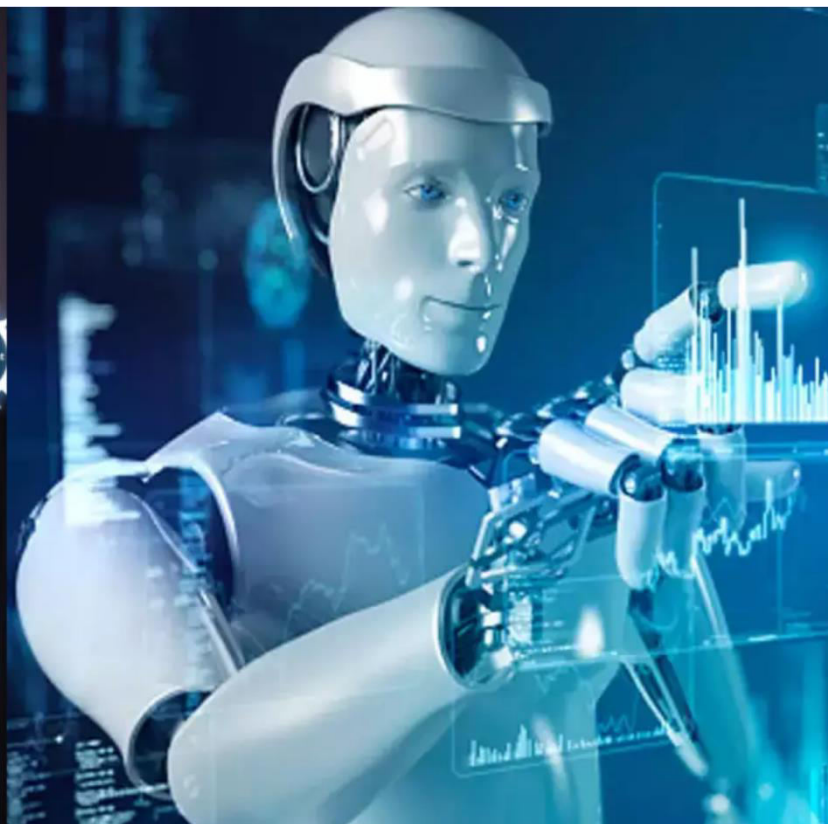




# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 3, March 2025**

# Quixify: An AI-Driven Platform for Optimized Household Services Delivery

Sahu Srihari<sup>1</sup>, Kori Devang<sup>1</sup>, Sajan Joseph<sup>1</sup>, Mistry Daxesh<sup>1</sup>, Ritu Agrawal<sup>2</sup>, Makhduma Saiyed<sup>3</sup>

U.G. Student, Department of Computer Engineering, Parul University, Vadodara, Gujarat, India<sup>1</sup>

Associate Professor, Department of Computer Engineering, Parul University, Vadodara, Gujarat, India<sup>2</sup>

Ph.D, EIT, GSMIEEE, Teaching Fellow, University of Regina, Canada<sup>3</sup>

**ABSTRACT:** A remote service collaboration platform facilitates connections between customers seeking household services and service providers such as cleaners, carpenters, electricians, and plumbers. It allows for seamless communication and collaboration regardless of geographical location, leveraging technology to streamline project management and service delivery. This platform enables users to access a global talent pool, offering diverse expertise tailored to specific needs. ‘Quixify’, for instance, exemplifies this concept by serving as a robust solution linking customers with skilled local service providers. It prioritizes reliability and customer satisfaction, setting new standards in remote collaboration. ‘Quixify’ prioritizes reliability, customer satisfaction, and operational efficiency, making it a pivotal tool in modern remote service collaborations.

**KEYWORDS:** Household services, MERN stack, AI recommendations, real-time tracking, service provider clustering, web application.

## I. INTRODUCTION

The emergence of digital platforms has revolutionized the way household services are delivered and managed [1]. With the advent of advanced technologies such as artificial intelligence (AI), and machine learning (ML), the landscape of household service provision has transformed significantly. These technologies facilitate the seamless integration of various services, enabling real-time data collection, predictive analytics, and personalized service recommendations. Despite these technological advancements, the household service industry faces several challenges. Traditional methods of finding and booking service providers are often time-consuming and inefficient [4].

Users frequently encounter difficulties in identifying reliable service providers, managing appointments, and ensuring service quality [3]. Moreover, the lack of a centralized platform for various household services results in fragmented and inconsistent user experiences [5]. Security and data privacy are also major concerns, as users are hesitant to share personal information and payment details on digital platforms [2].

Recent AI/ML developments enhance service recommendations and user satisfaction, giving these platforms a competitive edge. Secure payment options and data protection are essential, requiring a blend of advanced tech and user-centric design. With technological advancements, platforms now integrate booking systems, payment gateways, and mobile apps for greater convenience.

Existing platforms for household services often suffer from inefficiencies such as cumbersome booking processes, difficulty in finding reliable service providers, and fragmented user experiences. These platforms typically lack integration, resulting in separate systems for booking, payment, and service management, which can be both confusing and time-consuming. Security and data privacy concerns also pose significant issues, as users are wary of sharing personal information and payment details.

Platforms like ‘Quixify’ leverage these technological advancements to connect users with local service providers, enhancing convenience and efficiency in household management.

The main objective of this paper is to introduce Quixify, a comprehensive platform for household services that tackles the challenges outlined above. Quixify aims to:

- A comprehensive platform for household services that addresses current challenges in Industry.
- Development of a user-friendly interface that streamlines the process of browsing, booking, and managing household services, enhancing user experience and accessibility.
- Implementation of a rigorous verification system for service providers, ensuring reliability and quality of services offered through the platform.
- Design of a real-time tracking system for service status updates, coupled with a video consultation feature, improving communication and transparency between users and service providers.
- Establishment of robust security measures to protect user data and ensure secure transactions, addressing privacy concerns in digital service platforms.

The paper is organized as follows: Section II: 'Related Work', highlighting existing solutions and their limitations. Section III: 'Quixify Web Solution', outlining the approach and technologies used to develop and implement the platform. Section IV: 'Result', evaluations of Quixify. Finally, Section V: 'Conclusions', summarizing the key findings and offering suggestions for future research.

## II. LITERATURE SURVEY

Numerous research work has been done based on this field and it is getting more popular to the researchers. They show their interests on finding the ins and outs in this sector and sets the foundation for the development of Quixify.

Smith et al. [1] highlights the transition from traditional methods to digital platforms, emphasizing the increased convenience and efficiency these platforms offered. Early digital platforms provided basic listings of service providers but lacked advanced features such as user reviews, secure payment systems, and real-time service tracking. Over time, these platforms evolved to incorporate more sophisticated features, driven by user demand for better service quality and reliability. However, they still faced limitations in ensuring robust security, seamless real-time communication, and comprehensive service management.

Cababasay et al. [2] presented "HanAPP Buhay: Android Java-Based Mobile Commerce Application for Freelance Home Service Providers and Client Job Seekers," provided an Android app developed using Java, Android Studio, and Firebase for backend support. The system highlighted basic features such as listings of service providers, user reviews, secure payment systems, and real-time service tracking. It lacked advanced AI and ML capabilities to enhance service provider suggestions and better user satisfaction.

Tambe P.S. et al. [3] introduced an online system for home services, enabling convenient booking of household services such as plumbing and electrical work through a web-based platform or mobile application. The system offers features like service booking, secure payments, and a feedback mechanism. However, it faced limitations such as inflexible service booking, inability to choose time slots based on user convenience, limited provider recommendation, and system-scalability issues.

Gupta et al. [4] introduced an innovative Android application aimed at optimizing household service management. Constructed with Java within Android Studio, the application furnishes users with a user-friendly interface facilitating seamless interaction with service providers. Core features encompass service exploration, appointment scheduling, payment facilitation, and feedback provision. However, it faced limitations such as the absence of notification facilities, lack of real-time service tracking, and limited security measures.

Bhattacharjee et al. [5] proposed Service-Hub system addressed issues by integrating features like user and service provider accounts, administrative controls, geographical targeting, and a feedback system to improve communication and service delivery. However, it faced limitations in fully enhancing user satisfaction. The absence of an integrated payment system hindered seamless transactions, and the lack of AI and ML-driven service provider suggestions limited personalized and efficient recommendation.

Adekar et al. [6] mentioned, Household Veritas platform effectively addressed the demand for household services by offering features such as user registration, service provider registration, booking management, in-app messaging, payment integration, and ratings and reviews. Despite these advancements, the platform faced limitations. It lacked advanced AI and ML capabilities to enhance service provider suggestions and better user satisfaction.

Malewade et al. [7] discussed in 'Optimizing web applications using the MERN stack' that MERN stack offered improved performance and scalability through efficient, asynchronous programming and dynamic user interfaces, it also presents challenges such as complexity in managing asynchronous code, performance overheads, and a steep learning curve for developers.

Singh et al. [8] effectively addressed the demand for personalized web interactions through their exploration of data mining and machine learning techniques in their paper "Intelligent Techniques for Web Personalization." They detailed a comprehensive approach that included data collection, pre-processing, pattern discovery, evaluation methodologies, and real-time deployment strategies. These components collectively demonstrate significant strides in the field of web personalization, showcasing the potential for advancing user experience through sophisticated technological applications.

Gupta et al. [9] discussed different definitions of clusters, such as well-separated, centre-based, contiguous, density-based, and similarity-based clusters, each suited to specific applications. Hierarchical clustering, which created a nested sequence of partitions, and partitional clustering, like the K-means algorithm, were explored in detail. The literature also examined incremental and non-incremental methods, as well as overlapping and non-overlapping clustering, highlighting the flexibility required in real-world applications where objects might belong to multiple clusters. These insights emphasized the need for selecting appropriate clustering techniques and measures based on the data's characteristics and the desired analysis outcomes.

Pelivani et al. [10] conducted a comparative analysis of automation testing tools for mobile, web, and desktop applications, focusing on Selenium, Sikuli, Cypress, and Katalon Studio. Selenium was noted for its support for multiple languages and browsers, but it struggled with Windows alerts and parallel execution. Sikuli's image-based automation was found to be prone to failures due to visual changes. Cypress was praised for its network and load time management capabilities but was limited to Chrome and JavaScript. Katalon Studio was highlighted for its user-friendly interface and comprehensive testing capabilities across web, API, mobile, and desktop applications.

Majid et al. [11] proposed a P2P audio and video calling application using WebRTC and A system which provided real-time connection without relying on external plugins, thus reducing data usage and costs. This highlights the growing trend and necessity of integrating efficient, scalable, and real-time communication technologies in modern web applications.

Haque et al. [12] addressed various critical vulnerabilities in web applications, notably SQL Injection, Cross Site Scripting (XSS), and Broken Authentication and Session Management. SQL Injection remained a prevalent threat, allowing attackers to execute arbitrary SQL commands through vulnerable input fields, often mitigated through rigorous input validation and parameterized queries. XSS exploits involved injecting malicious scripts into web pages viewed by other users, necessitating robust sanitization of user inputs to prevent such attacks. Broken Authentication and Session Management flaws led to unauthorized access via session hijacking, emphasizing the need for secure session handling practices such as regenerating session IDs and properly managing cookies. The implementation of these security measures was demonstrated to significantly bolster the defences of web applications, mitigating the risks posed by these common vulnerabilities and enhancing overall web security.

Quixify has revolutionized household service management by addressing common issues of earlier platforms and applications. It offers customizable bookings, flexible scheduling, thorough notifications, live tracking, strong security and ensures compatibility across devices. Additionally, Quixify introduces cutting-edge features like AI and ML-driven service provider recommendations, ensuring users are matched with the best-suited providers, enhancing the overall user experience and setting a new standard in household service management.

### III. METHODOLOGY

Quixify is composed of several key modules, each responsible for specific aspects of the platform's functionality. These modules include:

- User Management
- Service Provider
- Service Booking
- AI/ML Recommendation
- Real-Time Communication
- Security & Compliance.

Together, these modules form a comprehensive system that ensures seamless user experiences, efficient service provider management, secure transactions.

#### A. User Management Module

The User Management module is central to Quixify, managing user registration, authentication, and profile management. It ensures users can easily sign up, log in, and manage their profiles securely. This module employs Firebase Authentication to support various sign-in methods and MongoDB to store user profiles. Using Node.js, it handles server-side scripting, ensuring efficient communication between the client and server. The goal is to provide a seamless user experience while maintaining robust security protocols to protect user data from unauthorized access. The figure 1, illustrates the overall functionality of the User Management module:

- **Firebase Authentication** [13]: Provides secure and reliable user authentication, supporting various sign-in methods such as email/password, social media accounts, and phone numbers.
- **MongoDB** [14]: A NoSQL database used to store user profiles, offering flexibility in data schema and scalability to accommodate a growing user base.
- **Node.js** [15]: Facilitates server-side scripting and handles user requests, ensuring efficient communication between the client and server.

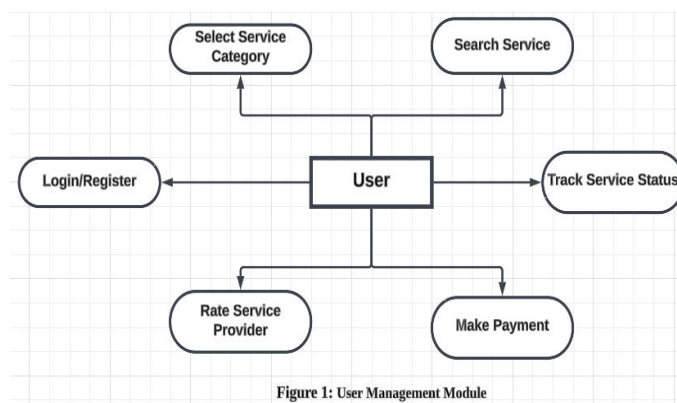


Figure 1: User Management Module

Fig. 1. illustrates the overall functionality of the User Management module

#### B. Service Provider Module

The Service Provider module manages the registration, verification, and profile management of service providers. This module ensures that only verified and qualified service providers are available to users. It uses Express.js to build the backend API, MongoDB to store detailed profiles, and React.js to provide a dynamic and responsive frontend interface. The module includes features for service providers to manage their availability, respond to service requests, and update their profiles. Cloudinary is integrated for efficient image storage, allowing service providers to upload and manage profile pictures and service-related images. Users can rate and review service

providers, which helps maintain quality and trust within the platform. The figure 2, illustrates the overall functionality of the Service Provider module:

- **Express.js** [16]: A web application framework for Node.js, used to build the backend API that handles service provider operations.
- **MongoDB** [14]: Stores detailed profiles of service providers, including their skills, ratings, and availability.
- **React.js** [17]: Provides a dynamic and responsive frontend interface for service providers to interact with the platform.
- **Cloudinary** [18]: A cloud Storage platform, help to store image and use in dynamic and responsive frontend interface.

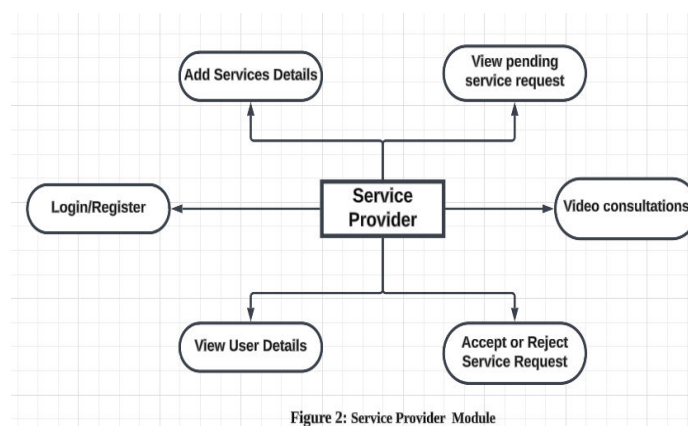


Figure 2: Service Provider Module

Fig. 2. Illustrates the overall functionality of the Service provider module

### C. Service Booking Module

The Service Booking module streamlines the process for users to browse, schedule, and manage service appointments. It integrates real-time availability checks, booking confirmation, and secure payment processing using Razorpay API. This module employs React.js for intuitive user interfaces, Node.js, and Express.js for backend operations. Users can search for services, select time slots, and make secure payments. Real-time notifications keep users updated on their booking status, enhancing the overall user experience and trust in the platform.

- **React.js** [17]: Develops intuitive user interfaces, allowing users to browse and book services effortlessly.
- **Node.js** [15] and **Express.js** [16]: Handle backend operations, including service availability checks, booking confirmations, and transaction processing.
- **Razorpay API** [19]: Ensures secure payment transactions, adhering to industry standards and enhancing user trust.

### D. AI and Recommendation Module

The AI and Recommendation module enhances user experience by leveraging machine learning algorithms to suggest suitable service providers based on user preferences and historical data. It uses Python for developing machine learning models, Flask for deploying predictions, and MongoDB for storing relevant data. The k-means clustering algorithm groups service providers based on similarity metrics. This module personalizes service recommendations, improving the matching process and ensuring users receive the most relevant suggestions for their needs.

- **Python** [23]: Used for developing machine learning models that analyse user data and service provider profiles.
- **MongoDB** [14]: Stores relevant data used for training and inference, ensuring seamless integration with other Quixify modules.

- **K-means Clustering Algorithm:** Groups and recommends service providers based on similarity metrics derived from user interactions and historical service data.

#### E. Real-Time Communication Module

The Real-Time Communication module enables video consultations between users and service providers, facilitating clear communication and quick issue resolution. It uses WebRTC for video conferencing, and React.js for the frontend interface. This module ensures users and service providers can communicate directly through the platform, enhancing engagement and satisfaction. Real-time notifications are handled by Twilio, keeping users informed of new messages, booking updates, and service changes.

- **WebRTC** [20]: Utilized for video conferencing functionalities, allowing seamless virtual consultations.
- **React.js** [17]: Integrated into the frontend to provide a user-friendly interface that supports real-time communication features, enhancing overall user engagement and satisfaction.
- **Twilio** [21]: It sends real-time notifications, keeping users informed about bookings, service status, and provider communications.

#### F. Security and Compliance Module

The Security module is dedicated to protecting user data and ensuring secure transactions by focusing on encryption and data storage practices. Passwords are encrypted using bcrypt, which ensures that user credentials are stored securely and are protected from unauthorized access. All application data, including user profiles and service details, is stored in MongoDB, benefiting from its flexible schema and robust security features to safeguard sensitive information. This module plays a crucial role in maintaining data integrity and privacy, thereby building trust and ensuring compliance with security standards.

- **Bcrypt** [22]: Used for hashing and encrypting passwords to protect user credentials.
- **MongoDB** [14]: Stores all application data securely, offering flexibility and robust security measures for sensitive information.

By integrating these modules with advanced technologies Quixify aims to revolutionize the household service industry, offering a seamless and secure platform that enhances user convenience and service provider efficiency.

- **Flow chart:**

In figure 3: Flow chart, provides a visual representation of the overall workflow within Quixify. It outlines the processes from user registration to service booking, payment, and feedback collection. This diagram highlights the interactions between clients and service providers, showcasing how Quixify ensures a seamless and efficient experience for all users and the integration of various modules to deliver optimized household services.

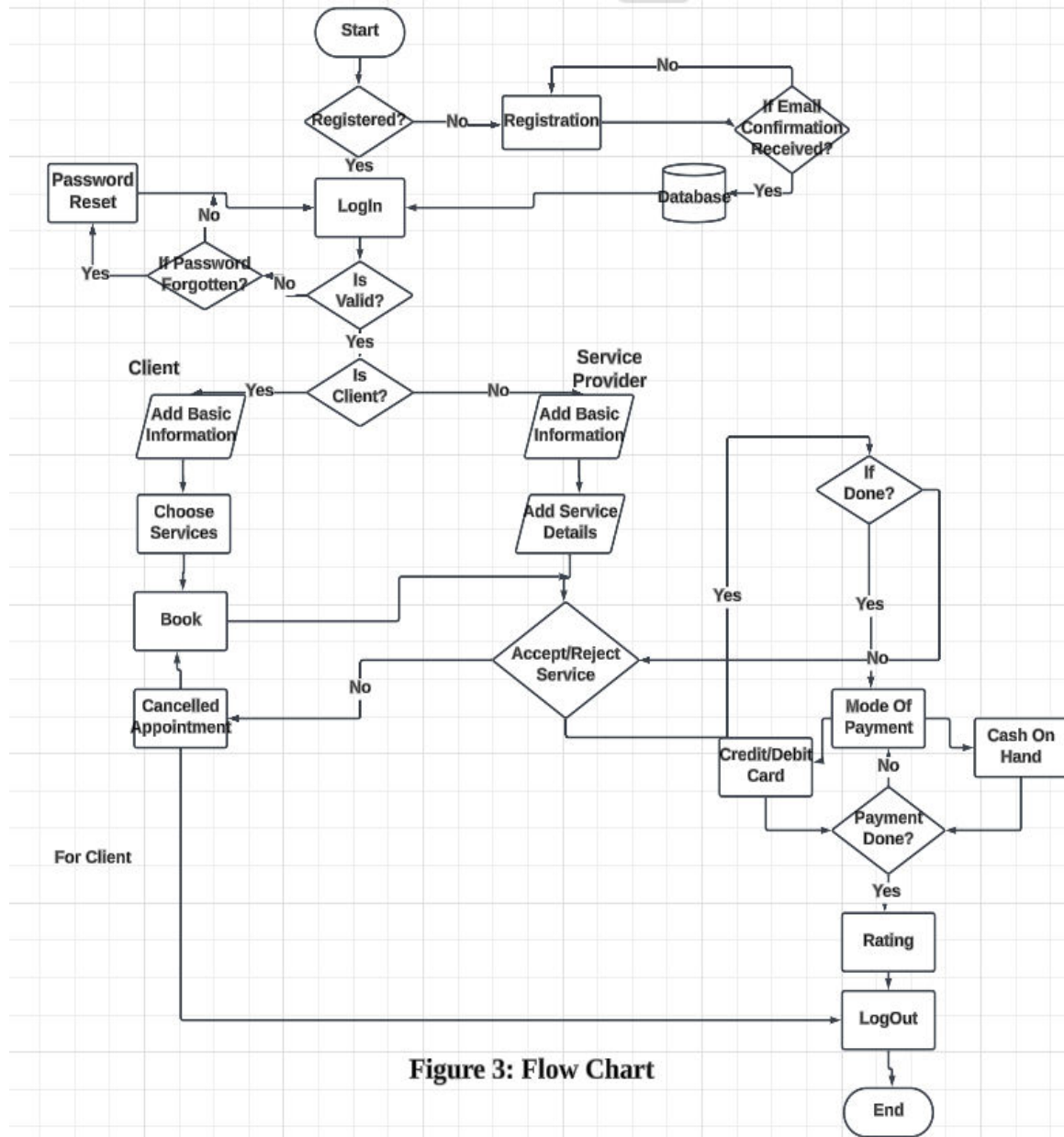


Figure 3: Flow Chart

Fig. 3. Illustrates the flow of Quixify

#### IV. EXPERIMENTAL RESULTS

Quixify has proven effective in enhancing household service delivery through advanced technology and modular design. The evaluation of Quixify's performance was conducted across several key metrics, including user satisfaction, service provider engagement, system reliability, and transaction security, AI and recommendation system, limitations and future scope. Below, we outline the results obtained from these assessments:

##### A. User Satisfaction and Engagement

Quixify's user-friendly interface and efficient management system have made a significant difference in user satisfaction. With Firebase Authentication, users can log in securely through various methods, and MongoDB's

flexible setup ensures the platform grows with the user base. The real-time notifications from Twilio keep users updated on service progress and bookings, enhancing overall engagement.

#### B. Service Provider Efficiency

The Service Provider module made it easier to manage profiles, availability, and service listings. Real-time communication via WebRTC improved interactions between clients and providers. Plus, the AI-driven recommendation system helped service providers by suggesting the best matches based on past data and user preferences.

#### C. System Reliability and Scalability

As Quixify continues to grow, its modular architecture, powered by technologies like Node.js, MongoDB, and React.js, has shown great potential for further scalability. This adaptability ensures that the platform can accommodate future expansions, such as the addition of new services or regions, without requiring major architectural changes.

#### D. Transaction Security

Transaction security is a key part of Quixify's success. By using Razorpay for payments, we ensure that all transactions are protected with top-notch encryption and fraud prevention. We also use bcrypt to keep passwords safe, reducing the risk of unauthorized access. Regular security audits confirm that Quixify meets the highest industry standards, making it a secure choice for both users and service providers.

#### E. AI and Recommendation System Performance

The AI-driven recommendation system uses k-means clustering to analyse user preferences and service provider profiles. The graph shows distinct clusters, visualized using Principal Component Analysis (PCA), shows distinct clusters where each colour represents a group of similar service providers. This clustering approach, based on historical booking data and user feedback, enables the platform to recommend service providers more effectively, matching users with relevant options. The system achieved an accuracy rate of 82.62%, consistently improving its suggestions, leading to higher booking completion rates and enhanced user satisfaction.

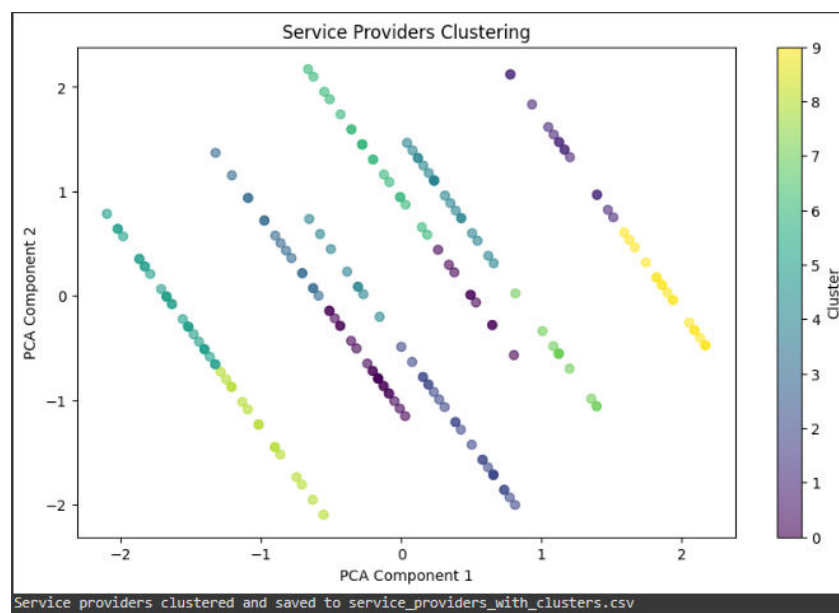


Fig. 4. Clustering of Service Providers using K-Means Algorithm

#### F. Limitations and Future Scope

While Quixify has demonstrated robust performance, there are a few areas for potential growth. Some edge cases in AI recommendations could be refined for better user satisfaction, especially in scenarios where user preferences are

unique or highly specialized. Additionally, while video consultations improve service clarity, there is an opportunity to further automate this feature using AI for initial problem diagnosis.

| Controller/Functionality                              | Tested Features                                                                                        | Key Observations                                                                                                                                                                                  |
|-------------------------------------------------------|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>User Signup &amp; Login Controller</b>             | User registration, login, email verification, password reset                                           | High success rate with all scenarios well-handled.                                                                                                                                                |
| <b>Service Provider Signup &amp; Login Controller</b> | Provider registration, login, profile verification                                                     | Strong validation for service provider authentication and verification.                                                                                                                           |
| <b>Booking Controller</b>                             | Booking creation, service availability checks, cancellations                                           | Booking process is smooth with real-time availability checks. Handled race conditions well.                                                                                                       |
| <b>Service Provider Fetch &amp; Recommendation</b>    | AI-based recommendations, filtering service providers                                                  | Accurate suggestions based on user preferences and historical data.                                                                                                                               |
| <b>Flexible Booking Controller</b>                    | Dynamic scheduling, rescheduling, and time-slot availability                                           | Seamless selection and modification of service timings.                                                                                                                                           |
| <b>Payment Controller</b>                             | Secure payments via Razorpay API.                                                                      | Payment processes were secure and efficient. Tested for multiple transaction types, including failures and retries.                                                                               |
| <b>Video Consultancy (Real-Time Communication)</b>    | WebRTC video consultations, Twilio notifications for updates                                           | Real-time video consultations worked smoothly. Notifications were triggered as expected.                                                                                                          |
| <b>Service Acceptance/Reject Controller</b>           | Service provider acceptance/rejection of service requests                                              | Providers can easily accept or reject requests.                                                                                                                                                   |
| <b>User Dashboard Controller</b>                      | Display user profile information, view/manage bookings, access service history/ratings, update profile | User profile info is accurate. Booking management is smooth. Profile updates are real-time.                                                                                                       |
| <b>Service Provider Dashboard Controller</b>          | Display provider profile info, view/manage bookings, access performance metrics, update details        | Provider profile and booking management are operational. Performance metrics are correct.                                                                                                         |
| <b>Admin Dashboard Controller</b>                     | View system metrics/statistics, manage users/providers, access/review reports, admin settings          | Admin dashboard provides clear system overview. Management functionalities are operational with appropriate controls. Added functionality for expanding business with new services and locations. |
| <b>Rating and Feedback Controller</b>                 | Submit ratings/feedback, display ratings, handle disputes/adjustments                                  | Submission process is smooth. Feedback display and aggregation are accurate.                                                                                                                      |

Table 1. Unit test results across various controllers and functionalities.

| Feature                            | Quixify                                                       | UrbanClap                                    | Handy                                        | TaskRabbit                                   |
|------------------------------------|---------------------------------------------------------------|----------------------------------------------|----------------------------------------------|----------------------------------------------|
| <b>Core Functionality</b>          | Booking services across categories with recommended providers | Booking services across categories           | Booking cleaning and handyman services       | Booking local task helpers                   |
| <b>Service Recommendation</b>      | AI-driven suggestions for local service providers             | User selects from service listings           | User selects from service listings           | User selects from service listings           |
| <b>Video Consultancy</b>           | Available before service acceptance                           | Not available                                | Not available                                | Not available                                |
| <b>Support for Local Providers</b> | Emphasizes local employment and engagement                    | Primarily uses large vendor networks         | Primarily uses large vendor networks         | Mix of local and large vendor networks       |
| <b>Problem Identification</b>      | Easy understanding through video and local communication      | No pre-service problem identification method | No pre-service problem identification method | No pre-service problem identification method |
| <b>Wide Range of Services</b>      | Very wide range (household services)                          | Very wide range (household services)         | Limited range (cleaning, handyman)           | Broad range (moving, cleaning, errands)      |
| <b>Scheduling Flexibility</b>      | Highly flexible scheduling with local providers               | Flexible scheduling with providers           | Limited to booking availability              | Flexible scheduling with providers           |
| <b>Pricing Transparency</b>        | Transparent, competitive pricing with local providers         | Transparent pricing, varies by service       | Fixed pricing model                          | Variable pricing based on tasks              |

Table 2. Highlighting the advantages over other similar platforms.

## V. DISCUSSION

Quixify represents a significant advancement in the household services sector by leveraging technology, community-driven solutions, and a modular architecture. The platform's focus on enhancing user experience through secure authentication, flexible scheduling, and personalized recommendations has set it apart from conventional service platforms like UrbanClap, Handy, and TaskRabbit. Quixify's integration of AI for service provider recommendations based on user preferences and past data offers a competitive edge, ensuring users are matched with the most suitable providers, thus increasing the likelihood of booking completions and user satisfaction.

The inclusion of video consultations before service acceptance has proven invaluable in providing clarity regarding service needs and ensuring that both users and providers are on the same page. This not only improves the accuracy of service delivery but also reduces misunderstandings, leading to fewer cancellations and higher user retention rates. The real-time communication facilitated by WebRTC has also streamlined interactions between clients and providers, enabling seamless collaboration and service execution.

Quixify's emphasis on local service providers has had a twofold impact: firstly, it provides employment opportunities to local workers, strengthening the local economy, and secondly, it fosters trust and improved communication between users and providers, given the lack of language and cultural barriers. This community-based approach is a unique selling point that distinguishes Quixify from larger, more generic platforms.

However, there are areas where Quixify can grow further. While the AI-driven recommendation system performs well in general scenarios, its accuracy diminishes when dealing with highly specialized or uncommon user preferences. This can be attributed to the system's reliance on historical data, which may not always account for niche cases. Improving the AI's handling of such cases could enhance user satisfaction even more. Additionally, further automation of the video consultation process using AI for preliminary problem identification could streamline the user experience.

In comparison to other platforms, Quixify's modular design, scalable architecture, and focus on transaction security, powered by technologies like Razorpay and bcript, provide a solid foundation for continued growth. However, challenges like further optimizing AI and expanding the platform to new regions or service types will require ongoing development and refinement.

## **VI. CONCLUSION**

Quixify has emerged as an innovative platform in the household services sector, combining technological advancements with a community-focused approach. By prioritizing local service providers and incorporating AI-driven recommendations, real-time communication, and secure transactions, the platform has successfully differentiated itself from competitors like UrbanClap, Handy, and TaskRabbit. Its modular architecture, scalability, and focus on security further strengthen its position in the market.

The integration of video consultations offers a unique feature that ensures transparency between users and service providers, while the flexible scheduling and wide range of services cater to diverse user needs. However, as with any growing platform, there are areas for improvement, particularly in optimizing the AI recommendation system for specialized cases and automating the consultation process.

Overall, Quixify has demonstrated significant potential to revolutionize the household services landscape. With continued focus on user feedback, AI advancements, and service expansion, Quixify is well-positioned for sustained growth and long-term success, benefiting both users and local service providers.

## **REFERENCES**

1. Smith J. (2022). Evolution of Online Platforms for Household Services. *Journal of Service Research*, 10(2), 123-136.
2. Cababasay K.C., Notarte M.J.C., Tan G.P., Catain J.A., Valdez A.P., Salvador G.T., Lao A.F.B., 2022. HanAPP Buhay: Android Java-Based Mobile Commerce Application for Freelance Home Service Providers and Client Job Seekers. In: 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), December 23-25, Karnataka, India.
3. Tambe P.S., Nikam P., Gunjal T., Jadhav P., Parakhe S., 2020. An Online System for Home Services. In: *International Journal of Scientific Development and Research (IJS DR)*, Vol. 5, Issue 9, September 2020, SVIT, Chincholi, Nashik, Maharashtra, India.
4. Gupta K., Tewari A., Basra R., 2021. Android App for Household Services. In: *International Journal of Innovative Research in Technology*, Volume 8 Issue 1, June 2021, Galgotias University, Greater Noida, India.
5. Bhattacharjee A., Tasnim T., Sohel M.S.R., Chakraborty A., Maruf A.A., Dutta S., Ishmam A.M., Toha T.R., Islam A.B.M.A.A., 2017. Service-Hub: A Better Approach for Developing the System of Online Marketing for Daily Services. In: 2017 IEEE 8th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), October 12-14, Vancouver, BC, Canada.
6. Adekar A., Dalvi A., Gharat P., Ratanghayra P., 2023. Household Veritas - A platform that provides household services. In: *International Research Journal of Engineering and Technology (IRJET)*, Volume 10 Issue 3, March 2023, Atharva College of Engineering, Malad West, Mumbai, India.
7. Malewade S.M., Ekbote A., 2021. Performance Optimization using MERN stack on Web Application. In: *International Journal of Engineering Research Technology (IJERT)*, Vol. 10 Issue 06, June-2021, Vasai, Palghar, India.
8. Sarabjot S. A. Bamshad M., Intelligent Techniques for Web Personalization, 2003 Center for Web Intelligence, School of Computer Science, Telecommunications and Information Systems, DePaul University, Chicago, Illinois, USA.

9. Er. Arpit G., Er. Ankit G., Er. Amit M., Research Paper On Cluster Techniques Of Data Variations,2011 Faculty Of Engineering and Technology (IT department), Mahatma Gandhi Chitrakoot Gramodaya Vishwavidyalaya. Chitrakoot-485780, Satna, Madhya Pradesh, India.
10. Pelivani, E., Cico, B., 2021. A Comparative Study of Automation Testing Tools for Web Applications. In: Proceedings of the 10th Mediterranean Conference on Embedded Computing (MECO), June 7-10, 2021, Budva, Montenegro.
11. Majid H.A., Samah A.A., Yusuf L.M., Nasien D., Cheah T.L. (2016). P2P audio and video calling application using WebRTC. ARPN Journal of Engineering and Applied Sciences, 11(3). FEBRUARY 2016, Skudai, Johor Bahru, Malaysia.
12. M. Fazlul H., M. Badrul A. M., Fuyad A. M., Enhancement of Web Security Against External Attack,2017 Department of ssInformation and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh-1902, Tangail, Bangladesh.
13. Firebase: <https://firebase.google.com/>
14. MongoDB: <https://www.mongodb.com/>
15. Node.js: <https://nodejs.org/en>
16. Express.js: <https://nodejs.org/en>
17. React.js: <https://legacy.reactjs.org/>
18. Cloudinary: <https://cloudinary.com/>
19. Razorpay API: <https://razorpay.com/>
20. WebRTC: <https://webrtc.org/>
21. Twilio: <https://www.twilio.com/>
22. Bcrypt: <https://www.npmjs.com/package/bcrypt>
23. Python: <https://www.python.org/>



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details