Optimizing Roommate Matching For New Residents By Harnessing The Power Of Chatgpt Conversational Guidance

Tamizhselvan[1], Sredesh[2] and Vinothiyalakshmi[3]
[1,2,3]Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, India

Abstract

Finding a compatible roommate, especially for university newcomers, is a persistent challenge. Current platforms rely on static profiles, limiting their ability to capture user preferences. This paper proposes a groundbreaking web application, the first to leverage Large Language Models (LLMs) for roommate matching. LLMs surpass traditional methods by enabling natural language dialogues. Users can explore potential matches based on lifestyle, communication styles, and cultural backgrounds through free-flowing conversation. This paper explores the theoretical foundation of the LLMpowered system, details its integration, and discusses the potential to revolutionize roommate matching, particularly for immigrant students.

Keywords:

Roommate Matching, Large Language Models (LLMs), Natural Language Processing (NLP), Conversational Search, University Students, Immigrant Students, Personalized Matching

1. INTRODUCTION

Securing a compatible roommate plays a pivotal role in shaping a person's living experience, particularly for students embarking on a new and often unfamiliar environment like college [1]. Pascarella & Terenzini (2005), in their comprehensive study "How College Affects Students: A Third Decade of Research," highlight the crucial role a positive living environment, including compatible roommates, plays in student well-being and academic success [1]. Imagine the excitement and trepidation a freshman feels as they move into a dorm room – a space that will be not only their sanctuary for sleep and study, but also a shared environment that can significantly impact their emotional and social well-being. Traditional roommate matching methods, however, often rely on static online profiles or impersonal questionnaires that fall short in capturing the intricate details that truly define compatibility. These methods, as RentCollegePads.com (2023) underscores, often focus on basic demographic information or simple preferences, neglecting the nuances of personality, communication styles, and lifestyle habits that can make or break a living arrangement [2].

This inherent limitation in traditional methods can lead to mismatched pairings, as evidenced by research on roommate conflict. Neuman & Rubin (2004), in their study "Why Can't We Be Friends? Understanding Roommate Conflict Among College Students," delve into the factors that contribute to roommate clashes within dorms [3]. Their research highlights the importance of personality compatibility beyond basic demographic information, revealing how seemingly minor differences in habits, communication styles, and personal routines can lead to friction and frustration. The consequences of such mismatches can be far-reaching, potentially leading to feelings of social isolation, diminished mental health, and even a decline in academic performance [3, 4]. Prendergast et al. (2014) explore these negative consequences in their study “The Impact of Roommate Conflict on Student Mental Health and Academic Performance”. Their findings demonstrate a correlation between roommate conflict and feelings of isolation, anxiety, and even a decline in academic achievement [4].

This paper proposes a revolutionary solution to address these shortcomings, a roommate matching system powered by Large Language Models (LLMs) to facilitate a more dynamic and nuanced matching process. LLMs, a transformative technology that has emerged in recent years, are capable of engaging in natural language conversations and demonstrating an understanding of complex human interactions [5]. Radford et al. (2021) explore the capabilities of LLMs for simulating human conversations, highlighting their potential for understanding the intricacies of human dynamics beyond the limitations of basic surveys [5]. By integrating LLMs into the system, users can engage in a conversational dialogue, exploring potential matches based on a broader spectrum of criteria than what static profiles can capture. These criteria could encompass not just basic habits like sleep schedules (early bird vs. night owl), but also delve deeper into communication styles (direct vs. indirect), cultural backgrounds, and even personal quirks (board game enthusiast vs. movie buff). This interactive approach allows users to express their needs and preferences in a comfortable and conversational way, fostering a more comprehensive understanding of compatibility between potential roommates.

In conclusion, this study significantly enhances our understanding of using LLMs to improve roommate matching and promote student wellbeing. The findings underscore the potential for substantial advancements in this area, offering valuable implications for both academic research and practical implementation.

2. RELATED WORKS

Traditional roommate matching has relied on static profiles and algorithms focused on achieving stable pairings, where roommates wouldn't have an incentive to switch [6]. While this approach aimed to prevent compatibility issues, it often overlooked the nuances of

users to express their preferences in detail [12].

individual preferences. Research has explored methods that incorporate more userspecific

By prompting the LLM with specific requirements and leveraging techniques like

criteria, such as the "master list" approach, which allows individuals to specify desired qualities in a roommate [7]. This provided a more structured way to consider personality traits, interests, and habits. However, it lacked the flexibility to adapt to user-specified details in realtime.

Another approach implemented machine learning to categorize users based on personality and lifestyle preferences using the K-Means algorithm [8]. This offered a data-driven way to group users with similar characteristics, facilitating the browsing and selection of potential roommates. However, this method relied on pre-defined categories and may not capture the full spectrum of user preferences. Large Language Models (LLMs) emerge as a potential solution to address these limitations. Unlike static profiles, LLMs can engage in natural language conversations, allowing matching process where users feel like they are actively involved in finding the right roommate.

Finally, LLMs can be fine-tuned to address potential biases in the matching process. Similar to the concept of "popularity bias" identified in recommendation systems, roommate matching algorithms might favor generic profiles [12]. By employing prompt engineering techniques, LLMs can be trained to prioritize user-specific preferences and ensure diverse, unbiased matching outcomes. This ensures that users are matched based on compatibility, not on generic characteristics that may not reflect their actual needs.

In our exploration of roommate recommendation systems, we also delved into evaluating fairness within such systems [13]. With the rise of recommendation paradigms driven by Large Language Models (LLMs), it becomes crucial to address potential biases and fairness concerns, particularly in delicate scenarios such as roommate matching.

In essence, LLMs offer a versatile and dynamic approach to roommate matching. Their ability to handle complex user preferences, integrate diverse techniques, and address potential biases paves the way for a more effective and user-friendly matching experience.

3. METHODOLOGY

hierarchical clustering, LLMs can offer more nuanced and personalized matching compared to pre-defined groupings [8]. This allows users to express their ideal living environment beyond pre-selected options, leading to more compatible matches.

Furthermore, LLMs have the capability to integrate diverse information sources concurrently [9]. They can combine techniques from various studies, such as "no-odd-ring" matching for stability and "master list" preferences, while also potentially utilizing algorithms like K-Means or K-Nearest Neighbors with Cosine Similarity [8, 9]. This comprehensive approach allows LLMs to consider various factors for optimal matching, drawing insights from multiple methodologies to create a more robust solution.

Inspired by conversational recommendation systems, LLMs can be used to create an interactive matching experience similar to having a conversation with a friend [12]. Users can provide feedback on suggested matches, allowing the LLM to refine recommendations and increase personalization. This fosters a more user-centric Traditional roommate recommendation methods rely on static profiles with pre-defined options, limiting users' ability to express their unique preferences. This approach often overlooks crucial aspects of compatibility, like movie marathon habits or board game enthusiasm. Our system breaks free from these limitations by employing Large Language Models (LLMs) as the driving force behind its matching prowess. LLMs act as sophisticated conversational partners, engaging users in a natural dialogue. Instead of a checklist, users can describe their ideal living environment in their own words, detailing their desired level of cleanliness, preferred social atmosphere, or even quirky habits.

This conversational approach unlocks a treasure trove of user preferences beyond the superficial. The LLM, with its mastery of natural language, delves into the subtleties of user desires. It can pick up on nuances in phrasing, identify dealbreakers embedded within casual conversation, and even gauge user sentiment to understand true priorities. Here's a table summarizing the key differences between traditional methods and LLM-based matching (Table 1).

Table 1. Comparison of Traditional Methods vs. LLM-based Matching.

Feature	Traditional Methods	LLM-based Matching
User Input	Fixed options, limited flexibility	Open-ended conversation, natural language
Data Captured	Basic demographicsdefined preferences	Rich details on lifestyle, personality, habits
Matching Criteria	Simple compatibility rules (e.g., smoker/nonsmoker)	Multifaceted matching based on user-expressed needs
Scalability	Limited to predefined options	Adapts to new user preferences and feedback
Bias Mitigation	Reliant on potentially biased options	Can be finetuned to reduce bias in matching algorithms
Weaknesses	Limited personalization, inflexible	Scalability Challenge (computational cost)

LLM-based matching offers several advantages beyond simply capturing more user data:

• Unveiling the "Why" Behind Preferences: During conversation, the LLM can uncover the underlying reasons behind user preferences. For example, a user might express a desire for a quiet roommate. Through conversation, the LLM might discover this is because they work from home and require focused silence during the day. This deeper understanding allows for more nuanced matching.

- Dynamic and Iterative: Unlike static profiles, LLM-based matching is a dynamic process. As users converse with the LLM, they can refine their preferences in real-time. This iterative approach ensures the system continuously adapts to capture the most accurate picture of user compatibility.

- Unveiling Hidden Gems: The LLM's ability to identify shared interests or complementary personality traits through conversation can unearth unexpected "perfect matches" that traditional methods might miss.

LLM-based matching represents a paradigm shift in roommate recommendation. It empowers users to express themselves authentically, leading to more fulfilling and long-lasting roommate relationships.

3.1 SYSTEM ARCHITECTURE

The roommate recommendation system leverages a three-tier architecture (Fig. 1) to facilitate user interaction, data storage, and intelligent matching powered by Large Language Models (LLMs).

- **User Interface (UI)**: This user-friendly interface serves as the entry point for users. It allows them to create profiles, specify their preferences (lifestyle, living habits, dealbreakers), search for potential roommates, and engage in LLMpowered conversations through a dedicated chat window.

- **Database**: This central repository acts as the system's memory, storing user profiles. Each profile is a rich tapestry of information, encompassing details explicitly provided by the user during profile creation and implicitly extracted by the LLM through conversations. This comprehensive data serves as the foundation for the LLM's matching process.

- **Large Language Model (LLM)**: The LLM acts as the system's intelligence engine. It interacts with users conversationally within the UI's chat window. By processing user queries and engaging in natural language dialogue, the LLM gleans valuable insights into the user's ideal roommate. It then retrieves profiles from the database that best match the user's preferences and identified compatibility factors.

The Flow of Information:

Fig. 1 depicts the seamless flow of information within the system:

1) **Raw Data Acquisition**: Users create profiles on the UI, furnishing details about their lifestyle, personality, interests, and housing preferences.

2) **Data Feeding to LLM**: This user-provided information is stored in the database, making it readily accessible to the LLM.

3) **Prompting the LLM**: Guided by prompts like "Describe your ideal roommate in terms of personality" or "What kind of living environment are you looking for?", users interact with the LLM through the chat window. These prompts encourage natural language conversation, allowing the LLM to delve deeper into user preferences.

4) **LLM Processing and Response Generation**: The LLM leverages its understanding of natural language and the user's conversation history to identify key compatibility factors. It then retrieves profiles from the database that best match these factors. The LLM may also ask clarifying questions during the conversation to refine its understanding and optimize the matching process.

5) **Recommendation Generation:** Based on the retrieved profiles, the system generates a shortlist of potential roommate matches for the user. This shortlist can be accompanied by explanations highlighting the key compatibility factors between the user and each recommended match.

Addressing Biases:

The system incorporates mechanisms to mitigate potential biases in the matching process:

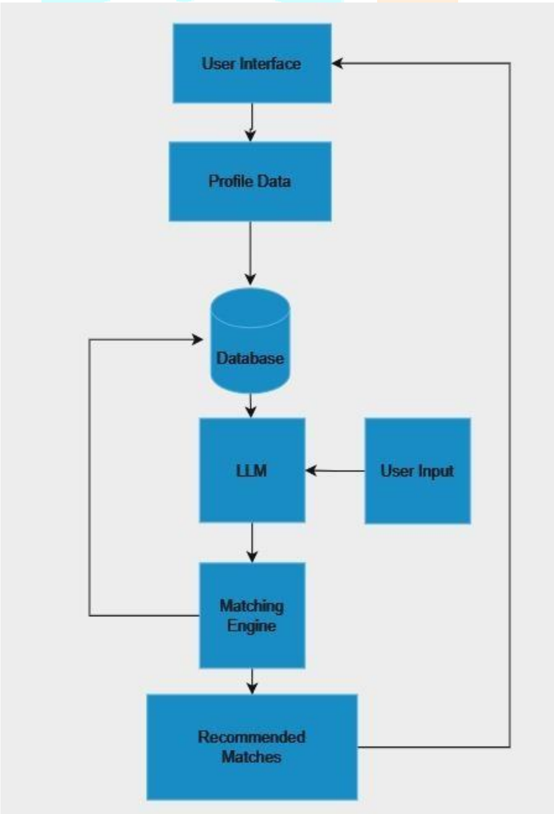
- **Prompt Engineering**: Carefully crafted prompts that avoid biased language steer the conversation towards user-specific preferences rather than stereotypical assumptions.

- **Dataset Curation**: The LLM is fine-tuned on a diverse dataset of roommate profiles to minimize the influence of biases present in real-world data.

- **User Feedback Integration**: The system allows users to provide feedback on the recommendations. This feedback is used to identify and address any emerging biases within the LLM's matching process.

By combining a user-friendly interface, a comprehensive database, and the power of LLMs, this system offers a robust and adaptable framework for roommate recommendation. It empowers users to express their preferences in a nuanced way, leading to more compatible and successful roommate pairings.

Fig. 1: Architectural Diagram of the Roommate Recommendation System



4. IMPLEMENTATION

The developed application seamlessly integrates advanced natural language processing techniques with user-centric design principles to provide users with a personalized and efficient roommate matchmaking experience.

The roommate recommendation system utilizes MongoDB for data storage, pymongo for database interaction, langchain_openai for natural language processing with OpenAI's language model, Flask for web server creation, and FAISS for efficient similarity search of text embeddings. Now, let's delve into a detailed exploration of the steps involved in the application's operation, elucidating the intricacies of its functionality.

The LLM-powered roommate recommendation system relies on a cohesive interplay between data retrieval and processing, natural language processing (NLP) with OpenAI, and a user-friendly web interface built with Flask. Here's a breakdown of the key implementation aspects:

1) **Data Retrieval and Processing**:

MongoDB Interaction: The system leverages pymongo to connect to a MongoDB Atlas cluster, the data repository for user profiles. It retrieves all user profiles stored within the specified database (e.g., "Dream_Nest"). Each user profile is a document containing details about the user's preferences and living habits.

Text Chunking: To enhance processing efficiency, the system breaks down the raw text from user profiles into smaller chunks using the CharacterTextSplitter class. This allows the LLM to handle the information more effectively.

Document Search Index: FAISS is employed to create a search index for the text chunks. The system utilizes OpenAI's language model to generate embeddings (compressed vector representations) that capture the semantic meaning of each chunk. By indexing these embeddings, the system facilitates efficient retrieval of documents with similar semantic content to a user's query.

2) **Natural Language Processing (NLP) with OpenAI**:

Embedding Text with OpenAI: The OpenAIEmbeddings class plays a crucial role in converting text inputs (user queries and user profile descriptions) into embeddings. These embeddings allow the LLM to grasp the underlying meaning and identify potential matches based on semantic similarity.

Loading Question Answering Model: The load_qa_chain function from the langchain package is responsible for loading a questionanswering (QA) model. This pre-trained model, fine-tuned with OpenAI's language model, empowers the system to answer user queries by extracting relevant information from the user profile embeddings stored in the database.

3) **Web Interface with Flask**:

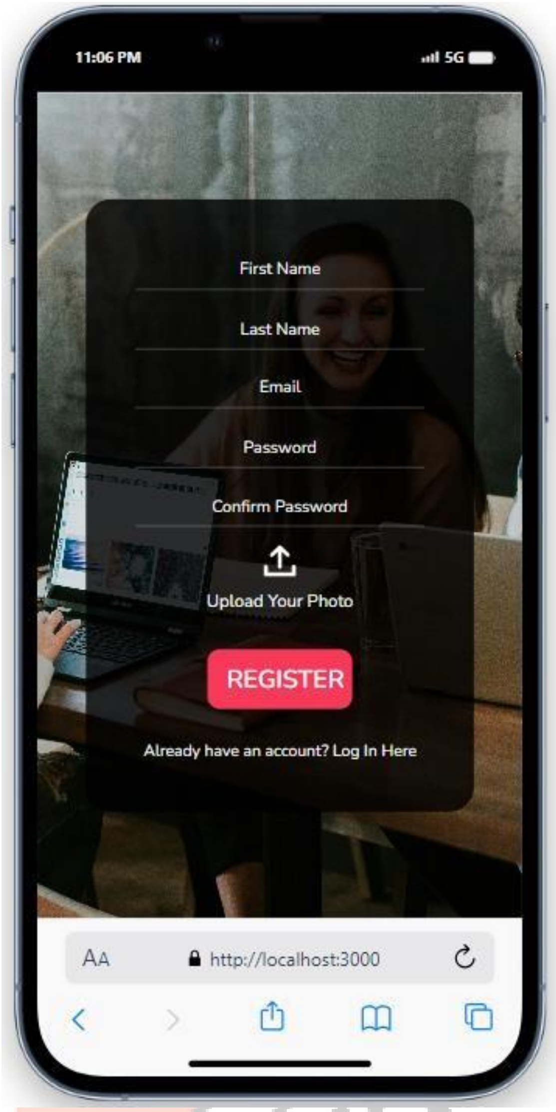
Flask Web Server: Flask serves as the foundation for the user interface. It creates a web server that listens for user requests and defines routes corresponding to specific functionalities. Common routes include one for the homepage (/) where users initiate the search and another for displaying results (e.g., "/result").

Rendering HTML Templates: Flask's render_template function brings the user interface to life by rendering pre-designed HTML templates. These templates define the layout and visual elements that users interact with, such as the search query form and the presentation of recommended roommate matches.

5. RESULT

The built application is presented along with various website routes that highlight its main pages and how they help with the roommate search. The program is based on the strong MERN stack, an effective set of technologies that serves as the foundation for a lot of contemporary online applications. This powerful combination provides a scalable and flexible architecture, perfectly suited for the demands of a dynamic web application. Additionally, various Python libraries are strategically integrated to enhance specific functionalities, ensuring an exceptional user experience.

Fig. 2: Registration Form



The Fig. 2 illustrates the registration form of the developed application.

This form serves as the initial step for users to gain access to the features of the web application.

Registration is a prerequisite for accessing certain functionalities, such as hosting or booking a room. Users are prompted to provide necessary information to create a unique account, enabling them to utilize the platform's services fully.

Fig. 3: Login Form

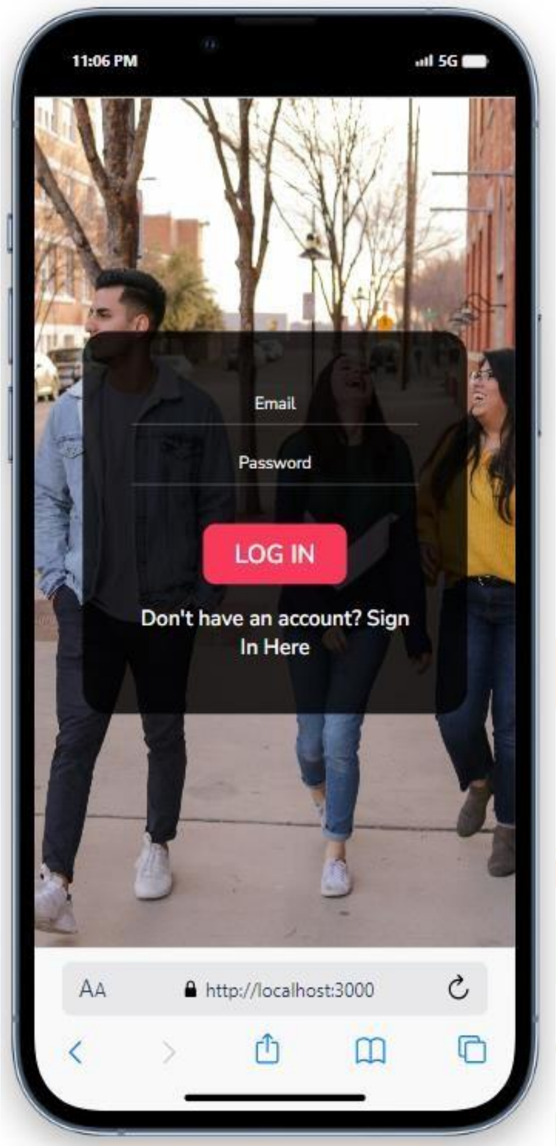


Fig. 3 depicts the login form, the gateway to the application's core functionalities for registered users. To ensure a seamless login experience, the system employs a secure authentication mechanism. Each user possesses unique login credentials, with the registered email address serving as a convenient username. Password security is of paramount importance; passwords are never stored in plain text. Instead, they undergo a one-way hashing process using industry-standard algorithms. This hashing process transforms the password into a unique, irreversible string, rendering it unreadable even in the event of a data breach. Additionally, JSON Web Tokens (JWTs) are employed to further enhance security by providing temporary, authenticated access after successful login. JWTs are self-contained tokens that contain essential user information and a digital signature. This approach mitigates the risk of session hijacking.

Fig. 4: Landing Page

Fig. 4 illustrates the landing page, the user's primary entry point after successful login. The landing page features a navigation bar that dynamically adapts based on the user's login status. For unregistered users, the navigation bar prompts them to register or log in. Conversely, logged-in users benefit from a comprehensive navigation menu, offering access to various functionalities:

- **Roommates List:** Provides details of rooms booked by the user.
- **Wish List:** Allows users to curate a list of preferred rooms encountered during their search.
- **Your Room:** Enables users to view and manage properties or rooms they own or host.

- **Booked List:** Presents details of user-hosted rooms that have been booked by other users.

- **Logout:** Provides a convenient option for users to log out of the web application.

Fig. 5: Top Categories Section Page

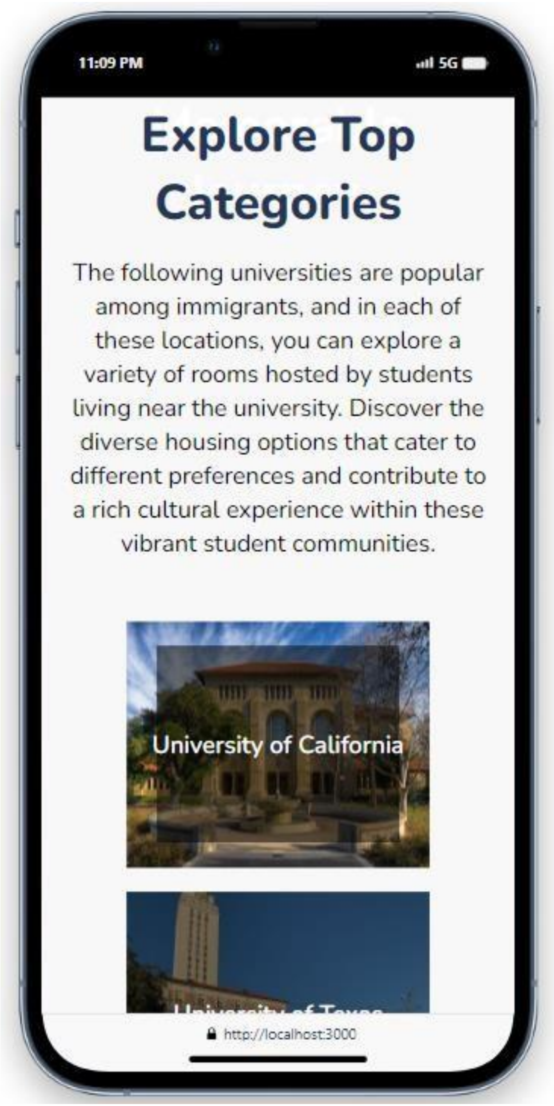


Fig. 5 showcases the top categories section page, designed with the specific needs of immigrant students in mind. Recognizing the challenges faced by this demographic in finding suitable housing, this section thoughtfully highlights universities popular among international students. Users can explore a variety of rooms hosted by students residing near these universities, ensuring convenient access to their academic pursuits. But the benefits go beyond just location. The page emphasizes the diverse housing options available, catering to students with a wide range of preferences. Whether a student seeks a quiet study environment conducive to focused learning, a social living space that fosters interaction and community, or a place that accommodates specific cultural dietary needs, the application offers a wealth of choices.

Fig. 6: Listing Cards Page

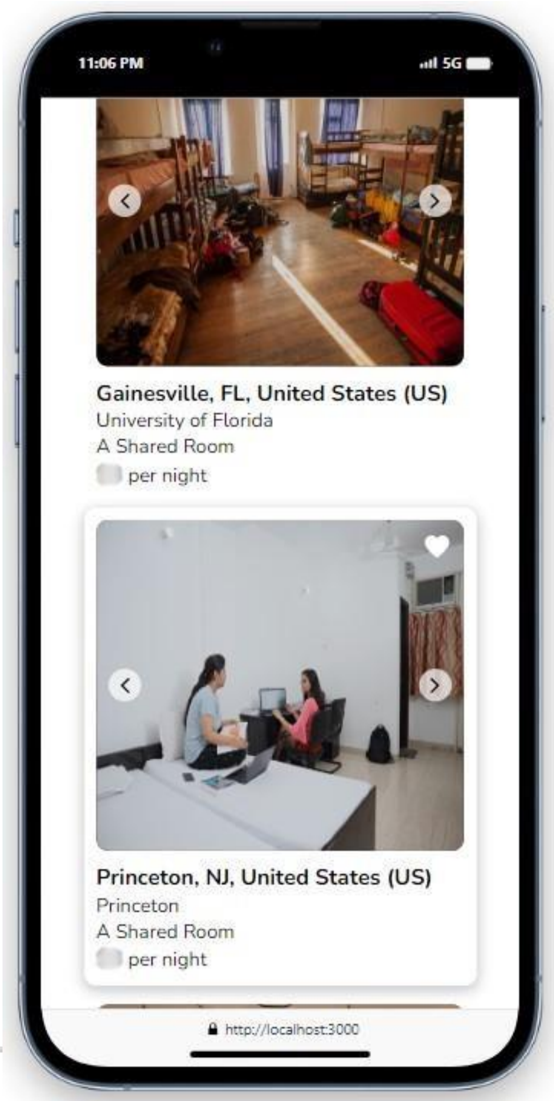


Fig. 6 depicts a sample listing card, a crucial element for displaying information about potential roommates and their rooms. All details pertaining to the roommate and the room are stored within these listing cards. Clicking on a listing card provides users with a comprehensive overview of the roommate's profile, including their preferences and the specific features of the room being offered. This detailed information is critical for the LLM (Large Language Model) component of the application, as it leverages this data to understand user queries and identify compatible roommate matches.

Fig.7: Sample output of the website

Refer to Fig.7 for a visual representation of the sample output. This illustration provides an overview of how the output appears in website.

6. CONCLUSION

Our innovative roommate recommendation system represents a paradigm shift in the field, capitalizing on the capabilities of Large Language Models (LLMs) to deliver a highly personalized and user-centric matching experience. Unlike conventional platforms, our system empowers users to articulate their preferences naturally through conversational interactions, facilitated by OpenAI's language model. By converting user queries and profile descriptions into embeddings, the system captures the nuanced essence of their needs, enabling semantic similarity matching for enhanced accuracy. In the realm of data retrieval and processing, our system excels through its utilization of MongoDB for robust data storage and pymongo for efficient database interaction. Text chunking optimizes LLM processing efficiency, while the integration of FAISS facilitates swift and effective search through the creation of a document search index based on text chunk embeddings. Central to our system's success is its user-friendly web interface built with Flask, which provides a seamless and intuitive platform for users to engage with. Through this interface, potential roommate matches are presented, with highlighted compatibility aspects based on the user's conversation, thereby simplifying the search process and enhancing user satisfaction. Moreover, our system addresses accessibility concerns by catering to users who may face language barriers or struggle to express their needs on static profiles, thereby fostering inclusivity and diversity within the roommate matching landscape.

In conclusion, our LLM-powered roommate recommendation system not only represents a significant advancement in personalized matching solutions but also holds the potential to transform the roommate search experience for a broader spectrum of users. By leveraging the power of LLMs, our innovative approach not only enhances user satisfaction but also strengthens the connections formed between future roommates, ultimately fostering a more harmonious and fulfilling living environment.

7. REFERENCES

[1] Pascarella, E. T., & Terenzini, P. T. (2005). How College Affects Students: A Third Decade of Research (Vol. 2). San Francisco: Jossey-Bass.

[2] Bartmer, A. (2020, April 14). How to avoid rental scams [Blog post]. RentCollegePads.com.

[3] Neuman, W., & Rubin, Z. (2004). Why Can't We Be Friends? Understanding Roommate Conflict Among College Students. Journal of College Student Development, 45(3), 342-357

[4] Prendergast, C., Proulx, S., & Lawler, M. (2014). The Impact of Roommate Conflict on Student Mental Health and Academic Performance. Journal of College Counseling, 17(2), 127-142

[5] Aher, Gati V., Rosa I. Arriaga, and Adam Tauman Kalai. "Using large language models to simulate multiple humans and replicate human subject studies." International Conference on Machine Learning. PMLR, 2023.

[6] Chung, K.-S. (2000). On the existence of stable roommate matchings. Games and Economic Behavior, 33(2), 206-230.

[7] Bredereck, R., Heeger, K., Knop, D., Niedermeier, R. (2020). Multidimensional Stable Roommates with Master List. In: Chen, X., Gravin, N., Hoefer, M., Mehta, R. (eds) Web and Internet Economics. WINE 2020. Lecture Notes in Computer Science (), vol 12495. Springer, Cham. [8] A. Gupta, I. Almeida, H. Balaji and M. Tiwari,
"DORMMATE - A Room-Mate Personality Matching Application," 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2022, pp. 1-5

[9] Rahman, S. Optimal room and roommate matching system using nearest neighbor algorithm with cosine similarity distribution. SSRN Electron. J. 2021, 3869826.

[10] Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. LLM Based Generation of Item-Description for Recommendation System. In Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23). Association for Computing Machinery, New York, NY, USA, 1204–1207.

[11] Ji, J. et al. (2024). GenRec: Large Language Model for Generative Recommendation. In: Goharian, N., et al. Advances in Information Retrieval. ECIR 2024. Lecture Notes in Computer Science, vol 14610. Springer, Cham.

[12] Spurlock, Kyle Dylan, et al. "ChatGPT for Conversational Recommendation: Refining Recommendations by Reprompting with Feedback." arXiv preprint arXiv:2401.03605 (2024).

[13] Zhang, Jizhi, et al. "Is ChatGPT fair for recommendation? evaluating fairness in large language model recommendation." Proceedings of the 17th ACM Conference on Recommender Systems. 2023.