International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Secure Resume Generation and Sharing System Using AES Encryption and Django Framework

SATYANARAYANA VEERAVA LLI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

K.Venkatesh

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

In the modern digital era, resume generation and sharing have become essential components of professional communication. However, the security and privacy of personal information remain major concerns when resumes are stored and shared online. This paper presents a **secure resume generation and sharing system** developed using the Django web framework and Advanced Encryption Standard (AES) encryption. The proposed system ensures confidentiality, integrity, and controlled access to user data.The system allows users to register, log in, and manage their personal and educational information. Based on user input, a professional resume is dynamically generated using visualization libraries such as Matplotlib and OpenCV. The generated resume is stored as an image file and encrypted using AES encryption before being saved in the system. This ensures that even if unauthorized access occurs, the stored data remains protected.A key feature of the system is its secure sharing mechanism. Users can share their resumes with selected individuals, and access control is maintained through a database-driven sharing model. Only authorized users can view or download shared resumes. During download, encrypted files are decrypted securely, ensuring data privacy.The system also integrates password-based key derivation techniques using PBKDF2 to strengthen encryption keys. This enhances resistance against brute-force attacks. Additionally, the use of Django ensures a scalable and maintainable architecture with robust backend support.

The proposed solution addresses limitations in traditional resume management systems by incorporating encryption, secure storage, and controlled sharing. It is particularly useful for job seekers, organizations, and academic institutions where data confidentiality is critical.Experimental results demonstrate that the system effectively secures sensitive data while maintaining usability and performance. The combination of web technologies and cryptographic techniques makes the system reliable and efficient for real-world applications.

Keywords: Resume Builder, AES Encryption, Django Framework, Secure Data Sharing, PBKDF2, Data Privacy, Web Application Security, Image Processing, Cryptography

I. INTRODUCTION

With the rapid advancement of digital technologies, the process of creating and sharing resumes has shifted from manual methods to online platforms. Resume builders have simplified the creation of professional documents, enabling users to generate structured and visually appealing resumes efficiently. However, these platforms often lack robust security mechanisms, exposing sensitive personal information such as contact details, educational background, and professional history to potential threats.Data privacy has become a significant concern in web-based applications. Unauthorized access, data breaches, and identity theft are common risks associated with storing personal information online. Therefore, it is essential to develop secure systems that protect user data while maintaining usability and efficiency.This project introduces a **secure resume generation and sharing system** that integrates cryptographic techniques with a web-based application. The system is built using the Django framework, which provides a scalable and secure backend for handling user requests and managing data. The primary focus of the system is to ensure that user information remains confidential during storage and transmission.

To achieve this, the system employs **Advanced Encryption Standard (AES)**, a widely used symmetric encryption algorithm known for its high security and efficiency. AES encryption ensures that resume files are protected from unauthorized access. Additionally, the system uses **PBKDF2 (Password-Based Key Derivation Function 2)** to generate strong encryption keys, enhancing resistance against attacks.The application allows users to create resumes by entering their details and uploading images. The generated resumes are converted into image formats using visualization libraries, ensuring a consistent and professional layout. These files are then encrypted before being stored in the system.Another important feature of the system is its **controlled sharing mechanism**. Users can share their resumes with specific individuals, ensuring that only authorized users can access the data. This feature is particularly useful in professional environments where secure document sharing is required.The system aims to provide a balance between security and usability. By integrating encryption, authentication, and access control, it offers a reliable solution for secure resume management. This project contributes to the development of secure web applications and highlights the importance of cryptographic techniques in protecting user data.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

The development of secure resume management systems is influenced by advancements in web technologies, cryptography, and data security. Various studies have explored different approaches to secure data storage and sharing in web applications.Traditional resume builder systems primarily focus on usability and design, often neglecting security

aspects. These systems store user data in plaintext or use minimal encryption, making them vulnerable to data breaches. Research has shown that lack of encryption is one of the major causes of data leakage in web applications.To address these issues, researchers have proposed the use of cryptographic algorithms such as AES, RSA, and hashing techniques. AES is widely preferred due to its efficiency and strong security. Studies indicate that AES provides faster encryption and decryption compared to asymmetric algorithms, making it suitable for real-time applications.Password-based key derivation techniques such as PBKDF2 have also been widely studied. PBKDF2 enhances security by generating strong keys from weak passwords, reducing the risk of brute-force attacks. It is commonly used in secure systems for password storage and encryption.

In addition to encryption, secure sharing mechanisms have been explored. Access control models, such as role-based and user-specific permissions, are used to restrict data access. These models ensure that only authorized users can view or modify data.Recent advancements in web frameworks such as Django have improved the security of web applications. Django provides built-in features such as authentication, session management, and protection against common attacks like SQL injection and cross-site scripting.Image-based document generation has also gained popularity. Libraries such as Matplotlib and OpenCV are used to create visually structured documents. These tools enable dynamic content generation, improving user experience.Despite these advancements, many existing systems still lack integrated solutions that combine encryption, secure sharing, and user-friendly interfaces. This highlights the need for a comprehensive system that addresses all these aspects.The proposed system builds upon these research findings by integrating AES encryption, PBKDF2 key generation, and Django-based web development. It provides a secure and efficient solution for resume generation and sharing, addressing the limitations of existing systems.

III. EXISTING SYSTEM

Existing resume generation systems are primarily designed to simplify the process of creating professional resumes. These systems provide templates and user-friendly interfaces for entering personal and professional details. While they are effective in improving productivity, they often lack robust security mechanisms.In most traditional systems, user data is stored in databases without proper encryption. This makes sensitive information vulnerable to unauthorized access and data breaches. Additionally, resume files are usually stored in plain formats such as PDF or DOC, which can be easily accessed if the system is compromised.Another limitation of existing systems is the lack of secure sharing mechanisms. Users often share resumes via email or public links, which can be accessed by unintended recipients. This increases the risk of data leakage and misuse of personal information.Furthermore, many systems do not implement strong authentication or access control mechanisms. This allows unauthorized users to access or modify data, compromising system integrity. The absence of encryption and secure key management further weakens the overall security.Scalability and customization are also challenges in traditional systems. Most platforms provide limited flexibility in resume design and do not support dynamic content generation.Overall, existing systems focus more on functionality and ease of use rather than security. This creates a need for a

secure and efficient solution that protects user data while maintaining usability. The proposed system addresses these limitations by incorporating encryption, secure storage, and controlled sharing mechanisms.

IV. PROPOSED METHOD

The proposed system is a **secure resume generation and sharing platform** developed using the Django framework and integrated with cryptographic techniques such as **Advanced Encryption Standard (AES)** and **PBKDF2 key derivation**. The primary objective of the system is to ensure the confidentiality, integrity, and controlled accessibility of user-generated resumes.In this system, users can register, log in, and input their personal, educational, and professional details. Based on this data, a structured resume is dynamically generated using visualization tools such as Matplotlib and OpenCV. The generated resume is saved as an image file to maintain formatting consistency.To enhance security, the system encrypts resume files using AES encryption before storing them on the server. AES is widely recognized for its strong security and efficiency in protecting sensitive data . Additionally, PBKDF2 is used to derive secure encryption keys from user-defined passwords, increasing resistance to brute-force attacks and improving overall data protection .The system also incorporates a **secure sharing mechanism**, allowing users to share their resumes with selected individuals. Access control ensures that only authorized users can view or download shared resumes. During download, encrypted files are decrypted securely, maintaining data confidentiality.Furthermore, the Django framework provides a secure and scalable backend with built-in protections against common web vulnerabilities . The integration of encryption, authentication, and controlled sharing makes the system highly reliable.Overall, the proposed system enhances traditional resume management by providing a secure, user-friendly, and efficient platform for generating and sharing resumes.

V. IMPLEMENTATION

The implementation of the proposed system is carried out using **Python programming language**, the **Django web framework**, and supporting libraries such as PyMySQL, Matplotlib, OpenCV, and cryptographic modules like pyaes and PBKDF2.The system is divided into multiple modules including user authentication, profile management, resume generation, encryption, and secure sharing. Initially, users register and log in using credentials stored in a MySQL database. Django handles HTTP requests, session management, and template rendering.Once authenticated, users can input their details such as personal information, education, skills, and projects. These inputs are processed and used to dynamically generate resumes. The system uses Matplotlib to design structured resume layouts and OpenCV for image processing and resizing.A key component of the implementation is **data encryption**. Before storing resumes, the system encrypts the generated image files using AES encryption. AES is widely used due to its high performance and strong security guarantees . The encryption key is derived using

PBKDF2, which strengthens the key by applying multiple iterations and salting techniques.The encrypted resumes are stored in the server directory, ensuring that even if unauthorized access occurs, the data remains unreadable. During download, the system decrypts the file and sends it to the user as an HTTP response.

The system also supports **resume sharing functionality**, where users can select other users to share their resumes. The sharing information is stored in a database table, and access control mechanisms ensure that only authorized users can view shared resumes.Additionally, validation mechanisms are implemented to ensure data integrity and prevent unauthorized access. Django's built-in security features help protect against vulnerabilities such as SQL injection and cross-site scripting.Overall, the implementation demonstrates the integration of web technologies and cryptographic techniques to build a secure and efficient application. The system ensures that user data is protected at every stage, from input to storage and sharing.

VI. ALGORITHMS

The proposed system uses several algorithms to ensure secure and efficient operation:

1. AES Encryption Algorithm

AES is a symmetric encryption algorithm used to secure resume files. It encrypts data using a secret key and ensures confidentiality. AES is widely adopted due to its strong resistance to cryptographic attacks and efficient performance in real-time applications .

2. PBKDF2 Key Derivation Algorithm

PBKDF2 generates a secure encryption key from a password by applying hashing multiple times with a salt value. This increases computational complexity, making brute-force attacks difficult. Recent studies show that adaptive PBKDF2 significantly improves security and entropy preservation .

3. Resume Generation Algorithm

This algorithm collects user input and formats it into a structured resume layout using Matplotlib. Text formatting, image insertion, and layout design are handled dynamically.

4. Encryption–Decryption Workflow

- Input data is converted into binary format
- AES encryption is applied
- Encrypted file is stored
- During download, AES decryption restores the original file

5. Access Control Algorithm

The system verifies whether a user is authorized to access a shared resume by checking database records.

6. File Handling Algorithm

Handles file storage, retrieval, and conversion between formats.

These algorithms collectively ensure secure data handling, efficient processing, and controlled access.

VII. SYSTEM DESIGN

The system follows a **modular layered architecture** consisting of frontend, backend, database, and security layers.

1. Architecture Overview

The system includes the following layers:

- **Presentation Layer:** Django templates for user interaction
- **Application Layer:** Handles business logic
- **Security Layer:** AES encryption and PBKDF2 key generation
- **Data Layer:** MySQL database and file storage

2. Modules

a) User Module

Handles registration, login, and session management. User data is stored securely in the database.

b) Profile Management Module

Allows users to update personal and educational details.

c) Resume Generation Module

Generates resumes dynamically using Matplotlib and OpenCV.

d) Encryption Module

Encrypts resume files using AES before storage.

e) Sharing Module

Manages resume sharing between users using access control mechanisms.

f) Download Module

Decrypts and provides resume files to authorized users.

3. Workflow

1. User registers and logs in
2. User enters resume details
3. Resume is generated as an image
4. File is encrypted using AES
5. Encrypted file is stored
6. User shares resume with others
7. Authorized users can view/download

4. Security Design

- AES ensures data confidentiality
- PBKDF2 strengthens encryption keys
- Django prevents web vulnerabilities

Modern frameworks like Django continuously improve security by addressing vulnerabilities and enforcing best practices .

5. Data Flow

User input → Django processing → Resume generation → Encryption → Storage → Decryption → Output

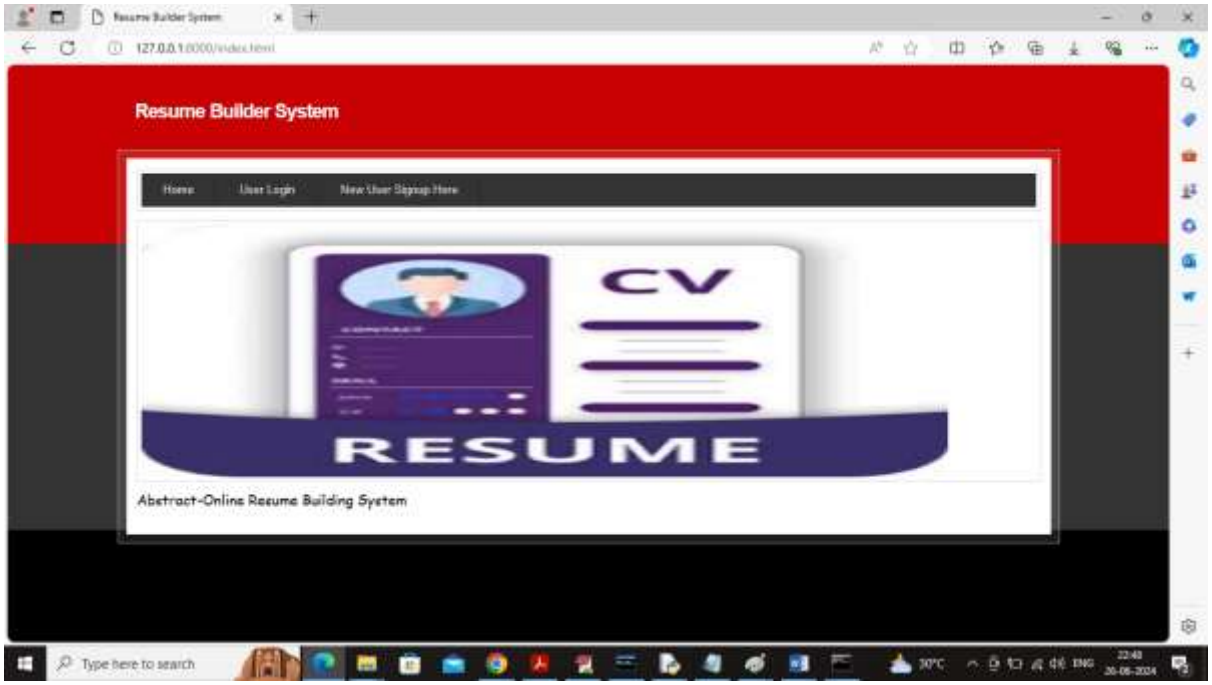
6. Advantages

- Secure data storage
- Controlled sharing
- User-friendly interface
- Scalable architecture

International Journal of Engineering Research and Science & Technology

SYSTEM DESIGN IMAGES

http://127.0.0.1:8000/index.html and press enter key to get below page

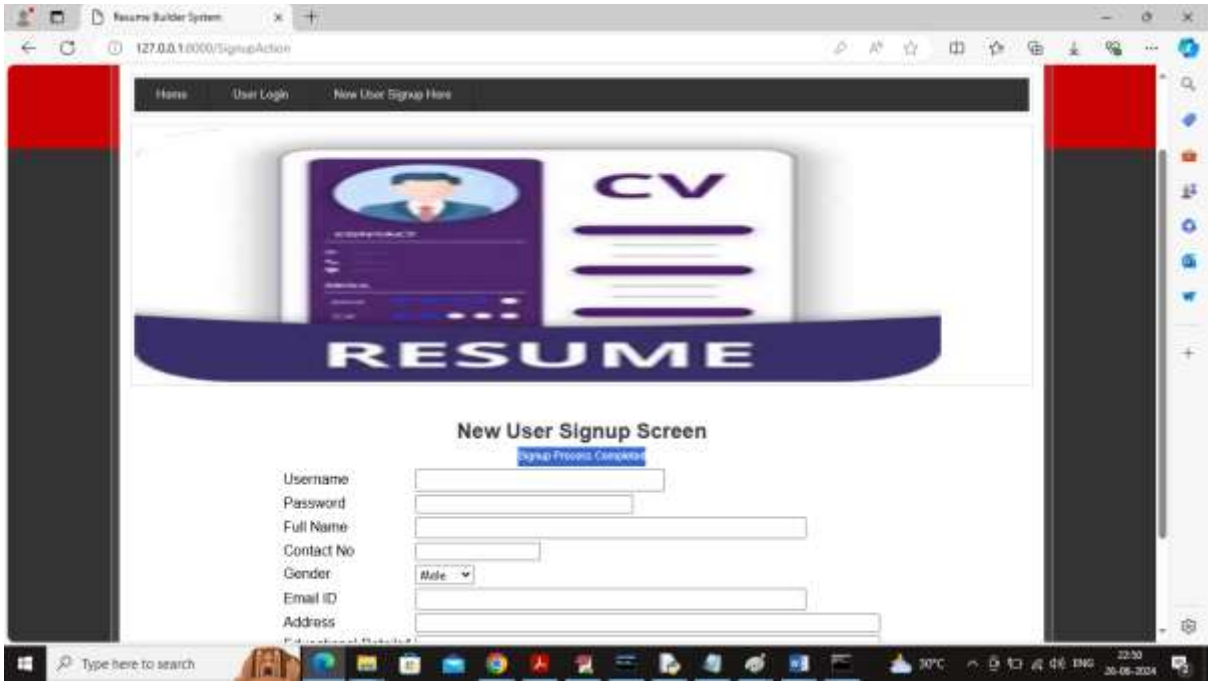


In above screen user can click on 'New User Sign up' link to get below page

International Journal of Engineering Research and Science & Technology

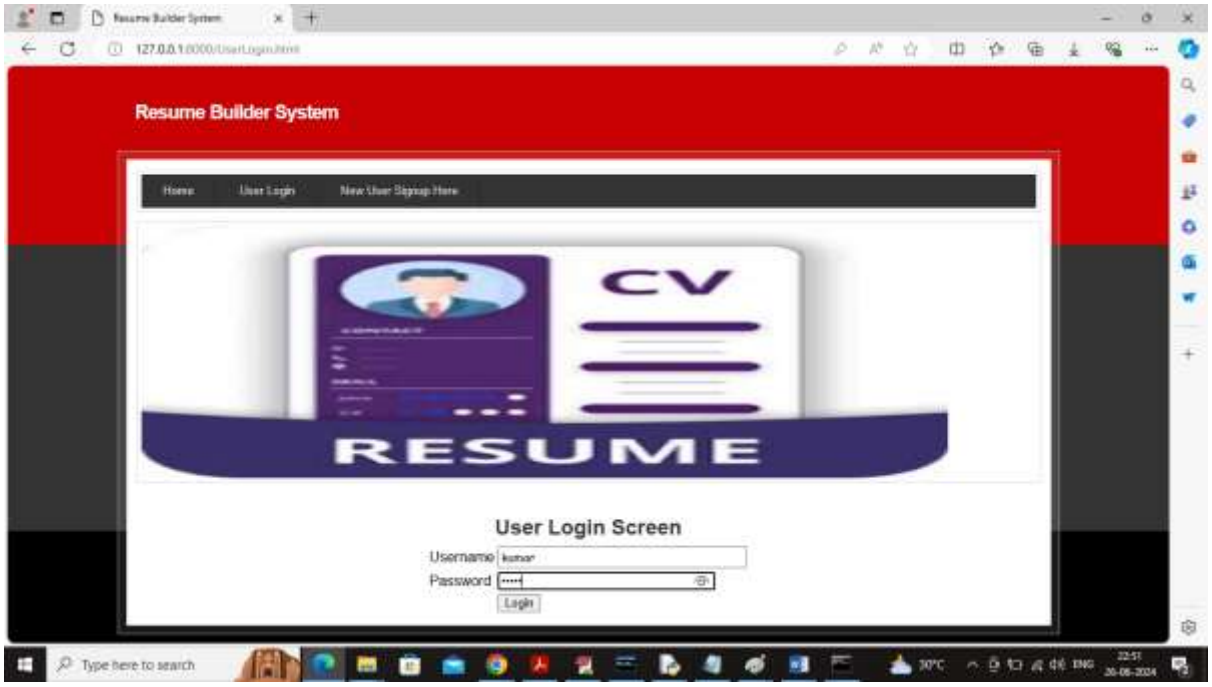


In above screen user is entering sign up profile details and then press button to save all user profile details to database and this details will be used to create user resume and user can update above details by login to system



In above screen user sign up process completed and now click on 'User Login' link to get below login screen

International Journal of Engineering Research and Science & Technology

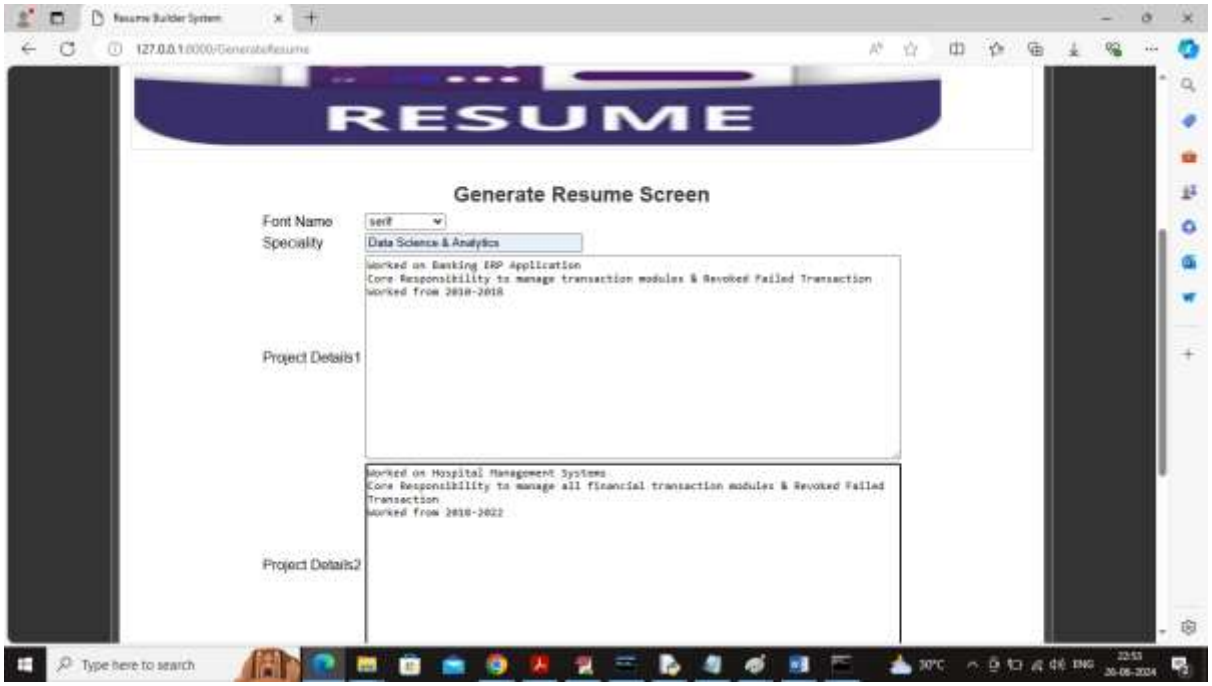


In above screen user is login and after login will get below page

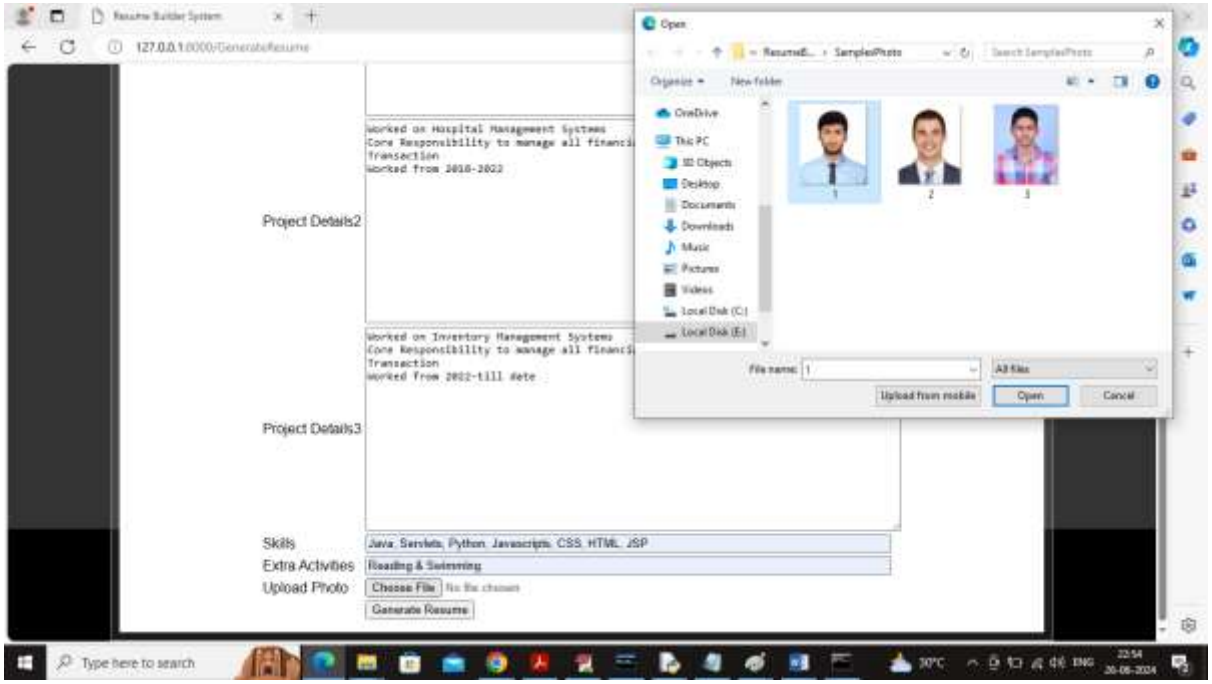


International Journal of Engineering Research and Science & Technology

In above screen user can view existing profile details and can update desired details and then press button to get below

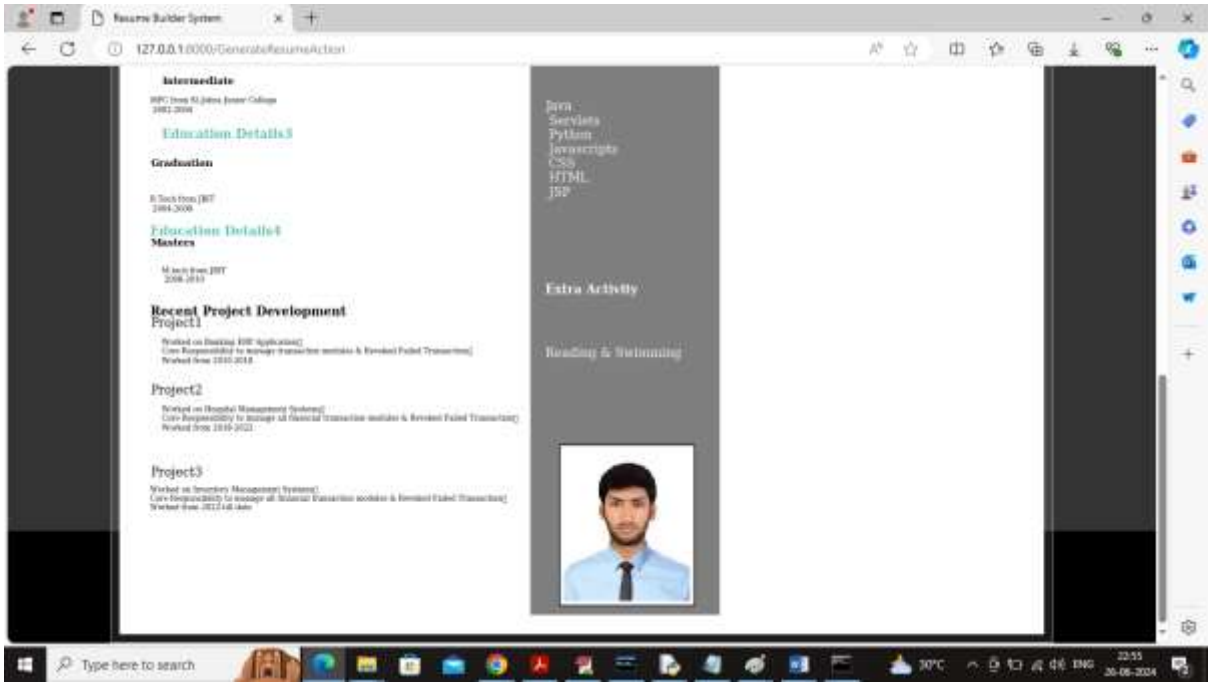


In above screen user will enter work experience details

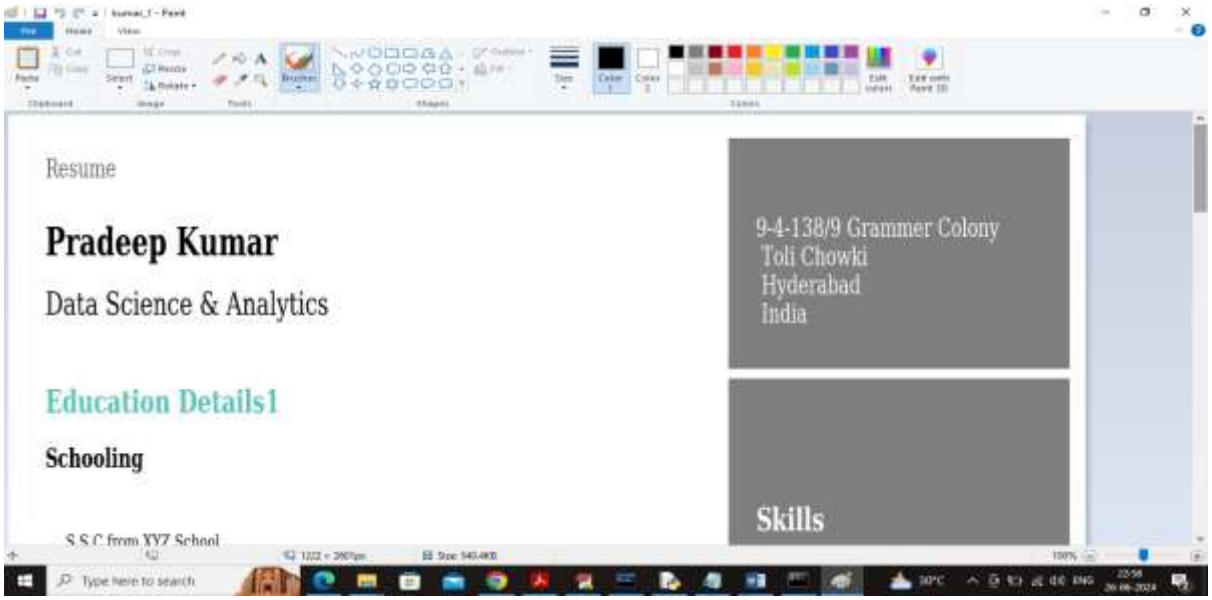


International Journal of Engineering Research and Science & Technology

In above screen user will enter all resume details and then upload photo and then click on 'Generate Resume' button to generate resume and can view preview like below screen



In above screen user resume created and can click on 'View Past Resume' link to view all generated resumes and get below page



In above screen user can view resume in Paint tool and in below screen can see all resumes are saved at server in encrypted format

Similarly by following above screens you can generate resumes

VIII. CONCLUSION

The proposed system successfully demonstrates a secure and efficient approach to resume generation and sharing using modern web technologies and cryptographic techniques. By integrating AES encryption and PBKDF2 key derivation with the Django framework, the system ensures strong protection of sensitive user data.The system addresses major limitations of traditional resume management platforms by providing encrypted storage and controlled sharing. AES encryption ensures that stored resumes remain confidential, while PBKDF2 enhances key security and resistance against brute-force attacks. Research confirms that combining AES with strong key derivation significantly improves data protection in modern applications .The use of Django provides a robust and scalable backend, enabling efficient handling of user requests and secure session management. The system's modular design allows easy extension and integration of additional features.Despite its advantages, challenges such as scalability and advanced key management remain areas for future improvement. Emerging technologies such as hybrid encryption and adaptive security models can further enhance system performance and security.Future enhancements may include cloud integration, mobile support, and advanced authentication mechanisms. Incorporating newer cryptographic techniques such as zero-knowledge proofs can further strengthen privacy.In conclusion, the system provides a practical solution for secure resume management and highlights the importance of integrating cryptographic techniques in web applications.

REFERENCES

1. Rosandi et al., 2026. *AES-256 File Encryption Application*
2. AlQahtani, 2025. *Dynamic PBKDF2 Model for Cryptographic Systems*
3. Springer, 2025. *Advanced Encryption Standard Survey*
4. Django Security Team, 2026. *Web Application Security Updates*
5. Darmawan et al., 2025. *Zero-Knowledge Encryption Systems*
6. Ahmad et al., 2025. *Hybrid Cryptographic Approaches*
7. NIST, 2023. *FIPS PUB 197: AES Standard*
8. NIST, 2010. *PBKDF2 Recommendation (SP 800-132)*
9. Stallings, 2023. *Cryptography and Network Security*
10. Ganesh, 2025. *Survey on AES Security*
11. Bayesh et al., 2025. *Dual-Layer AES Image Encryption Framework*
12. Mozo, 2025. *Quantum-Classical Hybrid Encryption Framework*

13. Chahar, 2025. *Future Trends in Cryptography*
14. Sharma et al., 2024. *Secure Hashing Mechanisms*
15. Xu, 2024. *AES Implementation in Modern Systems*