

SWIPEHIRE: A DUAL-MODE AI-DRIVEN JOB MATCHING PLATFORM FOR CANDIDATES AND RECRUITERS

Humera Siddiqua¹, Dr. Umar Farooq²

¹PG Scholar, Department of AI & DS, Shadan Women's College of Engineering and Technology, Hyderabad,
humerasiddiqua2k2@gmail.com

²Professor, Department of CSE, Shadan Women's College of Engineering and Technology
umarfarooq@gmail.com

ABSTRACT

The rapid growth of the digital job market has made traditional hiring platforms increasingly inefficient, with candidates spending excessive time browsing irrelevant listings and recruiters manually screening large volumes of unsuitable applications. In this work, we present SwipeHire, a dual-mode AI-supported hiring platform that reimagines job discovery and recruitment through a swipe-based interaction model inspired by modern consumer applications, combined with a transparent, rule-based skill-matching algorithm. The system supports two distinct user roles — candidates and recruiters — each guided through a tailored onboarding and profile-setup flow. Candidates browse job postings as swipeable cards annotated with an AI-generated match percentage derived from skill-overlap scoring, while recruiters review AI-ranked candidate profiles based on a composite fit score incorporating skill alignment, resume completeness, and relevant experience. A mutual right-swipe from both parties triggers an automatic match, unlocking a real-time chat channel built on WebSocket technology that supports text messaging, file sharing, and recruiter-initiated interview requests. The platform further provides candidates with a growth analytics dashboard visualizing application trends and in-demand skills, and recruiters with a recruiting analytics dashboard tracking shortlist rates and job performance. Administrative controls for user management and company verification, together with JWT-based authentication, password hashing, and rate-limiting, ensure the platform is both usable and secure. Implemented using the MERN stack (MongoDB, Express.js, React.js, Node.js) with Socket.IO for real-time communication, SwipeHire demonstrates that an explainable, algorithm-driven matching approach can deliver a fast, engaging, and trustworthy hiring experience for both job seekers and employers.

1. INTRODUCTION

The process of finding a job or hiring talent has traditionally relied on static listings, keyword-based search, and lengthy application forms that offer little immediate feedback to either party. Candidates are often left uncertain about how well their profile aligns with a given role, while recruiters are burdened with manually reviewing large numbers of applications, many of which are poor fits. This inefficiency is compounded by the absence of real-time communication channels between candidates and recruiters until relatively late in the hiring pipeline, delaying decisions and increasing drop-off rates. Meanwhile, consumer-facing applications in other domains have popularized swipe-based interaction models that reduce decision fatigue and increase engagement by presenting information in small, digestible units with clear binary actions. This project applies that same interaction paradigm to the hiring domain. SwipeHire is designed as a dual-mode platform: candidates swipe through job postings that display an AI-generated match percentage computed from the overlap between their listed skills and a job's required skills, while recruiters swipe through a queue of applicants ranked by a composite fit score. Crucially, the matching logic used throughout the platform is a deterministic, rule-based algorithm rather than an opaque machine-

learning model, making it fully explainable — a property that is particularly valuable in an academic and early-stage product context, where transparency and correctness matter as much as sophistication. When both a candidate and a recruiter express mutual interest, the platform automatically unlocks a real-time chat interface, allowing the two parties to communicate directly, share documents such as resumes, and schedule interviews without leaving the platform. The result is a hiring platform that is fast to use, transparent in its recommendations, and capable of taking a match all the way from first impression to interview scheduling within a single system.

2. EXISTING SYSTEM OF PROJECT

- Conventional job portals such as static listing boards primarily rely on keyword search and manual filtering, requiring candidates to read through lengthy job descriptions to judge their own suitability, and requiring recruiters to manually sift through résumés with no automated ranking of applicant quality.
- Existing platforms typically separate the application and communication stages: a candidate applies, and communication (if any) begins only after a recruiter

manually reviews the résumé, often through external email, introducing delay and reducing candidate engagement.

- Where recommendation or "matching" features do exist on established platforms, the underlying logic is usually a closed, opaque system whose scoring criteria are not visible to the user, making it difficult for a candidate to understand why a job was or was not suggested, or for a recruiter to trust a ranking without visibility into its basis.
- Resume screening on many existing systems is still substantially manual, which does not scale well for recruiters handling large applicant volumes and increases the time-to-hire for open positions.

EXISTING SYSTEM DISADVANTAGES:

- High cognitive load and time cost for candidates browsing large volumes of unfiltered listings
- No transparent or quantified indication of fit between a candidate and a job before applying
- Delayed, often external, communication channel between candidates and recruiters after application
- Manual, non-prioritized review queues for recruiters, with no built-in applicant ranking
- Lack of integrated post-match workflow (chat, file sharing, interview scheduling) within a single platform

3. RELATED WORK

Research and industry practice in online recruitment systems have explored several directions relevant to this project. Traditional applicant tracking systems (ATS) rely predominantly on keyword and Boolean search over résumé text, which, while computationally inexpensive, produces high false-negative rates when candidates describe equivalent skills using different terminology. More recent commercial platforms have introduced recommendation-style feeds that surface jobs based on a candidate's stated profile, but the scoring mechanisms behind these recommendations are rarely disclosed, limiting user trust and making it difficult for candidates to actively improve their match likelihood. Separately, the "swipe" interaction pattern popularized by consumer social and dating applications has been studied for its ability to reduce decision fatigue by presenting one item at a time with a simple binary choice, and several regional startups have experimented with applying this pattern to blue-collar and gig-economy hiring, typically pairing it with basic keyword matching rather than a documented scoring function. On the technical side, the MERN stack (MongoDB, Express.js, React.js, Node.js)

has become a standard architecture for building full-stack, real-time web applications due to its unified JavaScript ecosystem, and Socket.IO is widely used to add WebSocket-based real-time features — such as chat and live notifications — on top of a conventional REST API without requiring a separate messaging infrastructure. Building on these strands, SwipeHire combines the engagement benefits of swipe-based interaction with a fully transparent, rule-based skill-matching algorithm, and extends the interaction beyond matching into an integrated real-time communication and scheduling workflow, an area where most existing swipe-based hiring prototypes stop short.

4. PROPOSED SYSTEM

SwipeHire proposes an integrated, dual-mode hiring platform in which both candidates and recruiters interact with the same core matching engine from opposite ends. On the candidate side, each job posting is rendered as a swipeable card annotated with a match percentage calculated by comparing the candidate's declared skill set against the job's required skills using a normalized set-overlap algorithm; a right swipe records an application, a left swipe skips the listing, and a dedicated "save for later" action bookmarks a job without committing to either outcome. On the recruiter side, applicants for a given job are ranked using a composite fit score that combines the same skill-match percentage with secondary signals — résumé completeness and relevant work experience — allowing recruiters to review the strongest candidates first rather than in arrival order. When a candidate and a recruiter independently swipe right on each other for the same job, the system automatically creates a match record and unlocks a dedicated chat channel between the two parties, implemented using Socket.IO for real-time bidirectional communication. Within this channel, users can exchange text messages and share files such as résumés, and recruiters can issue a formal interview request that is surfaced to the candidate as a persistent banner within the conversation. Beyond the core matching loop, the system layers in supporting infrastructure: a multi-step profile wizard for candidates and a streamlined company-profile flow for recruiters; role-based JWT authentication with hashed password storage; real-time, Socket.IO-delivered notifications for matches, messages, and interview requests; analytics dashboards for both roles built on aggregated application, match, and shortlist data; a location, job-type, and work-mode filter layer on the candidate feed; and an administrative panel through which platform operators can verify recruiter companies, moderate job postings, and manage user accounts. Collectively, these components form a closed loop from

first discovery through to interview scheduling within a single, transparent system.

ADVANTAGES OF THE PROPOSED SYSTEM

- Transparent, Explainable Matching Algorithm
- Reduced Decision Fatigue Through Swipe-Based Interaction
- Integrated Real-Time Communication and Interview Scheduling
- Dual-Sided Analytics for Continuous Improvement
- Role-Based Security with Administrative Oversight

5. MODULES

1. Authentication & Onboarding Module

- Manages role-based account creation (candidate or recruiter) using email and password, with passwords secured through bcrypt hashing and session continuity maintained via JSON Web Tokens (JWT).
- Routes newly registered users through a role-specific profile setup wizard, ensuring both candidate and recruiter accounts are fully configured before entering the core matching experience.

2. Candidate Profile Module

- Captures structured candidate data including skills, education history, work experience, bio, location, expected salary, and preferred work mode (WFH/WFO/Hybrid).
- Handles résumé upload and storage via Multer-based file handling, with the résumé URL linked to the candidate's profile for recruiter review and made editable at any time after initial setup.

3. Recruiter & Company Profile Module

- Captures company-level information — name, website, industry, and description — associated with the recruiter's account, and initializes the company as unverified pending administrative review.
- Supports job posting with structured fields for title, description, responsibilities, required skills, location, job type, experience required, salary range, and work mode.

4. AI Matching Engine Module

- Computes a candidate-to-job match percentage using a normalized, case-insensitive set-overlap comparison between a candidate's skills and a job's required skills.
- Computes a recruiter-facing composite fit score by combining the match percentage with weighted secondary signals (résumé presence, relevant experience count), producing a ranked applicant queue per job.

5. Swipe Interface Module

- Presents candidates with a stacked, draggable card interface (built with Framer Motion) for browsing job postings, supporting right-swipe (apply), left-swipe (skip), and bookmark (save for later) actions, alongside location, job-type, and work-mode filters.
- Presents recruiters with an equivalent draggable card interface for reviewing AI-ranked applicants per job posting, supporting right-swipe (shortlist) and left-swipe (reject) actions.

6. Matching & Mutual-Match Detection Module

- Monitors swipe actions from both sides and automatically creates a match record the moment a candidate and a recruiter have both expressed interest in each other for the same job posting.
- Triggers real-time notification delivery to both parties upon match creation, and exposes a dedicated "My Matches" view listing all active matches with their computed match percentage.

7. Real-Time Chat & Interview Module

- Provides a Socket.IO-based real-time messaging channel scoped to each match, supporting persistent text message history, file attachment sharing (up to 10MB), and read-state tracking.
- Enables recruiters to issue a structured interview request directly within the chat interface, which is displayed to the candidate as a persistent status banner within the conversation.

8. Analytics & Growth Dashboard Module

- Generates candidate-facing analytics — total applications, total matches, average match percentage, application trend over the past six months, and top in-demand skills — visualized using Recharts.
- Generates recruiter-facing analytics — jobs posted, candidates shortlisted, shortlist rate, match trend over time, and per-job shortlist performance — supporting data-driven recruiting decisions.

9. Notification Module

- Delivers real-time, in-app notifications for key platform events (new match, new message, interview request) via a dedicated Socket.IO room per user, backed by a persistent notification history in the database.
- Provides an interactive notification bell with unread-count badge and mark-as-read functionality accessible from both candidate and recruiter dashboards.

10. Admin Panel Module

- Provides platform-wide statistics (total candidates, recruiters, jobs, verified/unverified companies) alongside management tables for

users, companies, and job postings, restricted to accounts flagged as administrators.

- Enables company verification toggling, job activation/deactivation, and user or job removal, supporting platform moderation and trust-building for the recruiter marketplace.

6. TECHNIQUES USED IN PROJECT

6.1 EXISTING TECHNIQUE:

Conventional hiring platforms depend primarily on keyword-based search and Boolean filtering over résumé and job description text, occasionally supplemented by opaque, closed-source recommendation scoring whose internal logic is not exposed to either candidates or recruiters. Communication between the two parties is typically deferred until after a manual résumé review, often occurring outside the platform entirely (through email or phone), which introduces delay and reduces the candidate's engagement with the process. Where ranking of applicants exists, it is frequently based on arrival order or basic keyword frequency rather than a documented, explainable scoring function, leaving recruiters to manually judge fit for each applicant individually. This combination of opaque matching and disconnected communication results in a slow, low-transparency hiring pipeline for both sides.

Drawbacks:

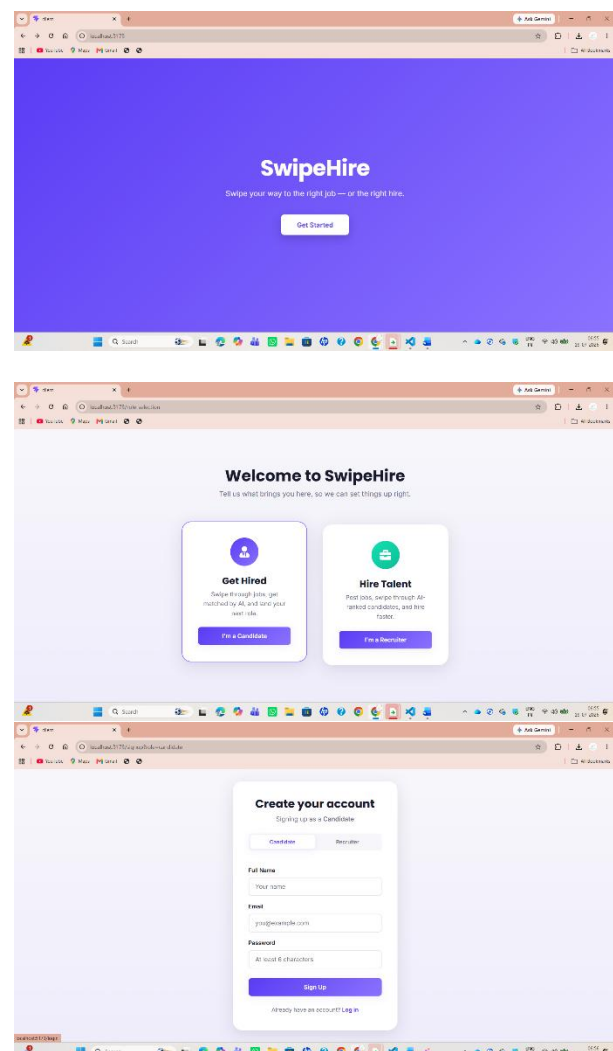
- No transparent or quantified fit score visible to the user before a decision is made
- Communication with the counterpart occurs late in the process and often outside the platform
- Applicant review order is not prioritized by actual suitability
- Limited or no integrated post-match workflow (scheduling, file sharing) within a single system.

6.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED

The proposed system uses a deterministic, rule-based skill-matching algorithm rather than a black-box machine learning model. Candidate skills and job-required skills are each normalized (trimmed and lower-cased) and compared as sets; the match percentage is computed as the proportion of a job's required skills that are present in the candidate's skill set, rounded to the nearest whole number. This approach is fully explainable — every match percentage shown to a user can be traced back to the exact set of overlapping and missing skills — which is a deliberate design choice favoring transparency and correctness over the complexity of a trained model,

particularly appropriate given the platform's current data scale. For recruiter-side ranking, this match percentage is extended into a composite fit score, weighting the skill match at 80% and adding small, capped bonuses for résumé presence and relevant experience count, giving recruiters a single sortable value that still privileges skill alignment as the primary signal. Real-time features (chat, notifications) are implemented using Socket.IO, which layers a WebSocket-based publish/subscribe messaging model over the existing Express.js REST API, allowing bidirectional, low-latency communication without replacing the underlying HTTP architecture. Authentication uses JSON Web Tokens (JWT) combined with bcrypt password hashing, and file uploads (résumés, chat attachments) are handled through Multer with server-side type and size validation.

7. RESULTS





International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X

www.ijdim.com

Original Research Paper

SwipeHire

Find your next role, and swipe at it.™

1 Tell us about yourself

2 Skills

3 Education & Experience

4 Resume

Tell us about yourself

bio

A short note about your professional background...

Location

city, country

Next

SwipeHire

Find your next role, and swipe at it.™

1 Tell us about yourself

2 What are your skills?

3 Education & Experience

4 Resume

What are your skills?

Type a skill and press Enter

e.g. React, Python, SQL...

Back

Next

SwipeHire

Find your next role, and swipe at it.™

1 Tell us about yourself

2 What are your skills?

3 Education & Experience

4 Resume

Education

Program (e.g. B.Tech, CSE)

Institution

Year

Experience (optional)

Job Title

Company

Duration

Back

Save & Continue

SwipeHire

Find your next role, and swipe at it.™

1 Tell us about yourself

2 What are your skills?

3 Education & Experience

4 Upload your resume

Upload your resume (max 5MB)

Choose file

Finish & Go to Dashboard

SwipeHire

Find your next role, and swipe at it.™

Filters

Save

Edit Profile

My Matches

Growth

Backend Developer

AcademyTech

Full-time

100%

A Backend developer builds and maintains the server-side infrastructure, databases, and APIs that power web applications. They write clean, scalable code to ensure that user-facing features operate seamlessly with the server. This role is responsible for ensuring the application performs efficiently under heavy traffic.

RESPONSIBILITIES

- Develop, test, and deploy server-side code.
- Manage databases and ensure data integrity.
- Collaborate with frontend developers to integrate user-facing features with the backend.

REQUIRED SKILLS

- Proficiency in a server-side language (e.g., Python, Java, Node.js).
- Experience with databases (e.g., MySQL, PostgreSQL).
- Understanding of web protocols and HTTP.

SwipeHire

Find your next role, and swipe at it.™

Filters

Save

Edit Profile

My Matches

Growth

Backend Developer

AcademyTech

Full-time

100%

A Backend developer builds and maintains the server-side infrastructure, databases, and APIs that power web applications. They write clean, scalable code to ensure that user-facing features operate seamlessly with the server. This role is responsible for ensuring the application performs efficiently under heavy traffic.

RESPONSIBILITIES

- Develop, test, and deploy server-side code.
- Manage databases and ensure data integrity.
- Collaborate with frontend developers to integrate user-facing features with the backend.

REQUIRED SKILLS

- Proficiency in a server-side language (e.g., Python, Java, Node.js).
- Experience with databases (e.g., MySQL, PostgreSQL).
- Understanding of web protocols and HTTP.

SwipeHire

Find your next role, and swipe at it.™

Filters

Save

Edit Profile

My Matches

Growth

Frontend Developer

AcademyTech

Full-time

100%

A Frontend developer builds the user-facing part of web and mobile applications. They are responsible for translating UI/UX designs into functional, responsive code, ensuring applications are easy to use and perform well across different devices and browsers.

RESPONSIBILITIES

- Develop and maintain the user interface of web and mobile applications.
- Collaborate with backend developers to integrate data and services.
- Ensure the application is responsive and works well on various devices.

REQUIRED SKILLS

- Proficiency in a frontend language (e.g., HTML, CSS, JavaScript).
- Experience with a frontend framework (e.g., React, Angular, Vue.js).
- Understanding of web protocols and HTTP.

SwipeHire

Find your next role, and swipe at it.™

Filters

Save

Edit Profile

My Matches

Growth

Saved Jobs

Jobs you bookmarked to apply to later.

Frontend Developer

AcademyTech

Full-time

100%

Remove



International Journal of DATA SCIENCE AND IOT MANAGEMENT SYSTEM

Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X
www.ijdim.com

Original Research Paper

Edit Your Profile

Bio
Computer Science Engineer with a background in Artificial Intelligence and Data Science, passionate about technology and continuous learning.

Location
Hyderabad, India

Expected Salary
Preferred Work Mode

Skills Type and press Enter

Education
B.Tech CSE SJCET 2020

Create your account

Signing up as a Recruiter

Full Name
Your name

Email
your@email.com

Password
At least 8 characters

Sign Up

Already have an account? Log In

SwipeHire
Find your next job, one swipe at a time.

Notifications

Frontend Developer
Academach

A Front End Developer builds the user-facing portion of websites and web applications. They are responsible for ensuring HTML, JavaScript, and CSS code is properly implemented and optimized for performance, responsiveness, and user experience.

Tell us about your company

This helps candidates navigate and trust your job postings.

Company Name
e.g. Acme Technologies

Company Website
https://yourcompany.com

Industry
e.g. Information Technology

Company Description
A short description of what your company does.

Complete Setup

Post a New Job

Fill in the details to create a new job posting.

Job Title
e.g. Frontend Developer

Job Description
Describe the role.

Location
Job Type

Salary Range
Work Mode

Minimum Experience Required (years)

Responsibilities Type and press Enter

Chat

Write a message...

Your Growth Dashboard
Monitor since July 2020

2 Jobs Applied
1 Match Received
20% Avg. Match Score
3 Jobs Saved

Applications Over Time

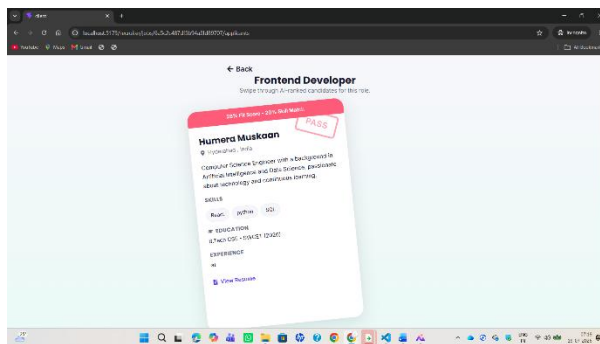
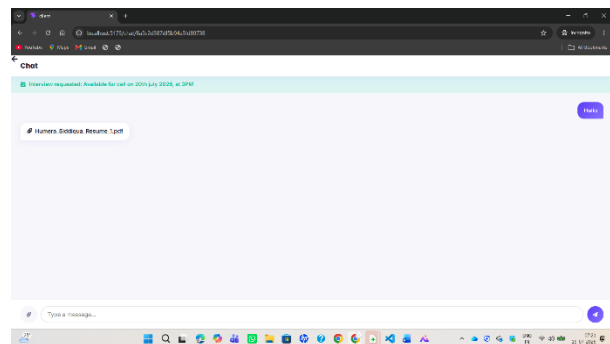
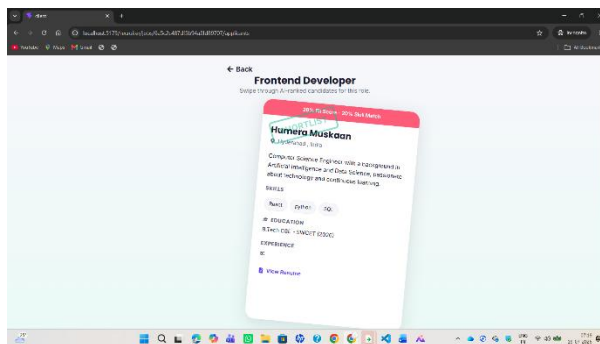
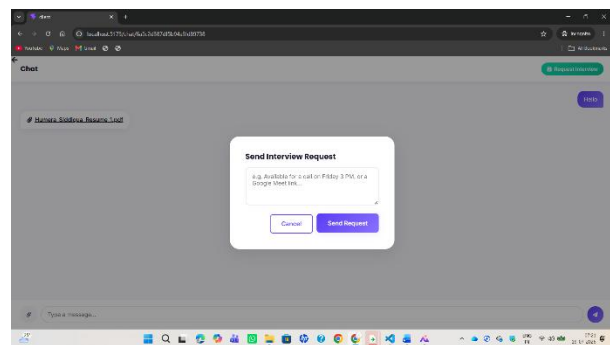
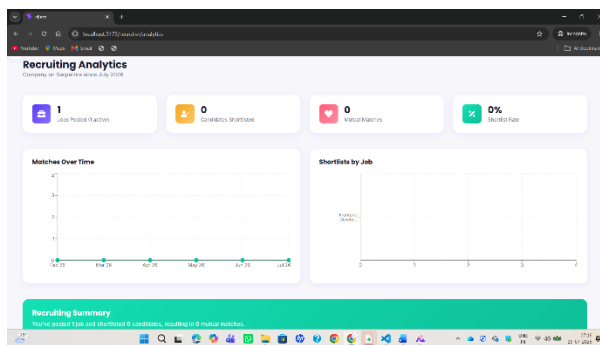
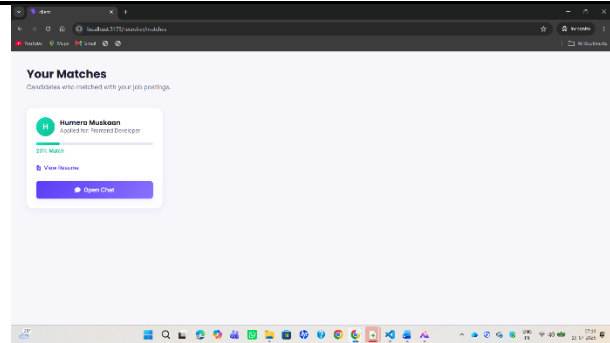
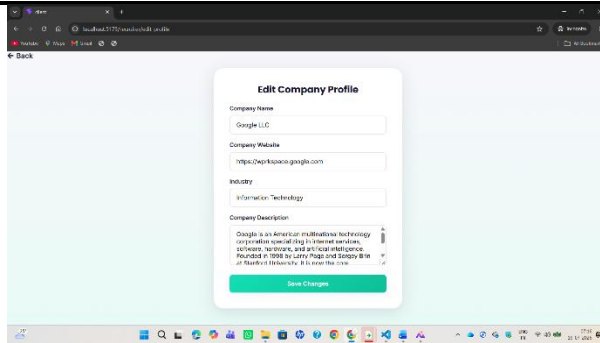
Your Most in-Demand Skills

Growth Summary
You've applied to 2 jobs and received 1 match so far. Your average skill match rate is 20%. Keep building your profile and applying to improve your standing.

Your Job Postings
Manage listings and see how candidates are matching.

Frontend Developer
Hyderabad, India - Full-time
A Front End Developer bridges the gap between design and technology, translating UI/UX wireframes into functional, interactive web applications.

Skills: HTML, CSS, JavaScript, React



8. CONCLUSION

The SwipeHire platform demonstrates that a transparent, rule-based matching algorithm combined with a familiar swipe-based interaction model can meaningfully streamline the job discovery and recruitment process for both candidates and recruiters. By computing and displaying an explainable match percentage at every stage — from the initial job feed through to recruiter-side applicant ranking — the system gives both parties a quantified, trustworthy basis for their decisions, rather

than relying on an opaque black-box score. The integration of real-time chat, file sharing, and recruiter-initiated interview requests directly within the matching flow closes a gap present in many existing swipe-style hiring prototypes, which typically stop at the match and leave subsequent communication to external channels. Supporting features — dual-sided analytics dashboards, job-feed filtering, save-for-later bookmarking, and an administrative panel for company verification and platform moderation — round out the system into a coherent, end-to-end product rather than a single-feature demonstration. Built on the MERN stack with Socket.IO for real-time communication, the platform is implemented as a fully functional, tested application covering authentication, dual-role profile management, AI-assisted matching in both directions, real-time interaction, and administrative oversight, validating the feasibility of a transparent, algorithm-driven approach to modern hiring platforms.

9. FUTURE ENHANCEMENT

The current skill-matching algorithm can be extended with weighted skill importance (allowing a job to mark certain skills as "must-have" versus "nice-to-have") and with semantic skill matching using embedding-based similarity, so that closely related skills (e.g., "React" and "React.js") are correctly recognized as equivalent even when phrased differently. The platform can be integrated with email notifications alongside the existing in-app alerts, ensuring users are reached even when not actively using the application. Video interview support could be added by auto-generating a meeting link (e.g., via a third-party video API) at the moment a recruiter sends an interview request, removing the need for manual link-sharing. A lightweight skill-verification layer — short, per-skill quizzes that award a "Verified Skill" badge — would strengthen the credibility of the AI match score by grounding it in demonstrated rather than self-reported ability. Finally, the recommendation logic could be extended over time from the current deterministic overlap algorithm toward a hybrid model that also incorporates collaborative signals (e.g., which job types similar candidates have historically matched with), while retaining the transparent overlap score as an explainable baseline shown alongside any learned recommendation.

REFERENCES

[1] S. Upadhyay and S. Khandelwal, "Applying artificial intelligence: implications for recruitment," *Strategic HR Review*, vol. 17, no. 5, pp. 255–258, 2018.

[2] I. Rahim, S. Ahmed, and R. Islam, "Job seekers' recommendation system based on collaborative filtering

and content-based filtering," in *Proc. Int. Conf. Comput. Adv. (ICCA)*, 2020, pp. 1–6.

[3] A. Gupta and R. Garg, "A review on employee-job matching using machine learning algorithms," *Int. J. Comput. Appl.*, vol. 178, no. 45, pp. 1–5, Jun. 2019.

[4] M. Al Zamil and S. Can, "A survey on job recommendation systems," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5195–5210, 2022.

[5] S. Chen, A. Zhang, and W. Liu, "Person-job fit: adapting the right talent for the right job with joint representation learning," *ACM Trans. Manage. Inf. Syst.*, vol. 9, no. 3, pp. 1–17, 2018.

[6] Y. Qin, Q. Liu, H. Zhu, and E. Chen, "Enhancing person-job fit for talent recruitment: an ability-aware neural network approach," in *Proc. 41st Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2018, pp. 25–34.

[7] A. Banerjee, "Building real-time applications with Socket.IO and Node.js," *Int. J. Eng. Res. Technol.*, vol. 9, no. 6, pp. 112–116, 2020.

[8] R. Verma and P. Sharma, "A comparative study of the MERN and MEAN stack for full-stack web development," *Int. J. Comput. Sci. Eng.*, vol. 8, no. 4, pp. 45–50, 2020.

[9] D. Crockford, "JSON Web Token (JWT) and its applications in modern authentication systems," *IEEE Softw. Eng. Notes*, vol. 44, no. 2, pp. 20–24, 2019.

[10] P. Kumar and N. Sharma, "MongoDB: a comprehensive study of NoSQL database and its applications," *Int. J. Adv. Res. Comput. Sci.*, vol. 11, no. 2, pp. 15–20, 2020.

[11] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, arXiv:1301.3781.

[13] S. Zhao, W. Du, X. Chen, and G. Wang, "Personalized job recommendation based on multi-source information fusion," *IEEE Access*, vol. 8, pp. 156158–156168, 2020.

[14] W. Yang, "Design and implementation of a swipe-based recommendation UI for mobile applications," *Int. J. Hum.-Comput. Interact.*, vol. 36, no. 15, pp. 1421–1432, 2020.

[15] React Team, "React – A JavaScript library for building user interfaces," Meta Open Source, 2023. [Online]. Available: <https://react.dev>

[16] Socket.IO Team, "Socket.IO Documentation," 2023. [Online]. Available: <https://socket.io/docs>