WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Worklyn

***G. Hema Prabha* [*1], *J. Shiva Surya* [*2]**

[*1] Department of Computer Science and Engineering, Bachelor of Engineering, Sri Shakthi Institute of Engineering and Technology Coimbatore-641062, Tamil Nadu, India.

[*2] Department of Computer Science and Engineering, Bachelor of Engineering, Sri Shakthi Institute of Engineering and Technology Coimbatore-641062, Tamil Nadu, India.

A B S T R A C T:

WorkLyn is a smart local service marketplace platform developed to connect customers with nearby service professionals such as electricians, plumbers, carpenters, mechanics, cleaners, and other skilled workers. The platform aims to simplify the process of finding reliable professionals through intelligent search, location-based filtering, and real-time communication. It provides three different user roles: customers, workers, and administrators. Customers can search for professionals, submit booking requests, communicate through real-time chat, share locations, and provide reviews. Workers can create professional profiles, manage bookings, update availability, and showcase their previous work. Administrators can manage users, verify workers, monitor complaints, and maintain platform settings. The system is developed using React.js, Node.js, Express.js, MongoDB, and Socket.IO. It also incorporates a Progressive Web Application (PWA) architecture, allowing users to install the platform on mobile and desktop devices without developing a separate native application. WorkLyn AI provides an integrated and accessible digital solution for improving local service discovery and communication.

Keywords: WorkLyn, Local Service Marketplace, React.js, Node.js, MongoDB, Artificial Intelligence, Progressive Web Application, Socket.IO.

1. Introduction

Finding reliable local service professionals is a common challenge for customers. People often depend on personal recommendations, phone calls, or local advertisements to identify suitable workers for household and maintenance services. These traditional methods can be time-consuming and may not provide sufficient information about worker availability, experience, skills, or customer feedback.

WorkLyn AI is developed to address these challenges through a centralized digital platform that connects customers with nearby service professionals. The system provides an organized environment where customers can discover workers, submit service requests, communicate directly, and manage their bookings.

The platform supports different service categories, including electricians, plumbers, carpenters, mechanics, cleaners, and technicians. It uses location-based search and intelligent matching techniques to help customers identify suitable professionals.

The application is developed using the MERN technology stack and supports real-time communication through Socket.IO. It is also designed as a Progressive Web Application, providing an application-like experience across supported devices.

1.1 Objectives

The main objectives of WorkLyn are:

1. To develop a centralized platform for connecting customers and local workers.
2. To simplify worker discovery through intelligent search and location-based filtering.
3. To provide secure registration and authentication for different user roles.
4. To enable real-time communication between customers and workers.
5. To provide booking, review, and complaint management facilities.
6. To develop an installable PWA for mobile and desktop users.
7. To provide administrators with tools to manage and monitor the platform.

1.2 Existing System

In the existing system, customers generally find local service professionals through personal contacts, advertisements, and offline recommendations. These approaches have several limitations, including limited worker information, difficulty comparing professionals, lack of centralized booking management, and delays in communication.

1.3 Proposed System

The proposed WorkLyn platform provides a centralized solution for local service discovery and management. Customers can search professionals based on service requirements, location, ratings, and other criteria. Workers can manage their profiles and booking requests, while administrators can monitor the overall platform.

The system integrates intelligent worker matching, location-based search, real-time chat, secure authentication, and PWA installation to improve accessibility and usability.

2. System Architecture and Technologies

WorkLyn follows a full-stack client-server architecture. The frontend communicates with the backend through RESTful APIs, while MongoDB stores application data. Socket.IO supports real-time communication between connected users.

2.1 Frontend Technologies

The frontend is developed using React.js 18 and Vite. React Router handles application navigation and protected routes, while the Context API manages global application states. Tailwind CSS is used for responsive interface design, and Framer Motion provides animations.

2.2 Backend Technologies

The backend is developed using Node.js and Express.js. It handles authentication, user management, worker search, bookings, reviews, complaints, and other application operations through RESTful APIs.

2.3 Database Management

MongoDB is used as the primary database, with Mongoose providing schema definitions, validations, and database operations. The system maintains collections for customers, workers, administrators, bookings, reviews, complaints, categories, and notifications.

2.4 Real-Time Communication

Socket.IO is integrated to provide instant communication between customers and workers. It supports messaging, typing indicators, read receipts, and location sharing.

2.5 Progressive Web Application

WorkLyn is configured as an installable Progressive Web Application using a web app manifest and service worker. Users can install the platform on supported mobile and desktop browsers and access it through a standalone application window.

3. System Modules

3.1 Customer Module

The customer module allows users to register, log in, search for local workers, view professional profiles, submit booking requests, communicate with workers, save favorites, provide reviews, and submit complaints.

3.2 Worker Module

The worker module enables service professionals to create profiles, specify their professional categories, provide experience details, upload portfolio images, manage availability, and accept or reject booking requests.

3.3 Admin Module

The administrator module provides centralized control over the platform. Administrators can manage customers and workers, verify professional accounts, manage service categories, monitor complaints, and configure platform settings.

3.4 Booking Management Module

The booking module manages the complete service request process. Customers can submit booking requests, and workers can accept, reject, or update booking statuses. Customers can monitor their bookings through the dashboard.

3.5 Real-Time Chat Module

The chat module enables direct communication between customers and workers. It supports instant messages, typing indicators, read receipts, and location sharing.

3.6 Review and Complaint Module

Customers can submit reviews and ratings based on their service experiences. They can also report issues through the complaint management system, allowing administrators to review and resolve reported problems.

4. Intelligent Worker Matching and Location-Based Search

4.1 Natural Language Search

WorkLyn AI provides an intelligent search mechanism that processes customer service requirements expressed in natural language.

For example, a customer may enter: "I need an emergency plumber near me under ₹1000."

The search system identifies relevant information such as service category, budget, urgency, and location to support worker discovery.

4.2 Worker Matching Algorithm

The platform uses a weighted scoring approach to rank suitable workers according to multiple factors.

Matching factor	Weight
Skill/Profession Match	40
Customer Rating	25
Verification Status	10
Availability	15
Experience	10
Total	100

The matching score helps identify workers who best satisfy the customer's requirements.

4.3 Location-Based Search Using Haversine Formula

The Haversine formula is used to calculate the geographical distance between customer and worker coordinates.

The formula is:

$$ a=\sin^2(\Delta\phi/2)+ \cos(\phi_1)\cos(\phi_2)\sin^2(\Delta\lambda/2) $$

$$ d=2R\arcsin(\sqrt{a}) $$

Where:

- d = Distance between two locations.
- R = Earth's radius.

- \(\phi_1,\phi_2\) = Latitude coordinates.
- \(\lambda_1,\lambda_2\) = Longitude coordinates.

This calculation supports nearby worker identification without requiring a paid distance calculation API.

5. Security and Authentication

Security is an important part of the WorkLyn AI platform. JWT (JSON Web Token) is used for authentication and protected API access. BcryptJS is used to hash user passwords before storing them in the database.

Role-based authorization restricts access to customer, worker, and administrator functionalities. Helmet provides HTTP security headers, while Express Rate Limit helps control excessive API requests.

The application also uses middleware for authentication, role verification, request validation, and error handling.

6. Development and Deployment

The application is developed using a separate frontend and backend structure.

The React frontend is configured using Vite and runs on port 5173 during development. The Express backend runs on port 5000 and communicates with MongoDB through Mongoose.

The frontend is deployed using Vercel, while the backend is hosted on Render. MongoDB Atlas is used for cloud database storage.

The system uses environment variables to configure database connections, authentication secrets, email services, file storage, and API endpoints.

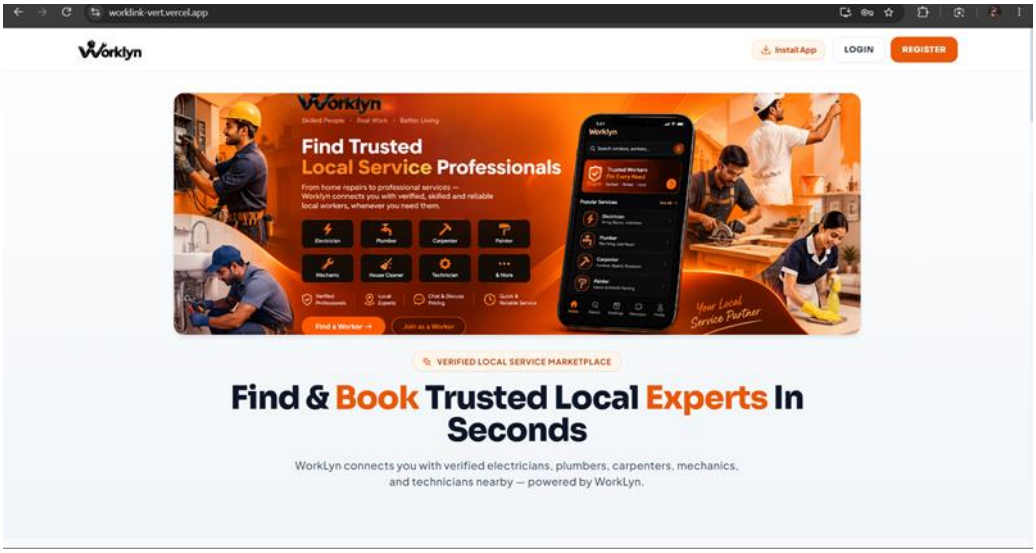
7. Results and Discussion

WorkLyn AI integrates customer management, worker management, booking operations, intelligent search, and real-time communication into a single platform.

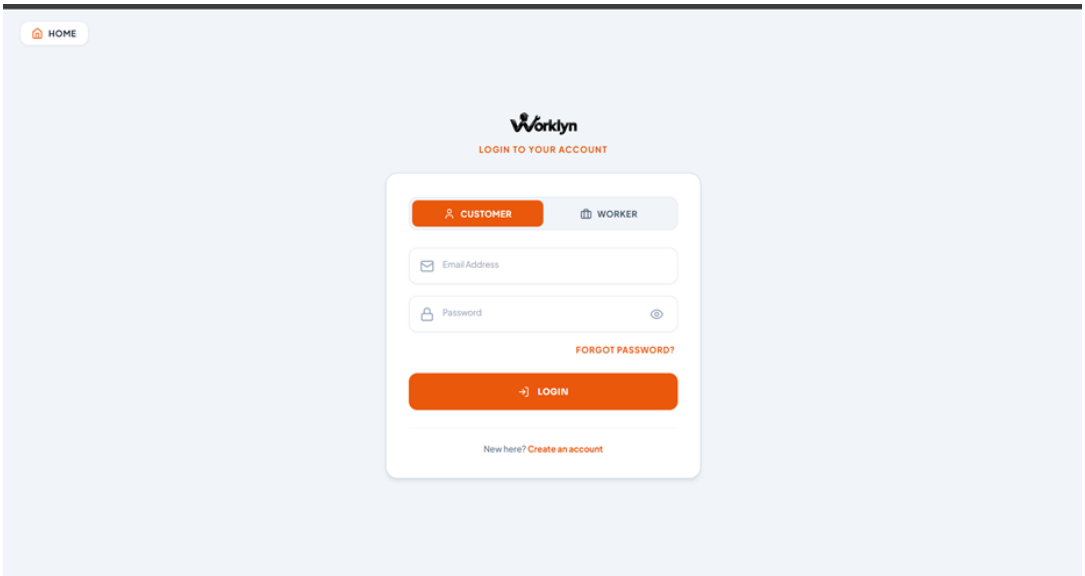
The three-role architecture provides separate interfaces for customers, workers, and administrators. Location-based search helps customers discover nearby professionals, while the matching algorithm ranks workers according to relevant criteria.

The PWA implementation improves accessibility by allowing users to install the website on supported devices without requiring a separate native Android or iOS application.

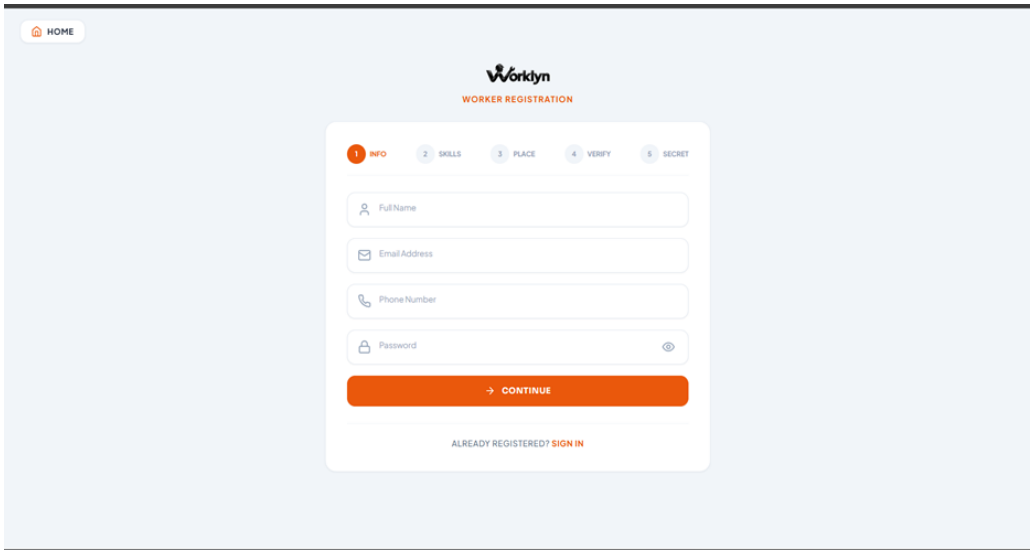
The modular architecture also supports future improvements and additional service categories.



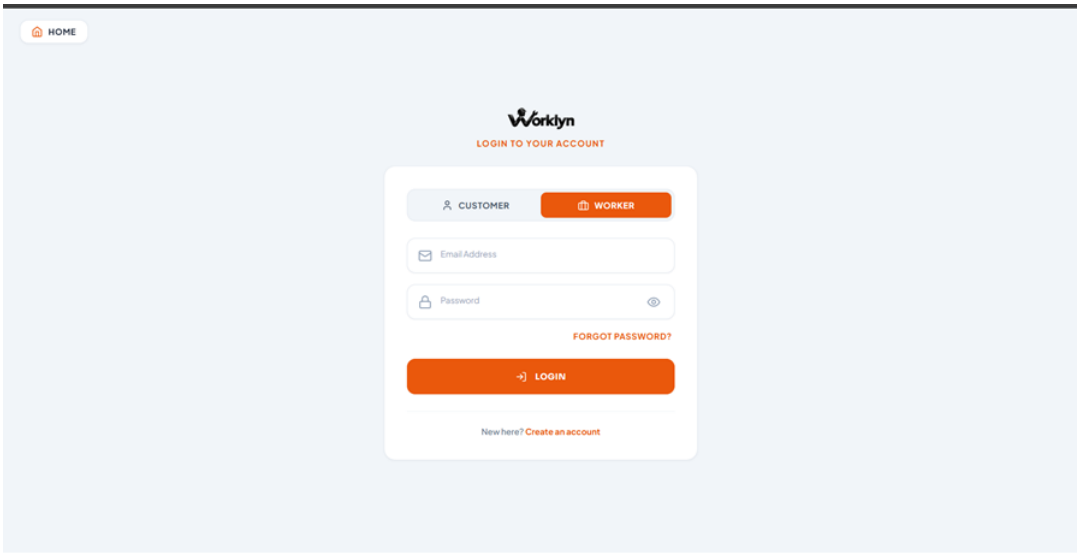
HOME PAGE



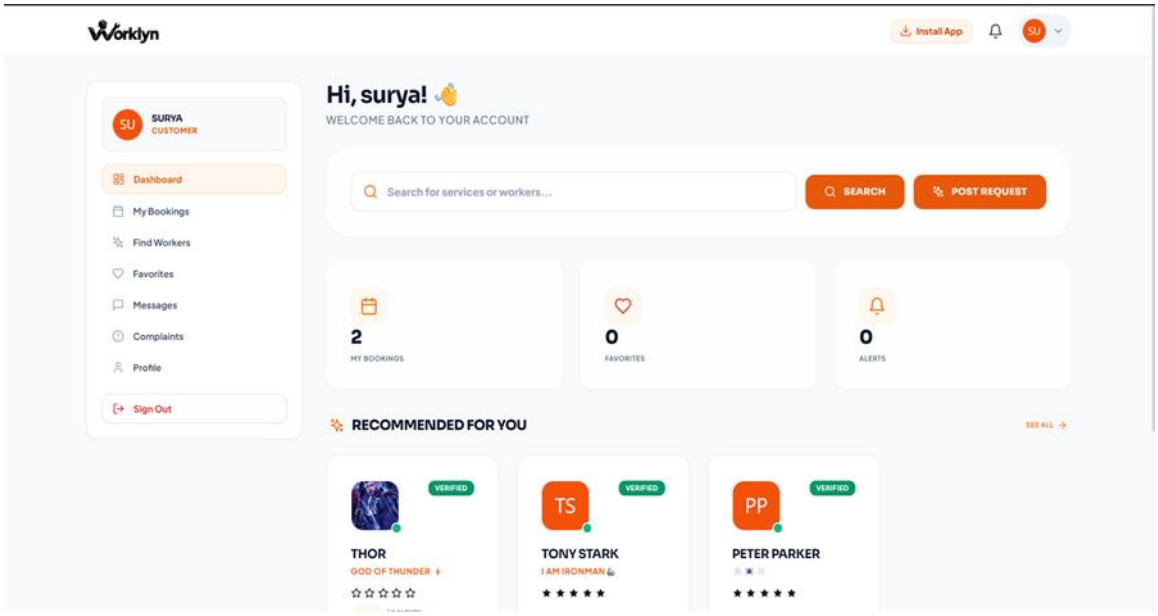
CUSTOMER REGISTRATION



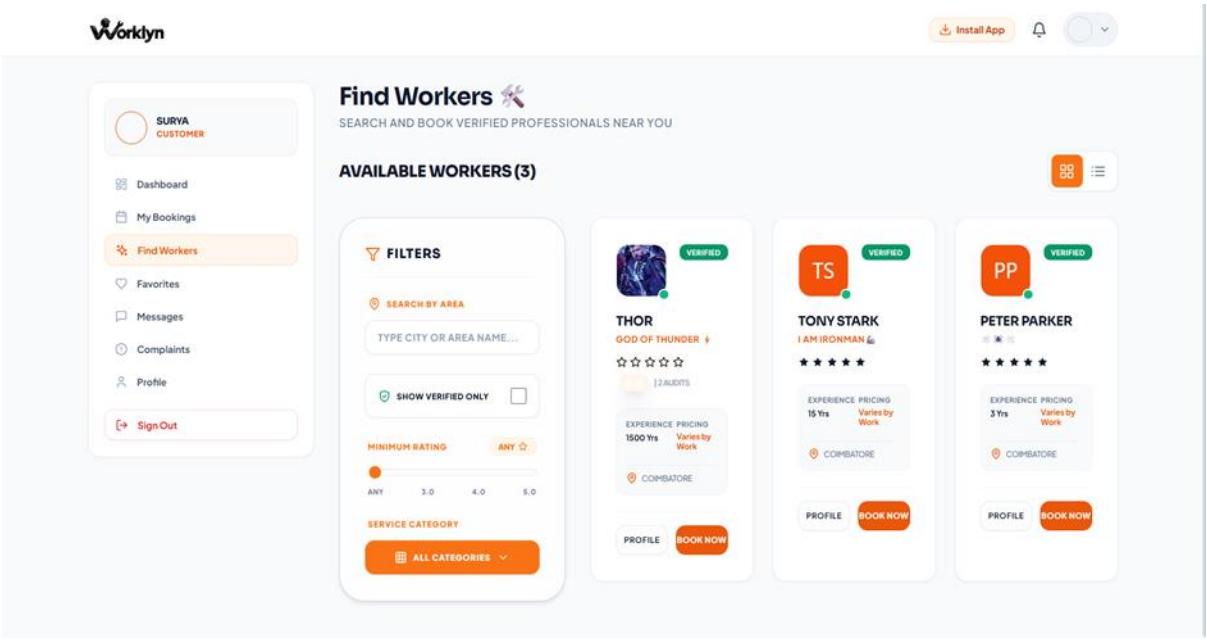
WORKER REGISTRATION



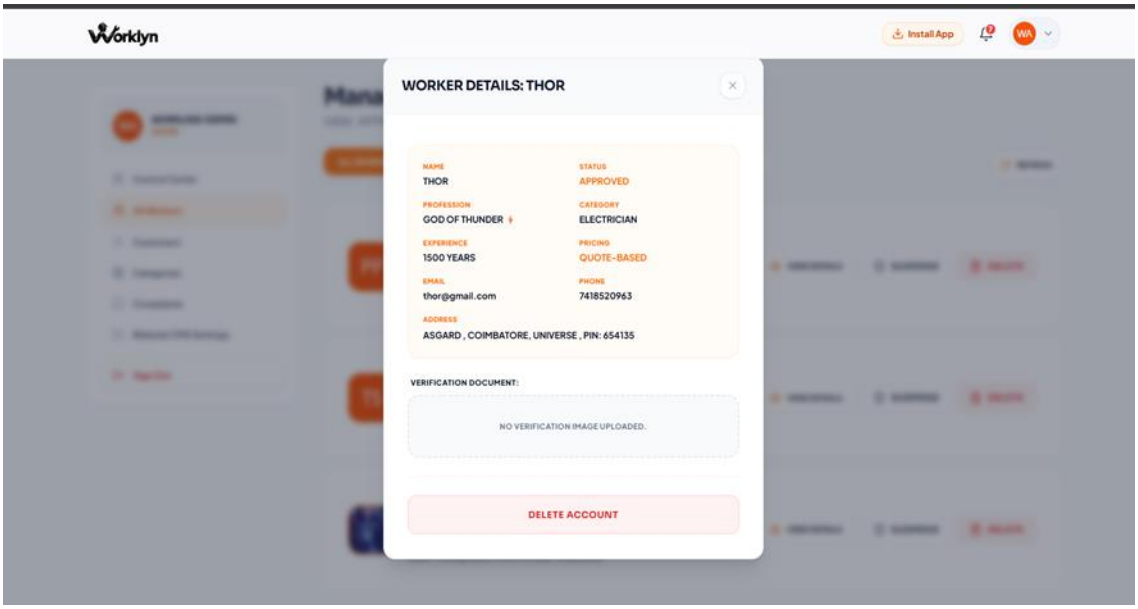
LOGIN PAGE



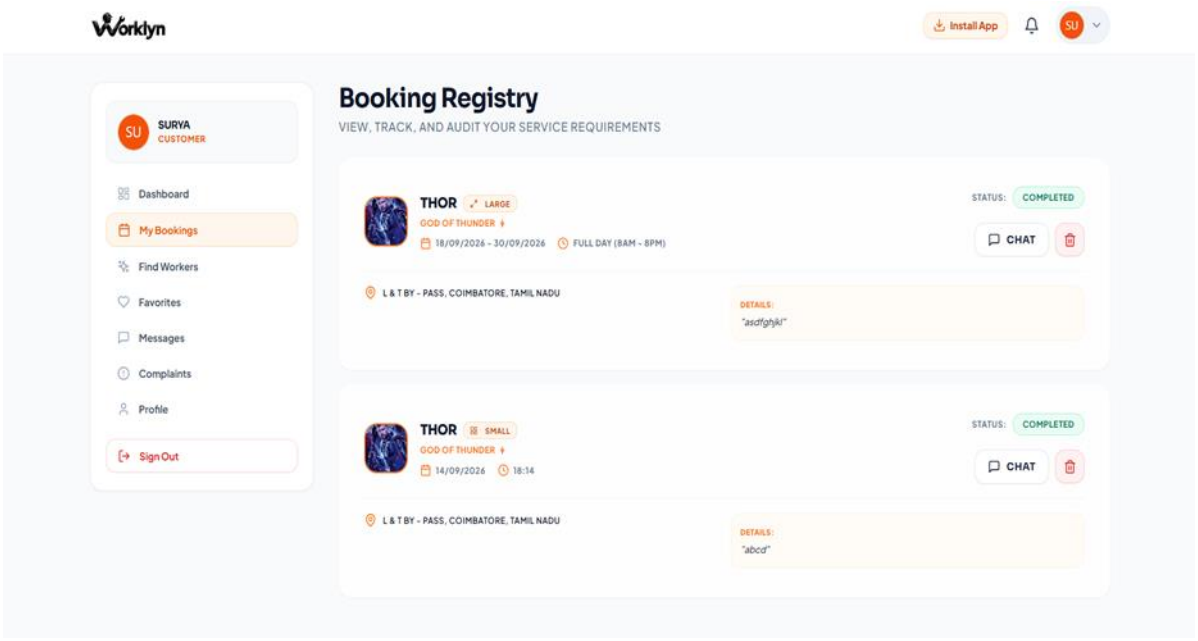
CUSTOMER DASHBOARD



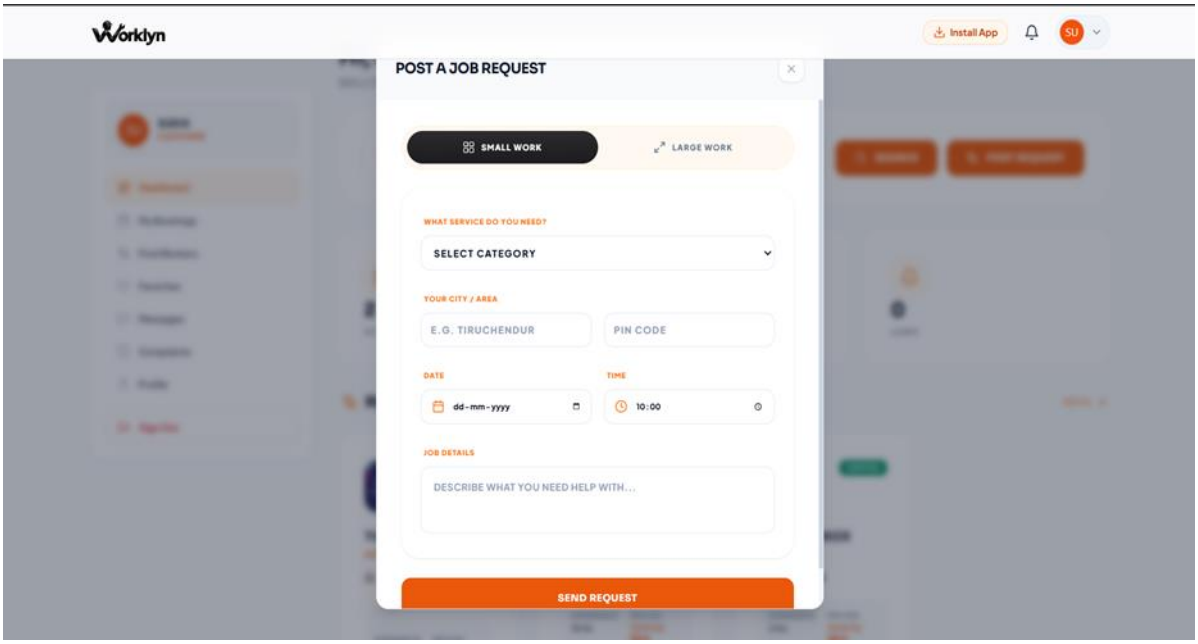
WORKER SEARCH



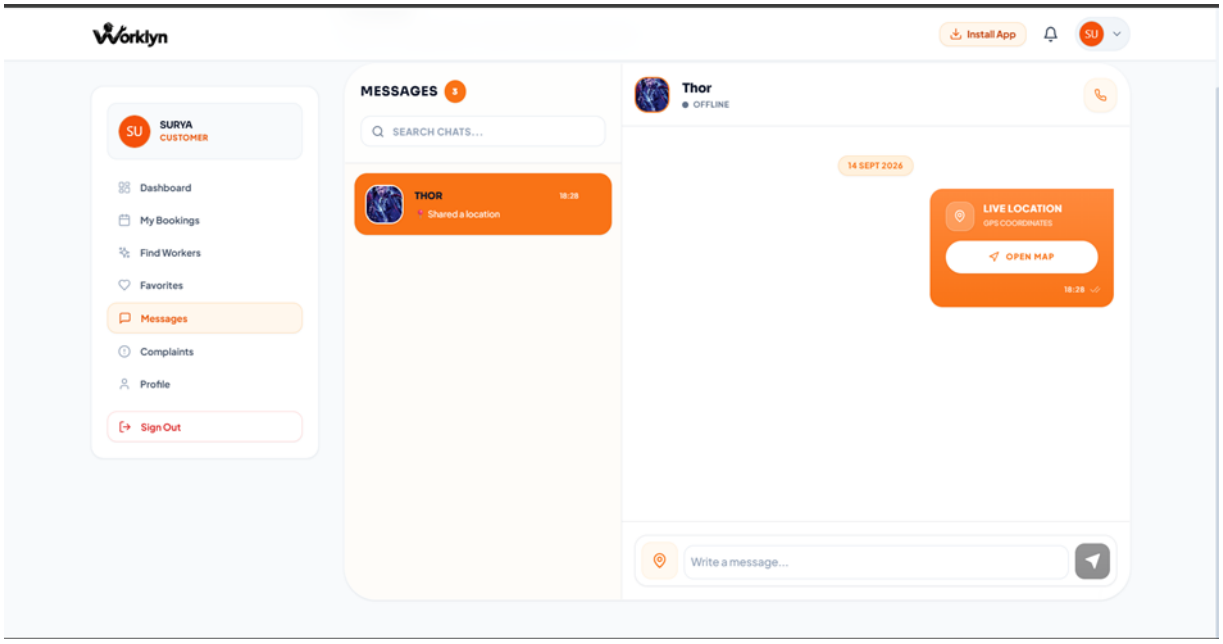
WORKER PROFILE PAGE



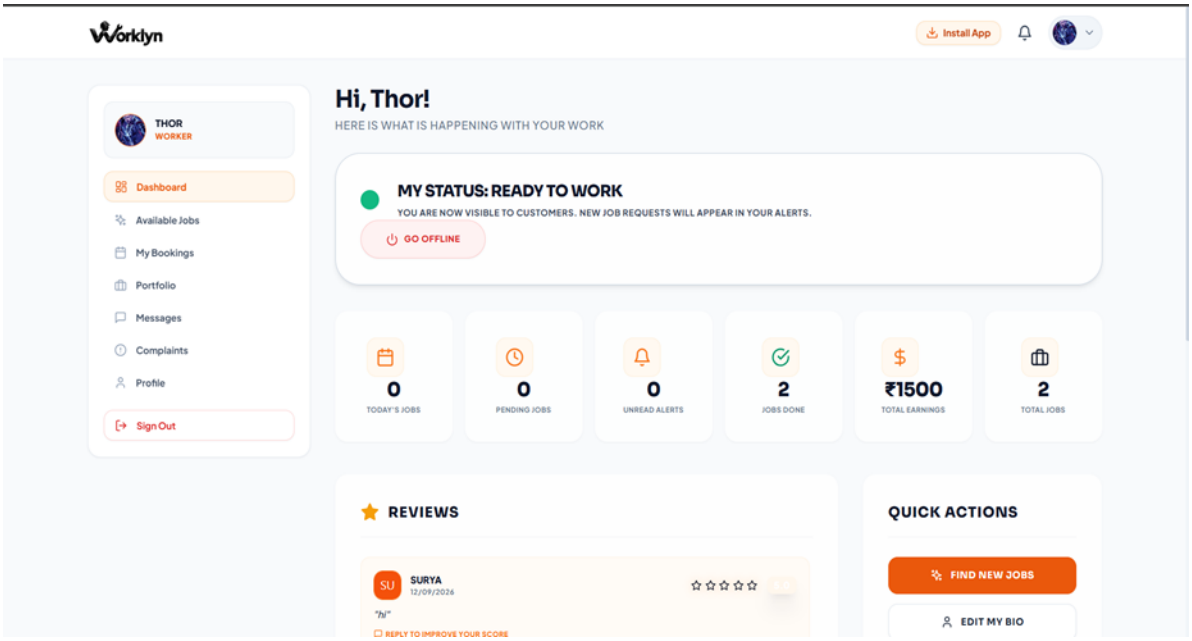
DIRECT BOOKING DIALOG



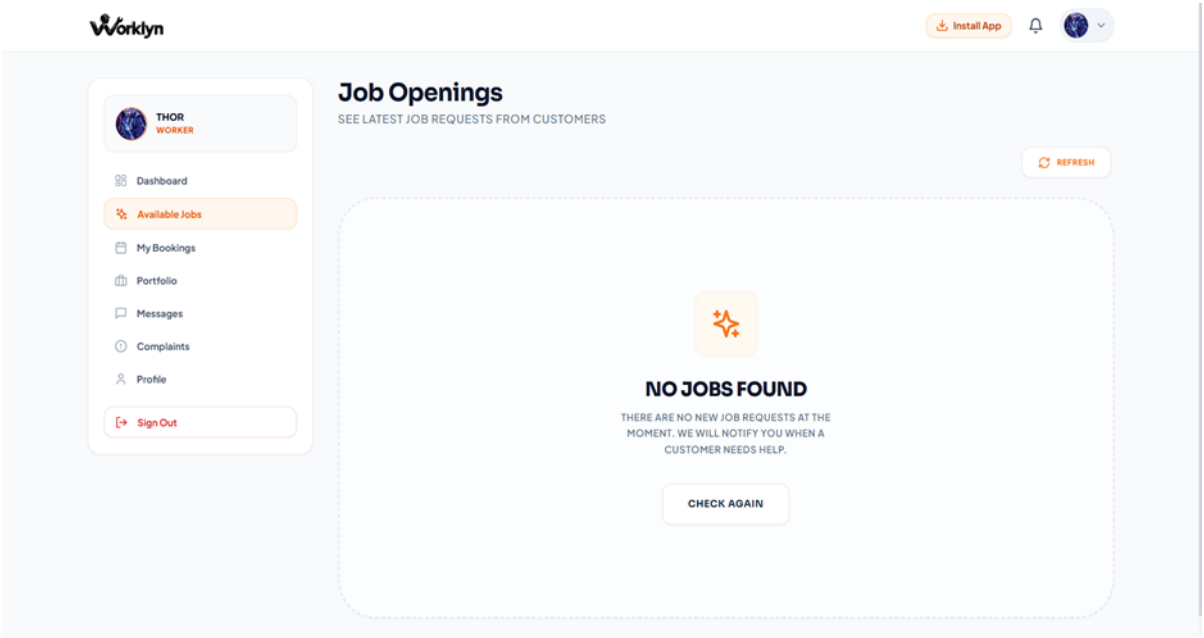
BROADCAST BOOKING DIALOG



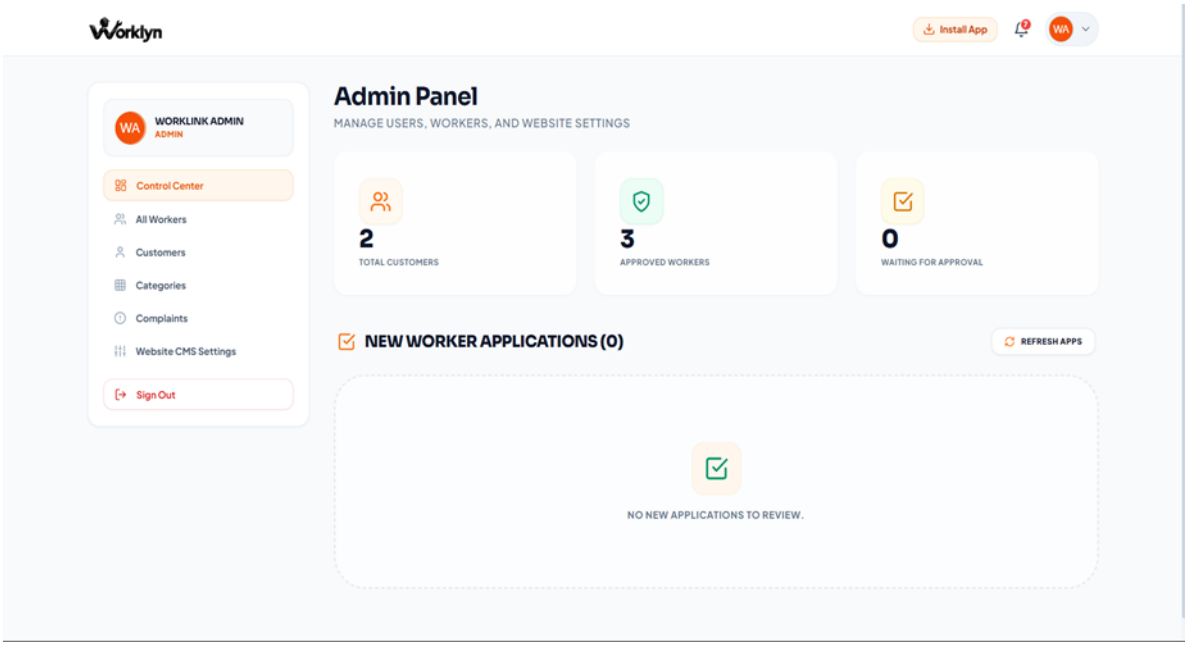
REAL-TIME CHAT WITH LOCATION



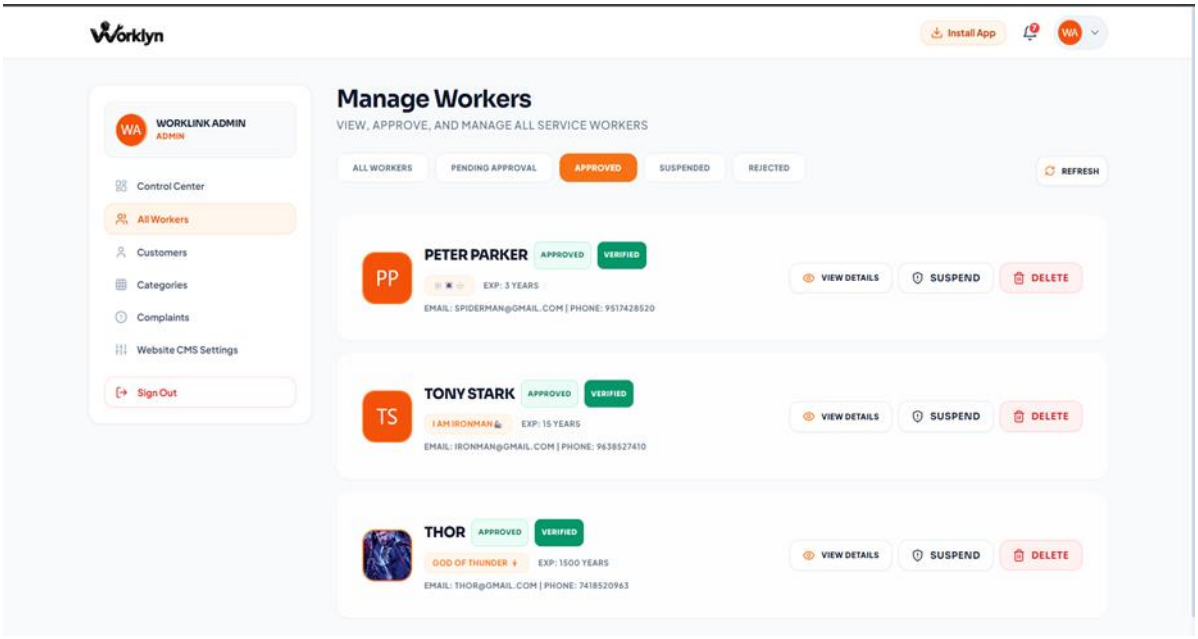
WORKER DASHBOARD



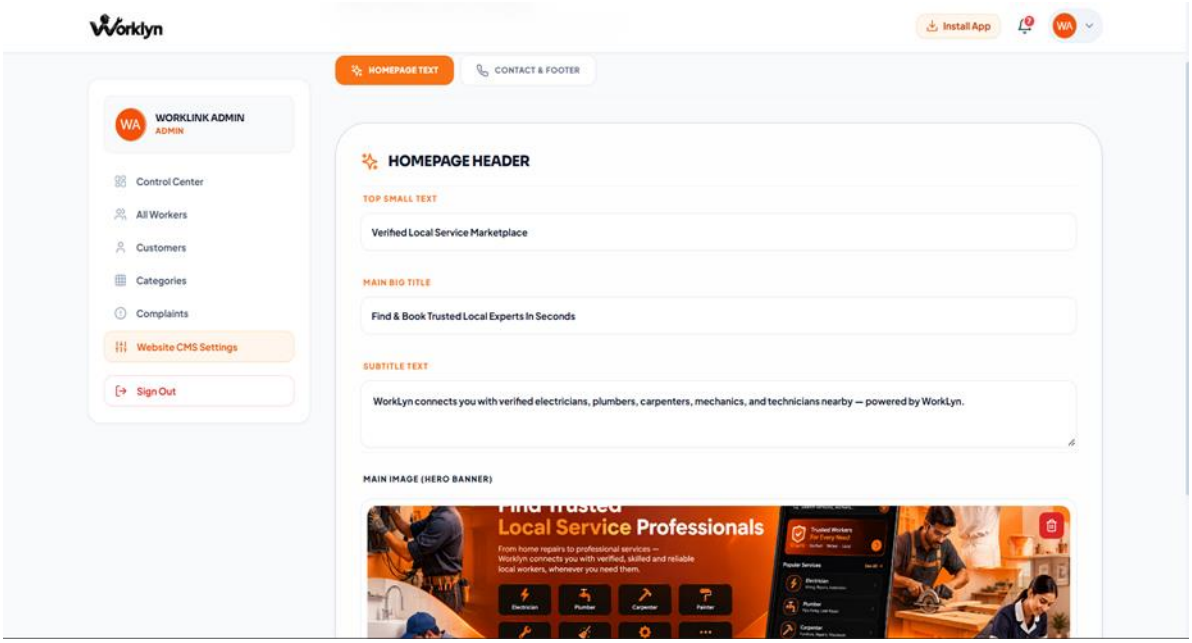
AVAILABLE JOBS



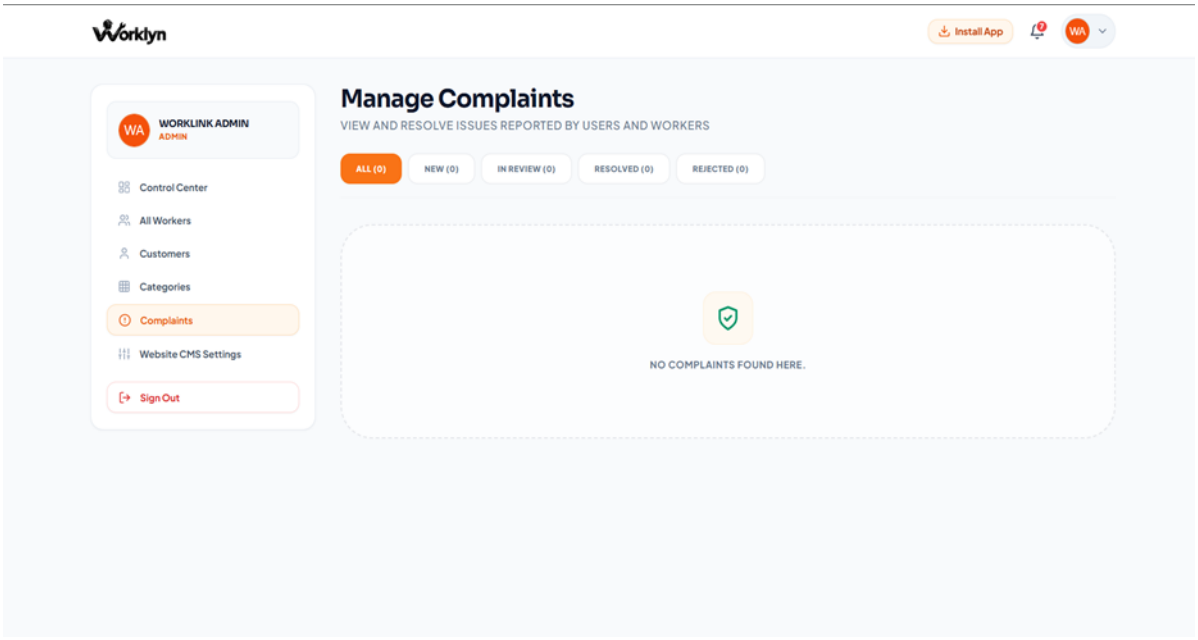
ADMIN DASHBOARD



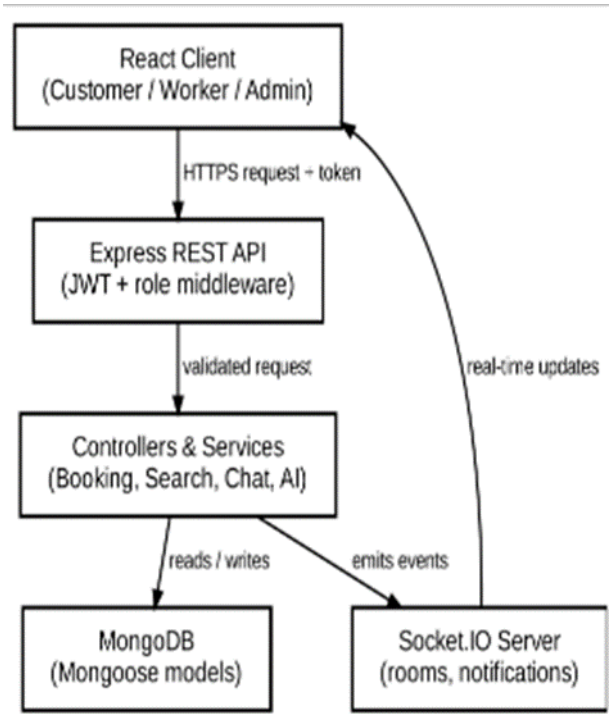
ADMIN WORKER APPROVAL



ADMIN SITE SETTINGS



COMPLAINTS PAGE



Flowchart of Worklyn

8. Future Enhancements

The following enhancements can improve the platform in future versions:

1. Integration of secure online payment facilities.
2. Push notifications for booking updates.
3. Voice-based intelligent search.
4. Support for multiple regional languages.
5. Advanced worker recommendation algorithms.
6. Improved analytics and reporting.
7. Offline access to selected application features.
8. Advanced appointment scheduling.

9. Conclusion

WorkLyn AI provides a centralized digital platform for connecting local service professionals with customers. The application combines intelligent worker matching, location-based search, booking management, secure authentication, and real-time communication to simplify local service discovery. By using React.js, Node.js, Express.js, MongoDB, and Socket.IO, the system provides a modular and scalable architecture. Its Progressive Web Application implementation further improves accessibility by offering an installable application-like experience without requiring a separate native application. WorkLyn AI demonstrates the practical application of modern web technologies in addressing everyday local service requirements.

Acknowledgements

The authors would like to express their sincere gratitude to the management, Head of the Department, faculty members, and project guide of Sri Shakthi Institute of Engineering and Technology for their valuable guidance, encouragement, and support throughout the development of this project.

REFERENCES

1. MongoDB Inc., "MongoDB Documentation," https://www.mongodb.com/docs/.

2. Meta Platforms Inc., "React Documentation," https://react.dev/.

3. OpenJS Foundation, "Node.js Documentation," https://nodejs.org/docs/.

4. OpenJS Foundation, "Express.js Documentation," https://expressjs.com/.

5. Socket.IO, "Socket.IO Documentation," https://socket.io/docs/v4/.

6. MDN Web Docs, "Progressive Web Apps," https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.

7. IETF, "JSON Web Token (JWT)," RFC 7519, 2015.