Development of a Machine Learning-Integrated MERN Job Portal for Sustainable Informal Employment

Saras Jadhav[1], Shashikant V. Athawale[2], Neha Kolambe[3], Gargee Joshi[4], Riya Cornel[5]

[1]AISSMS COE, Savitribai Phule University, Pune, India

[2]Associate Professor, Department of Computer Engineering, AISSMS COE, Savitribai Phule Pune University

[3]AISSMS COE, Savitribai Phule University, Pune, India

[4]AISSMS COE, Savitribai Phule University, Pune, India

[5]AISSMS COE, Savitribai Phule University, Pune, India

Abstract: India's informal sector employs over 450 million blue- and pink-collar workers, yet most remain disconnected from digital employment platforms designed primarily for white-collar jobs. This research presents HireEasy, an intelligent job portal built using MERN stack (MongoDB, Express.js, React.js, Node.js) to facilitate long-term, skill-based employment for informal workers. The system integrates a machine learning–powered recommendation engine that matches workers to clients based on skills, location, and availability, while also supporting manual filtering for user-driven control. To ensure inclusivity, HireEasy introduces offline worker onboarding through admin-assisted registration for individuals lacking smartphones or internet access. Key modules include role-based authentication, OTP-based login, job posting and booking workflows, and an admin dashboard for verification and analytics. Experimental evaluation demonstrates improved worker–client match accuracy and streamlined job discovery. This platform contributes toward equitable, technology-driven employment access for India's underserved workforce.

Keywords: Job Portal, MERN stack, machine learning, informal workforce, recommendation system, inclusive employment

I. Introduction

India's workforce is predominantly informal, with over 450 million individuals employed in blue- and pink-collar occupations such as domestic help, driving, construction, and maintenance services. Despite their significant contribution to the economy, these workers remain largely excluded from organized digital employment platforms. Existing job portals such as Naukri.com and LinkedIn primarily cater to white-collar professionals, while gig-based applications like Urban Company emphasize short-term service engagements rather than sustainable, long-term employment. As a result, the informal labor sector lacks digital representation, leading to

inefficient job discovery, limited trust, and a fragmented hiring process.

To address this gap, HireEasy is proposed as an inclusive and intelligent job portal designed specifically for the informal workforce. The system utilizes the MERN stack (MongoDB, Express.js, React.js, Node.js) for scalability, responsiveness, and cross-platform accessibility. It integrates a machine learning–based recommendation engine that matches workers to clients based on their skills, location, and availability, improving both hiring efficiency and job visibility. Moreover, to ensure inclusivity, HireEasy incorporates offline worker onboarding through admin-assisted registration, thereby bridging the digital divide for users without smartphone access. This research aims to design, implement, and evaluate HireEasy's architecture and recommendation framework to facilitate equitable access to employment.

II. Literature Review

Recent studies have demonstrated substantial progress in the field of intelligent job recommendation systems, with a growing emphasis on personalization, fairness, and hybrid machine learning techniques. Duygu Çelik Ertuğrul and Selin Bitirim (2025) conducted a systematic literature review and taxonomy of 57 core papers published between 2010 and 2023, highlighting the evolution of Job Recommender Systems (JRS). Their work categorized methodologies such as content-based, collaborative, and hybrid filtering, emphasizing that while JRS research is expanding, there remains limited exploration beyond candidate–job matching, especially in human resource optimization and fairness-aware systems.

Yoosof Mashayekhi et al. (2024) addressed congestion and exposure fairness in job recommendation by proposing ReCon, a multi-objective optimization model that integrates collaborative and content-based filtering with optimal transport theory. Using large-scale datasets such as VDAB and CareerBuilder, their study demonstrated that ReCon significantly improves recommendation fairness and scalability, though it introduces complexity in hyperparameter tuning and computational overhead.

Hao Wang and Wenchuan Yang (2023) introduced an Improved Heterogeneous Graph Convolutional Network (IHGCN) for early job seekers using LinkedIn data. The model captures multi-relational links between resumes, skills, and job postings, enabling better contextual recommendations through meta-path-based learning. While IHGCN achieved over 10% performance improvement compared to traditional GCNs, it remains resource-intensive and less interpretable for real-time systems.

Collectively, these studies reveal that while advanced job recommender systems show strong performance in data-rich, structured domains, they are less suited for informal job markets characterized by sparse data and limited textual inputs. This gap underscores the need for a lightweight, inclusive, and hybrid approach such as the proposed HireEasy model, which integrates content-based filtering, collaborative filtering, and LightGBM ranking to deliver context-aware

recommendations for the informal workforce.

III. Proposed System

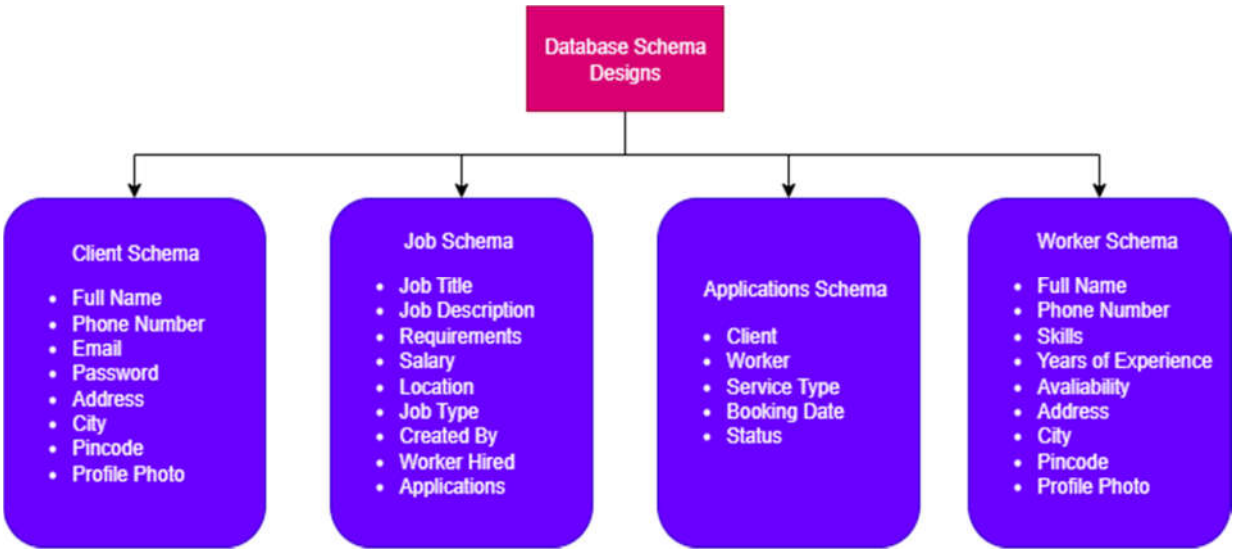
A. System Overview

The proposed system HireEasy, is an inclusive job portal designed to bridge the digital employment gap for India's blue and pink-collar workforce. The platform enables clients such as households, local businesses to hire skilled workers for long-time roles. Unlike conventional job portals that usually focus on white-collar and short-term gigs, HireEasy emphasizes on sustainable employment and inclusivity by supporting both online and offline worker onboarding.

HireEasy is developed using MERN stack (MongoDB, Express.js, React.js, Node.js) to ensure scalability, modularity and responsiveness across devices. The system incorporates a machine learning powered recommendation engine that suggests sustainable workers based on their skills, location and availability, while also allowing clients to manually filter candidates according to specific requirements.

The platform consists of three major roles: Clients, who post or requests services. Workers, who register skills and availability and the admin, who manages analytics and offline registrations. This enable being transparent and efficient ecosystems that connects job seekers and employers.

B. Database Design



(Fig. 3.1 Database Schema Design)

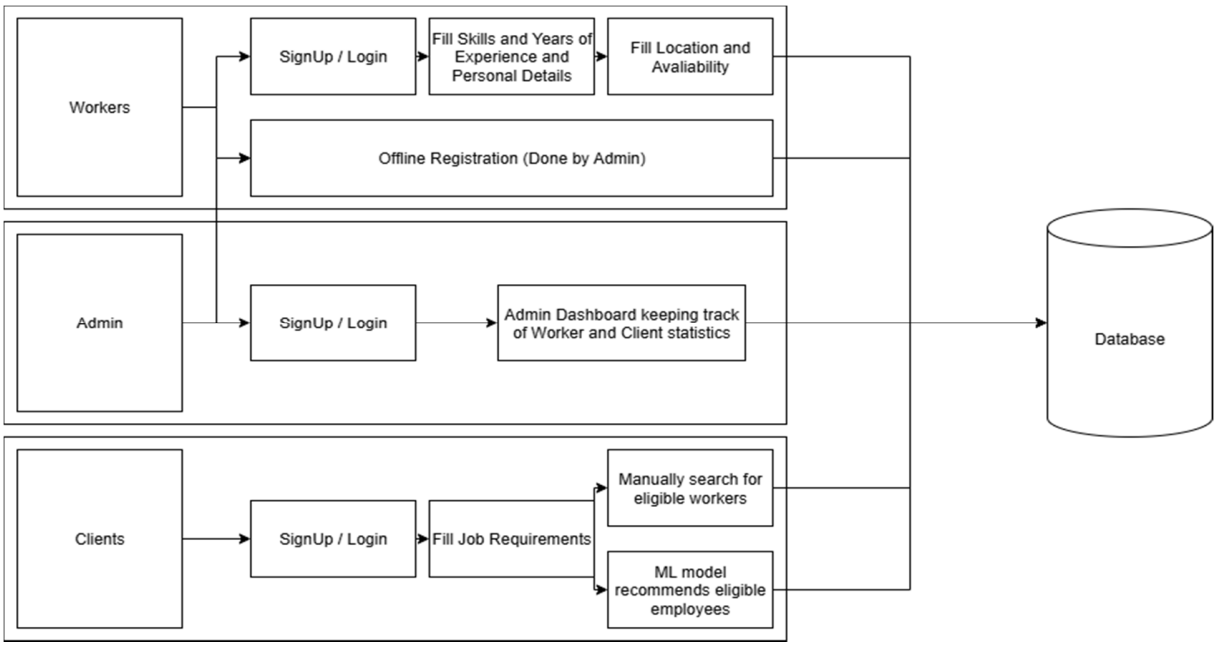
The database is implemented using MongoDB, a NoSQL document-oriented database that provides flexibility in handling heterogeneous and dynamic datasets. The design follows a modular and relational approach through referencing, ensuring efficient storage, scalability, and

fast retrieval of data across the system's modules. The database comprises four key schemas: Client, Worker, Job, and Application, each serving a specific functional purpose within the system.

1. Client Schema
   The client schema stores essential details of users seeking services. It includes attributes such as full name, email, phone number, city, pincode, address, password, and profile photo. These attributes enable client authentication and personalized service discovery. This schema ensures that clients can securely log in, manage their profiles, and post job requests.
2. Worker Schema
   The worker schema captures the personal and professional details of informal workers. Unlike the client schema, it prioritizes phone number as the primary contact field instead of email to accommodate workers with limited digital literacy. Other attributes include full name, address, skills, years of experience, availability, and profile photo. This schema facilitates efficient worker discovery and recommendation by the ML module.
3. Job Schema
   The job schema manages information about the job postings created by clients. It includes fields such as title, description, requirements (services requested), salary, location, and job type.
   The schema maintains multiple references to ensure relational consistency:
   createdBy: References the Client schema to identify which client posted the job.
   workerHired: References the Worker schema to record which worker has been hired for the job.
   applications: References the Application schema to link all application submitted for the job's post. This relational mapping enables seamless tracking of each job's lifecycle from posting to hiring.
4. Application Schema
   The application schema serves as the bridge between clients and workers by capturing the details of every service request. It includes:
   Client: Reference to the Client schema (who requested the service).
   Worker: Reference to the Worker schema (who applied or was hired).
   serviceType: Specifies the type of service requested, e.g., cook, maid, driver, etc.
   bookingDate: Indicates the scheduled date for the service.
   Status: An enumerated field with values such as Pending, Confirmed, Completed and Cancelled representing the state of the application. This schema ensures that each service between clients and workers is properly logged and traceable.

The integration of these four schemas ensures data integrity and streamline the interactions between the different user roles.

C. System Workflow



(Fig. 3.2 System Workflow of HireEasy)

The workflow outlines the interaction between clients, workers, and the admin through a structured data flow that connects the frontend applications, backend APIs, and the central database. As illustrated in diagram, the workflow represents both online and offline registration processes, job posting mechanisms, and the functioning of the planned machine learning module.

1. Worker Workflow
   Workers begin by signing up or logging into the system through the Worker React App. They enter personal details such as name, contact information, and address, along with professional details including skills, years of experience, and availability status. For workers lacking access to smartphones or digital literacy, offline registration is handled by the Admin, who inputs their data manually into the system. All worker-related data is stored securely in the MongoDB database for subsequent retrieval and matching operations.
2. Admin Workflow
   The admin functions as the supervisory entity within HireEasy. After logging into the Admin React App, the admin can verify newly registered workers, validate client postings, and monitor user statistics through a dedicated dashboard. The admin module also maintains control over the data consistency between workers, clients, and ongoing job requests. In cases of offline registrations, the admin inputs worker data directly into the system database, ensuring inclusivity for non-digital users.

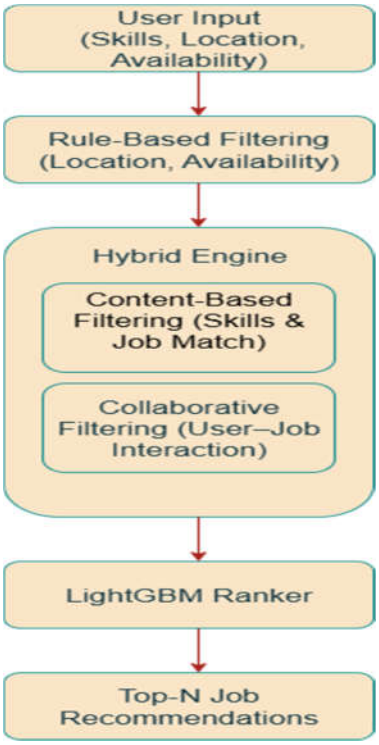
3. Client Workflow
   Clients register and log in through the Client React App, where they specify job requirements, such as service category, job location, and salary range. Once a job request is submitted, the system provides two possible approaches:
   Manual Search: Clients can browse and filter eligible workers based on predefined attributes such as skill set, availability, and proximity.
   ML-Based Recommendation: The machine learning model analyzes the stored worker and job data to automatically recommend suitable workers based on similarity scoring between the client's requirements and worker profiles.

All workflows converge at the MongoDB database which serves as the central repository for the client, worker and job-related data. This integrated structure allows efficient querying, reliable data synchronization, and smooth operation between all user modules.

IV. Proposed Machine Learning Model

Several existing studies have explored machine learning and hybrid approaches for enhancing job recommendation accuracy personalization.

While the earlier studies do demonstrate strong performance in structured, data-rich environments, they are less suited to informal labor markets where user data is limited and less textual. To address this gap, HireEasy proposes a Hybrid Recommendation Model that combines Content-Based Filtering (CBF), Collaborative Filtering (CF), and a Light Gradient Boosting Machine (LightGBM) Ranking Model.
In the proposed approach, CBF analyzes worker attributes such as skills, location, and availability to identify candidates similar to the client's requirements. Collaborative Filtering incorporates implicit feedback (e.g., booking frequency, worker ratings, prior selections) to refine matching patterns among users with similar preferences. The LightGBM ranking layer serves as the final optimization stage, where it learns to rank worker–job pairs using engineered features from both CBF and CF outputs. This gradient boosting approach enables the model to handle sparse data efficiently, reduce overfitting, and deliver ranked recommendations based on a learned relevance score.



(Fig. 4.1 Proposed hybrid recommendation model)

Model Overview:

The hybrid model consists of three main stages:

1. Pre-filtering:
   Filters out irrelevant jobs based on worker's skills, availability and location preference, reducing the candidate job pool and computation time.
2. Hybrid Recommendation Engine:
   Combines:
   a. Content-Based Filtering (CBF): Matches workers and jobs based on the similarity between skill vectors.
      Cosine similarity is used to measure the overlap between worker skill sets and job requirements.
   b. Collaborative Filtering (CF): Learns from worker–job interaction history to identify jobs preferred by similar workers using matrix factorization.
3. Ranking Layer (LightGBM):
   The final ranked list of jobs is generated using LightGBM, which learns an optimal ranking function based on multiple features — such as skill match percentage, geographical distance, and worker availability score.

The hybrid architecture ensures that recommendations remain accurate, adaptive, and explainable, even in environments with limited data. As more user interactions accumulate, the

LightGBM ranking layer will dynamically update feature importance, continuously improve match precision while maintaining low computational overhead.

V. Expected Outcomes

HireEasy aims to create a scalable and inclusive digital platform that connects clients with blue- and pink-collar workers through a unified, data-driven system. By integrating a MERN-based architecture with a machine learning recommendation engine, it enables accurate and personalized worker–client matching based on skills, location, and availability. The platform supports both online and offline onboarding, ensuring accessibility for non-digital users, while offering secure, role-based interfaces for clients, workers, and admin. With its focus on bridging the employment gap and promoting long-term opportunities, HireEasy seeks to empower India's informal workforce, reduce digital inequality, and foster sustainable socio-economic inclusion through a reliable and scalable digital ecosystem.

VI. Conclusion

HireEasy presents an innovative approach to addressing the employment challenges faced by India's blue- and pink-collar workforce. By leveraging the MERN stack for system development and integrating a machine learning–based recommendation engine, the platform provides a scalable and intelligent framework for connecting informal workers with potential employers. The system's modular design—comprising client, worker, and admin interfaces—ensures seamless communication, secure data handling, and efficient job management.

Unlike existing job portals that primarily target white-collar or short-term gig workers, HireEasy emphasizes inclusivity and sustainability by enabling offline worker onboarding and long-term job connections. The architecture's flexibility supports continuous expansion through future modules such as multilingual support, and real-time analytics. HireEasy aims to bridge the gap between informal labor and digital employment ecosystems, promoting equitable access and economic empowerment for millions of underserved workers in India.

VII. Acknowledgement

We would like to express our sincere gratitude to our project guide, Dr. Shashikant V. Athawale, for his valuable guidance, constant encouragement, and insightful feedback throughout the completion of this research work.

We are also thankful to the Department of Computer Engineering, AISSMS College of Engineering, for providing the necessary facilities and support for carrying out this research.

VIII. References

[1] A. Shaikym, Z. Zhalgassova, and U. Sadyk, "Design and evaluation of a personalized job recommendation system for computer science students using hybrid approach," *in Proc. IEEE Int. Conf. Electronics, Computer and Communication (ICECCO)*, 2023, pp. 1–7. doi: 10.1109/ICECCO58239.2023.10147147.

[2] D. Sahoo, S. C. Nayak, N. R. Paul, S. Saha, P. R. Patra, and P. K. Kuanr, "Designed framework for advanced intelligent job recommendation system," *in Proc. 21st OITS Int. Conf. on Information Technology (OCIT)*, 2023, pp. 1–6. doi: 10.1109/OCIT59427.2023.10430867.

[3] H. Wang, W. Yang, J. Li, J. Ou, Y. Song, and Y. Chen, "An improved heterogeneous graph convolutional network for job recommendation," *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107147, 2023. doi: 10.1016/j.engappai.2023.107147.

[4] Y. Mashayekhi, B. Kang, J. Lijffijt, and T. De Bie, "Scalable job recommendation with lower congestion using optimal transport," *IEEE Access*, vol. 12, pp. 1–12, 2024.

[5] D. Ç. Ertuğrul and S. Bitirim, "Job recommender systems: A systematic literature review, applications, open issues, and challenges," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 1–34, 2023.

[6] L. O. Badru, V. Vasudevan, G. I. Lingam, and M. G. M. Khan, "MERN stack web-based education management information systems for Pacific Island countries," *SN Computer Science*, vol. 4, no. 1, p. 70, 2023.

[7] S. C. Nayak, N. R. Paul *et al.*, "Designed framework for advanced intelligent job recommendation system," in *Proc. 2023 IEEE OCIT*, Dec. 2023, doi: 10.1109/OCIT59427.2023.10430867.

[8] Q. Wang, "College employment recommendation based on improved K-means clustering and SimRank algorithm in college employment management," *IEEE Access*, vol. 12, pp. 1–12, 2024.