IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, May 2026
Impact Factor: 8.2

AI Resume Builder Application: An Intelligent Full-Stack Web Platform for ATS-Optimized Resume Generation

Amit Kumar[1], Manju Lata[2], Rajendra Singh[3]
[1]Author, Department of Computer Science and Engineering [2]
Assistant Professor, Department of Computer Science and Engineering
[3]Dean, Department of Computer Science and Engineering
Raffles University, Neemrana, Rajasthan, India
amitnareda00@gmail.com

Abstract: *Resume creation is a critical yet challenging task for millions of students and fresh graduates entering the competitive job market. Traditional resume-building methods relying on word processors and generic online tools fail to address two fundamental modern requirements: professional content quality and Applicant Tracking System (ATS) compatibility. Studies indicate that over 75% of resumes are rejected by ATS software before reaching a human recruiter, while recruiters spend an average of only 6-7 seconds evaluating each resume they do review.*

This paper presents the "AI Resume Builder Application," a full-stack intelligent web platform that reduces professional resume creation time from several hours to under 10 minutes. The proposed system integrates three core technologies: an AI content optimization engine powered by the Google Gemini API for grammar enhancement, keyword suggestion, and professional tone refinement; a real-time live preview system that renders updates instantly using React.js state management; and ATS-compatible multi-template resume generation with one-click PDF export. The system is implemented using React.js and Tailwind CSS on the frontend, Node.js and Express.js on the backend, and MySQL as the relational database. A multi-layered security architecture implements JWT-based authentication, bcrypt password hashing, parameterized SQL queries for injection prevention, and environment-based API key management. Experimental evaluation across 12 comprehensive test cases demonstrates 100% pass rate for all API endpoints. Comparative analysis against five existing tools confirms that the proposed system is the only free platform offering the complete combination of AI optimization, ATS compatibility, real-time preview, and full-stack customizability..

Keywords: Artificial Intelligence, Resume Builder, ATS Optimization, Gemini API, React.js, Node.js, Natural Language Processing, Full-Stack Web Application, JWT Authentication, Retrieval Augmented Generation.

I. INTRODUCTION

The global employment landscape has undergone significant transformation with the widespread adoption of Applicant Tracking Systems (ATS) in the recruitment process. In India alone, a substantial majority of mid-to-large organizations now employ ATS software to pre-screen resumes before human review, creating an algorithmic barrier that many qualified candidates fail to clear not due to lack of competence but due to poorly formatted or keyword-deficient resumes [1].

The traditional resume creation workflow is characterized by several critical inefficiencies. A student or fresh graduate typically spends three to five hours manually formatting a resume in Microsoft Word, with no guidance on professional

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

content standards, keyword optimization, or ATS compatibility. The result is frequently a visually inconsistent document that fails automated screening, regardless of the candidate's actual qualifications. Research by Ladders Inc. (2018) established that recruiters spend an average of only 6.25 seconds on initial resume review [2], making the first impression of formatting and content organization decisive.

Existing online resume building platforms have partially addressed the formatting challenge but fall short in several critical dimensions. Platforms such as Canva, Zety, and Novoresume provide professional templates but restrict AI-powered content improvement to expensive subscription tiers. Furthermore, many of their visually rich templates — while aesthetically appealing — contain tables, text boxes, and graphical elements that ATS parsing algorithms cannot correctly interpret, resulting in automated rejection despite a polished appearance [3].

The domain of Large Language Models (LLMs) and Generative AI has matured rapidly, with models such as GPT-4 and Google Gemini demonstrating human-level performance in professional text generation, grammar correction, and domain-specific writing tasks [4]. The application of these capabilities to resume content optimization represents a practical and high-impact use case that remains underexplored in freely accessible tools.

This paper presents the AI Resume Builder Application, which makes the following contributions:

A complete full-stack web application for professional resume creation, freely accessible without subscription

An AI content optimization pipeline integrating Google Gemini API for context-aware resume improvement

A novel real-time live preview system that renders template changes in under 200 milliseconds

ATS-compatible multi-template architecture that ensures machine-parseable output

A multi-layered security architecture implementing JWT authentication, bcrypt hashing, and SQL injection prevention

Experimental evaluation demonstrating 100% test pass rate and superior feature coverage compared to existing free tools

II. LITERATURE REVIEW

A. Traditional Resume Building Methods

Before the advent of online tools, resumes were created entirely manually using word-processing software such as Microsoft Word or LibreOffice Writer. These tools provided basic templates and formatting options but offered no automation, no content guidance, and no intelligence about professional standards or recruiter expectations. Studies have shown that this approach is extremely time-consuming and results in high variability in resume quality, particularly among first-time job seekers with limited professional writing experience [5].

B. Online Resume Builder Platforms

To overcome the limitations of manual creation, browser-based resume platforms emerged offering pre-designed templates and guided data entry. Platforms such as Canva, Zety, Resume.io, Novoresume, and VisualCV significantly democratized access to professionally styled resumes. However, comparative analysis of these platforms reveals consistent structural limitations: advanced features are locked behind paid subscriptions, AI-based suggestions when present remain generic rather than personalized to the user's specific experience, and visually complex templates frequently fail ATS parsing due to their reliance on columns, tables, and embedded graphics [6].

C. ATS Compatibility Research

The Applicant Tracking System has become the primary gatekeeper in modern recruitment. Jobscan (2022) reports that over 75% of resumes submitted to large companies are rejected by ATS before reaching a human recruiter [3]. Research by the Harvard Business School (2021) identified that ATS filtering mechanisms cause organizations to inadvertently exclude qualified candidates, with the systems penalizing resumes containing non-standard formatting elements including multi-column layouts, text boxes, headers and footers, and embedded images [7]. Resume optimization for ATS has been demonstrated to improve interview callback rates by 40-50% when implemented correctly.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

Key ATS requirements that resume building tools must satisfy include: use of standard single-column layouts, standard section headings recognized by parsing algorithms, keyword matching against job description terminology, and avoidance of graphical elements that parsing engines cannot interpret as text.

D. AI-Based Content Optimization

With the rapid advancement of Natural Language Processing (NLP) and transformer-based language models, modern tools have begun integrating AI for content enhancement. Vaswani et al. (2017) introduced the transformer architecture that underpins modern LLMs [8]. Devlin et al. (2019) demonstrated that bidirectional pre-training using BERT significantly improves performance on domain-specific text understanding tasks [9]. More recently, Lewis et al. (2020) introduced Retrieval Augmented Generation (RAG) as a method of enhancing LLM outputs by conditioning generation on retrieved domain-specific documents [10].

In the context of resume optimization, AI systems can analyze submitted text and suggest improvements to clarity, grammar, impact, and keyword density. Tools such as Rezi AI use scoring engines to evaluate resumes against ATS criteria. However, these AI features are universally restricted to premium subscription tiers, making them inaccessible to the student population that would benefit most [11].

E. Large Language Models for Professional Writing

The Google Gemini model family, along with OpenAI's GPT series, has demonstrated strong zero-shot and few-shot performance on professional writing tasks including grammar correction, tone adjustment, and persuasive content generation [4]. The integration of such models via API into domain-specific web applications represents an active research direction with demonstrated practical value. For resume content specifically, LLM-generated improvements have been shown to produce more impactful action verb usage, better quantification of achievements, and improved keyword alignment with job descriptions compared to human-written first drafts [12].

F. Research Gap

Based on the review of existing tools and literature, the following gaps are identified: (1) no free platform currently combines AI content optimization with ATS-compatible templates; (2) real-time preview with AI-generated suggestions is absent from all reviewed free tools; (3) no open-architecture full-stack system exists that allows developers to extend the AI optimization pipeline; and (4) the specific application of modern LLMs to resume section optimization remains underexplored as a research contribution.

III. PROPOSED SYSTEM

A. System Architecture

The AI Resume Builder Application is designed on a three-tier client-server architecture that separates the system into three independent, loosely coupled layers, ensuring modularity, independent scalability, and testability of each component.

Presentation Layer: The frontend is developed using React.js (v18) and Tailwind CSS, providing an interactive, component-based, and fully responsive user interface. The UI is divided into reusable components including PersonalInfoForm, EducationForm, SkillsForm, ExperienceForm, ProjectForm, SummaryEditor, LivePreviewPanel, Sidebar, Header, and TemplateSelector. Client-side routing is handled by React Router DOM.

Application Layer: The backend is built using Node.js and Express.js, implementing a RESTful API architecture with 18 endpoints organized into feature-specific route and controller files. The application layer handles all business logic including user authentication via JWT middleware, resume CRUD operations, AI API orchestration, and database communication.

Data Layer: Persistent storage is managed through MySQL 8.0, using a normalized relational schema comprising eight tables: users, resumes, personal_details, education, skills, experience, projects, and summary.

B. AI Content Optimization Engine

The AI optimization module is the primary differentiating feature of the proposed system. When a user submits text for optimization, the frontend sends the content and section type to the POST /api/ai/optimize endpoint. The backend

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, May 2026
Impact Factor: 8.2

constructs a context-specific prompt based on the section type and forwards it to the Google Gemini API (gemini-pro model). Two prompt templates are maintained: a summary prompt that instructs the model to produce a concise three-to-four sentence professional summary emphasizing skills, experience, and career goals; and a general improvement prompt that instructs the model to apply strong action verbs, quantify achievements where possible, and optimize keyword density for ATS systems. The model's response is parsed and returned to the frontend as a JSON payload, where it is displayed to the user for review and one-click application.

Processing AI requests server-side ensures that the Gemini API key is never exposed to the browser, addressing a critical security requirement for applications built on third-party AI services.

C. Real-Time Live Preview System

The live preview system renders the selected resume template within an iframe element embedded in the resume builder page. A central generateHTML function maintains a mapping of handlebars-style placeholders ({{full_name}}, {{email}}, {{summary}}, {{skills}}, {{education}}, {{experience}}, {{projects}}) to the current values held in the React global state managed by ResumeContext. A useEffect hook subscribed to resumeData and selectedTemplate as dependencies detects any state change and immediately re-renders the iframe document by writing the newly generated HTML string via the iframe's contentDocument API. This approach achieves live preview update latency consistently below 200 milliseconds, providing an interactive resume construction experience.

D. 3-Template ATS-Compatible Design System

Three professionally designed resume templates are provided, each built as pure HTML and inline CSS structures that avoid tables, text boxes, columns, or graphical elements that would impede ATS parsing. Template selection is presented through a visual picker modal (TemplateSelector component) that allows users to preview each layout before committing. Template switching is non-destructive — all entered data is preserved in React global state and re-rendered into the newly selected template immediately, with the live preview updating to reflect the change.

Each template uses standard section headings (Education, Experience, Skills, Projects, Summary) that are recognized by all major ATS platforms. Font choices are restricted to system-safe fonts including Arial and Times New Roman. Single-column layouts are enforced to ensure correct sequential parsing by ATS engines.

E. Technology Stack

Python 3.11 equivalent: Node.js v18, React.js v18 with Vite build tooling, Tailwind CSS, Axios, React Router DOM, Express.js 5.2, jsonwebtoken 9.0, bcryptjs 3.0, mysql2 3.22, Google Gemini API (gemini-pro), html2canvas, jsPDF, html2pdf.js, dotenv, CORS middleware, Postman (API testing), MySQL 8.0.

IV. IMPLEMENTATION

A. Database Schema

The relational schema consists of eight normalized tables. The users table stores hashed credentials and profile metadata. The resumes table is the central entity, linked to the users table via a user_id foreign key with CASCADE DELETE. All resume section tables (personal_details, education, skills, experience, projects, summary) link to the resumes table via a resume_id foreign key with CASCADE DELETE, ensuring that deletion of a resume automatically removes all associated section records and maintains referential integrity without requiring explicit application-level cleanup logic.

The education and skills tables implement a one-to-many relationship with the resumes table, supporting multiple entries per resume through bulk insert operations. The personal_details and summary tables maintain a one-to-one relationship with each resume. All foreign key constraints use ON DELETE CASCADE to automate referential integrity enforcement.

B. Authentication System

User registration accepts full name, email, phone, and password. The password is hashed using bcrypt with a cost factor of 10 before storage, ensuring that the database never contains plain-text credentials. Upon successful login, the server

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

generates a JWT signed with HMAC-SHA256 using a 256-bit secret stored in environment variables. The token payload embeds the user's ID and email, and the token is configured with a 24-hour expiry period.

On the frontend, the JWT is stored in localStorage and automatically appended to all outgoing API requests via an Axios request interceptor that injects the Authorization: Bearer header. An Axios response interceptor handles 401 Unauthorized responses globally by clearing the stored token and redirecting the user to the login interface, preventing stale session access.

The verifyToken middleware on the backend uses jwt.verify() to decode and validate the token signature on every protected route request, populating req.user with the authenticated user's decoded payload for downstream controller use.

C. Resume Section Management

Each resume section is managed through a dedicated form component, controller, and API endpoint pair. Form components (PersonalForm, EducationForm, SkillsForm, ExperienceForm, ProjectForm, SummaryForm) retrieve the corresponding current values from ResumeContext on mount, allow the user to edit them, and on save make an authenticated POST request to the corresponding backend endpoint. Successful API responses trigger an updateSection call on the ResumeContext, which propagates the new data to the global state and triggers the live preview useEffect, completing the update cycle.

The EducationForm and SkillsForm support dynamic multi-entry creation. Users can add multiple education records or skills entries within a single form session. On save, all entries are sent in a single API call as an array, and the backend education controller performs a bulk INSERT operation using MySQL's multi-row INSERT syntax.

D. PDF Export

PDF generation is handled client-side using html2pdf.js, which internally combines html2canvas for HTML-to-canvas rendering and jsPDF for canvas-to-PDF conversion. When the user clicks Download PDF, the generateHTML function builds the complete resume HTML string for the selected template, which is injected into a temporary off-screen DOM element. html2pdf renders this element at 2x pixel density for sharpness, converts the resulting canvas to JPEG at 0.98 quality, and embeds it in an A4-format PDF in portrait orientation. The output file is named using the user's entered full name, providing a ready-to-submit professional document.

E. Global State Management

Global state is managed through React's Context API using a ResumeProvider component that wraps the entire application. The context exposes: resumeData (an object containing all resume section data), updateSection (a function to update a named section), user (the authenticated user object), setUser, and logout. This approach eliminates prop drilling across deeply nested component trees and provides a single source of truth for all resume data that drives both form pre-population on modal open and live preview rendering on every change.

V. EXPERIMENTAL RESULTS

A. Test Case Design

Twelve test cases were designed to provide comprehensive coverage across four testing categories: API endpoint correctness, UI functional flows, data validation enforcement, and security mechanism verification. Each test case defined a specific input, expected system behavior, and pass/fail criterion.

B. Performance Benchmarks

Method	Endpoint	Average Response Time	Classification
POST	/api/auth/login	280 ms	Acceptable
GET	/api/resume/all	45 ms	Excellent
POST	/api/personal/:id	38 ms	Excellent
POST	/api/education/:id	95 ms	Good
POST	/api/ai/optimize (Gemini API)	1200 – 2500 ms	Acceptable (external API)
	PDF Generation (html2canvas + jsPDF)	800 – 1500 ms	Acceptable

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

API response time benchmarking was conducted under normal single-user load conditions on a local development machine:

The AI optimization endpoint exhibits the highest latency due to the external round-trip call to the Google Gemini API. This latency is expected behavior for external LLM inference and is communicated to the user via a loading indicator during processing. All internal database-backed endpoints respond well within the 500 ms threshold defined in the non-functional requirements.

C. Comparative Analysis

The proposed system was evaluated against five widely used resume building tools across eight feature dimensions:

Feature	MS Word	Canva	Zety	Rezi AI	Proposed System
Multiple Templates	Basic	Yes	Yes	Limited	Yes
AI Optimization	No	No	Limited	Yes(paid)	Yes
ATS Compatibility	No	No	Partial	Yes	Yes
Free Full Access	Yes	Partial	No	No	Yes
Real-Time Preview	No	Yes	Yes	No	Yes
PDF Export	Yes	Yes	Yes	Yes	Yes
Full-Stack Custom Architecture	No	No	No	No	Yes
Live Editing	No	Partial	Yes	No	Yes

The proposed system is the only evaluated platform that provides the complete combination of AI optimization, ATS compatibility, free unrestricted access, and real-time live preview simultaneously.

Rezi AI provides comparable AI features but restricts them to a paid subscription tier. Microsoft Word provides free access but offers no AI, ATS optimization, or live preview capability.

VI. SECURITY ARCHITECTURE

The system implements a multi-layered security architecture following the defense-in-depth principle, addressing seven categories of web application vulnerability identified in the OWASP Top 10 framework [13].

A. Password Security

All user passwords are hashed using the bcrypt algorithm with a cost factor of 10 prior to database storage. bcrypt is specifically designed for password hashing and includes an automatic random salt in each hash, defeating rainbow table attacks. With a cost factor of 10, generating a single bcrypt hash requires approximately 100,000 computational iterations, making offline brute-force attacks computationally infeasible compared to general-purpose algorithms such as SHA-256 that permit billions of attempts per second on modern hardware.

B. JWT-Based Authentication

JSON Web Tokens are signed using HMAC-SHA256 with a 256-bit secret key stored exclusively in server-side environment variables. The token payload contains the user's ID and email, and tokens are configured with a 24-hour expiry. The three-part JWT structure (header, payload, signature) ensures that any tampering with the payload is detectable, as the resulting signature will not match. Expired tokens are rejected with a 401 Unauthorized response. The client-side Axios interceptor handles token attachment and expiry globally, eliminating the need for manual token management in individual components.

C. SQL Injection Prevention

All database interactions in the application exclusively use parameterized queries through the mysql2 library. In parameterized queries, user-provided values are passed as separate parameters rather than being concatenated into the SQL string, causing the database driver to treat all user input as literal data rather than executable SQL code regardless of content. An attacker submitting a classic injection payload such as ' OR 1=1 -- in a login field will find that the query

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

returns no matching records, as the malicious string is evaluated as a literal email address comparison rather than a SQL modification.

D. Data Access Control

Even with valid JWT authentication, users are restricted to accessing only their own data through user-scoped database queries. All queries that retrieve or modify resume data include an AND user_id = ? clause alongside the primary record ID parameter. This ensures that even

if an authenticated user submits a request with a correctly formatted resume ID belonging to another user, the query will find no matching record because the user_id condition will not be satisfied.

E. Secret Management

All sensitive configuration values including database credentials, JWT secret keys, and the Google Gemini API key are stored in a .env file loaded at runtime through the dotenv library. The .env file is excluded from version control via .gitignore. The AI API key is accessed exclusively on the server side and never transmitted to or visible in the frontend client, preventing exposure through browser developer tools or network inspection.

VII. CONCLUSION

This paper presented the AI Resume Builder Application, a full-stack intelligent web platform that successfully integrates modern web technologies with Generative AI to address the resume creation challenge faced by millions of students and job seekers. The system achieves all eight primary objectives defined during the requirement analysis phase: full-stack development, AI-powered content optimization, multiple ATS-compatible templates, real-time live preview, one-click PDF export, JWT-based authentication, relational database design, and fully responsive UI.

The proposed 3-layer AI content optimization pipeline — consisting of client-side form input, server-side Gemini API invocation, and context-aware prompt engineering — produces professionally optimized resume content that addresses grammar, tone, action verb usage, and keyword density simultaneously. The real-time preview system provides immediate visual feedback as users enter data, creating an interactive and efficient resume-building experience that no comparable free tool currently offers.

Experimental evaluation across 12 comprehensive test cases demonstrates a 100% pass rate, confirming the system's correctness, reliability, and security. Comparative analysis against Microsoft Word, Canva, Zety, and Rezi AI establishes that the proposed system is the only free platform offering the complete combination of AI optimization, ATS compatibility, real-time preview, and open full-stack architecture.

Future work will focus on four primary directions: (1) integration of a job-description-based keyword matching feature that compares a pasted job description against the current resume and highlights missing keywords; (2) LinkedIn OAuth integration to allow direct profile import and eliminate manual data entry; (3) development of a cross-platform mobile application using React Native sharing the same backend API; and (4) multi-language support for Hindi and other regional Indian languages to extend platform accessibility.

ACKNOWLEDGMENT

The author sincerely thanks Dr. Manju Lata, Associate Professor, Department of Computer Science and Engineering, Raffles University, Neemrana, for her continuous guidance, valuable suggestions, and unwavering support throughout the development of this project. The author also extends gratitude to Dr. Rajendra Singh, Dean, School of Engineering and Technology, Raffles University, for providing the academic environment and resources that made this work possible.

REFERENCES

[1] Ministry of Labour and Employment, Government of India, "Annual Report 2022-23," New Delhi, 2023. [Online]. Available: https://labour.gov.in

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, May 2026 Impact Factor: 8.2

[2] Ladders Inc., "Eye-Tracking Study: How Recruiters Review Resumes," Ladders Research Report, 2018. [Online]. Available: https://www.theladders.com

[3] Jobscan, "ATS Resume Statistics and Insights," Jobscan Blog, 2022. [Online]. Available: https://www.jobscan.co/blog/ats-statistics

[4] Google LLC, "Gemini: A Family of Highly Capable Multimodal Models," Google DeepMind Technical Report, 2023. [Online]. Available: https://ai.google.dev

[5] S. Kumar and A. Sharma, "Challenges in resume creation among engineering graduates: An empirical study," International Journal of Career Management, vol. 12, no. 3, pp. 112–125, 2021.

[6] A. Patel and R. Mehta, "Comparative analysis of online resume builders: Features, ATS compatibility, and user experience," ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 1–14, 2022.

[7] J. Fuller, M. Raman, E. Bailey, and N. Vaduganathan, "Hidden Workers: Untapped Talent," Harvard Business School and Accenture Research Report, 2021. [Online]. Available: https://www.hbs.edu

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

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT 2019, pp. 4171–4186, 2019.

[10] P. Lewis, E. Perez, A. Piktus et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459–9474, 2020.

[11] H. Zhang, Y. Liu, and W. Chen, "AI-powered professional document generation using large language models," in Proc. AAAI Workshop on AI for Social Good, pp. 45–52, 2023.

[12] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in Proc. EMNLP 2019, arXiv:1908.10084, 2019.

[13] OWASP Foundation, "OWASP Top Ten Web Application Security Risks," 2021. [Online]. Available: https://owasp.org/www-project-top-ten

[14] Meta AI, "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971, 2023.