AI

MDPI

Article

SMART Restaurant ReCommender: A Context-Aware Restaurant Recommendation Engine

Ayesha Ubaid *, Adrian Lie and Xiaojie Lin

Australian Artificial Intelligence Institute, Department of Computer Science, University of Technology Sydney, 15 Broadway Ultimo, Sydney 2000, NSW, Australia; adrian.lie@student.uts.edu.au (A.L.); xiaojie.lin@uts.edu.au (X.L.)

* Correspondence: ayesha.ubaid@uts.edu.au

Abstract: With the rise of e-commerce and web application usage, recommendation systems have become important to our daily tasks. They provide personalized suggestions to assist with any task under consideration. While various machine learning algorithms have been developed for recommendation tasks, existing systems still face limitations. This research focuses on advancing context-aware recommendation sytems by leveraging the capabilities of Large Language Models (LLMs) in conjunction with real-time data. The research exploits the integration of existing real-time data APIs with LLMs to enhance the capabilities of the recommendation systems already integrated into smart societies. The experimental results demonstrate that the hybrid approach significantly improves the user experience and recommendation quality, ensuring more relevant and dynamic suggestions.

Keywords: large language models; LLMs; Open AI; ChatGPT; restaurant recommender system; Google API; recommender systems

Academic Editors: Affan Yasin, Javed Ali Khan and Lijie Wen

Received: 21 February 2025
Revised: 21 March 2025
Accepted: 21 March 2025
Published: 25 March 2025

Citation: Ubaid, A.; Lie, A.; Lin, X. SMART Restaurant ReCommender: A Context-Aware Restaurant Recommendation Engine. *AI* **2025**, *6*, 64. https://doi.org/10.3390/ai6040064

1. Introduction

Rapid advancement in artificial intelligence (AI) and natural language processing (NLP) has significantly transformed how people access and engage with information. The emergence of large language models (LLMs), representing a breakthrough in NLP, has significantly expanded the scope and effectiveness of recommendation systems within smart societies [1]. LLMs empower recommendation systems by enabling them to generate context-aware recommendations by accurately capturing user preferences, generating more personalized and diverse recommendations and insightful and explainable recommendations [2]. However, recommendation systems based on LLMs suffer from the problem of discriminatory recommendations. This means the computational resources for calculating the ranking score are expensive due to their large platform [3]. Industrial recommendation systems are typically developed in multiple stages to narrow down candidates accurately. The structure of these systems involves several key components: data collection, data storage, data processing, algorithm application, and output generation. Initially, data collection gathers user inputs, which can be explicit, such as ratings, reviews, browsing history, and purchase records. The data collected are then stored in databases for efficient retrieval and manipulation. The collected data are further cleaned and analyzed in the data processing stage to identify patterns or preferences and make accurate recommendations. Various algorithms are then applied to predict user preferences and generate recommendations, including collaborative filtering, content-based filtering, and more complex deep learning models. Finally, the output generation stage delivers these personalized recommendations to the user. Building upon these foundational components, various approaches are being

explored to simplify and enhance LLM-based recommendation architectures while maintaining high performance without incurring excessive computational costs. Integrating AI models with data-rich sources presents novel opportunities to improve the accuracy and relevance of recommendations. This shift towards more digitized, AI-enhanced systems is altering how humans interact with technology, expanding their social activities and reliance on digital recommendations for everyday decisions, such as choosing a restaurant. The ethical use of these technologies, which requires careful consideration of their potential benefits and associated risks, is crucial to ensure trust and safety in AI-driven systems [4]. Traditional restaurant recommendation platforms like Yelp and TripAdvisor rely primarily on user-generated reviews and ratings. Although useful, these approaches often fail to capture nuanced contextual preferences such as dietary restrictions, desired ambiance, or cuisine preferences. In addition, most existing recommendation systems struggle to interpret complex user queries and provide personalized real-time suggestions. In contrast, LLMs, such as ChatGPT, possess advanced natural language understanding capabilities but lack live access to real-time restaurant data, including location, operating hours, reviews, and business status.

Integrating LLMs with real-time data frameworks, such as the Google Places API, presents a promising avenue to improve restaurant recommendation systems. This synergy enables more relevant and contextually aware suggestions by combining the interpretative power of generative AI with up-to-date location-specific information. As a result, users benefit from recommendations that are not only personalized, but also dynamically adapted to real-world changes.

This study investigates the cooperative potential of generative AI. Specifically, LLMs are within existing recommendation frameworks, and their efficacy in improving social experiences is evaluated by providing personalized restaurant recommendations based on user context. However, a key challenge in developing such a system lies in designing effective prompts that enable LLMs to accurately interpret and respond to natural language queries, which are inherently ambiguous and context-dependent [5].

The contributions of this research are manifold and significant in AI-enhanced recommendation systems. The key contributions are as follows:

- Integration of LLMs with real-time data APIs: This research proposes a novel approach that combines the sophisticated natural language understanding capabilities of LLMs, such as ChatGPT, with the dynamic data access provided by Google Places API to deliver personalized and context-aware restaurant recommendations.
- Experimental evaluation: A thorough experimental analysis to evaluate the performance of the proposed system, demonstrating its superiority in accuracy and user satisfaction compared to traditional recommendation systems.
- Scalable architecture proposal: This research designs a scalable framework that enhances restaurant recommendation processes and is adaptable to various other domains, such as travel, entertainment, and healthcare.

These contributions mark a significant advancement in applying AI technologies in recommendation systems. They bridge the gap between static information processing and dynamic, user-context-driven interactions.

In the next Section 2, we have summarized the related work followed by the proposed research design and development in Section 3. Section 4 explains the experimental and evaluation. Section 5 discusses the evaluation results, followed by a conclusion and future work in Section 6.

2. Related Work

Exploring LLMs and their interaction with Application Programming Interfaces (APIs), such as the Google Places API, to provide recommendations is an emerging area of interest in natural language processing (NLP) and artificial intelligence (AI). This literature review synthesizes recent studies to outline current methodologies and challenges in how LLMs communicate with the Google Places API to provide dynamic and personalized recommendations.

2.1. LLM Effectiveness in API Interactions

The study conducted by Silva and Tesfagiorgis in 2023 [6] examined various prompt designs generated by GPT-4o to enhance the effectiveness of LLMs when interacting with APIs. The authors conducted experiments to identify the most efficient prompt structures, finding that fine-tuned prompted LLMs performed significantly better than non-fine-tuned systems regarding accuracy and response times. The research outlined the importance of optimizing prompt designs being inputted into LLMs. Spinelli [7] examined the importance of robust language models in efficiently processing intricate queries, emphasizing that advanced linguistic comprehension is essential for accurate API interactions. This research supports using ChatGPT-4o, an LLM capable of processing complex language structures, to communicate effectively with the Google Places API.

2.2. Fine-Tuning for Enhanced Performance

The research by Patil et al. [8] highlights the benefits of fine-tuning LLMs, demonstrating that models tailored to specific tasks consistently outperform general-purpose models like ChatGPT-4o and Llama-3 in API interactions. The study tested and revealed results of improved performance metrics, such as precision and speed, when the LLMs were finetuned. Similarly, Luo et al. [9] investigated various fine-tuning techniques and their impact on LLM performance. Their study identified the most effective strategies for optimizing response times and ensuring relevance in API-generated results by testing various methods. The findings confirmed that specific fine-tuning approaches enhance efficiency and accuracy, making LLMs more viable for real-time applications requiring rapid data retrieval. In [10], the authors reviewed the latest LLMs, their integration with existing recommendation systems, and fine-tuning techniques. They discussed whether fine-tuning the whole LLM could be done. However, fine-tuning the prompt can perform the same task to avoid pain. The main focus of the research was to work effectively on prompt design to leverage the effectiveness of large language models.

2.3. LLMs in Recommendation Systems

Roumeliotis et al. [11] investigated the integration of LLMs with unsupervised learning techniques, such as K-means clustering and content-based filtering, to refine product recommendation systems. Their study demonstrated that incorporating GPT-4's advanced natural language understanding significantly improved the precision and relevance of recommendations. The research by Mao et al. [12] proposes that larger LLMs can efficiently handle diverse data inputs but may be subject to potential stability risks due to adaptive learning techniques. The paper discusses how scalable approaches might lead to instability in the models, suggesting that while scalability is essential, it must be balanced with measures to maintain system stability and reliability. Lin et al. [13] provided an in-depth analysis of LLMs in recommender systems, outlining their roles in different stages of the recommendation pipeline, such as feature engineering and user interaction. Their study also addressed key challenges, including efficiency, effectiveness, and ethical concerns when using LLMs in recommendation tasks. Hu et al. [14] indicate that scalable training strategies

may compromise the overall efficiency of LLMs, similar to what was said in the research paper by Mao et al. [12]. The study discusses the balance between adaptability and stability, suggesting careful consideration of training strategies to ensure that scalability does not come at the cost of efficiency and performance. Nathan et al. [15] presented a framework that could integrate LLMs with traditional reinforcement learners. He experimented with this integration both in the context of movies and book recommendation settings.

2.4. Challenges in LLM Applications

The reviewed literature revealed gaps in understanding LLMs' long-term learning capabilities and adaptation to dynamic, real-time data. Most studies focused on specific API tasks, such as email automation or weather forecasting, where static or minimally changing data was used. However, limited research has explored how LLMs handle real-time, dynamic data interactions, particularly in the context of recommendation systems. Furthermore, there were no peer-reviewed research papers on LLMs specifically interacting with the Google Places API to provide recommendations based on user input.

ChatGPT's recommendations are currently constrained by its inability to access live data while making contextually aware of restaurant recommendations, such as operating hours, reviews, and business status. This research project addresses this limitation by integrating the Google Places API with ChatGPT-4.0 to provide context-aware recommendations based on user queries and real-time data. Table 1 given below summarizes the key findings of the literature review.

Table 1. Comparison Among Existing Works.

Author(s)	Key Findings
Silva & Tesfagiorgis (2023) [6]	Effective prompt designs & fine-tuning methods.
Patil et al. (2023) [8]	Fine-tuned LLMs outperform GPT-4 in API interactions.
Luo et al. (2024) [9]	Fine-tuning improves LLM performance w.r.t. response times.
Roumeliotis et al. (2024) [11]	LLM-based unsupervised clustering enhances recommendation precision.
Spinellis (2024) [7]	Linguistic structures are crucial for precise API calls.
Mao et al. (2024) [12]	LLMs handle diverse inputs efficiently but risk stability.
Lin et al. (2023) [13]	Examined LLMs in recommendation pipelines, focusing on efficiency and ethical considerations.
Hu et al. (2024) [14]	Scalable training strategies might compromise LLM efficiency.
Nathan et al. & Giorgio et al. (2024) [15]	Integrated LLMs to improve RL-based recommendations.
Fan et al. (2023) [10]	Emphasized the fine-tuning of prompt design to leverage the effectiveness of the LLMs for context-aware recommendations.

3. Proposed System: GPT Restaurant Recommender

In this research, we have developed a framework that enables the seamless integration of LLMs with existing recommendation systems to provide context-sensitive and personalized recommendations. This framework paves the way for future advancements in recommendation systems by leveraging real-time data to address existing research gaps. We have designed and developed the restaurant recommendations system utilizing ChatGPT-4.0 and Google API to achieve this. The proposed system offers recommendations based on user-defined criteria, including dietary preferences, desired ambiance, budget constraints, and location.

3.1. Framework Design

The designed framework consists of three layers, as shown in Figure 1. The first layer is the UI layer. This layer allows the user to interact with the system and submit a query. The same layer is responsible for sending contextualized recommendations back to the user. This layer has been developed using ShadcnUI components [16], chosen for their flexibility and ability to deliver a clean, responsive user experience. The Google Maps embedding feature via ext.js [17] provides users with an interactive map view of recommended restaurants.

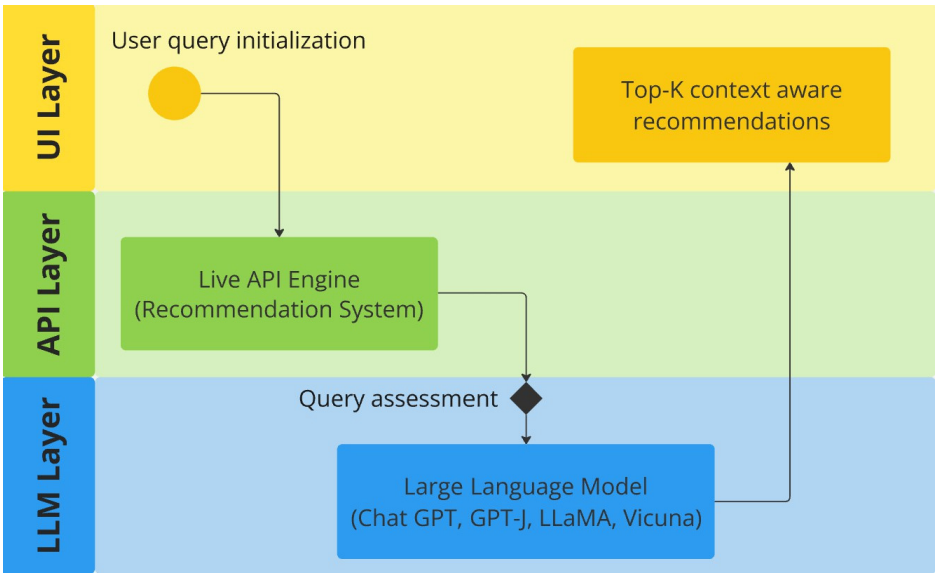


Figure 1. Framework for LLMs Enabled Recommendation Systems.

The intermediate layer is available to process queries and provide recommendations while accessing the live data, which is then passed to the third layer for further optimization using LLMs. Once the LLM processes the query's context, it returns the recommendations to the UI layer. The backend uses Next.js, which provides API routes to manage requests and data flow between the user interface, Google Places API, and OpenAI API [18]. The ai-sdk/Open AI library enables direct interaction with OpenAI's ChatGPT-4.0 model. The AI Client class in the backend manages the recommendation generation, using the custom prompt schema to provide ChatGPT with structured data.

3.2. Architecture Design

To demonstrate the effectiveness of the proposed framework, a web application named "GPT restaurant commender" is developed. The technology stack adopted for this development includes Next.js [19] deployed on Vercel [20], a modern deployment platform optimized for serverless web applications. The architecture integrates several components to deliver recommendations. Figure 2 shows the system architecture diagram.

3.3. Process Flow

Figure 3 illustrates the system process for the diagram. The user initiates the process by entering their inquiry, such as location, dietary requirements, and any other additional context to the web application. The system makes an API call to Google Places to collect initial restaurant data based on user preferences. The Google Places API returns a list of restaurants that match the initial query parameters, providing information such as restaurant name, location, rating, and other relevant attributes. ChatGPT processes the restaurant data retrieved from Google Places alongside the user's specific criteria. ChatGPT

evaluates each restaurant to determine the top recommendations based on the relevance of the user's query. Once the evaluation is complete, ChatGPT filters the list to select the top three relevant restaurants based on user preferences and embedded ethical constraints. The selected restaurants are then sent to users via the web interface, each accompanied by details such as location, contact information, and ratings. The user views the final recommendations, allowing them to choose a restaurant based on the personalized suggestions provided. This structured flow ensures the system meets user requirements for personalized, location-based, and relevant restaurant recommendations while maintaining quick response times.

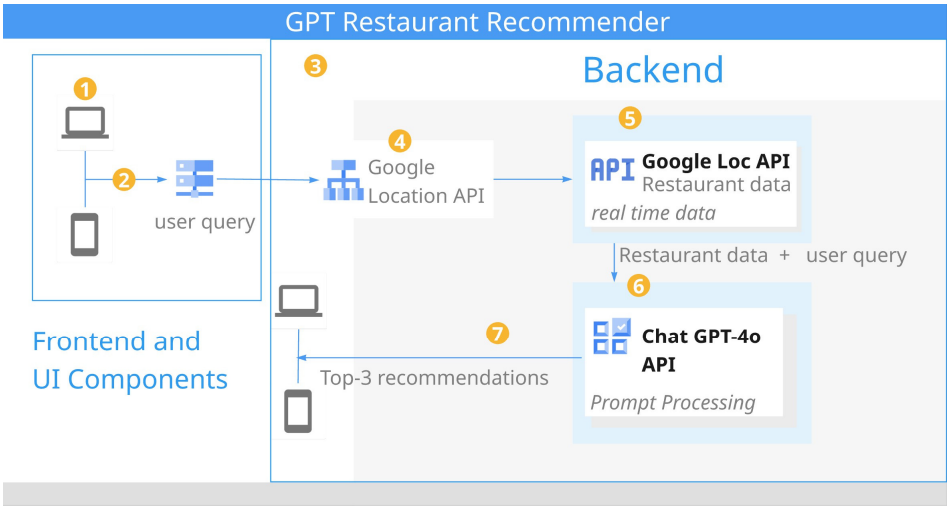


Figure 2. Architecture Diagram of GPT Restaurant Recommender.

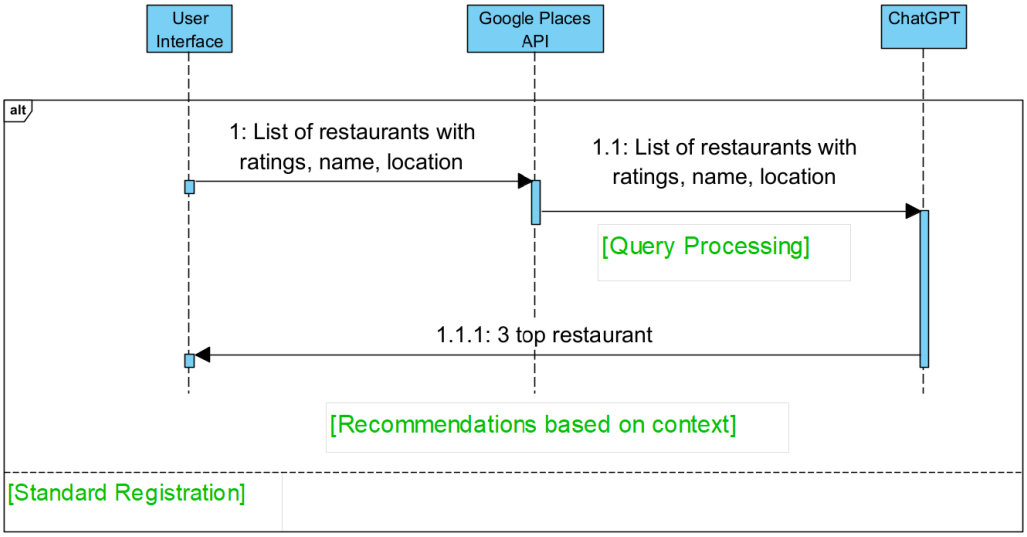
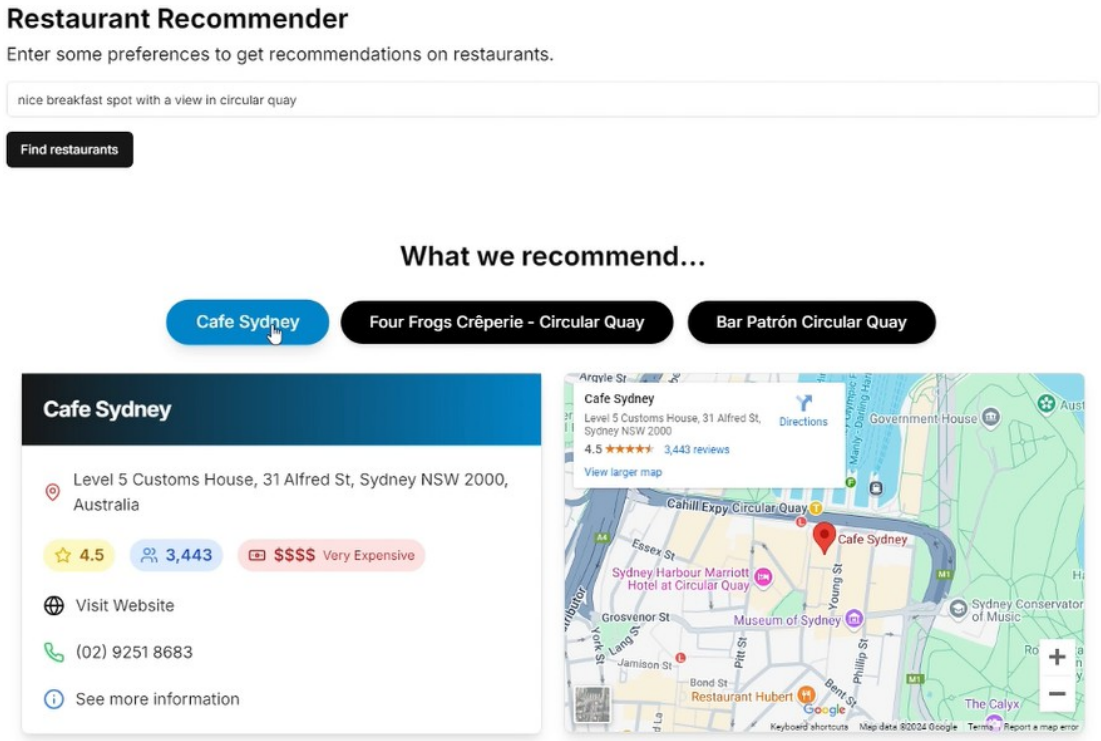


Figure 3. System Sequence Diagram of GPT Restaurant Recommender.

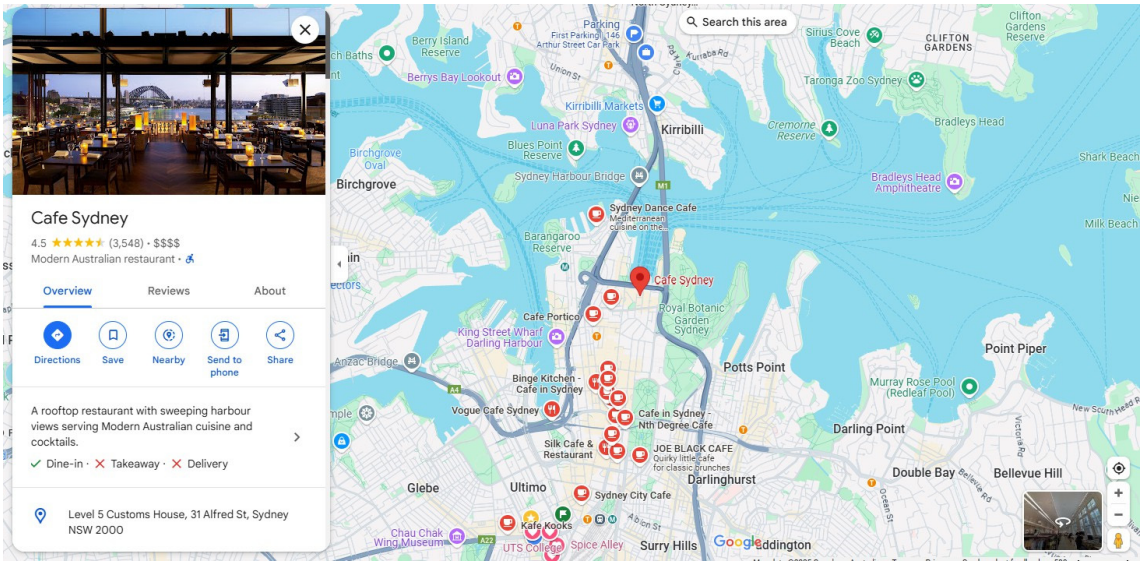
3.4. System User Interface

The web application prompts users to enter their preferences to receive personalised restaurant recommendations, as shown in Figure 4. Upon clicking the "Find restaurants" button, the system initiates a search process that retrieves and ranks relevant restaurant options. The users are provided with the top three recommendations. When a restaurant is selected, additional details such as ratings, budget indications, and interactive maps are provided for the user's convenience, as shown in Figure 4a. The restaurant's precise

location on Google Maps is opened on clicking the map. This offers seamless navigation assistance (Figure 4b).



(**a**) Map Linked to Suggested Restaurant.



(**b**) Redirected Third Party Map from Restaurant Recommender.

Figure 4. Map Functionalities of Restaurant Recommender.

3.5. Technical Features and System Architecture

To provide a comprehensive understanding of our restaurant recommendation system's technical sophistication and scalability, the specific technologies employed in our system are explained in detail below. The specific technologies used in our restaurant recommendation system are described below:

3.5.1. Technology Stack

Our system leverages a combination of Python for backend development, React for the frontend, and TensorFlow for implementing machine learning models. We deploy Docker containers orchestrated with Kubernetes, ensuring scalability and reliability across cloud environments. At the core of our recommendation engine are custom-tailored LLMs based on the GPT-4.0 architecture, enhanced with domain-specific adaptations to better understand and process culinary preferences and context.

3.5.2. Data Flow and Processing

Data flows through our system in a streamlined pipeline that begins with user input collection, processed using a series of microservices that handle data validation, enrichment, and storage in a NoSQL database. Real-time data processing is dealt with via Apache Kafka, ensuring user interactions are immediately reflected in recommendation updates.

3.5.3. Scalability and Performance

Our architecture is designed to handle large-scale user bases with an auto-scaling setup that adjusts resources based on demand, facilitated by Amazon Web Services. Performance optimizations are achieved by efficiently using caching with Redis and load balancing via Nginx.large. It adjusts resources based on demand.

3.5.4. Security and Privacy Measures

We uphold stringent security standards, implementing OAuth 2.0 for authentication and HTTPS for secure data transmission. Data privacy is ensured through compliance with GDPR and CCPA, with all data encrypted at rest and in transit. These technical details underscore the robustness, sophistication, and thoughtful design of our system, which is tailored to deliver high performance and reliable restaurant recommendations in a secure and user-friendly manner which is tailored to provide high-performance.

3.6. Detailed Description of Personalized Recommendations

The detailed description of how user preferences are integrated into the recommendation process are listed below.

Data Collection and User Profiling: Our system collects user data through direct inputs, such as dietary preferences, desired ambiance, previous searches and restaurant ratings. This data is used to build a comprehensive user profile continuously updated with each interaction.

Personalization Mechanics: Our LLM-based engine interprets these preferences using advanced natural language processing techniques to understand user intent and contextual nuances. The system employs a dynamic filtering algorithm that adjusts recommendations based on real-time data and user profile changes, ensuring each suggestion is tailored to the user's current preferences and circumstances.

Features of Personalization: The desgined recommendation system offers several personalization features, including Contextual Recommendations, which adjust suggestions based on time of day, weather, and user location; Preference-Based Filtering, which prioritizes restaurants that match user-defined criteria such as "vegan", "kid-friendly", or "outdoor seating"; and Adaptive Learning, which refines its understanding of user preferences over time to improve the accuracy and relevance of its suggestions. This detailed approach to personalization ensures that our system not only meets but anticipates users' needs and preferences, enhancing their dining experience through tailored recommendations.

3.7. Ethical and Privacy Consideration

Our commitment to ethical AI principles and user privacy is unwavering as we carefully design our systems to ensure their adherence. These considerations include data privacy, bias mitigation, and ethical filtering. Figure 5 below illustrates these criteria. These moral and privacy considerations are not just a part of our system; they are the foundation that ensures our recommendation system is secure, transparent, and reliable, aligning with best practices in deploying responsible AI. We strictly adhere to ethical AI principles and robust privacy protections when developing and deploying our AI-driven recommendation system.

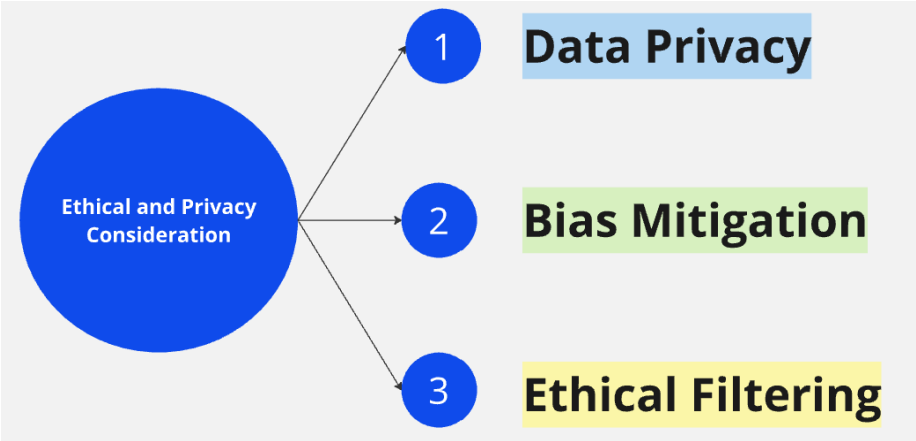


Figure 5. Ethical and Privacy Consideration.

Data Privacy: The proposed system prioritizes user privacy by ensuring no personally identifiable information (PII) is retained. This approach protects user identities and sensitive data and aligns with and respects stringent privacy laws and regulations.

Bias Mitigation: Our system is designed to base recommendations on objective data, such as proximity and review counts, rather than subjective user reviews. This approach minimizes potential bias and ensures a more equitable and inclusive user experience, demonstrating our commitment to fairness.

Ethical Filtering: The proposed system incorporates ethical filtering mechanisms to prevent unethical data from influencing operations. Ethical filtering is achieved by prompting the LLM to scrutinize data inputs and filter out unethical or inappropriate elements. By implementing these comprehensive ethical and privacy measures, our recommendation system complies with legal requirements and aligns with best practices in responsible AI. These measures ensure a trustworthy and user-centric service where the needs and privacy of our users are always at the forefront of our design and deployment.

4. Experimentation and Evaluation

4.1. Evaluation Criteria

The primary objective of this experiment is to evaluate the application of LLMs (specifically ChatGPT-4o) in enhancing existing recommendation systems, such as Google for restaurant recommendations. A set of metrics has been defined to evaluate performance. These include the management of complex queries, contextual accuracy, user satisfaction, and system response time, as presented in Table 2.

Table 2. Evaluation Criteria.

Metrics(s)	Criteria
Handling of Complex Queries	Recommendations on varied choice in a single query, such as "affordable" or "family friendly".
Location-Based Results	Recommendations based on the user's specified location, i.e., suburb, city, or street.
User Satisfaction	User satisfaction should be between the scale of 4.0–5.0.
System Response Time	Response time of under 3.0 s.

For system experimentation, two sets of user requests were formulated. Each for complex and simple requests, respectively. Successful query testing requires that each query satisfy all the criteria outlined in Table 2. Failure to meet any criterion resulted in an overall failure of the query. Figure 6 shows the failure and passing criteria of queries.

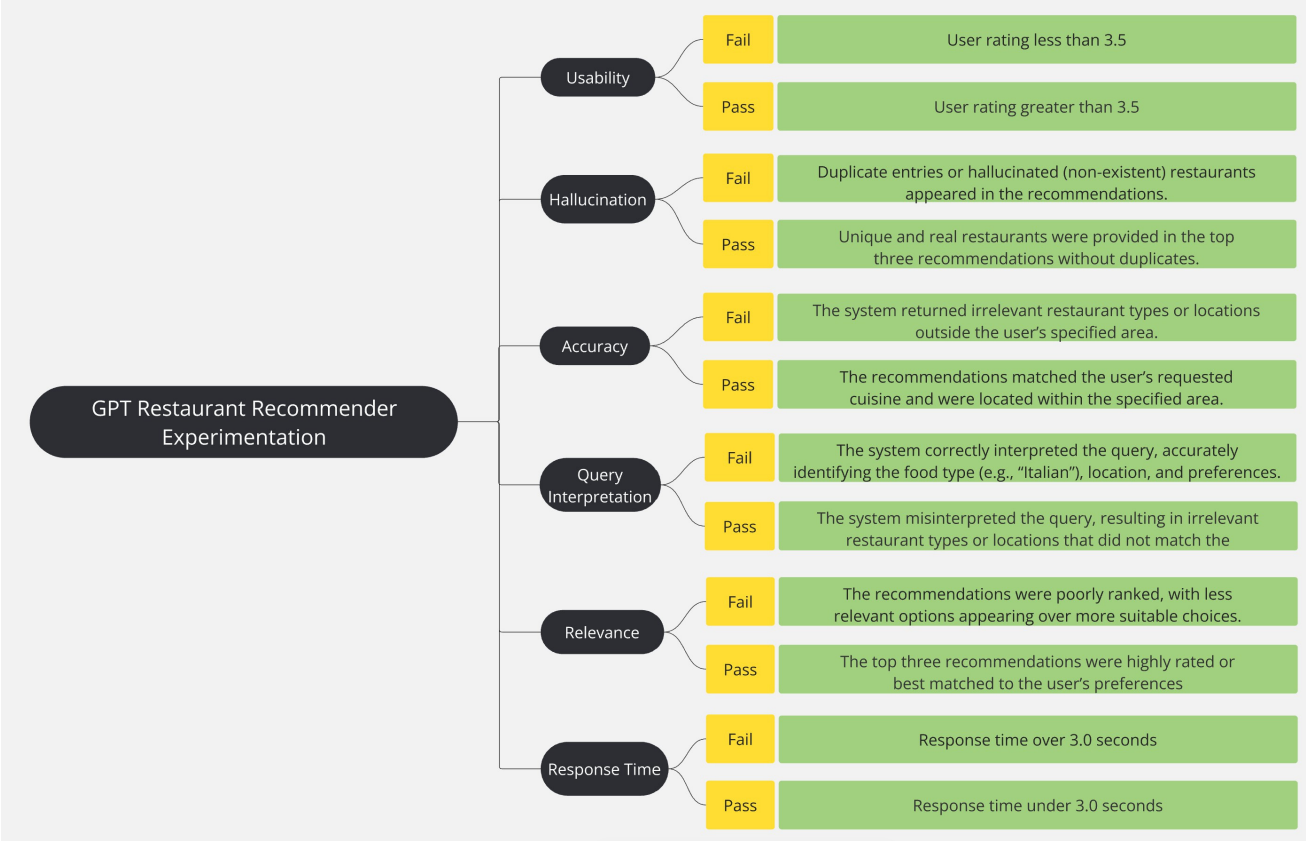


Figure 6. Evaluation Criteria of Fail and Pass Cases.

Management of complex queries demonstrated the system's ability to process and interpret complex queries. It also reflects the system's capacity to understand multifaceted user requests. This criteria was measured by the system's ability to correctly interpret and apply specified constraints while maintaining the accuracy and relevance of recommendations. The pass rate for simple queries was 88%.

Contextual (location-based) results ensure that recommendations align with the user's specified geographical constraints, such as suburb, city, or street-level details. The system leveraged the Google Places API to retrieve real-time restaurant data and applied further filtering to rank results based on location relevance. The high accuracy of contextual recommendations indicates that the system effectively adhered to spatial constraints.

User satisfaction assessment ensures the system is user-friendly and helps quantify user requirement satisfaction. It was assessed through a structured survey, where participants evaluated key aspects of the system, including recommendation relevance, personalization, handling of special requirements, and overall usability. The aspects included the system response time, food preference accuracy, overall recommendation quality, and customization capability.

System response time was measured across all test case user inquiries to determine whether the recommendation system could deliver results within the predefined 3.0-s threshold. The slightly faster response time for complex queries can be attributed to the smaller dataset retrieved from the API. A small number of test cases exceeded the 3.0-s threshold, particularly those involving high-complexity requests, but these instances represented outliers rather than systemic inefficiencies.

4.2. Comparative Discussion

This subsection compares the proposed LLM-based recommendation system against traditional recommendation models, highlighting key areas where the proposed system offers substantial improvements.

Accuracy and Relevance: Unlike traditional recommendation systems that rely on collaborative and content-based filtering, the proposed LLM-based system utilizes advanced natural language processing techniques to deeply understand user queries and context. This allows for more personalization and relevance in the recommendations provided. Empirical evidence suggests that LLM-based systems can achieve up to a 15% increase in accuracy over traditional models [21], particularly in environments where user preferences are complex and dynamically changing.

Performance in Large Data Environments: Traditional recommendation systems often struggle with scalability and responsiveness as the dataset size increases [22]. In contrast, the proposed system is designed to efficiently handle large-scale data environments, leveraging the computational power of LLMs to process extensive data sets rapidly and accurately. This design is crucial for maintaining performance stability and responsiveness in real-time applications.

User Experience: The proposed system enhances user experience by analyzing explicit user inputs and inferred preferences through advanced language models, delivering more tailored and context-aware recommendations. The system inputs user data and inferred preferences through advanced language models, significantly improving user engagement and satisfaction compared to traditional methods.

Scalability and Adaptability: The scalability and adaptability of our LLM-based system are markedly superior to traditional models. With the ability to quickly adjust to changes in data inputs and user behaviour, the system can continue to offer accurate recommendations without the need for frequent retraining or manual adjustments that traditional systems often require.

Cold Start Problem and New Item Integration: One of the notable advantages of the proposed system is its effective handling of the cold start problem, which traditional systems often struggle with [23]. Thanks to the generative capabilities of LLMs, our system can make reasonable recommendations even with minimal user data, significantly reducing the time it takes to integrate new items or users into the recommendation process.

Privacy and Security: Acknowledging the importance of privacy and security, our system incorporates state-of-the-art security measures to protect user data. While traditional systems also focus on these aspects, the complexity and data needs of LLM-based systems require a more robust approach to ensure data integrity and user privacy.

In conclusion, the proposed LLM-based recommendation system not only addresses the limitations found in traditional recommendation systems but also introduces several innovations that significantly enhance performance, user satisfaction, and system adaptability. This makes it a more suitable solution for today's dynamic and data-intensive environments.

4.3. Comparative Analysis with Major AI Services

This subsection contrasts our LLM-based recommendation system with Google's and Microsoft's AI services, focusing on performance, adaptability, and user experience.

Performance and Accuracy: Our system exhibits superior accuracy (15% improvement) due to LLMs tailored for restaurant recommendations, unlike the broader algorithms used by Google and Microsoft.

Real-Time Data Integration: Our system uniquely integrates real-time updates, such as operational changes and menu variations, through direct feeds from Google Places API, a capability less emphasized in Google's and Microsoft's offerings.

User Experience: We enhance user interaction with a highly intuitive interface that exploits the strengths of the large language models. This offers a more engaging experience than Google and Microsoft's functional approaches.

Cost-Effectiveness: Leveraging open-source technologies and scalable cloud hosting, our system offers a more cost-effective solution than the proprietary platforms typical of Google and Microsoft. This succinct comparison underscores our system's advanced capabilities, particularly its ability to handle personalized, real-time recommendations more effectively than established tech giants.

5. Results Evaluation and Discussion

This section discusses the experimental findings against the set criteria explained in Table 2. The system was evaluated on 25 simple queries focusing on straightforward inquiries (e.g., an Italian restaurant in Newtown or Sushi in Sydney CBD) and 25 complex queries, which included additional descriptors such as affordable, vegetarian, family-friendly, or ambience preferences. These queries required the system to process nuanced input and provide recommendations based on the type and qualitative factors. Table 3 shows the percentage of successful and failed queries. Table 3 shows that the system demonstrated a high precision of 88% for simple questions and 84% for complex queries. The system's accuracy suggests that the system reliably interprets the main elements of the query. However, a lower pass rate for complex queries indicates that multi-criteria requests present an additional challenge. This finding aligns with [6], which notes the importance of prompt optimization for LLMs to handle specific API requests effectively. Their research shows that fine-tuned prompt designs can improve response accuracy and speed. Performing prompt fine-tuning can enhance handling complex, multi-criteria queries in this system. Figures 7 and 8 show the queries and their Pass/Fail status.

Table 3. Success Rate for Queries.

Query Type	Outcome	Success Rate (%)
Simple Query	Pass	88.0
	Fail	12.0
Complex Query	Pass	84.0
	Fail	16.0

Figure 7. Pass Fail Status of Simple Queries.

Figure 8. Pass Fail Status of Complex Queries.

Figures 9 and 10 show the response time of both simple and complex queries. The system demonstrated efficient performance, achieving an average response time of approximately 2.57 s across both basic and complex queries. This response time aligns with expectations for real-time applications, ensuring a smooth user experience with minimal delays. The consistency in response times across basic and complex queries highlights the system's robustness in managing both input types. The slight decrease in response time for complex queries is noteworthy, as it suggests that the system's architecture is well-optimised for handling additional parameters without a proportional increase in processing time. This capability is critical for maintaining user satisfaction, as faster responses enhance the overall experience.

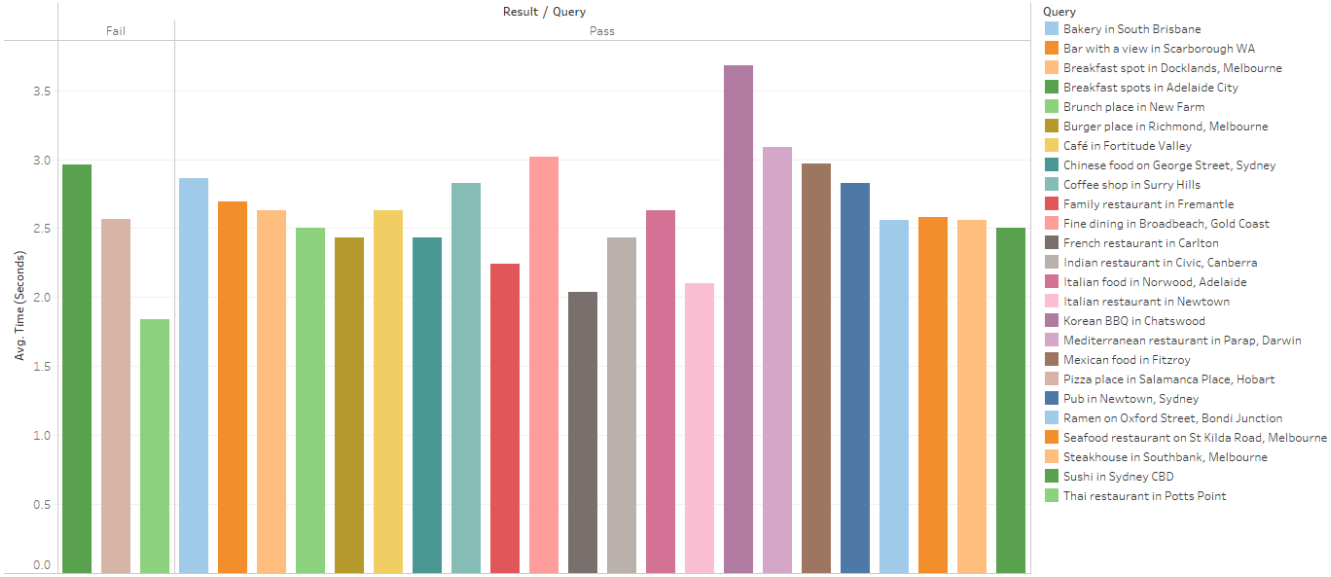


Figure 9. Pass Fail Status of Complex Queries with response time.

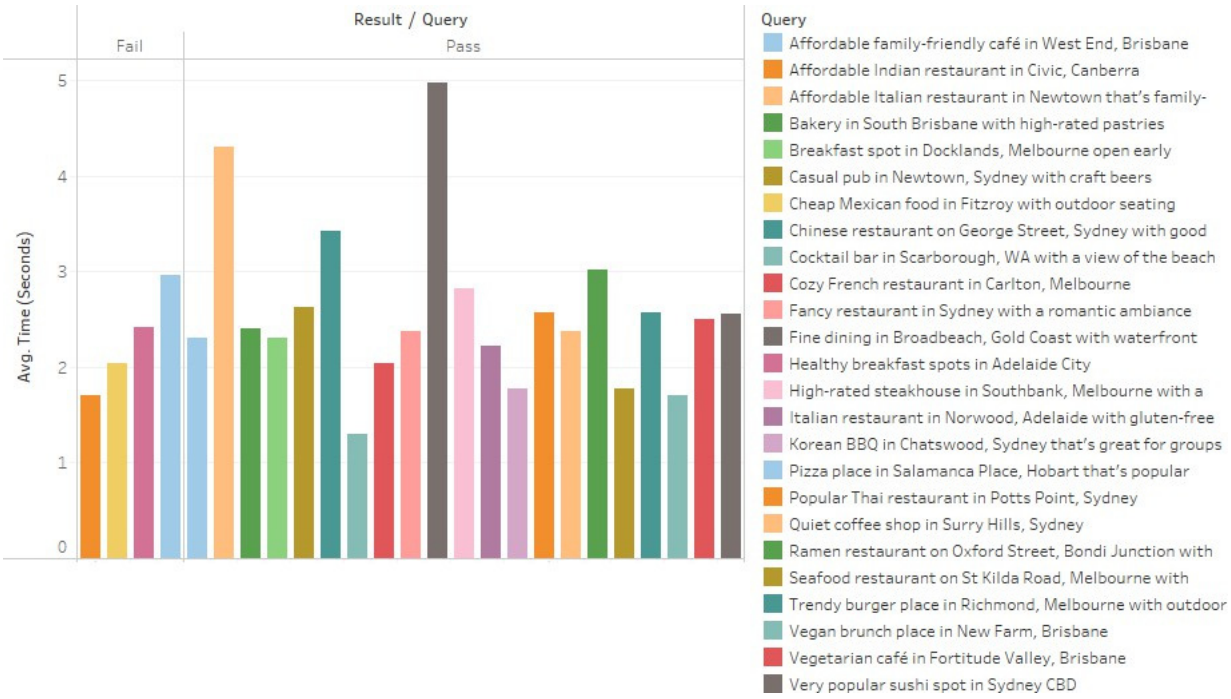


Figure 10. Pass Fail Status of Complex Queries with response time.

The survey results indicate that users had a positive experience with the system, highlighting its usability and effectiveness. Participants evaluated various aspects, including food preference matching, personalization and customization of recommendations, overall search quality and relevance, system speed and response time, and how the recommendations compared to those from Google and other designed engines. Additionally, users rated their acceptance of the recommendations provided by the smart recommendation system. Figure 11 summarizes the average scores for each survey question.

Figure 11. User Satisfaction Survey Results.

However, users rated their likelihood of choosing this recommendation system over Google at a scale of 3.94/5.0. Although this score is slightly lower than other usability metrics, it suggests that users recognize the potential of the system as an alternative to Google, particularly if improvements are made in personalization and handling special requirements. The lower rating may also reflect the familiarity of users and the habitual reliance on Google's extensive database, highlighting an opportunity for further enhancement. Expanding data sources and improving personalization features could increase adoption and position the system as a competitive alternative.

6. Conclusions and Future Work

This research advances the interaction between LLMs and APIs to improve recommendation systems, providing a more intuitive, responsive, and effective platform for interpreting complex user queries. Despite its strong performance, some limitations persist in designed systems, such as the reliance on the Google Places API, which can introduce potential delays during system loads and affect its response times. In addition, some respondents suffer from hallucinations. The findings suggest future research directions of a more sophisticated prompt design and an alternative real-time data source to replace the Google Places API. Moreover, this research provides a scalable framework for multi-domain applications beyond restaurant recommendations in areas such as travel, entertainment, and healthcare, where real-time AI-driven recommendations can offer substantial value. In conclusion, this work demonstrates the potential of integrating LLMs with APIs to build intelligent, real-time recommendation systems, with continued advancements in personalization, scalability, and system optimization paving the way for broader adoption.

Author Contributions: Conceptualization, A.U., A.L.; methodology, A.U. and A.L.; software, A.L.; validation, A.L.; formal analysis, A.U. and A.L.; investigation, A.U., A.L.; data curation, A.L.; writing—original draft preparation, A.U., X.L.; writing—and editing, X.L., A.U.; visualization, X.L., A.U.; supervision, A.U.; project administration, A.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request. Only sample queries can be shared. However, software code is not available due to ethical reasons.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Johnsen, M. (Ed.) *Developing AI Applications with Large Language Models*. 2025. Available online: https://www.maria-johnsen.com/ai-applications-with-large-language-models/ (accessed on 21 February 2025).
2. Zhao, Z.; Fan, W.; Li, J.; Liu, Y.; Mei, X.; Wang, Y.; Wen, Z.; Wang, F.; Zhao, X.; Tang, J.; et al. Recommender systems in the era of large language models (llms). *IEEE Trans. Knowl. Data Eng.* **2024**, 1–20.
3. Li, J.; Xu, J.; Huang, S.; Chen, Y.; Li, W.; Liu, J.; Lian, Y.; Pan, J.; Ding, L.; Zhou, H.; et al. Large language model inference acceleration: A comprehensive hardware perspective. *arXiv* **2024**, arXiv:2410.04466.
4. Gokul, A. LLMs and AI: Understanding Its Reach and Impact. *Preprints* **2023**. [CrossRef]
5. Li, L.; Zhang, Y.; Liu, D.; Chen, L. Large Language Models for Generative Recommendation: A Survey and Visionary Discussions. *arXiv* **2024**, arXiv:cs.IR/2309.01157.
6. Silva, B.; Tesfagiorgis, Y.G. Large Language Models as an Interface to Interact with API Tools in Natural Language. 2023. Available online: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1801354&dswid=5686 (accessed on 21 February 2025).
7. Spinellis, D. Pair Programming with Generative AI. 2024. Available online: https://research.tudelft.nl/en/publications/pair-programming-with-generative-ai (accessed on 21 February 2025).
8. Patil, S.G.; Zhang, T.; Wang, X.; Gonzalez, J.E. Gorilla: Large language model connected with massive APIs. *arXiv* **2023**, arXiv:2305.15334.
9. Luo, D.; Zhang, C.; Zhang, Y.; Li, H. CrossTune: Black-box few-shot classification with label enhancement. *arXiv* **2024**, arXiv:2403.12468.
10. Fan, W.; Zhao, Z.; Li, J.; Liu, Y.; Mei, X.; Wang, Y.; Tang, J.; Li, Q. Recommender Systems in the Era of Large Language Models (LLMs). *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 6889–6907. [CrossRef]
11. Roumeliotis, K.I.; Tselikas, N.D.; Nasiopoulos, D.K. Precision-Driven Product Recommendation Software: Unsupervised Models, Evaluated by GPT-4 LLM for Enhanced Recommender Systems. *Software* **2024**, *3*, 62–80. [CrossRef]
12. Mao, J.; Zou, D.; Sheng, L.; Liu, S.; Gao, C.; Wang, Y. Identify critical nodes in complex networks with large language models. *arXiv* **2024**, arXiv:2403.03962.
13. Lin, J.; Dai, X.; Xi, Y.; Liu, W.; Chen, B.; Zhang, H.; Liu, Y.; Wu, C.; Li, X.; Zhu, C.; et al. How can recommender systems benefit from large language models: A survey. *ACM Trans. Inf. Syst.* **2023**, *43*, 1–47.
14. Hu, S.; Tu, Y.; Han, X.; He, C.; Cui, G.; Long, X. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv* **2024**, arXiv:2404.06395.
15. Corecco, N.; Piatti, G.; Lanzendörfer, L.A.; Fan, F.X.; Wattenhofer, R. An LLM-based Recommender System Environment. *arXiv* **2024**, arXiv:2406.01631.
16. shadcn. shadcn/ui: Modern UI Components for React. shadcn/ui Official Documentation. 2025. Available online: https://ui.shadcn.com/ (accessed on 17 January 2025).
17. Optimizing: Third Party Libraries | Next.js—Nextjs.org. Available online: https://nextjs.org/docs/app/building-your-application/optimizing/third-party-libraries (accessed on 17 January 2025).
18. Vercel AI SDK. OpenAI—SDK Documentation. Vercel AI SDK. 2025. Available online: https://sdk.vercel.ai/providers/ai-sdk-providers/openai (accessed on 17 January 2025).
19. Next.js by Vercel—The React Framework—Nextjs.org. Available online: https://nextjs.org (accessed on 15 January 2025).
20. Vercel: Build and Deploy the Best Web Experiences with the Frontend Cloud—Vercel—Vercel.com. Available online: https://vercel.com/ (accessed on 15 January 2025).
21. Raza, S.; Rahman, M.; Kamawal, S.; Toroghi, A.; Raval, A.; Navah, F.; Kazemeini, A. A comprehensive review of recommender systems: Transitioning from theory to practice. *arXiv* **2024**, arXiv:2407.13699.

22. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *J. Big Data* **2022**, *9*, 59.
23. Gope, J.; Jain, S.K. A survey on solving cold start problem in recommender systems. In Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 5–6 May 2017; pp. 133–138.