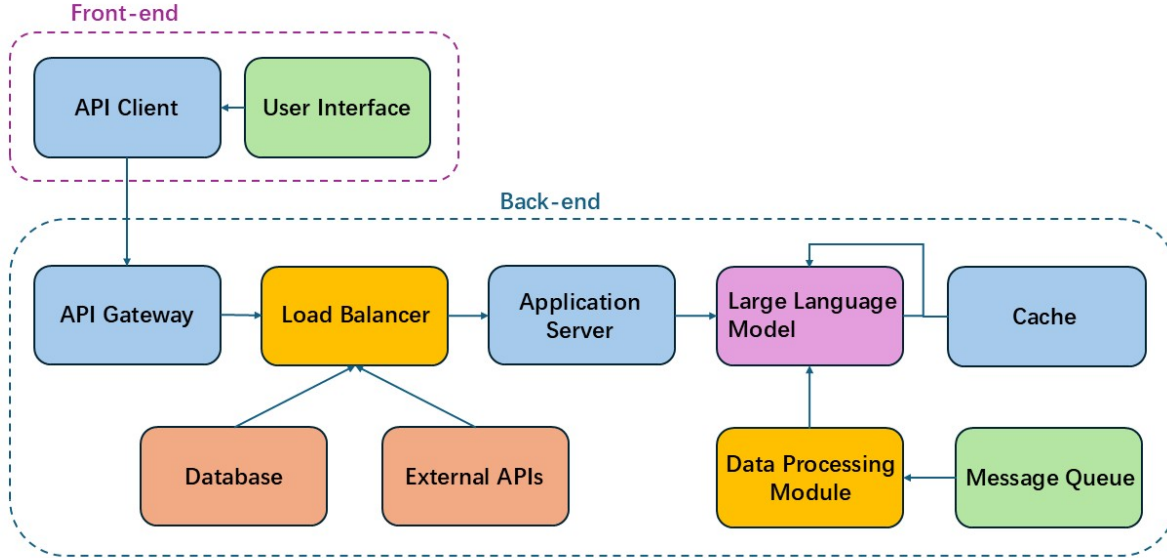


Figure 1: Detailed system architecture of the SimpleTripAdvisor AI platform.

The system was built using a modular architecture, incorporating a Model-View-Template (MVT) structure and a Backend-for-Frontend (BFF) pattern. Figure 1 illustrates the overall system architecture of the web-based application. The frontend and backend are hosted on separate servers. The frontend delivers an engaging chat-based interface, while the backend handles user authentication, optimizes data traffic distribution, and interacts with the LLM. Communication between the frontend and backend was established mainly using RESTful APIs and partially WebSocket APIs via HTTPS protocol.

The BFF pattern was chosen for its scalability benefits, particularly in anticipation of future expansion to a mobile platform. By decoupling the frontend and backend, the system allows the development of dedicated interfaces for different platforms while reusing the same backend infrastructure. In contrast, a tightly coupled architecture would hinder this scalability, as introducing a new frontend (e.g., mobile) would require significant backend modifications, leading to code duplication and increased maintenance effort. While the BFF pattern introduces additional initial

complexity and development overhead, such as managing two separate servers, it provides long-term flexibility and maintainability.

2.4 Object-Oriented Data Model: Class Design and Entity Relationships

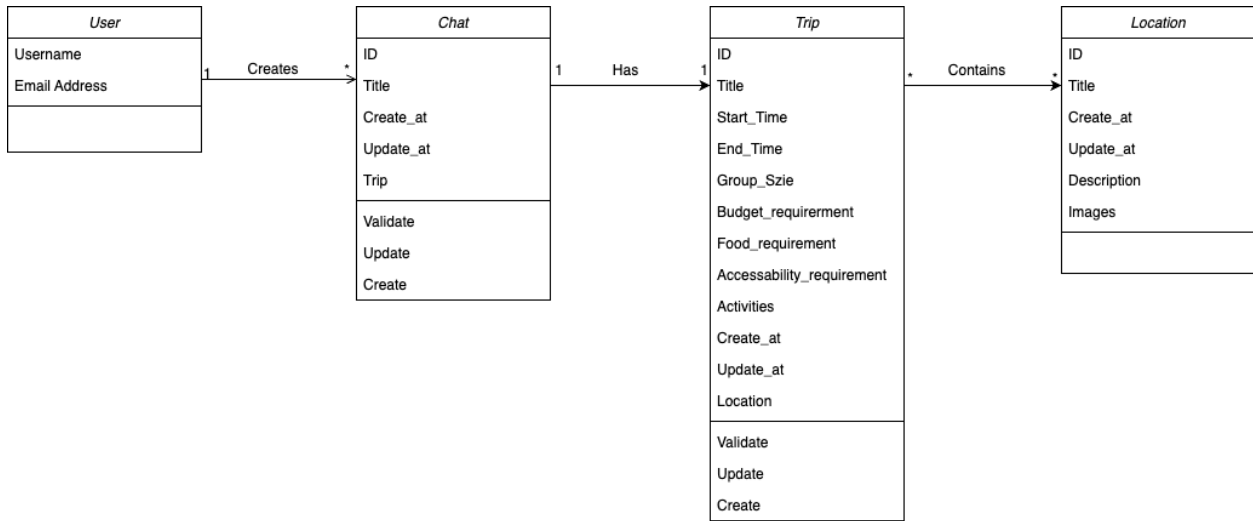


Figure 2: Backend class diagram of the platform, showing the structure and associations between key entities.

The Design Class Diagram (DCD) of the backend is presented in Figure 2, illustrating the relationships and interactions among the core entities: **User**, **Chat**, **Trip**, and **Location**.

The **User** entity initiates a **Chat**, which is associated with a single **Trip**. In turn, each **Trip** can include multiple **Locations**, representing various destinations planned during the trip. This design supports personalized, AI-guided interactions, where user preferences are dynamically captured and reflected in the generated recommendations.

Each class in the diagram includes relevant attributes and methods. The **User** entity stores essential authentication data, such as *Username* and *Email Address*, ensuring proper identification. The **Chat** entity logs planning sessions through attributes like *Title*, *Create_at*, and *Update_at*, and is responsible for creating or updating linked trips. The **Trip** entity manages planning details including *Start_Time*, *End_Time*, *Group_Size*, *Budget_requirement*, etc. Lastly, the **Location** entity holds information about destinations, such as *Title*, *Description*, and associated *Images*.

2.5 LLM Selection and Conversational AI Integration

The conversational AI for this project is powered by the Gemini-2.0-Flash-exp model, integrated via its API. During model selection, the team compared it primarily against GPT-4o-mini. Gemini-2.0-Flash-exp was ultimately chosen for its superior performance across several key dimensions, including knowledge representation, multilingual support, larger context window, faster response times, cost-efficiency, and native web search integration. As illustrated in Figures 3 and 4, Gemini outperforms GPT-4o-mini in all evaluated metrics, as supported by Artificial Analysis [7]. Notably, Gemini offers a 1 million token context window and free-tier access, while GPT-4o-mini charges \$0.150 per million input tokens and \$0.600 per million output tokens, with a smaller 128K context window. Furthermore, Gemini’s direct integration with Google Search removes the need for separate function calls, unlike GPT-4o-mini.



Figure 3: Comparison of capabilities between Gemini-2.0-Flash-exp and GPT-4o-mini.

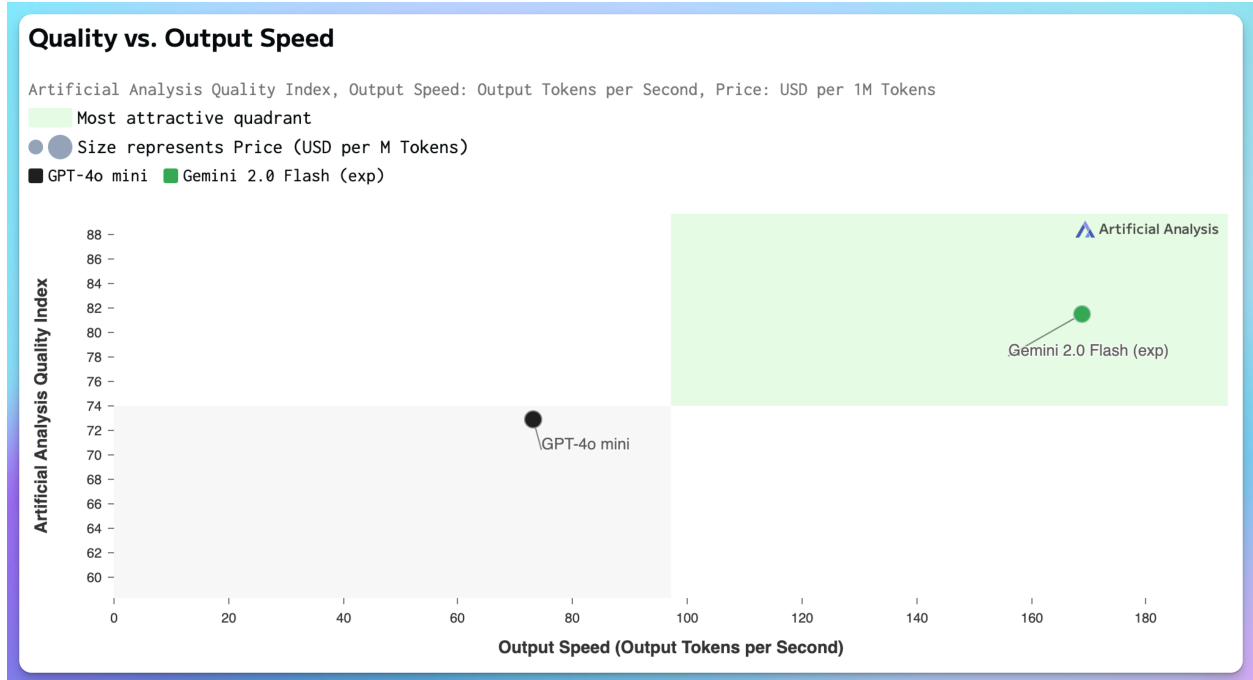


Figure 4: Comparison of output quality and processing speed between Gemini-2.0-Flash-exp and GPT-4o-mini.

Despite its advantages, integrating Gemini’s API came with certain challenges. During development, the team implemented both Gemini and OpenAI API connections in parallel. However, Gemini’s more complex and fragmented documentation, split between genai (for individuals) and vertex (for enterprises), slowed the integration. Additionally, third-party LLMs such as Deepseek [8] and Qwen [9] currently offer OpenAI-compatible APIs but not Gemini support, reducing future model-switching flexibility. Moreover, while Gemini’s free tier suited the development needs, scaling to production may introduce costs as the user base grows.

2.6 Incorporation of External APIs

To enhance the platform’s overall functionality, the integration of weather and geolocation services is essential for delivering real-time insights and personalized recommendations.

Rather than utilizing a standalone weather API such as OpenWeather, the platform will leverage the built-in Google Search functionality available within Gemini-2.0-Flash-exp to retrieve weather-

related information. This method enables quick and reliable access to up-to-date weather conditions without the need for additional API dependencies. It simplifies the development workflow and minimizes potential integration challenges, thereby improving efficiency and maintainability.

For location-based functionalities, Mapbox has been adopted as the primary geolocation service provider due to its customizable map styling, efficient rendering performance, and developer-friendly API ecosystem. The integration enables precise geolocation visualization, such as dropping pins on specific destinations and displaying detailed geographic contexts. While Mapbox does not natively provide user-generated reviews like Google Maps, its modular design allows developers to integrate external data sources, such as TripAdvisor or Yelp, based on specific use case needs. Compared to all-in-one platforms, Mapbox offers greater flexibility and a lighter footprint.

3 Application Workflow and System Showcase

This section presents the functional outcome of the project, including the user interaction flow and a demonstration of the developed web-based application. The goal is to illustrate how users engage with the platform and to showcase the key features and interface design.

3.1 Interactive Planning Flow

As illustrated in Figure 5, the system adopts a structured, step-by-step dialogue approach to guide users through the travel planning process. Through a series of dynamically generated questions, the platform gathers essential user preferences, such as travel dates, budget, interests, and accessibility needs, and refines its recommendations in real time. By the end of the interaction, a personalized itinerary is generated based on the user's inputs and feedback throughout the session. All sessions are saved to the database, allowing users to revisit their previous interactions at any time. Users can either view past recommendations or re-engage with the conversation to further refine and customize their travel plans.

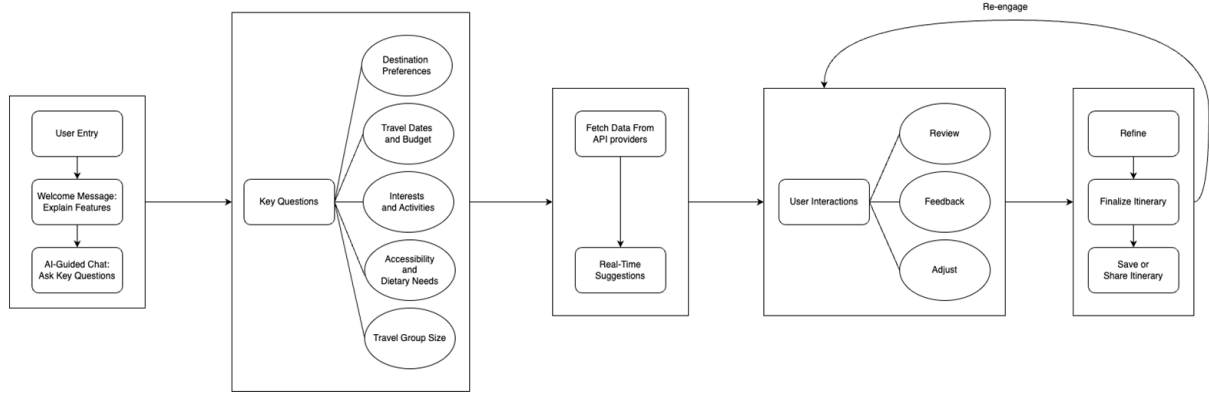


Figure 5: The user interaction flow within the platform.

3.2 Interactive Use-Case Walk-through: A Typical User Journey in SimpleTripAdvisor AI

The following section provides a comprehensive walkthrough of a typical interaction with SimpleTripAdvisor AI, showcasing how its core features work together to support travel planning. From initiating a conversation to generating itineraries, exploring destinations, checking flights, and retrieving helpful information like weather and time. This flow serves both as a functional demonstration and as a practical user guide. All features of the platform can be accessed via a unified conversational interface, designed to be intuitive, responsive, and supportive of multilingual inputs.

For readers interested in implementation details, source code and technical documentation are available on the project’s GitHub repository ¹.

¹<https://github.com/KiwiGaze/SimpleTripAdvisor>

Initiating a Trip – A Minimalist Conversational Gateway

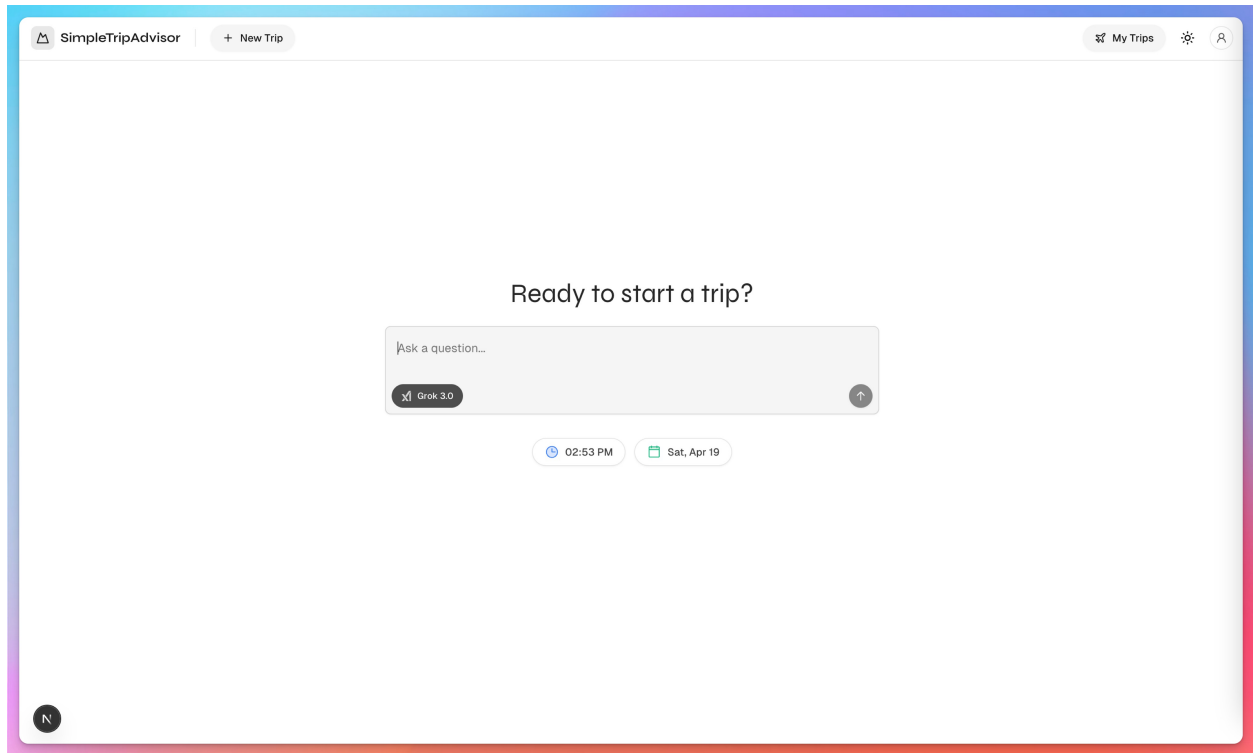


Figure 6: Initial interface inviting users to begin a trip with natural-language input.

Upon entering the platform, users are greeted by a clean interface with a central prompt: "Ready to start a trip?" (Figure 6). A single input field allows users to initiate interaction through free-form natural language, encouraging flexibility in how queries are expressed. The system can parse and respond to a wide range of travel-related inputs, including incomplete or exploratory questions. This entry point is designed to be intuitive and welcoming, with responsive support across all major browsers and adaptive light/dark themes for enhanced accessibility.

Itinerary Generation – Structured Planning Based on Personalized Requirements

Create a 7-day solo travel itinerary in Hong Kong on a moderate budget.

Sources Found37 Results

7-day solo travel Itinerary Hong Kong moderate budget 2025

best attractions in Hong Kong for solo

The Best Hong Kong Solo Trav...

www.theladywhotravels.com

Set aside \$100–\$150 per day. This covers mid-range accommodation, local food like dim sum or noodles, and efficient public...

Hong Kong Travel Guide...

www.nomadicmatt.com

A backpacking budget covers staying in a hostel dorm, eating fast food occasionally but mainly cooking your own meals, using public...

H

w

Missing:

10 Day Hong Kong Itinerary

TOUR OF HONG KONG

A traditional junk boat with red sails glides through the water in front of a

A traditional Chinese junk boat sails in vibrant turquoise waters,

SimpleTripAdvisor AI

I'm excited to help you plan a 7-day solo travel itinerary in Hong Kong on a moderate budget. Below, I've crafted a detailed plan that balances iconic attractions, cultural experiences, and budget-friendly options for accommodations, dining, and activities. I've assumed a moderate daily budget of approximately 800-1,200 HKD (around 100-150 USD) per day, covering mid-range accommodations, local food, public transportation, and entry fees to attractions [The Lady Who

Ask a new question...

Ask 3.0

SimpleTripAdvisor AI

I'm excited to help you plan a 7-day solo travel itinerary in Hong Kong on a moderate budget. Below, I've crafted a detailed plan that balances iconic attractions, cultural experiences, and budget-friendly options for accommodations, dining, and activities. I've assumed a moderate daily budget of approximately 800-1,200 HKD (around 100-150 USD) per day, covering mid-range accommodations, local food, public transportation, and entry fees to attractions [1]. Since specific travel dates weren't provided, this itinerary is flexible and can be adjusted based on your schedule. I've also prioritized solo-traveler-friendly activities and neighborhoods.

Trip Overview

Destination:

Hong Kong

Duration:

7 Days

Budget:

Moderate (800-1,200 HKD / 100-150 USD per day)

Travel Style:

Solo, focusing on a mix of iconic sights, local culture, food, and affordable experiences

Transportation:

Primarily public transport (MTR, buses, ferries) using an Octopus Card for convenience

Preparation Tips

Octopus Card:

Purchase this rechargeable card (around 100 HKD with 50 HKD deposit) for seamless travel on MTR, buses, trams, ferries, and even some shops. It's a must for getting around efficiently [2].

Accommodation:

I've suggested staying in a budget-friendly hotel or hostel in a central area like Tsim Sha Tsui or Causeway Bay for easy access to attractions. Estimated cost: 400-600 HKD (50-75 USD) per night.

Ask a new question...

Ask 3.0

Figure 7: System-generated response to a user query for a solo 7-day Hong Kong itinerary, combining web-sourced inspiration and a structured travel plan.

When a user enters a natural-language query such as “Create a 7-day solo travel itinerary in Hong Kong on a moderate budget,” the system responds with a two-phase output (Figure 7). On the left, the interface presents a curated selection of external sources and visual materials, including travel blog entries, guide suggestions, and image carousels to inspire the user. On the right, the assistant delivers a clearly formatted response containing practical planning elements: destination summary, budget estimation in both HKD and USD, transportation methods, accommodation suggestions, and travel tips. The output is designed to support solo travelers with cost-conscious recommendations and easily actionable information.

Location-Based Discovery – Integrating Maps and Contextual Guidance

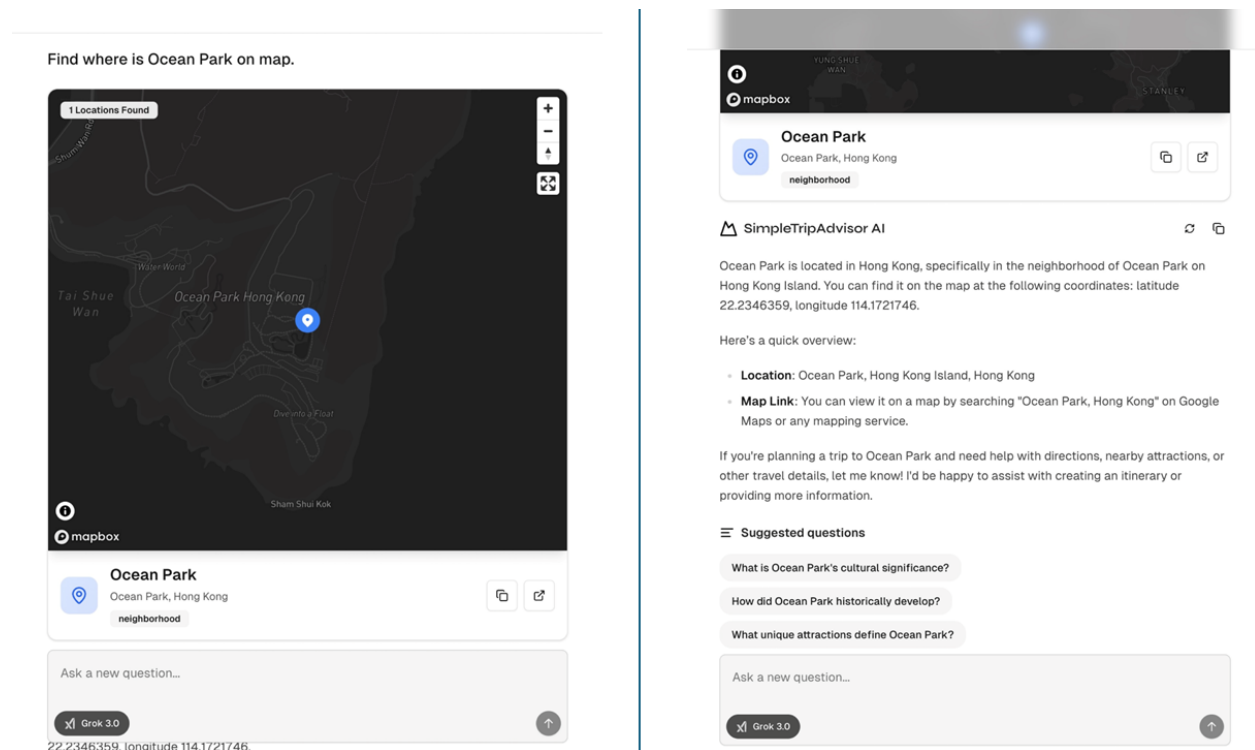


Figure 8: Map and assistant-generated information for Ocean Park in Hong Kong, combining geolocation, a brief overview, and follow-up question suggestions.

To support geographic awareness during trip planning, the system responds to natural-language queries with integrated map visualization and contextual information. In response to the input “Find where Ocean Park is on the map,” the assistant returns an interactive dark-themed map with a pinpointed marker for Ocean Park on Hong Kong Island (Figure 8). Alongside the map, the interface provides a concise location overview, including neighborhood information and geographic coordinates (latitude 22.23, longitude 114.17). To enhance engagement, the system also presents follow-up questions related to the destination’s cultural significance, history, and attractions, guiding users toward deeper exploration or itinerary refinement.

Flight Planning Assistance – From Availability Search to Real-Time Tracking

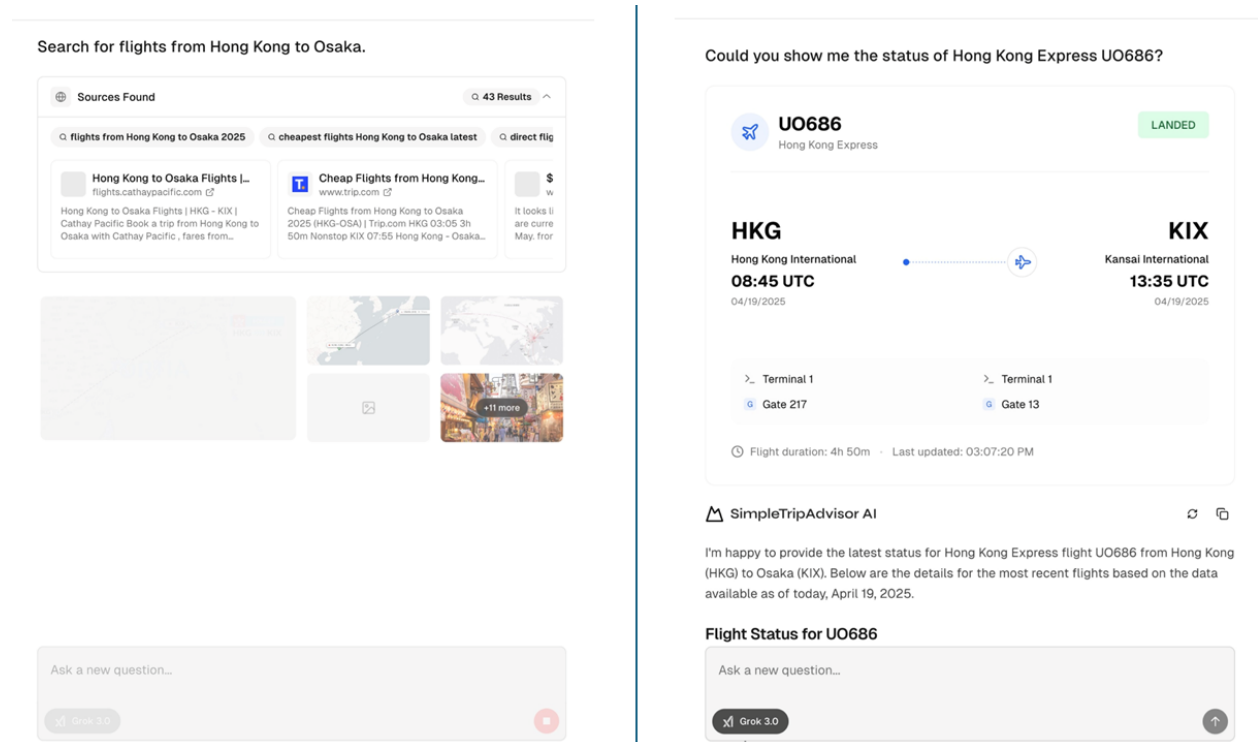


Figure 9: Flight search and tracking interaction. Left: query results for flights between Hong Kong and Osaka. Right: real-time flight status for UO686, including departure/arrival details and gate information.

Air travel is a central part of trip planning, and the system offers comprehensive support throughout the process. When a user searches for “flights from Hong Kong to Osaka,” the assistant retrieves a curated list of flight options from third-party sources, including airline names, durations, and fare estimates (Figure 9, left). Upon follow-up with a specific query such as “What is the status of Hong Kong Express UO686?”, the assistant returns real-time flight tracking information, displaying terminal, gate, departure and arrival times in UTC, and flight duration (Figure 9, right).

Next Steps

If you have a specific date in mind for tracking UO686 or need assistance with booking or other flight details, please let me know:

- Are you looking for the status on a future date?
- Do you need help setting up alerts for this flight?
- Would you like information on booking this flight or alternative options?

I'm here to assist with any further travel planning needs!

≡ Suggested questions

What is Hong Kong Express's history?

How does Osaka culture influence visitors?

What are notable landmarks near Kansai Airport?

Ask a new question...

 Grok 3.0



Figure 10: Proactive follow-up from the assistant, offering next steps such as alert setup, alternate travel dates, or booking assistance.

In the subsequent step (Figure 10), the system proactively suggests next actions, such as setting alerts, checking future dates, or receiving booking assistance, demonstrating conversational continuity and context-aware planning support.

Auxiliary Features – Responding to Time and Weather Queries

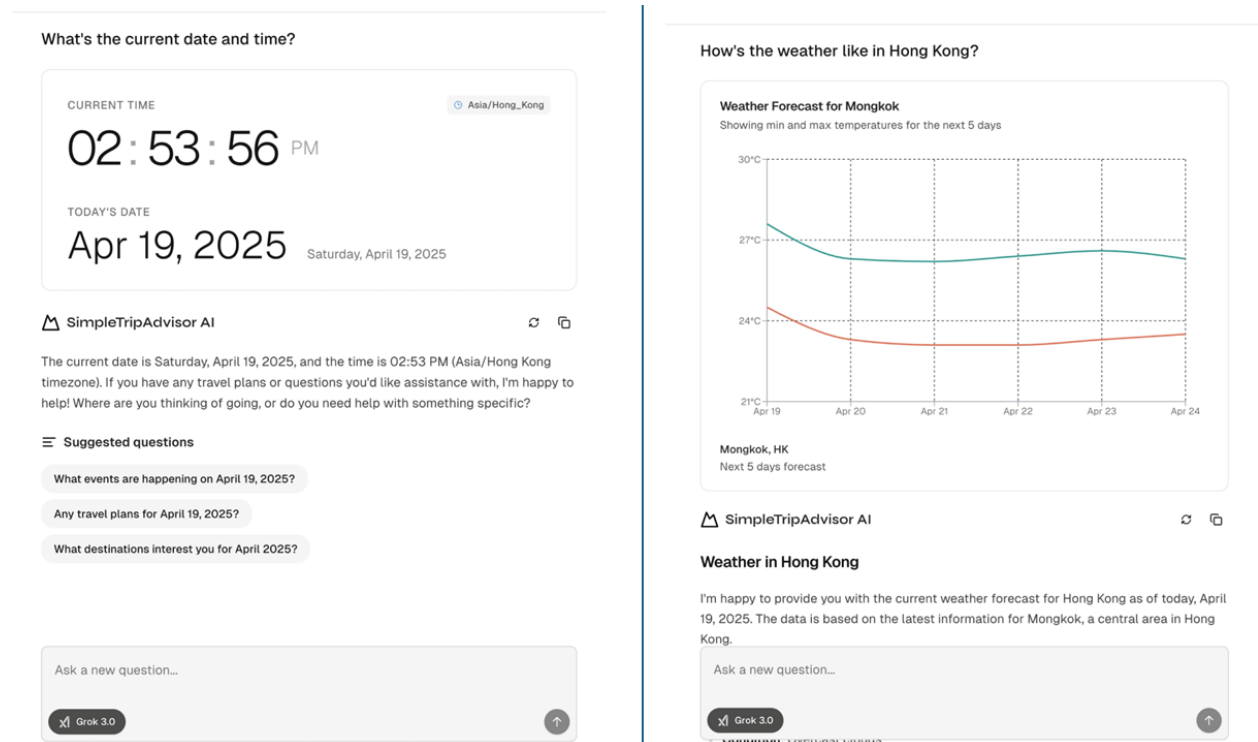


Figure 11: Responses to auxiliary user queries. Left: current date and time in the user's time zone. Right: 5-day weather forecast for Mongkok, Hong Kong, shown with temperature trends.

Beyond core itinerary planning, the system is equipped to handle a wide range of auxiliary travel-related queries. These include, but are not limited to, checking local time, retrieving weather forecasts, and finding upcoming events. Figure 11 illustrates two such examples. On the left, the assistant responds to a query about the current date and time by providing localized time zone information and a precise timestamp. On the right, it returns a 5-day weather forecast for Mongkok, Hong Kong, using a temperature trend chart to convey both minimum and maximum values. These capabilities ensure that travelers receive timely and situationally relevant information throughout their planning process.

4 Future Enhancements and Expected Challenges

This section outlines the potential future directions for enhancing the SimpleTripAdvisor AI platform, focusing on two key areas: **(1)** developing a specialized neural network-based recommendation system, and **(2)** improving the performance and contextual understanding of the conversational interface through prompt engineering techniques. Each of these areas presents both opportunities for system evolution and practical challenges that must be addressed.

4.1 Toward a Specialized Recommendation Engine

Currently, the itinerary recommendations are generated directly by an LLM based on user prompts. While this approach allows for quick deployment and flexible responses, it relies heavily on the model’s pre-trained knowledge, which may not always be up-to-date, domain-specific, or context-aware. Furthermore, no publicly available LLM has been explicitly trained for the task of travel planning. This results in limitations when it comes to recommendation precision and personalization.

To overcome these issues, the team plans to develop a dedicated neural network (NN)-based recommendation engine. This model will interpret structured user preferences extracted via the natural language interface and generate personalized itineraries accordingly. Neural collaborative filtering, particularly latent factor models, is well-established for such tasks due to their scalability and performance in capturing user-item interactions [10]. However, these models typically struggle with the cold start problem—i.e., generating recommendations for new users or items with little or no historical interaction data [11].

To mitigate this, the team plans to incorporate DropoutNet [12], which augments latent factor models with neural network layers specifically trained to improve generalization under cold-start conditions. DropoutNet enables the system to make reasonable inferences even with sparse user data by simulating missing information during training. This technique can be integrated without modifying the core architecture of the recommender system, making it a practical and effective

enhancement.

Table 1 summarizes the characteristics of different recommendation approaches considered in this project.

Table 1: Comparison of Recommendation Techniques for Travel Planning

Technique	Advantages	Limitations
Rule-Based Recommendation	Simple logic, interpretable, easy to implement	Inflexible, hard to scale or personalize
LLM-Based Generation (e.g., Gemini)	Fast prototyping; no labeled data required; flexible input handling	Trained on general text corpora; lacks adaptation to travel planning and personalized context
Neural Network-Based Collaborative Filtering	Scalable, trainable on user data, supports personalization	Requires structured datasets and feature engineering; still challenged by data sparsity

4.2 Challenges in Data Collection and Feature Engineering

Developing a recommendation system also brings considerable challenges in data preparation. One of the primary obstacles lies in designing effective input and output formats for the model. Unlike typical recommendation tasks that focus on single-item suggestions (e.g., recommending a movie), travel planning requires generating multi-step itineraries involving various categories like attractions, accommodations, transportation, and events. Standardizing and structuring such diverse data into a unified feature space is non-trivial.

A key requirement will be to tag and grade thousands of possible entities (e.g., restaurants, hotels, landmarks) with consistent feature representations, such as price level, popularity, category, and accessibility. This is a labor-intensive process, particularly for a small development team. Due to the lack of high-quality public datasets in this domain, the team also expects to rely on synthetic data generation techniques to augment the training set and improve model generalizability.

Feature engineering will also involve encoding categorical variables (e.g., city names, travel themes, accommodation types) into numerical formats while preserving their semantic relationships. Moreover, the model may suffer from the curse of dimensionality if too many features are used, leading to overfitting or degraded performance. To address this, dimensionality reduction techniques, such as Principal Component Analysis (PCA) or feature selection via regularization, may be used, alongside careful hyperparameter tuning during training to strike a balance between model complexity and predictive power.

4.3 Enhancing Prompt Quality and Contextual Understanding in LLMs

As the recommendation engine evolves into a separate component, the role of the LLM will shift toward extracting structured user preferences from natural-language input. This will require better prompt design and improved intent recognition to ensure that key information, such as destination, travel dates, group size, and budget, is accurately captured.

One promising direction is the use of intent-based prompting, where user inputs are categorized into predefined intent types (e.g., request for hotel recommendations, flight inquiries, or activity suggestions). Each intent category would trigger a tailored prompt format designed to elicit more accurate and relevant responses from the AI-driven assistant. However, this approach requires reliably identifying user intent from free-form text, which can be difficult, especially when multiple overlapping intents appear in a single message.

To tackle this, self-seeding and multi-intent self-instructing techniques have been proposed in recent research [13]. These approaches allow the LLM to recursively generate and refine intent-aware prompts by learning from its own outputs. Such methods can enhance the conversational agent’s ability to understand and disambiguate complex user inputs, ultimately leading to more intelligent, coherent, and personalized dialogue.

5 Conclusion

This report presents the development of SimpleTripAdvisor AI, an intelligent travel planning platform that integrates a natural-language interface with an LLM to deliver personalized and seamless itinerary suggestions. By combining conversational AI with modular web-based architecture, the platform enables users to plan trips through intuitive, dialogue-based interactions rather than relying on fragmented manual tools.

The current system features a web-based responsive interface and a modular backend that supports real-time dialogue, itinerary generation, map integration, flight tracking, etc. These features are delivered through a unified conversational interface, reducing the learning curve for new users and offering a more engaging user experience. While natural-language interaction has become more common, this project highlights its potential to simplify information-intensive tasks such as travel planning.

This report also outlines several areas for future enhancement. These include developing a dedicated neural network-based recommendation engine, addressing cold start and data representation challenges, and improving prompt strategies for better intent recognition. Although this combination of LLMs and task-specific models is not yet realized, it remains a key direction for future work.

In conclusion, this project offers a functional and user-friendly prototype that demonstrates how LLM-powered interfaces can reshape travel planning workflows. It also lays the groundwork for future research in combining general-purpose language models with domain-specific reasoning systems, opening possibilities for more intelligent and personalized digital experiences.

References

- [1] P. M. Addo, D. Guegan, and B. Hassani, “Credit risk analysis using machine and deep learning models,” *Risks*, vol. 6, no. 2, p. 38, 2018.
- [2] K. A. Bhavsar, J. Singla, Y. D. Al-Otaibi, O.-Y. Song, Y. B. Zikria, and A. K. Bashir, “Medical diagnosis using machine learning: a statistical review,” *Computers, Materials and Continua*, vol. 67, no. 1, pp. 107–125, 2021.
- [3] W. Tizani, “A review of trip planning systems.” 1992.
- [4] B. Pan, X. R. Li, L. G. Zhang, and W. W. Smith, “An exploratory study on the satisfaction and barriers of online trip planning to china: American college students’ experience,” *Journal of Hospitality & Leisure Marketing*, vol. 16, no. 1-2, pp. 203–226, 2008.
- [5] A. Følstad and P. B. Brandtzæg, “Chatbots and the new world of hci,” *interactions*, vol. 24, no. 4, pp. 38–42, 2017.
- [6] S. Laoyan, “What is Agile Methodology? (A Beginner’s Guide) [2024] • Asana,” 2 2024. [Online]. Available: <https://asana.com/resources/agile-methodology>
- [7] A. Analysis, “AI Model API Providers Analysis — Artificial Analysis.” [Online]. Available: https://artificialanalysis.ai/?models_selected=gpt-4o-mini%2Cgemini-2-0-flash-experimental
- [8] D. Team, “DeepSeek-R1 Release — DeepSeek API Docs.” [Online]. Available: <https://api-docs.deepseek.com/news/news250120>
- [9] A. C. Team, “Qwen API reference - Alibaba Cloud Model Studio - Alibaba Cloud Documentation Center.” [Online]. Available: <https://www.alibabacloud.com/help/en/model-studio/developer-reference/use-qwen-by-calling-api>
- [10] H. Ko, S. Lee, Y. Park, and A. Choi, “A survey of recommendation systems: recommendation models, techniques, and application fields,” *Electronics*, vol. 11, no. 1, p. 141, 2022.

- [11] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, “Addressing cold-start problem in recommendation systems,” in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008, pp. 208–211.
- [12] M. Volkovs, G. Yu, and T. Poutanen, “Dropoutnet: Addressing cold start in recommender systems,” *Advances in neural information processing systems*, vol. 30, 2017.
- [13] A. Askari, R. Petcu, C. Meng, M. Aliannejadi, A. Abolghasemi, E. Kanoulas, and S. Verberne, “Self-seeding and multi-intent self-instructing llms for generating intent-aware information-seeking dialogs,” *arXiv preprint arXiv:2402.11633*, 2024.