International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

Gen AI Job Preparation Web Application: An Intelligent Full-Stack Platform for Personalized Career Readiness and AI-Driven Interview Preparation

Mr. Amit Mohapatra
Dept. of CSE
GIFT Autonomous
Bhubaneswar, Odisha, India

Mr. Laxmikanta Mohanta
Dept. of CSE
GIFT Autonomous
Bhubaneswar, Odisha, India

Lecturer Allupati Chakradhar Patro
Dept. of CSE
GIFT Autonomous
Bhubaneswar, Odisha, India

Abstract—The rapid advancement of digital technologies has significantly transformed the education and career development sectors by increasing the demand for intelligent, scalable, and user-friendly platforms. Traditional job preparation methods and existing online platforms often operate independently, suffering from fragmentation, generic content, and a lack of real-time personalized feedback. This research presents the comprehensive design, development, and implementation of the Gen AI Job Preparation Web Application, an AI-integrated full-stack platform that bridges this gap by providing a unified ecosystem for AI-powered mock interviews, resume analysis, skill assessments, and performance tracking. Engineered using modern frontend technologies including React.js, TypeScript, and Tailwind CSS, alongside a resilient backend powered by Node.js, Express.js, and MongoDB, the system establishes a centralized digital career preparation environment. Key functionalities include OTP-based secure user authentication governed by JSON Web Tokens (JWT), intelligent resume evaluation through the Gemini AI API, adaptive aptitude and technical assessments, and granular performance analytics. Experimental analysis demonstrates significant improvements in personalized interview preparation and user engagement, with stable performance across all implemented modules. The platform successfully resolves the fragmentation of existing systems, providing a secure, intelligent, and multifunctional environment suitable for students, fresh graduates, and professionals seeking career advancement.

Keywords— Generative AI, Job Preparation Platform, MERN Stack, React.js, Gemini API, OTP Authentication, JWT, Role-Based Access Control, Resume Analysis, Mock Interviews, Full-Stack Web Development, Adaptive Learning, Performance Analytics.

I. INTRODUCTION

A. Background and Context

In today's highly competitive employment landscape, effective preparation for job interviews, aptitude assessments, and resume screening is critical for students and professionals alike. Traditional preparation methods, including static textbooks, coaching classes, and pre-recorded video content, fail to deliver personalized, adaptive guidance or real-time feedback. With the rapid advancement of Generative AI, intelligent systems can now offer dynamic, interactive, and highly personalized preparation support tailored to individual user needs.

Generative AI-powered platforms are fundamentally transforming the way users prepare for careers by enabling adaptive learning experiences, AI-generated interview simulations, and intelligent resume evaluation. These systems continuously analyze user performance to identify strengths and weaknesses, delivering targeted recommendations for improvement. As technology evolves, integrating Generative AI into job preparation platforms has emerged as a highly innovative solution for bridging the gap between academic learning and industry requirements.

B. Problem Statement

Existing job preparation platforms face significant architectural, operational, and personalization limitations. Most systems offer generic question banks, static study materials, and limited AI integration. Users frequently lack access to real-time feedback on mock interviews, intelligent resume analysis, or detailed performance analytics. The fragmentation between preparation tools—requiring users to navigate multiple disconnected applications for different preparation needs—creates inefficiencies and degrades the overall learning experience.

A critical deficiency in many contemporary platforms is their inability to adapt to the user's individual skill

Received: 20-04-2026 | Accepted: 29-05-2026 | Published: 04-06-2026

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

level and preparation history. Furthermore, the management of dynamic AI-generated content alongside secure user data requires a modern, scalable architecture capable of handling complex requests efficiently. There is, therefore, a distinct and pressing need for a unified, full-stack application that natively integrates AI-powered preparation tools, secure authentication, and comprehensive performance tracking within a single, cohesive platform.

C. Research Objectives

The major objectives driving the development of the proposed Gen AI Job Preparation Web Application are:

- Architectural Unification: To design and deploy a centralized digital ecosystem utilizing a modern, decoupled MERN-style tech stack (React.js, TypeScript, Tailwind CSS for the client; Node.js, Express.js, MongoDB for the server).
- AI-Powered Interview Preparation: To develop a dynamic mock interview module leveraging the Gemini AI API to generate personalized, domain-specific technical and HR interview questions with real-time feedback.
- Intelligent Resume Analysis: To implement an AI-driven resume evaluation engine capable of parsing uploaded resumes and providing structured feedback on content quality, ATS compatibility, and skill alignment.
- Advanced Identity Management: To implement a cryptographically secure authentication module featuring OTP-based user registration, secure password recovery pipelines, and strict Role-Based Access Control (RBAC) via stateless JWT middleware.
- Comprehensive Performance Analytics: To develop a persistent analytical tracking system that maintains detailed user performance records, generates progress reports, and delivers personalized learning recommendations.
- Future-Ready Extensibility: To structure the backend data models and routing controllers to natively support future integrations, including voice-based AI interviews, multilingual support, and advanced machine learning analytics.

D. Scope of the Project

The Gen AI Job Preparation Web Application is designed to serve a diverse demographic including students, fresh graduates, working professionals seeking career transitions, educational institutions, and placement training centers. The platform provides distinctly tailored interfaces for three primary user tiers: Students/Users, Administrators, and System-level operations. The architecture is engineered for horizontal scalability, ensuring stable performance as the concurrent user base grows, while adhering to modern web security standards for secure data processing and privacy protection.

II. LITERATURE REVIEW

A. Evolution of Digital Job Preparation Systems

Job preparation has evolved significantly from traditional classroom coaching and printed study materials to online learning platforms and digital assessment tools. Early systems provided static question banks and recorded lectures, while modern platforms now offer adaptive learning, live assessments, and AI-driven recommendations. The integration of Generative AI has further enhanced personalization and automation in job preparation, enabling platforms to dynamically generate content, evaluate user responses, and recommend improvement strategies based on individual performance data.

B. Role of Artificial Intelligence and Machine Learning

Artificial Intelligence and Machine Learning play a crucial role in modern job preparation systems. AI is used for generating personalized interview questions, evaluating candidate responses, analyzing resumes, recommending learning content, and tracking performance trajectories. Machine Learning algorithms identify user weaknesses and adapt content accordingly. Recent literature highlights the immense potential of integrating large language models (LLMs) such as GPT-based systems and Google's Gemini into educational platforms, enabling conversational assessment, intelligent feedback generation, and personalized study plan creation based on collaborative and content-based filtering techniques.

C. Web Application Architecture and Security

Modern full-stack web application development frameworks, specifically the MERN stack (MongoDB, Express.js, React, Node.js) integrated with TypeScript for compile-time type safety, provide the ideal foundational architecture for building highly scalable, cohesive digital ecosystems. Standard session-based

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

authentication scales poorly under high concurrent user loads. This limitation has been largely superseded by stateless JSON Web Tokens (JWT), which enable horizontally scalable, distributed role verification without database lookups. The incorporation of One-Time Password (OTP) verification pipelines significantly mitigates unauthorized access by ensuring user registration and account recovery workflows are cryptographically secured through verified communication channels.

D. Limitations of Existing Systems

Several job preparation platforms such as LinkedIn Learning, LeetCode, and InterviewBit provide technical and aptitude preparation content. However, most existing systems offer limited personalization, lack real-time AI-generated interview feedback, and do not provide an integrated end-to-end preparation experience within a single platform. Current systems often focus only on coding or aptitude preparation without offering intelligent resume analysis, adaptive mock interviews, or comprehensive performance dashboards. There remains a clear need for an integrated AI-powered system that provides personalized, interactive, and complete job preparation support.

III. SYSTEM OVERVIEW AND ARCHITECTURE

A. Comprehensive Proposed System

The Gen AI Job Preparation Web Application functions as a highly centralized, intelligently governed career preparation platform. The application seamlessly integrates responsive frontend interfaces, a highly secure RESTful API backend, Generative AI-driven interview and resume modules, cryptographically secured OTP authentication, and a comprehensive performance analytics engine. By unifying these disparate preparation tools into a single full-stack repository, the platform resolves the operational friction, data redundancy, and personalization gaps typical of fragmented existing systems.

Users can securely register via an email OTP verification loop, access AI-generated mock interview sessions, upload resumes for intelligent evaluation, attempt adaptive aptitude and technical assessments, and track their preparation progress through a personalized analytics dashboard. The platform simultaneously provisions an administrative tier equipped with comprehensive management tools for overseeing user accounts, question banks, AI configurations, and system analytics.

B. Multi-Tier System Architecture

The proposed system follows a multi-layer web application architecture consisting of four primary tiers: the Presentation Layer, Application Layer, Database Layer, and AI Integration Layer. This separation of concerns ensures highly maintainable, scalable, and secure data flows from the database through the backend to the client interface.

Presentation Layer (Frontend): Constructed using React.js and TypeScript, scaffolded with Vite for near-instantaneous build times and optimized production bundles. The interface leverages Tailwind CSS for rapid, utility-first, mobile-responsive design. Key frontend components include the User Dashboard, Mock Interview Interface, Resume Upload Module, Quiz and Assessment Pages, and JWT-based session management. The application is designed as a Single Page Application (SPA) utilizing React Router for seamless client-side navigation.

Application Layer (Backend): Built on the asynchronous, event-driven Node.js runtime utilizing the Express.js framework. This layer acts as the secure gateway for all system operations, implementing RESTful API routing, JWT-based authentication middleware, OTP generation and verification logic, role-based access control, and asynchronous communication with the Gemini AI API for interview generation and resume analysis.

Database Layer: Employs MongoDB, a highly scalable NoSQL document database, to store user profiles, interview histories, assessment scores, resume data, and progress reports. Mongoose ODM enforces strict schema-based definitions, manages relational lookups, and maintains database integrity while preserving the flexibility required for dynamic AI-generated data structures.

AI Integration Layer: Powered by the Google Gemini AI API, this layer processes user inputs including parsed resume text, quiz scores, and interview responses to generate structured, actionable intelligence. The backend dynamically handles asynchronous AI request-response cycles, parsing complex AI payloads into user-friendly interview questions, resume feedback, and personalized learning recommendations.

C. Key Functional Modules

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

The platform is meticulously partitioned into several autonomous, yet highly communicative modules:

User Module: Manages all user-facing activities including secure registration and login, profile management, dashboard access, and interaction with AI preparation tools. Users can participate in mock interviews, attempt assessments, upload resumes, and view personalized progress reports.

Admin Module: Enables administrators to manage users, monitor platform performance, maintain question banks, configure AI settings, and generate comprehensive system reports. Provides centralized control over the entire application.

Interview Preparation Module: Provides AI-generated mock interview questions and evaluates user responses with personalized feedback. Simulates realistic HR, technical, and aptitude interview scenarios based on user-selected domain and skill level.

Assessment Module: Conducts aptitude tests, technical quizzes, and coding assessments. Implements automated score calculation, timer functionality, and performance reporting to identify strengths and weak areas.

Resume Analysis Module: Analyzes uploaded resumes using the Gemini AI API, evaluating structure, formatting, keywords, technical skills, and ATS compatibility to provide improvement suggestions.

Authentication Module: Handles secure login, registration, OTP generation and verification, password encryption via bcrypt, JWT session management, and Role-Based Access Control.

IV. METHODOLOGY AND ALGORITHMIC DESIGN

A. OTP-Based Authentication Workflow

To ensure uncompromising platform security, user registration and all critical account recovery actions are strictly guarded by an OTP verification pipeline. This methodology prevents unauthorized account creation and guarantees that the user possesses the claimed email address.

Algorithm 1: OTP Generation, Hashing, and Verification

Require: User Email (E), User Password (P_raw)

- Backend receives registration payload (E, P_raw) and validates syntax via schema validator.
- Database is queried; if User exists with E, return 409 Conflict.
- Hash Password: P_hash ← bcrypt(P_raw, saltRounds=10).
- Generate secure 6-digit OTP: O ← crypto.randomInt(100000, 999999).
- Hash OTP for storage: O_hash ← SHA-256(O). Store with TTL index of 600 seconds.
- Dispatch raw O to E via SMTP Transport (Nodemailer).
- User submits (E, O_input) to /api/auth/verify endpoint.
- If document not found or TTL expired, return 400 Bad Request.
- Compare Hashes: If SHA-256(O_input) ≠ O_hash, return 401 Unauthorized.
- Verification Success: Create User Document, generate JWT, return 200 OK with token.

B. JWT Generation and Stateless Session Management

The system utilizes JWTs for stateless session management. A JWT consists of three parts: Header, Payload, and Signature. The cryptographic signature is generated using the HMAC-SHA256 algorithm as follows:

Signature = HMAC-SHA256(base64UrlEncode(Header) + "." + base64UrlEncode(Payload), SecretKey)

This cryptographic signature guarantees that the token has not been tampered with in transit. If a client attempts to elevate their role from 'user' to 'admin' in the base64 payload, the signature verification on the server immediately fails and the request is rejected, ensuring strict enforcement of Role-Based Access Control across all API endpoints.

C. AI Integration and Resume Analysis Methodology

Resume analysis is performed through a multi-stage pipeline. When a user uploads a resume, the backend middleware (Multer) intercepts the multipart file upload and buffers the document in memory. A text extraction engine (pdf-parse or equivalent) converts the uploaded PDF or DOCX to a clean text string. This extracted text is then encapsulated within a structured prompt and transmitted to the Gemini AI API via an asynchronous HTTP request. The AI model processes the input and returns a structured JSON payload containing categorized feedback covering content

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

quality, ATS compatibility, keyword optimization, and skill gap analysis. The backend parses this payload and stores the results in MongoDB linked to the user's profile for persistent tracking and future reference.

D. Mock Interview Generation Pipeline

The interview preparation module leverages the Gemini AI API to dynamically generate domain-specific and role-specific interview questions. When a user initiates a mock interview session, the frontend transmits the selected domain (e.g., Software Engineering, HR Behavioral, Aptitude), difficulty level, and any previously answered questions to the backend. The application layer constructs a structured prompt incorporating this contextual information and dispatches it to the Gemini API asynchronously. The returned questions are parsed from the AI response payload, formatted, and transmitted to the React frontend for display. User responses are evaluated by a secondary AI call for scoring and feedback generation, with results stored in the Interview History collection in MongoDB.

V. SYSTEM DESIGN AND DATA MODELING

A. Database Schema and Entity Relationships

The MongoDB environment is strictly structured via Mongoose schemas to ensure rapid retrieval of user data, AI-generated content, and performance analytics while maintaining referential integrity across collections.

User Schema: Stores registered user information including full name, email address, encrypted password (bcrypt hash, select: false), role enumerators (Admin/User), OTP verification status boolean (isVerified), and references to associated interview and assessment records.

Interview Schema: Defines individual mock interview sessions. Contains user reference (ObjectId ref 'User'), selected domain, AI-generated question arrays, user response text, AI-generated feedback strings, calculated performance scores (Float), and session timestamps.

Resume Schema: Stores uploaded resume metadata including user reference, file storage path, extracted text content, and the AI-generated structured analysis result (JSON document) containing ATS score, keyword recommendations, and content improvement suggestions.

Assessment Schema: Records quiz and aptitude test sessions. Contains user reference, test type (Technical/Aptitude/Coding), question arrays, user answers, calculated marks (Float), accuracy metrics, and date_taken timestamp.

Progress Schema: Acts as a transactional ledger mapping User IDs to longitudinal performance data across interviews, assessments, and resume evaluations. Tracks improvement trends and generates aggregated analytics for dashboard visualization.

B. RESTful API Architecture

The backend implements a strict RESTful routing design following the MVC architectural pattern. The core API endpoints governing the platform are presented in Table I.

TABLE I

CORE RESTFUL API ENDPOINT DEFINITIONS

Method	Endpoint Route	Functionality / Access
POST	/api/auth/register	Initiates OTP workflow. (Public)
POST	/api/auth/verify	Validates OTP, issues JWT. (Public)
GET	/api/interview/generate	Generates AI interview questions. (Protected)
POST	/api/resume/analyze	Uploads and analyzes resume via AI. (Protected)
GET	/api/assessment/start	Retrieves assessment questions. (Protected)
POST	/api/assessment/submit	Submits answers, returns score. (Protected)
GET	/api/dashboard/progress	Returns user performance analytics. (Protected)
PUT	/api/users/profile	Updates user profile information. (Protected)

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

C. Data Flow and Security Interceptor Pattern

The system's data flow is secured via an Axios Interceptor pattern on the frontend and custom middleware on the backend. When a user requests a protected resource: the React frontend's Axios instance intercepts the outbound HTTP request and attaches the JWT as a Bearer token in the Authorization header. The Express API routes the request through the authMiddleware, which extracts and verifies the cryptographic signature against the server's private secret (process.env.JWT_SECRET). Upon successful verification, the decoded payload (User ID and Role) is attached to the request object, and the controller fetches the requested data via Mongoose queries. Invalid or expired tokens immediately trigger a 401 Unauthorized response, redirecting users to the authentication interface.

VI. IMPLEMENTATION DETAILS

A. Frontend Development Strategy

The frontend architecture was rigorously developed using React.js and TypeScript. TypeScript interfaces enforce strict API contract adherence, providing immediate IDE feedback when accessing undefined payload properties and significantly reducing debugging time in large-scale component trees. Vite was selected as the primary build tool for its native ES module support during development, offering instantaneous Hot Module Replacement (HMR) without losing component state. For production, Vite employs Rollup for aggressive tree-shaking, producing highly optimized, chunked static assets.

UI components are styled using Tailwind CSS utility classes applied directly within JSX, ensuring a consistent, modular design system that adapts responsively across mobile, tablet, and desktop environments. State management employs React Context API and custom hooks to manage complex application states including authentication status, AI response loading states, assessment timers, and real-time form validation for OTP inputs.

B. Backend Integration and Middleware Pipeline

The Node.js and Express backend enforces a strict MVC architectural pattern with a robust middleware pipeline:

- Helmet.js: Sets critical HTTP headers protecting against XSS and Clickjacking vulnerabilities.
- CORS Middleware: Configured to strictly accept requests only from the verified frontend domain.
- Express.json(): Parses incoming JSON payloads with a strict size limit to prevent DoS attacks.
- Rate Limiting: Custom middleware using express-rate-limit restricts IPs to prevent brute-force attacks on OTP generation and login routes.
- express-mongo-sanitize: Strips prohibited characters from request bodies to neutralize NoSQL injection vectors.

C. AI API Integration

The Gemini AI API is integrated at the application layer through a dedicated service module. All AI requests are managed as asynchronous operations using JavaScript Promises, preventing blocking of the Node.js event loop. Structured prompts are constructed dynamically based on user context (domain, skill level, resume text) to maximize AI output relevance and accuracy. API response payloads are validated, parsed, and transformed into standardized internal data structures before storage or transmission to the frontend, ensuring consistency in AI-generated content presentation.

VII. TESTING AND VALIDATION

A. Testing Strategy

Comprehensive testing was conducted across all implemented modules using a structured, multi-stage testing strategy to verify correctness, security, performance, and integration of all system components.

Unit Testing: Individual modules including authentication, AI interview generation, resume analysis, and assessment scoring were tested independently. Examples include testing user login functionality with valid and invalid credentials, verifying OTP expiry logic, and validating quiz score calculation accuracy.

Integration Testing: Frontend-backend communication, backend-database connectivity, and AI API integration were verified. This confirmed smooth data flow across system layers, including correct transmission of resume text to the Gemini API and accurate storage of AI-generated feedback in MongoDB.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

System Testing: End-to-end testing of complete user workflows including the full interview session lifecycle (login → domain selection → question generation → response submission → feedback display) confirmed overall system reliability.

Security Testing: Simulated brute-force attacks on the login endpoint verified rate-limiting effectiveness. IDOR testing confirmed that users cannot access other users' resumes or interview records through URL manipulation. JWT tampering tests verified that signature verification correctly rejects modified tokens.

B. Test Case Summary

TABLE II

SYSTEM TEST CASE RESULTS

C. Tools Used

Browser Developer Tools provided real-time DOM inspection and network monitoring for frontend debugging. Postman served as the primary backend validation utility for isolated testing of RESTful API endpoints. MongoDB Compass provided visual inspection and query analysis of database collections. Console debugging tools across both React.js and Node.js runtime environments enabled tracing of asynchronous execution flows and variable states during active development sessions.

VIII. RESULTS AND PERFORMANCE ANALYSIS

A. System Performance Evaluation

Comprehensive testing of the Gen AI Job Preparation Web Application demonstrated stable and efficient performance across all implemented modules. The React.js frontend maintained responsive, sub-second rendering times for critical user interfaces including the main navigation, interactive assessment modules, and administrative control panels. The application layer's asynchronous architecture prevented event-loop blocking during concurrent data transactions.

TABLE III

SYSTEM PERFORMANCE METRICS

Metric	Result
AI Interview Question Generation	< 2 seconds average response time
Resume Analysis Processing	Successfully handled PDF and DOCX uploads
Authentication (JWT)	Secure login with < 100ms token validation
Dashboard Load Time	Sub-second rendering under normal load
Concurrent User Handling	Stable performance under test load conditions
OTP Verification Pipeline	100% delivery rate during simulated tests

B. Key Results

Test Case ID	Module	Test Scenario	Expected Result	Status
TC01	Login	Valid User Login	User redirected to dashboard	Pass
TC02	Registration	New User Signup	Account created successfully	Pass
TC03	Resume Upload	Upload Resume File	Resume stored and analyzed	Pass
TC04	Interview Module	Generate AI Questions	Questions displayed successfully	Pass
TC05	Assessment	Submit Quiz	Score calculated and saved	Pass
TC06	Dashboard	View Progress Report	Analytics displayed correctly	Pass

The AI-powered mock interview generation module successfully produced contextually relevant and domain-specific questions across technical, HR, and aptitude categories. Resume analysis delivered structured, actionable feedback covering ATS compatibility, keyword optimization, and content quality with satisfactory accuracy. The JWT validation middleware proved highly efficient, adding minimal latency overhead to protected route requests. Secure OTP delivery achieved a high dispatch success rate during simulated load testing without dropped SMTP connections or TTL failures. Performance tracking dashboards loaded comprehensive historical analytics

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

efficiently, enabling users to identify preparation trends and target specific weak areas.

C. Comparison with Existing Systems

Table IV presents a comparative analysis of the proposed system against existing job preparation platforms, demonstrating significant improvements across critical evaluation dimensions.

TABLE IV

COMPARISON: PROPOSED SYSTEM VS. EXISTING SYSTEMS

Feature	Existing System	Proposed System
Interview Preparation	Static questions and materials	AI-generated dynamic mock interviews
Feedback Mechanism	Limited or manual feedback	Instant AI-based personalized feedback
Resume Analysis	Basic or unavailable	Intelligent AI resume evaluation
Personalization	Generic learning content	Adaptive personalized recommendations
Performance Tracking	Limited progress monitoring	Detailed analytics and reports
Accessibility	Limited functionality	Accessible anytime via web browser
AI Integration	Minimal or absent	Advanced Generative AI (Gemini API)
User Interaction	Less interactive	Interactive, user-friendly interface
Scalability	Limited feature expansion	Modular, easily scalable architecture

IX. CONCLUSION

This paper presented the comprehensive design, development, and evaluation of the Gen AI Job Preparation Web Application, an intelligent full-stack platform that addresses the significant limitations of existing job preparation systems through the integration of Generative AI technologies with a modern, scalable web architecture.

The platform successfully unified AI-powered mock interview generation, intelligent resume analysis, adaptive skill assessments, secure OTP-based authentication, and detailed performance analytics within a single, cohesive system. The MERN stack architecture, augmented with the Gemini AI API, enabled dynamic content generation, real-time feedback delivery, and persistent user progress tracking. Experimental results demonstrated stable performance, secure authentication workflows, and satisfactory AI output quality across all implemented modules.

The system effectively resolves the fragmentation and personalization deficiencies characteristic of existing platforms, delivering an accessible, cost-effective, and intelligent career preparation experience available 24/7 through a responsive web interface. The modular architecture provides a scalable foundation for future enhancements including voice-based AI mock interviews, multilingual support, mobile application development, company-specific preparation modules, job portal integration, and advanced machine learning-driven performance analytics. The Gen AI Job Preparation Web Application establishes a strong foundational platform for the next generation of AI-powered career development ecosystems.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Hoboken, NJ: Pearson, 2020.

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[3] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008, 2017.

[4] Google Cloud, "Gemini API Overview," Google Developer Documentation, 2025. [Online]. Available: https://ai.google.dev/gemini-api/docs. [Accessed: May 17, 2026].

[5] T. B. Brown et al., "Language models are few-shot learners," in Proceedings of the 34th International Conference on Neural Information Processing Systems, pp. 1877–1901, 2020.

[6] E. Kasneci et al., "ChatGPT for good? On the potential of large language models for education," Learning and Individual Differences, vol. 103, p. 102274, Apr. 2023.

[7] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020.

International Journal of
AI Electronics and Nexus Energy
Peer Reviewed, Referred & Indexed Journal
ISSN: 3070-0515 www.zesterapublications.com Original Research Paper

[8] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, Node.js in Action, 2nd ed. Shelter Island, NY: Manning Publications, 2017.

[9] K. Chodorow, MongoDB: The Definitive Guide, 3rd ed. Sebastopol, CA: O'Reilly Media, 2019.

[10] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, May 2015. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7519.

[11] OWASP Foundation, "OWASP Top 10:2021," OWASP, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/.

[12] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. Boston, MA: Pearson, 2020.

[13] I. Sommerville, Software Engineering, 10th ed. Boston, MA: Pearson, 2015.

[14] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY: McGraw-Hill, 2020.

[15] Mozilla Developer Network (MDN), "MDN Web Docs," MDN, 2025. [Online]. Available: https://developer.mozilla.org. [Accessed: May 17, 2026].