

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 11 ISSUE : 07 Print / Issue Publication Date: November 2026

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2026.v11i07.002

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

CAREEROS: AN INTEGRATED AI-ASSISTED PLATFORM FOR PERSONALIZED CAREER PREPARATION AND JOB APPLICATION MANAGEMENT

Soham Mali, Rushikesh Waghmare, Abhay Gaikwad, Sachin Nanvare, and Prasad Pratape
Department of Computer Science and Engineering,
MIT College of Railway Engineering & Research,
Barshi, Maharashtra, India,
Affiliated to Punyashlok Ahilyadevi Holkar Solapur University,
Solapur, Maharashtra, India

Abstract—**Career preparation for technology students spans resume improvement, evidence of practical skills, interview rehearsal, structured learning, portfolio development and job-application tracking. These tasks are often handled in separate tools, leaving students to carry information manually between services. This paper presents CareerOS, a full-stack web application that integrates resume analysis, GitHub profile analysis, AI-assisted text and voice mock interviews, career-profile aggregation, personalized roadmaps, adaptive quizzes, portfolio publishing and job-search/application tracking. The client uses React, React Router and Vite with Tailwind-based components; a Node.js/Express API coordinates business logic, and MongoDB/Mongoose stores user-owned records. Groq-based language-model calls generate structured feedback and learning content. The optional voice workflow combines speech transcription with speech synthesis, while Cloudinary handles supported uploaded files and audio assets. The evaluation in this paper is artefact-based: source files and supplied interface screenshots were inspected to document modules and visible states. This is not a controlled user study or an accuracy benchmark. The contribution is a description of how career evidence, practice, learning plans and job-search workflows are integrated in one workspace, together with an explicit account of what remains to be validated.**

Index Terms—**career guidance, resume analysis, large language models, mock interview, skill-gap analysis, adaptive learning, job recommender, web application.**

I. INTRODUCTION

The transition from academic learning to employment requires more than completing coursework. Students must explain technical experience in a resume, identify gaps against a role, practise interview responses, maintain a visible portfolio and manage applications. Separate tools may support these activities individually, but the learner must copy information between them and progress can become difficult to interpret as one plan. CareerOS addresses this coordination problem by connecting several career-preparation workflows under one account.

The platform is intended as a student-facing preparation workspace, not an automated hiring-decision system. Resume scores, interview evaluations and job-match percentages are advisory outputs. Research on language-model resume matching shows that generated ratings can differ from human judgments [3], and automated interview assessments require careful evidence of reliability, validity and fairness [5], [6]. For this reason, CareerOS is described as an integration artefact, not as a validated predictor of employment.

The work makes three practical contributions. First, it documents a modular full-stack architecture joining career evidence, learning practice and job-application workflows. Second, it explains the interpretable skill-overlap and quiz-difficulty rules used by the implementation. Third, it separates observations available from source code and screenshots from performance claims that would require controlled evaluation.

II. RELATED WORK

A. Career guidance and job recommendation

Career-guidance research has explored systems that connect student skills with possible information-technology roles. C3-IoC combines student skill profiling with role matching and network visualization to support career exploration [2]. CareerOS follows the broad objective of relating existing skills to future steps, while adding resume analysis, interview rehearsal, portfolio publishing and application tracking within the same workflow. A systematic review of

job recommender systems categorizes content-based, collaborative, hybrid and knowledge-based approaches and identifies open issues involving evaluation, data quality and bias [1]. The current CareerOS matcher uses normalized skill overlap because it is easy to explain, but it cannot interpret every synonym or infer skill proficiency from context.

B. Resume matching and recruitment

Resume analysis typically combines document extraction with an evaluation against a role or job description. Vaishampayan et al. examined LLM-based and human resume ratings across 736 resumes and reported limited correlation between the two rating sources [3]. Herold and Roedenbeck reviewed AI-supported recruitment and selection and described the role of natural-language processing and machine-learning methods alongside ethical concerns [4]. These findings motivate a cautious interpretation of CareerOS output: the analysis can suggest missing keywords or improvements, but it should not be treated as a definitive assessment of a candidate.

C. Interview assessment and learning practice

Automated video-interview research has evaluated the reliability, validity and generalizability of personality or cognitive assessments under different modelling conditions [5], [6]. CareerOS implements interview-question generation, answer-level feedback and a final debrief for practice; it does not claim psychometric validation. Distributed-practice research reports that the spacing interval and the desired retention interval interact in determining recall [7]. The project uses a simple accuracy-based review schedule. That rule is an implementation choice that requires longer-term learner testing before stronger learning-outcome claims can be made.

D. Responsible AI and secure web implementation

The NIST AI Risk Management Framework recommends identifying, governing and managing risks throughout the design and use of AI-enabled systems [8]. For a career-preparation product, responsible use includes explaining the intended purpose of scores, retaining only necessary personal data, providing correction pathways and testing for inconsistent feedback across different writing styles. The React, Node.js and MongoDB documentation describe the foundational tools used to implement the client and persistence layers [9]–[11]. OWASP guidance on authentication and token handling informs the web-security design [12].

TABLE I. QUALITATIVE COMPARISON OF RELATED APPROACHES

Approach	Strength	Limitation
Standalone resume checker	Fast document feedback and keyword suggestions.	Usually disconnected from learning and application tracking.
Job recommender [1]	Ranks opportunities from profile or content signals.	Needs reliable labels, freshness checks and bias evaluation.
Career guidance graph [2]	Relates student skills to possible roles.	Needs maintained role-skill mappings.
CareerOS	Connects evidence, practice, learning and applications.	Integrated workflows remain to be benchmarked at scale.

III. RESEARCH METHOD AND EVALUATION SCOPE

This work is a software design-and-implementation study, not a randomized or controlled user experiment. The primary artefact is the submitted frontend and backend source archive, including React pages, Express routes and controllers, Mongoose models, external integrations and backend test files. The secondary evidence is the set of eight supplied interface screenshots covering the dashboard, resume analysis, administration page, adaptive quiz, voice interview, roadmap, job tracker and light theme.

The inspection procedure had four steps. First, the main modules were mapped to the relevant source folders and route/service groups. Second, screenshots were examined to identify which workflows and example UI states were visible. Third, source-level data models and integration modules were compared with those visible workflows. Fourth, claims were classified as implementation evidence, screenshot observation or a question requiring additional testing. The automated test suite was not re-executed while preparing this paper; therefore no test pass count is asserted here as a new result.

The evidence boundary is important. A visible control can demonstrate that an interface was built, but it does not prove that every API response is fresh, that authorization is secure or that an AI score is accurate. No labelled resume corpus, expert-rated feedback set, interview-rater dataset, usability dataset or job-placement outcome data was supplied. Results below are therefore descriptive, and proposed evaluation metrics are presented as future work rather than as completed experiments.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

Evaluation plan
A full evaluation should use synthetic or consented resumes, clearly labelled job descriptions, test accounts with separate ownership, and multiple human reviewers for interview and resume feedback. Measurements should include text-extraction precision and recall, expert agreement with generated recommendations, precision and recall for job relevance, task completion rate, response latency, external-service failure rate, and accessibility checks. Test data should avoid real candidate identifiers whenever possible.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

CareerOS uses a client–API–data structure. The React client handles routing, forms, page state and feedback. The Node.js/Express REST API validates requests and delegates business logic to route, controller and service modules. MongoDB persists users and user-owned records through Mongoose schemas. Separate integrations connect the platform to AI generation, audio transcription, speech synthesis, public GitHub data, job-search providers and file storage. The implementation's language-generation integration is based on the Groq API [13], and Adzuna documentation describes one of the configured job-search providers [14]. A browser request passes from a protected page to the API, where authentication, validation and ownership checks should run before data access.

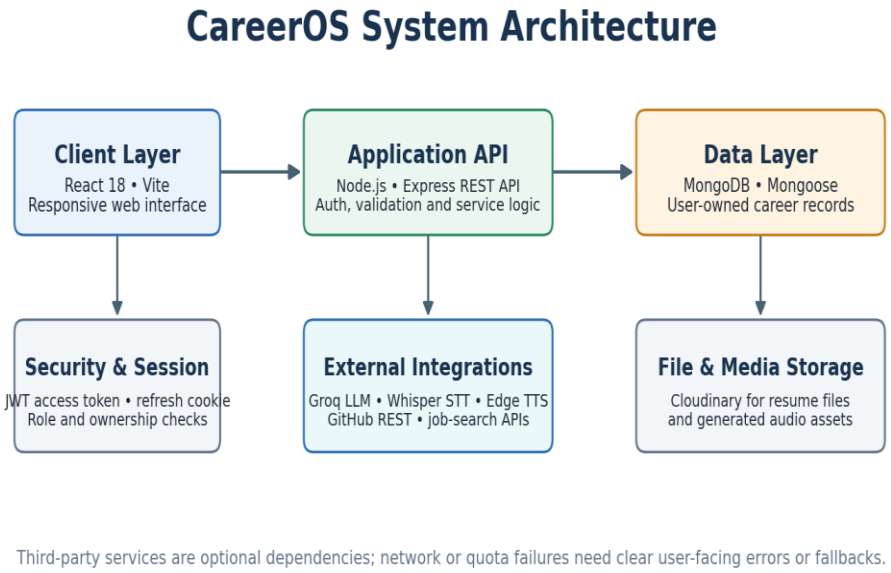


Fig. 1. Conceptual architecture of the CareerOS client, API, data store and external-service layers.

A. Main application modules
The resume service accepts a supported document, extracts text, validates that the content resembles a resume, stores file metadata and requests structured feedback. The GitHub service gathers available public repository and language evidence before producing a human-readable summary. The career-profile aggregator combines available outputs from resume, GitHub and completed interviews. The roadmap service uses the career profile to identify known and missing skills and generates an ordered list of learning nodes. The portfolio module stores user-editable content and exposes a public slug when published. The job-search service retrieves listings from configured providers or documented fallbacks and calculates skill-overlap indicators. The dashboard consolidates summaries from these modules, while administrative pages show aggregate platform activity.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

TABLE II. MODULE-LEVEL INPUTS AND EVIDENCE BOUNDARY

Module	Input and output	Evidence boundary
Resume	Document + optional job description → structured feedback.	Screenshot shows form/history; extraction accuracy not measured.
Interview	Role and answer history → questions and debrief.	Recorder and answer UI shown; rating validity unknown.
Roadmap/quiz	Target role or topic → skill nodes/questions.	Controls visible; learning gains not measured.
Jobs/portfolio	Profile skills and listings → match labels/application pipeline.	UI visible; job freshness and relevance not benchmarked.

B. Persistence and ownership

The core user record is the ownership root for resume analyses, GitHub analyses, interview sessions, career profiles, roadmaps, portfolio content, quizzes, quiz attempts, job postings and applications. Refresh-token records are separate and use hashed token material with expiry data. For private resources, the authenticated identity should be checked against the owner field for each read or update; this is a central invariant because career documents and interviews may contain personal data. Schema-based validation also helps keep AI-generated JSON and user-submitted fields within expected shapes.

C. AI and voice workflow

Text-generation requests are routed through a Groq integration and ask for structured output that the service can validate before it reaches the user. The interview module selects questions based on role, difficulty, interview type and target skills, records the answer history, and requests answer-level feedback. The optional voice path combines transcription through a Whisper-based service and spoken output through an Edge Neural TTS integration. Browser microphone permission and supported speech APIs vary by environment, so the system should always expose a recoverable typed-answer path.

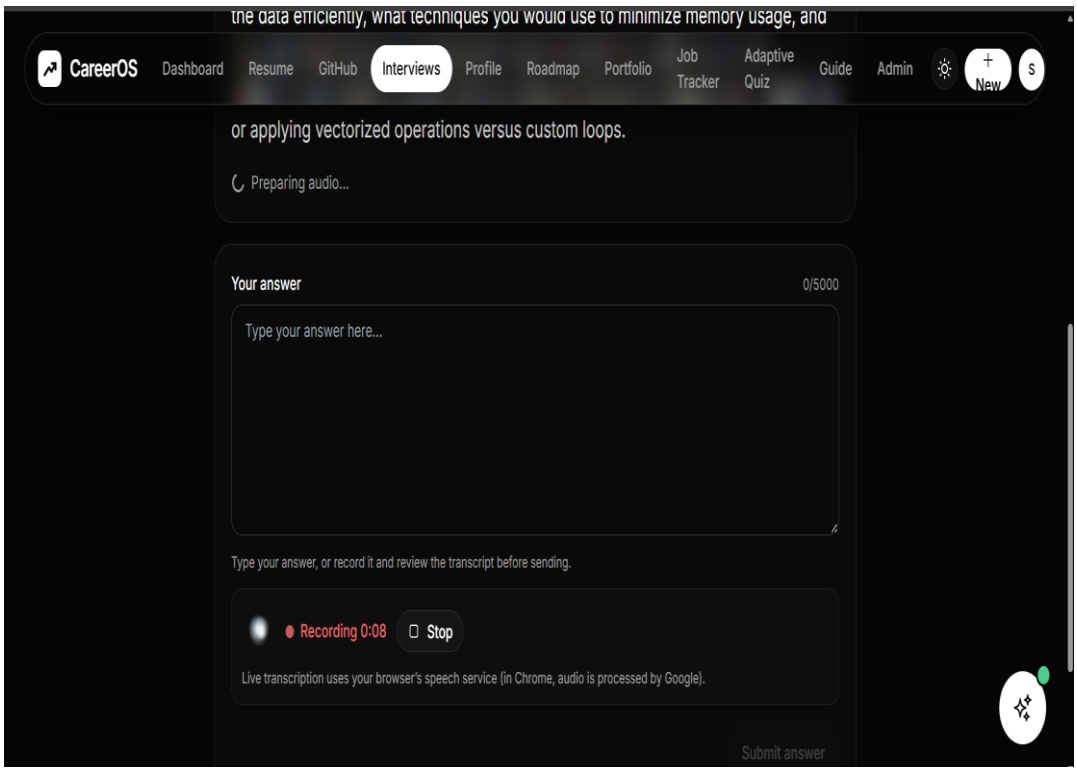


Fig. 2. Mock-interview answer interface with a recording state and transcription guidance.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

D. Data lifecycle and failure handling

A normal workflow begins with an authenticated request from the browser. The API identifies the user, validates the request shape and passes the request to a service. When a file is involved, the upload middleware applies file constraints before a storage integration is called. The service stores metadata and the structured output in records associated with the authenticated user. The response sent to the client should contain only fields needed by the interface. A request identifier and a sanitized error category can help operators trace failures without retaining entire resumes, interview answers or authentication material in application logs.

External providers create a boundary between internal records and external processing. AI prompts may include extracted resume text or interview answers; those inputs can contain sensitive details. A deployment should disclose this processing and send only the context necessary for the requested output. API credentials should stay on the server, use environment configuration and be rotated if exposure is suspected. Retry logic should be bounded to avoid duplicate records or repeated charges. When a provider is unavailable, the client should distinguish an actual empty result from a failure or a demonstration fallback. These practices improve clarity, but a complete privacy and security assessment remains outside the evidence available for this paper.

E. Design trade-offs

A shared profile makes the user experience coherent because a resume analysis or interview report can inform a roadmap and skill-gap display. The same integration also increases the importance of data ownership: an incorrect authorization check could expose multiple categories of personal career information rather than a single feature. A modular route/service structure therefore helps isolate provider integrations and apply consistent checks, but code organization alone does not guarantee security. Similarly, structured AI output is easier to display and validate than free-form text, yet a syntactically valid response can still be factually inaccurate. The client should allow users to revise their data and treat generated feedback as one source of guidance rather than a substitute for human judgment.

V. JOB MATCHING AND ADAPTIVE PRACTICE

A. Transparent skill-overlap score

The skill-gap utility trims skill labels, converts them to lowercase and collapses repeated spaces before comparing the required-skill list with the normalized career-profile skill set. Let R denote the set of normalized required skills extracted for a vacancy and S denote the set of skills in the profile. For non-empty R, the displayed match percentage is:

$$\mathbf{M = 100 \times |R \cap S| / |R|}$$

For example, if a posting lists five required skills and the profile contains three exact normalized matches, the simple overlap is 60%. This value is an explanation of overlap, not a probability of receiving an interview. The implementation returns 100% by convention when the required-skill list is empty; that edge case should be disclosed or revised because an absent requirement list is not evidence of a complete match. Exact matching is interpretable, but it fails to capture synonyms, context, depth of experience or adjacent competencies. Hybrid recommender-system research describes approaches that combine multiple recommendation methods, which may address limitations of single-method matching but require their own evaluation [15].

B. Adaptive quiz and spaced-practice logic

The quiz generator reviews recent attempts for the selected topic. When there is no attempt history, it selects an easier question distribution. Higher recent accuracy (at least 80%) or a sufficient number of recent attempts selects a mastery distribution; moderate performance (at least 60%) or multiple attempts selects a balanced distribution. The prompt includes recent question text to discourage duplication, and the server grades submitted answers so the active quiz need not expose the correct option. The debrief identifies missed questions and explanations.

The browser-side spaced-practice helper schedules the next review after one day when recent accuracy is below 80% and after two days when accuracy is at least 80%. This is a deterministic heuristic that is easy to explain and test. It should not be described as an optimized memory model: an evaluation would need delayed recall tests over multiple sessions, a baseline schedule and an agreed measure of retained knowledge [7].

C. Roadmap generation

Roadmap generation expects a target role and uses the current career profile to determine the skills already represented and the gaps that remain. The AI service is asked for structured learning nodes with order, priority, prerequisite, estimated duration, checklist items, resources and practice material. A filtering step removes roadmap titles that exactly duplicate a normalized skill already held by the user. The remaining path is stored so the user can mark node status and return to progress later. Exact title matching can miss semantic duplicates, so future versions should use a controlled skill taxonomy and explicit aliases.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

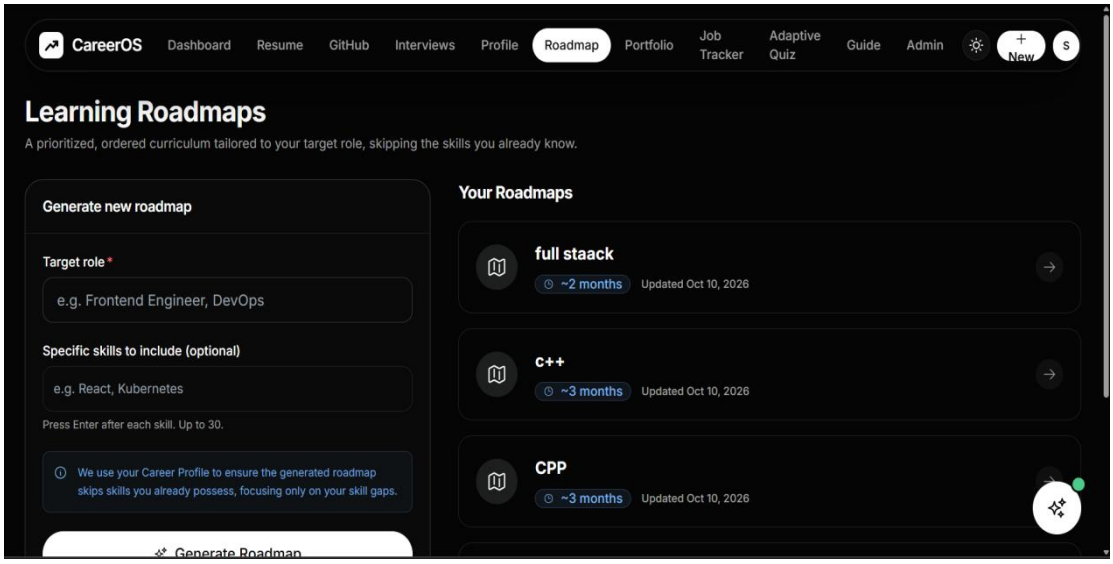


Fig. 3. Roadmap interface for choosing a target role and generating a skill-based learning path.

VI. SECURITY AND RESPONSIBLE USE

The implementation uses password hashing, short-lived access tokens and rotating opaque refresh tokens stored as hashes on the backend; the refresh token is delivered through an HTTP-only cookie while the client keeps the access token in memory. Request handling includes protected routes, administrative role checks, resource-ownership checks, Zod validation, Helmet, a configured CORS policy and rate limiting. External credentials should be stored in environment variables rather than committed to source files. These design choices reduce common risks but do not prove that a deployed system is free from vulnerabilities; a dedicated security assessment is still required [12].

Career data can include contact details, education, work history, resume contents, interview recordings and generated assessments. Users should understand when content is transmitted to third-party AI or storage services, how long it is retained, and how they can correct or delete inaccurate records. Model-generated scores should be framed as coaching suggestions and should not be the sole basis for hiring, admissions or other high-impact decisions. Responsible deployment also requires testing for accessibility, privacy, fairness and different input styles [8].

Operational safeguards

A pre-release checklist should confirm cross-account isolation, refresh-token rotation, safe error responses, upload validation, API rate limits, HTTPS cookie settings, deletion/retention behavior and provider-failure handling. Logs should capture service health and error categories without recording complete resumes, recorded audio, secrets or access tokens. Administrator access should be restricted at the API layer, not solely by hiding a client navigation item.

VII. RESULTS AND DISCUSSION

The supplied screenshots show the breadth of the implemented interface. One dashboard snapshot displays a resume ATS-style score of 78/100 based on five analyses, three completed mock interviews, an average interview score of 49/100, 0% roadmap progress, and a published portfolio with 53 views. An administration screenshot shows 14 total users, 7 interviews, 4 roadmaps, 2 resumes, 2 GitHub analyses and 4 quizzes. These numbers are captured UI values only; they do not establish production-scale use, predictive validity or sustained engagement.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 07, ISSN No. 2455-2143, Pages 09-17
Published Online November 2026 in IJEAST (http://www.ijeast.com)

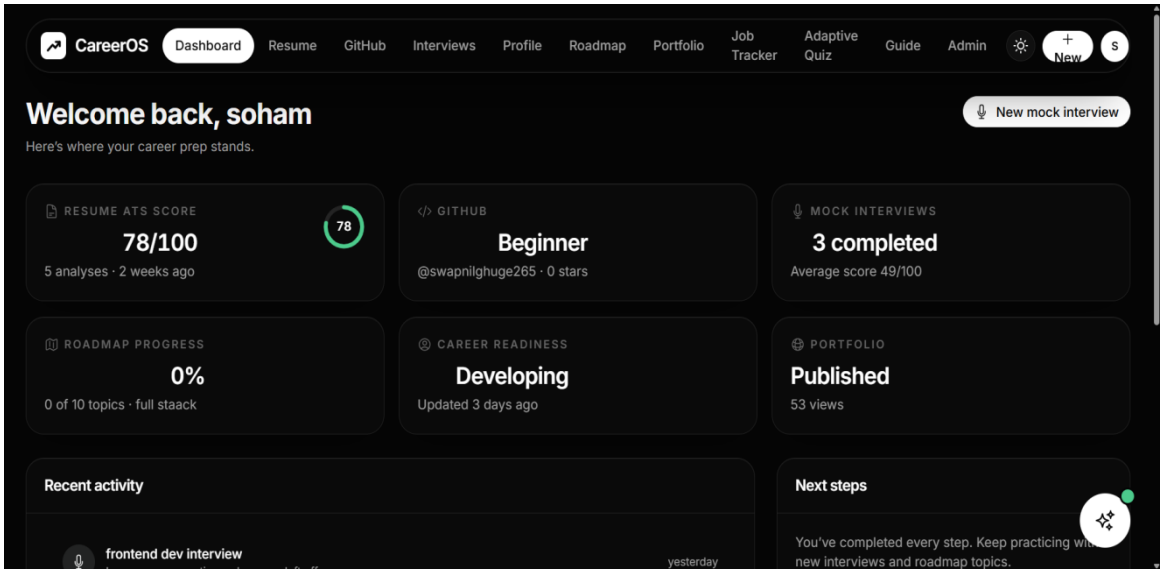


Fig. 4. Dashboard snapshot showing example module summaries and career-preparation indicators.

The job-tracker screenshot displays full-stack job cards with 100% skill-match badges and JavaScript/Node.js labels. Since matching relies on normalized skill overlap, evaluation should confirm which required skills were extracted for each listing and whether the required list is complete. The resume page shows historical analysis scores, while the interview screen demonstrates answer entry and an active recording state. Roadmap and quiz views show learning controls, but they do not demonstrate long-term knowledge retention.

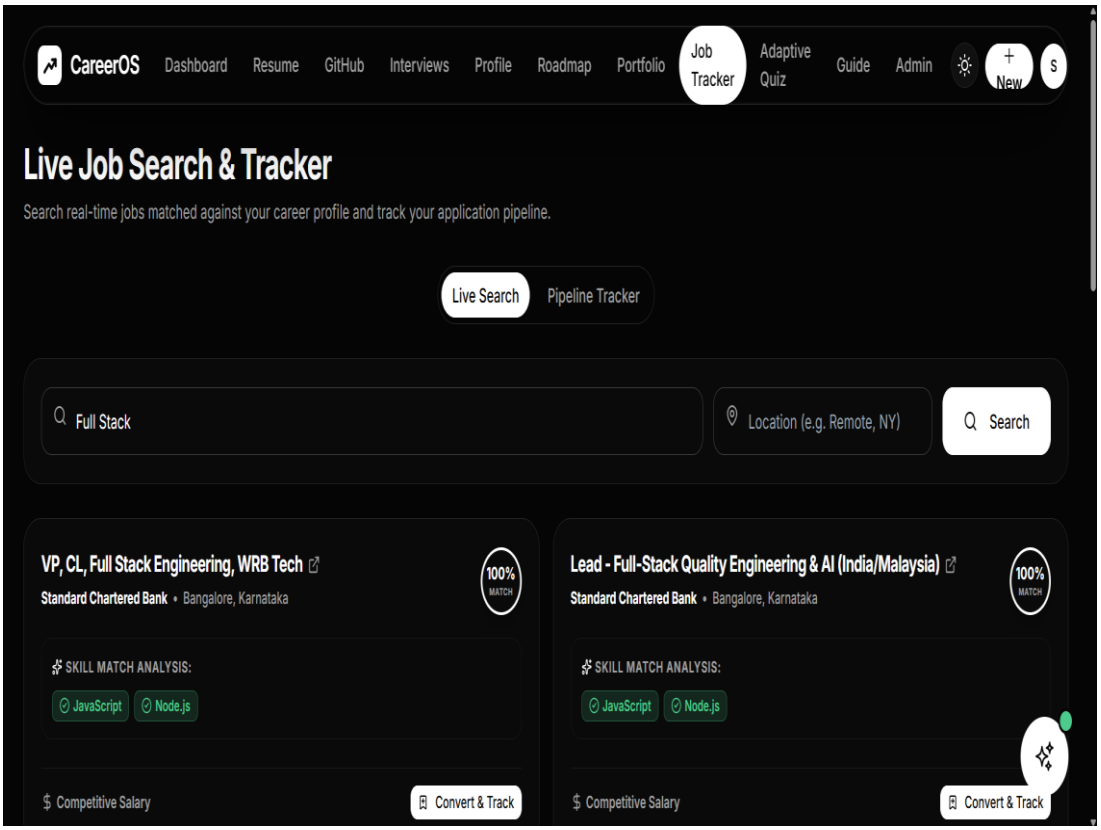


Fig. 5. Job-search screen with job cards and skill-match indicators.

TABLE III. SNAPSHOT OBSERVATIONS AND LIMITATIONS

Observed UI state	Directly visible	Not established by the screenshot
Dashboard	78/100 resume score; 3 interviews; average score 49/100.	Scoring accuracy, real-world usage or employability impact.
Admin page	Aggregate counters for users and module records.	Longitudinal adoption or complete data integrity.
Job tracker	Job cards with 100% match labels.	Freshness, relevance or vacancy availability.

The strongest result is architectural: several career-preparation activities are accessible through one interface and are linked by user-specific career data. The key research gap is empirical. Resume feedback could be reviewed by qualified human raters; job relevance could be assessed using a labelled vacancy set and metrics such as precision@k, recall@k and normalized discounted cumulative gain; and interview scores could be compared across independent raters. Response latency, provider failures, accessibility and data isolation should also be measured under a documented test setup.

VIII. LIMITATIONS AND FUTURE WORK

The available evidence does not include a benchmark dataset, a controlled user study, measured latency under load or independently verified outcome data. AI services and job-search APIs can be unavailable or rate-limited. The job matcher is not semantic and does not account for proficiency. Generated roadmaps and interview evaluations can be inaccurate or inconsistent. Browser speech-recognition support varies, and the Edge TTS integration depends on a provider interface that can change. The spaced-review rule is heuristic, with no observed long-term retention result available.

Priority future work is to run the test suite in continuous integration; develop a consented evaluation set; compare resume feedback with expert ratings; measure job-match relevance; test cross-account security and token lifecycle; add skill aliases and proficiency evidence; expose job-data freshness; provide consent, deletion and export controls; and evaluate mobile accessibility. A before-and-after learner study could examine whether the integrated workflow helps students maintain a more consistent preparation routine, provided that the study design defines participants, tasks, outcome measures, consent and analysis in advance.

IX. CONCLUSION

CareerOS brings resume analysis, GitHub evidence, mock interviews, learning roadmaps, quizzes, portfolio publishing and job-application tracking into a common career-preparation workspace. Its main contribution is the integration of these workflows through a shared profile and modular API/data design. The evidence presented here documents implementation structure and screenshot-visible states; it does not establish accuracy, improved employability or production-level reliability. Human review, labelled evaluation data, privacy controls and systematic software testing are necessary next steps.

ACKNOWLEDGEMENT OF AI ASSISTANCE

ChatGPT, an AI language model developed by OpenAI, was used as an assistance tool during the preparation of this paper for language refinement, organization of ideas, and drafting support. The authors reviewed and edited the generated material and remain responsible for the paper's content, interpretations, and conclusions.

X. REFERENCES

[1]. Çelik Ertuğrul D., and Bitirim S. (2025). Job recommender systems: a systematic literature review, applications, open issues, and challenges, Journal of Big Data, DOI: 10.1186/s40537-025-01173-y.

[2]. José-García A., Sneyd A., Melro A., Ollagnier A., Tarling G., Zhang H., Stevenson M., Everson R., and Arthur R. (2023). C3-IoC: A Career Guidance System for Assessing Student Skills using Machine Learning and Network Visualisation, International Journal of Artificial Intelligence in Education, DOI: 10.1007/s40593-022-00317-y.

[3]. Vaishampayan S., Leary H., Alebachew Y. B., Hickman L., Stevenor B., Beck W., and Brown C. (2025). Human and LLM-Based Resume Matching: An Observational Study, Findings of the Association for Computational Linguistics: NAACL 2025, DOI: 10.18653/v1/2025.findings-naacl.270.

[4]. Herold M., and Roedenbeck M. R. H. (2025). AI-Driven Research in the Recruitment and Selection Process: Application of an AI Taxonomy With a Systematic Literature Review, SAGE Open, DOI: 10.1177/21582440251361746.

[5]. Hickman L., Bosch N., Ng V., Saef R., Tay L., and Woo S. E. (2022). Automated Video Interview Personality Assessments: Reliability, Validity, and Generalizability Investigations, Journal of Applied Psychology, DOI: 10.1037/apl0000695.

[6]. Hickman L., Tay L., and Woo S. E. (2025). Are Automated Video Interviews Smart Enough? Behavioral Modes, Reliability, Validity, and Bias of Machine Learning Cognitive Ability

Assessments, Journal of Applied Psychology, DOI: 10.1037/apl0001236.

[7]. Cepeda N. J., Pashler H., Vul E., Wixted J. T., and Rohrer D. (2006). Distributed Practice in Verbal Recall Tasks: A Review and Quantitative Synthesis, Psychological Bulletin, DOI: 10.1037/0033-2909.132.3.354.

[8]. Tabassi E. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, National Institute of Standards and Technology, DOI: 10.6028/NIST.AI.100-1.

[9]. React Team (n.d.). React Documentation, Available at: https://react.dev/ (accessed 23 October 2026).

[10]. MongoDB, Inc. (n.d.). MongoDB Node.js and React Application Documentation, Available at: https://www.mongodb.com/docs/drivers/node-frameworks/react/ (accessed 23 October 2026).

[11]. OpenJS Foundation (n.d.). Node.js Documentation, Available at: https://nodejs.org/docs/latest/api/ (accessed 23 October 2026).

[12]. Open Worldwide Application Security Project (OWASP) (n.d.). JSON Web Token for Java Cheat Sheet and Authentication Guidance, Available at: https://cheatsheetseries.owasp.org/ (accessed 23 October 2026).

[13]. Groq (n.d.). API Documentation, Available at: https://console.groq.com/docs (accessed 23 October 2026).

[14]. Adzuna (n.d.). Job Search API, Available at: https://developer.adzuna.com/ (accessed 23 October 2026).

[15]. Burke R. (2002). Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, DOI: 10.1023/A:1021240730564.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

- Soham Mali
- Rushikesh Waghmare
- Abhay Gaikwad
- Sachin Nanvare
- Prasad Pratape

ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H-INDEX	i10-INDEX	IJEAST CITATIONS	PEER-REVIEWED & OPEN ACCESS
33	154	9,647 (IJEAST CITATIONS)	PUBLISHED MONTHLY

PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.

AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

editor@ijeast.com
www.ijeast.com
India

2455-2143