# Software Requirements Specification (SRS) for Secure File Sharing

## 1.Introduction

### 1.1 Document Purpose

The Software Requirements Specification (SRS) document outlines the requirements for the "Secure File Sharing" web application. It defines the functional and non-functional needs for developers, testers, and project managers who will contribute to the development lifecycle of this application.

### 1.2 Product Scope

The "Secure File Sharing" platform provides a secure method for users to upload, encrypt, and share files using time-limited encrypted links. It aims to protect sensitive data against unauthorized access by implementing end-to-end encryption and token-based download verification. The platform primarily serves students, educators, and professionals who require secure data sharing capabilities.

### 1.3 Document Overview

This document includes the following sections:

- Section 2: Product overview.
- Section 3: Comprehensive definition of specific requirements.
- Section 4: References.

### 1.4 Definitions, Acronyms, and Abbreviations

AES: Advanced Encryption Standard
API: Application Programming Interface
DB: Database
UI: User Interface

## 2. Overall Description

### 2.1 Product Perspective

"Secure File Sharing" is an independent web-based application that enables encrypted file uploads and controlled sharing through secure links. It integrates frontend technologies (HTML, CSS, JavaScript) with a backend server built using Node.js and Express.js, connected to a MongoDB database.

### 2.2 Product Functions

- User registration and login authentication via Passport.js.
- Upload interface for users to submit files for encryption.
- AES-based encryption of uploaded files.
- Token-based generation of secure links.

- Decryption of files with valid tokens.
- Expiration management.
- Error handling.
- User dashboard.

### 2.3 User Characteristics

- End Users: Students, teachers, business professionals.
- Administrators: Backend administrators for maintenance.

### 2.4 Constraints

- Must use AES-256 encryption.
- File size limit: 100MB.
- HTTPS-only access.
- 99.5% uptime requirement.

### 2.5 Assumptions and Dependencies

- Active internet connection.
- Server uses HTTPS.
- MongoDB configured securely.
- Frontend connects to APIs properly.

## 3. Specific Requirements

### 3.1 External Interfaces

- User Interface: HTML, CSS, JS frontend.
- Database Interface: MongoDB.
- Security Interfaces: AES encryption, Passport.js authentication.
- API Services: RESTful APIs.

### 3.2 Functional Requirements

- User registration and login.
- AES file encryption.
- Token-based secure downloads.
- Token expiration and validation.
- MongoDB file metadata storage.
- Error messages for expired/invalid tokens.

### 3.3 Non-Functional Requirements

- Performance: Upload/download under 5s for 50MB.
- Security: HTTPS, AES encryption.
- Usability: Simple interface.
- Scalability: 500 concurrent users.
- Maintainability: Modular backend.

## 4. Supporting Information

### 4.1 References

1. Z. Chen and Y. Zhao, “Secure Data Sharing with Encryption Techniques,” 2020 IEEE 5th International Conference on Big Data Analytics (ICBDA).
Available at: https://ieeexplore.ieee.org/document/9124623

2. S. Turner, “Best Practices for Secure File Transfer,” SANS Institute Research Paper, 2022.Available at: https://www.sans.org/white-papers/secure-file-transfer-best-practices/

3. M. Nabil et al., “Secure File Storage in Cloud using AES Encryption,” International Journal of Computer Applications (IJCA), 2021.
Available at: https://www.ijcaonline.org/archives/volume178/number15/nabil-2019-ijca-917423.pdf

Member Contributions:

| Team Members | Contributions |
|---|---|
| Dhruvil Bhanderi | Drafted Section 1 (Introduction) and contributed to Section 2.1 (Product Perspective). |
| Kliona Kennet | Drafted Section 2 (Product Functions, User Characteristics, Constraints, and Assumptions). |
| Vedang Kathiriya | Drafted Section 3 (Specific Requirements) including External Interfaces and Functional Requirements. |
| Binul Bijo | Drafted Section 3.3 (Non-Functional Requirements) and Section 4 (Supporting Information - References) and conducted final document review and formatting. |