





PROJECT AND TEAM INFORMATION**PROJECT TITLE:**

SECUREVAULT: A File Encryption and Decryption System

STUDENT / TEAM INFORMATION:

Team Name	The Mavericks
Team member 1 (Team Lead) (Last Name, name: student ID: email, picture):	Pokhriyal, Anuj – 220113037 anujpokhriyal6@gmail.com 
Team member 2 (Last Name, name: student ID: email, picture):	Kumar, Samrat- 220111019 samratkumatchauhan@gmail.com 
Team member 3 (Last Name, name: student ID: email, picture):	Safdar, Zoya – 22152649 zoyasafdar136@gmail.com 

Team member 4 (Last Name, name: student ID: email, picture):	Pandey,Anshu – 220113001 anshupandey968@gmail.com 
--	--

PROPOSAL DESCRIPTION

MOTIVATION:

With increasing cyber threats and data breaches, sensitive files stored on local or cloud storage are vulnerable to unauthorized access. Individuals and businesses need a reliable, user-friendly tool to protect their confidential data from cybercriminals, unauthorized modifications, and accidental exposure.

Why is it important?

- Privacy Protection: Prevents unauthorized access to personal and corporate files.
- Compliance with Regulations: Meets security standards such as GDPR, HIPAA, and PCI-DSS.
- Secure File Sharing: Ensures that only intended recipients can access encrypted files.
- Data Integrity: Prevents data tampering or corruption.

State of the Art / Current solution:

Currently, encryption solutions exist, but they come with limitations:

1. BitLocker (Windows), VeraCrypt, and AxCrypt – Strong encryption but limited cross-platform compatibility.
2. Cloud-Based Encryption Services – Secure but depend on third-party storage providers.
3. Command-line encryption tools (e.g., OpenSSL, GnuPG) – Powerful but complex for non-technical users.

How Our Solution Improves on Existing Ones:

- >Cross-platform support (Windows, Linux, macOS).
- > User-friendly web-based interface (No complex commands required).
- > Multi-layer security (AES for file encryption, RSA for key management).
- > Secure local storage & cloud integration (Optional).

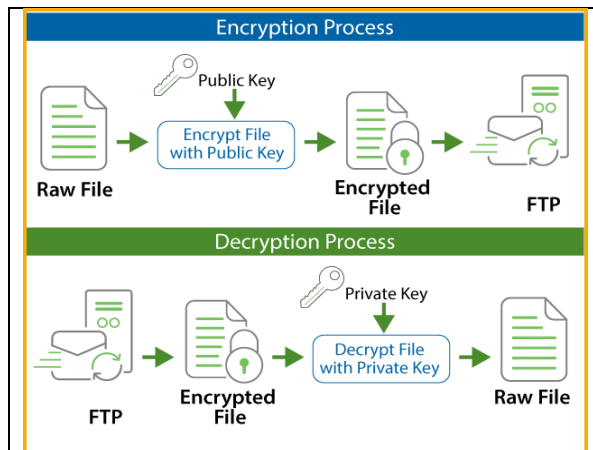
Project Goals and Milestones:

<input type="checkbox"/> Develop a secure, efficient, and user-friendly file encryption and decryption tool.
<input type="checkbox"/> Implement AES-256 encryption for file security.
<input type="checkbox"/> Use RSA encryption for secure key exchange and management.
<input type="checkbox"/> Develop a web-based UI for accessibility.
<input type="checkbox"/> Ensure cross-platform compatibility and OS-level file security.

Project Approach:

Component	Technology	Purpose
Backend (Encryption & File Processing)	Python (Django, PyCryptodome)	Handles encryption, decryption, and secure file processing.
Frontend (Web UI)	HTML, CSS, JavaScript (Django Templates, Bootstrap)	Provides a user-friendly interface for file uploads and management.
Database (For Metadata & User Management)	PostgreSQL / SQLite	Stores encrypted file metadata, user credentials, and encryption keys securely.
Security Algorithms	AES-256 (File Encryption), RSA (Key Management), SHA-256 (Integrity Check)	Ensures data confidentiality, integrity, and secure key management.
Operating System Integration	Python OS Module (os, shutil), File Permissions (chmod, chown in Linux)	Manages secure file handling, storage, and access permissions.

System Architecture (High Level Diagram):



Project Outcome / Deliverables:

A fully functional File Encryptor & Decryptor with:

- ✓ Encryption & Decryption Functionality
- ✓ Secure Key Management
- ✓ Web-based User Interface
- ✓ OS-Level Security Features
- ✓ Performance Optimization for Large Files
- ✓ User Guide & Documentation.

Assumptions:

- ☐ The user has the necessary decryption key to access encrypted files.
- ☐ The encryption process does not degrade file integrity or performance.
- ☐ The system ensures strong security without compromising ease of use.

References:

- ☐ AES-256 Encryption: <https://www.geeksforgeeks.org>
- ☐ RSA Cryptography: [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))
- ☐ "Django for Beginners" by William S. Vincent.