CODE4CAREER: AI-Integrated E-Learning Platform for Coding, Placement, And Interview Training using Google Gemini API

Author Details

Satya Prem[1], Shivam Singh Tomar[2], Pooja Basal[3]
[12]Department of Computer Science & Engineering,
Shri Ram Institute of Technology, RGPV, Jabalpur, Madhya Pradesh, India
[4]Professor, Department of Computer Science & Engineering,
Shri Ram Institute of Technology, RGPV,Jabalpur, Madhya Pradesh, India

Corresponding Author: **Vansh Soni**

Email: vansh16tanay@gmail.com

https://doi.org/10.55041/ijsmt.v2i5.329

Cite this Article: Prem, S., Tomar, S. S. & Basal, P. (2026). CODE4CAREER: AI-Integrated E-Learning Platform for Coding, Placement, And Interview Training using Google Gemini API. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.329

Abstract—
The rapid advancement of large language models (LLMs) offers transformative potential for personalized education, yet existing e-learning platforms for coding and placement training remain largely static and generic. This paper presents an AI-integrated platform that leverages the Google Gemini API to deliver adaptive coding challenges with intelligent hints, automated code evaluation, realistic mock interviews with dynamic feedback, resume parsing and job matching, and personalized course recommendations. Built on a modern full-stack architecture (React, Node.js, MongoDB, Google OAuth), the system maintains a longitudinal learner profile that tracks progress across all modules. A pilot study with 30 final-year computer science students over four weeks demonstrated significant improvements: coding scores increased by 16.4 points, interview scores rose by 2.4 points, and resume ATS scores improved by 23%. The paper details prompt engineering strategies, system architecture, implementation challenges (hallucination, over-reliance, privacy), and mitigation techniques. This work provides a replicable blueprint for integrating state-of-the-art LLMs into vocational training, transforming one-size-fits-all preparation into adaptive, learner-centric skill development.

Keywords— Large language models; Google Gemini API; coding education; placement training; mock interview; personalized learning; resume parsing; intelligent tutoring systems.

1. INTRODUCTION

The demand for skilled software developers continues to outpace supply, yet traditional training methods – online courses, static coding platforms, and generic placement preparation – fail to address individual learning needs. Students face several persistent challenges: coding exercises that do not adapt to their skill level, lack of real-time feedback on code quality and logic, inadequate interview preparation with unrealistic questions, manual and subjective resume screening, and disconnected tools that treat coding, interview prep, and placement guidance as siloed activities [1], [2].

Large Language Models (LLMs) like Google Gemini offer a promising solution. Gemini's multimodal and reasoning capabilities have been benchmarked against GPT-4, showing competitive performance in natural language understanding and generation [3]. Recent systematic reviews confirm that LLMs enhance academic performance, engagement, and skill development when integrated into educational tools [4]. Specifically, Thelwall

[5] found that Gemini 1.5 Flash achieves a Spearman correlation of 0.645 with human scores for evaluation tasks, validating its use for automated assessment in educational contexts.

Despite these advances, few systems integrate end-to-end placement training – combining coding practice, mock interviews, resume analysis, and course recommendation – under a single AI-powered platform. Existing commercial platforms (LeetCode, HackerRank, InterviewBit) rely on static content or rule-based evaluation, lacking adaptive personalization and longitudinal tracking.

This paper presents a novel AI-integrated e-learning platform that uses the Google Gemini API as its core AI engine to address these gaps. The platform provides four integrated modules: (1) adaptive coding practice with intelligent hints and automated code evaluation, (2) role-specific mock interview generation with real-time feedback, (3) resume parsing and job matching with ATS scoring, and (4) personalized course recommendations based on skill gaps. The system maintains a persistent user profile via Google OAuth and MongoDB, enabling longitudinal tracking of skill development. We describe the architecture, prompt engineering strategies, implementation details, pilot study results, and challenges encountered.

The remainder of this paper is organized as follows: Section II reviews related work. Section III details the methodology and system architecture. Section IV presents results from the pilot study. Section V concludes with implications and future directions.

2. LITERATURE REVIEW

Several research streams inform this work: LLM applications in education, automated code evaluation, AI-driven interview systems, and resume parsing.

LLMs in Education. Shi et al. [4] systematically reviewed 88 empirical studies on LLMs in education (November 2022 – March 2025). They identified six primary application categories: Chatbots, Learning Content Generation, Automated Assessment and Feedback, Task Support Tools, Learning Support Tools, and Intelligent Tutoring Systems (ITS). ITS was the most prominent, with reported improvements in academic performance, motivation, and cognitive skill development. However, the review noted challenges such as student over-reliance on LLM outputs, hallucination, assessment fairness, and privacy concerns.

Gemini Performance in Evaluation. Thelwall [5] compared Google Gemini 1.5 Flash with ChatGPT 4o-mini on research quality evaluation. Gemini achieved a higher correlation with human scores when processing full-document PDFs. This supports the use of Gemini for automated assessment in educational contexts.

Automated Code Evaluation. Cambaz and Zhang [6] reviewed AI-driven code generation models in programming education. They found that LLM-based systems can provide immediate feedback on correctness, efficiency, and coding style, though challenges remain in handling complex logic and preventing solution regurgitation. Our platform extends this by using Gemini for code evaluation and progressive hint generation.

AI Mock Interview Systems. The AutomatedRecruitment project [7] demonstrated Gemini for resume parsing and interview assessment. Another platform [8] used Gemini for adaptive question generation and feedback, improving student confidence. Our platform integrates mock interviews with coding and placement modules under a unified learner profile.

Resume Parsing and Job Matching. Deepa et al. [9] proposed M-LPERM, achieving high resume parsing accuracy. Traditional methods rely on TF-IDF and keyword matching, which often fail to capture semantic meaning. Our platform uses Gemini for context-aware skill extraction and job matching, supported by TF-IDF baseline scoring.

Research Gap. Existing systems mainly focus on isolated functionalities such as coding practice, mock interviews, or resume analysis. Few provide an integrated platform combining coding practice, interview preparation, resume analysis, and course recommendations with persistent learner tracking. Our work addresses this gap by developing a modular, AI-driven platform powered by Gemini across all components..

4. METHODOLOGY

A. System Architecture

The platform follows a client-server model with three core layers:

Frontend: React 18 with Vite, Tailwind CSS, Axios, and @react-oauth/google for Google Login.

Backend: Node.js with Express, exposing REST APIs. Authentication uses Google OAuth 2.0, and user data is stored in MongoDB.

AI Engine: Google Gemini API (Gemini 1.5 Flash for standard tasks and Gemini 1.5 Pro for complex evaluations). API calls are asynchronous and include retry logic.

Fig. 1 illustrates the architecture.

[React Frontend] ↔ [Node.js Backend] ↔ [MongoDB]
↕ ↕
[Gemini API] [Google OAuth]

Fig. 1. System architecture showing frontend, backend, database, and AI integration.

B. Module Design

Four integrated modules were implemented:

1. Coding Practice Module

Adaptive problem generation: Gemini generates coding problems based on the user's skill level and previously practiced topics.

Code evaluation: Submitted code is evaluated using a rubric covering correctness, efficiency, readability, and edge cases. Gemini returns a score with strengths and weaknesses.

Hint generation: Progressive hints are generated without revealing the full solution.

2. Mock Interview Module

Question generation: Gemini generates technical, behavioral, and system design questions based on the selected job role.

Answer evaluation: Responses are evaluated on relevance, clarity, and depth with actionable feedback.

3. Resume & Job Matching Module

Resume parsing: Resume text is extracted from uploaded PDFs and structured into fields such as skills, education, and experience.

Job matching: Skills are matched with job descriptions using TF-IDF cosine similarity and re-ranked using Gemini for final scoring.

4. Personalized Course Recommendation Module

Skill gap analysis: Coding performance and resume skills are analyzed to identify weak areas and recommend relevant online courses.

All user interactions are stored in MongoDB, enabling longitudinal skill tracking and personalized dashboards.

C. Prompt Engineering

Chain-of-thought prompting and few-shot examples were used to improve Gemini's output reliability. For critical tasks such as code evaluation, JSON output schemas and validation rules were enforced.

Example prompt for code evaluation:

System: You are an expert software engineering interviewer. Evaluate the solution strictly using the rubric.

User: Problem: {problem_description}
Solution: {user_code}
Rubric: correctness (40%), efficiency (30%), readability (20%), edge cases (10%).

Return JSON:

```
{
  "score": int,
  "strengths": list,
  "weaknesses": list,
  "improved_code": str
}
```

D. Data Collection and Pilot Study

A pilot study was conducted with 30 final-year computer science students over four weeks. Participants completed coding sessions, mock interviews, and resume analyses using the platform.

Pre- and post-intervention measures included:

Coding scores

Interview scores

Resume ATS scores

User satisfaction was collected using a 5-point Likert scale survey. Ethical approval and informed consent were obtained, and all collected data were anonymized before analysis.

E. Technical Implementation

The backend was developed using Node.js v20 and Express, while MongoDB Atlas was used for database hosting. The Gemini API was accessed through the @google/generative-ai SDK with rate limiting support.

Google OAuth 2.0 was configured with authorized origins:

http://localhost:5173

http://localhost:3000

The frontend was built using Vite and deployed locally. The source code is available in a public repository (link omitted for blind review).

5. RESULTS AND DISCUSSION

A. Quantitative Outcomes

Table I summarizes the pre- and post-intervention metrics for the 30 participants.

Table I: Performance Improvement After 4 Weeks

Metric	**Pre-intervention**	**Post-intervention**	**Improvement**
Average coding score (0–100)	58.2	74.6	+16.4
Average interview score (0–10)	4.8	7.2	+2.4
Average time to solve medium problem (minutes)	22	14	-36%
Resume ATS score (%)	45	68	+23%

A paired t-test showed that all improvements were statistically significant ($p < 0.01$). The large effect size (Cohen's $d = 1.24$ for coding score) indicates strong practical impact.

Fig. 2 illustrates the distribution of coding score improvements. Most participants showed measurable improvement, with the highest gain observed among students who initially struggled with recursion-based problems.

B. User Feedback

Survey responses collected using a 5-point Likert scale are summarized in Table II.

Table II: User Satisfaction Survey

Statement	**Mean Score (1–5)**
"The AI hints helped me learn without giving away answers."	4.6
"Mock interview questions were realistic and feedback useful."	4.4
"Resume gap analysis showed me exactly what skills to learn."	4.8
"The platform was easy to use and responsive."	4.5
"I would recommend this platform to peers."	4.7

Participants reported that the hint system improved conceptual understanding and that resume analysis provided clear guidance on missing skills and project requirements.

C. Technical Performance

Average Gemini API response times were:

Coding evaluation: 1.8 seconds

Interview generation: 2.4 seconds

Resume parsing: 3.1 seconds

System uptime during the pilot study was 99.2%. Average operational cost was approximately $0.02 per user session using Gemini 1.5 Flash, increasing during complex evaluations requiring Gemini 1.5 Pro.

D. Comparison with Existing Platforms

The proposed platform improves upon existing systems in several ways:

LeetCode/HackerRank: Provide coding practice but lack adaptive AI-generated hints, interview preparation, and resume analysis.

InterviewBit: Includes interview preparation but relies on pre-defined questions without AI-based evaluation.

Standalone AI tutors: Offer conversational assistance but do not provide longitudinal learner tracking or integrated placement workflows.

The proposed system combines coding practice, mock interviews, resume analysis, and personalized recommendations within a unified learner profile.

E. Challenges and Mitigation

Three major challenges were identified during implementation:

Hallucination: Gemini occasionally generated incorrect feedback or invalid test cases. This was mitigated using chain-of-thought prompting, validation rules, and fallback rule-based checks.

Over-reliance on AI: Some users depended heavily on generated solutions. To address this, an "explain in your own words" step was added before hints were displayed.

Privacy Concerns: User resumes and interview answers involved sensitive information. Mitigation strategies included anonymization, explicit consent collection, local transcript storage, and minimal Gemini data retention settings.

6. CONCLUSION

This paper presented an AI-integrated e-learning platform that leverages the Google Gemini API to transform coding, placement, and interview training. Unlike static systems, our platform provides adaptive coding challenges with intelligent hints, realistic mock interviews with dynamic feedback, resume parsing and job matching, and personalized course recommendations – all under a single user profile with longitudinal skill tracking. A four-week pilot study with 30 students demonstrated significant improvements in coding scores (+16.4 points), interview scores (+2.4 points), and resume ATS scores (+23%), with high user satisfaction.

The platform offers a replicable blueprint for integrating state-of-the-art LLMs into vocational training. Future work will extend the system with speech-based interview evaluation (using Google Speech-to-Text), multimodal coding (diagram-to-code using Gemini Pro Vision), institutional deployment with teacher dashboards, and a larger longitudinal study measuring actual placement success rates. As LLMs continue to evolve, such AI-integrated platforms will become essential for scalable, equitable, and effective career preparation in the digital economy.

ACKNOWLEDGMENT

The authors sincerely thank all 30 student participants who contributed to the pilot study and provided valuable feedback throughout the development and evaluation phases of the platform. Their active participation played a significant role in improving the system's usability and effectiveness.

The authors would also like to express their heartfelt gratitude to **Prof. Pooja Basal Mam** for her continuous guidance, encouragement, and academic support during the course of this work. Her valuable suggestions and mentorship greatly contributed to the successful completion of the project.

We also acknowledge Google for providing access to the Gemini API through its educational research support program, which enabled the implementation of the AI-driven features used in this platform.

REFERENCES

[1] P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A machine learning approach for automation of resume recommendation system," *Procedia Computer Science*, vol. 167, pp. 2318–2327, 2020.

[2] S. Maheshwary and H. Misra, "Matching resumes to jobs via deep Siamese network," in *Companion Proceedings of the Web Conference 2018*, 2018, pp. 87–88.

[3] E. Y. Wang et al., "Gemini in reasoning: Unveiling commonsense in multimodal LLMs," *arXiv preprint arXiv:2312.17661*, 2023.

[4] Y. Shi, K. Yu, Y. Dong, and F. Chen, "Large language models in education: A systematic review of empirical applications, benefits, and challenges," *Computers and Education: Artificial Intelligence*, 2025.

[5] M. Thelwall, "Is Google Gemini better than ChatGPT at evaluating research quality?," *Journal of Data and Information Science*, vol. 10, no. 2, pp. 1–5, 2025.

[6] D. Cambaz and X. Zhang, "Use of AI-driven code generation models in teaching and learning programming: A systematic literature review," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education (SIGCSE 2024)*, 2024, pp. 172–178.

[7] AutomatedRecruitment Team, "Gemini API for end-to-end hiring," Google AI Developer Competition, 2024. [Online].Available: https://ai.google.dev/competition/projects/automatedrecruitment

[8] "AI-based job interview preparation platform using MERN stack and Gemini API," *International Journal of Scientific Research in Engineering and Development (IJSRED)*, vol. 8, no. 6, 2025. [Online]. Available: https://zenodo.org/records/14965297

[9] R. Deepa et al., "Survey on resume parsing models for JOBCONNECT+: Enhancing recruitment efficiency using natural language processing and machine learning," *International Journal of Computational and Experimental Science and Engineering*, vol. 10, no. 1, 2024.