IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

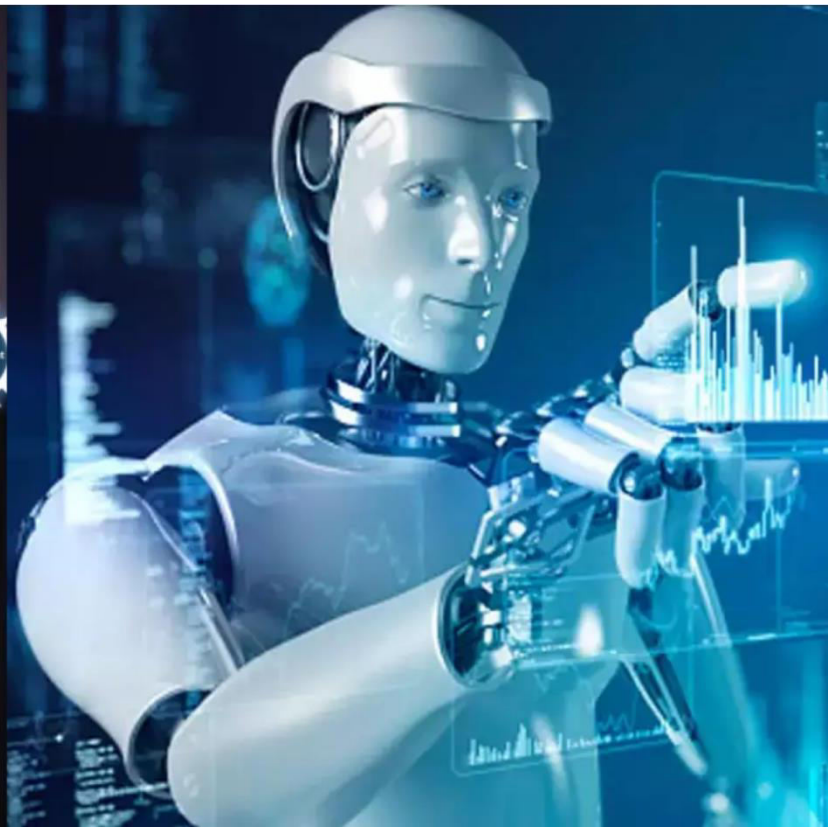
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 10, October 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

HireForge: An AI-Based Placement Readiness and Skill-Gap Analyser with Adaptive Assessment, Mock Interview Preparation and Resume Analysis

Eshwari More[1], Raj Pawar[1], Samarth Payghan[1], Aniket Basate[1], Prof.Meghraj Chougule[2]

B. Tech Final Year Student, Department of Computer Science and Engineering, MIT College of Railway Engineering & Research, Barshi, India[1]

Guide, Department of Computer Science and Engineering, MIT College of Railway Engineering & Research, Barshi, India[2]

ABSTRACT: Placement preparation in engineering colleges is fragmented: students practise aptitude on one site, rehearse interviews on another and check resumes on a third, while nobody gives them one clear picture of how ready they are for a target role. This paper presents HireForge, a web-based career readiness platform that brings an AI readiness assessment with skill-gap analysis, an assessment system, an AI-assisted mock interview module and a resume analyser into one application. The front end is written in HTML5, CSS3 and JavaScript and is deployed on Vercel; the back end is written in Python with Django and the Django REST Framework (DRF). Data is stored in SQLite in the current implementation, with MySQL planned for the integrated application, and AI-generated questions, feedback and assessments come from the Google Gemini API. For a chosen target role, the readiness module returns a score out of 100 together with strengths, skill gaps and recommendations, and the resume analyser reports overall quality, ATS readiness, detected skills, a role-specific skill gap and a learning plan. The platform was exercised end to end on its deployed instance, and representative outputs are reported.

KEYWORDS: career readiness, skill-gap analysis, placement preparation, mock interview, resume analysis, ATS, Django REST Framework, Google Gemini API, JavaScript.

I. INTRODUCTION

Campus placement is the most visible outcome of an engineering degree, and preparing for it is a long process that spans aptitude tests, data-structure practice, core-subject revision, interviews and resume writing. Institutions usually support this process with a placement cell, occasional training sessions and spreadsheets of past results. Students, however, prepare alone and across many disconnected tools.

A student rarely knows which weakness is actually lowering the chance of selection for the role they want. Existing tools address one activity at a time. Aptitude sites score quizzes, resume checkers count keywords, and interview simulators ask questions without knowing the student's profile. None of them joins these signals into a single view, and none tells the student, in plain words, what to learn next for a specific role. HireForge addresses this by placing assessment, interview practice, resume analysis and a readiness assessment inside one web application that stores its data in one database. A large language model, reached through the Google Gemini API, generates interview questions and written feedback, and the platform presents the results as scores, strengths, skill gaps and a learning plan that the student can act on. The rest of this paper describes the architecture, the technology stack, the modules, the evaluation protocol and the observed outputs.

The key objectives of this research are:

- To build one web platform that combines a readiness assessment, quizzes, mock interviews and resume analysis.
- To produce a readiness score and a ranked skill gap for a chosen target role from the student's profile.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

- To analyse resumes for overall quality, ATS readiness, contact links, sections and detected skills, and to compare them with a target role.
- To run mock interviews by role, interview type and difficulty, with typed or spoken answers and saved sessions.
- To expose the modules through REST APIs built with the Django REST Framework.
- To use open, widely available tools (HTML5, CSS3, JavaScript, Python, Django and SQLite or MySQL) so that a college can host the platform at low cost.

II. RELATED WORK

Placement-prediction studies have compared decision trees, naive Bayes, support vector machines, random forests and neural networks on institutional records, and typically report accuracy on a single split of a spreadsheet. They seldom connect the prediction to a student-facing workflow that tells the student what to do next. Resume checkers score keyword overlap and format, interview simulators present generic questions, and aptitude sites do not link results to a target role.

Large language models have changed what such tools can offer. Brown et al. [1] showed that a large model can perform new text tasks from a few examples, and the Gemini family [2] makes comparable capability available through an API, so personalised questions, feedback and recommendations can be generated without training a model. Such output is advisory and can be wrong, and Raghavan et al. [3] show that claims about fairness in algorithmic hiring need careful scrutiny, which is why HireForge presents its results as guidance for the student and not as a selection decision.
A key limitation across existing work is fragmentation: tools address one preparation activity, and none joins assessment, interview and resume signals into one role-specific view. Table 1 summarises the studies discussed above and the gap each leaves. Table 2 compares the main preparation approaches with HireForge.

TABLE 1 LITERATURE REVIEW

Ref.	Year	Methodology	Key finding	Limitation
[1]	2020	Large language model, few-shot learning	New text tasks learned from a few examples	Output can be incorrect; needs checking
[2]	2023	Gemini multimodal model family	General-purpose LLM available through an API	Depends on an external service
[3]	2020	Review of algorithmic hiring and bias	Fairness claims need careful scrutiny	Evaluates vendors; builds no system
[R-x]	[year]	[Placement prediction study to be added]	[finding]	[limitation]
[R-y]	[year]	[Resume analysis study to be added]	[finding]	[limitation]
[R-z]	[year]	[Interview simulation study to be added]	[finding]	[limitation]

TABLE 2 COMPARISON OF PREPARATION APPROACHES

Approach	Readiness analysis	Assessment	Interview	Resume	Unified data
Placement-cell spreadsheets	No	No	No	No	No
Standalone aptitude sites	No	Yes	No	No	No
Standalone resume checkers	No	No	No	Yes	No
Generic mock-interview tools	No	No	Yes	No	No
Offline placement-prediction models	Yes	No	No	No	No
HireForge	Yes	Yes	Yes	Yes	Yes

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

III. SYSTEM ARCHITECTURE

HireForge follows a layered architecture in which a browser front end talks to a Django back end through REST APIs (Fig. 1). It consists of four layers:

1. Client Tier (HTML5, CSS3, JavaScript):
Web pages are written in HTML5, styled with CSS3 for responsive layouts and made interactive with JavaScript [5]. The pages call the back end with JavaScript requests and show the dashboard, quizzes, interview sessions and resume results. The browser's Web Speech API [10] can capture spoken interview answers when it is enabled.

2. Application Tier (Django):
Django [4] handles requests, authentication and business logic. The Django REST Framework exposes the REST APIs, particularly for assessments, and returns JSON to the client.

3. Data Tier (SQLite, MySQL):
SQLite [6] stores the data of the current implementation. MySQL [7] is planned as the shared database of the integrated application, and Django's object-relational mapper allows the switch without rewriting the models.

4. Intelligence Services (Google Gemini API):
The Google Gemini API [8], called only from the back end, generates interview questions, evaluates answers, writes resume feedback and produces the readiness assessment, the skill-gap analysis and the career recommendations.

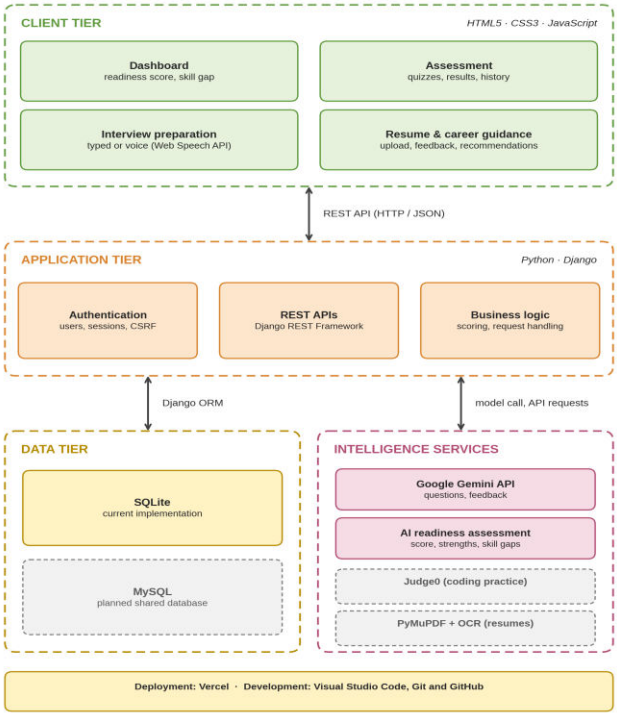


Fig. 1 Layered architecture of HireForge

Two planned components are shown as dashed boxes in Fig. 1: Judge0 for executing coding-practice submissions, and PyMuPDF with OCR for extracting text from PDF and scanned resumes. They are not part of the evaluated system. A design rule runs through the architecture: the large language model writes feedback and assessments, its output is advisory, and the student stays in control of every action.

The functional behaviour of the system is captured by the use-case diagram in Fig. 2. Two actors use the system. A student registers, maintains a profile, takes quizzes, runs the AI readiness assessment, practises mock interviews and

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

uploads a resume for analysis. An administrator logs in to manage users and the question bank and to view platform analytics.

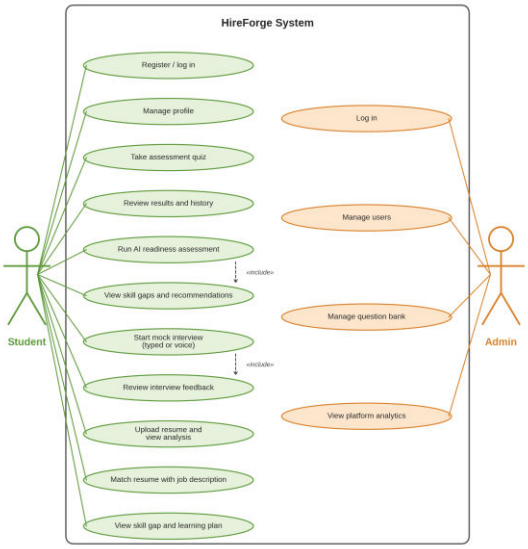


Fig. 2 Use-case diagram for the two roles

IV. TECHNOLOGY STACK

Table 3 lists the technologies used in HireForge, what each is used for, and whether it is part of the current implementation or planned.

TABLE 3 TECHNOLOGY STACK

Layer	Technology	Purpose	Status
Front end	HTML5	Structure of web pages	Used
Styling	CSS3	Page design and responsive layouts	Used
Client-side	JavaScript	User interactions and front-end functionality	Used
Back end	Python	Server-side logic and data processing	Used
Web framework	Django	Application and request handling	Used
API	Django REST Framework	REST APIs, particularly for assessments	Used
Database	SQLite	Local data storage in the current implementation	Used (development)
Database	MySQL	Shared database for the integrated application	Planned
AI	Google Gemini API	Interview questions, answer evaluation, resume feedback, readiness assessment, career recommendations	Used
Speech	Web Speech API	Browser speech recognition for spoken answers	Optional
Code execution	Judge0	Running coding-practice submissions	Planned
Resume text	PyMuPDF, OCR	Text extraction from PDF and scanned resumes	Planned
Tools	Visual Studio Code	Code development and debugging	Used
Version control	Git and GitHub	Source control and collaboration	Used
Deployment	Vercel	Hosting the web application	Used

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

V. SYSTEM MODULES

The proposed system is composed of the following modules, which together provide assessment, interview practice, resume analysis and career guidance.

5.1 Landing Page, Profile and Dashboard
The landing page introduces the platform and leads the student to sign up. After sign-in, the navigation bar gives access to the Dashboard, Assessments,

Mock Interview and Resume pages. The student keeps a profile with skills, projects and a target role, which the other modules use when they build their requests.

5.2 Assessment Module
Questions are held in a question bank with category (Programming, Data Structures, Aptitude and Core Subjects), difficulty and explanation. The REST API starts a quiz, validates answers on the server and returns the score, percentage and section-wise results. Quiz history and section performance are stored so that the student can track improvement.

5.3 Interview Preparation Module
The set-up screen asks for a target job role (which may be left blank when the profile already supplies one), an interview type such as technical or project-based, and a difficulty level. The student can start a new interview, load saved interviews, and open a previous interview to continue reviewing it. Answers are typed or, when the browser supports it, spoken through the Web Speech API. The Django REST API passes each request to the Google Gemini API, which generates role-based questions and evaluates the answers, and the student receives feedback on their performance. Each saved interview lists its role, type and status, for example in progress.

5.4 Resume Analyser
The student uploads a resume, optionally pastes a job description and presses a button to match the resume with the job. The analysis page reports an overall quality score, an ATS readiness score and the word count, followed by strengths and prioritised improvements marked high or medium. It checks that contact links (email, phone, LinkedIn and GitHub) and the standard sections (summary or objective, education, skills, experience, projects and certifications) are present, and it lists the detected skills grouped into categories such as databases, data and AI, CS fundamentals, tools and cloud, web development, professional skills and programming languages. For the chosen role it shows the missing core skills and the nice-to-have skills, and it ends with a learning plan that gives one study task for each missing skill. Text extraction with PyMuPDF and OCR for scanned resumes is planned.

5.5 Placement and Career Guidance
The AI readiness assessment is the module that gives the platform its placement-prediction function. For a selected target role, the student presses a button to predict and analyse skill gaps. The back end builds a request from the student's profile and sends it to the Gemini API, and the page shows a readiness score out of 100, a short summary of the student's current preparation, strengths, skill gaps and practical recommendations. A saved assessment is stored with its date and is reloaded the next time the student opens the page, and a link leads back to the profile so that the student can improve it and run the assessment again.

5.6 Data Model and Security
All modules write to one database through Django models, with migrations owned by the module that creates each table. Records are kept for student profiles, quiz attempts and answers, interview sessions and answers, resumes with their analysis results, and saved readiness assessments. Django authentication protects every page and API, CSRF protection and input validation are enabled, and the Gemini API key is stored on the server in an environment variable, never in browser code.

VI. METHODOLOGY AND EVALUATION PROTOCOL

HireForge was developed by a team of four in an iterative plan, in which each member owned complete features (front end, API, database and logic) rather than a single layer, so that a working application existed early. Code was written in

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

Visual Studio Code and managed with Git and GitHub using a branch for each member and pull requests for merging. The web application was deployed on Vercel. The system was evaluated on its deployed instance. Each module was exercised end to end through its pages and APIs, and the outputs of the AI features were checked by hand for one test profile and one resume.

VII. RESULTS AND DISCUSSION

7.1 Functional Evaluation

The functional flows covered registration and login, profile update, quiz start to finish, interview set-up and question generation, typed and spoken answers, saved-interview loading, resume upload and analysis, job matching and the AI readiness assessment. Representative test cases are in Table 4.

TABLE 4 REPRESENTATIVE TEST CASES

ID	Test case	Expected result	Status
TC-01	Register and log in	Session created; dashboard opens	[Pass/Fail]
TC-02	Complete a quiz	Score and section scores shown	[Pass/Fail]
TC-03	Start an interview with role, type and difficulty	Questions generated through the Gemini API	[Pass/Fail]
TC-04	Submit a typed answer	Feedback returned and saved	[Pass/Fail]
TC-05	Submit a spoken answer	Speech converted to text, then evaluated	[Pass/Fail]
TC-06	Load saved interviews	Earlier sessions listed and can be reopened	[Pass/Fail]
TC-07	Upload a resume	Scores, strengths, improvements and skills shown	[Pass/Fail]
TC-08	Match a resume with a job description	Match result and missing skills shown	[Pass/Fail]
TC-09	Run the AI readiness assessment	Score out of 100, strengths and skill gaps shown	[Pass/Fail]
TC-10	Open an API without logging in	Access denied	[Pass/Fail]

7.2 Observed Outputs

Table 5 reports the outputs observed on the deployed instance for one test profile and one resume. They show what the student sees and are not a measure of accuracy.

TABLE 5 OBSERVED OUTPUTS FOR ONE TEST PROFILE

Module	Measure	Observed value
Resume analyser (target: Data Scientist / ML Engineer)	Overall quality	92
Resume analyser	ATS readiness	99
Resume analyser	Word count	280
Resume analyser	Skills detected	16
Resume analyser	Contact links found	4 of 4 (email, phone, LinkedIn, GitHub)
Resume analyser	Standard sections found	6 of 6
Resume analyser	Missing core skills	4 (Statistics, Pandas, NumPy, Scikit-learn)
Resume analyser	Nice-to-have skills missing	6 (Deep Learning, TensorFlow, PyTorch, NLP, Data Visualization, Docker)
AI readiness assessment (target: Data Scientist)	Readiness score	45 / 100

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

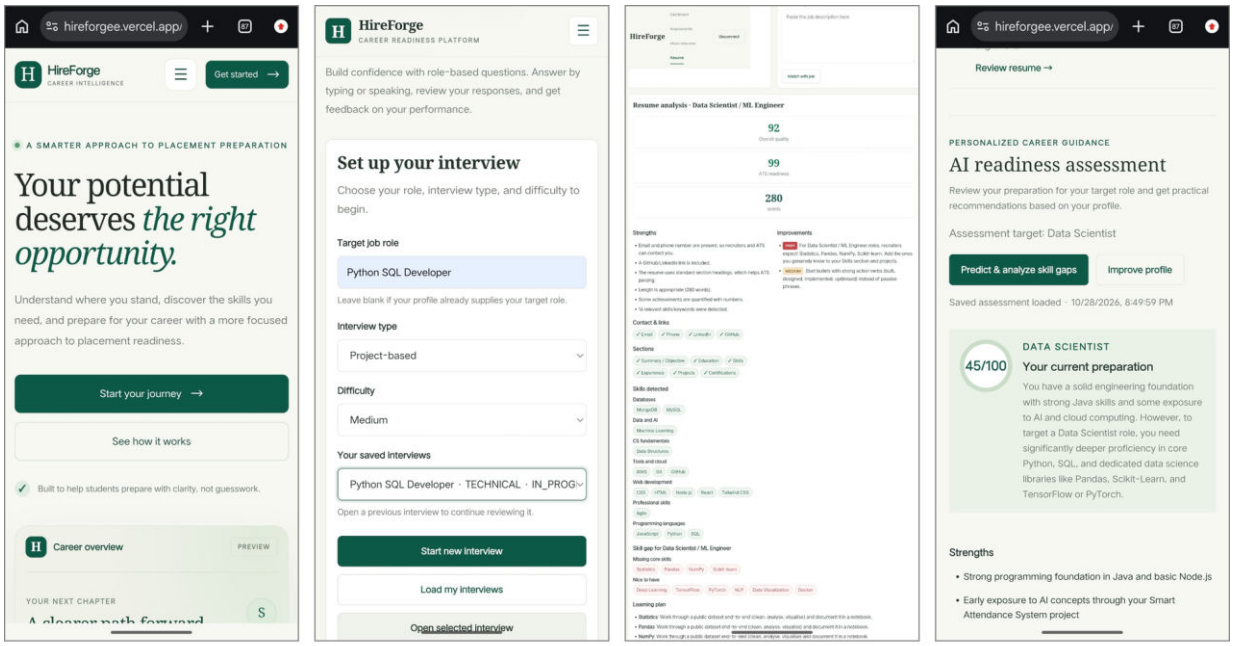
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

7.3 Application Screenshots

Fig. 3 shows four pages of the deployed application on a mobile browser: the landing page, the interview set-up screen, the resume analysis page and the AI readiness assessment.



(a) Landing page (b) Interview set-up (c) Resume analysis (d) AI readiness assessment

Fig. 3 Screenshots of the HireForge web application: (a) landing page, (b) interview set-up, (c) resume analysis, (d) AI readiness assessment

7.4 Discussion

The observed outputs show that the platform turns a profile and a resume into role-specific, actionable guidance: the resume analysis names the four core skills that are missing for a data-science role and gives a study task for each, while the readiness assessment summarises the gap as a score out of 100 with strengths and recommendations. Keeping every result tied to a target role and a next step is what separates HireForge from tools that only score a document. Table 6 maps each objective to its outcome.

TABLE 6 OBJECTIVES AND ACHIEVED OUTCOMES

Objective	Outcome
Unified platform	Assessment, interview, resume and readiness modules in one Django application
Readiness score and skill gap	Score out of 100, strengths and skill gaps for a chosen role
Resume analysis	Quality and ATS readiness scores, skills, role-specific gap and learning plan
Mock interviews	Role, type and difficulty selection; typed or spoken answers; saved sessions
REST APIs	Django REST Framework APIs used by the browser front end
Low-cost, open stack	HTML5, CSS3, JavaScript, Python, Django, SQLite / MySQL, Vercel

7.5 Limitations

The scores and assessments are produced with the help of a large language model and have not been validated against actual placement outcomes, so they should be read as guidance. The observed outputs come from one test profile and one resume, and the platform has not yet been evaluated with a larger group of students. The AI features depend on an external service, so they need an internet connection and are subject to API limits, and resume content is sent to that

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 10, October 2026

|DOI: 10.15680/IJIRSET.2026.1510107|

service. SQLite suits development, and the planned MySQL migration has not been evaluated. Judge0 and PyMuPDF with OCR are planned and not part of the evaluated system.

VIII. CONCLUSION AND FUTURE WORK

In this paper, HireForge has been presented as an integrated career readiness platform built with HTML5, CSS3, JavaScript, Python, Django and the Django REST Framework. It combines a quiz-based assessment system, AI-assisted mock interviews, a resume analyser and an AI readiness assessment that returns a score, strengths and a ranked skill gap for a target role, using the Google Gemini API for its AI features. The observed outputs on the deployed instance show that the platform turns a student's profile and resume into role-specific guidance.

Future work includes migrating the shared database from SQLite to MySQL, adding a coding-practice module that executes submissions with Judge0, extracting text from PDF and scanned resumes with PyMuPDF and OCR, evaluating the platform with a larger group of students, and comparing the readiness scores with real placement outcomes.

The main contributions of this work are:

- A single Django platform that joins assessment, interview, resume and readiness data around a target role.
- A resume analyser that reports quality and ATS readiness, checks contacts and sections, groups detected skills and produces a role-specific gap and learning plan.
- AI-assisted mock interviews with role, type and difficulty selection, typed or spoken answers and saved sessions.
- A REST API design with the Django REST Framework that serves a plain HTML5, CSS3 and JavaScript front end.

IX. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, MIT College of Railway Engineering & Research, Barshi, for its guidance and support throughout this project.

REFERENCES

[1] T. B. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.
[2] Gemini Team, Google, “Gemini: A Family of Highly Capable Multimodal Models,” arXiv:2312.11805, 2023.
[3] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, “Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices,” in Proc. ACM FAT*, 2020, pp. 469–481.
[4] Django Software Foundation, “Django and Django REST Framework Documentation.” [Online]. Available: https://docs.djangoproject.com
[5] Mozilla, “MDN Web Docs: HTML, CSS and JavaScript.” [Online]. Available: https://developer.mozilla.org
[6] SQLite Consortium, “SQLite Documentation.” [Online]. Available: https://www.sqlite.org/docs.html
[7] Oracle, “MySQL Reference Manual.” [Online]. Available: https://dev.mysql.com/doc
[8] Google, “Gemini API Documentation.” [Online]. Available: https://ai.google.dev
[9] Vercel, “Vercel Documentation.” [Online]. Available: https://vercel.com/docs
[10] Mozilla, “Web Speech API,” MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
[11] [Add placement-prediction and resume/interview studies from your own literature search, in IEEE format.]



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details