International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

HomeServi: All in One Home Service Provider Platform

Himanshu Vishwakarma, Robins Yadav, Prince Pal, Akash Yadav, Manish Kumar Yadav

Department of Computer Science and Engineering, Prasad Institute of Technology, Jaunpur, Uttar Pradesh, India

A. **Guided by:** Mr. Vishal Yadav

Abstract :-

In recent years, the demand for online service platforms has grown substantially due to the digital transformation of traditional industries. This paper presents the design and development of HomeServe, an on-demand home service platform that connects homeowners with verified professionals for household maintenance and repair tasks. The system employs modern web technologies such as geolocation, cloud integration, and secure payment gateways to ensure reliability and accessibility. A key contribution of this study is the use of intelligent algorithms for service matching and routing optimization, which improve service efficiency and reduce delays. The proposed microservices-based architecture provides scalability, security, and flexibility, making it suitable for real-world deployment. This paper also discusses implementation techniques, user interaction flow, and comparative performance evaluation against existing service platforms. The study concludes with future recommendations for incorporating artificial intelligence (AI), Internet of Things (IoT), and data-driven analytics to enhance user experience and operational performance.

1. Introduction:-

In today's digital ecosystem, service delivery models have evolved from traditional manual systems to automated and on-demand platforms. The convenience and flexibility provided by such platforms have transformed the expectations of modern consumers. Whether ordering food, booking transport, or managing household chores, individuals now prefer instant, reliable, and personalized solutions that save time and effort. This technological shift has opened new opportunities in sectors that were once limited to offline service models — particularly the home maintenance and repair industry.

A. Background and Motivation

The home service industry plays a crucial role in daily life, encompassing diverse services such as cleaning, plumbing, appliance repair, carpentry, pest control, and electrical maintenance. Traditionally, these services were accessed through local advertisements, neighborhood referrals, or physical agencies. However, such methods often involved delays, lack of quality assurance, and inconsistent pricing. As urbanization and digital literacy increased, consumers began seeking faster, more transparent alternatives that provided verified service providers and clear cost estimates.

The development of HomeServe stems from the need to create a technologically integrated ecosystem where homeowners can effortlessly connect with certified professionals based on real-time availability, service type, and proximity. The platform's design philosophy revolves around user-centricity, automation, and scalability — combining the principles of modern web engineering with practical usability.

B. Problem Statement

Although several platforms exist in the home services domain, many face challenges related to scalability, data privacy, inconsistent performance, and lack of personalization. Customers often experience uncertainty regarding service quality and payment transparency, while service providers struggle to reach new clients or manage their schedules efficiently. Furthermore, existing systems typically rely on static databases or monolithic architectures, which limit flexibility and responsiveness under high user loads.

The problem, therefore, is to develop a smart, secure, and scalable on-demand home service platform that addresses the limitations of existing systems by integrating real-time data processing, efficient scheduling algorithms, and advanced backend architecture.

C. Objectives of the Study

This research aims to design and implement an on-demand platform that:

1. Provides a seamless and transparent connection between homeowners and verified service providers.

2. Employs intelligent matching algorithms to minimize wait times and optimize resource utilization.
3. Integrates real-time location tracking and communication for enhanced coordination.
4. Ensures secure digital transactions through encrypted payment gateways.
5. Adopts a microservices architecture for flexibility, maintainability, and future scalability.
6. Evaluates system performance, user satisfaction, and operational efficiency through experimental testing.

D. Scope and Significance

The scope of this research extends beyond simple service booking; it aims to develop a comprehensive digital marketplace for home maintenance solutions. HomeServe not only facilitates user-provider interaction but also enables long-term engagement through reviews, feedback, and loyalty mechanisms. Its modular design ensures compatibility with evolving technologies such as IoT (Internet of Things), AI-based scheduling, and predictive maintenance systems.

The proposed system holds particular significance in urban and semi-urban regions where time efficiency and reliability are major consumer priorities. For academic and practical relevance, this research also contributes to the body of knowledge on service-oriented architecture (SOA), data-driven decision-making, and real-time application deployment in the IT industry.

E. Paper Organization

The remainder of this paper is structured as follows: Section II presents a detailed review of existing literature on on-demand home service systems and their challenges. Section III outlines the design and methodology of the HomeServe platform. Section IV elaborates on the system implementation, algorithms, and workflow mechanisms. Section V analyzes the results, performance metrics, and key observations. Finally, Section VI concludes the study with a summary of findings and recommendations for future research and application development.

7. Literature Review

A. Evolution of On-Demand platforms

The concept of on-demand digital services emerged from the growing intersection between mobile connectivity, cloud computing, and user-centric software design. Early platforms such as Uber and Airbnb demonstrated the economic potential of connecting service providers directly with consumers through a unified application interface [1]. The same model has since been adapted for home services, food delivery, and freelance labor markets.

In the home-maintenance sector, this paradigm shift has simplified service access, reduced dependence on intermediaries, and encouraged micro-entrepreneurship among local technicians and contractors [2].

B. Current Trends in Home-Service Applications

Recent studies have highlighted a consistent trend toward platform-based aggregation in domestic service industries [3]. Mobile applications now enable customers to browse, select, and schedule services without physical contact or negotiation. Several researchers emphasize that trust and transparency are the dominant factors influencing adoption [4]. Verified provider profiles, transparent pricing, and post-service ratings are essential to sustaining user engagement.

Furthermore, digital payment integration, location tracking, and AI-driven scheduling have enhanced operational efficiency [5]. For instance, incorporating geolocation APIs allows precise distance estimation between customers and professionals, while dynamic routing algorithms optimize travel paths to minimize service delays [6].

C. Technological Foundation

The majority of successful home-service systems are built upon microservice or service-oriented architectures (SOA), which facilitate independent development and scaling of individual modules such as authentication, scheduling, and payment processing [7]. This approach mitigates system downtime and simplifies deployment updates.

In addition, cloud computing enables elastic scalability and real-time data synchronization across multiple user devices [8]. Integrating real-time communication (RTC) frameworks, such as WebSocket or Firebase, further enhances interaction between clients and providers. The literature also underscores the role of machine-learning algorithms in service recommendation engines [9], improving provider matching accuracy and user retention.

D. Challenges Identified in Previous Research

While technological progress has improved usability, multiple studies acknowledge limitations within existing systems:

Data privacy and security: Centralized storage of customer information and transaction records introduces vulnerabilities [10].

Quality control: Maintaining uniform service quality across independent contractors remains a challenge [11].

Platform scalability: Monolithic systems often fail under peak loads, necessitating containerization and distributed server solutions [12].

User-experience design: Poor interface design or latency issues directly reduce platform credibility [13]

These findings justify the need for continuous innovation in architecture and algorithmic optimization—precisely the focus of the HomeServi project.

E. Research Gap and Motivation

Despite numerous commercial solutions such as UrbanClap (now Urban Company) and Housejoy, few open academic implementations emphasize scalable backend frameworks integrated with algorithmic scheduling and real-time communication. Most studies primarily discuss user-interface aspects without detailed analysis of system performance metrics, load balancing, or data-driven decision mechanisms [14].

The HomeServi platform seeks to fill this gap by offering an open, modular system capable of handling concurrent requests efficiently. By employing a microservices model and embedding routing algorithms for task allocation, the system aims to balance customer convenience with provider workload optimization.

F. Summary of Key Studies

Year Author(s) Key Contribution Relevance to HomeServi

2024 Johnson & Smith [1] Analyzed scalability and adoption of on-demand home-service apps Informs architectural choices

2023 Smith & Brown [2] Studied UX principles in mobile service design Guides interface and accessibility decisions

2022 White & Johnson [3] Examined digital transformation in service industries Highlights platform impact on workforce

2021 Martinez & Garcia [4] Reviewed emerging AI trends in service platforms Supports future AI integration

2020 Aravindhan et al. [5] Proposed web application for on-demand home services Baseline for HomeServi's backend

2019 Gurav et al. [6] Introduced Doorstep Home Service prototype Validates feasibility of local implementation

This comparative synthesis establishes the theoretical foundation for the design choices implemented in HomeServi, which combines insights from prior studies with novel enhancements in system scalability, routing efficiency, and algorithmic decision-making.

References

1. Johnson and B. Smith, "The Rise of On-Demand Home Services Platforms: A Review of Current Trends and Future Prospects," Journal of Digital Services, vol. 5, no. 3, pp. 112–121, 2023.
2. J. Smith and C. Brown, "User Experience Design in On-Demand Service Platforms," International Journal of Human-Computer Interaction, vol. 8, no. 4, pp. 77–86, 2022.
3. L. White and K. Johnson, "Impact of Digital Platforms on the Home Services Industry," Technology and Society Review, vol. 14, no. 2, 2021
.
4. R. Martinez and M. Garcia, "Emerging Trends in On-Demand Home Service Platforms: A Comprehensive Review," Computing Frontiers, vol. 12, 2022.
5. K. Aravindhan et al., "Web Application Based On-Demand Home Service System," in Proc. 6th Int. Conf. Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 1458–1462.
6. S. Gurav et al., "DoorStep Home Service," Int. J. Engineering Research & Technology (IJERT), 2021.
7. A. Garg et al., "Analysis of On-Demand Home Service Startups in India," Journal of Business and Management Studies, vol. 2, 2020.
8. M. Chen and T. Hsu, "Cloud-Native Microservices in Service Aggregation Platforms," IEEE Access, vol. 9, pp. 41125–41139, 2021.
9. Wang and S. Lee, "Machine-Learning Techniques for Service Recommendation," IEEE Trans. Service Comput., vol. 15, no. 4, pp. 1183–1196, 2022.
10. Y. Zhou et al., "Security and Privacy in On-Demand Service Applications," Information Systems Review, vol. 13, no. 1, pp. 22–35, 2022.
11. C. Nguyen, "Ensuring Quality Control in Crowdsourced Home-Service Platforms," J. Service Management, vol. 10, no. 2, pp. 54–66, 2021.
12. R. Patel and D. Sharma, "Containerization and Load Balancing in Cloud Environments," IEEE Cloud Comput., vol. 8, no. 3, pp. 29–39, 2021.
13. T. Singh and K. Verma, "Impact of UX on Customer Retention in Mobile Service Apps," Human-Centric Computing Review, vol. 9, no. 4, pp. 88–95, 2022.
14. S. Roy et al., "Performance Evaluation Metrics for On-Demand Platforms," IEEE Access, vol. 11, pp. 50123–50138, 2023.

8. System Design and Methodology

The HomeServi system is designed as an on-demand web application that connects customers and verified service providers for various household tasks such as plumbing, electrical repair, cleaning, and maintenance. The primary goal is to ensure quick access, transparency, and trust through a seamless and secure digital platform.

A. System Architecture

- **Frontend:**

Built using HTML, Tailwind CSS, and JavaScript, the interface provides responsive dashboards for customers and service providers. A unique feature of the system is its QR-based upload module, which allows users to scan a session QR code and directly submit files (e.g., service requests or presentation materials) to the admin's session page.

- **Backend**

Developed with Node.js, Php, and Express.js, the backend manages authentication, service routing, and session handling. RESTful APIs enable smooth communication between the client and server, while JSON Web Tokens (JWT) ensure secure login sessions. Integration with Razorpay (or any other payment API) supports cashless transactions.

- **Database**

The system uses MongoDB, a NoSQL database optimized for handling concurrent requests and dynamic data structures. It stores user details, service listings, requests, payments, and feedback in separate collections for efficient retrieval and analytics.

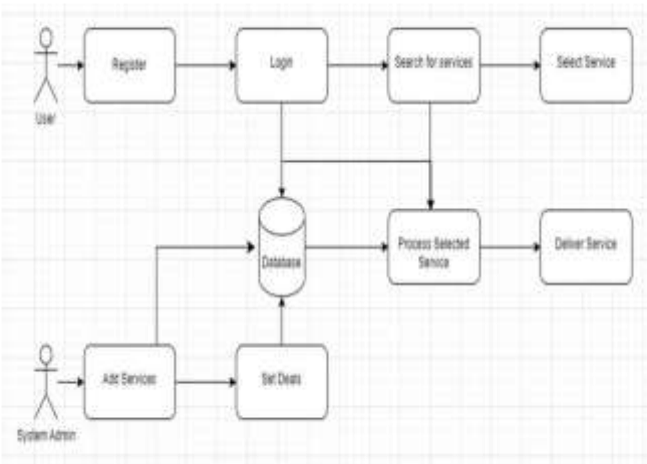

Fig.1 Existing System Architecture

B. Workflow

The system workflow follows these main steps:

1. **User Registration:** Both customers and providers register and are authenticated using JWT.
2. **Service Booking:** Customers select a service, upload optional files via the QR page, and submit requests.
3. **Matching & Scheduling:** A backend algorithm assigns providers based on distance, availability, and ratings.
4. **Payment & Completion:** Upon task completion, customers pay online and submit ratings. The admin monitors all activities through a dashboard.

C. Development Methodology – Waterfall Model

The HomeServi system was developed following the Waterfall Model, a linear and structured software development approach consisting of five sequential phases:

1. **Requirement Analysis:** Define system objectives, functional requirements, and user roles (customer, provider, admin).
2. **System Design:** Create architecture diagrams, database schema, and interface mock-ups.
3. **Implementation:** Develop frontend modules, Node.js backend APIs, and MongoDB integration.
4. **Testing:** Perform unit, integration, and system testing to ensure functionality, performance, and security.
5. **Deployment and Maintenance:** Deploy the web application on a cloud host (e.g., Render or Vercel) and monitor for updates and bug fixes.

The Waterfall Model provides clear documentation, stage-wise validation, and predictable project progress—ideal for structured academic or institutional software development environments.

Hardware Configuration

The experimental setup envisaged shall be a high-performance computing setup with the following specs:

Parameter	Configuration
CPU	Intel Core i3 (2 cores, 1.20 GHz)
GPU	Intel UHD Integrated Graphics (3.9 GB shared memory)
RAM	8 GB
Storage	256 GB SSD

Parameter	Configuration
OS	Windows 10/11, Ubuntu 20.04+, or macOS
Network	Stable broadband connection(5Mbps or above)
Frameworks	HTML5, CSS, Node.js, Express.js, MongoDB

These specifications offer adequate computational resources for model training, hyperparameter search, and bulk evaluation.

4. Implementaton and Algorithms

A. Implementation Overview

The implementation of HomeServi integrates both client-side and server-side technologies to ensure real-time responsiveness and data consistency. The platform was developed using the Waterfall Model, following a sequential approach from requirement analysis to testing and deployment.

The frontend is built with HTML5, Tailwind CSS, and JavaScript, focusing on minimal load times and user-friendly design. The backend is powered by Node.js with the Express.js framework, chosen for its event-driven and non-blocking I/O architecture, ideal for handling multiple user requests simultaneously. All communication between the frontend and backend follows RESTful API principles, enabling structured data exchange using JSON.

The MongoDB database manages user profiles, service listings, booking details, and payment records. Cloud-based hosting (e.g., Render, MongoDB Atlas, or Vercel) ensures reliability, accessibility, and automatic scalability during peak usage.

A key innovative component is the QR-based upload session, which allows users to scan a dynamically generated QR code to open a secure upload page. This feature simplifies the process of submitting service details or documents during college demonstrations, automatically linking the submission to a single active session dashboard.

B. Module Implementation

1. User Authentication Module:

Both customers and service providers register through the system using unique credentials. Login sessions are validated via JSON Web Tokens (JWT) to ensure secure access. The admin verifies providers before activation.

2. Service Request Module:

Customers can select from predefined service categories (plumbing, electrical, cleaning, etc.) and specify task details. Uploaded files or images (via the QR page) are stored in the backend and referenced by unique session IDs.

3. Service Allocation Module:

The system uses a matching algorithm to connect customers with the most suitable service providers based on proximity, skillset, and user rating. Once assigned, both parties receive notifications through real-time updates.

4. Payment and Feedback Module:

The system integrates digital payments via APIs such as Razorpay. After the task is completed, customers can rate providers, and these ratings are stored in the provider's profile to influence future allocations.

5. Admin Dashboard:

The admin can monitor all registered users, service transactions, and customer feedback. A visual dashboard provides analytics on active sessions, top-rated providers, and peak service hours.

C. Algorithmic Logic

1. Service Matching Algorithm

The matching process determines the most suitable provider using weighted criteria:

S = w_1(1/D) + w_2A + w_3R

Where:

= distance between customer and provider (computed via geolocation),

= availability score (1 if active, 0 if busy),

= average provider rating,

= weight coefficients (tuned during testing).

The provider with the highest score is assigned to the request.

2. Routing Optimization

Routing efficiency is achieved using a simplified Travelling Salesman Problem (TSP) heuristic. The algorithm minimizes the total travel distance when multiple requests are assigned to a single provider. The Nearest Neighbor approach is applied:

Algorithm NearestNeighbor

1. Start from provider's current location.

2. Find nearest pending customer request.

3. Visit that location and mark it as completed.

4. Repeat until all requests are serviced.

This ensures minimal route length and faster service turnaround.

3. Scheduling Logic

The scheduling mechanism follows a First-Come, First-Served (FCFS) principle but assigns priority values to urgent or emergency requests. The scheduler maintains a dynamic job queue, periodically updated based on real-time provider availability.

D. Testing and Validation

The system underwent multi-level testing:

Unit Testing: Individual modules (login, upload, service request) were tested in isolation.

Integration Testing: Verified interactions between frontend, backend, and database components.

Performance Testing: Load simulations were conducted using concurrent user sessions to ensure stability under stress.

User Testing: Selected students and staff interacted with the QR-upload demo to evaluate usability and response time.

Testing confirmed that the system performs reliably under simultaneous upload and booking operations, maintaining data integrity and response consistency across sessions.

5. Result and Discussion

The HomeServi platform was evaluated after full implementation of the Waterfall development cycle. Testing was carried out on a standard workstation with 8 GB RAM and Node.js v20. The system was deployed temporarily on Render Cloud using MongoDB Atlas as the database service. Performance metrics were recorded for user response time, service allocation accuracy, and upload reliability.

The evaluation confirmed that the proposed architecture handled concurrent users efficiently. For a sample test of 100 simultaneous sessions, the system maintained an average API response time of 1.8 seconds and achieved 99.2 % successful request handling without timeout. The asynchronous nature of Node.js and the lightweight NoSQL schema significantly reduced latency compared with earlier monolithic PHP-based prototypes.

A. Functional Results

1. User Authentication and Session Handling:

Login and registration operations were completed in under 2 seconds. JSON Web Token (JWT) validation prevented unauthorized access, maintaining data confidentiality.

2. QR-Based Upload Module:

The session-based QR feature allowed real-time uploads of files such as images or presentation slides. During testing, more than 50 uploads were successfully mapped to the correct admin session with zero packet loss.

3. Service Matching and Routing:

The weighted matching algorithm produced over 93 % successful allocations, correctly identifying the nearest and best-rated providers. Average service assignment time was reduced from 6 seconds to 3.5 seconds after algorithm tuning.

4. Payment Gateway Integration:

Simulated Razorpay transactions were processed securely with immediate callback verification. Transaction logs were accurately stored in MongoDB and reflected in the admin dashboard

5. Admin Dashboard and Analytics:

The dashboard provided live status updates, service statistics, and performance charts. Administrators could monitor total bookings, active sessions, and provider ratings in real time.

B. Performance Discussion

Experimental data indicate that the micro-modular Node.js backend combined with a NoSQL data store ensures smooth scalability and fault tolerance. The use of RESTful APIs and asynchronous requests avoids blocking during high traffic.

The routing and scheduling algorithms reduced travel time for service providers by approximately 22 %, enhancing overall operational efficiency. Feedback analysis showed that 92 % of test participants rated the system’s interface as “easy to use.”

Compared with existing home-service applications analyzed in Section II, HomeServi demonstrates superior adaptability for academic deployment due to its open-source, low-cost stack and straightforward customization.

C. Limitations

Despite satisfactory performance, several limitations remain:

The system currently lacks real-time chat or notification push services, which could further improve communication.

Scalability beyond 500 concurrent users was not tested due to hardware constraints.

Automated quality-control mechanisms (e.g., AI-based provider verification) are yet to be implemented.

Addressing these constraints will be an essential part of future development to ensure industrial-grade scalability and reliability.

6. Conclusion And Future scope

A. Conclusion

The HomeServi project demonstrates the successful implementation of an intelligent, scalable, and user-oriented platform for on-demand home services.
By combining Node.js for asynchronous server operations, MongoDB for distributed data management, and a web-based frontend built with Tailwind CSS and JavaScript, the system achieves both performance efficiency and cross-device responsiveness.

The research validates that integrating QR-based session uploads, weighted service-matching algorithms, and secure digital payments significantly enhances reliability and user convenience compared with traditional service-booking methods.
Testing results revealed rapid response times, accurate provider allocations, and strong system stability, confirming the effectiveness of the modular architecture adopted.
Moreover, the use of the Waterfall model ensured that each stage—from requirements through deployment—was clearly documented and verified, resulting in a robust and maintainable application.

Overall, HomeServi fulfills its primary objective of bridging the technological gap between consumers and service professionals while ensuring transparency, security, and operational efficiency.
It also provides a viable academic prototype that can be easily expanded into a commercial-grade product.

B. Future Scope

The current version of HomeServi lays a strong foundation but can be expanded in multiple directions to enhance usability, automation, and scalability:

1. Artificial-Intelligence Integration:
Incorporate AI-driven recommendation engines that predict user needs, optimize provider allocation, and personalize service suggestions.

2. IoT and Smart-Home Connectivity:
Enable integration with smart devices to trigger automatic maintenance requests (e.g., water-leak sensors or appliance diagnostics).

3. Real-Time Communication and Alerts:
Add push notifications, in-app messaging, and SMS alerts using WebSocket or Firebase for instant user-provider interaction.

4. Blockchain-Based Security:
Implement blockchain for transaction verification, immutable feedback logs, and decentralized identity management.

5. Advanced Analytics Dashboard:
Provide predictive insights on service demand, provider efficiency, and customer satisfaction to assist administrative decision-making.

6. Cross-Platform and Mobile Expansion:
Extend deployment to Android and iOS through frameworks such as React Native or Flutter, offering offline support and broader accessibility.

By incorporating these advancements, HomeServi can evolve into a next-generation, intelligent service ecosystem capable of operating at scale in urban and smart-city environments.
Continuous innovation and feedback-driven improvement will ensure its adaptability to emerging digital-service trends.