

AI-Powered Marketplace For Home-Based Skilled Services

AVS Radhika¹, Koppa Bramani², Kajal Kumari³, Thotapally Kalyani⁴

¹Assistant Professor; Department Of Computer Science And Engineeringbhoj Reddy Engineering College For Women Hyderabad, India.

^{2,3,4}B.Tech Students; Department Of Computer Science And Engineeringbhoj Reddy Engineering College For Women Hyderabad, India.

Mail Id; koppabramanis@gmail.com², kalyanithotapally755@gmail.com⁴, kajalkumarisinha2004@gmail.com³

Abstract

In today's fast-growing digital economy, connecting local skilled workers with potential customers continues to be a major challenge. Many home-based professionals, including tailors, artisans, and other service providers, often face difficulties in gaining visibility and accessing appropriate job opportunities. The proposed AI-Powered Marketplace for Home-Based Skilled Services presents a web-based platform designed to effectively bridge this gap by facilitating seamless interaction between service providers and customers. Existing service platforms frequently lack advanced location-based filtering, personalized recommendations, and efficient communication features, which can lead to poor user experience and limited service discovery. To overcome these shortcomings, the proposed system integrates artificial intelligence-driven recommendation techniques along with location-aware filtering. Algorithms such as K-Nearest Neighbors (KNN) are employed to suggest the most relevant nearby service providers, thereby improving accuracy and convenience for users. The platform enables skilled workers to create detailed profiles, display their portfolios, manage bookings, and communicate directly with customers. At the same time, users can search for services, schedule appointments, make secure payments, and submit feedback. Furthermore, sentiment analysis is applied to customer reviews to evaluate service quality and build trust within the platform. By combining intelligent recommendation mechanisms with an intuitive user interface, the proposed solution improves accessibility, encourages self-employment, and enhances the overall service experience. Ultimately, the system supports local economic development by empowering individuals and strengthening community-based service networks.

Keywords— Artificial Intelligence, KNN, Recommendation System, Sentiment Analysis, Location-Based Services, Web Marketplace, Skilled Workers, Personalized Services.

Introduction

In many local communities, home-based skilled workers such as tailors, craft makers, and small service providers face difficulties in reaching

potential customers and promoting their services effectively. Limited access to digital platforms and inadequate visibility restrict their opportunities for growth. Most existing solutions do not provide efficient local search capabilities, skill presentation options, or simplified booking mechanisms, making it challenging for both workers and customers to connect. To address these issues, the proposed AI-powered marketplace introduces an online platform designed to connect skilled home-based professionals with nearby customers. The system allows service providers to create detailed profiles, showcase their work, and receive bookings through a secure interface. Customers can easily search for services, view recommendations, and complete transactions safely. The platform also enhances communication between customers and service providers, ensuring smooth coordination and improved service delivery. By increasing visibility for talented individuals and supporting self-employment, the system contributes to strengthening local economies. Overall, the proposed solution promotes convenience, reliability, and mutual benefits for both workers and customers.

Existing System

Current online platforms primarily focus on general freelance or remote job opportunities and provide limited support for home-based skilled workers such as tailors, artisans, and local service providers. These systems often lack dedicated features such as location-based discovery, personalized recommendations, and integrated booking or payment facilities tailored to local services. As a result, skilled workers face challenges in showcasing their expertise, while customers struggle to identify reliable service providers within their vicinity.

Proposed System

The proposed system consists of three primary modules: Admin, Talent Provider, and Customer, supported by artificial intelligence features to improve usability and efficiency. The Admin module is responsible for managing users, monitoring transactions, and overseeing platform activities. Talent Providers can create profiles, upload work samples, manage bookings, update service status, and interact with customers. Customers can search for service providers based on

skills and location, book services, make secure payments, and provide feedback. AI-based recommendation techniques are incorporated to match customers with suitable nearby talents, enhancing personalization, visibility, and overall system performance.

Requirement Analysis

Functional Requirements

Functional requirements define the core operations and services that the system must perform. These requirements specify the expected behavior of the application, including user interactions, inputs, outputs, and system responses. The proposed platform is structured into three main modules: Admin, Customer, and Talent Provider.

Non-Functional Requirements

Non-functional requirements define the quality attributes of the system and describe how the system performs under different conditions. These requirements ensure efficiency, reliability, and usability of the platform.

- **Scalability:** The system should support an increasing number of users and services without performance degradation.
- **Usability:** The application should provide an intuitive and user-friendly interface for all types of users.
- **Reliability:** The system must ensure high availability and stable performance during peak usage.
- **Security:** User data, transactions, and communications must be protected through secure authentication and authorization mechanisms.
- **Compatibility:** The platform should function effectively across modern web browsers and devices.

Computational Resource Requirements

Hardware Requirements

The minimum hardware specifications required for efficient system operation include:

- Processor: Intel Core i5 or higher
- RAM: 8 GB or above
- Storage: 256 GB Hard Disk

Software Requirements

The software environment required for development and deployment includes:

- Operating System: Windows 10 or Windows 11
- IDE: Visual Studio Code

Design

Architecture

System architecture defines the structure of the application, including its components and the flow of information between them. It explains how requests are processed, which modules participate in execution, and the sequence of operations. A well-designed architecture supports scalability, security, and maintainability. The proposed system incorporates both software and technical architectural perspectives.

Software Architecture

Software architecture outlines the logical organization of the application and the interaction between modules. A secure architecture is essential to minimize vulnerabilities during development and deployment. Proper architectural planning helps identify potential risks, analyze dependencies, and ensure robust system behavior.

The proposed system follows a layered architecture consisting of:

- Presentation Layer (User Interface)
 - Application Logic Layer
- This separation improves modularity and simplifies maintenance. Security considerations are incorporated to reduce risks such as unauthorized access, data leakage, and improper input handling. The architecture also supports efficient debugging and performance optimization.

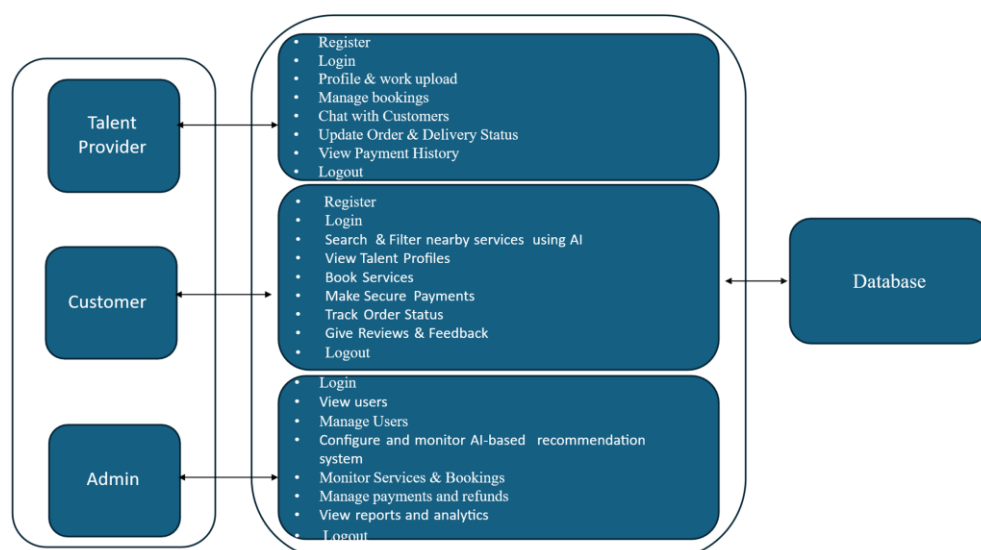


Fig. 1 illustrates the software architecture of the proposed system.

Technical Architecture

Technical architecture defines the physical deployment of system components and their interactions. It describes the technologies, servers, databases, and communication mechanisms involved in system execution. The proposed

architecture uses a web-based client-server model where users interact through a browser interface. Requests are processed by backend services, which communicate with the database to retrieve or store information. This structure ensures scalability and efficient resource utilization



Fig 2 shows the technical architecture of the system.

Testing

Software testing is the process of evaluating a software application to verify whether it satisfies the specified requirements and to identify defects before deployment. The primary objective of testing is to ensure that the developed system performs correctly, reliably, and securely. A well-tested application improves software quality and minimizes the risk of failures after release.

With the rapid growth of digital services such as online banking, e-commerce, and service booking platforms, users increasingly depend on software systems in their daily activities. Even a minor defect can lead to significant financial loss, reduced performance, and damage to user trust. Therefore, testing plays a crucial role in delivering a dependable product. The major benefits of software testing include:

- Cost effectiveness
- Customer satisfaction

Initially, debugging was the primary testing method. Over time, testing evolved into a structured quality assurance activity integrated into the software development life cycle. Modern testing approaches combine manual and automated testing to improve accuracy and reduce delivery time. After individual modules are verified, the integrated system is tested to ensure overall functionality.

Dimensions of Testing

Software testing involves multiple dimensions to ensure complete system validation:

- Application Layers: Database, API, and user interface testing
- Testing Scale: Unit testing, module testing, integration testing, and system testing

- Testing Types: Functional, performance, and security testing
 - Testing Methodology: Exploratory testing, manual scripted testing, and automated testing
- Characteristics of Software Testing Life Cycle (STLC)
- STLC is a subset of the Software Development Life Cycle (SDLC) focusing only on testing activities.
 - Testing begins once requirements are finalized.
 - STLC provides a structured approach for ensuring software quality.

During early stages, the testing team defines the scope, prepares test cases, and establishes entry and exit criteria. Early preparation reduces testing time and helps detect issues at an initial stage. Once development is completed, test execution begins, enabling early identification of defects.

Stages of Testing

The Software Testing Life Cycle consists of six major phases:

1. Requirement Analysis

In this phase, testing requirements are identified and analyzed. The testing team collaborates with stakeholders to understand functional and non-functional requirements. These requirements guide the preparation of test strategies.

2. Test Planning

Test planning defines testing objectives, tools, resources, schedules, and responsibilities. A well-prepared plan reduces risks and ensures efficient testing execution.

3. Test Case Development

Test cases are designed based on system requirements. Each test case includes input data, expected results, and execution steps. Test case review ensures completeness and accuracy.

4. Test Environment Setup

A testing environment similar to the production environment is prepared. This includes configuring hardware, software, database, and network settings. Proper setup ensures realistic testing conditions.

5. Test Execution

During execution, test cases are run on the system. Actual results are compared with expected outcomes. Any deviation is logged as a defect.

6. Test Closure

Test closure involves analyzing results, documenting defects, evaluating coverage, and preparing test reports. This phase confirms completion of testing activities.

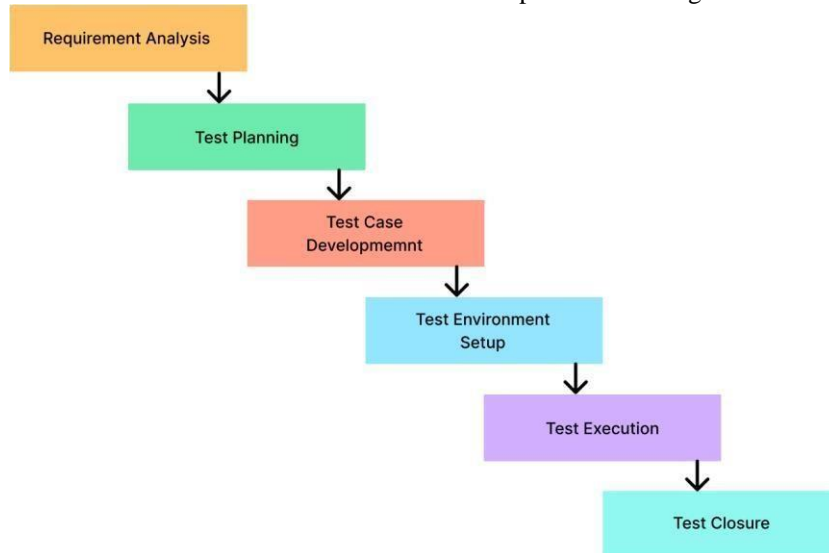


Fig. 3 illustrates the phases of testing.

Types of Testing

Black Box Testing

Black box testing evaluates system functionality without examining internal code. The tester focuses on input and output behavior. This method is applicable at unit, integration, system, and acceptance testing levels.

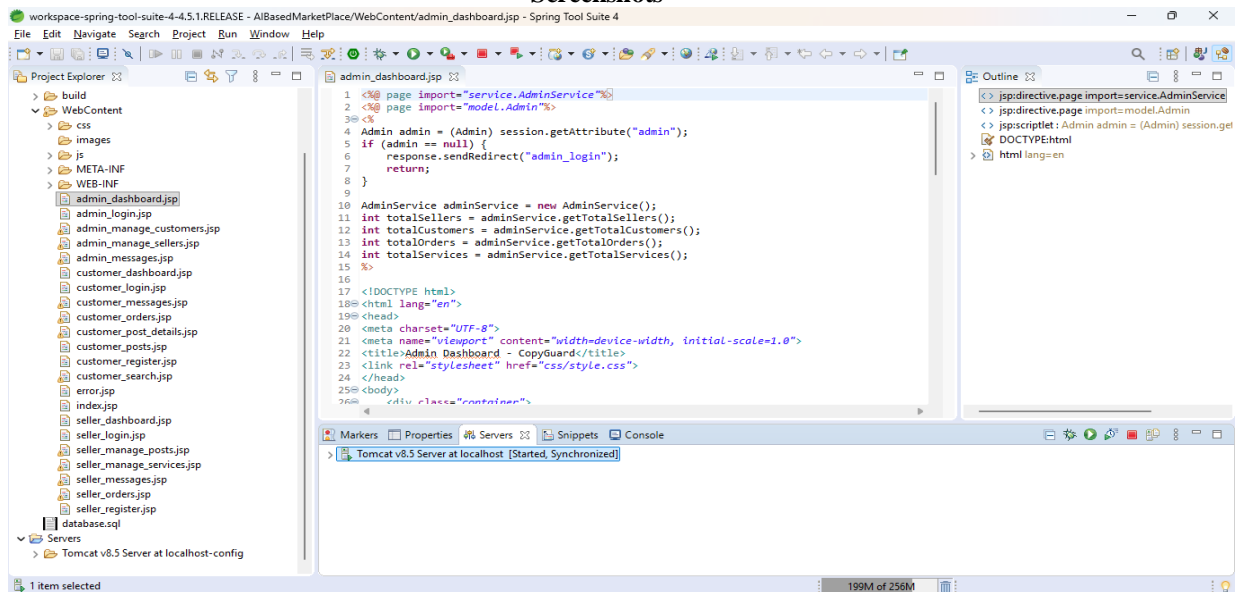
White Box Testing

White box testing examines the internal logic and structure of the code. Test cases are designed based on control flow and code paths. This testing is typically performed at the unit level.

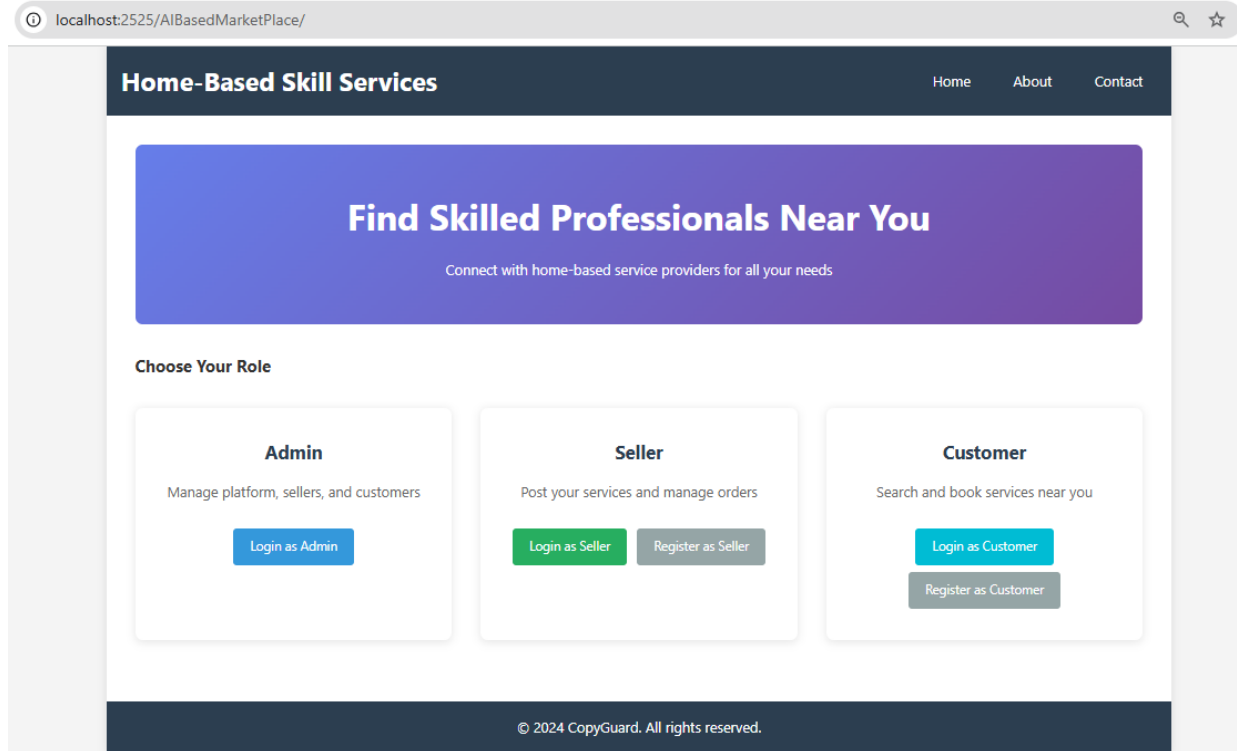
White box testing techniques include:

- Statement coverage
- Branch coverage

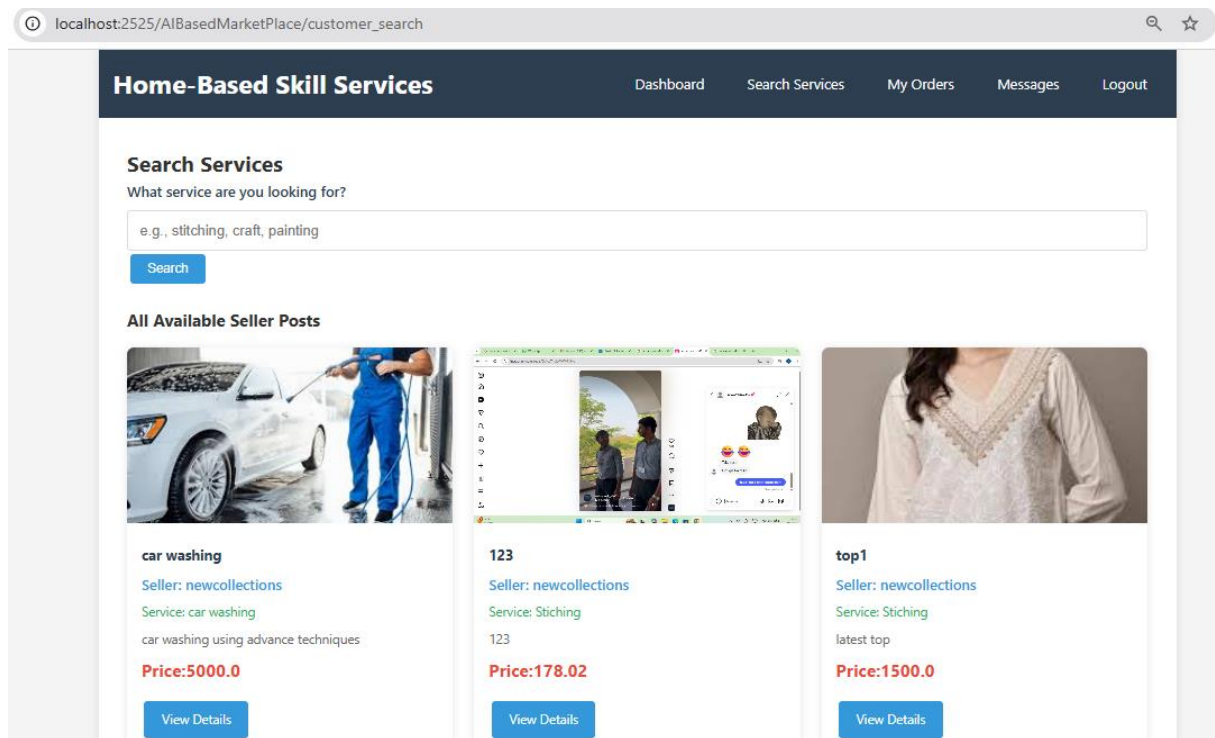
Screenshots



Screenshot 1 Run the web application on Apache Tomcat server



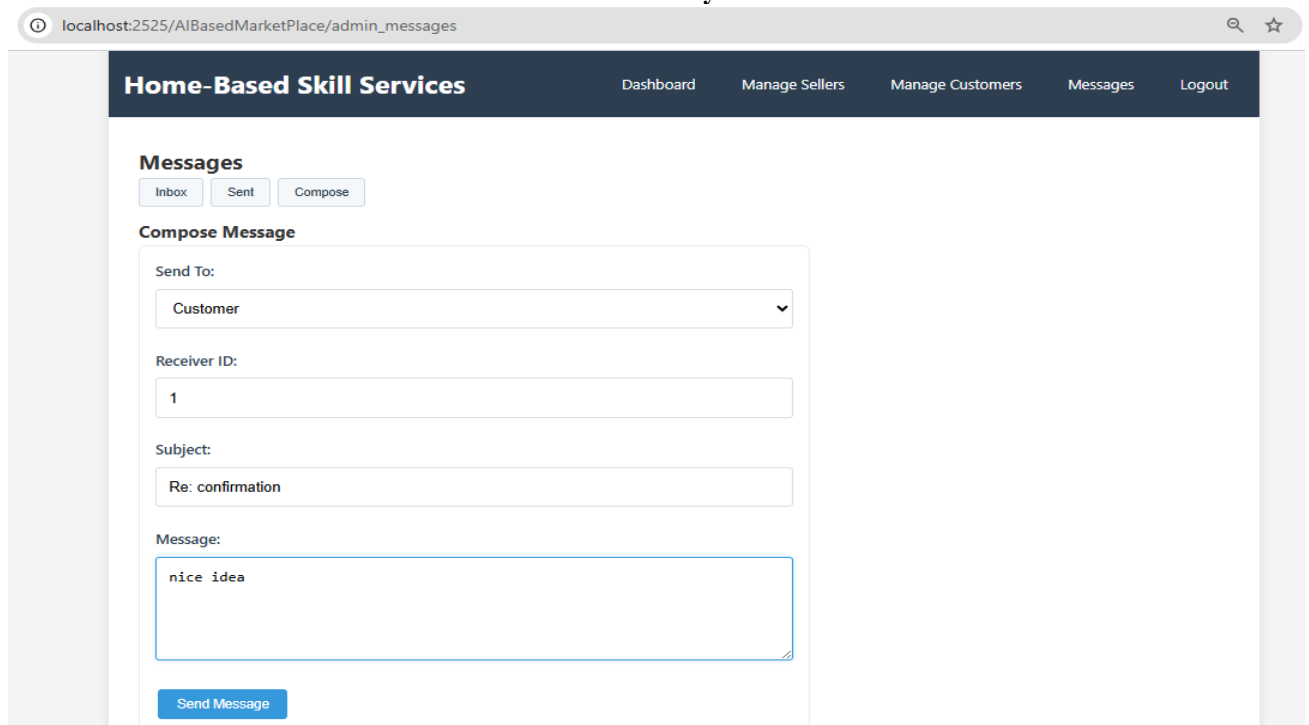
Screenshot 2 Web Page



Screenshot 3 Search Services by Customer



Screenshot 4 Book service by Customer



Screenshot 5 Compose Messages by Admin

Conclusion

The proposed system effectively connects home-based skilled professionals with potential customers through a structured online marketplace. By providing a dedicated platform, service providers are able to present their skills, increase visibility, and access improved employment opportunities. Simultaneously, customers benefit from the ability to easily discover reliable local services tailored to their requirements.

The inclusion of features such as personalized profiles, intelligent recommendation mechanisms,

and streamlined communication enhances overall user experience and fosters trust between users. The feedback and rating system further promotes transparency, allowing customers to make informed decisions while encouraging service providers to maintain quality standards. In addition, the platform supports self-employment, strengthens local service ecosystems, and contributes to community-level economic development. Overall, the system offers a scalable and efficient solution for empowering skilled individuals and improving access to local services.

Future Scope

The proposed platform can be extended with advanced artificial intelligence techniques to improve recommendation accuracy and deliver a more personalized experience. Future enhancements may include the development of a dedicated mobile application to improve accessibility and convenience. Integration of real-time chat functionality between customers and service providers can further enhance communication.

Additional improvements may involve secure online payment gateways, expansion to support multiple service categories, and coverage across broader geographic regions. Features such as live service tracking and instant booking confirmation can also be incorporated to improve usability. Furthermore, multilingual support can be introduced to make the platform accessible to users from diverse linguistic backgrounds. These enhancements will increase system adoption and improve overall functionality.

References

- [1] C. S. Rajpoot, V. Tiwari, and S. K. Vishwakarma, "Emerging Trends of Recommender System for E-Commerce: A Comprehensive Review," *Discover Computing*, vol. 29, no. 63, pp. 1–20, Jan. 2026.
- [2] S. Y. Lim, N. Hashim, and L. T. Thanh, "Recommender Systems: A Comprehensive Review of Models, Approaches and Evaluation Metrics," *Journal of Informatics and Web Engineering*, vol. 4, no. 3, pp. 1–15, Oct. 2025.
- [3] L. Kang and Y. Wang, "Personalized Product Recommendation System for E-Commerce Platforms," *ITM Web of Conferences*, vol. 74, pp. 1–11, Feb. 2025.
- [4] Y. H. Alfaifi, "Recommender Systems Applications: Data Sources, Features, and Challenges," *Information (MDPI)*, vol. 15, no. 10, pp. 660, Oct. 2024.
- [5] Y. C. Hua, P. Denny, J. Wicker, et al., "A Systematic Review of Aspect-Based Sentiment Analysis: Domains, Methods, and Trends," *Artificial Intelligence Review*, vol. 57, pp. 1–30, Sept. 2024.
- [6] Z. Tang, "Research on Web Text Sentiment Analysis and Application Based on Deep Learning," *Advances in Engineering Innovation*, vol. 16, no. 5, pp. 85–89, May 2025.