

Labour Connect: A Digital Platform for Verified Skilled Labour Hiring

Nikhil Jagtap¹ Omkar Ambekar² Krishna Kardile³ Anushka Kale⁴ Ms. R.B. Gawale⁵

^{1,2,3,4,5}Department of Information Technology

^{1,2,3,4,5}MVP's Rajarshi Shahu Maharaj Polytechnic, Nashik, Maharashtra, India

Abstract — Skilled and semi-skilled workers in construction and allied trades are still hired mostly through informal channels such as labour chowks, middlemen and word of mouth. This causes unverified identities, uncertain skills, wage disputes, long searching time for contractors and no reliable work history for workers. This paper presents LabourConnect, a mobile-first Progressive Web Application (PWA) that connects verified skilled workers directly with contractors and employers. A worker registers by scanning a QR code displayed at a site or gate, verifies a mobile OTP, uploads documents and pays a small verification fee. After administrator approval the worker receives a QR-enabled Digital Skill Card. Contractors post job requirements and an AI-based weighted matching engine ranks the best workers using trade, experience, distance, expected salary, rating and availability. Contractors shortlist, unlock contact details through a secure payment and hire. Ratings and a trust score build reputation over time. The system is designed with React.js, FastAPI, Supabase (PostgreSQL), Firebase and Razorpay. This paper describes the problem, the proposed system, architecture, modules, database design, matching logic, security measures and the planned evaluation.

Keywords: Labour Hiring, Digital Skill Card, QR Code, Progressive Web App, AI Matching, Workforce Management

I. INTRODUCTION

Construction, plumbing, electrical work, welding, carpentry and similar trades depend on a large pool of skilled workers. In most Indian cities a contractor who needs a pipe fitter or a mason goes to a labour chowk or calls a known middleman. The worker, on the other side, waits at the roadside without knowing whether work will be available that day or what wage is fair. Neither side can easily check the other's identity, skill level or past record.

Digital platforms for white-collar jobs are common, but workers in the unorganised sector have little access to them because of language, literacy and device limitations. A usable solution must therefore be simple, quick to register, mobile-friendly and trustworthy for both sides.

LabourConnect is proposed to fill this gap. It is a digital marketplace in which every worker is verified, every skill is documented in a QR-enabled Digital Skill Card, every job requirement is posted in a structured form and every hire can be rated. The aim is to reduce hiring time, reduce fraud and give workers a portable proof of their skills and work history.

A. Objectives

- 1) To design a simple QR-based registration process for workers.
- 2) To verify identity and skill documents through an admin-controlled process.
- 3) To issue a Digital Skill Card with a scannable QR code.
- 4) To rank suitable workers for each job using a transparent weighted algorithm.

- 5) To provide secure payment, real-time notifications, ratings and work history.

II. LITERATURE REVIEW

Online job portals such as Naukri and LinkedIn mainly serve office and professional roles. Blue-collar platforms such as WorkIndia and Apna provide job listings for a wide range of low-skill jobs, and service marketplaces such as Urban Company connect customers with home-service professionals. However, most of these are either general listing boards or customer-facing service apps. They do not focus on contractor-to-worker hiring at construction sites with site-level QR onboarding, trade-wise skill cards and a ranked shortlist.

At the government level, the e-Shram portal of the Ministry of Labour and Employment creates a national database of unorganised workers and the Building and Other Construction Workers Act, 1996 provides for welfare of construction workers [6][7]. These initiatives register workers but do not offer direct job matching with contractors. Aadhaar-based identification is widely used for verification, while the Digital Personal Data Protection Act, 2023 requires careful handling of personal data [8], which is why LabourConnect stores only a masked Aadhaar number.

On the technical side, REST architecture [1], JSON Web Tokens for stateless sessions [2], and machine-learning libraries such as scikit-learn [3] provide proven building blocks. Modern web frameworks such as React [4] and FastAPI [5] allow fast development of responsive PWAs with a Python back end, which makes it easy to integrate a Python-based matching engine. The gaps found in the existing approaches are summarised in Table I.

Parameter	Traditional / Existing	LabourConnect
Hiring channel	Labour chowk, middlemen, word of mouth	Online PWA for workers and contractors
Verification	Mostly none or verbal	Document check by admin, OTP, masked Aadhaar
Skill proof	None	Digital Skill Card with QR code
Worker search	Manual and time consuming	AI-based ranked matching
Work history	Not recorded	Stored with ratings and trust score
Payment	Cash, disputes common	Razorpay verification and contact-unlock fees
Alerts	Phone calls only	Push, SMS and e-mail notifications

Table I: Comparison Of Existing and Proposed Approach

III. EXISTING SYSTEM

In the existing practice, a contractor visits a labour chowk or contacts a middleman, negotiates the wage on the spot and hires workers without any record. Workers have no proof of skill, cannot show previous work and depend on the middleman for opportunities. Limitations are: (1) no identity or skill verification, (2) no transparent wage information, (3) time lost in searching, (4) no complaint or rating mechanism, (5) no digital record of employment and (6) unsafe or fake hiring in some cases. General job portals do not solve these issues because they are not designed for trade-wise, location-wise and day-wise hiring of site workers.

IV. PROPOSED SYSTEM

LabourConnect is a web-based platform with three types of users: Worker (Labour), Contractor / Employer and System Administrator. The platform is delivered as a Progressive Web App so it works on low-cost smartphones and desktops without a separate installation from an app store, and can cache pages for weak networks.

A worker scans a QR code placed at a site or gate and reaches the registration form. After the mobile OTP is verified through Firebase Authentication, the worker fills in trade, experience, expected salary and city, and uploads identity and skill documents. A one-time verification fee of Rs. 99 is paid through Razorpay. The administrator reviews the documents and approves the profile, after which a Digital Skill Card with a QR code is generated. A contractor can scan this QR code to view the verified details of the worker.

Contractors post a job requirement with trade, number of workers, experience, daily rate and location. The AI matching engine calculates a score for each verified worker and notifies the best matches. Workers apply, the contractor shortlists and unlocks the contact through a secure payment, and after the work is completed, both sides can rate each other.

V. SYSTEM ARCHITECTURE

The architecture follows a layered, API-driven design (Fig. 1). The client layer is a React.js PWA with separate interfaces for workers, contractors and administrators. It talks to a FastAPI back end through REST APIs over HTTPS. The back-end handles authentication, job management, application handling, payment integration and the matching API. Specialised services are attached to the back end: Firebase Authentication for OTP login, Razorpay for payments, a Python machine-learning engine for matching, Google Maps API for distance calculation and Firebase Cloud Messaging for alerts. All persistent data is kept in a Supabase PostgreSQL database, and files such as documents, certificates and profile pictures are stored in Supabase Storage.

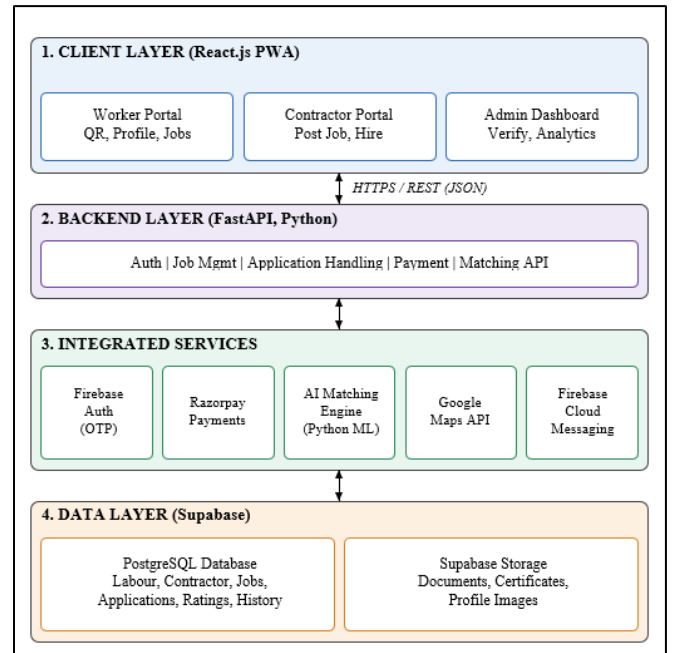


Fig. 1: System Architecture of LabourConnect

Fig. 2 shows the context-level data flow. The Worker, Contractor, Admin and Payment Gateway are external entities that exchange data with the LabourConnect system.

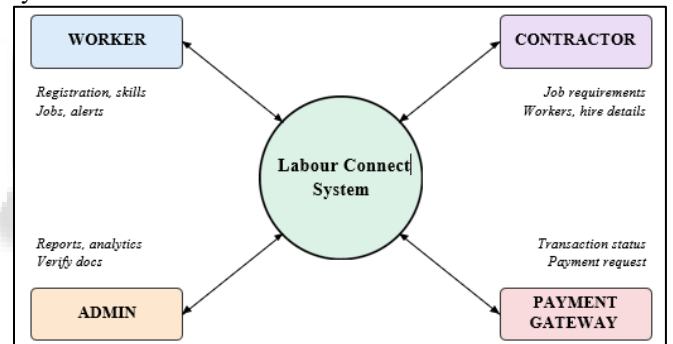


Fig. 2: Level-0 Data Flow (Context) Diagram

VI. WORKING FLOW

The complete working flow consists of eleven steps, shown in Fig. 3. Steps 1 to 7 cover worker onboarding, from QR scan to issue of the Digital Skill Card. Steps 8 to 11 cover job posting, matching, application and hiring. Each step is colour-coded by the actor responsible.

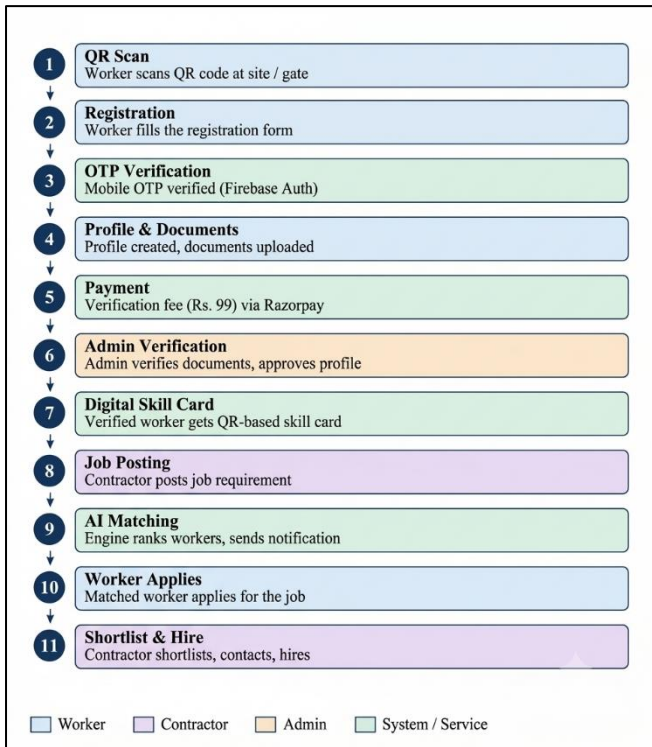


Fig. 3: Complete Working Flow of LabourConnect

VII. MODULES

The system is divided into the following modules, listed in Table II.

Module	Main Functions
Worker	Scan QR, register, OTP login, build profile, upload documents, pay verification fee, view Digital Skill Card, browse and apply for jobs, track applications, view ratings
Contractor	Sign up, company profile with GST number, post job requirement, search and filter workers, view ranked matches, shortlist, unlock contact, hire, rate worker
Administrator	Login, verify documents and approve or reject profiles, approve jobs, handle disputes and reports, view analytics dashboard
Matching Engine	Fetch relevant workers, compute weighted score, rank workers, send notification to top matches
Payment	Verification fee, contact-unlock fee, secure transactions and receipts through Razorpay
Notification	Job alerts, application updates and general notices using Firebase Cloud Messaging, SMS and e-mail

Table II: Modules And Their Functions

VIII. AI MATCHING LOGIC

When a job is posted, the engine fetches all verified workers of the same or a related trade and calculates a

matching score between 0 and 100 using a weighted sum of six factors (Fig. 4, Table III):

$$\text{Score} = 0.40 T + 0.20 E + 0.15 D + 0.10 S + 0.10 R + 0.05 A$$

where T is trade match, E is experience, D is distance, S is expected-salary fit, R is rating and A is availability, each normalised between 0 and 1. Workers are sorted by score and the top matches receive a notification.

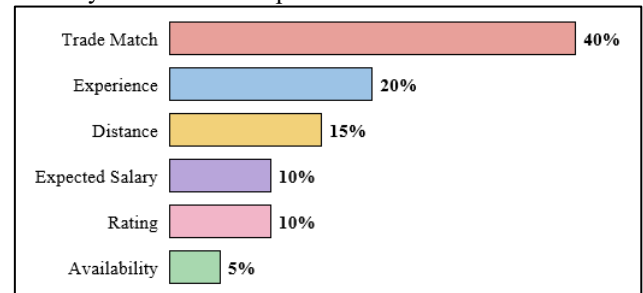


Fig. 4: Weights of Matching Factors

Factor	Weight	How it is measured
Trade (T)	40%	Same trade = 1, related trade = partial, otherwise 0
Experience (E)	20%	Worker years / required years, capped at 1
Distance (D)	15%	Distance from Google Maps; nearer gives higher value
Salary (S)	10%	Expected salary compared with offered daily rate
Rating (R)	10%	Average rating / 5, along with trust score
Availability (A)	5%	1 if available on job dates, else 0

Table III: Matching Factors

Illustrative example. For a Pipe Fitter job requiring 5 years of experience at Rs. 1000/day, a worker with the same trade (T = 1.0), 6 years of experience (E = 1.0), 5 km distance (D = 0.8), salary within budget (S = 1.0), rating 4.5/5 (R = 0.9) and available (A = 1) scores $0.40 + 0.20 + 0.12 + 0.10 + 0.09 + 0.05 = 0.96$, i.e. 96%. The values used in this example are only illustrative.

In the first version the score is a transparent rule-based weighted algorithm. As real hiring data accumulates (previous applications, hires and ratings), the engine can be extended with machine-learning ranking models trained with a library such as scikit-learn [3].

IX. DATABASE DESIGN

A relational database is used because the data is structured and has clear relationships between workers, contractors, jobs and applications. The main tables are shown in Table IV. Primary keys (PK) and foreign keys (FK) maintain integrity: one contractor can post many jobs, one job receives many applications, and each application is linked to one worker. The Work History and Ratings tables record completed jobs and feedback.

Table	Key Attributes
Labour	id (PK), name, mobile, masked Aadhaar, trade, experience, expected salary, city, documents, verification status, rating, trust score

Contractor	id (PK), company name, GST number, contact person, mobile, e-mail, address, verification status
Jobs	id (PK), contractor id (FK), trade, description, required workers, experience required, daily rate, location, status
Applications	id (PK), job id (FK), labour id (FK), status, applied at, shortlisted at, hired at
Work History	id (PK), job id (FK), labour id (FK), start date, end date, status
Ratings	id (PK), from user, to user, rating, review, created at

Table IV: Main Database Entities

X. TECHNOLOGY STACK

The technologies selected for implementation are listed in Table V. They are mostly open-source or have a free tier, which suits a student or start-up project, and they are scalable if the number of users grows.

Layer	Technology	Purpose
Frontend	React.js, Tailwind CSS	Responsive UI and PWA
Backend	FastAPI (Python)	REST API and business logic
Database	Supabase (PostgreSQL)	Structured data storage
Storage	Supabase Storage	Documents and images
Authentication	Firebase Authentication	OTP login, user management
Payment	Razorpay	Verification and unlock fees
AI / ML	Python, scikit-learn	Matching and recommendation
Maps	Google Maps API	Distance and location search
Notifications	Firebase Cloud Messaging	Push alerts
Analytics	Chart.js	Dashboards and reports
Deployment	Vercel (front end), Railway / Render (back end)	Hosting
Tools	GitHub, Postman, Swagger	Version control, API testing, docs

Table V: Technology Stack

XI. SECURITY AND PRIVACY

Because the system stores identity data, security is built into every layer:

- HTTPS/TLS for all communication and JWT-based sessions [2].
- Role-based access control for worker, contractor and administrator.
- Only a masked Aadhaar number is stored, in line with data-minimisation principles [8].
- Input validation, API rate limiting and secure file-upload checks.

- Private storage buckets and controlled access to uploaded documents.
- Contact details of a worker are visible to a contractor only after shortlisting and payment.
- Payments handled entirely by the Razorpay gateway; card data is never stored.
- Audit logs of administrator actions.

XII. TESTING PLAN

The system will be tested module-wise using Postman and Swagger for API testing and manual testing for user interfaces. Representative test cases are given in Table VI. The results of the pilot deployment will be reported in the extended version of this work.

ID	Test Case	Expected Result
T1	Worker scans QR and opens registration	Registration form loads on mobile
T2	Enter wrong OTP	Access denied, error shown
T3	Upload unsupported file type	Upload rejected
T4	Admin approves profile	Digital Skill Card with QR generated
T5	Post job and run matching	Workers returned in descending score order
T6	Contractor views contact before payment	Contact hidden
T7	Successful Razorpay payment	Contact unlocked, transaction saved
T8	Worker opens another user's data by URL	Request blocked by access control

Table VI: sample test cases

XIII. ADVANTAGES

- 1) Faster hiring: contractors find suitable workers in minutes instead of hours.
- 2) Trust: verified profiles, Digital Skill Card and ratings reduce fraud.
- 3) Easy onboarding through QR code and a mobile-first PWA.
- 4) Fair opportunity: workers get visibility based on skill, not on middlemen.
- 5) Permanent work history that the worker can carry to any employer.
- 6) Scalable and modular architecture that can add new trades and cities.

XIV. LIMITATIONS AND FUTURE SCOPE

The current design depends on a smartphone with internet access and on manual document verification by the administrator, which can become a bottleneck. Future work includes multi-language and voice-based interfaces for low-literacy users, automatic document verification using OCR and DigiLocker integration, a native Android application, offline-first operation, wage and attendance tracking, insurance and welfare-scheme integration, and machine-learning ranking trained on real hiring data.

XV. CONCLUSION

This paper presented LabourConnect, a digital platform for verified skilled labour hiring. By combining QR-based registration, OTP authentication, admin verification, a Digital Skill Card, AI-based weighted matching, secure payment and ratings, the system replaces the informal and unverified hiring process with a transparent and trustworthy one. The layered architecture using React.js, FastAPI, Supabase and Firebase is scalable and affordable. The platform can improve earning opportunities for workers, save time for contractors and create a reliable digital record of the skilled workforce.

REFERENCES

- [1] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, University of California, Irvine, 2000.
- [2] M. Jones, J. Bradley and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, Internet Engineering Task Force, May 2015.
- [3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
- [4] React Documentation, Meta Open Source. [Online]. Available: <https://react.dev>
- [5] FastAPI Documentation, S. Ramirez. [Online]. Available: <https://fastapi.tiangolo.com>
- [6] Government of India, "The Building and Other Construction Workers (Regulation of Employment and Conditions of Service) Act, 1996."
- [7] Ministry of Labour and Employment, Government of India, "e-Shram Portal." [Online]. Available: <https://eshram.gov.in>
- [8] Government of India, "The Digital Personal Data Protection Act, 2023."