



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

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



**Impact Factor: 9.935**

**Volume 15, Issue 10, October 2026**

# SkillBridge AI – AI-Based Daily Wage Worker Connect System

Mrs Rajeswari G , Shyamkumar S, Santhoshkumar A

Assistant Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,  
Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,  
Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Pottapalayam,  
Tamil Nadu, India

**ABSTRACT:** Daily wage workers such as plumbers, electricians, carpenters, masons and painters provide essential services, but most of them find work informally, and customers find it difficult to identify skilled, trustworthy workers nearby and to agree on a fair price. This paper presents “SkillBridge AI – AI-Based Daily Wage Worker Connect System”, a full-stack on-demand platform that connects informal workers with nearby customers fairly, securely and transparently. The frontend is developed with React, Vite and Tailwind CSS, the backend is a Spring Boot REST API secured with Spring Security and JSON Web Tokens (JWT) using a PostgreSQL database, and a Python FastAPI microservice provides machine learning, OCR document processing, sentiment analysis and anomaly detection. Workers are verified through mobile OTP, OCR-scanned identity and certification documents and administrative approval. Customers create standard or emergency requests with GPS-assisted location, and the AI engine ranks nearby workers by trade skill, proximity, availability and historical ratings. OpenStreetMap-based OSRM routing computes outbound and return road distances, which together with bike mileage, petrol price and travel allowance give an itemised price that the customer and worker can negotiate and mutually lock. The job is tracked through live status updates and completed by a customer-held six-digit OTP verified by the worker on-site, after which reviews are analysed using NLP and audit records are stored.

**KEYWORDS:** Daily wage workers, On-demand services, Worker verification, OCR, AI recommendation, OSRM, Price negotiation, OTP, Sentiment analysis, Anomaly detection, Spring Boot, FastAPI.

## I. INTRODUCTION

Daily wage workers such as plumbers, electricians, carpenters, masons and painters form the backbone of household maintenance and small construction work. Most of them work in the informal sector and find jobs through personal contacts or local labour gatherings. Customers, in turn, find it difficult to identify skilled, trustworthy workers nearby at short notice, and wages, materials and travel charges are often settled through hurried bargaining that leads to disputes. SkillBridge AI – AI-Based Daily Wage Worker Connect System is a full-stack, AI-powered on-demand service platform developed to bridge informal workers and nearby customers fairly, securely and transparently. The system uses a distributed architecture: a React, Vite and Tailwind CSS frontend [9], [10], a Spring Boot REST API with Spring Security, JWT and a PostgreSQL database [8], [11], [13], and a specialised Python FastAPI microservice [12] that provides machine learning, OCR document processing, sentiment analysis and anomaly detection.

The platform follows an end-to-end, trust-driven workflow. Workers register, verify their mobile number by OTP, upload identity and certification documents that are scanned by OCR, and are approved by an administrator. Customers create standard or emergency requests using the browser’s Geolocation API and interactive maps, and the AI engine recommends the best matching nearby workers. Road-based travel distances from OSRM [4], [14] are used to prepare an itemised price, which the customer and worker negotiate and lock together. The worker navigates to the customer with live status updates, and the job is closed by a secure, customer-held six-digit OTP that the worker verifies on-site.

The paper is organised as follows. Section II reviews related work. Section III describes the existing system and its limitations. Section IV presents the proposed system. Section V explains the system architecture, UML diagrams, database design and requirements. Section VI describes the implementation modules. Section VII discusses the results, Section VIII covers system testing, and Section IX presents the conclusion and future enhancements.

#### **A. Problem Statement**

Informal workers and customers operate with very little reliable information. Customers cannot easily verify the identity or skill of a worker, workers cannot be sure that the agreed wage will be respected, and travel cost is either ignored or fixed arbitrarily. There is also no secure way of confirming that a job has been completed. A platform is therefore needed that verifies workers, recommends suitable ones, makes the price transparent and negotiable, and confirms completion in a way that both parties can trust.

#### **B. Objectives**

- To verify workers through mobile OTP, OCR-based document scanning and administrative approval.
- To allow standard and emergency service requests with GPS-assisted location capture.
- To recommend nearby workers using trade skill, proximity, availability and historical ratings.
- To compute outbound and return road distances using OSRM and derive a transparent travel charge.
- To provide itemised prices with offers, counter-offers and mutual price locking.
- To track job status live and confirm completion through a customer-held six-digit OTP.
- To analyse reviews with NLP and detect anomalies so that administrators can maintain audit records.

## **II. LITERATURE SURVEY**

Research on gig-economy and on-demand platforms studies how workers should be matched with jobs. Liu et al. show that improved matching technology can reveal information about labour demand and change workers' participation decisions, so matching design must consider worker behaviour [1]. Ma et al. analyse sequential job offers and show that ranking workers greedily by match quality can be optimal under suitable conditions [2]. Sekiya et al. report a field experiment on a spot-work platform in which controlling the exposure of recommendations improved realised matches [3]. These works are mostly theoretical or concern shift-based jobs, and they do not combine worker verification, travel-cost calculation, price negotiation and secure completion for individual skilled trades.

For travel cost, Huber and Rust describe how OSRM and OpenStreetMap data can be used to calculate travel distance and time without commercial map providers [4]. Their tool is intended for batch analysis in a statistical package, whereas the proposed system integrates OSRM into a web backend to obtain real road distances for every job.

Tesseract is an open-source OCR engine that converts document images into text [5]. OCR extracts text but does not decide whether a document is genuine, so extracted text must be reviewed by a person. For reviews, Hutto and Gilbert present VADER, a rule-based sentiment analysis model that performed well on social media text [6]; such models are designed for English and may need adaptation for local-language reviews.

Isolation Forest is an unsupervised anomaly detection method that isolates unusual observations quickly [7]. It flags statistical outliers but does not explain them, so flagged cases need human review. JSON Web Tokens provide a compact way of carrying identity claims between a client and a server [8], but their security depends on proper signing, expiry and the use of HTTPS. The proposed system combines these ideas in a single workflow and keeps administrators in the loop for sensitive decisions.

## **III. EXISTING SYSTEM**

Existing arrangements for hiring daily wage workers rely largely on personal contacts, local labour gatherings, telephone calls and general classified listings. Some online home-service marketplaces exist, but they are generally oriented towards organised service providers, use fixed price lists, add platform fees and give limited visibility into how workers are ranked. These approaches share the following limitations:

- Workers are discovered through informal contacts, which limits their work opportunities and income.
- There is no reliable verification of identity or skill before a worker visits a customer's home.
- Prices are settled by hurried bargaining and are not itemised into labour, materials and travel.
- Travel distance and fuel cost are ignored or fixed arbitrarily, which is unfair to workers who travel far.
- Either party can change the agreed price after the work begins, which causes disputes.
- There is no secure way of confirming that the work has been completed.
- Reviews are not analysed, and unusual behaviour is neither monitored nor recorded.

#### **IV. PROPOSED SYSTEM**

The proposed SkillBridge AI is a web-based system that provides an integrated workflow combining worker verification, GPS-assisted job creation, AI-based worker recommendation, road-based price estimation, mutual price negotiation, live tracking, OTP-based completion, review analysis and audit. It is developed using React, Vite and Tailwind CSS [9], [10] for the user interface, Spring Boot with Spring Security and JWT [8], [11] for the REST API, PostgreSQL [13] for data storage, Python FastAPI [12] for the AI microservice, and OpenStreetMap and OSRM [4], [14] for maps and routing.

##### **A. Multi-Step Worker Verification**

A worker registers and first verifies the mobile number through a one-time password. The worker then uploads identity and certification documents, which are scanned by the OCR service in the FastAPI microservice to extract the text. An administrator reviews the extracted information and the original documents and approves or rejects the worker. Only approved workers can receive job requests.

##### **B. GPS-Assisted Job Creation**

A customer creates a standard or emergency service request by selecting the trade, describing the work and confirming the location. The location is captured using the browser's Geolocation API and can be adjusted on an interactive map based on OpenStreetMap.

##### **C. AI Worker Recommendation**

The AI engine in the FastAPI microservice ranks nearby approved workers and recommends the best matches. The ranking considers trade skill, proximity, availability and historical ratings, so that the customer sees the most suitable workers first.

##### **D. Road Travel Distance and Price Estimation**

After a worker is selected, the backend calls OSRM to calculate the outbound road distance from the worker to the customer and the return distance. These distances are combined with a configurable bike mileage, the live petrol price and a travel allowance to compute the fuel travel charge, which is shown with the labour and material charges as an itemised breakdown (Table I).

**TABLE I. COMPONENTS OF THE ITEMISED PRICE**

| <b>Component</b> | <b>Basis</b>                                                       | <b>Description</b>                                        |
|------------------|--------------------------------------------------------------------|-----------------------------------------------------------|
| Labour           | Trade and work involved                                            | Charge for the worker's skilled work                      |
| Materials        | Materials required for the job                                     | Cost of materials needed to complete the work             |
| Fuel travel      | OSRM outbound and return distance, bike mileage, live petrol price | Fuel cost of the worker's travel to and from the customer |
| Travel allowance | Configurable allowance                                             | Allowance added to the travel charge                      |
| Total            | Sum of the components                                              | Amount that the customer and worker negotiate and lock    |

##### **E. Mutual Price Negotiation and Price Locking**

The itemised estimate is the starting point for negotiation. The customer and the worker can propose offers and counter-offers until both accept the same amount. Once the price is locked, neither party can change it unilaterally, which removes a common cause of wage disputes.

#### F. Live Navigation and OTP-Based Completion

The worker navigates to the customer and updates the job status as ON\_THE\_WAY, ARRIVED and IN\_PROGRESS in real time. When the service ends, the customer generates a cryptographically secure six-digit OTP, which the worker must verify on-site; authentication practices follow general security guidelines [8], [15]. The job is marked complete only if the OTP is correct, which protects both customer satisfaction and fair payment.

#### G. Reviews, Sentiment Analysis, Anomaly Detection and Audit

After completion, the customer can submit a review, and its text is analysed by the sentiment analysis component. The anomaly detection component monitors jobs, prices and behaviour and flags unusual patterns, and important events are stored as audit records for administrators. Flagged cases are reviewed by a person and are not acted upon automatically.

### V. SYSTEM ARCHITECTURE AND REQUIREMENTS

#### A. System Architecture

The system follows a layered, distributed architecture (Fig. 1). The React single page application provides the customer dashboard, worker dashboard, administrator console and map components, and it communicates with the Spring Boot REST API over HTTPS using JSON and a JWT. The Spring Boot application manages authentication and roles, worker verification, job management, travel and price calculation, negotiation, tracking, OTP completion, reviews and audit, and stores data in PostgreSQL. It calls the FastAPI microservice for recommendation, OCR, sentiment analysis and anomaly detection, and it uses external services such as OSRM, OpenStreetMap tiles, an OTP gateway and a petrol price source.

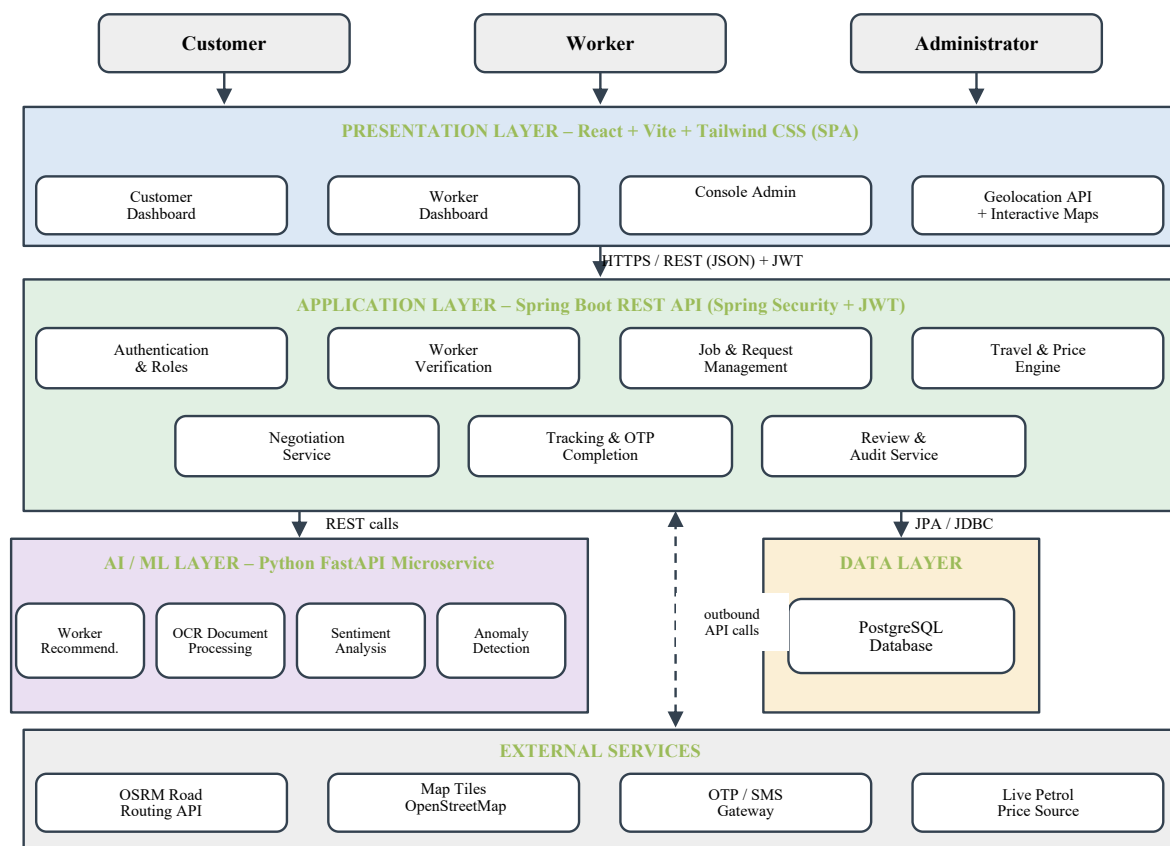


Fig.1. System Architecture

## B. Data Flow

The data flow begins when a worker registers and submits documents, which are stored and approved by the administrator (Fig. 2). A customer's service request is saved and passed to the AI recommendation process, and the selected worker is passed to the travel and price estimation process. The itemised estimate goes to the negotiation process, and the locked price is passed to the tracking and OTP completion process. Completed jobs flow to the review and audit process, which stores reviews and audit logs that the administrator can view.

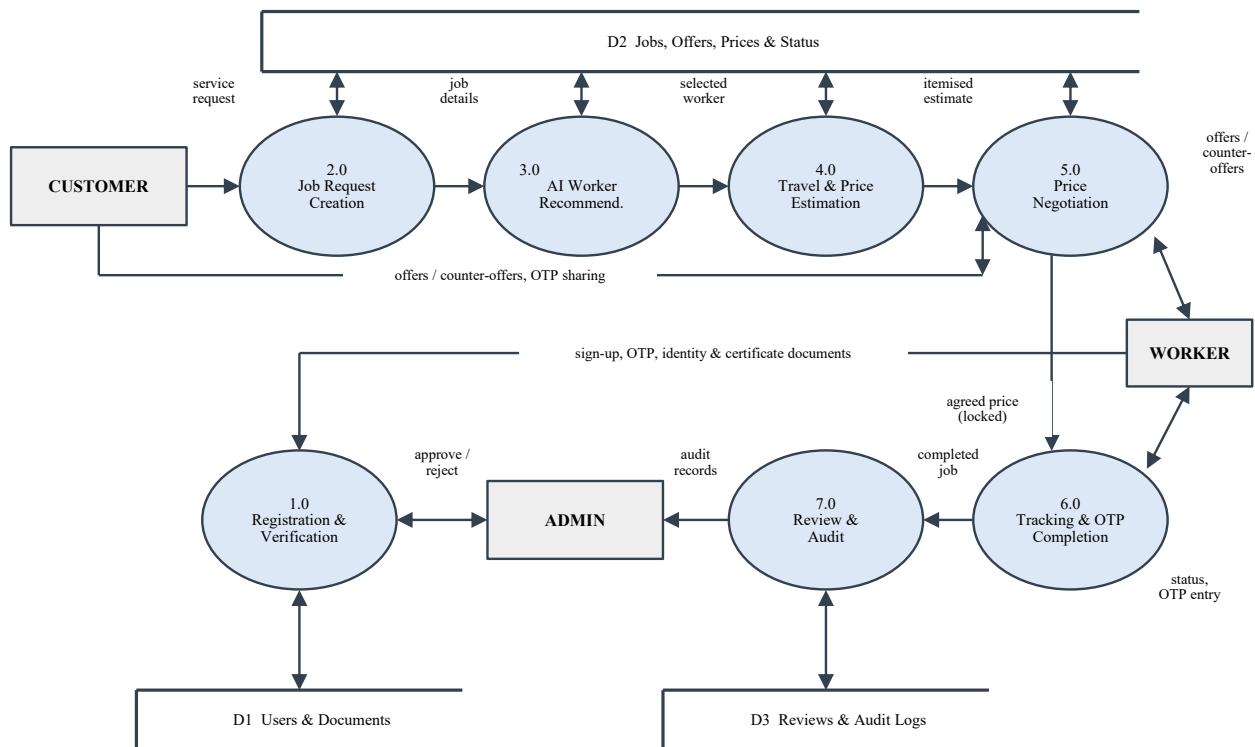


Fig.2. Data Flow Diagram

## C. UML Diagrams

Fig. 3 shows the use case diagram with three actors. The customer registers, creates requests, views recommended workers and price breakdowns, negotiates, tracks the worker, shares the completion OTP and submits a review. The worker registers and verifies, accepts jobs, proposes counter-offers, updates status and verifies the OTP. The administrator approves workers, monitors anomalies, views audit records and configures rates.

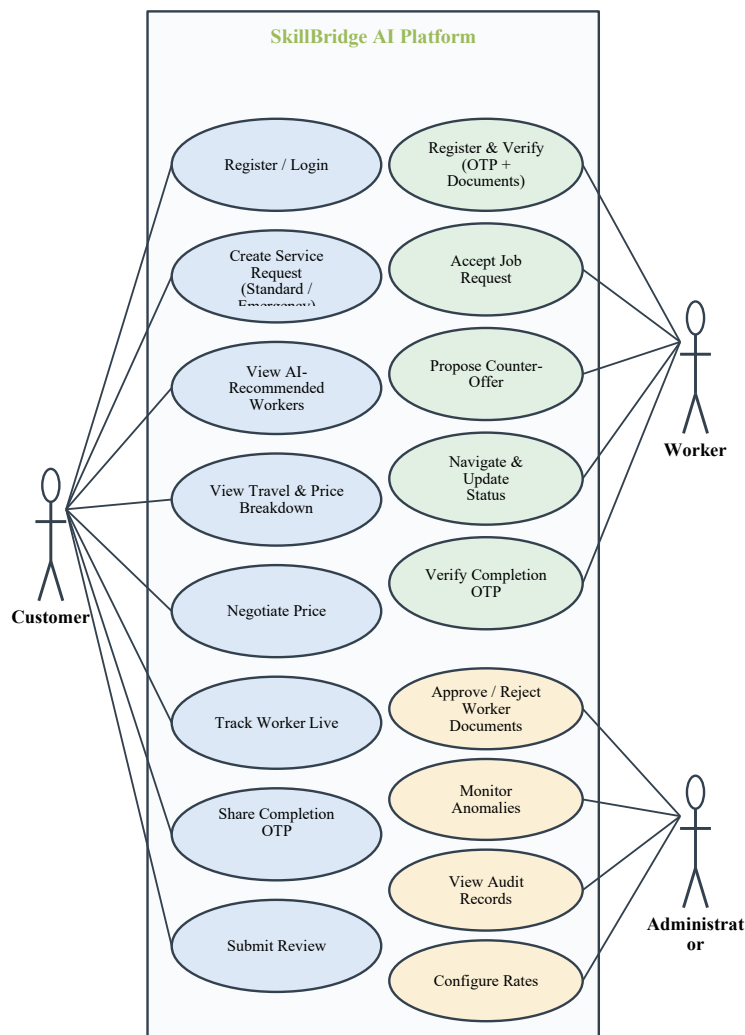


Fig.3. Use Case Diagram

The activity diagram in Fig. 4 shows the workflow from login and request creation to AI ranking, price estimation, negotiation, price locking, status updates, OTP verification and completion. Fig. 5 shows the sequence of messages between the customer, the React frontend, the Spring Boot API, the FastAPI AI service, OSRM and the database, and the worker.

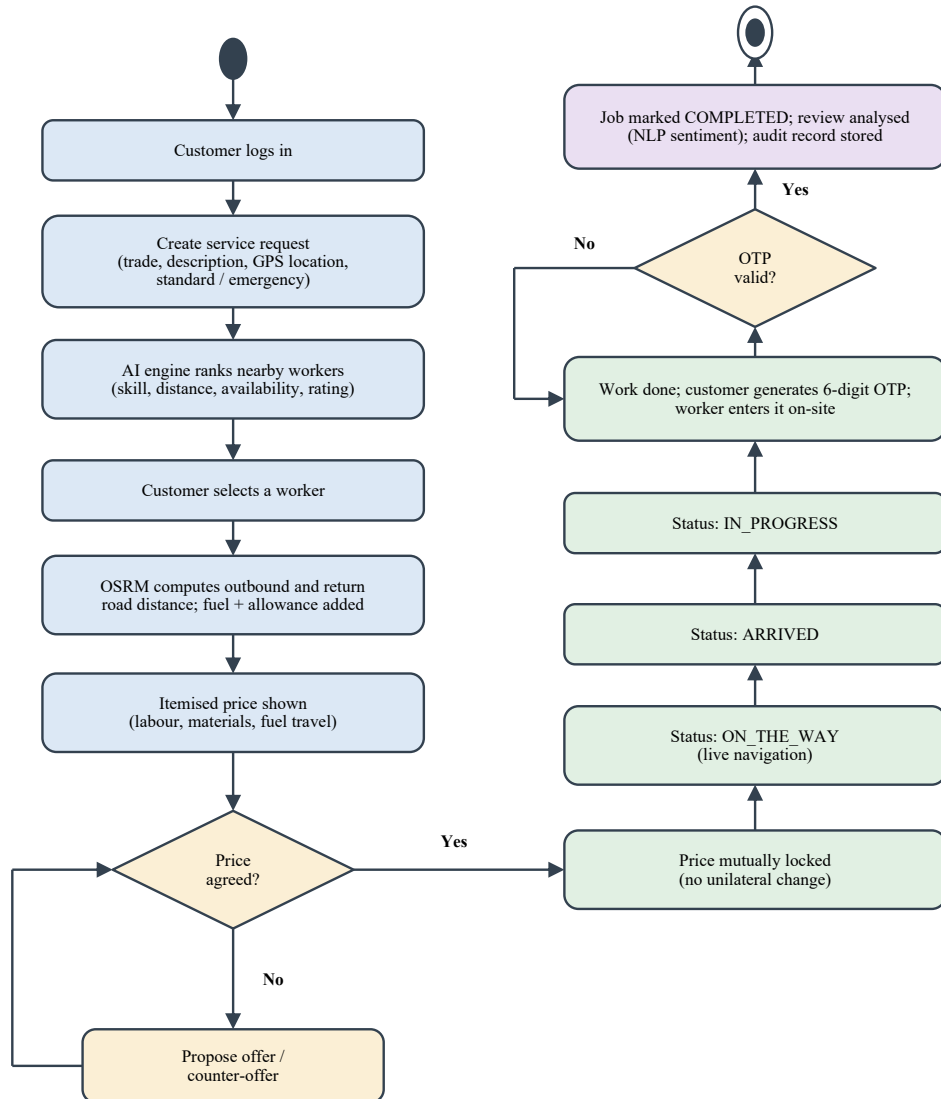


Fig.4. Activity Diagram

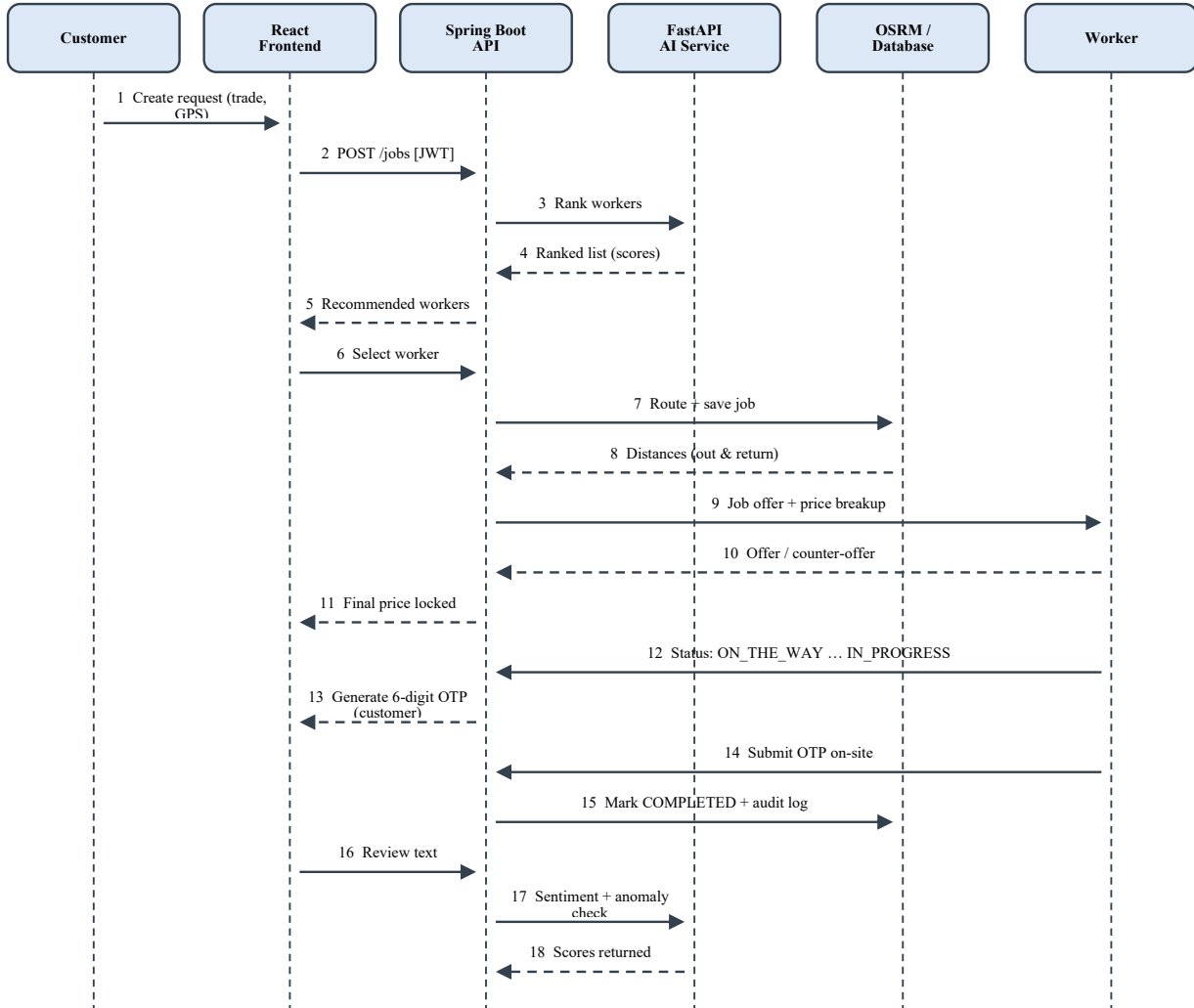


Fig.5. Sequence Diagram

#### D. Database Design

The PostgreSQL database stores the data needed by all modules. The principal logical entities are summarised in Table II. PLACEHOLDER: confirm the entity, table and column names against the implemented PostgreSQL schema before submission.

TABLE.II. PRINCIPAL DATABASE ENTITIES

| Entity           | Purpose                                                                       |
|------------------|-------------------------------------------------------------------------------|
| Users            | Account details and role (customer, worker or administrator) of every user    |
| Worker profiles  | Trade, availability, location, rating and verification status of workers      |
| Worker documents | Uploaded identity and certification documents, OCR text and approval decision |
| Jobs             | Service requests with trade, location, request type, status and locked price  |
| Price offers     | Offers and counter-offers with labour, material and travel components         |
| Status history   | Time-stamped job status changes from ON THE WAY to COMPLETED                  |
| Reviews          | Ratings, review text and sentiment result for completed jobs                  |
| Audit logs       | Important events and anomaly flags for administrative review                  |

#### E. Hardware and Software Requirements

The system requires a computer with an Intel Core i5 or equivalent processor, 8 GB or more RAM, a stable internet connection for OSRM, OpenStreetMap and OTP services, and a display of at least  $1366 \times 768$  resolution; customers and workers use a smartphone or computer with a modern browser and location support. PLACEHOLDER: confirm the actual hardware used for development and deployment. The software components are listed in Table III.

TABLE.III. SOFTWARE REQUIREMENTS

| Component         | Technology                                                                                                                  |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Operating System  | Windows / Linux                                                                                                             |
| Frontend          | React, Vite, Tailwind CSS                                                                                                   |
| Backend           | Spring Boot (Java), Spring Security                                                                                         |
| Authentication    | JSON Web Token (JWT), mobile OTP                                                                                            |
| Database          | PostgreSQL                                                                                                                  |
| AI Microservice   | Python, FastAPI                                                                                                             |
| AI / ML Functions | Worker recommendation, OCR, sentiment analysis, anomaly detection<br>PLACEHOLDER: list the actual libraries and models used |
| Maps and Routing  | Browser Geolocation API, OpenStreetMap, OSRM                                                                                |
| Development Tool  | PLACEHOLDER: IDE / editor used                                                                                              |

#### VI. SYSTEM IMPLEMENTATION

The system is implemented as ten interconnected modules that follow the sequence: User Authentication → Worker Verification → GPS-Assisted Job Creation → AI Worker Recommendation → Travel and Price Estimation → Price Negotiation → Live Tracking → OTP-Based Completion → Reviews and Sentiment Analysis → Admin Audit.

- **User Authentication Module:** registers customers and workers, issues a JWT after login and applies role-based access through Spring Security.
- **Worker Verification Module:** verifies the mobile number by OTP, accepts identity and certification documents, passes them to the OCR service and records the administrator's approval or rejection.
- **Job Creation Module:** captures the trade, description, request type and GPS location, which the customer can adjust on an interactive map.
- **AI Recommendation Module:** ranks nearby approved workers by trade skill, proximity, availability and historical ratings using the FastAPI microservice.
- **Travel and Price Estimation Module:** obtains outbound and return road distances from OSRM and adds bike mileage, live petrol price and travel allowance to produce the itemised price.
- **Negotiation Module:** stores offers and counter-offers and locks the price when both parties accept the same amount.
- **Tracking Module:** records live status updates (ON\_THE\_WAY, ARRIVED, IN\_PROGRESS) and shows them to the customer.
- **OTP Completion Module:** generates a secure six-digit OTP for the customer and completes the job only when the worker verifies it on-site.
- **Review and Sentiment Module:** accepts reviews for completed jobs and analyses the review text using NLP sentiment analysis.
- **Admin and Audit Module:** supports worker approval, rate configuration, anomaly flags and audit records.

#### VII. RESULTS AND DISCUSSION

The developed system provides an integrated web interface for customers, workers and administrators. Customers create requests, view recommended workers with the itemised price, negotiate and track the job. Workers complete verification, respond to offers, update their status and verify the completion OTP, and administrators approve workers and inspect audit records. The outputs of each stage are shown in Figs. 6 to 8.

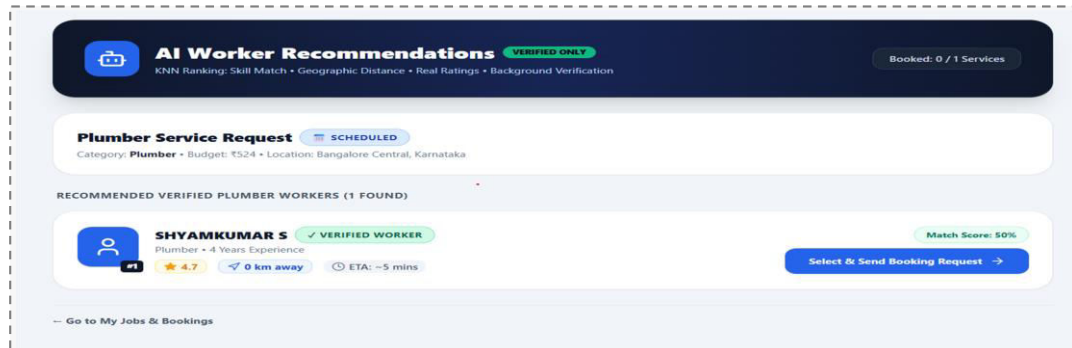


Fig.6. Job Creation and AI-Recommended Workers

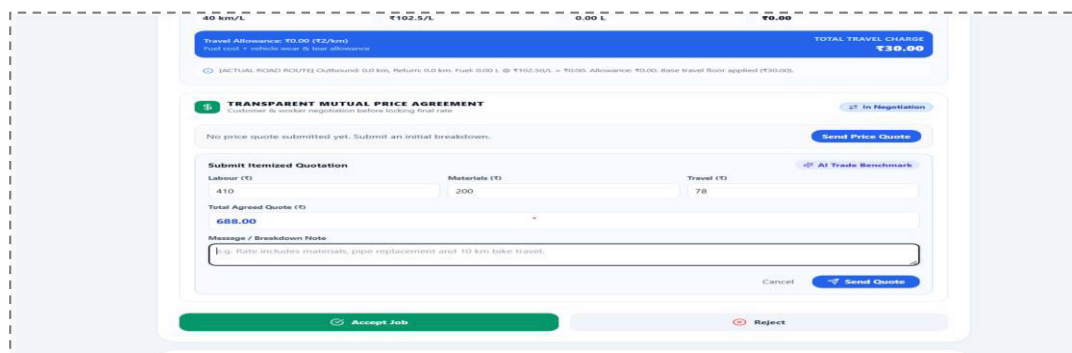


Fig.7. Itemised Price and Negotiation

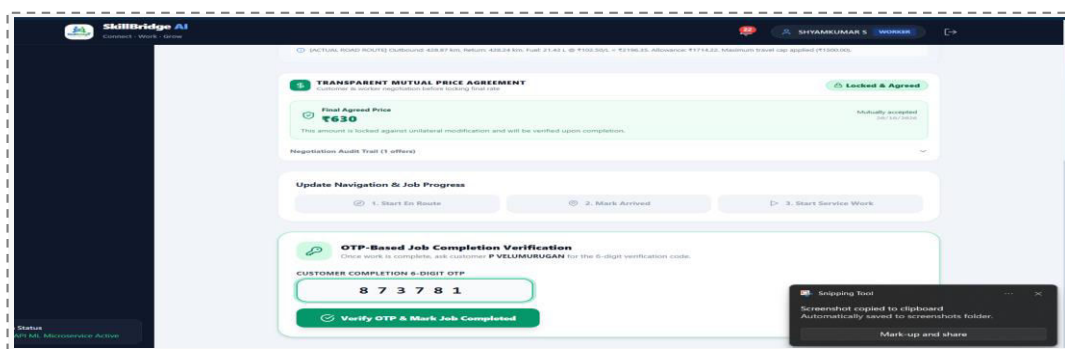


Fig.8. Completion OTP Verification

The itemised price makes the discussion between customer and worker transparent, because the travel charge is derived from real road distances and configurable rates instead of a guessed amount. The mutual price lock prevents unilateral changes, and the customer-held OTP prevents a job from being closed without the customer's participation. The AI outputs are intended to assist customers and administrators and not to replace human judgement: recommendations are ranked suggestions, OCR text is reviewed by an administrator, and anomaly flags are examined by a person. PLACEHOLDER: insert actual measured results from your testing, for example recommendation response time, OCR extraction accuracy and sentiment analysis outcomes. No measured values have been assumed in this report.

## VIII. SYSTEM TESTING

The testing plan covers five levels. Unit testing verifies individual modules, functional testing executes the complete workflow from registration to review, integration testing verifies data flow between the React frontend, Spring Boot

API, FastAPI microservice, OSRM and PostgreSQL, performance testing observes response times under multiple users, and security testing checks role-based access, JWT handling, price locking and OTP verification. Representative unit test cases are given in Table IV. PLACEHOLDER: record the actual test outcomes before submission.

**TABLE .IV.SAMPLE UNIT TEST CASES**

| Module              | Test Performed                                   | Expected Result                                                                      |
|---------------------|--------------------------------------------------|--------------------------------------------------------------------------------------|
| Login               | Valid / invalid credentials                      | User receives a JWT and opens the dashboard / access is denied with an error message |
| Worker Verification | Correct / wrong mobile OTP                       | Correct OTP verifies the number / wrong OTP is rejected                              |
| Document Upload     | Identity document image                          | Text is extracted by OCR and shown to the administrator                              |
| Job Creation        | Standard and emergency request with GPS location | Request is saved with coordinates and request type                                   |
| Recommendation      | Request for a given trade                        | Only approved workers of that trade are ranked and listed                            |
| Price Estimation    | Known distances and rates                        | Fuel travel charge and total match the manual calculation                            |
| Negotiation         | Offer, counter-offer and acceptance              | Price is locked and cannot be changed afterwards                                     |
| OTP Completion      | Correct / wrong / expired OTP                    | Job is completed only with the correct OTP                                           |
| Review              | Review for a completed job                       | Review is saved and a sentiment result is produced                                   |
| Audit               | Open audit records as administrator              | Events and anomaly flags are listed with user, action and time                       |

## **IX. CONCLUSION AND FUTURE ENHANCEMENT**

SkillBridge AI – AI-Based Daily Wage Worker Connect System provides an integrated and trust-driven approach for connecting informal workers with nearby customers. By combining multi-step worker verification, GPS-assisted job creation, AI-based recommendation, OSRM road-based price estimation, mutual price locking, live tracking and customer-held OTP completion, it addresses the problems of trust, pricing and confirmation of completion in the informal service sector. NLP-analysed reviews, anomaly detection and audit records further strengthen accountability, and the system is intended to assist administrators rather than replace their judgement.

Future enhancements include online payment integration, multilingual and voice-based interfaces for workers with limited literacy, native mobile applications with push notifications, scheduled bookings and additional trades and regions, exposure balancing so that newly joined workers receive fair visibility, a combined trust score for workers, and richer anomaly detection using administrator feedback.

## **REFERENCES**

- [1] Y. Liu, B. Lou, X. Zhao and X. Li, “Unintended Consequences of Advances in Matching Technologies: Information Revelation and Strategic Participation on Gig-Economy Platforms,” *Management Science*, vol. 70, no. 3, pp. 1729–1754, 2024.
- [2] H. Ma, W. Ma and M. Romero, “Sequential Offering in On-Demand Platforms: On the Optimality of Greedy Ranking,” *arXiv:2609.08001*, 2026.
- [3] K. Sekiya, S. Otani, Y. Komatsu, Y. Fujii, S. Ozeki and S. Noda, “Designing Recommendation Exposure and Favorite Lists: A Field Experiment in a Spot-Work Platform,” *arXiv:2606.17397*, 2026.
- [4] S. Huber and C. Rust, “Calculate Travel Time and Distance with OpenStreetMap Data Using the Open Source Routing Machine (OSRM),” *The Stata Journal*, vol. 16, no. 2, pp. 416–423, 2016.
- [5] R. Smith, “An Overview of the Tesseract OCR Engine,” in *Proc. Ninth International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.
- [6] C. J. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” in *Proc. Eighth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2014, pp. 216–225.

- [7] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," in Proc. Eighth IEEE International Conference on Data Mining (ICDM), 2008, pp. 413–422.
- [8] M. Jones, J. Bradley and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force (IETF), 2015.
- [9] React Documentation, "React – The Library for Web and Native User Interfaces," documentation for component-based user interface development.
- [10] Tailwind CSS Documentation, "Tailwind CSS," documentation for utility-first styling, together with the Vite build tool documentation.
- [11] Spring Documentation, "Spring Boot and Spring Security Reference," documentation for REST API development and security.
- [12] FastAPI Documentation, "FastAPI," documentation for building Python APIs and microservices.
- [13] PostgreSQL Documentation, "PostgreSQL: The World's Most Advanced Open Source Relational Database," documentation for relational data storage.
- [14] Project OSRM and OpenStreetMap Documentation, "Open Source Routing Machine" and "OpenStreetMap," documentation for road routing and map data.
- [15] OWASP Foundation, "OWASP Application Security Verification Standard," security guidelines for authentication and web application development.



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