AVE Trends in Intelligent Computing Systems

FindMyDorm: A Deep Reinforcement Learning Approach for Personalized Recommendation Systems

T. S. Abiraami[1], R. Hari[2], S. Surya[3], R. Angeline[4,*], Sureshkumar Somayajula[5]

[1,2,3,4]Department of Computer Science and Engineering in AIML, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[5]Department of Computer Science and Technology, Sunlife Canada Financials, Toronto, Ontario, Canada.
at2742@srmist.edu.in[1], hr6416@srmist.edu.in[2], ss5749@srmist.edu.in[3], angelinr1@srmist.edu.in[4], suresh.kumar.somayajula@sunlife.com[5]

*Corresponding author

Abstract: The process of finding appropriate hostel accommodation for students is a very important and time-consuming task that students in universities need to undertake, entailing the analysis of a variety of constraints and requirements that need to be met, including affordability, proximity to institutions of learning, security, and availability of basic amenities. There is a primary reliance on a straightforward listings-based strategy that does not provide relevant or legitimate information when used with existing web solutions for searching for college hostels. FindMyDorm is a web-based dormitory recommendation system developed for this research paper. Its purpose is to facilitate efficient, individualized housing searches. To address the problems raised, this approach was designed to overcome them. The system presented aims to create recommendations that are pertinent by enabling the systematic collection and processing of user-specific input on geographical location, financial constraints, required facilities, and proximity to educational institutions. This enables the formulation of relevant recommendations.

Keywords: Deep Reinforcement Learning; Personalized Recommendation System; Student Accommodation; Sequential Decision-Making; Sentiment Analysis; Popularity Bias Mitigation.

Cite as: T. S. Abiraami, R. Hari, S. Surya, R. Angeline, and S. Somayajula, "FindMyDorm: A Deep Reinforcement Learning Approach for Personalized Recommendation Systems," *AVE Trends in Intelligent Computing Systems*, vol. 3, no. 2, pp. 77–85, 2026.

Journal Homepage: https://avepubs.com/user/journals/details/ATICS

Received on: 07/04/2025, **Revised on:** 02/08/2025, **Accepted on:** 23/09/2025, **Published on:** 05/05/2026

DOI: https://doi.org/10.64091/ATICS.2026.000257

1. Introduction

The rapid rise of digital platforms has significantly changed how users search for products and services. Areas such as housing, social media, and e-commerce can leave users feeling overwhelmed by the sheer number of choices and lack of clear navigation. This situation, known as information overload, makes it hard for users to find options that fit their individual preferences, limitations, and situational needs. Different approaches may appear faster, but they cannot cover all the angles or dimensions of dynamic user preferences. User-generated data, such as ratings and reviews, requires better methods, including collaborative filtering, content-based filtering, and hybrid recommendation models. These methods will be very useful because they consider the user's personal and past preferences and provide search results accordingly. Therefore, this will serve as a better, more optimal method that provides personalized results. This can be very efficient. In modern times, recommendation engines have grown significantly, as they are found in almost all apps that researchers use, and various factorization and boosting methods have been very useful for building optimal algorithms. Many platforms, such as search engines and social media apps, operate on the principle of providing personalized recommendations to each user. They use this to personalize the user experience as much as possible by providing relevant content based on their search history. Providing information that is necessary to the user, such as the location, pricing, neighborhood, cleanliness, etc., is essential to make the user feel more comfortable, since this is an application focused primarily on addressing the concerns of a student, and including all this information is crucial. An important part of research in the recommendation engine field is the use of data from real-time users, such as real-time reviews and feedback. The main goal is to improve preference modeling.

The information from the user reviews has tremendous value that goes beyond numerical ratings. Analyzing user reviews and reflecting on their quality of search results can help the system understand users' diverse views and needs, as well as the subtext in the reviews, thereby improving the quality of recommendations. On top of that, recommendation approaches that combine multiple domains aim to leverage cross-domain knowledge, helping address data sparsity and enabling better personalization for new users. In recent years, conversational recommendation systems have tended to perform better by combining dialogue systems with recommendation engines. Conversational systems aim to improve user engagement. User preferences are often taken into consideration to provide better recommendations. Systems that are interactive in this manner are useful in complex situations where accommodations need to be selected, when users cannot list all their preferences at once. In a wide range of research, developing advanced recommendation systems for diverse users has become increasingly important. Systems that are designed for student accommodations and user personalization, etc which, establish the importance of recommendations and optimized solutions. Students searching for accommodations often fail to find something that caters to their needs. This has become a huge problem for modern-day students seeking accommodations near their respective universities.

The atmosphere and the environment of a student are extremely important for personal growth. Recommendations tend to be vague and do not cater to individual needs or students' expectations. If a student wants to pursue a course at a particular university, they might prefer to stay in private accommodations in the university's neighborhood. Different students have different needs. One student might prioritize the neighborhood, some might have dietary requirements for the food and refreshments served in the private accommodation, and some might prioritize spacious rooms. Therefore, each student must receive recommendations based on their top priorities. Modern-day recommendation websites often try to define what a good accommodation is and do not personalize their recommendations based on users' personal needs, leaving users confused when choosing a place to stay. This is why a change in recommendation engines is required across domains to cater to the needs of every user. Despite significant improvements, there is a growing demand for solutions that integrate multiple approaches, including effective data collection, innovative machine learning methods, and user-friendly design, to deliver trustworthy, personalized recommendations. The inspiration for this paper is to provide an efficient system that can be a great help to students searching for accommodations tailored to their requirements.

2. Literature Review

A lot of research has been done on recommendation systems. The main goal is to build a robust platform that is accurate and delivers personalized results to students. This literature survey reviews previous research in this domain. Su and Khoshgoftaar [1] researched the relationship between the product and the user’s preferences by collecting information from various users in the form of a survey. This work serves as the basis of personalized recommendation systems. This has had a major impact on the field of recommendation systems, which have revolutionized the provision of user recommendations based on individual preferences. This way, each user can have an experience that satisfies their requirements. Koren et al. [2] developed a primitive filtering approach for search results and sought to optimize it to produce results efficiently. This work proved to be highly efficient while reducing computational requirements. This can be extremely helpful when dealing with large-scale implementations. Since older filtering methods were not as efficient as they needed to be, this research introduced a perspective that combined multiple filtering techniques. At the same time, it reduces the computational resources required. This served as a major inspiration for further recommendations of engine models. Pazzani and Billsus [3] experimented with various filtering methods and developed a hybrid recommendation model. This proved to be very efficient in terms of performance and in delivering personalized user content across various scenarios. Since users may have different preferences, combining filtering methods helps us achieve a result that combines the best features of each.

This way, a highly efficient system can be built. Lin [4] conducted research in the academic field using clustering techniques and various filtering methods to represent it as vectors. This reduced dimensionality, and the recommendations proved very efficient and accurate, providing significant help for large-scale implementations. Wang and Zhang [5] studied how machine learning, clustering methods, and related techniques play a significant role in social media algorithms and their recommendation engines. Often, social media algorithms use clustering techniques, which provide each user with a personal bubble of recommendations tailored to their needs and preferences. This was a great help in concluding that machine learning models can understand various patterns and provide user-specific feedback, which can be very useful in this field. This being one of the most widely used applications of recommendation engines, it serves as a huge model for upcoming or developing models. Han et al. [6] highlighted the importance of integrating unsupervised and supervised learning methods for user behavioral analysis. Combining different learning methods can be highly beneficial, providing a unique experience for each user. By combining supervised and unsupervised learning methods, researchers can extract key attributes from each technique, yielding an optimized learning model. The research demonstrated that in terms of user satisfaction and prediction accuracy, hybrid learning methods are more effective than traditional heuristic-based recommendation techniques.

McAuley and Leskovec [7] emphasized the need to combine users' text reviews for this. This was a great way to understand what users prefer; it was an efficient way to study and analyze patterns from actual text reviews. This could be used to personalize each user's recommendations as accurately as possible. This way, the recommendations would feel more relevant to the user. This led to improved personalized recommendations, providing a better understanding of context and preference-based ratings. Choi and Kim [8] developed review-based recommendation models for cross-domain environments. Through knowledge transfer, a key factor in understanding user needs, the study demonstrated that user reviews can effectively reduce cold-start issues and improve recommendation accuracy across various domains. Often, users encounter results that are not relevant to their needs, leading to cold-start issues. Through this study, researchers can see how internal domain knowledge transfer can enhance user experience and improve the accuracy of the recommendations. Wu et al. [9] specifically focus on recipe recommendation platforms in the domain-specific recommendation models. Such complex processes require extremely quick adaptation techniques. This is because the amount of data available for processing is always limited in every instance. Therefore, it can become difficult to offer extremely specific recommendations for the limited data available.

The study integrates collaborative filtering and content analysis and highlights the role of hybrid models that combine rule-based constraints to satisfy dietary and nutritional requirements. Li [10] proposed a content filtering method based solely on the web, incorporating ensemble learning and combining it with various clustering methods. This ensured the model was efficient, could make better recommendations, and could be used in real-world applications. The amount of information available on the web is enormous, and it can therefore be used as context to predict user preferences when limited data is available. Categorizing users by common preferences is a great way to offer personalized recommendations when data is limited. This study offers insights into how to efficiently provide recommendations using different learning techniques. Zhang et al. [11] developed a model that combined the characteristics of both dialogue moderation and recommendation engines. This served as a base for future developments of a model with conversational capabilities, which can improve the user experience. This can be an efficient way to engage users and fulfill their needs. Various Social media platforms in modern-day seem to be adopting this integrated modeling technique to deliver optimized results without compromising professionalism and ensuring a pleasant user experience. The research conducted here was a great inspiration for the development of modern-day recommendation engines used on popular platforms and search engines. Wu et al. [12] investigated the domain of making recommendation models more conversational, so that this conversational model would help provide a personalized experience for users using a data structure called a knowledge graph.

The use of a data structure such as a knowledge graph, when combined with a conversational model, can be extremely efficient, as it is often used in the development of interactive chatbots on various platforms to assist users with their grievances. By adapting this method, providing personalized recommendations to each user can become much simpler. This study offers a unique perspective on recommendation engine models. Liang et al. [13] proposed a method for integrating attention networks with recommendation engines. This way, the model can learn from users' preferences through its attention mechanism while remaining unbiased. The key is being able to relate the user's past preferences to the current state. Establishing a relationship between the user's current and past preferences can be highly efficient for improving the user experience and delivering accurate results that meet the user's requirements. This is a way for researchers to make recommendation engines suitable for sound decision-making. Pazzani [14] identified difficulties in recommendation systems, including data trustworthiness and applicability to real-world problems. In theory, personalizing the user's content based on their needs and curating it might seem simple enough with the available data. Still, it is important to consider open-source information when building a recommendation engine. This work focused on integrating multiple domains to improve the system's reliability. Kazemi et al. [15] sought to examine the modifications and improvements in how recommendation engines have evolved, as well as the difficulties that need to be addressed. This research/study demonstrated the importance of considering the applications of algorithms to real-world problems.

3. Proposed Methodology

3.1. Data Collection and Integration

FindMyDorm will be a system for collecting and combining student accommodation data, organized and partially organized from various sources. The sources include the private residence records near universities, public rental listings, government property databases, and user reviews. The FindMyDorm system uses a data pipeline that adds details such as location, price ranges, amenities, neighborhood, and safety information. Researchers start with preprocessing to ensure the data is consistent and reliable. The preprocessing stage ensures the records have units, consistent formats, and the same timestamps. The preprocessing stage removes entries and redundant listings using validation checks and duplicate-removal steps. The preprocessing stage also cleans the user reviews' text. The procedures clean the text, split it into words, remove stop words, and convert words to their base forms. The structured preparation helps with topic modeling. Researchers also note that structured preparation improves sentiment analysis and enables me to evaluate the results. Structured preparation is the key to the evaluation.

3.2. Platform Architecture and User Flow

FindMyDorm uses a web-based design that runs on the stack (React, Supabase). The user interaction flow starts with registration and authentication. The user interaction flow then moves to an onboarding process that asks the user for preferences. The user tells FindMyDorm about budget constraints, proximity to campus, room-sharing requirements, gender-based housing needs, and lifestyle preferences. As shown in Figure 1, the information is sent to the backend system via APIs. The recommendation service processes the inputs. The backend pulls listings from the database. The backend returns a ranked list of accommodations. Each listing carries metadata. The metadata includes images, utility charges, property details, neighborhood details, and combined ratings. The metadata lets users make informed decisions. This ensures they can use the platform because it supports actions. The platform lets you search in time and filter results. This platform can be very helpful because users can find many options and choose the ones that best meet their needs.

3.3. Recommendation System Design

The recommendation engine is a part of FindMyDorm. At first, users can set a bunch of filters based on their priorities. Some people might prioritize hygiene, others might prioritize food options; similarly, neighborhood, safety, cleanliness, budget, etc. By allowing users to prioritize what they want, researchers can provide a personalized experience.

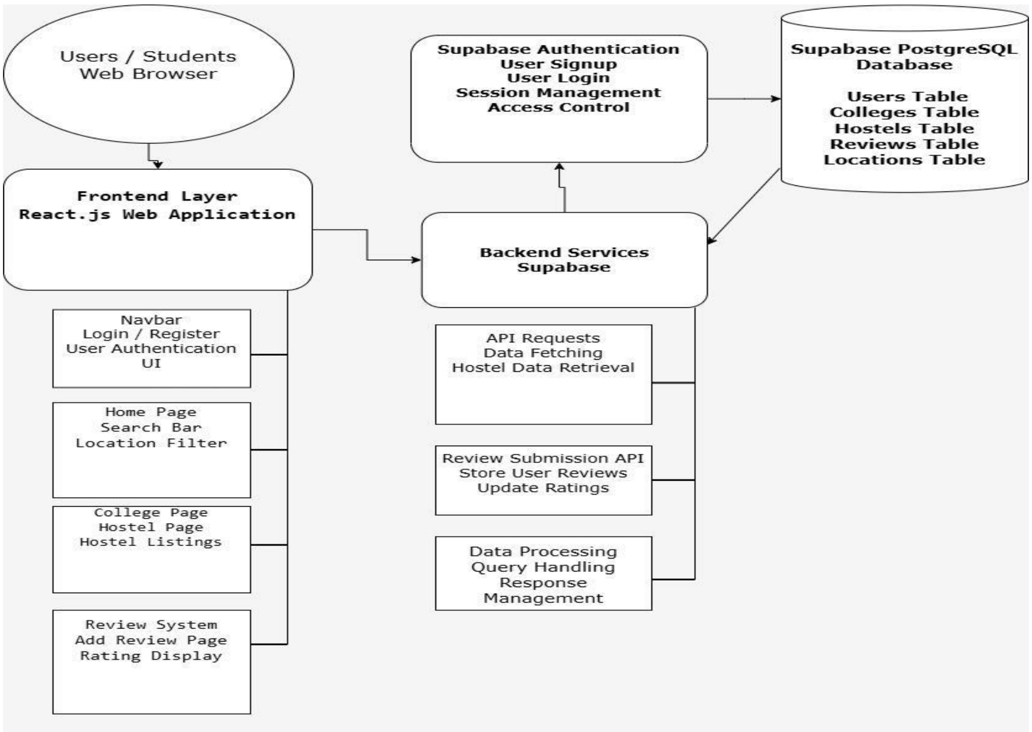

Figure 1: Architecture diagram

The recommendation engine uses preferences such as safety ratings and proximity to transportation. The recommendation engine ranks the listings based on those scores. In the phases, researchers add the hybrid recommendation technique that combines content-based filtering and collaborative filtering. Content-based filtering matches user preferences with listing attributes. Collaborative filtering uses interaction data, such as clicks, ratings, and browsing history, to identify patterns of similarity among users. The matrix factorization uncovers hidden relationships between the users and the dorm listings. The matrix factorization then helps the system suggest the accommodations that the users prefer. To improve personalization, researchers add a reinforcement learning algorithm to the model. The system treats recommendations as a step-by-step decision-making process. The user profile and the interaction history become the system state. The recommended listings become the actions. The user engagement signals, such as clicks, saves, inquiries, and bookings, become the reward signals. The algorithm gets updated with each step. The RL agent continuously updates the policy. This updates the policy to maximize user satisfaction over time. This will ensure that the feedback mechanism guides the system and that the system refines the strategies and improves the recommendation accuracy as more interaction data arrives.

3.4. Review Processing

Researchers read user-generated reviews to get ideas for dormitory listings. Researchers use computer tools that read text to extract sentiment scores, topics such as environment, supportive landlord, and key keywords. Researchers would then extract sentiment scores, topics, and keywords from user-generated reviews. Researchers then combine the extracted features. Show the extracted features as review summaries, as sentiment signs, or as word clouds. The review summaries, sentiment indicators, and word clouds help prospective tenants quickly read the information. The review processing is carried out by the reinforced PG Ranking algorithm in the following manner:

Final Score = (0.4 × Average Rating) + (0.25 × Category Score) + (0.2 × Authenticity Score) + (0.15 × Proximity Score)

(This model reinforces trusted reviews and student-relevant factors while still considering distance from campus.)

Student Preference Optimization Preference Score = (Food × w_1) + (Safety × w_2) + (Cleanliness × w_3) + (Rent Transparency × w_4)

(Where $w_1 + w_2 + w_3 + w_4 = 1$, depending on student priority.)

In later versions, researchers will add the aspect-based sentiment analysis so that the system can connect sentiment scores to each dorm attribute. The dorm attributes include internet speed, cleanliness, security, and facility maintenance. Aspect-based sentiment analysis will give a view of the feedback. The exact view of the feedback will help the recommendation engine. The recommendation engine will use user experiences when ranking the dorms. The recommendation engine will become stronger because user experience is now part of the ranking.

3.5. Testing, Evaluation, and User Feedback

The system uses a check method based on the user's domain when they try to sign in. If the user has an academic domain such as “edu.in”, the system labels them as a verified user. If they don’t have an academic domain in their email, they will not be labeled as a verified user. The reviews can be trustworthy. Researchers run performance tests on the system by monitoring search delays, API response times, and system throughput. Performance testing on the system measures how quickly the search returns results, how quickly the API responds, and how much work the system can handle at once. Researchers measure the effectiveness of recommendations within the system using information retrieval scores. The effectiveness of recommendations in the system tells me how well it recommends items. User reviews are collected to assess the originality of the recommendations. Information gathered from various reviews will be integrated into a continuous development cycle to improve the personalization of recommendation results.

4. Results and Discussion

FindMyDorm is a system that helps users choose accommodation based on their preferences without being overwhelmed by vast amounts of information. Each listing includes main details such as the rental price, location, available amenities, safety ratings, and user reviews. This helps users to identify and shortlist their PGs/Hostels based on their key requirements. As everyone has their own priorities when selecting PGs/Hostels, this feature lets users choose the best fit for their requirements.

Table 1: University database

ID	Name	Location_id
618f8fb4-ab0d-4a7c-ae28-c44cb8c7ccd1	SRM Institute of Science and Technology,	NULL
659dbc23-34dc-43f6-ae15-cf52b85c1512	Indian Institute of Technology, Madras	NULL
806c1628-de85-424a-9785-289341c22087	Anna University	NULL
9b623597-0dbc-4b2d-aaeb-8d468b7504c	SRM Institute of Science and Technology	90a2daeb-6de6-4b2c-b290-90ba...
d8984a1c-7b90-4681-80ac-8eaeece0ad68	Vellore Institute of Technology, Chennai	NULL

Table 1 shows the database where university information is stored. As shown, each university has a unique ID in the Supabase DB that can be used to identify that university. This comes into play when users are searching for accommodations near their desired university. As Researchers map each university using a primary key or unique key, researchers ensure there are no duplicate listings and that the search for PGs/Hostels can be seamlessly centered on the desired College. This also helps us map all PGs/Hostels within the proximity of a particular college to their respective college search. Table 2 is a database containing details of private accommodations, such as name, address, price range, and the accommodation's status on our website. The status column is for authorizing any new accommodation requests from the users. This way, the details of accommodations can be listed clearly when hosted on the frontend. Such details can help users choose an accommodation based on their needs. This helps us store all the relevant details users need to make fair decisions without unauthorized influence. When researchers receive a request to add a PG, they verify the details, then set the status to approved, ensuring students can select trusted PGs/Hostels with legitimate details.

Table 2: Accommodation database

Name	Address	Price_Min.	Price_Max.	Status
Rainbow Men's Hostel / Branch-2	No1/3a, 14th Cross St, Venkateshwara Ne	5000	5000	Approved
Frankfurt Living PAVO	Valluvar Salai, near Grace Super Market,	5000	12000	Approved
Kms Men's Paying Guest	No 1/39C, Near DLF and SRM College, Ve	6000	15000	Approved
RSD HIFI PG FOR MEN	No 2/6, Kalaignar Street, Shanthi Nagar,	5850	16999	Approved

Table 2 shows the database used to store user details for platform sign-ups. Based on the details they provide during registration, they will be categorized as verified or non-verified users in the database. If the user’s email address has “. edu.in”, their profile will be categorized as true in the is_verified column, and if not, it will be categorized as false in the is_verified column. This ensures researchers lend credibility to students who post reviews, and that users know an actual student has posted the review, making their decision-making process easier.

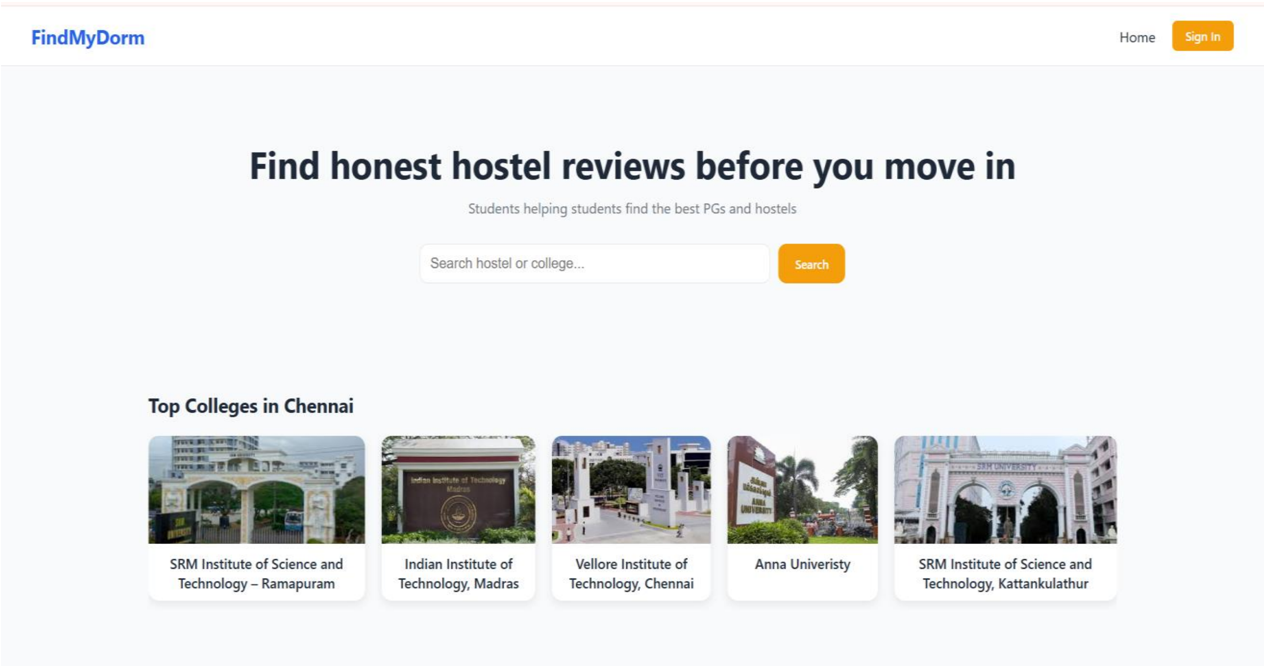

Figure 2: Homepage

Figure 2 and Figure 3 represent the workflow of the website, Figure 2 is the homepage of the website where the universities are listed so that the users can choose the university and search for accommodations in that particular neighbourhood they can either select one the displayed top College or if they don't find their College in that list they can use the search bar at top to find their specific College once they click on their respective College they can explore PGs around their College and as it can be seen from Figure 3, once the users selects one of the universities, the list of PGs/Hostels near that particular universities are listed. Users can browse for details depending on their needs.

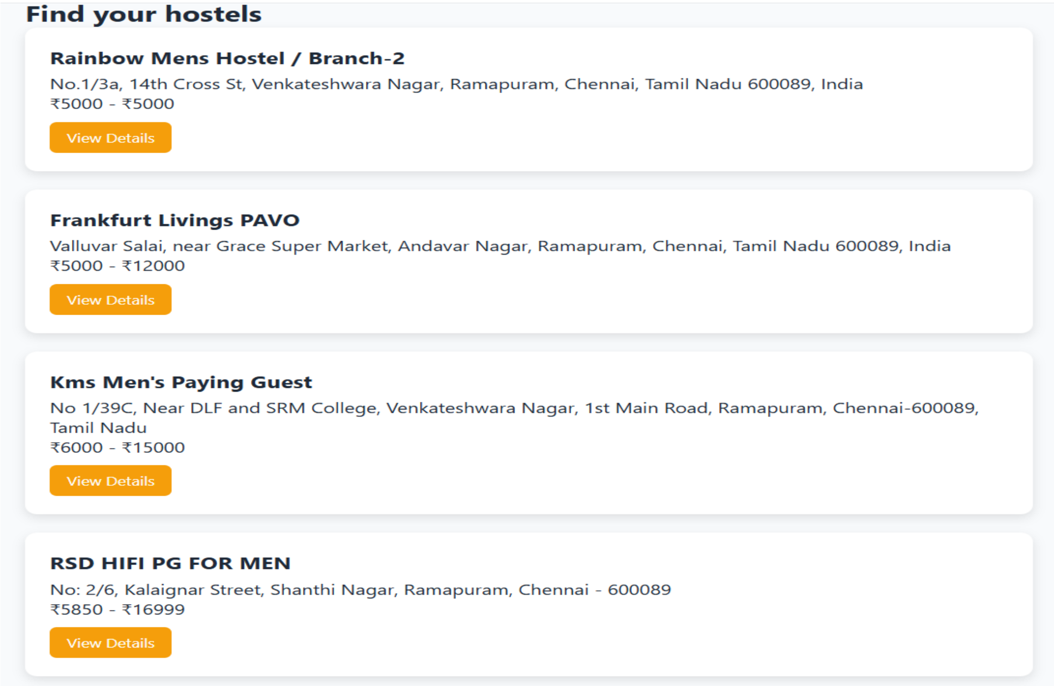

Figure 3: Hostel listings

The image shows that the surrounding PGs/Hostels are listed with relevant details, including name, address, price range, and a button to view further details.

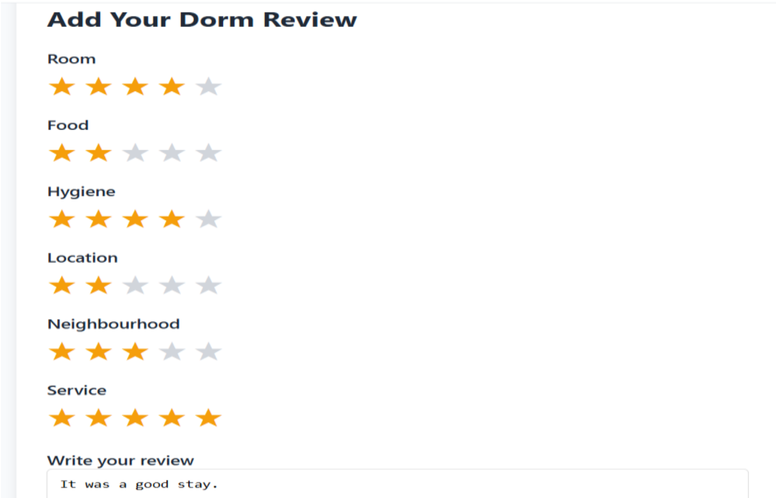

Figure 4: Review page

Figure 4 is the review page where users can rate the accommodation or stay based on five different criteria, such as the room, food, hygiene, location, service, and then write out the text review to make sure that they can explain everything that they want to. This way, the reviews can become transparent and reliable. When more users are browsing for accommodations, they can use reviews from other users to make their decision. As more users leave reviews, it becomes easier for new users to make more informed choices.

Table 3: Tabulation of results

PG Name	Distance from SIM	Avg. Rating	Bed	Cleanliness	No. of Reviews
Green Nest PG	0.8	4.2	4.1	4.1	25
Sai Residency PG	1.2	3.8	3.6	3.6	25

Comfort Stay PG	0.6	4.5	4.5	4.3	40
Sri Lakshmi PG	1.5	3.6	3.5	3.5	18
Metro Living PG	0.9	4.1	4.0	4.0	29
Ramapuram PG	0.7	4.3	4.2	4.1	36
City Comfort PG	1.8	3.9	3.6	3.5	20
Sunshine Residency	1.0	4.0	3.9	3.8	27
Lakshmi Ladies PG	0.5	4.4	4.4	4.2	38
Urban Nest PG	1.3	3.9	3.8	3.8	22

Table 3 presents an evaluation of multiple private accommodations based on publicly available information. This analysis of existing information about various accommodations helps understand user needs and how they prioritize different categories based on their personal preferences, and serves as a useful tool to improve service delivery to users.

5. Conclusion

FindMyDorm is a platform that helps students find off-campus housing. It makes finding a place to live easier. FindMyDorm combines information about housing and users' opinions of it with smart computer methods. All of this is on a website built with ReactJS and Supabase. The platform uses some ways to help students. It uses filtering to narrow down options. It also uses hybrid recommendation modeling and Deep Reinforcement Learning to suggest housing. This means FindMyDorm gets better at suggesting housing all the time. FindMyDorm provides accommodation suggestions tailored to each student. One important thing about this research is that it makes the recommendations more personal. Most recommendation systems use the old information about what a user likes. Our system looks at recommendations as a series of choices. It uses user behavior to improve recommendations. The system keeps updating its list of recommendations in real time, trying to keep the user interested for a long time and make them happy. The recommendation system is always trying to achieve this. The results of the experiment show that this adaptive approach makes recommendations more consistent and eliminates irrelevant options in subsequent sessions. This shows that the personalized platforms researchers use today need to be updated constantly. The adaptive approach is good at doing this. It helps personalized platforms give recommendations to users. The adaptive approach is very important for the platforms. This helps us understand how people feel and what they are talking about.

Acknowledgment: N/A

Data Availability Statement: The relevant data and supporting materials may be accessed through the corresponding author upon reasonable request and in accordance with repository guidelines.

Funding Statement: The authors collectively confirm that no external financial support, sponsorship, or grant funding was received for conducting this research or preparing the manuscript.

Conflicts of Interest Statement: All authors declare that there are no conflicts of interest, financial or personal, related to this study. Proper acknowledgment, citations, and references have been included for all sources and materials used in the research.

Ethics and Consent Statement: The authors affirm that necessary permissions, institutional approvals, and participant consents were obtained during the data collection process. The study was conducted in compliance with accepted ethical standards and research guidelines.

References

1. X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in Artificial Intelligence*, vol. 2009, no. 1, pp. 1–19, 2009.
2. Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
3. M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The Adaptive Web, *Springer*, Berlin, Heidelberg, 2007.
4. N. Lin, “Personalized book intelligent recommendation system design for university libraries based on IBCF algorithm,” *IEEE Access*, vol. 12, no. 6, pp. 82015–82032, 2024.
5. J. Wang and J. Zhang, “Application of machine learning algorithms in user behavior analysis and a personalized recommendation system in the media industry,” *Journal of ICT Standardization*, vol. 13, no. 1, pp. 41–66, 2025.

6. J. Han, M. Kamber, and J. Pei, “Data Mining: Concepts and Techniques,” 3rd ed., *Morgan Kaufmann*, Burlington, Massachusetts, United States of America, 2012.
7. J. McAuley and J. Leskovec, “From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews,” *arXiv:1303.4402*, 2013. [Accessed by 12/02/2025].
8. Y. Choi and C. K. Kim, “Generalization of knowledge transfer with user reviews for cross-domain recommendation,” *IEEE Access*, vol. 13, no. 10, pp. 172954 –172971, 2025.
9. Y. Wu, L. Kang, Q. Guo, and L. Zhao, “A review of recipe recommendation methods,” *IEEE Access*, vol. 13, no. 5, pp. 92483 –92494, 2025.
10. X. Li, “Design of a web content personalized recommendation system based on collaborative filtering improved by combining k-means and LightGBM,” *Advanced Practice in Web Engineering in Asia*, vol. 24, no. 2, pp. 267–290, 2025.
11. Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft, “Towards conversational search and recommendation: System ask, user respond,” in *Proc. 27th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, Torino, Italy, 2018.
12. F. Wu, G. Zhao, T. Li, J. Shen, and X. Qian, “Improving conversational recommendation systems through personalized preference modeling and knowledge graph,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8529 – 8540, 2024.
13. R. Liang, Q. Zhang, J. Wang, and J. Lu, “A hierarchical attention network for cross-domain group recommendation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3859 – 3873, 2024.
14. M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artificial Intelligence Review*, vol. 13, no. 5, pp. 393–408, 1999.
15. F. Kazemi, M. Lamothe, and S. McIntosh, “Characterizing the prevalence, distribution, and duration of stale reviewer recommendations,” *IEEE Transactions on Software Engineering*, vol. 50, no. 8, pp. 2096 – 2109, 2024.