IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

EMPLOYMENT HIRING PLATFORM USING AI

Sanjay Kumar[1], Prince Pandey[2], Md Shahjad[3], Deepak Kumar[4]

[1]UG student of the Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh

[2]UG student of the Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh

[3]UG student of the Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh

[4]Assistant Professor, Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh

Abstract: This paper presents JobVista, a full stack employment hiring platform designed to support students, job seekers, and recruiters through a role-aware digital recruitment workflow. The platform provides secure authentication, candidate profile management, recruiter company management, job posting, job browsing, application tracking, applicant review, and assistant-driven guidance through JobMate. The implemented system uses React, Vite, Redux Toolkit, Express.js, Node.js, MongoDB, Mongoose, JWT-based cookie authentication, and bcrypt password hashing. JobVista also improves early platform usefulness by showing selected external opportunities and by providing template and Gemini-backed assistant responses for resumes, interviews, cover letters, job posts, and screening questions. The paper discusses the problem background, proposed methodology, architecture, database design, module implementation, testing, results, limitations, and future enhancements. The final system demonstrates that a college-level job portal can be technically meaningful, visually usable, and extensible without becoming unnecessarily complex.

Keywords: JobVista, job portal, employment hiring platform, React, Node.js, Express.js, MongoDB, role-based access control, application tracking, JobMate, AI career assistant

1. INTRODUCTION

Digital recruitment has become a major part of campus placement, internship search, and early-career hiring. Students need one place where they can build a useful profile, discover opportunities, and apply quickly. Recruiters need a structured workflow for company information, job posting, applicant review, and decision tracking.

JobVista was developed as a full stack hiring platform that joins these needs inside one web application. The system supports two main roles: student and recruiter. Students can register, complete their profile, search jobs, view job details, apply for openings, and receive career guidance through JobMate. Recruiters can create companies, post jobs, see applicants, and update application status.

The project is academically useful because it combines frontend engineering, backend API design, database modeling, authentication, authorization, external API usage, and assistant-driven support. Instead of being only a job-listing page, JobVista represents an end-to-end hiring workflow.

2. PROBLEM STATEMENT

Many small academic job portals provide only login, registration, and a basic job list. Such systems do not properly model the difference between candidates and recruiters, and they often do not include application tracking or meaningful profile data.

Another problem is fragmentation. Students may search for jobs in one place, maintain resumes elsewhere, and ask for interview help through a separate tool. Recruiters may post jobs but still manage applicants manually. JobVista

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

addresses this gap by bringing job discovery, profile data, recruiter workflows, application records, and career assistance into one maintainable platform.

3. OBJECTIVES

- Provide secure registration, login, logout, and JWT cookie-based session handling.
- Support separate workflows for students and recruiters using role-based access.
- Store rich candidate profile data such as skills, education, projects, internships, links, and resume information.
- Allow recruiters to manage companies, publish jobs, review applicants, and update application status.
- Provide JobMate guidance for resumes, interviews, cover letters, job posts, and candidate screening questions.
- Keep the codebase modular so that future features can be added without redesigning the whole system.

4. PROPOSED METHODOLOGY

The proposed methodology follows a layered MERN-style architecture. The React frontend manages user interaction and shared state. The Express backend validates requests, applies authentication and role rules, and performs database operations. MongoDB stores users, companies, jobs, and applications as related documents.

The development process was iterative. Authentication was implemented first, followed by profile handling, company management, job posting, job browsing, application tracking, applicant review, and JobMate assistant support. This order reduced integration risk because each new module could build on stable identity and data foundations.

Component	Implementation in JobVista
Frontend	React, Vite, Redux Toolkit, React Router, Tailwind CSS, Axios, reusable components.
Backend	Node.js and Express.js routes, controllers, middleware, and REST API handlers.
Database	MongoDB with Mongoose models for User, Company, Job, and Application.
Security	JWT cookies, bcrypt password hashing, authentication middleware, and role checks.
AI Support	JobMate template responses with optional Gemini API generation and safe fallback behavior.
External Jobs	Selected external opportunities are merged with platform jobs to improve discovery.

5. SYSTEM ARCHITECTURE

The architecture separates presentation, state management, API logic, business rules, persistence, and optional external services. This makes the project easier to test, explain, and extend.

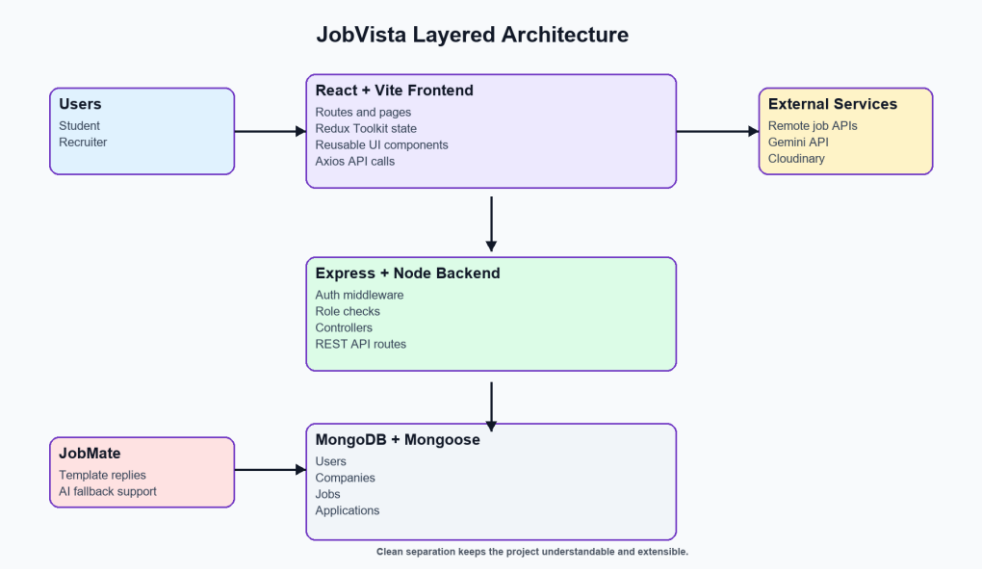

Figure 1. Layered architecture of JobVista

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

6. DATABASE DESIGN

The database design uses four main entities. The User model stores identity, role, and detailed profile data. Company records are owned by recruiters. Jobs are linked to a company and recruiter. Applications connect a student applicant with a job and maintain the current recruitment status.

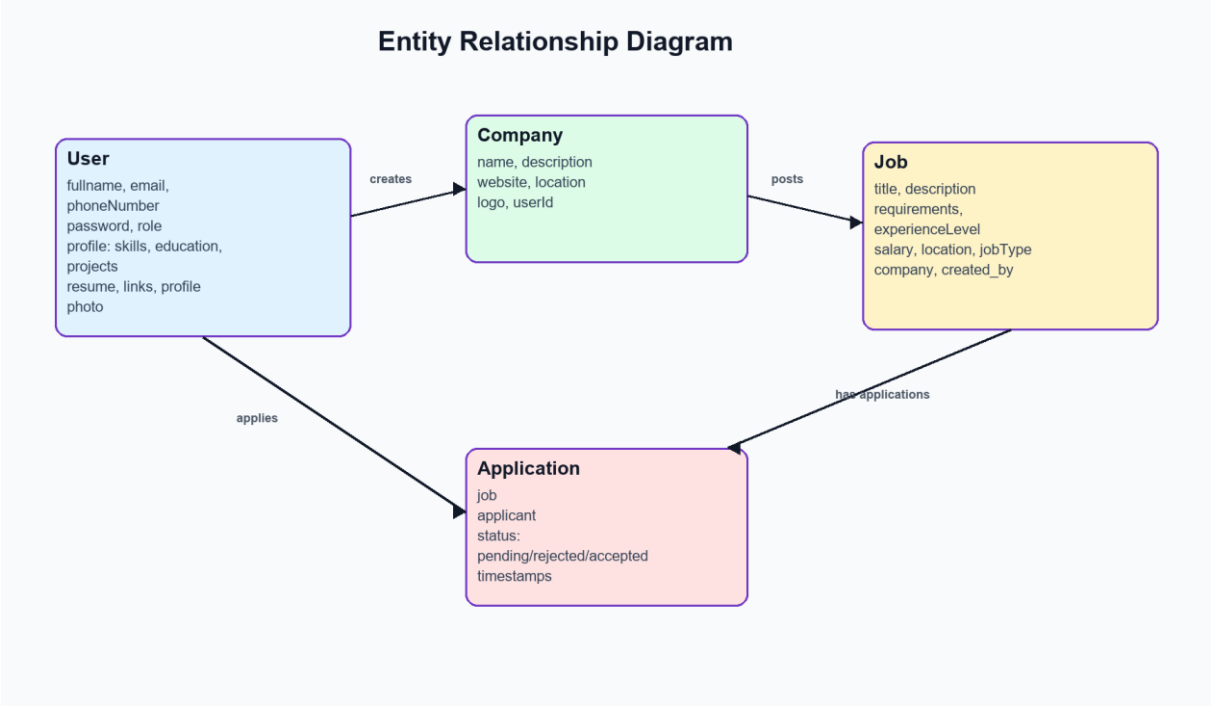

Figure 2. Entity relationship diagram

7. DATA FLOW AND WORKFLOW

The data flow begins when a student or recruiter sends a request through the frontend. Protected operations pass through authentication middleware before controllers access MongoDB. Optional services, such as external jobs and JobMate, enrich the experience without breaking the main application when they are unavailable.

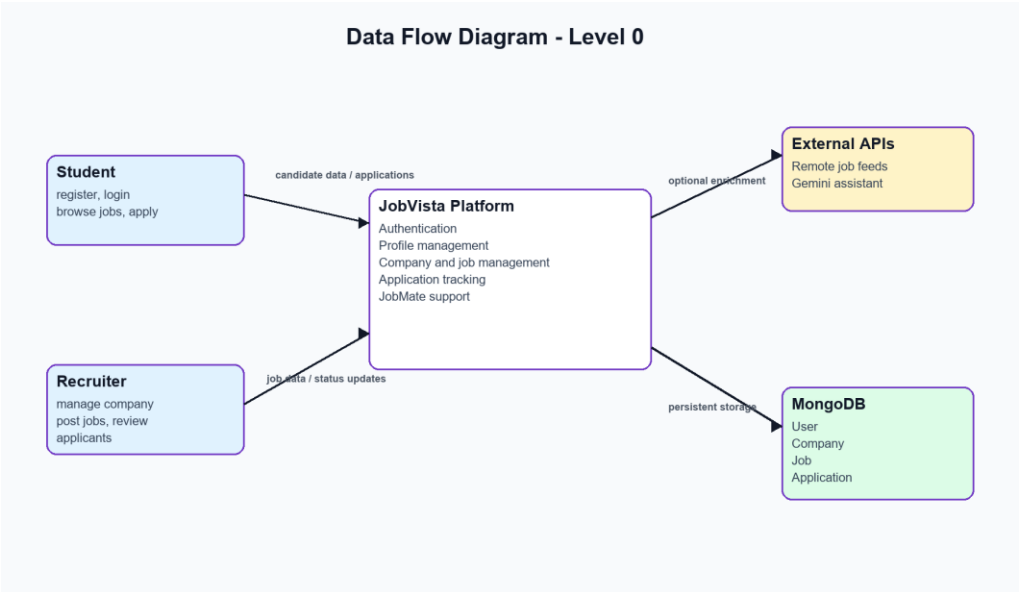

Figure 3. Data flow diagram level 0

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

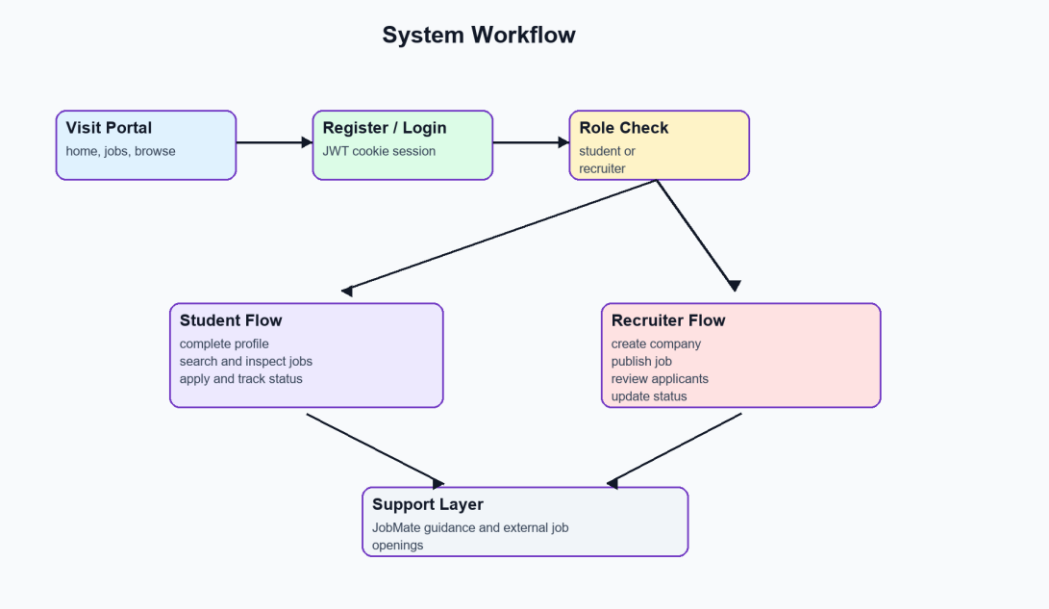

Figure 4. Overall workflow of the JobVista system

8. MODULE IMPLEMENTATION

8.1 Student Module

The student module supports registration, login, profile completion, job search, job detail viewing, and application submission. Profile fields include skills, education, projects, internships, experience, location, portfolio, LinkedIn, GitHub, and resume details.

8.2 Recruiter Module

The recruiter module supports company creation, company update, job posting, applicant viewing, and status updates. This workflow gives recruiters continuity from company identity to job publication and candidate review.

8.3 JobMate Assistant

JobMate provides practical guidance for resumes, interviews, cover letters, job posts, applications, and candidate screening questions. The backend first uses reliable template replies for common queries. If a Gemini API key is configured, it can generate additional responses. If the external service fails, the system returns a safe demo response instead of crashing.

8.4 Application Tracking

The Application model connects a job and an applicant and stores status as pending, rejected, or accepted. This gives both sides a clearer representation of hiring progress than a simple submit-only form.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

9. IMPLEMENTATION SCREENS

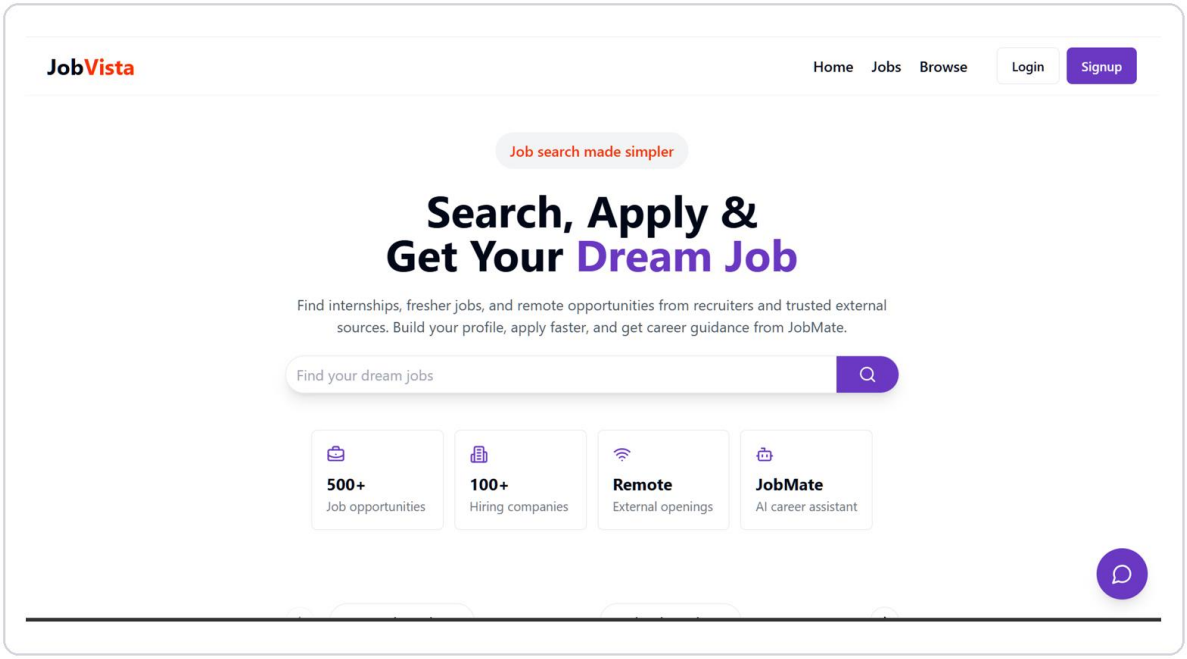

Figure 5. Home page of JobVista with search, navigation, and platform statistics

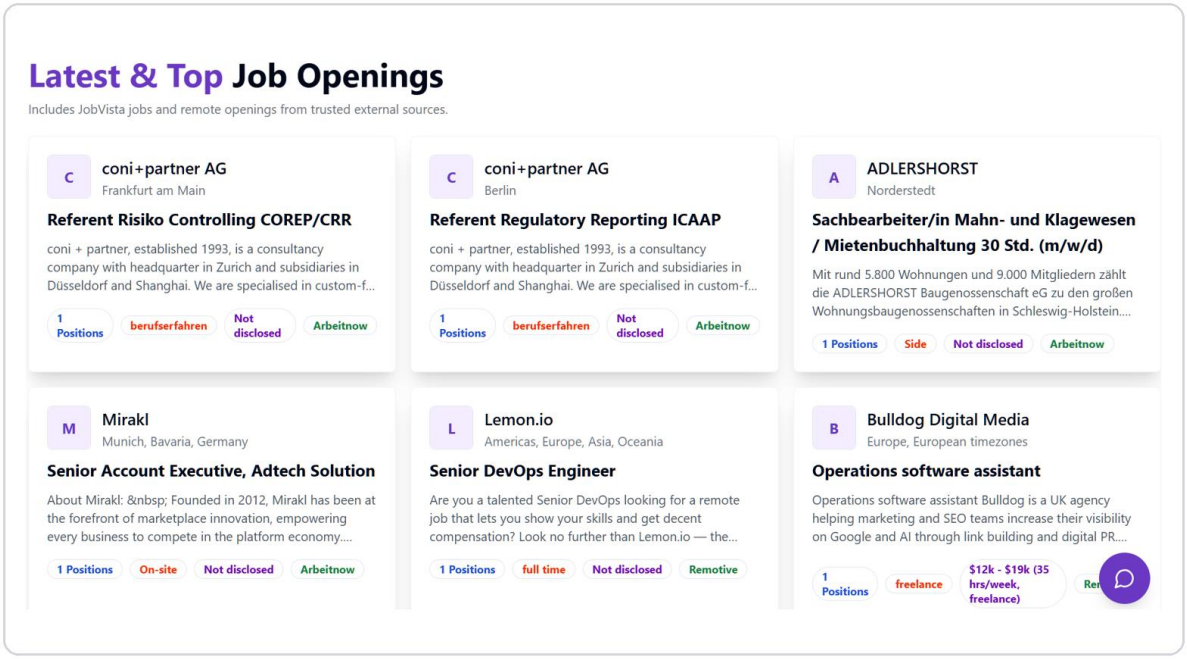

Figure 6. Latest and top job openings displayed through responsive job cards

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

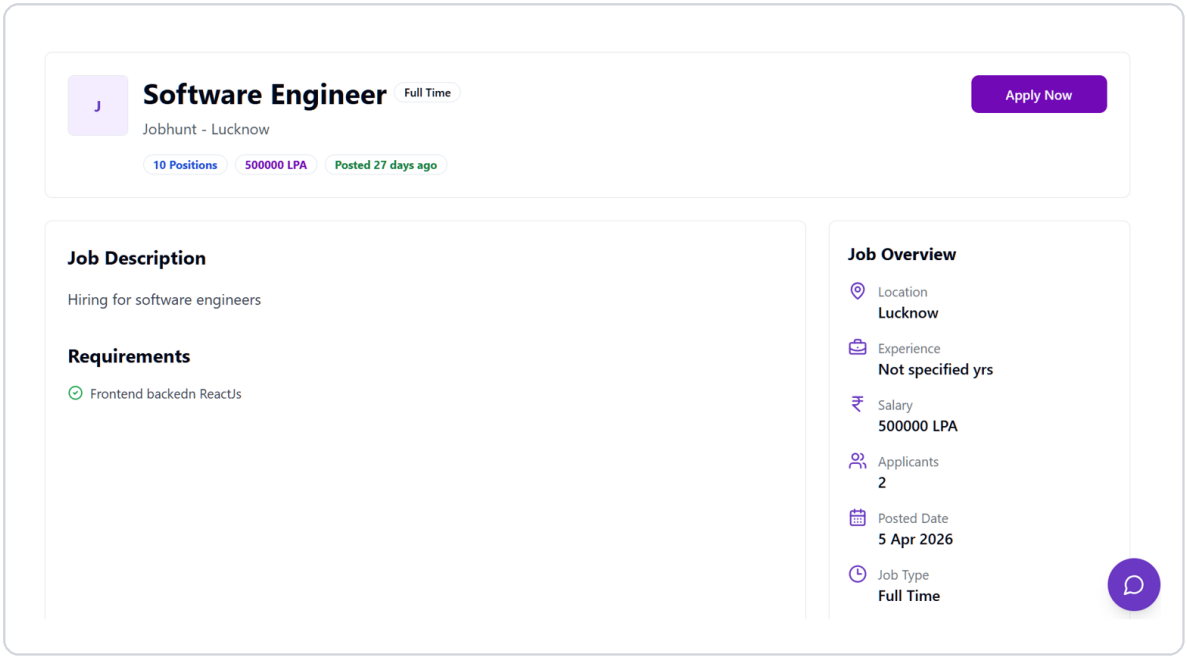

Figure 7. Job details page with description, requirements, overview, and apply action

10. TESTING AND VALIDATION

- Registration and login were checked for both student and recruiter roles.
- Protected routes were checked to confirm that unauthenticated users cannot access restricted actions.
- Student profile update, job browsing, job detail view, and application submission were verified manually.
- Recruiter company creation, job posting, applicant listing, and status update flows were verified.
- JobMate was checked with resume, interview, cover letter, recruiter, application, and screening-question prompts.
- External service fallback behavior was checked so the application remains usable when an optional API is unavailable.

11. RESULT

The implemented JobVista system successfully demonstrates a practical employment hiring platform with separate student and recruiter workflows. The frontend presents a clean interface for job discovery and job details, while the backend maintains structured records and secure access rules.

The system result is stronger than a basic CRUD portal because it includes recruiter-side continuity, detailed candidate profiles, application tracking, assistant support, and external opportunity enrichment. The screenshots show that the platform has a usable visual interface, and the diagrams show that the internal structure is modular and explainable.

12. LIMITATIONS

- The system currently uses manual testing more than automated test coverage.
- Advanced recommendation logic and resume parsing are not yet implemented.
- Recruiter analytics, notification systems, interview scheduling, and saved jobs are future enhancements.
- External job and AI features depend on third-party service availability, although fallback behavior is included.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15526

13. FUTURE SCOPE

Future versions can add saved jobs, notifications, profile completion scoring, skill-based job recommendations, resume parsing, recruiter notes, interview scheduling, analytics dashboards, automated tests, and deployment monitoring.

JobMate can also be improved by using profile and job context to generate more personalized resume suggestions, interview preparation notes, and recruiter screening material.

14. CONCLUSION

JobVista provides a meaningful implementation of a modern employment hiring platform. It combines secure authentication, role-aware navigation, student profiles, recruiter company and job workflows, application tracking, external job discovery, and assistant-driven support inside one modular full stack system.

The project satisfies the objective of building an academically presentable and technically useful job portal. Its single-column research-paper format, clean diagrams, and real implementation screenshots make it clearer for college submission, seminar explanation, and viva discussion.

REFERENCES

[1] React Documentation, official references on component-based user interface development.

[2] Redux Toolkit Documentation, official references on predictable state management.

[3] Vite Documentation, official references on frontend tooling.

[4] Node.js Documentation, official references on server-side JavaScript runtime behavior.

[5] Express.js Documentation, official references on routing and middleware.

[6] MongoDB and Mongoose Documentation, official references on document database design and ODM usage.

[7] JWT and bcrypt documentation, references on authentication tokens and password hashing.

[8] Google Gemini API documentation, references on generative assistant integration.

[9] JobVista project codebase, including frontend routes, backend controllers, middleware, and Mongoose models.