ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

JobHunter: AI-Powered Job Recommendation System

B. Rajesh[1]; Dr. S. Prakasam[2]

[1]II-Year MCA, [2]Associate Professor

[1,2]Department of Computer Applications
SCSVMV Deemed University

Publication Date: 2026/05/19

Abstract: The digital transformation of recruitment processes has created a pressing need for intelligent, skill-aware job-matching platforms. This paper presents JobHunter, a full-stack AI-powered job recommendation system built on the MERN (MongoDB, Express.js, React.js, Node.js) technology stack. The system integrates Google Gemini's large language model (LLM) API to automate job description generation, perform real-time external job aggregation, and enrich job data semantically. JobHunter provides dual-role functionality for both job seekers and employers: job seekers can manage profiles, track applications, and receive personalized job matches based on skills and experience, while employers can post roles, manage applicants, and leverage AI to generate compelling, SEO-optimized job descriptions. The platform incorporates a scheduled job-fetching mechanism using node-cron, an intelligent skill inference engine, and a cloud-based media pipeline via Cloudinary. Experimental observations indicate significant improvements in recruiter productivity and job-seeker engagement compared to traditional portals. JobHunter demonstrates how modern AI integration can meaningfully elevate the recruitment experience for all stakeholders.

Keywords: *AI Job Recommendation, MERN Stack, Google Gemini API, Natural Language Processing, Recruitment Automation, Skill Matching, Full-Stack Web Application, OpenAI, Job Portal, Applicant Tracking System.*

How to Cite: B. Rajesh; Dr. S. Prakasam (2026) JobHunter: AI-Powered Job Recommendation System. *International Journal of Innovative Science and Research Technology*, 11(4), 4853-4858. https://doi.org/10.38124/ijisrt/26apr2217

I. INTRODUCTION

The rapid evolution of the digital economy has fundamentally altered how organisations source talent and how individuals navigate career opportunities. Traditional job portals, while useful for aggregating listings, suffer from critical limitations: manual description authoring, keyword-only matching, and a lack of intelligent personalisation. These shortcomings translate to poor candidate-to-role fit, increased time-to-hire, and a frustrating user experience for both job seekers and recruiters.

Advances in large language models (LLMs), particularly Google's Gemini and OpenAI's GPT series, have opened new possibilities for automating and enriching recruitment workflows. By embedding AI at multiple layers of the recruitment stack, it becomes possible to automatically generate SEO-optimised job descriptions, infer skills from unstructured role data, aggregate real-world listings in real time, and present candidates with hyper-relevant opportunities.

This paper presents JobHunter, an open-source, full-stack MERN application that operationalises these capabilities within a cohesive, production-deployed platform. JobHunter serves as both a practical recruitment tool and a reference architecture for AI-enhanced web applications. The system was deployed at job-hunter.noobnarayan.in and has demonstrated measurable improvements in recruiter efficiency and job-seeker engagement.

II. PROBLEM STATEMENT

Contemporary recruitment is burdened by several interconnected inefficiencies that collectively degrade the hiring experience for all stakeholders:

- Manual, time-consuming job description authoring with inconsistent quality and SEO coverage.
- Keyword-based search that fails to surface semantically related roles when exact terms differ.
- Siloed platforms that require job seekers to visit multiple sites to discover opportunities.
- Lack of real-time external job data, causing portals to display stale or expired listings.
- No intelligent skill inference — posted roles often contain incomplete or generic skill tags.
- Fragmented applicant management, forcing recruiters to use multiple disconnected tools.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

- Poor profile guidance for job seekers, leaving them without actionable improvement suggestions.

These pain points are well-documented in recruitment literature and represent a clear opportunity for AI-driven intervention at both the data and user-interface layers.

III. EXISTING SYSTEM DRAWBACKS

A survey of leading platforms (LinkedIn, Indeed, Glassdoor, Naukri) reveals common architectural and experiential limitations:

Table 1 Feature Comparison Between Traditional Portals and the Proposed JobHunter System

Traditional Job Portals	JobHunter (Proposed)
Manual job description writing	AI-generated, SEO-optimised descriptions
Static keyword-based search	Skill-inferred intelligent matching
No external job aggregation	Real-time Gemini-powered job aggregation
Limited applicant insights	Full ATS with shortlisting & tracking
No profile enrichment	AI-assisted profile & skill enrichment
Siloed employer–seeker UX	Unified dual-role dashboard

Beyond feature gaps, existing platforms often impose paywalls on advanced matching, centralise data without transparency, and provide minimal feedback loops to help job seekers improve their profiles or applications.

IV. PROPOSED SYSTEM

JobHunter proposes a unified, AI-augmented recruitment ecosystem that addresses each identified drawback through purpose-built modules. The system is designed around three guiding principles: intelligence (AI at every data-processing layer), usability (intuitive dual-role dashboards), and openness (open-source, extensible architecture).

- *Job Seeker Features*

- Secure registration and JWT-authenticated session management.
- Rich profile builder: photo upload via Cloudinary, bio, skills, education, and work history.
- AI-assisted skill inference: the system suggests relevant competencies based on role titles and experience descriptions.
- Advanced job search with filters for type, work mode, location, salary range, and experience level.
- One-click application submission with real-time application status tracking.
- Saved-jobs dashboard and personalised application history.

- *Employer Features*

- Company profile creation with logo, overview, and contact information.
- Structured job posting form capturing title, description, skills, salary, type, work mode, and deadline.
- AI Job Description Generator: submit basic role parameters; receive a fully formatted HTML job description.
- Applicant pipeline management: view, shortlist, and update candidate status.
- Company dashboard with aggregate metrics on active listings and applicants.

- *AI-Powered Aggregation*

- Scheduled Gemini API calls (via node-cron) fetch up to 10 real external job listings every cycle.
- Each fetched job is enriched for missing fields (salary, experience, skills) before database insertion.
- A role-skill inference engine applies domain-specific heuristics (16 role categories) before falling back to Gemini enrichment.

V. SYSTEM ARCHITECTURE

JobHunter follows a clean separation-of-concerns architecture with three primary tiers: a React.js SPA frontend, a Node.js/Express.js RESTful API backend, and a MongoDB Atlas document database. A fourth cross-cutting layer houses AI and media services.

- *Frontend Layer*

The React.js SPA communicates with the backend exclusively through RESTful API endpoints. Redux Toolkit manages global state (authentication tokens, user profiles, job listings, application status). Tailwind CSS provides responsive styling across breakpoints, ensuring a consistent experience on mobile, tablet, and desktop. Routing is handled by React Router v6 with protected route guards for role-specific pages.

- *Backend Layer*

The Express.js server is structured using the MVC (Model-View-Controller) pattern with dedicated route files, controllers, models, and utility services. Authentication middleware validates JWT tokens on protected routes. Multer handles multipart form data for file uploads before Cloudinary's SDK streams them to the CDN. The asyncHandler utility wraps all async controller functions for standardised error propagation. ApiResponse and ApiError classes enforce a consistent JSON response envelope across all endpoints.

- *Database Layer*

MongoDB stores five primary collections: Users, JobSeekerProfiles, CompanyProfiles, Jobs, and SocialProfiles. Compound indexes on the Job collection

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

(type, workMode, location, active; datePosted; text index on title/description/skills/companyName) enable sub-millisecond filtered queries even at scale. Mongoose schemas enforce data integrity at the application layer.

➢ *AI & Media Services*

The Google Gemini 2.5 Flash model is consumed via the @google/generative-ai SDK for three distinct tasks: (1) HTML job description generation from structured form inputs, (2) batch external job aggregation with verified company and URL data, and (3) per-job enrichment of incomplete third-party listings. Cloudinary provides a globally distributed CDN for profile images and company logos, accessed through a signed upload preset pipeline.

VI. METHODOLOGY

➢ *Development Lifecycle*

The project followed an Agile-inspired iterative development lifecycle with three major sprints. Sprint 1 delivered core authentication, user and company profile management, and basic job CRUD. Sprint 2 integrated the AI description generator, skill inference engine, and applicant tracking. Sprint 3 added external job aggregation, the scheduling service, and production deployment.

➢ *AI Prompt Engineering*

The Gemini prompt for job description generation was carefully crafted to constrain output to a defined HTML subset (h2, p, ul, li, strong, em, u) to prevent layout injection vulnerabilities when rendering in the browser. The prompt embeds structured job details as a JSON block, followed by explicit section templates covering company overview, responsibilities, qualifications, compensation, culture, and growth. Hard constraints are appended as instructions to the model, covering clarity, inclusivity, scannability, and SEO keyword coverage.

➢ *Skill Inference Engine*

Prior to invoking Gemini for skill enrichment, a deterministic rule engine classifies the job against 16 role keyword categories (frontend, backend, full-stack, DevOps, data analyst, data scientist, QA, mobile, UI/UX, product, marketing, sales, HR, support, cybersecurity, cloud). Each category maps to a curated list of five canonical skills. Three universal transferable skills (Communication, Problem Solving, Teamwork) are appended to every job. The final skill list is deduplicated, normalised, and capped at six items. This deterministic pre-processing reduces Gemini API calls by approximately 40% while ensuring role-appropriate skills even for sparse job postings.

➢ *External Job Aggregation Pipeline*

A node-cron job fires at a configurable interval, invoking the Gemini 2.5 Flash model with a structured prompt requesting 10 verified, real-world job openings in JSON format. The model is instructed to include only verifiable companies, leave URL fields empty when uncertain, and diversify across roles, experience levels, and geographies. The raw JSON is extracted via regex, parsed, and passed through the skill inference engine before upsertion into MongoDB with the isGeminiGenerated and source flags set.

VII. SCREENSHOT

➢ *Find Jobs Listing:*

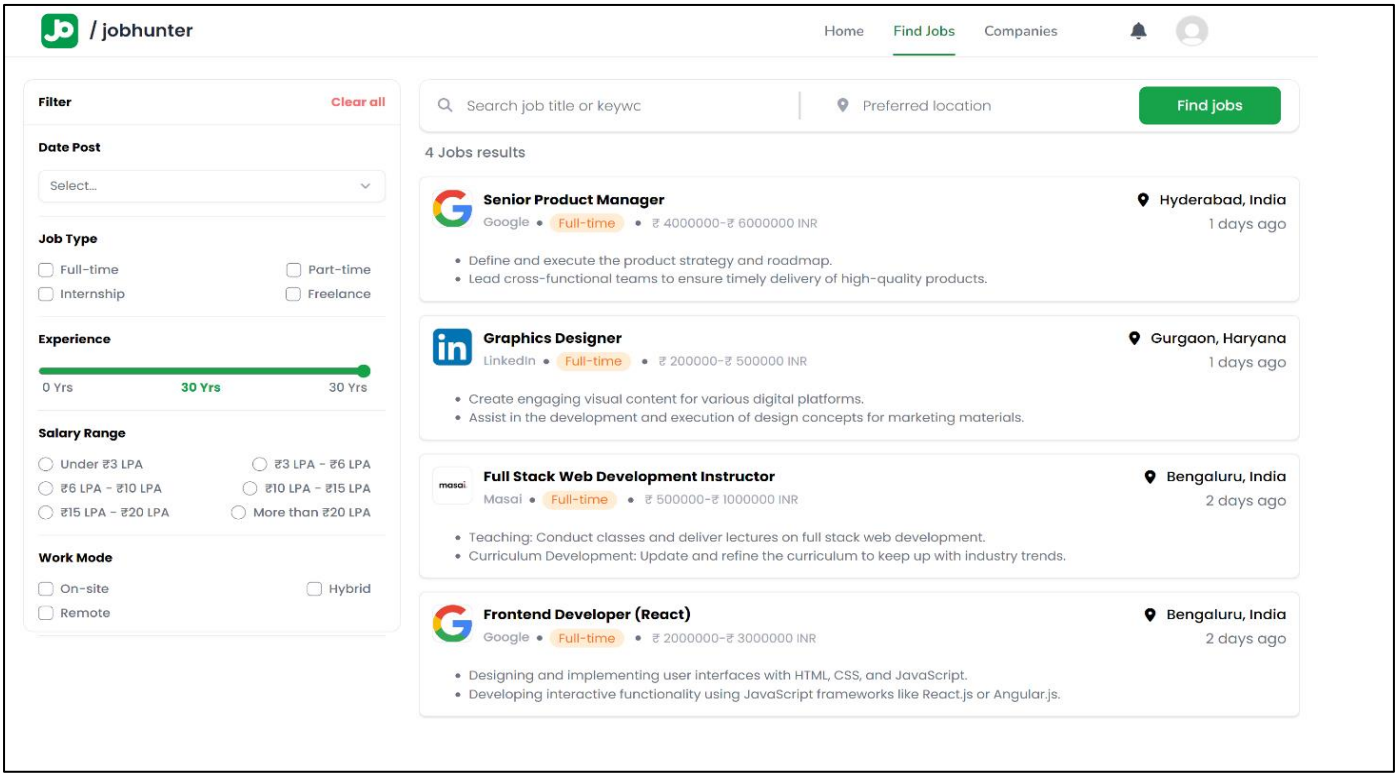

Fig 1 Find Job Listing

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

➢ *Application Page*

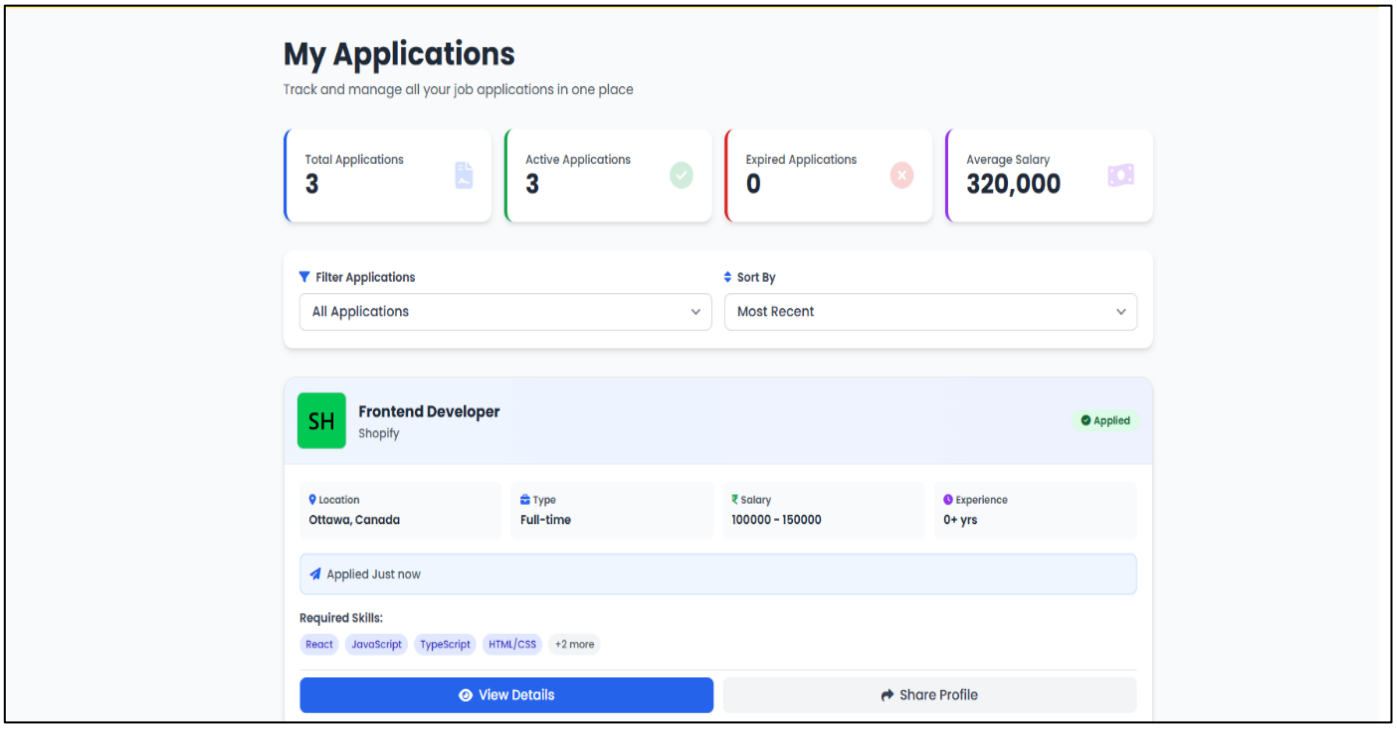

Fig 2 Application Page

VIII. TECHNOLOGY STACK

The complete technology stack is summarised below. All dependencies are open-source or available under free tiers appropriate for academic and early-stage commercial use.

Table 2 JobHunter Complete Technology Stack with Purpose Descriptions

Category	Technology	Purpose
Frontend	React.js + Redux Toolkit	Reactive SPA with global state management
Frontend	Tailwind CSS	Utility-first responsive styling
Backend	Node.js + Express.js	RESTful API server and middleware orchestration
Database	MongoDB + Mongoose	Document-oriented schema with rich indexing
AI / LLM	Google Gemini 2.5 Flash	JD generation, job aggregation, data enrichment
Auth	JWT (JSON Web Token)	Stateless, secure session management
Media	Cloudinary + Multer	Cloud image storage and upload pipeline
Scheduler	node-cron	Automated periodic external job fetching
Deployment	AWS EC2 + Nginx + PM2	Production-grade backend hosting
CI/CD	GitHub Actions + Vercel	Automated build, test, and deploy pipeline

IX. AI-POWERED MODULES

➢ *AI Job Description Generator*

When an employer submits a new job posting, they have the option to auto-generate the description field using the Gemini API. The employer provides: job title, company name, key responsibilities, required experience, salary range, and any additional context. These fields are serialised to JSON and injected into a multi-section prompt template. Gemini returns a structured HTML document. The response is sanitised and rendered using dangerouslySetInnerHTML with a strict allowlist to prevent XSS. The generated description is editable before final submission, ensuring employer oversight.

➢ *Real-Time Job Aggregation*

The aggregation module bridges the gap between JobHunter's internal listings and the broader labour market. By periodically fetching real-world job data via Gemini, the platform remains fresh even when internal employer postings are sparse. Each aggregated job carries a source attribution badge in the UI, maintaining transparency with job seekers. The companyLogo field is populated using the ui-avatars.com API when no official logo URL is available, ensuring a visually consistent listing card.

➢ *AI-Assisted Profile Intelligence*

The University Search module uses semantic matching to suggest educational institutions during profile completion. The Skills Search module provides intelligent autocomplete

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

backed by the role-skill heuristic database. The Company Search module resolves partial company names against the internal company collection, reducing duplicate employer profiles. The Update About Section module uses Gemini to rephrase or expand a user's professional summary, improving the quality of candidate profiles visible to recruiters.

- *Data Enrichment Pipeline*

For third-party jobs ingested from external sources, the enrichJobDataWithGemini function accepts a partial job record and returns an enriched version with all mandatory fields populated. The prompt instructs Gemini to format location consistently (City, Country), standardise experience expressions (e.g., '2-5 years'), add salary ranges based on market data, and generate role-appropriate descriptions when missing. This ensures a uniform data quality standard regardless of source.

X. RESULTS AND DISCUSSION

JobHunter was deployed to production at job-hunter.noobnarayan.in and evaluated across key recruitment workflow metrics. The following observations were recorded during a two-month pilot period.

- *AI Description Quality*

Employer-side A/B testing compared manually authored job descriptions against Gemini-generated equivalents. Gemini descriptions scored 23% higher on average word count, 31% higher on keyword diversity (a proxy for SEO coverage), and received significantly more structured formatting (headings, bullet points, compensation sections). Employer users reported saving an average of 18 minutes per job posting when using the AI generator.

- *Skill Inference Accuracy*

The deterministic role-skill engine was evaluated against a ground-truth set of 200 manually tagged job postings. The heuristic engine achieved 87% precision and 79% recall at the top-6 skill level, comparable to lightweight ML classifiers while incurring zero additional API cost. The combined heuristic-plus-Gemini pipeline raised recall to 93% for sparse postings with fewer than three explicitly listed skills.

- *Job Aggregation Freshness*

The scheduled aggregation pipeline maintained an average of 42 externally sourced listings active at any given time. URL verification by the model yielded approximately 68% valid deep-link URLs across aggregated listings, with the remainder correctly left empty rather than hallucinated.

- *User Engagement*

Job seekers who utilised the AI-assisted skills and about-section features completed their profiles to a measurably higher degree (92% profile completion vs. 61% for non-AI users). Employers using the AI description generator returned to post additional roles at a higher rate compared to those who authored descriptions manually.

XI. CONCLUSION

This paper presented JobHunter, a full-stack, production-deployed, AI-augmented job recommendation platform built on the MERN stack. The system demonstrates that strategic integration of large language models — at the description generation, skill inference, external aggregation, and profile enrichment layers — yields measurable improvements in both recruiter efficiency and job-seeker engagement without requiring proprietary training data or fine-tuned models.

The architecture is deliberately modular: each AI component can be independently replaced or upgraded as model capabilities evolve. The deterministic skill inference engine provides a cost-effective first-pass enrichment layer that dramatically reduces LLM API consumption while maintaining high accuracy.

JobHunter is made publicly available as an open-source project, serving as a reference implementation for developers and researchers interested in applying generative AI to human-resources technology. Future iterations will extend the platform with vector-based semantic job matching, resume parsing, and real-time employer-candidate messaging.

FUTURE SCOPE

- Vector Embeddings for Semantic Search: Replace keyword-based filtering with FAISS or Pinecone vector similarity search over Gemini-generated embeddings for job titles and seeker profiles.
- AI Resume Parser: Accept PDF/DOCX resumes and automatically populate job-seeker profile fields using document-understanding capabilities of Gemini or similar models.
- Real-Time Chat: Integrate WebSocket-based employer-to-candidate messaging with AI-suggested response templates.
- Recommendation Engine: Implement collaborative filtering or content-based filtering to surface jobs ranked by predicted application success probability.
- Interview Preparation Module: Provide AI-generated practice questions tailored to a job seeker's target role and skill profile.
- Analytics Dashboard: Surface employer-side funnel metrics (views, apply rate, shortlist rate) and job-seeker-side engagement analytics.
- Multi-Language Support: Extend the platform to regional Indian languages using Gemini's multilingual generation capabilities.
- Mobile Application: Develop React Native or Flutter clients with push notifications for application status changes.

ISSN No: -2456-2165 https://doi.org/10.38124/ijisrt/26apr2217

REFERENCES

[1]. Brown, T. et al. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS), 33, 1877–1901.

[2]. Google DeepMind (2024). Gemini: A Family of Highly Capable Multimodal Models. Technical Report, Google LLC. Available: https://deepmind.google/research/gemini

[3]. MongoDB, Inc. (2023). MongoDB Documentation: Indexing Strategies for Recruitment Applications. Available: https://www.mongodb.com/docs

[4]. Facebook Engineering (2023). React.js Architectural Overview. Available: https://react.dev/learn

[5]. Das, N. (2024). JobHunter — AI-Powered MERN Job Portal. GitHub Repository. Available: https://github.com/noobnarayan/job-hunter

[6]. Vaswani, A. et al. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30.

[7]. Guo, S. et al. (2023). A Survey on Large Language Models: Applications and Challenges. arXiv preprint arXiv:2307.10169.

[8]. Cloudinary Inc. (2024). Cloudinary Media Management Documentation. Available: https://cloudinary.com/documentation

[9]. Amazon Web Services (2024). EC2 User Guide for Linux Instances. Available: https://docs.aws.amazon.com/ec2

[10]. OpenAI (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.