

MyHouseHelp: An Intelligent Marketplace for On-Demand Services

Aditya Shinde¹, Suraj Fulare², Janhavi Patil³, Cheyansh Sahu⁴, Sharvari Tele⁵,
Dr. M. A. Potey⁶, Prof. S. Y. Arude⁷

Students, Computer Engineering, D. Y. Patil College of Engineering, Akurdi, Pune, India¹⁻⁵
Dean Collaborations, Computer Engineering, D. Y. Patil College of Engineering, Akurdi, Pune, India⁶
Assistant Professor, Computer Engineering, D. Y. Patil College of Engineering, Akurdi, Pune, India⁷
aditya.shinde@gmail.com, surajfulare123@gmail.com, janhavipatil4103@gmail.com,
cheyanshsahu2004@gmail.com, sharvaritele2964@gmail.com

Abstract: *The urban home services sector is largely unorganized, presenting significant challenges for customers in finding reliable, skilled, and fairly-priced professionals. MyHouseHelp is a unified, intelligent platform that connects verified service providers with customers for a wide range of household needs. The system integrates a robust three-tier architecture for user management, service booking, and secure payments. At its core, MyHouseHelp leverages machine learning to enhance the user experience and operational efficiency. This paper presents the system's design, methodologies, and the specific algorithms used to deliver a trustworthy and intelligent home service marketplace.*

Keywords: On-Demand Services, Recommender Systems, Service Marketplace, Machine Learning, Django.

I. INTRODUCTION

A. Background

The rapid pace of urbanization has prompted the related demand for household services, such as cleaning, plumbing, and electrical services. Customers often experience challenges locating trustworthy and competent service professionals, knowing what the likely pricing will be in an overly opaque market, and being confident in the quality of service. Traditional processes such as word-of-mouth referrals or local phone book advertisements are often ineffective or can be unreliable. MyHouseHelp is designed to offer an easy-to-use way for the marketplace to address these issues through the establishment of a centralized digital marketplace. The platform provides an effective way for customers to identify, book and pay for services. For service providers, the platform provides a structured way to list their skills, manage their schedules, and create an opportunity for achieving broader customer reach. The administrative backend ensures that all providers are vetted and offers tools to manage disputes and the platform. The most significant value-add of MyHouseHelp from a technology perspective is the integration of machine learning to create a smarter ecosystem comprised of data-driven models as opposed to a simple listing platform. A recommendation engine matches users to the best provider for their needs, a dynamic pricing algorithm adjusts provider pricing to reflect fair market rates, sentiment analysis of a customer's review provides insight on maintaining a high quality of service, and demand forecasting informs a proactive allocation of resources. The platform leverages machine learning to enhance user experience through intelligent recommendations, dynamic pricing, and sentiment analysis.

B. Problem Statement

Despite the growing demand for home services, many issues still persist in the current ecosystem:

- Small and independent service providers have no affordable digital infrastructure to list services, manage bookings, track analytics, and receive payments without depending on third-party aggregators.
- Platforms operating in a closed, proprietary ecosystems without any transparency.



• Centralized platforms imposing higher service commissions resulting in creating an inequitable financial structure that disproportionately disadvantages independent workers who already operate on thin margins. These restrictions call for an intelligent, transparent, and open home service marketplace that empowers independent providers and consumers equally.

C. Contribution of Proposed System

The proposed MyHouseHelp system addresses these challenges by:

- Implementing a role-based access system for Customers, Providers, and Admins.
- Enabling end-to-end booking lifecycle management from creation to completion.
- Integrating a UPI-based payment flow with QR code scanning and confirmation.
- Incorporating ML-powered features including service recommendations, sentiment analysis on reviews, demand forecasting, and dynamic pricing.
- Providing providers with analytics dashboards for earnings, ratings, and booking trends.

II. RELATED WORK

A. Traditional Platforms

Earlier, home service solutions were dependent upon classified listing websites such as JustDial and Sulekha, where customers could discover providers but had no booking management, payment integration, or verification mechanisms. While useful for discovery, these platforms offered no structured workflow and placed the entire burden of trust and transaction on the end users.

B. Aggregator-Based Platforms

Modern platforms such as Urban Company and Housejoy introduced structured booking and provider onboarding. However, these systems function as centralized aggregators that impose heavy commission structures, restrict provider pricing autonomy, and offer limited transparency in dispute resolution and review moderation. Additionally, ML-driven features such as demand forecasting and dynamic pricing remain inaccessible to individual providers.

C. Research Gap

- No direct peer-to-peer UPI payment support between customer and provider.
- Absence of provider-level ML tools for pricing and demand prediction.
- Independent service providers have no affordable platform to manage services end-to-end without commission dependency.

III. SYSTEM ARCHITECTURE

A. Complete Design

MyHouseHelp follows a layered MVT (Model-View-Template) architecture built on Django 4.2.7:

- Presentation Layer: HTML5 templates extending base.html with a unified indigo-coloured theme, responsive navigation, and a floating chatbot widget.
- Business Layer: Django view functions handling role-based logic across accounts, services, bookings, reviews, payments, and the ML engine.
- Data Layer: SQLite database which manages all entities with the help of Django ORM with a clearly defined foreign key relationships across entire users, bookings, services, reviews, and payments sections.

B. Technology Stack

- Language: Python 3.13, JavaScript (Vanilla).
- Framework: Django 4.2.7.



- Frontend: HTML5, CSS3 with custom CSS variables.
- Libraries: TextBlob, scikit-learn, Google Fonts (Inter).
- Database: SQLite (development).

C. Modules Description

1. Accounts Module

- Supports role-based user registration and authentication (Customer / Provider / Admin).
- Separate admin login for administrative control.

2. Services Module

- Provider services listing with description, category, pricing, and availability management.
- ML-powered recommendations integrated into service browsing.

3. Bookings Module

- End-to-end booking lifecycle: pending → confirmed → in-progress → completed.
- Status history tracking, cancellation, and rescheduling support.

4. Reviews Module

- Multi-dimensional ratings across quality, professionalism, timeliness, and communication.
- Automatic sentiment analysis triggered on every review submission.

5. Payments Module

- Direct UPI-based peer-to-peer payment flow between customer and provider.
- Customer marks payment with optional transaction ID and screenshot; provider confirms receipt.

6. ML Engine Module

- Recommendation Engine, Sentiment Analyzer, Demand Forecaster, and Dynamic Pricing running as separate individual components.

D. Data Flow

Customer Request → URL Router → View Function → ORM Query → Template Render → HTTP Response.

Booking Flow: Service Selection → Booking Form → Provider Notification → Status Updates → Payment → Confirmation.

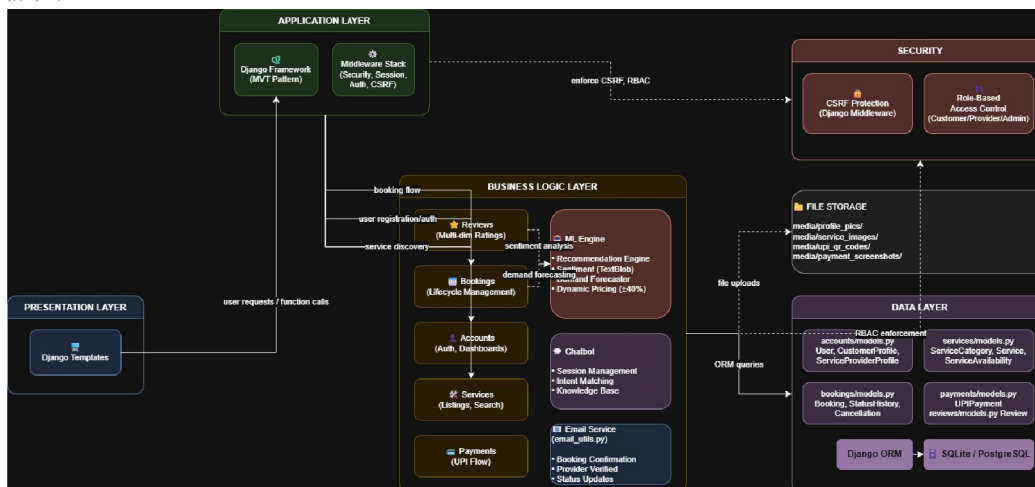


Fig III.1 System Architecture



IV. PROPOSED METHOD

The suggested system optimizes every part of the service experience, from the user’s input to the service’s completion. The system utilizes a friendly user interface and intelligent backend finder to provide ongoing connections for customers with the appropriate professionals for their needs. The service experience begins with a simplified and intuitive user input interface through which the customer provides the services they would like. The workflow involves several key steps:

- 1) **User Input:** While the user interacts with the input interface on the front end, the backend finder begins its work. It uses a machine learning recommendation engine to scan for qualified skilled workers in a network and immediately provide a list of suitable professionals.
- 2) **Intelligent Backend Finder:** The customer is able to view the list of workers available, accompanied by their profiles, rating, and reviews, allowing the customer to choose the best-serving worker based on the provided information.
- 3) **Available Workers List:** Once the customer selects a preferred worker, the system will provide the means to schedule the service at the customer’s convenience. The system will provide an easy way for everyone to communicate and keep everyone in the same understanding.
- 4) **Booking and Scheduling:** Once the customer selects a preferred worker, the system will provide the means to schedule the service at the customer’s convenience. The system will provide an easy way for everyone to communicate and keep everyone in the same understanding.
- 5) **Service Provision:** When the time arrives for the service, the customer will see the selected worker arrive prepared to supply the requested service.
- 6) **End of Service and Feedback:** Finally, after the service is provided to the customer’s satisfaction, feedback and ratings can be provided by the customer, to help to maintain quality and trust in service experience across the platform.

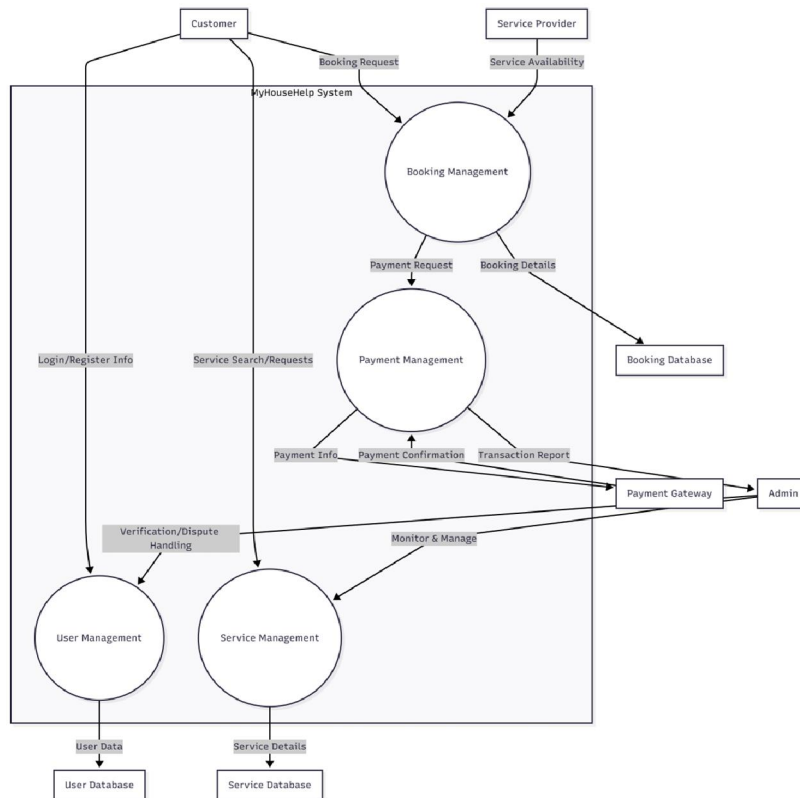


Fig IV.2 DFD Level-1



V. ML MODELS IMPLEMENTATION

A. Recommendation Engine

The recommendation engine in MyHouseHelp is built on collaborative filtering, a widely used technique in recommender systems that identifies patterns from user-service interaction history rather than relying on explicit service attributes.

Input Representation: The system constructs a user-service interaction matrix R where each entry $R(u, s)$ represents the number of times user u has booked service s :

$$R(u, s) = \text{frequency of bookings by user } u \text{ for service } s$$

Similarity Computation: Cosine similarity is used to measure the closeness between two users u and v :

$$\text{sim}(u, v) = (R_u \cdot R_v) / (\|R_u\| \times \|R_v\|)$$

where R_u and R_v are the booking history vectors of users u and v respectively.

Prediction Score: The predicted preference score for user u for an unbooked service s is computed as:

$$P(u, s) = \sum \text{sim}(u, v) \times R(v, s) / \sum |\text{sim}(u, v)|$$

where the summation is over all users v who have previously booked service s .

Cold Start Handling: For new users where no booking history exists, the system falls back to a popularity score:

$$\text{PopScore}(s) = \text{total bookings of service } s / \text{total bookings across all services}$$

Services are then ranked by PopScore and the top-N are recommended to the new user.

B. Sentiment Analyser

The sentiment analyzer is triggered automatically upon every review submission and processes the free-text review provided by the customer.

- **Polarity Scoring:** TextBlob computes a polarity score p for a given review text T using its internal lexicon-based model:

$$p(T) \in [-1.0, +1.0]$$

where -1.0 represents the most negative sentiment and $+1.0$ represents the most positive sentiment.

- **Classification Rule:** The polarity score is mapped to a sentiment label using the following threshold-based classification:

$$\text{Label}(T) = \text{Positive, if } p(T) > 0.1 \text{ Label}(T) = \text{Negative, if } p(T) < -0.1 \text{ Label}(T) = \text{Neutral, if } -0.1 \leq p(T) \leq 0.1$$

- **Subjectivity Score:** In addition to polarity, TextBlob computes a subjectivity score $\sigma(T) \in [0.0, 1.0]$, where 0.0 indicates a fully objective statement and 1.0 indicates a highly subjective opinion. This score is stored alongside the sentiment label to give context to the polarity measurement.

- **Output Storage:** For each review object, the following fields are populated automatically upon submission:

$$\text{sentiment_score} \leftarrow p(T) \text{ sentiment_label} \leftarrow \text{Label}(T)$$

These values are accessible to the platform admin for monitoring provider quality trends and flagging providers with persistently negative sentiment scores.

C. Demand Forecaster

The demand forecaster analyzes historical booking data over a 90-day rolling window to compute a demand score for each service across different time periods.

Historical Booking Frequency: For a given service s , the raw booking count on day d is denoted as:

$$B(s, d) = \text{number of bookings for service } s \text{ on day } d$$

Day-of-Week Pattern: The system aggregates bookings by day of the week to compute a normalized day weight $W(s, \text{dow})$ for day-of-week $\text{dow} \in \{0, 1, \dots, 6\}$:

$$W(s, \text{dow}) = \sum B(s, d) \text{ for all } d \text{ where } \text{weekday}(d) = \text{dow} / \sum B(s, d) \text{ for all } d \text{ in } 90\text{-day window}$$

Seasonal Factor: A seasonal adjustment factor $SF(s, m)$ is applied based on the month m to account for periodic demand fluctuations:



$$\text{AdjustedDemand}(s, \text{dow}, m) = W(s, \text{dow}) \times \text{SF}(s, m)$$

Demand Score Normalization: The final demand score $DS(s)$ is normalized to a $[0, 1]$ scale:

$$DS(s) = \frac{\text{AdjustedDemand}(s) - \min}{\max - \min}$$

A higher demand score indicates periods of peak service demand, allowing providers to prepare resources accordingly and the platform to apply dynamic pricing multipliers.

D. Dynamic Price Engine

The dynamic pricing engine computes a suggested price for each service by applying a composite multiplier to the provider's base price, ensuring market-responsive and fair pricing within defined bounds.

Base Price: Let BP denote the provider-defined base price for a service.

Composite Multiplier: The final price multiplier M is computed as the product of four independent factors:

$$M = M_{\text{demand}} \times M_{\text{competition}} \times M_{\text{urgency}} \times M_{\text{time}}$$

where:

- M_{demand} — derived from the normalized demand score $DS(s)$; higher demand increases the multiplier.
- $M_{\text{competition}}$ — reflects the number of competing providers offering the same service category in the area; more competition lowers the multiplier.
- M_{urgency} — based on the booking priority field; urgent bookings carry a higher multiplier than standard bookings.
- M_{time} — accounts for the time of day; peak hours such as mornings and evenings carry a slightly higher multiplier than off-peak hours.

Suggested Price Computation:

$$\text{SuggestedPrice} = BP \times M$$

Price Bounding: To prevent unreasonable price fluctuations, the suggested price is clamped within $\pm 40\%$ of the base price: $\text{FinalPrice} = \max(0.6 \times BP, \min(1.4 \times BP, \text{SuggestedPrice}))$

This ensures that no service is ever priced below 60% or above 140% of its provider-defined base price, maintaining fairness for both customers and providers while still allowing market-responsive adjustments.

VI. RESULTS

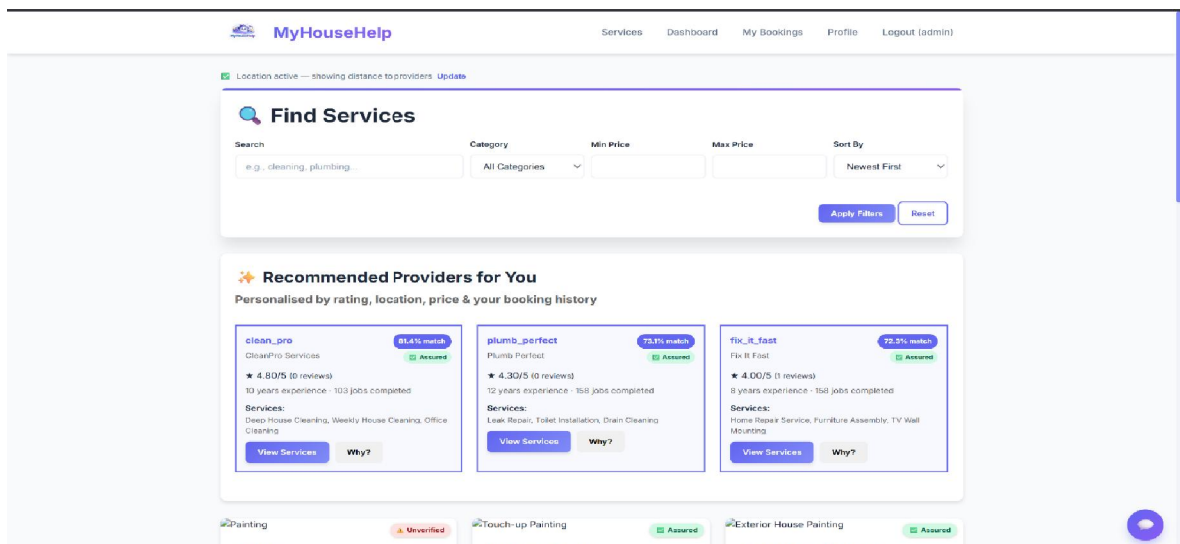


Fig VI.1 Service Browsing



Fig VI.2 Admin Panel

Fig VI.3 Booking Completion



VII. FUTURE WORK

Although the MyHouseHelp system effectively addresses many issues, specific opportunities for future development and research are present in the on-demand domain. The following areas indicate directions for potential future improvement and expansion:

1. **Mobile Application Development:** A cross-platform mobile application can improve the accessibility and usability of the platform. If developed using certain frameworks, such as React Native or Flutter, this task can be accomplished relatively easy. Features such as push notifications, live booking tracking, and offline access can further enhance the overall user experience and in-app interaction.
2. **Advanced AI-Driven Features:** Future work can utilize deep learning models for improved recommendation accuracy and dynamic pricing of services. Additionally, intelligent chatbots utilizing Natural Language Processing (NLP) can also assist in automating a portion of the customer interaction processes, as well as offer service support based on predefined procedures.
3. **Enhanced Security and Privacy Controls:** Future development can also focus on stricter security mechanisms, such as multi-factor authentication, end-to-end encryption or even using AI-based fraud detection, to ensure protection of user data while maintaining integrity of the system.
4. **Multilingual and Regional Support:** Expanding language options, regions, interfaces or even organization-specific interfaces can make the platform more agile and accessible, which encourages user adoption across various regions into the domestic-gig-economy sector.
5. **Scalability through Cloud Infrastructure:** Future migration to a primarily cloud-native system can enhance the platforms scalability, availability and overall increased fault tolerance during “high-use” time period, using systems such as AWS, Azure or Google Cloud.

VIII. CONCLUSION

The MyHouseHelp on-demand web application’s development is a major step forward in the continuing development of service delivery and technology. MyHouseHelp incorporates a collaborative space for customers, administrators, and workers while meeting the challenges posed by a disorganized actual home services industry. The primary aspects of the application such as user-friendly interfaces and profiles; as well as machine learning elements work together to create a better user experience which brings convenience, transparency, and trust – all integral elements of useful on-demand services. The application also possesses a scalable architecture that would allow it to grow and change as it responds to user needs which is a kind of flexibility that can constantly be addressed across a wide variety of user considerations. MyHouseHelp allows users to discover solutions for challenges at their home for the least amount of cost and in the least amount of time form factor.

ACKNOWLEDGEMENTS

The authors sincerely thank the Department of Computer Science and Engineering of our institution for providing the necessary support and infrastructure to complete this project. We express our heartfelt gratitude to our project guide for their continuous guidance, encouragement, and valuable suggestions throughout the development of this work. Additionally, we would like to express our gratitude to the Django, TextBlob, and scikit-learn developers as well as the open-source community for providing the tools and libraries that enabled the creation of MyHouseHelp. We thank the Python ecosystem’s contributions, whose frameworks and tools served as the foundation for this platform.

REFERENCES

- [1]. K. ManojSenthil, P. Nirmala Devi, S. Ravi, E. S. Kirushnamurthy, D. Kowsik, and S. Jayasurya, (2024). ‘On-Demand Home Service based Web Application’. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT).



- [2]. K. Aravindhan, K. Periyakaruppan, T. S. Anusa, S. Kousika, and A. L. Priya, (2020). 'Web Application Based On Demand Home Service System'. In 2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 1458-1462.
- [3]. M. Chowdhury and A. Ahmad, (2023). 'Cheer Up: An Advanced Mobile Application Featuring Chef Selection, Recipe Search, and Chef Assistance facilities for the Citizens of Bangladesh'. In 2023 International Conference on Communication, Circuits, and Systems (IC3S).
- [4]. A. Chakraborty, G. Kathuria, M. Chawla, and D. Chakraborty, (2022). 'Urban Company: The Road from Local to Global'. Delhi Business Review, Vol. 23, No. 2, pp. 81-85.
- [5]. J. Yu, et al., (2021). 'Collaborative Filtering Recommendation with Fluctuations of User' Preference'. In 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE), pp. 222-226.
- [6]. R. N. Tortor, R. T. Alviar, E. E. G. Dacian, P. A. R. Pelaez, and J. L. Sales, (2024). 'DSERV: Home Service Booking System with Geolocation for Serviceable Area Detection'. In 2024 22nd International Conference on ICT and Knowledge Engineering (ICT&KE).

