**ZERO TRUST NEXTGEN RESUME PLATFORM**



**Submitted To : Sir Waqas Ali**

**Submitted by:**

**Noor Asghar 2022-CS-140**

**Ghania Khan Niazi 2022-CS-156**

**Department of Computer Science University of Engineering and Technology, Lahore, Pakistan**

**Table of Contents**

[**Zero Trust Information Security Project Report: NextGen Resume Platform** 3](#_Toc197450547)

[**Introduction** 3](#_Toc197450548)

[**Importance of Zero Trust Security** 3](#_Toc197450549)

[**What is Zero Trust?** 3](#_Toc197450550)

[**Why It Matters in Resume Platforms:** 3](#_Toc197450551)

[**Related Work** 3](#_Toc197450552)

[**Core Security Architecture:Alignment with Zero Trust Principles** 4](#_Toc197450553)

[**Signup & Login Security Features** 4](#_Toc197450554)

[**Student Dashboard Functionalities** 5](#_Toc197450555)

[**Navigation Tabs:** 5](#_Toc197450556)

[**Check ATS: Secure Resume Analysis** 5](#_Toc197450557)

[**Feedback Form** 5](#_Toc197450558)

[**Resume History** 5](#_Toc197450559)

[6](#_Toc197450560)

[**Resume Generator** 6](#_Toc197450561)

[**Admin Panel** 7](#_Toc197450562)

[**Admin Profile Management** 7](#_Toc197450563)

[**Feedback Analytics** 7](#_Toc197450564)

[**User Statistics** 7](#_Toc197450565)

[**Backend: Flask + spaCy + AES-GCM** 7](#_Toc197450566)

[**Firestore Security Rules** 7](#_Toc197450567)

[**Future Enhancements** 7](#_Toc197450568)

[**Conclusion** 9](#_Toc197450569)

[References 9](#_Toc197450570)

# **Zero Trust Information Security Project Report: NextGen Resume Platform**

**Introduction**

**NextGen** is an AI-powered Resume Evaluation and Generation Platform that integrates **Zero zero-trust security**, modern **encryption techniques**, and **NLP-based resume analysis** to provide a secure, intelligent, and user-centric experience.This document examines how **Zero Trust Security principles** can be implemented within the NextGen Resume Platform a sophisticated, full-stack web application aimed at helping students and professionals create, enhance, and analyze their resumes. The platform leverages modern frontend technologies (**React.js**), backend services (**Python Flask**), cloud infrastructure (**Firebase**), and **AI/NLP** components **(spaC**y) to provide functionalities such as resume parsing, ATS scoring, real-time feedback, and administrative analytics. Access control that is role-based and identity-aware through Firebase Authentication.Use of **AES-GCM encryption** for **secure** resume storage and communication. Implementation of strict access policies in Firestore and safe server-side **APIs**.Separation of administrative functionalities and **real-time analytics** with regulated privileges.

**Importance of Zero Trust Security**

**What is Zero Trust?**

**Zero Trust Security** is a cybersecurity model that assumes **no implicit trust** even from inside the network. Every user, device, and action must be **explicitly verified** before access is granted.

**Why It Matters in Resume Platforms:**

* Resume data includes **sensitive personal and professional information**.
* In a world of frequent data breaches, **encrypting data and strictly controlling access** ensures user privacy.
* Helps align with modern security expectations from both users and industry standards (e.g., GDPR, HIPAA).
* Prevents internal threats by **enforcing least-privilege access** at all layers.

# 

# **Related Work**

Various platforms provide services for resume building, job matching, and resume evaluation. Nevertheless, the majority do not implement Zero Trust principles or emphasize robust security measures. Services such as Zety, Novo resume, and Resume.io offer resume templates and basic analytics, yet typically fail to include secure authentication, role-based access management, or encrypted data processing. This section contrasts these platforms with the NextGen Resume Platform to underscore the distinctiveness of its security-oriented framework.

# **Core Security Architecture: Alignment with Zero Trust Principles**

|  |  |
| --- | --- |
| Zero Trust Principle | Implementation in Platform |
| Verify Explicitly | Firebase Authentication, UID validation, Firestore access rules |
| Least Privilege Access | Scoped Firestore rules, frontend UI guards, admin-only operations |
| Assume Breach | AES-GCM encryption, input sanitization, CORS enforcement |
| Microsegmentation | Role-based React components, separate routes for users/admins |
| Contextual Access | Admin access via email verification; plans for device/session-based policies |
| Secure Data at Rest/Transit | AES-GCM encryption, HTTPS communication, secure Firebase storage |
| End-to-End Visibility | Admin dashboard with real-time updates and feedback logging |
| Logging & Error Reporting | User-friendly messages, toast notifications, console logging |

**Signup & Login Security Features**

* **Email format** validation via Regex.
* **Password complexity check**: uppercase, digit, special character.
* **No admin self-assignment**: frontend restricted from setting admin roles.
* **Friendly error messages** using err. Code.
* **Duplicate account detection**.
* **Optional terms and conditions** checkbox.
* **Config.js for secrets**: Keeps credentials out of source code.
* **Admin access** protected using hardcoded email (can be replaced with Firebase custom claims).

**Student Dashboard Functionalities**

**Navigation Tabs:**

* **Check ATS** – Resume evaluation and feedback
* **Generate Resume** – Customizable resume builder
* **History** – Tracks ATS scores and resume history
* **Feedback** – User experience input
* **Settings** – Theme toggle and customization

**Check ATS: Secure Resume Analysis**

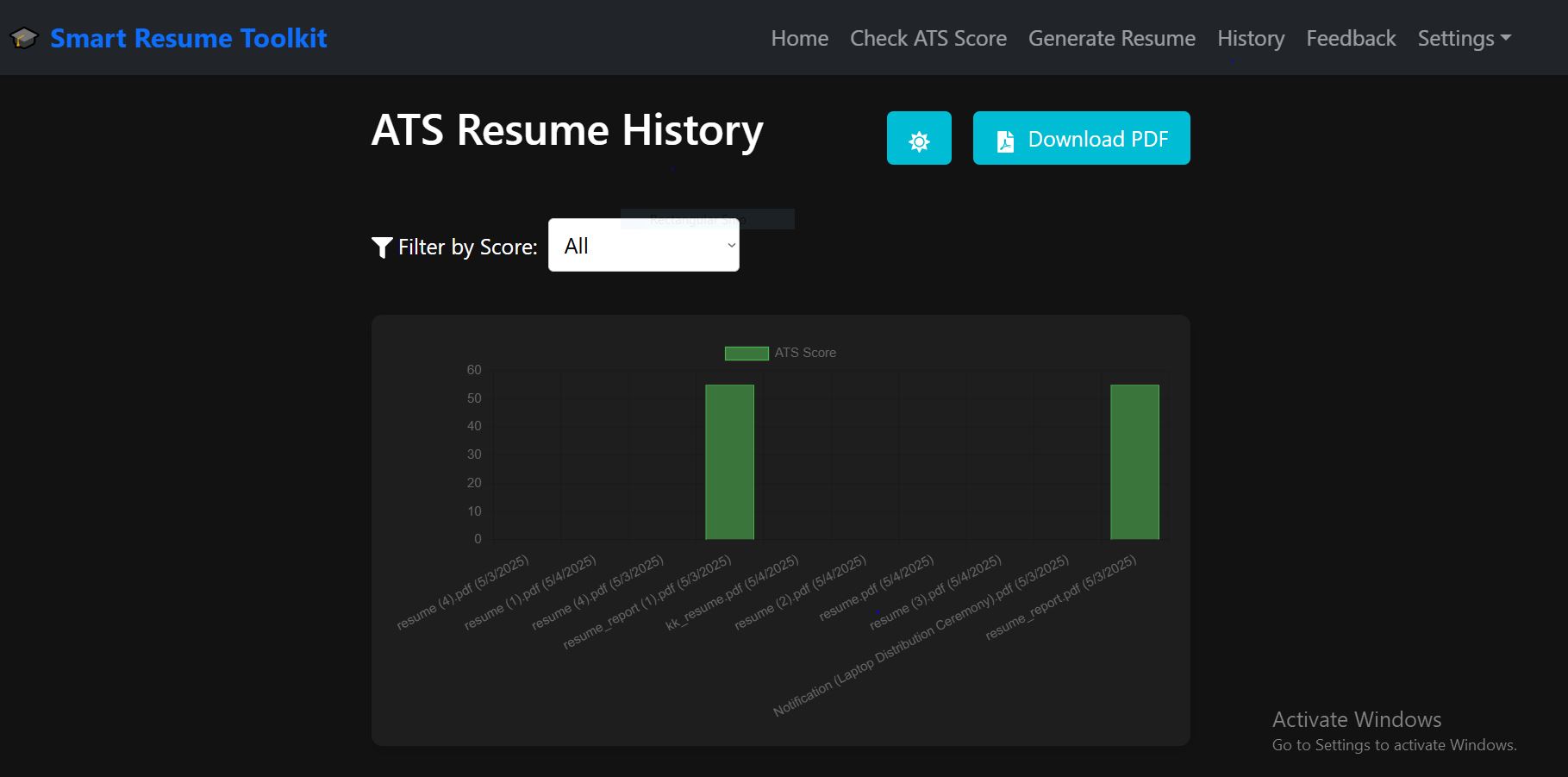
|  |  |
| --- | --- |
| **Feature** | **Details** |
| File Type & Size Validation | Only allows PDF/DOC/DOCX, max 2MB |
| Duplicate Resume Name Check | Prevents same file name uploads |
| Upload Button Disable | Prevents repeated submissions |
| Spinner + ARIA Labels | Accessibility and feedback |
| Toast Notifications | UI feedback for actions |

**Feedback Form**

* Users can submit **ratings and written feedback**.
* Data stored securely in Firestore.
* Admin analyzes responses for platform improvement.

**Resume History**

|  |  |
| --- | --- |
| Feature | Details |
| Authentication-guarded fetch | Only logged-in user sees their history |
| Score Filtering | Categorized by ATS score thresholds |
| Visualizations | Bar charts via react-chartjs-2 |
| Modal Suggestions | View improvement areas |
| Pagination | 5 items per page |
| Theme Toggle | Dark/light mode saved in local Storage |
| PDF Export | Uses jsPDF |

****

**Resume Generator**

|  |  |
| --- | --- |
| **Security Considerations** | Implementation |
| |  |  | | --- | --- | | Data Encryption |  | | |  |  | | --- | --- | |  | crypto-js AES encrypts all resume content | |
| |  |  | | --- | --- | | Input Sanitization |  | | |  |  | | --- | --- | |  | Prevents XSS and injection | |
| |  |  | | --- | --- | | Export Options |  | | |  |  | | --- | --- | |  | PDF via html2pdf.js, DOCX via docx & file-saver | |
| |  |  | | --- | --- | | Templates |  | | |  | | --- | | Multiple styles with live preview | |
| |  |  | | --- | --- | | Authentication |  | | |  |  | | --- | --- | |  | React-Firebase-Hooks for secure access | |

**Admin Panel**

**Admin Profile Management**

* View/edit profile details: name, phone, email, picture
* Real-time sync via update Profile and setDoc
* Profile data **encrypted** using crypto-js AES

**Feedback Analytics**

* Categorized feedback (positive, neutral, negative)
* Pie charts using **Recharts**

**User Statistics**

* Real-time user counts and active users
* Bar charts for trends

**Backend: Flask + Spacy + AES-GCM**

|  |  |
| --- | --- |
| **Module** | **Details** |
| Resume Parsing | spaCy extracts Name, Email, Skills, Education, Experience |
| Resume Scoring | Based on semantic presence of critical keywords |
| Feedback Suggestions | Based on missing or weak sections |
| Encryption | AES-256-GCM ensures confidentiality + integrity |
| Secure File Handling | secure filename, controlled directory |
| CORS | Only allows http://localhost:3000 |
| Error Handling | Graceful error reporting with user-friendly messages |

**Firestore Security Rules**

* Enforces **strict data isolation**.
* Only authenticated users can read/write their documents.
* Admin access scoped by **email** (upgradeable to role-based access).

# **Future Enhancements**

* **Firebase Custom Claims:** Allow detailed management of user roles.
* **OAuth with Google/GitHub:** Ensure secure authentication using single sign-on providers.
* **Cloud Functions:** Implement server-side logic to review or verify data submissions.
* **ReCaptcha v3:** Block bot attacks without adding user friendly

**Conclusion**

**NextGen** embodies the principles of modern cybersecurity by fully embracing the **Zero Trust Security model**. This project showcases how **AI, encryption, and secure design patterns** can be integrated into a practical and user-friendly platform. By eliminating assumptions of implicit trust and encrypting all sensitive data, NextGen stands out as a **secure-by-design** resume platform.

# **References**

1. Explosion AI. spacy nlp: Industrial-strength natural language processing, 2025. Available at: https://spacy.io.
2. CryptoJS Documentation. Cryptojs: Javascript cryptography library. Available at: https://github.com/brix/crypto-js.
3. Firebase Documentation. Firebase authentication and firestore docs, 2025. Available at: https://firebase.google.com/docs.
4. Firebase Security Rules Documentation. Firebase fire- store and storage security rules, 2025 , Available at: https://firebase.google.com/docs/rules.
5. PyCryptodome Documentation. Pycryptodome: A self-contained cryptographic library for python. Available at: [https://www.pycryptodome.org.](http://www.pycryptodome.org/)
6. R. Johnson and P. Lee. Automating resume screening: An ats perspective. *Journal of Human Resource Technologies*, 29(3):45–58, 2023.
7. National Institute of Standards and Technology (NIST). Aes- gcm specification (nist sp 800-38d), 2020. Available at: https://csrc.nist.gov/publications/detail/sp/800-38d/final.
8. National Institute of Standards and Technology (NIST). Sp 800-207: Zero trust architecture, 2020. Available at: https://csrc.nist.gov/publications/detail/sp/800-207/final.
9. OWASP. Owasp top 10: The ten most critical web application security risks, 2021. Available at: https://owasp.org/www-project-top-ten/.
10. Forrester Research. Zero trust extended ecosystem, 2020. Available at: https://go.forrester.com/research/.
11. The Flask Documentation Team. Flask documentation, 2025. Avail- able at: https://flask.palletsprojects.com/.