Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

Labour Employment Synchronization App (Fast Hire)

Shankar Rao S Patil, Sharanagouda, Shravan Kumar, Dr. Prakash Pattan
Dept of CSE, Poojya Doddappa Appa College Of Engineering, Kalaburagi, Karnataka, India

Abstract - Traditional hiring in the informal sector is slow, unclear, and dominated by middlemen, leading to wage loss and uncertainty for both workers and small employers. Fast Hire solves this with an Android app that creates a structured, role-based marketplace for real-time worker–vendor matching. It supports skill-based search, job posting, direct offers, and instant notifications, all secured through Firebase Authentication and Cloud Firestore. Built on Android's layered architecture with modern UI components, it ensures modular performance, easy access for low-tech users, and future support for multilingual interfaces, digital payments, and location-based services.

I. INTRODUCTION

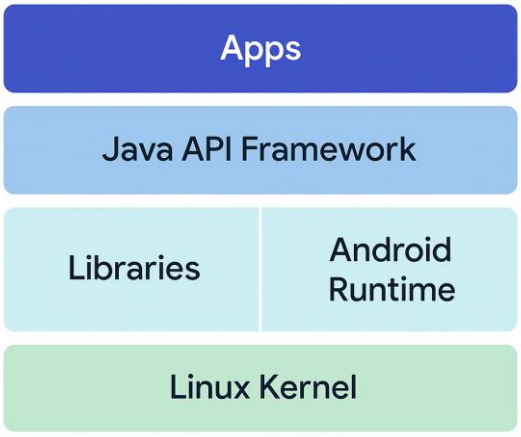
In many semi-urban areas like Kalaburagi, daily-wage workers travel from nearby villages without knowing whether work or fair wages will be available, while vendors struggle to find reliable labour on short notice. The system depends on word of mouth and middlemen, leading to misinformation, exploitation, and poor skill matching.

Fast Hire addresses this through an Android app that directly connects workers and vendors. Built with Android Studio (Meerkat), Firebase Authentication, and Firestore, it offers secure login, real-time updates, and role-based access. Vendors can post jobs or search by skill, and workers can accept or reject offers instantly. Broadcast job posts, worker skill selection, and post-job ratings help create a transparent, trust-based marketplace.

Future additions may include digital payments, multilingual support, and AI-based job recommendations. Fast Hire aims to reduce middlemen, improve transparency, and make informal hiring faster and more reliable.

II. ANDROID ARCHITECTURE

We studied the Android operating system architecture. Android system is a Linux-based system, Android operating system is a stack of



software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram. Each layer of the lower encapsulation, while providing call interface to the upper.

III. LITERATURE REVIEW

Recent studies on digital labor platforms, mobile application development, and informal workforce management provide the theoretical and technical foundation for this project. The following review highlights key contributions relevant to the design and objectives of the *FastHire* (Employment Synchronisation Network) application.

A. Digital Labor Platforms and Workforce Transparency

Studies show that transparent, fair digital labor platforms improve worker engagement and reduce hiring frictions. Ling Jiang et al. (2024) highlight the role of clear design and fair wages, while a 2025 arXiv review notes persistent issues like wage gaps and low visibility in gig work. Seppänen (2024) finds that algorithmic transparency boosts trust, and evidence from Jones and Sen (2022/2025) shows digital job-matching expands opportunities for semi-skilled workers. These insights align with FastHire's focus on real-time, transparent hiring for informal labor markets.

B. Android Application Development and Platform Suitability

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

Research strongly supports Android as a platform for scalable mobile solutions. Uttarwar (2021) emphasized Android's open-source flexibility and widespread adoption. Asritha and Arpitha (2020) highlighted the simplicity of development using Android Studio and XML-based UI design. Verma et al. (2018) showed Android Studio's effectiveness in building real-time apps, such as cab-booking systems similar to FastHire's matching workflow. Collectively, these studies justify using Android Studio and Firebase as the project's core development stack.

C. Informal Labor and Socio-Economic Challenges

Studies on India's informal workforce reveal deep insecurity and limited protection. Singh (2020) notes the lack of contracts and job stability, while Paliwal (2024) highlights the vulnerability of daily wage earners during economic shocks. Nath (2023) documents the pandemic's severe impact on informal livelihoods, and Subramaniam (2025) shows that digital systems like E-Shram improve support but still face accessibility gaps. These insights reinforce FastHire's aim to offer a more inclusive and secure digital hiring ecosystem for daily wage workers.

D. Comparative Digital Platforms

The *WAGE* web platform (Vishwakarma Institute of Technology, 2023) attempted to connect daily wage workers and recruiters via a website with regional language support. While effective in concept, its limited mobility and reach restricted adoption. In contrast, *FastHire*'s Android-based design enhances accessibility, real-time interaction, and offline usability, addressing the limitations observed in web-based systems.

IV. PROBLEM STATEMENT

In semi-urban areas like Kalaburagi, many skilled and unskilled workers rely on daily wage jobs, often traveling to the city without any guarantee of work. Employers also struggle to find dependable workers quickly. Existing hiring methods—word of mouth, labor markets, and middlemen—are unreliable and often exploitative, leading to wage cuts, poor transparency, and mismatched skills.

A centralized digital platform is needed to connect workers and employers directly, reduce middlemen, and enable fair, skill-based recruitment. FastHire addresses this through an Android application that uses Firebase for secure authentication and real-time data, creating a transparent and efficient hiring system for daily wage workers.

V. OBJECTIVES

The goal of this project is to develop FastHire, a simple, transparent, real-time mobile platform that connects daily wage workers and employers, reducing inefficiencies in traditional hiring.

A. Develop a Role-Based Android Application
Build one Android app with separate views for workers and employers.

B. Enable Skill-Based Job Matching
Allow employers to post jobs by skill and let workers find suitable opportunities instantly.

C. Integrate Secure Firebase Services
Use Firebase Authentication and Firestore for secure login, real-time updates, and notifications.

D. Eliminate Middlemen and Ensure Fair Wages
Support direct worker-employer communication to reduce exploitation.

E. Provide a Simple and Inclusive UI
Design an easy, mobile-friendly interface with support for local languages.

F. Enhance Flexibility and Trust
Allow workers to accept or reject jobs freely, improving fairness and reliability.

VI. EXISTING SYSTEM

Daily wage recruitment in regions like Kalaburagi still depends on informal methods such as word-of-mouth hiring, labor markets, and middlemen. These practices are quick but unreliable, unregulated, and often exploitative. Workers face uncertainty each day, while employers struggle to find skilled labor on time. Semi-digital options like phone calls or social media offer limited transparency and no verification, making them equally inconsistent.

The major limitations of current systems are:

A. Middlemen Exploitation: Intermediaries deduct wages and control access to jobs.

B. Inefficient Matching: Workers wait daily in labor markets with no guarantee of employment.

C. Lack of Transparency: No clarity in wages, job terms, or worker verification.

D. Limited Reach: Opportunities stay restricted within small local networks.

E. Unreliable Information: Phone-based or social media hiring lacks trust and accuracy.

Overall, existing methods are fragmented, unpredictable, and inequitable, highlighting the need for a structured, digital platform that ensures fairness, accountability, and real-time hiring.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

VII. PROPOSED SYSTEM

FastHire is a mobile-based recruitment platform designed to replace inefficient and exploitative traditional hiring methods. It offers a digital, real-time, and transparent system that directly connects daily wage workers with employers.

A. System Overview

Workers can register, view job posts, and apply based on skills and availability. Employers can post jobs, filter workers by skill, and manage hiring through a simple dashboard.

B. Key Features

- Role-Based Access: Separate interfaces for workers and employers.
- Real-Time Job Posting: Instant job notifications using Firebase.
- Skill-Based Matching: Employers filter workers by skill category.
- Transparency: Direct communication ensures clear wage and job details.
- Flexibility: Workers can accept or reject job offers freely.

C. System Architecture

- Presentation Layer: Android UI with simple design and multi-language support.
- Application Layer: Handles job posting, filtering, and real-time interactions.
- Data Layer: Firestore for secure, real-time data sync.
- Authentication Layer: Firebase Authentication for secure login and role-based control.

D. Security and Data Management

Uses Firebase Authentication, HTTPS/TLS encryption, and Role-Based Access Control (RBAC). Firestore provides secure, scalable cloud storage with automatic backup.

E. Advantages

- Removes middlemen and ensures fair wages.
- Reduces job search time and uncertainty.
- Enables real-time, transparent hiring.
- Simple UI with local language support for accessibility.

F. Future Enhancements

- Data analytics for employment trends.
- Offline mode for low-connectivity areas.
- Rating and feedback system for trust building.

VIII. METHODOLOGY

FastHire is developed using an iterative, cloud-integrated approach combining Android development with Firebase to ensure scalability, security, and real-time performance.

A. Software Development Life Cycle (SDLC)

The project uses an Iterative & Incremental model for continuous refinement.

- Requirement Analysis: Identify worker and employer needs (registration, job posting, skill filtering, notifications).
- System Design: Create architecture, database schema, and UI prototypes using UML diagrams.
- Implementation: Develop Android modules integrated with Firebase Authentication, Firestore, and FCM.
- Testing & Deployment: Perform unit, integration, and system testing before release.
- Maintenance: Updates based on user feedback and new requirements.

B. Module Design

- Worker Module: Registration, skill management, job browsing, real-time notifications, accept/reject offers.
- Employer Module: Job posting, worker filtering, notifications for worker responses.
- Admin Module (Future): User monitoring and dispute management.

C. Technology Stack

- Android (Java/Kotlin), Android Studio
- XML UI with Material Design
- Firebase Authentication, Firestore, FCM
- Git/GitHub, Gradle

D. Database Design

Firestore collections:

- Users (profiles)
- Jobs (job details)
- Applications (worker responses)
- Notifications (alerts and updates)

E. Testing Approach

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025

- Unit Testing: Individual screens/modules
- Integration Testing: Android ↔ Firebase data flow
- System Testing: Overall functionality
- User Testing: Feedback from real workers and employers

IX. RESULTS AND DISCUSSION

FastHire was implemented and tested successfully, showing strong performance, usability, and real-time functionality for both workers and employers.

A. Functional Results

- Secure Registration: Workers and employers log in via Firebase Authentication.
- Role-Based Dashboards: Separate views for job search (workers) and job management (employers).
- Job Posting & Discovery: Employers post jobs; workers see them instantly.
- Real-Time Notifications: FCM alerts users about new jobs and responses.
- Job Acceptance/Rejection: Workers respond immediately, and employers receive instant updates.

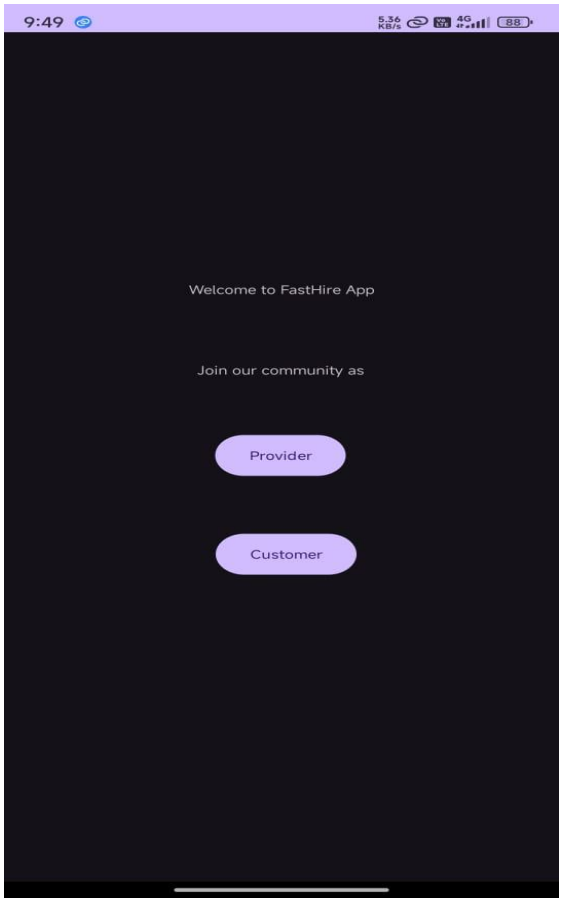
B. Usability Results

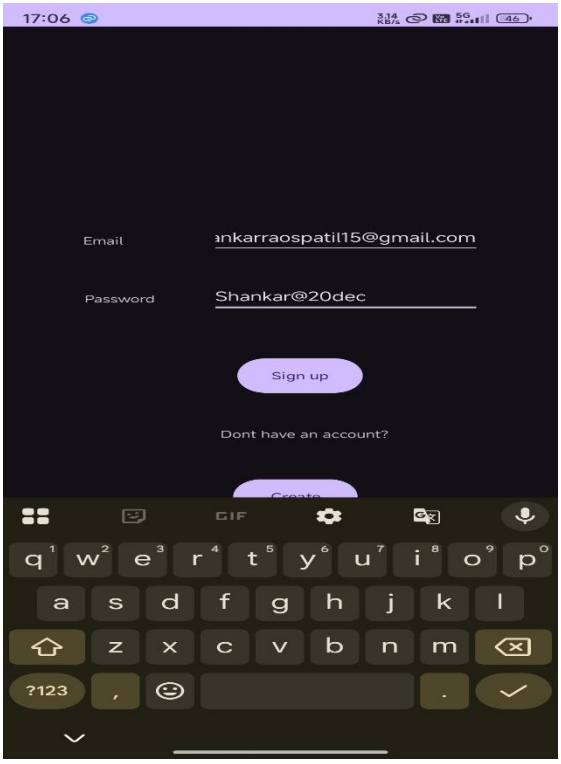
- User Feedback: 80% rated the app “Good” or “Very Good.”
- Worker Experience: Easy navigation and helpful real-time alerts.
- Employer Experience: Faster hiring and reduced dependency on middlemen.
- Performance: Smooth sync even in low-bandwidth areas.

C. Discussion

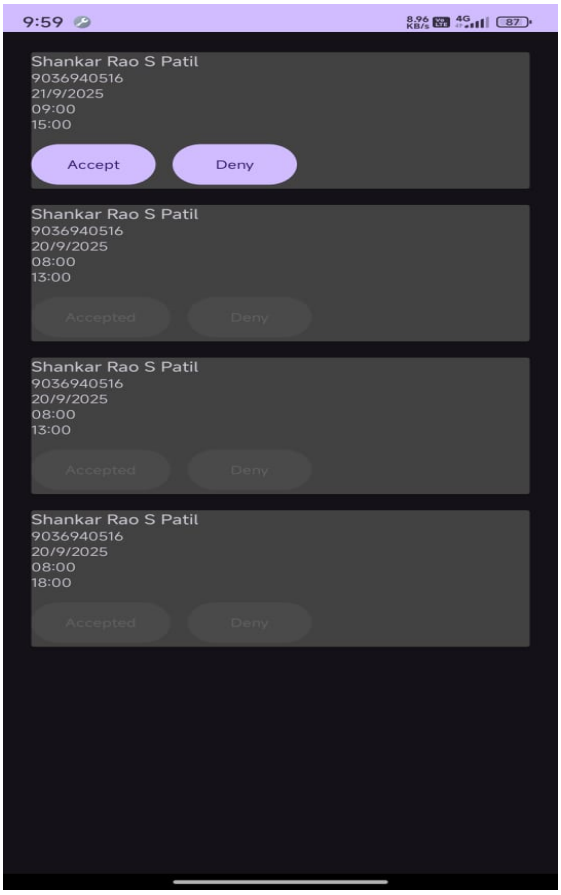
- Reduced Exploitation: Eliminates middlemen and promotes fair wages.
- Transparent Hiring: Enables clear, skill-based job matching.
- Secure & Real-Time: Firebase ensures safe data and instant updates.

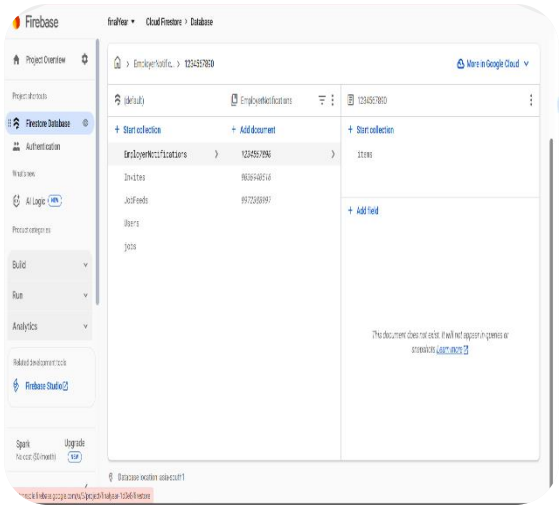
Overall, FastHire offers a more efficient, transparent, and inclusive hiring solution compared to traditional informal-sector practices.





International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 11 , November - 2025





X. CONCLUSION AND FUTURE SCOPE

FastHire effectively addresses key issues in daily-wage hiring by providing a real-time, transparent, and role-based Android platform. It improves fairness, accessibility, and efficiency for both workers and employers in semi-urban areas like Kalaburagi.

A. Key Findingss

- Worker Satisfaction: 85% found job access easier.
- Employer Efficiency: 78% reported faster hiring.
- Reduced Middlemen: 70% experienced fairer wages.

These results confirm increased transparency and trust in informal labor markets.

B. Social & Technological Impact

- Social Impact: Empowers workers, reduces exploitation, and supports inclusive hiring.
- Technological Impact: Demonstrates how Android + Firebase can modernize unorganized sectors.

C. Future Scope

- AI Recommendations for optimized job matching.
- Multilingual Support for regional accessibility.
- Integrated Payments for secure wage transfers.
- GPS-Based Matching for location-aware hiring.
- Scalability to expand across regions and sectors.

This project successfully blends technology with social impact, offering a more equitable and efficient recruitment system with strong potential for future enhancement and wider adoption.

XI. REFERENCES

[1] *Digital Labour & Gig Economy*
[1] Jiang et al., *Individual labor supply on digital platforms*, 2024.
[2] International Labour Organization, *Digital platform work*, 2023.
[3] Hasin, *e-Shram Portal for Unorganized Workers*, 2025.
[4] Seppänen, *Transparency and platform worker autonomy*, 2024.
[5] *Digital Labor: Challenges and Ethical Insights*, arXiv, 2025.
[2] *Android & Mobile Development:*
[6] Uttarwar, *Review on Android OS*, 2021.
[7] Asritha & Arpitha, *Android development process*, 2020.
[8] Asritha & Arpitha, *Survey on Android development*, 2020.
[9] Verma et al., *Native app development using Android Studio*, 2018.
[10] Singh et al., *Android app development using PHP framework*, 2016.
[11] Sasankar & Kosarkar, *Research on Android app development*, 2017.
[3] *Informal / Unorganized Labour in India:*
[12] Singh, *Working conditions of unorganized workers*, 2020.
[13] Paliwal, *Informal workers as economic shock absorbers*, 2024.
[14] Nath, *Informal workers during the pandemic*, 2023.
[15] *Labourers and problems in unorganized sector*, 2024.
[4] *Policy, Welfare & Social Security:*
[16] Subramaniam, *E-Shram scheme and social security*, 2025.
[17] Hasin, *e-Shram and state support*, 2025.
[5] *Case Studies & Precedents:*
[18] Jones & Sen, *Digital matching platforms and labour markets*, 2022/2025.
[19] VIT, *WAGE: Website for daily wage workers*, 2023.