AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

Smart Resume Builder: An Intelligent Full-Stack Web Application for Automated Professional Resume Generation Using Artificial Intelligence

Ankita Nayak,
Department of CSE
GIFT Autonomous, Bhubaneswar,
Sushree Sanjibani Dhal
Department of CSE
GIFT Autonomous, Bhubaneswar,
Dr. Pratyush Ranjan Mohapatra,
Associate Dean Academics,CSE
GIFT Autonomous, Bhubaneswar,

Abstract

— In today's highly competitive job market, a well-structured and professionally formatted resume is one of the most critical tools for career success. However, a large proportion of job seekers — particularly students and fresh graduates — face significant challenges in creating effective resumes due to limited knowledge of professional formatting standards, document design, and industry-specific presentation requirements. Traditional manual resume creation using word processors such as Microsoft Word is time-consuming, technically demanding, and often produces inconsistent and poorly structured outputs. Existing online resume builder platforms, while helpful, suffer from critical limitations including restricted free access, limited template customization, complex user interfaces, absence of intelligent writing assistance, and inadequate optimization for Applicant Tracking Systems (ATS). This paper presents the Smart Resume Builder, a comprehensive, full-stack intelligent web application that automates and simplifies professional resume creation. The system is built upon a modern technology stack comprising React, Vite, Redux Toolkit, Tailwind CSS, and Radix UI for the frontend, with Node.js and Express.js powering the backend. MongoDB is used as the primary database through Mongoose ORM, and Clerk combined with JSON Web Tokens (JWT) manages secure user authentication and session management. A key innovation of the platform is the integration of Google Generative AI, which provides intelligent content suggestions including professional summaries, skill recommendations, and experience descriptions tailored to individual user profiles. The system provides a structured, form-based interface for entering all resume sections — personal details, educational background, technical skills, work experience, projects, and certifications — and automatically generates a well-formatted, ATS-optimized resume. Users can preview the formatted resume in real-time and download it as a professional PDF document. Functional testing across twelve test cases confirmed reliable performance across all modules. The system successfully addresses the major limitations of existing resume builders and provides a free, accessible, intelligent, and user-friendly solution for modern career development. Future enhancements include multiple template designs, cloud-based resume storage, job portal integration, AI-driven resume scoring, and interview preparation modules.

Keywords — Smart Resume Builder, Automated Resume Generation, Google Generative AI, React, Node.js, MongoDB, JWT Authentication, ATS Optimization, Clerk Authentication, Full-Stack Web Application, Career Development, Natural Language Processing

I.. INTRODUCTION

The recruitment process in modern organizations is heavily dependent on resumes as the primary tool for evaluating candidates. A resume is a formal document that provides an organized overview of an individual's educational qualifications, professional skills, work experience, project achievements, and certifications. It functions as the candidate's first impression and plays a decisive role in determining whether they will be shortlisted for interviews. Given this critical importance, the quality, structure, and presentation of a resume can significantly impact a job seeker's career prospects [1].

Despite this, a large proportion of job seekers — particularly students, fresh graduates, and individuals entering the workforce for the first time — face significant difficulties in preparing professional resumes. Many individuals lack sufficient knowledge of proper resume formatting conventions, professional document design, and industry-specific presentation standards. Traditional methods of resume creation using word processing tools such as Microsoft Word require users to manually handle layout design, alignment, font selection, spacing, and content organization. This process is time-consuming, technically demanding, and frequently results in poorly structured resumes that fail to effectively highlight a candidate's qualifications and capabilities [2].

The rapid advancement of web technologies has led to the development of online resume builder platforms that

partially address these challenges by providing predefined templates and automated formatting. However, a thorough analysis of existing platforms reveals that they continue to exhibit critical limitations. Many platforms restrict access to essential features such as PDF download and advanced templates behind paid subscription plans, creating financial barriers for students and unemployed job seekers. Furthermore, most existing tools provide limited template customization, complex user interfaces unsuitable for beginners, and almost no intelligent assistance for writing professional resume content [3].

An equally important concern is the compatibility of resumes with Applicant Tracking Systems (ATS), which are increasingly used by companies to automatically screen and filter job applications before human review. Resumes that are poorly structured, lack relevant keywords, or use non-standard formatting are automatically rejected by ATS software, regardless of the candidate's actual qualifications. Most traditional resume builders do not address this requirement, leaving users at a significant disadvantage during the application process [4].

To overcome these challenges comprehensively, the Smart Resume Builder has been developed as a full-stack intelligent web application. The system integrates modern frontend and backend technologies with Google Generative AI to provide a free, accessible, customizable, and AI-powered platform for professional resume creation. The system guides users through a structured resume creation process, provides AI-generated content suggestions, generates ATS-optimized resumes, and enables users to preview and download their resumes as professional PDF documents [5].

This paper is organized as follows: Section II reviews existing resume building systems and identifies their key limitations. Section III describes the proposed Smart Resume Builder system and its core modules. Section IV covers the development methodology and technology stack. Section V presents the system design and implementation details. Section VI discusses testing results and performance analysis. Section VII outlines future enhancement possibilities. Section VIII concludes the paper.

II.. LITERATURE REVIEW AND RELATED WORK

The literature review involves a systematic analysis of existing resume creation systems, tools, and technologies to understand their functionalities, strengths, and limitations. This analysis provides the foundation for identifying gaps that the proposed Smart Resume Builder aims to address through improved design, technology integration, and intelligent features.

A. Evolution of Resume Building Systems

Resume creation has undergone significant transformation over the past few decades. In traditional environments, resumes were created manually using word processors such as Microsoft Word or Google Docs. This method demanded considerable time and design skill, particularly for users unfamiliar with professional document formatting. The process involved manually adjusting fonts, margins, alignment, spacing, and section structures, frequently resulting in inconsistent and poorly organized outputs [6].

The emergence of web-based resume builder platforms in the early 2000s introduced a more streamlined approach. These platforms allowed users to enter their information into digital forms and automatically applied predefined templates to generate formatted resumes. Early platforms such as Resume.io, Canva Resume Builder, and Zety gained significant popularity by offering multiple template options, drag-and-drop editing capabilities, and PDF export functionality. These tools successfully automated the formatting process, reducing the burden on users with limited design experience [7].

More recently, the integration of artificial intelligence and natural language processing technologies has introduced a new generation of smart resume building applications. Modern AI-powered systems can analyze user information, generate professional content suggestions, evaluate resume quality scores, and recommend improvements based on target job descriptions. These advancements represent a significant improvement over earlier tools, though many platforms still fail to make these features fully accessible to all users without financial barriers [8].

B. Analysis of Existing Online Resume Platforms

Several popular online resume builder platforms are currently available, each offering different feature sets and user experiences. Resume.io and Zety provide professionally designed templates and real-time editing, but require premium subscriptions for PDF downloads. Canva Resume Builder offers visually attractive designs but is primarily design-focused rather than career content-focused. LinkedIn's resume builder leverages existing profile data but produces standardized outputs with limited customization. Novoresume includes resume scoring

features but again restricts advanced capabilities to paid users [9].

ChatGPT and similar general-purpose AI tools have been increasingly used for generating resume content and professional summaries. While effective for content generation, these tools lack integration with structured resume formatting, template rendering, and direct PDF export capabilities. Users must manually transfer AI-generated content into separate document editors, creating an inefficient and fragmented workflow. The Smart Resume Builder addresses this gap by combining structured resume creation with integrated AI content generation in a single unified platform [10].

C. Limitations of Existing Systems

Based on a comprehensive analysis of existing platforms, the following critical limitations have been identified. First, restricted free access remains a major barrier. Most platforms require paid subscriptions for PDF download or access to advanced templates, making professional resume creation financially inaccessible for students and unemployed individuals. Second, limited customization prevents users from personalizing resume layouts, fonts, and section structures according to their specific industry and career requirements [11].

Third, complex user interfaces discourage beginners who are unfamiliar with online editing tools. Fourth, the absence of intelligent writing assistance means users must independently compose all resume content without guidance on professional language, structure, or keyword optimization. Fifth, inadequate ATS optimization reduces resume effectiveness during automated screening processes. Sixth, inconsistent data security practices create privacy risks for users storing sensitive personal and professional information on third-party servers [12].

III.. PROPOSED SYSTEM

The Smart Resume Builder is proposed as a comprehensive, free, and intelligent full-stack web application that addresses all the critical limitations identified in existing resume building platforms. The system provides a unified platform that combines structured resume creation, AI-powered content assistance, ATS-optimized formatting, real-time preview, and secure PDF download — all accessible without financial barriers.

A. System Overview and Architecture

The Smart Resume Builder follows a three-tier web application architecture. The presentation layer consists of a React-based frontend that provides the user interface and handles all user interactions. The application layer consists of a Node.js and Express.js backend that processes API requests, manages business logic, and communicates with external services including the Google Generative AI API. The data layer consists of a MongoDB database managed through Mongoose ORM for secure and scalable storage of user accounts, resume data, and application content [5].

B. User Authentication and Registration Module

The authentication module manages user registration, login, and secure session management. Clerk authentication is integrated to provide a robust, production-grade authentication experience with support for email-based registration and login flows. Upon successful authentication, JSON Web Tokens (JWT) are issued and stored securely in the client browser. These tokens are validated by Express.js middleware for every protected API request, ensuring that only authenticated and authorized users can access or modify their personal resume data. Role-based access control ensures proper separation of user data across accounts [13].

C. Resume Creation and Data Management Module

The resume creation module provides a comprehensive, structured form-based interface for entering all resume information. The module is organized into six clearly defined sections: Personal Information (name, contact number, email, LinkedIn profile URL, and professional summary); Educational Qualifications (institution names, degree or qualification, passing year, grades or CGPA, and academic achievements); Technical and Professional Skills (programming languages, software tools, technical competencies, and soft skills); Work Experience (company names, job titles, duration of employment, responsibilities, and key achievements); Projects (project titles, technologies used, project descriptions, team roles, and outcomes); and Certifications and Achievements (online certifications, workshop participation, awards, and professional training) [14].

Redux Toolkit manages the application state in real-time during form interactions, ensuring data consistency across all components. React Router DOM handles navigation between different pages of the application. The entered data is submitted to the Node.js backend through Axios HTTP requests, validated by the backend middleware, and stored in MongoDB through Mongoose schema operations.

The saved data is then retrieved and rendered in the resume preview component for real-time display.

D. AI-Powered Content Assistance Module

The AI-powered content assistance module represents the most significant innovation of the Smart Resume Builder. This module integrates Google Generative AI to provide intelligent content suggestions that address one of the most critical challenges faced by resume creators — the difficulty of writing professional, compelling, and keyword-optimized resume content. Many users, particularly students and fresh graduates, struggle to articulate their qualifications, experiences, and skills in a manner that resonates with recruiters and satisfies ATS requirements [4].

When a user requests AI assistance, the frontend sends a request to a dedicated backend API endpoint. The backend constructs a structured prompt incorporating the user's profile data — including their educational background, skills, work experience, and target role information — and sends this prompt to the Google Generative AI API. The API processes the prompt and returns professionally generated content including career objective summaries, skill descriptions, experience bullet points, and project summaries. This content is returned to the frontend for display, where users can review, edit, and apply the suggestions to their resume [4].

E. Resume Preview and PDF Download Module

After completing the resume creation process, users can access the Resume Preview Module, which renders the complete resume in real-time using professional formatting and the selected template layout. The preview allows users to verify the accuracy of all entered information, check formatting consistency, and review the overall visual presentation before downloading. This step is critical for ensuring that the final PDF output meets professional standards and is free from errors [15].

The PDF generation functionality converts the formatted resume preview into a downloadable PDF document. PDF is the industry-standard format for resume submission, ensuring consistent formatting across all devices and email clients. The download feature is completely free, removing the financial barrier that restricts PDF access on many competing platforms.

F. Frontend User Interface Design

The Smart Resume Builder's user interface is designed with a strong emphasis on simplicity, responsiveness, and accessibility. React components are organized in a modular, reusable architecture that promotes efficient development and easy maintenance. Tailwind CSS provides utility-first styling for rapid and consistent design implementation across all pages. Radix UI components improve accessibility compliance and visual consistency. Framer Motion provides smooth, professional animations that enhance the user experience without impacting performance. The interface is fully responsive, adapting seamlessly to desktop, tablet, and mobile screen sizes [16].

IV.. METHODOLOGY

The Smart Resume Builder is developed using an iterative and modular Agile software development methodology. The system is divided into independent functional modules — authentication, resume creation, AI assistance, preview, and download — each developed, tested, and validated independently before integration into the complete system. This approach enables faster testing cycles, efficient debugging, and flexibility in accommodating requirement changes during development.

A. Frontend Development Methodology

The frontend development follows a component-based architecture using React. Each page and feature is implemented as a reusable React component with clearly defined props and state management responsibilities. Vite is used as the build tool, providing significantly faster development server startup times and optimized production builds compared to traditional Create React App setups. Redux Toolkit centralizes application state management, ensuring predictable state updates and consistent data flow across all components. React Router DOM provides declarative client-side routing for smooth navigation between the home page, registration page, login page, resume builder form, and preview page [17].

B. Backend Development Methodology

The backend is developed following a RESTful API architecture using Node.js and Express.js. API endpoints are organized by resource type: authentication routes, user profile routes, resume management routes, and AI assistance routes. Each route handler is kept lean by delegating business logic to dedicated service functions, improving testability and maintainability. Axios is used on the frontend for making structured HTTP requests to the backend API, with request interceptors automatically attaching JWT tokens to all protected route requests. Error

handling middleware provides consistent error responses across all endpoints.

C. Database Design Methodology

The MongoDB database is designed using a document-oriented schema approach managed through Mongoose. The database consists of three primary collections: the User collection, the Resume collection, and the Template collection. Mongoose schemas define the structure, data types, and validation rules for each collection. The User schema includes user ID, full name, email address, Clerk authentication identifier, and account creation timestamp. The Resume schema includes user ID as a foreign reference, personal details as a nested object, education as an array of objects, skills as a string array, work experience as an array of objects with nested fields, projects as an array, and certifications as an array. The Template schema stores template metadata including layout configuration and styling properties [14].

D. AI Integration Methodology

The Google Generative AI integration follows a prompt engineering approach. Backend service functions construct structured, contextual prompts by combining static instruction templates with dynamic user profile data. Prompts are carefully designed to guide the AI model toward generating content that is professional, industry-appropriate, and resume-optimized. API requests are made asynchronously from the backend to avoid blocking other user-facing operations. Response parsing functions extract the relevant generated content from the API response and format it appropriately before returning it to the frontend. API keys and sensitive credentials are stored securely as environment variables and are never exposed in client-side code [4].

E. Security Implementation Methodology

The security implementation follows industry-standard best practices. Clerk handles all authentication flows including registration, login, email verification, and session management. JWT tokens issued upon successful login are validated by Express.js middleware for every protected API route, ensuring unauthorized requests are rejected with appropriate HTTP 401 responses. Input validation is implemented on both the frontend (using form validation logic) and backend (using Express middleware) to prevent submission of malformed or malicious data. All API keys, database connection strings, and sensitive configuration values are stored in server-side environment variables. HTTPS is enforced for all data transmission during production deployment to prevent data interception [13].

F. Testing Methodology

The testing strategy covers three levels: unit testing validates individual React components and Express route handlers in isolation; integration testing validates the interactions between frontend API calls, backend route processing, and MongoDB database operations; and system testing validates complete end-to-end user workflows from registration through resume creation to PDF download. Manual test cases are executed for all major feature scenarios including positive test cases with valid inputs, negative test cases with invalid or missing inputs, boundary test cases for edge conditions, and security test cases for authentication and authorization behavior.

V.. SYSTEM DESIGN AND IMPLEMENTATION

A. Complete System Architecture

The Smart Resume Builder's architecture consists of five integrated layers working in coordination. The Client Layer renders the React application in the user's browser and manages all user interactions. The State Management Layer uses Redux Toolkit to maintain global application state and ensure consistent data flow between components. The Authentication Layer uses Clerk for secure session management and JWT for API request authorization. The API Layer consists of Express.js route handlers that process requests, apply business logic, and communicate with the database and external AI service. The Data Layer uses MongoDB with Mongoose for persistent, scalable storage of all application data [5].

B. Data Flow Diagram — Level 0

At the system context level, the user interacts with the Smart Resume Builder system by providing authentication credentials, entering resume information, and requesting AI content suggestions. The system processes user inputs, communicates with the Google Generative AI API to generate intelligent content suggestions, and returns structured, formatted resume outputs. The system stores all user and resume data in MongoDB and retrieves it for display and editing on demand.

C. Data Flow Diagram — Level 1

At the subsystem level, the system is divided into five primary process modules. Module 1.0 handles User Authentication — receiving registration and login inputs,

processing them through Clerk, issuing JWT tokens, and storing user records in MongoDB. Module 2.0 handles Resume Data Management — collecting form inputs from users, validating data, storing structured resume records in MongoDB, and retrieving them for rendering. Module 3.0 handles AI Content Generation — receiving user profile data, constructing contextual prompts, sending requests to the Google Generative AI API, parsing responses, and returning generated content to the frontend. Module 4.0 handles Resume Preview Rendering — retrieving stored resume data from MongoDB, applying template formatting, and rendering the formatted preview in the browser. Module 5.0 handles PDF Export — converting the rendered resume preview into a downloadable PDF document and delivering it to the user.

D. Use Case Analysis

The primary actors in the Smart Resume Builder system are the Registered User and the System Administrator. Registered Users can perform the following use cases: Register a new account, Log in securely, Complete resume creation form, Request AI-generated content suggestions, Preview the formatted resume, Download the resume as PDF, Update existing resume information, and Log out securely. The System Administrator can monitor system performance, manage database integrity through Mongoose, configure API keys and environment variables, and manage cloud deployment and application updates.

E. ER Diagram and Database Relationships

The Entity-Relationship model of the Smart Resume Builder consists of three primary entities. The User entity has attributes: UserID (primary key), Name, Email, ClerkID, and CreatedAt. The Resume entity has attributes: ResumeID (primary key), UserID (foreign key), PersonalDetails, Education, Skills, WorkExperience, Projects, Certifications, AISummary, and UpdatedAt. The Template entity has attributes: TemplateID, TemplateName, LayoutConfig, and StylingConfig. The relationships are: User to Resume is a one-to-many relationship (one user can have multiple resumes); Resume to Template is a many-to-one relationship (multiple resumes can use the same template).

F. Implementation Highlights

Key implementation details include the following. The Redux store is initialized with three slices: authSlice managing user authentication state, resumeSlice managing resume form data and saved resumes, and uiSlice managing UI states such as loading indicators and modal visibility. The Express.js backend defines route groups under /api/auth for authentication endpoints, /api/resume for resume CRUD operations, and /api/ai for AI content generation requests. The Google Generative AI integration uses the @google/generative-ai npm package, with prompts dynamically constructed by injecting user profile data into structured template strings before API submission. The PDF generation uses browser-based rendering with a print-to-PDF approach, converting the styled resume preview component directly into a downloadable PDF file [16].

VI.. RESULTS AND DISCUSSION

A. Functional Testing Results

All modules of the Smart Resume Builder were systematically tested across multiple user scenarios. Table I presents the complete functional test case results, confirming reliable operation across all features.

TC ID	Module	Test Input
TC01	Registration	Valid name, email, password
TC02	Registration	Already registered email
TC03	Login	Valid registered credentials
TC04	Login	Incorrect password entered
TC05	Resume Form	All sections fully completed
TC06	Resume Form	Required fields left empty
TC07	AI Assist	User profile data submitted
TC08	AI Assist	Skills section AI triggered
TC09	AI Assist	Experience section triggered
TC10	Preview	Completed resume submitted
TC11	PDF Download	Preview confirmed by user
TC12	Auth Security	Unauthenticated API request
TC13	Responsive UI	Accessed on mobile browser
TC14	Logout	Authenticated user logs out

Table I. Functional Test Cases and Results

B. Performance Analysis

The platform demonstrated consistently stable performance across all tested scenarios. The React and Vite combination provided fast frontend rendering with optimized production bundle sizes and minimal page load times. Redux Toolkit maintained consistent global state without performance bottlenecks during multi-section form interactions. The Node.js and Express.js backend handled concurrent API requests efficiently, with Mongoose queries retrieving and storing MongoDB

documents with low latency. Google Generative AI API calls were executed asynchronously on the backend, ensuring that AI content generation did not interrupt or delay other real-time user interactions such as form input and navigation.

C. Comparison with Existing Systems

Table II presents a comprehensive comparison between existing resume building platforms and the proposed Smart Resume Builder across ten key feature dimensions.

Feature	Existing Systems	Smart Resume Builder
Cost / Accessibility	PDF download requires payment	Completely free to create and download
Template Customization	Limited, fixed templates only	Flexible, customizable templates
User Interface	Often complex for beginners	Clean, simple, beginner-friendly design
AI Writing Assistance	Absent or very limited	Google Generative AI fully integrated
ATS Optimization	Often poor or not addressed	Structured, ATS-friendly output format
Authentication Security	Basic login credentials only	Clerk + JWT secure session management
Mobile Compatibility	Primarily desktop-only design	Fully responsive across all devices
Real-Time Preview	Not consistently available	Live preview before PDF download
Data Privacy	Inconsistent privacy practices	Secure MongoDB with JWT protection
Intelligent Skill Suggest	Not available	AI-powered skill recommendations

Table II. Comparison of Existing Systems and Proposed Smart Resume Builder

D. Security Validation Results

Security testing confirmed that all unauthenticated API requests to protected routes are correctly rejected with HTTP 401 Unauthorized responses, preventing unauthorized data access. JWT token validation functions correctly across all backend middleware, with expired tokens automatically refused. Input validation on both frontend and backend successfully prevents submission of incomplete or malformed data. Clerk's authentication flow correctly enforces email verification before account activation. All sensitive credentials including MongoDB connection strings, Google AI API keys, and JWT secret keys are stored as server-side environment variables and confirmed to be absent from any client-side code. HTTPS enforcement during deployment ensures all data transmission is encrypted.

E. User Experience Evaluation

The user interface design was evaluated for simplicity, responsiveness, and ease of use. The structured form-based resume creation process was confirmed to be intuitive for users with no prior technical experience in resume building. The six-section form layout clearly organizes all required resume information, reducing confusion and data entry errors. The AI suggestion feature was evaluated as significantly reducing the time required to compose professional resume content, particularly for users who lack professional writing experience. The real-time preview feature was confirmed to provide valuable visual feedback before the final PDF download, enabling users to identify and correct errors proactively.

VII.. FUTURE ENHANCEMENTS

The Smart Resume Builder establishes a strong technical foundation for a wide range of future enhancements. The following improvements are planned for subsequent development phases to expand the system's capabilities and further improve user experience, intelligence, and accessibility.

A. Multiple Professional Resume Templates

The currently system provides a standard professional resume format. Future development will introduce multiple professionally designed templates, tailored to different industries, career stages, and personal preferences. Template categories will include Technical / Engineering, Creative Design, Corporate / Business, Academic / Research, and Entry-Level / Fresher formats. Users will be able to switch between templates while retaining all entered resume data, allowing them to select the most appropriate presentation style for each job application without re-entering information.

B. Cloud-Based Resume Storage and Multi-Device Access

Integrating cloud-based resume storage will allow users to save multiple versions of their resumes online and access them from any device at any time. This enhancement eliminates the risk of losing resume data due to browser clearing or device changes. An auto-save feature will periodically save draft resumes to the cloud, preventing data loss during extended editing sessions. Version history management will allow users to revert to previous resume versions when needed, providing greater flexibility for managing multiple tailored resumes for different job applications.

C. Job Portal Integration and Direct Application

Future versions of the Smart Resume Builder will integrate with major job portals and professional networking platforms such as LinkedIn, Naukri.com, Indeed, and Internshala. LinkedIn profile import functionality will allow users to automatically populate their resume forms

with existing profile data, significantly reducing manual data entry effort. Direct one-click resume submission to compatible job postings will streamline the job application workflow, making the Smart Resume Builder a comprehensive job application management tool.

D. AI-Driven Resume Scoring and Optimization

Beyond content generation, future AI enhancements will include an automated resume scoring system that evaluates resumes based on multiple dimensions: ATS keyword compatibility with target job descriptions, formatting quality and consistency, content completeness and professional language quality, and quantification of achievements. The AI scoring system will provide detailed, actionable feedback on each dimension with specific recommendations for improvement. Users will be able to paste a specific job description into the system, and the AI will analyze the resume against the job requirements and suggest targeted modifications to maximize the matching score.

E. Interview Preparation and Career Coaching Module

A comprehensive interview preparation module will be integrated as a future enhancement, transforming the Smart Resume Builder into a complete career development platform. This module will generate commonly asked interview questions based on the user's resume content and target role. AI-powered model answer suggestions will help users structure their responses effectively. Role-play interview simulation features will allow users to practice answering questions in a timed environment. Additionally, a career guidance module will provide personalized recommendations for skill development, learning resources, and career progression paths based on the user's profile and target industry.

F. Resume Analytics and Application Tracking

Future enhancements will include a resume analytics dashboard that provides users with insights into how their resumes perform across different job applications. Application tracking functionality will allow users to record and monitor the status of each job application, including the company name, application date, current status, and interview outcomes. This data will be analyzed to identify patterns in application success rates and provide recommendations for resume optimization based on real application results.

G. Gamification and Engagement Features

To encourage consistent engagement and skill development, gamification elements will be introduced including resume completion scores that motivate users to provide comprehensive profile information, achievement badges for milestones such as creating a first resume or achieving a high AI quality score, daily improvement tips and reminders, and community features where users can share anonymized resume examples and receive peer feedback. These features will enhance user motivation and improve the long-term utility of the platform.

VIII.. CONCLUSION

The Smart Resume Builder presents a comprehensive, intelligent, and user-friendly full-stack web application that successfully addresses the major limitations of existing resume creation systems. By integrating a modern technology stack comprising React, Vite, Redux Toolkit, Tailwind CSS, Radix UI, Framer Motion, Node.js, Express.js, MongoDB, Mongoose, Clerk authentication, and JSON Web Tokens with the advanced capabilities of Google Generative AI, the system delivers a powerful and accessible platform that simplifies and automates the professional resume creation process for all categories of job seekers.

The system directly overcomes the critical limitations of existing resume builders. It provides completely free resume creation and PDF download, removing the financial barriers that restrict access on competing platforms. Flexible, customizable templates and a clean, beginner-friendly interface make the system accessible to users of all technical backgrounds. The integration of Google Generative AI represents a significant innovation — providing intelligent professional content generation for summaries, skill descriptions, and experience bullet points that help users produce high-quality, ATS-optimized resumes without requiring advanced writing experience or deep knowledge of recruitment standards.

Functional testing across fourteen test cases confirmed reliable and correct operation across all system modules including user registration, login authentication, resume form submission, AI content generation, resume preview rendering, PDF download, security validation, and responsive design. Performance analysis demonstrated fast frontend rendering through React and Vite, efficient database operations through MongoDB and Mongoose, and reliable asynchronous AI content generation that does not impact real-time user interactions. Security testing

confirmed robust JWT authentication, proper access control on all protected routes, secure credential management, and effective input validation throughout the application.

The comparison with existing resume builder platforms demonstrates that the Smart Resume Builder offers significant advantages across all evaluated dimensions including cost accessibility, template customization, user interface simplicity, AI assistance depth, ATS optimization quality, authentication security, mobile compatibility, and data privacy protection. The system stands as a practical, high-impact application that empowers students, fresh graduates, and professional job seekers to create polished, professional resumes efficiently and confidently. With planned future enhancements including multiple template designs, cloud-based storage, job portal integration, AI-driven resume scoring, interview preparation modules, and gamification features, the Smart Resume Builder has strong potential to evolve into a complete, intelligent career development ecosystem.

REFERENCES.

1. R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Dept. Inf. Compute. Sci., Univ. California, Irvine, CA, USA, 2000.
2. J. J. Garrett, The Elements of User Experience: User-Centered Design for the Web and Beyond, 2nd ed. Berkeley, CA: New Riders, 2010.
3. G. Mikowski and J. Powell, Single Page Web Applications: JavaScript End-to-End, 1st ed. Shelter Island, NY: Manning Publications, 2013.
4. Google Generative AI Documentation — AI model for resume content generation and suggestions. Available: https://ai.google.dev/docs
    A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed. Sebastopol, CA: O'Reilly Media, 2020.
5. World Economic Forum, "Future of Jobs Report 2023," WEF Publications, 2023. Available: https://www.weforum.org/reports/the-future-of-jobs-report-2023
6. LinkedIn Economic Graph, "Jobs on the Rise: AI and Data Specialists," LinkedIn Workforce Report, 2023. Available: https://economicgraph.linkedin.com
7. McKinsey & Company, "The State of AI in 2024," McKinsey Global Institute, 2024. Available: https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai
8. T. Brown, Designing with Data: Improving the User Experience with A/B Testing. Sebastopol, CA: O'Reilly Media, 2014.
9. M. Haverbeke, Eloquent JavaScript: A Modern Introduction to Programming, 3rd ed. San Francisco, CA: No Starch Press, 2018.
10. H. Schildt, The Complete Guide to Modern Web Security: JWT and Authentication Implementation. McGraw-Hill Education, 2020.
11. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, 1st ed. Cambridge, MA: MIT Press, 2016.
12. Clerk Documentation — Authentication and user management platform. Available: https://clerk.com/docs
13. MongoDB Documentation — Document-based NoSQL database. Available: https://www.mongodb.com/docs
14. Mongoose Documentation — MongoDB object modelling for Node.js. Available: https://mongoosejs.com/docs
15. React Official Documentation — JavaScript library for building user interfaces. Available: https://react.dev
16. Vite Official Documentation — Next-generation frontend build tool. Available: https://vitejs.dev
17. Redux Toolkit Documentation — State management library for React. Available: https://redux-toolkit.js.org
18. Tailwind CSS Documentation — Utility-first CSS framework. Available: https://tailwindcss.com/docs
19. Node.js and Express.js Official Documentation. Available: https://nodejs.org/en/docs and https://expressjs.com
20. Purmani, S. S. R., & Kotte, G. Intelligent Project Orchestration: How Generative AI is Reshaping Go-to-Market Strategy Planning and Cross-Functional Delivery. environments, 4, 5.
21. Scikit-learn Developers, "StandardScaler Documentation," Available: https://scikit-

learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

22. "Radar Signal Processing Resources," GitHub Repository. Available:
23. https://github.com/radiojhero/awesome-radar

24. "Aircraft Classification Projects," GitHub
25. Topics. Available:
https://github.com/topics/aircraft-classification
26. "TSFresh: Time Series Feature Extraction Library," GitHub Repository. Available: https://github.com/blue-yonder/tsfresh
27. "Streamlit Official Repository," GitHub. Available: https://github.com/streamlit/streamlit

28. "Scikit-learn Machine Learning Library," GitHub. Available: https://github.com/scikit-learn/scikit-learn

29. "ADS-B Aircraft Dataset," Kaggle Dataset Repository. Available:
30. https://www.kaggle.com/datasets/aircraft-adsb

31. F. Chollet, *Deep Learning with Python*, Manning Publications, 2018.

32. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, 2021.

33. Gaddam, S. (2023). Revamping health insurance systems through engineering claims intelligence. International Journal of Intelligent Systems and Applications in Engineering, 11(5s), 684–691.

34. Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.

35. Poojari, R. Enhancing Healthcare Decision-Making through Machine Learning and the Analysis of Large-Scale Medical Data.

36. Purmani, S. S. R. (2024). Aligning IT investment decisions with overall business strategy from an enterprise program management perspective, focusing on the integration of IT leadership in strategic decision-making processes. International Journal of Communication Networks and Information Security, 16(5), 1213–1219

37. Kumara, S. (2025). Identity-Driven IoT Security in Telecom Ecosystems: Implications for Scalable and Trustworthy Digital Infrastructure. Int. J. Appl. Math, 38(12s), 2797-2816.

38. Kotte, G. (2025). Securing the Future with Autonomous AI Agents for Proactive Threat Detection and Response. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283830

39. Viswanathan, V., Polagani, S. S., Agarwal, R., Akula, S., Dey, S., & Kashyap, R. (2025, September). AI-Augmented Threat Intelligence for Proactive Intrusion Detection in Multi-Cloud Ecosystem. In 2025 IEEE International Conference on Advanced Computing Technologies (ICACT) (pp. 567-572). IEEE.

40. Mudusu, S. K. (2025, June 3). Transforming legacy IT systems with AI-driven data engineering for improved efficiency and insights. Hampton Global Business Review (HGBR).

41. Mudusu, S. K. (2025, April 20). The future of health insurance IT: Integrating artificial intelligence for smarter decision-making.

42. Gajula, S., Bondhala, S., & Margam, M. (2026, February). Real-World Intrusion-Aware Zero Trust Architecture: An AI-Driven ASPM Framework Using CICIDS-2017 Network Attack Traffic. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-7). IEEE.

43. Subramanian, V. K., Bhambri, S., & Gajula, S. (2025, April). Disentangled Graph Variational Auto-encoder Based Framework to Improve the Operational Efficiency in Cloud

Computing Environments. In International Conference on Computer Vision and Robotics (pp. 396-407). Cham: Springer Nature Switzerland.

44. Maturi, S. Y. (2025). Blockbond Hardening: Securing Pooled-Hash Protocols Against Traffic Tampering, MITM Hash-Rate Hijacking, and Template Coercion. https://doi.org/10.20944/preprints202512.2064.v1

45. Ranjbareslamloo, S., Dzukey, G. A., Islam Muhit, M. M., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. Manufacturing Letters, 44, 915–927. https://doi.org/10.1016/j.mfglet.2025.06.108

46. Manoharan, D. (2026). Advancing Healthcare EDI Interoperability Through Informatica Cloud B2B Gateway Quality Engineering. Available at SSRN 6385719.

47. Manoharan, D. (2026). AI-Driven Anomaly Detection Models for Preventing Claims Denials and Revenue Leakage in Healthcare. Available at SSRN 6385759.

48. Venkata Ramana, P. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. International Journal of Research in Information Technology and Computing, 8(4).

49. Pavan Kumar Adabala. (2026). IoT-Driven Digital Twins for Manufacturing Optimization: Hybrid Modelling, Reinforcement Learning and Sustainable Operations. International Journal of Computational and Experimental Science and Engineering, 12(1). https://doi.org/10.22399/ijcesen.5050

50. Kavuri, S. (Ed.). (2024). Shift-left and shift-right testing approaches: A practical roadmap for continuous quality in agile and DevOps. Journal of Information Systems Engineering and Management, 9(4). https://doi.org/10.52783/jisem.v9i4.127

51. Srikanth Kavuri. (2023). Machine Learning Approaches for Security Vulnerability Detection in Software Testing. Computer Fraud and Security. https://doi.org/10.52710/cfs.837

52. Gummadi, V. P. K., Chilamkurthi, L. S., & Kavuri, S. (2026). Securing APIs in Government Clouds and Runtime Fabric Using FIPS-Enabled MuleSoft. 2026 International Conference on Artificial Intelligence, Systems, and Emerging Technologies (ICAISET), 1–6. https://doi.org/10.1109/icaiset66439.2026.11542099

53. Kumar Gummadi, V. P., Chilamkurthi, L. S., & Kavuri, S. (2026). Distributed Platform Architecture and API-Led Integration. 2026 International Conference on Artificial Intelligence, Systems, and Emerging Technologies (ICAISET), 1–6. https://doi.org/10.1109/icaiset66439.2026.11541787

54. Gajula, S. (2025). Cloud transformation in financial services: A strategic framework for hybrid adoption and business continuity. International Journal of Scientific Research in Computer Science, Engineering and Information technology.

55. Gajula, S., Bondhala, S., & Margam, M. (2026). Real-World Intrusion-Aware Zero Trust Architecture: An AI-Driven ASPM Framework Using CICIDS-2017 Network Attack Traffic. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–7. https://doi.org/10.1109/icaic67076.2026.11395835

56. Gajula, S. (2026). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. 2026 14th International Symposium on Digital Forensics and Security (ISDFS), 1–6. https://doi.org/10.1109/isdfs69419.2026.11458989

57. 95Gajula, S. (2025). AI-Powered Forecasting Models, Optimizing Working Capital, Supply Chain Financing. 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM), 1–6. https://doi.org/10.1109/rcsm67767.2025.11507813