Recipe Radar: A Voice-Driven Hybrid Recommender System for Recipes Using NLP, Text-to-SQL, and Cloud-Native Infrastructure

Aleksandra Kolevska[1], Natasha Blazheska-Tabakovska[1]

[1] *University "St. Kliment Ohridski" - Bitola, Faculty of Information and Communication Technology, Studentska bb, Bitola, R.North Macedonia*

aleksandrakolevska1@hotmail.com; natasa.tabakovska@uklo.edu.mk

Abstract

In the era of exponential data growth and digital habits, personalised recommendations have become a necessity. This paper presents an intelligent recipe recommendation application that combines voice interaction (Speech-to-Text), natural language processing, Text-to-SQL, and a hybrid recommendation approach (demographic + collaborative filtering). Users can express their preferences naturally through speech or text, and the system automatically translates them into SQL queries, fetches data from Supabase tables, and returns personalised results. The solution is implemented with Angular (frontend), Python (backend), OpenAI for SQL generation, and Supabase (PostgreSQL) for storage and authentication. This demonstrates how integrating multiple technologies and intelligent methods leads to more relevant recommendations and improved user experience.

Keywords:

Recommender systems, Voice-Driven, Hybrid Recommender System, NLP, Speech-to-Text, Text-to-SQL

1. Introduction

Data is no longer merely a byproduct of digital activity, it has become a critical asset that drives decision-making, innovation, and economic growth. It is generated continuously and at an exponential rate [1], with recent studies estimating that approximately 2.5 quintillion bytes of data are produced every day [2], a figure that continues to rise. This explosive growth is largely driven by the proliferation of the internet and the integration of connected technologies across nearly all aspects of daily life. In this context, the need for robust and intelligent data life cycle management has become increasingly important, not only for efficient storage and processing but also for extracting actionable insights. One of the most prominent applications of such insights is the development of personalised recommendation systems, which utilise user behaviour, preferences, and contextual information to provide relevant and timely suggestions. Personalised recommendations take into account the user's digital footprint along with detailed product information—such as specifications, user feedback, and comparisons with similar products—before generating a suggestion [3]. These systems have proven effective in enhancing user engagement, improving conversion rates, and fostering long-term retention. A knowledge-based recommender system differs from traditional approaches by generating recommendations in response to specific user-defined queries or constraints, rather than relying on historical ratings or behavioral data. In recent years, recommender systems have evolved considerably, driven by significant advances in machine learning, deep learning, and big data technologies. As intelligent software solutions, they aim to reduce cognitive load and personalise content or product offerings, making them a fundamental component of modern digital ecosystems.

Effective personalised recommendations begin with the collection of comprehensive data, including explicit user inputs as well as implicit behaviours such as browsing history, purchase activity, and interaction patterns. This rich dataset is then processed using advanced algorithms and analytical tools designed to uncover meaningful patterns, trends, and individual preferences. Leveraging these insights, predictive models forecast the products or services that a user is most likely to value next. Finally, these personalised recommendations are delivered seamlessly to the user through appropriate channels.

This paper focuses specifically on recipe recommendation, where natural language and voice commands serve as intuitive interfaces between the user and the underlying data infrastructure.

The paper is structured as follows: Following the introduction, the related work section provides a comprehensive overview of recommender systems and emerging technologies. The subsequent section, system architecture**,** describes the overall design of the system, highlighting the integration of voice recognition, NLP, automatic SQL generation, user behaviour analysis, and a hybrid recommendation engine. The next section details the technical realisation of the system, covering the development of the Angular-based frontend, the Python backend, and integration with the Supabase database. The results and discussion section demonstrate the system's ability to process natural language input and generate personalised, real-time recommendations. It also evaluates the system's performance in addressing key challenges, such as the cold-start problem and user accessibility. Finally, the paper concludes with a summary of the key findings and contributions, reflections on the results, and recommendations for future research and development. Additionally, it discusses the limitations and challenges encountered.

2. Related Work

Francesco, Rokach, and Bracha highlight that recommender systems can be categorised based on the domain they operate in, the knowledge they leverage, and most notably, the algorithms used to predict the relevance or usefulness of a recommendation [4]. Differences also arise in how these systems compile and present recommendations in response to user queries.

According to Aggarwal, recommender systems are generally grouped into four main categories: collaborative filtering, content-based filtering, demographic filtering, and hybrid models [5]. Collaborative filtering assumes that users with similar past preferences will share similar tastes in the future [6]. It can be further divided into user-based and item-based approaches. In user-based collaborative filtering, recommendations are derived by identifying users with similar rating patterns [7], while item-based filtering calculates item similarities and recommends items similar to those a user has already liked [8].

Content-based filtering, on the other hand, builds user profiles by analysing item attributes and using labelled item descriptions as training data. This approach recommends items similar to those the user has rated positively in the past [5].

Demographic filtering classifies users based on demographic characteristics such as age, gender, or location. Users within the same demographic category are assumed to share similar preferences. For new users, recommendations are generated by identifying the demographic group they belong to and applying the preferences of similar users in that group [9].

In contrast, knowledge-based recommender systems rely on explicit domain knowledge and predefined rules or constraints [10]. Unlike collaborative or content-based systems that infer preferences from user interactions, knowledge-based systems are designed to deliver recommendations based on user-defined queries or specific requirements. As noted by Aggarwal, these systems do not depend on collective user behaviour or machine learning models, but instead provide transparent and explainable results [5]. This makes them particularly suitable for domains where precision, trust, and interpretability are essential [11].

Hybrid recommender systems integrate multiple recommendation techniques to capitalise on their strengths and mitigate individual limitations. Bell and Koren emphasise the success of hybrid models in the Netflix Prize competition, where combining predictions from diverse models led to an 8.43% improvement over Netflix's original algorithm, Cinematch [12]. The continued evolution of hybrid systems has been driven by the need for more accurate, robust, and personalised recommendations [6].

Beyond traditional recommendation algorithms, the integration of natural language processing (NLP) and voice technologies has significantly expanded the functionality and accessibility of recommender systems. Speech-to-text and text-to-speech technologies allow users to interact with systems using voice commands, enabling multimodal communication and enhancing the user experience [13].

NLP serves as a critical interface between human language and machine interpretation, enhancing human-computer interaction through techniques such as tokenisation, natural language understanding (NLU), and natural language generation (NLG). According to Stryker and Holdsworth, NLP systems can be implemented using rule-based approaches, statistical models, or neural architectures based on deep learning [14].

Another important advancement is the development of Text-to-SQL technologies, which allow users to interact with databases using natural language instead of formal query languages. Zhu et al. point out that SQL can be a barrier for non-technical users and even challenging for practitioners when dealing with domain-specific queries. Text-to-SQL systems translate natural language questions into structured SQL queries, significantly lowering the threshold for accessing and manipulating data [15].

These technological advancements form the foundation for the proposed system, Recipe Radar, which combines voice interfaces, NLP, Text-to-SQL translation, and a hybrid recommendation model to provide intelligent, accessible, and user-centric recipe recommendations.

3. System Architecture

The developed system presents a comprehensive integration of advanced technologies, combining voice recognition (Figure 1), natural language processing (NLP), automatic SQL generation (Text-to-SQL), user behaviour analysis, and a hybrid recommendation engine to deliver personalised recipe suggestions. The conceptual architecture of the system illustrates the interaction between these components, enabling seamless communication from user input to data retrieval and presentation.

Figure 4: Conversion of human speech to text using the browser native Web Speech API

On the frontend, the application is built using Angular and Angular Material, which together offer a modern, responsive, and accessible user interface. The frontend is modularised into distinct components, including: `web-speech`, `recommended-recipes`, `recipe-details`, `profile`, `add-recipe`, and `charts`. Each module is designed to handle a specific aspect of the user experience, from voice input and recipe browsing to profile management and data visualisation. Angular Material further enhances usability by ensuring consistent design patterns and accessibility compliance.

The backend is implemented in Python, serving as the middleware that processes user inputs, orchestrates logic, and facilitates communication with the database. It also handles the execution of SQL queries generated via natural language and the delivery of results back to the frontend.

The relational database is hosted on Supabase, a cloud-native backend-as-a-service built on PostgreSQL. The database schema consists of four primary tables: `recipes`, `ratings`, `profiles`, and `shopping_list`. These tables respectively capture information on recipe content, user-generated ratings, demographic and behavioural profiles, and user-curated shopping items. This structured data model supports both collaborative filtering and demographic-based recommendations, enabling a multi-dimensional recommendation strategy.

A key innovation in the system is the incorporation of Text-to-SQL functionality, which bridges the gap between natural language input and structured database queries. (Figure 2) The process involves three core steps: (1) Schema extraction – the system dynamically extracts the database schema and provides it as structured context to the language model; (2) Prompt engineering – Carefully designed prompts guide the model (e.g., a large language model) to generate accurate SQL queries in a structured JSON format; Query execution – The resulting SQL is parsed and executed against the Supabase database, and the response is formatted and returned to the user.

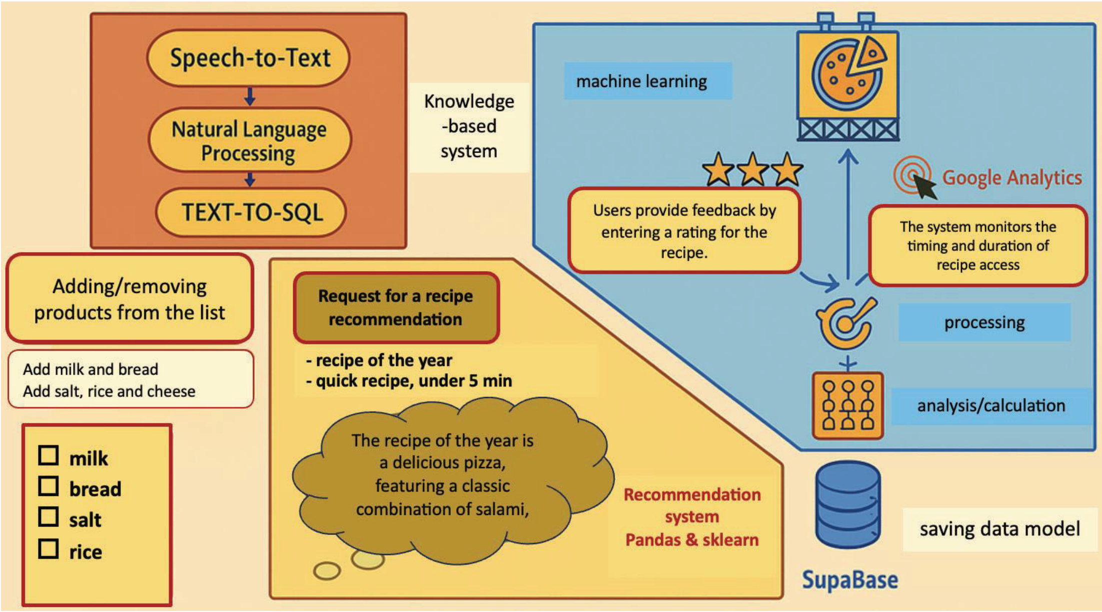

Figure 5: Architecture of the system

The speech interface enhances user interaction by enabling natural, voice-based communication. Spoken commands are processed via speech-to-text technology and passed through the NLP and Text-to-SQL pipeline, allowing even non-technical users to retrieve recipe data without needing to understand or write SQL.

Together, these components form a robust and user-centric system, Recipe Radar, that leverages modern AI and cloud-native technologies to deliver intelligent, personalised culinary recommendations through a multimodal interface.

4. Implementation

The frontend of the system is developed using Angular, leveraging reactive forms for recipe submission, modular visual components for recipe display, and seamless integration with Supabase via RESTful APIs. This design ensures a responsive, user-friendly interface capable of handling dynamic data interactions in real-time. The visual components are structured to enhance usability and accessibility, supporting an intuitive flow from user input to data presentation.

The backend is implemented in Python, functioning as the core processing layer of the application. It handles both textual and spoken user input, utilising OpenAI's language models to perform natural language to SQL (Text-to-SQL) translation (Figure 3). The generated SQL queries are executed against a PostgreSQL database hosted on Supabase, with the retrieved results relayed back to the frontend for display. This architecture enables natural, multimodal user interaction without requiring technical knowledge of database query languages.

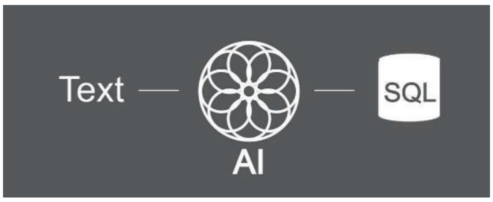

Figure 6: Text-to-SQL process: schema extraction, SQL generation, and execution.

The system employs a hybrid recommendation strategy (Figure 4) that integrates both demographic filtering and collaborative filtering techniques to enhance the relevance and diversity of suggestions.

- Demographic filtering identifies users with similar attributes—specifically, matching gender, city, and an age range within ±5 years—to address the cold-start problem often encountered in recommendation systems.
- Collaborative filtering constructs a user–recipe interaction matrix based on user-generated ratings. This matrix is processed using cosine similarity and the NearestNeighbors algorithm, allowing the system to identify users with overlapping taste profiles. Recommendations are then generated by aggregating highly rated recipes from a user's nearest neighbours.

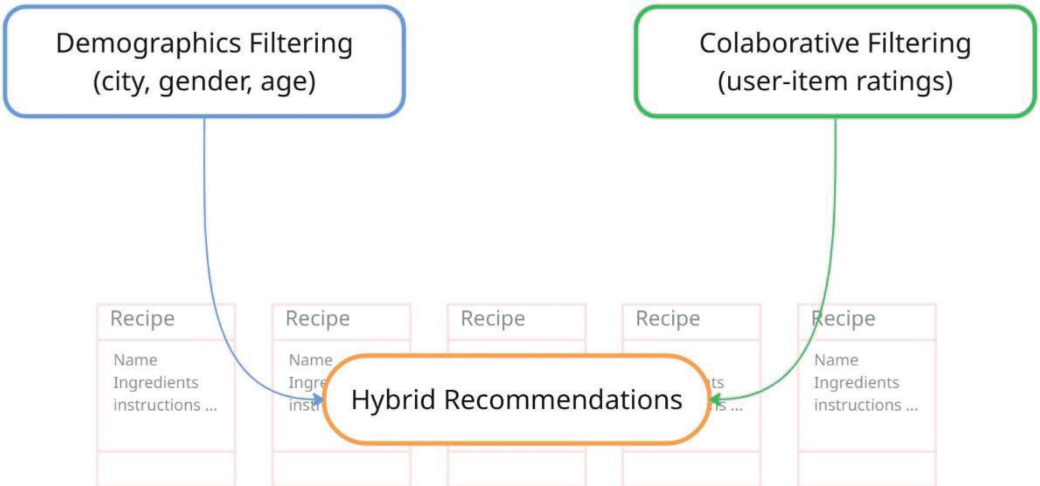

Figure 7: Hybrid filtering approach

For instance, if a user consistently rates vegetarian dishes positively, the collaborative filtering component will prioritise vegetarian recipes favoured by users with similar preferences. Simultaneously, the demographic filter ensures that the recommendations are contextually appropriate by incorporating users with similar demographic profiles.

This hybrid approach enhances recommendation accuracy and robustness across varying user scenarios, improving both personalisation and system adaptability. In the user interface, the final recommendations and query results are presented using visually engaging cards, each displaying an image, title, and brief description of the recipe. This presentation format supports a rich and immersive user experience, effectively bridging the gap between data-driven logic and user-centric design.

5. Results and Discussion

An illustrative example of system interaction involves a user issuing the voice command: *"Add 3 oranges to the shopping list."* This input is captured via the speech interface, transcribed into text, and processed through the Text-to-SQL pipeline. The system subsequently generates the appropriate SQL query, updates the relational database, and synchronously reflects the change in the user interface. This interaction exemplifies the seamless integration of natural language processing with structured database operations, enabling users to perform complex tasks without direct interaction with database query languages.

A core innovation of the system lies in its hybrid recommendation engine, which combines both demographic and collaborative filtering techniques (Figure 5). This dual approach addresses the limitations inherent in using a single strategy. For new users with limited interaction history, a common challenge known as the cold-start problem, demographic filtering provides a basis for initial recommendations by leveraging user attributes such as age, gender, and location. In contrast, for users with extensive activity logs, collaborative filtering draws upon shared behavioural patterns to generate more refined suggestions. This ensures that the system remains effective across a spectrum of user profiles, from novices to highly engaged users.

Another notable feature is the system's Text-to-SQL interface, which significantly lowers the barrier to entry for non-technical users by enabling interaction through everyday language. This is complemented by the use of Angular Material, which adheres to modern UI/UX standards and accessibility guidelines, ensuring that the application remains both usable and inclusive.

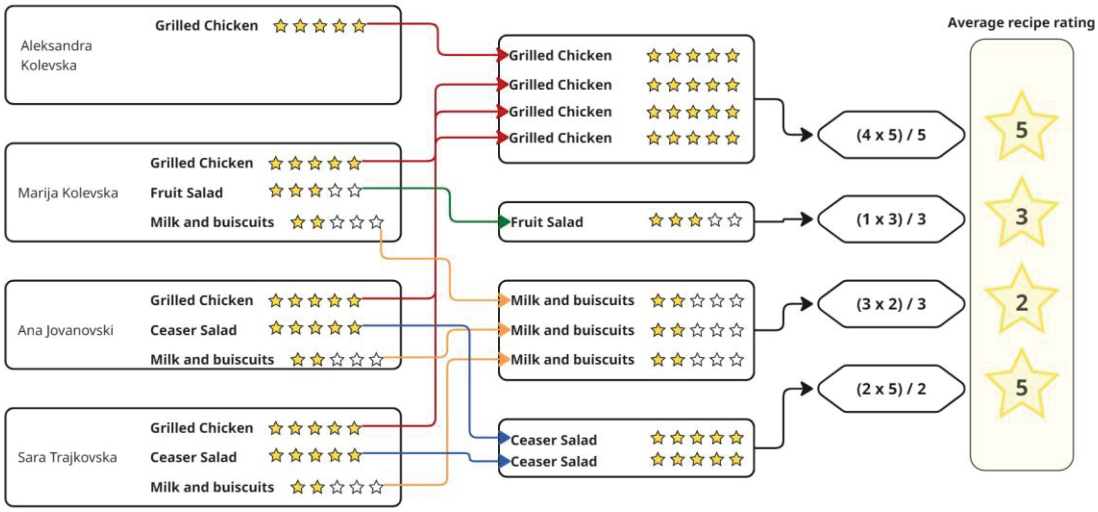

Figure 8: *Recommendation flow: demographic and rating-based filtering*

The design philosophy aligns with practices observed in large-scale commercial platforms. For example, Amazon employs item-to-item collaborative filtering supplemented with content-based signals, while Netflix integrates deep learning within its hybrid recommendation systems to capture complex user preferences. These real-world analogues underscore the scalability, relevance, and practical applicability of the hybrid approach adopted in this work.

Despite its strengths, the system exhibits several limitations**.** The accuracy of the speech-driven interface is contingent upon reliable speech recognition performance, which may be affected by accents, background noise, or pronunciation variations. Additionally, the platform requires stable internet connectivity to ensure real-time interaction with the Supabase backend. Furthermore, the efficiency and correctness of SQL query generation remain critical for delivering accurate responses, particularly in edge cases or ambiguous queries.

6. Conclusion

This study presents the design and implementation of an integrated, intelligent, and user-centric recipe recommendation system that leverages a multimodal interaction framework incorporating speech recognition**,** natural language processing (NLP)**,** automatic SQL generation**,** and cloud-native data storage**.** By combining these technologies, the system allows users, regardless of technical expertise, to interact with a structured relational database using natural language, either through text or voice input. This contributes to a more inclusive and accessible user experience, especially for non-technical or visually impaired individuals.

At the core of the system lies a hybrid recommendation engine**,** which fuses demographic filtering and collaborative filtering to offer robust and context-sensitive recipe suggestions. Demographic filtering utilises user metadata (such as age, gender, and geographic location) to provide personalised recommendations in scenarios with limited historical data, addressing the well-known cold-start problem**.** In contrast, collaborative filtering leverages historical rating data to uncover latent preferences and similarities between users, offering deeper personalisation as user interaction grows. The integration of these two methods creates a flexible recommendation architecture capable of adapting to both novice and experienced users.

The system architecture, comprising an Angular frontend, Python-based backend, and Supabase-hosted PostgreSQL database, demonstrates the feasibility of deploying scalable, cloud-native AI applications using modern web technologies. The implementation of Text-to-SQL pipelines, powered by large language models, enables semantic interpretation of natural language queries into executable SQL commands. This functionality significantly reduces the barrier to data retrieval, empowering users to access relevant content through intuitive interaction modalities.

The practical utility of this system extends beyond the culinary domain. The architectural design and underlying methodologies are readily applicable to other fields that require personalised content delivery, such as healthcare (e.g., personalised diet planning or medication tracking), education (e.g., personalised learning content recommendations), and e-commerce (e.g., tailored product suggestions). This demonstrates the system's potential for cross-domain applicability and scalability in real-world deployments.

Nevertheless, certain limitations persist. These include a dependency on high-quality speech recognition, the need for a reliable internet connection, and the potential for inaccuracies in SQL generation under ambiguous or poorly structured user queries. Addressing these challenges is essential for enhancing system reliability and user trust.

Future research directions will focus on enhancing the recommendation engine through the incorporation of deep learning-based models**,** such as neural collaborative filtering or graph neural networks, to capture more complex user-item interactions. Additionally, context-aware filtering mechanisms**,** taking into account temporal patterns, location data, dietary restrictions, and seasonal availability, may further refine personalisation. Expanding the scope of the system to include **non-recipe** domains and evaluating its performance in real-world user studies will also contribute to a deeper understanding of its usability, generalizability, and societal impact.

However, several limitations were observed during system evaluation. Ambiguous or loosely structured queries such as 'find a healthy recipe' or 'show me something my friends like for dinner' proved challenging for the Text-to-SQL module to interpret accurately, occasionally resulting in incomplete or irrelevant SQL generation. Furthermore, speech recognition accuracy may vary depending on background noise, accent, or pronunciation, which can impact query interpretation. The system’s reliance on stable internet connectivity and the limited diversity of the dataset also affect its consistency and adaptability across broader culinary domains.

The Text-to-SQL component was refined through a structured prompt engineering process. Prompts were iteratively optimized to include database schema information, example SQL structures, and constraints on output format using JSON. This guided the language model to generate more accurate and syntactically valid SQL queries while reducing hallucination and mismatched column references. The process also involved few-shot learning examples to improve query reliability and semantic alignment between natural language inputs and database structure.

In future research, the proposed model will be evaluated through the inclusion of quantitative metrics. This will involve measuring and comparing task completion times, accuracy, and user satisfaction between the voice-driven interface and a traditional graphical user interface (GUI). Such analysis will help validate the practical advantages of voice-enabled interaction in terms of efficiency and accessibility. Additionally, user experience evaluations based on standardized metrics such as the System Usability Scale (SUS) and interaction efficiency (number of steps per query) will be conducted to provide measurable insights into system performance.

References:

[1] M. Phiri, "Exponential Growth of Data," Medium, 19 11 2022. [Online]. Available: https://medium.com/@mwaliph/exponential-growth-of-data-2f53df89124.

[2] A. S. Roy, "How does facebook handle the 4+ petabyte of data generated per day? Cambridge Analytica - facebook data scandal.," 16 9 2020. [Online]. Available: https://medium.com/@srank2000/how-facebook-handles-the-4-petabyte-of-data-generated-per-day-ab86877956f4.

[3] S. K. Gorakala and M. Usuelli, Building a Recommendation System with R, Birmingham, UK: Packt Publishing Ltd., September 2015.

[4] R. Francesco, L. Rokach and S. Bracha, Recommender Systems Handbook, New York: Springer.

[5] Aggarwal, Recommender Systems: The Textbook, Yorktown Heights, NY, USA: Springer International Publisinh Switzerland, 2016.

[6] M. Techlabs, "Types of Recommendation Systems & Their Use Cases," Medium, 19 8 2021. [Online]. Available: https://marutitech.medium.com/what-are-the-types-of-recommendation-

systems-3487cbafa7c9.

[7] M. Harsh, "User-Based Collaborative Filtering," Geeksforgeeks, 22 January 2023. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/user-based-collaborative-filtering/.

[8] M. Harsh, "Item-to-Item Based Collaborative Filtering," Geeksforgeeks, 26 Septermber 2024. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/item-to-item-based-collaborative-filtering/.

[9] R. Iateilang and L. Chameikho, "Recommender Systems: Types of Filtering Techniques," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ,* vol. 3, no. 11, 11.2014.

[10] R. Burke, "Knowledge-Based Recommender Systems," *ncyclopedia of library and information systems. 69. ,* vol. 69, 2000.

[11] T. Siljanoska and N. Blazheska-Tabakovska, "Upgrading Traditional E-Commerce Systems with A Knowledge- Based Recommendation System," in *14th International Conference on Applied Internet and Information Technologies AIIT 2024*, Zrenjanin, 2024.

[12] R. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," in *IEEE International Conference on Data Mining (ICDM'07)*, 2007.

[13] N. Blazheska-Tabakovska, S. Savoska, B. Ristevski, I. Jolevski and D. Gruevski, "Web Content Accessibility for People with Cognitive Disabilities," in *IX International Conference on Applied Internet and Information Technologies AIIT 2019*, Zrenjanin, Serbia, 2019.

[14] C. Stryker and J. Holdsworth, "What is NLP (natural language processing)?," IBM, 11 08 2024. [Online]. Available: https://www.ibm.com/think/topics/natural-language-processing.

[15] X. Zhu, Q. Li, L. Cui and Y. Liu, "Large Language Model Enhanced Text-to-SQL Generation: A Survey," arXiv:2410.06011v1 [cs.DB] 08 Oct 2024, 8 10 2024. [Online]. Available: https://arxiv.org/html/2410.06011v1#bib.