

**Musanze, August 2025**

B.P. 155

Ruhengeri

Rwanda

**T** : (+250) 788 90 30 30

  : (+250) 788 90 30 32

**W** : [www.ines.ac.rw](http://www.ines.rw)

**E** : info@ines.ac.rw

***Scientia et Lux***

**By: NYIRANSENGIYUMVA Angelique**

**Reg. No: 23/22281**

**Supervisor: Dr. NIRERE Adria**

**Co-Supervisor: Dr. NTEZIRIZA NKERABAHIZI Josbert**

A dissertation presented in partial fulfillment of requirements for The award of a bachelor's degree with honors of Science in computer science, option of network engineering.

**DESIGN AND IMPLEMENTATION OF PRIVATE LOCAL AREA NETWORK FOR FILE SHARING SYSTEM VIA HYPERTEXT TRANSFER PROTOCOL SECURE (HTTPS)**

**Case study: INES-Ruhengeri**

**FACULTY OF SCIENCES AND INFORMATION TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE**

**OPTION OF NETWORK ENGINEERING**

**INSTITUT D’ENSEIGNEMENT SUPÉRIEUR DE RUHENGERI**

***Accredited by Ministerial Order N° 005/2010/MINEDUC of 16 June 2010***

# DECLALATION OF ORIGINALITY

I do hereby declare that work presentation within this dissertation is my own contribution to the best of my knowledge. The same work has never been submitted to any other university or institution. I, therefore declare the work is my own for the partial fulfilment of the award of a Bachelor’s degree with honors in Computer Science Department, Network Engineering option at INES-Ruhengeri University.

The candidate name: **NYIRANSENGIYUMVA Angelique**

Signature of the candidate: ...............................................

Date of submission: ...........................................................

# APPROVAL

This is to certify that this dissertation entitled “Design and implementation of private local area network for file sharing system via Hyper-Text Transfer Protocol Secure (HTTPS)” is an original study conducted by NYIRANSENGIYUMVA Angelique under my supervision.

The supervisor’s name: **Dr. NIRERE Adria**

Signature of the supervisor: ……………………..

Date of submission: ………………………..

# DEDICATIONS

**ACKNOWLEGEMENTS**

The study is conducted through the contribution of several people who has spent their time helping me to carry out this study. For that, words are not enough to express my deepest appreciation to the almighty God to whom I owe my living; the deep sense of gratitude is due to Dr. NIRERE Adria, supervisor of this study for his valuable guidance, collaboration and constructive suggestions which helped me to come to the successfully completion of this study my sincere thanks are conveyed to INES-Ruhengeri staff, particularly the applied fundamental sciences, computer science department lecturers to providing professional supports at various successful stages during the studies program and for facilitating me to carry out this research and keep guidance. Also, special thanks go to my parents, parent’s friend heartfelt gratitude to my loving parent and fiends for financial, moral successful emotional support, my classmates and friend who helped in various ways for the successful completion of this work, God bless you all.

# ABSTRACT

While digital file management systems are essential in modern organizations, many fail to adequately protect sensitive information, especially when users have excessive permissions or when unauthorized access occurs. Understanding how to limit access without reducing usability is crucial for improving both security and efficiency. However, current systems often overlook the need for fine-grained control that prevents unauthorized downloading or saving of files while still enabling collaborative work. The main objective of this study was to design and implement a secure file access system that uses role-based access control to allow file uploading, viewing, and editing while restricting downloading and saving actions. This project employed a case study methodology that involved system analysis, prototype development, and real-time testing. The system was built with HTTPS encryption and included file-sharing capabilities tailored to users or departments. The findings revealed that users could effectively collaborate and manage documents without compromising data security. The role-based access model proved effective in limiting access based on privileges while ensuring usability through a web-friendly interface. These results suggest that institutions should adopt similar models to improve data protection, foster user accountability, and support secure internal collaboration among departments and users.

**Keywords:** secure file access, role-based access control, data confidentiality, user permissions, HTTPS encryption.

**Table of Contents**

[DECLALATION OF ORIGINALITY i](#_Toc204080471)

[APPROVAL ii](#_Toc204080472)

[DEDICATIONS iii](#_Toc204080474)

[ABSTRACT v](#_Toc204080475)

[CHAPTER 1: GENERAL INTRODUCTION 1](#_Toc204080476)

[1.1 Background of the study 1](#_Toc204080477)

[1.2 Problem statement 2](#_Toc204080478)

[1.3 Research objectives 2](#_Toc204080479)

[1.4 Research questions 3](#_Toc204080480)

[1.5 Research hypotheses 4](#_Toc204080481)

[1.6 Choice of the study 4](#_Toc204080482)

[1.7 Significance of the study 4](#_Toc204080483)

[1.8 Delimitation of the study 5](#_Toc204080484)

[1.9 Research methodology 5](#_Toc204080485)

[1.10 Organization of the study 5](#_Toc204080486)

[1.11 Summary 5](#_Toc204080487)

[CHAPTER 2: LITERATURE REVIEW 7](#_Toc204080488)

[2.1 Introduction 7](#_Toc204080489)

[2.2 Definition of key terms 7](#_Toc204080490)

[2.2.1 Hypertext transfer protocol secure (HTTPS) 7](#_Toc204080491)

[2.2.2 Local Area Network (LAN) 8](#_Toc204080492)

[2.2.3 Transport Layer Security (TLS) 8](#_Toc204080493)

[2.2.4 Role-Based Access Control (RBAC) 8](#_Toc204080494)

[2.2.5 File sharing system 9](#_Toc204080495)

[2.2.6 Data confidentiality 9](#_Toc204080496)

[2.2.7 Access control mechanisms 10](#_Toc204080497)

[2.3 Related works 10](#_Toc204080498)

[2.4 Final reflection 13](#_Toc204080499)

[2.5 Summary 14](#_Toc204080500)

[CHAPTER 3: METHODOLOGY AND MATERIALS 15](#_Toc204080501)

[3.1 Introduction 15](#_Toc204080502)

[3.2 Case study 15](#_Toc204080503)

[3.3 Data collection 16](#_Toc204080504)

[3.3.1 Techniques used to collect data 16](#_Toc204080505)

[3.3.2 Documentation 16](#_Toc204080506)

[3.3.3 Observation 16](#_Toc204080507)

[3.3.4 Interview 16](#_Toc204080508)

[3.4.1 System requirements 17](#_Toc204080509)

[3.4.2 Functional requirements 17](#_Toc204080510)

[3.4.3 Non-functional requirements 18](#_Toc204080511)

[3.5 Hardware requirements 18](#_Toc204080512)

[3.6 Software requirements 18](#_Toc204080513)

[3.7 System development methodology 19](#_Toc204080514)

[CHAPTER4: Design and implementation 22](#_Toc204080515)

[4.1 Introduction 22](#_Toc204080516)

[4.2 System architecture 22](#_Toc204080517)

[4.2.1 Overview of system architecture 22](#_Toc204080518)

[4.2.2 HTTPS LAN File Sharing System Architecture Diagram 23](#_Toc204080519)

[4.3 Flowchart diagram 24](#_Toc204080520)

[4.4 System implementation 25](#_Toc204080521)

[4.4.1 Admin dashboard implementation 26](#_Toc204080522)

[4.4.2 User management interface 26](#_Toc204080523)

[4.4.3 File sharing interface 27](#_Toc204080524)

[4.4.4 Upload interface panel 28](#_Toc204080525)

[4.4.5 Document creation panel 29](#_Toc204080526)

[4.4.6 System landing page 30](#_Toc204080527)

[4.4.7 User login interface 31](#_Toc204080528)

[4.4.8 Starred files panel 32](#_Toc204080529)

[4.4.9 Main dashboard overview 33](#_Toc204080530)

[4.4.10 File viewer interface 34](#_Toc204080531)

[4.5 Summary 35](#_Toc204080532)

[CHAPTER 5: CONCLUSION AND RECOMMENDATIONS 36](#_Toc204080533)

[5.1 Conclusion 36](#_Toc204080534)

[References 37](#_Toc204080535)

[APPENDICES 41](#_Toc204080536)

**LIST OF FIGURES**

[**Figure 1:** HTTPS 7](#_Toc204150542)

[**Figure 2:** Local Area Network (LAN) 8](#_Toc204150543)

[**Figure 3:** INES Ruhengeri 15](#_Toc204150544)

[**Figure 4:** Agile model 20](#_Toc204150545)

[**Figure 5:** HTTPS-secure LAN file haring system architecture 23](#_Toc204150546)

[**Figure 6:** HTTPS-Secure LAN file sharing system flowchart 25](#_Toc204150547)

[**Figure 7:** View of admin panel 26](file:///D:\temp\NYIRANSENGIYUMVA%20Angelique%2023%2022281%20corrected.docx#_Toc204150548)

[**Figure 8:** Admin panel user management interface 27](#_Toc204150549)

[**Figure 9:** User interface shared files panel 28](#_Toc204150550)

[**Figure 10:** User interface upload panel 29](#_Toc204150551)

[**Figure 11:** Create document panel 30](#_Toc204150552)

[**Figure 12:** System homepage 31](#_Toc204150553)

[**Figure 13:** Secure share -login interface 32](#_Toc204150554)

[**Figure 14:** Starred files section 33](#_Toc204150555)

[**Figure 15:** Main user dashboard 34](#_Toc204150556)

[**Figure 16:** File preview interface (read-only viewer) 35](#_Toc204150557)

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS**  **ACL**  **API**  **CRUD**  **DB**  **HTTPS**  **IP**  **IT**  **RBAC**  **SQL**  **UI**  **URL** | **ACRONYMS**  Access Control List  Application Programming Interface  Create, Read, Update, Delete  Database  Hypertext Transfer Protocol Secure  Internet Protocol  Information Technology  Role-Based Access Control  Structured Query Language  User Interface  Uniform Resource Locator |
|  |  |

##### LIST OF APPENDICES

[**Appendix A:** Flask app a](#_Toc204080080)

[**Appendix B:** Questions for data collection d](#_Toc204080081)

# CHAPTER 1: GENERAL INTRODUCTION

## Background of the study

The digital transformation of educational institutions has revolutionized how information is shared and accessed within organizational environments. In academic settings like Ines-Ruhengeri, sensitive information including research materials, financial records, student data, and personnel information requires secure exchange mechanisms. However, conventional file sharing methods such as email attachments, USB drives, and public cloud services frequently present significant security vulnerabilities, including data interception, unauthorized access, and potential information leakage (Nissim et al., 2017; Santhoshkumar et al., 2024). During routine operations, these insecure channels expose critical institutional data to both internal and external threats, characterized by weak encryption, inadequate authentication mechanisms, and exposure to network-based attacks (Santhoshkumar et al., 2024). These issues often lead to confidentiality breaches, compromise of data integrity, and diminished trust in institutional information systems.

Various traditional file sharing solutions have been implemented to address these concerns, such as network folder sharing and third-party cloud storage services (Abwnawar, 2020). However, they typically lack adequate security controls specific to organizational boundaries, often failing to prevent unauthorized access and data exfiltration (Aldossary & Allen, 2016). Addressing these challenges requires a more sophisticated and security-focused approach to institutional file sharing systems. One promising approach involves establishing a private, contained file sharing environment that leverages strong encryption protocols while restricting access to authorized users within organizational network boundaries (Ngwenya et al., 2024).

A potential solution lies in implementing Hypertext Transfer Protocol Secure (HTTPS) within a controlled Local Area Network (LAN) environment, a technology that provides encrypted communications and secure identification of network servers (Cangialosi et al., 2016). HTTPS employs Transport Layer Security (TLS) to encrypt all data transferred between clients and servers, offering a robust mechanism to protect file transfers and prevent unauthorized interception within institutional networks (Sakthivel & Ramalingam, 2025).

By implementing HTTPS within a private LAN configuration, it becomes possible to establish a secure file sharing ecosystem that restricts access to authorized organizational users while encrypting all data transmissions, thereby preventing confidentiality breaches and maintaining system integrity (Abwnawar, 2020; Ngwenya et al., 2024). This approach not only enhances data security during file transfers but also optimizes access control across the organizational network infrastructure. Specifically, in the context of this research, HTTPS-secured LAN file sharing have been explored as a comprehensive solution for implementing a secure file exchange framework that enforces encryption, authentication, and authorization while maintaining accessibility for legitimate users. The system monitors access patterns and enforce security policies in real-time, ensuring that sensitive institutional data remains protected from both internal and external threats (Aslan et al., 2023).

Furthermore, the system integrates role-based access control mechanisms, allowing administrators to define granular permissions based on organizational roles and responsibilities, reducing the risk of unauthorized access to sensitive files. This programmable security framework allows the institution to adapt to evolving security requirements while ensuring resources remain accessible to authorized personnel. The expected outcome of this study is a robust HTTPS-secured LAN file sharing system capable of maintaining data confidentiality, integrity, and availability while providing seamless access to authorized users. The system dynamically authenticates users and encrypt communications, reducing the probability of data breaches and significantly improving the security, reliability, and privacy of critical institutional data.

## 1.2 Problem statement

Digital file sharing has become essential in academic institutions such as INES-Ruhengeri, facilitating efficient collaboration among students, lecturers, and administrative departments. However, the institution currently depends on traditional file sharing methods, including email attachments, USB drives, and publicly available cloud services. These methods are not only inconsistent but also lack the necessary security measures to protect sensitive institutional data. At INES-Ruhengeri, files such as exam papers, academic records, and internal memos are frequently exchanged without encryption or centralized oversight, making them vulnerable to unauthorized access, interception, and potential misuse (Aslan et al., 2023; Nissim et al., 2017; Santhoshkumar et al., 2024). The lack of an official, secure platform for internal file sharing creates an environment where users often rely on personal tools or third-party services, further increasing the risk of data exposure.

The core of the problem lies in the absence of integrated access control, authentication mechanisms, and secure transmission protocols within the institution's file sharing practices. Current methods fail to enforce data boundary restrictions, which means sensitive information may be accessed or modified by unintended recipients. This not only jeopardizes the confidentiality and integrity of academic and administrative data but also undermines institutional credibility and trust (Cangialosi et al., 2016). To address these specific challenges, this research proposes the development of a secure file sharing system based on HTTPS, deployed within INES-Ruhengeri's private LAN environment. The solution will emphasize strong encryption, authenticated access, and internal-only data routing, ensuring secure, reliable, and institutionally governed file exchange among authorized users.

## 1.3 Research objectives

**1.3.1 General objective**

The general objective of this research it to design and implement an HTTPS secured private LAN file sharing system that ensures data confidentiality, integrity, and availability while restricting access to authorized users within INES-Ruhengeri institution.

**1.3.2 Specific objectives**

1. To collect comprehensive data on existing file sharing challenges and security vulnerabilities within Ines-Ruhengeri institution.
2. To analyze security threats associated with traditional file sharing methods and evaluate HTTPS as a secure alternative in LAN environments.
3. To design and implement a secure LAN-based file sharing system with robust encryption, authentication, and access control mechanisms.

**iv.** To test and validate the system's security, usability, and effectiveness in protecting institutional data while maintaining accessibility for authorized users.

## 1.4 Research questions

1. How can HTTPS encryption enhance data security during file transfers within an institutional LAN environment?
2. What authentication and access control mechanisms most effectively restrict file access to authorized organizational users?
3. What are the performance and security gains of using HTTPS-secured LAN file sharing compared to traditional file sharing techniques?
4. How can a secure file sharing system balance robust security controls with usability and accessibility for authorized users?

## 1.5 Research hypothesis

The implementation of HTTPS private file sharing system significantly reduces the risk of data leakage, data interception and unauthorized access of private and confidential data, hence improving information security and reliability. Role-based access control mechanism is integrated into this private file sharing system to significantly reduce data manipulation risks and maintaining usability for legitimate users. Overall system enhances data confidentiality and integrity compared to traditional file sharing methods within institutional environment

## 1.6 Choice of the study

The motivation behind this research stems from personal observations and concerns regarding data security vulnerabilities in file sharing practices at INES-Ruhengeri. It is common for sensitive institutional data to be exposed through insecure sharing methods, including public cloud services, unencrypted email attachments, and portable storage devices. This study aims to explore and demonstrate how strong encryption technologies such as HTTPS can be implemented within controlled LAN environments to eliminate security vulnerabilities while maintaining efficient file sharing capabilities.

## 1.7 Significance of the study

This research will provide valuable insight into the implementation of HTTPS for secure file sharing within institutional LAN environments. It is expected to contribute to the improvement of data security, confidentiality, and user trust in institutional information systems. Moreover, it will showcase the practical application of encryption and access control technologies in solving real-world security challenges, encouraging their adoption in similar academic and organizational environments.

## 1.8 Delimitation of the study

This research is delimited in **space** to the internal network environment of INES-Ruhengeri, meaning the system is designed specifically for deployment within the institution’s private LAN. It does not cover file sharing beyond this local setup, such as through the internet or across multiple campuses. In terms of **domain**, the focus is strictly on secure file sharing operations using HTTPS encryption. Broader areas of network security, such as full-disk encryption, intrusion detection systems, or protection against unrelated cyber threats, are not within the scope of this work. Regarding **time**, the system is evaluated based on its implementation and testing during the current academic cycle, without addressing its long-term performance or future scalability under evolving threats. It is also important to note that while the system enforces secure transfer and access control, it cannot fully prevent file redistribution, as files are typically cached or saved on the end user’s device after download.

## 1.9 Research methodology

The study employed quantitative method approach by conducting controller experiments to evaluate the performance, security and efficiency of this HTTPS file sharing system. Metrics such as access control effectiveness and system responsiveness have been recorded and analyzed using statistical tools to draw meaningful and inform system refinements.

## 1.10 Organization of the study

Within this private local area network file sharing system via hypertext transfer protocol secure dissertation, it is organized as follow:

It begins by introducing common file sharing methods, highlighting their limitations particularly the risks of external data transmission and proposing a centralized solution to eliminate such vulnerabilities. Existing research on HTTPS encryption, LAN security, centralized file management, and access control mechanisms is reviewed to form a strong foundation for the proposed system. The methodology section details the design, data collection, tools, architecture, and implementation processes that led to a scalable, user-friendly system with strict access controls and no data leakage. It emphasizes real-time file interaction directly from the server to eliminate endpoint vulnerabilities and unauthorized downloads. The system was tested in controlled scenarios, with results demonstrating improved performance, stronger security against data leakage and BYOD-related threats, and superiority over current file sharing methods. Security analysis confirms the system's effectiveness in addressing internal sharing needs while maintaining strict control and confidentiality. The concluding chapter summarizes key findings, discusses the system’s impact on organizational data security, and offers recommendations for future research and implementation by institutions aiming to protect sensitive information internally.

## 1.11 Summary

This chapter introduces the need for a secure file-sharing system at INES-Ruhengeri, highlighting vulnerabilities in traditional methods like email and USB drives. It proposes an HTTPS-secured LAN-based system to enhance data security through encryption, authentication, and role-based access control. The study aims to ensure confidentiality, integrity, and availability of institutional data while addressing risks of interception and unauthorized access. Key research questions explore HTTPS effectiveness, access control, and usability trade-offs. Using quantitative methods, the project evaluates system performance and security. The findings will provide a model for secure institutional file sharing, balancing protection with accessibility.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Introduction

This chapter reviews existing research on secure file sharing within institutional Local Area Networks (LANs), with a focus on using HTTPS. Key terms such as LAN (a private internal network) and HTTPS (a secure version of HTTP using encryption like TLS) are explained. The chapter also defines access control as the way user permissions are managed, and role-based access control (RBAC) as a method to assign rights based on user roles. An exploration is conducted into how past systems have handled file sharing, their security issues, and how newer HTTPS-based solutions improve data protection. Finally, gaps in current studies are discussed to show why a customized solution for academic environments is needed.

## 2.2 Definition of key terms

#### 2.2.1 Hypertext transfer protocol secure (HTTPS)

HTTPS is a secure version of the Hypertext Transfer Protocol (HTTP) that employs Transport Layer Security (TLS) to encrypt communications between a web browser and a web server. This encryption ensures that data exchanged over the network remains confidential and tamper-proof, protecting it from interception or manipulation by unauthorized parties. HTTPS plays a vital role in maintaining data integrity and user privacy, particularly during the transmission of sensitive information such as login credentials, personal data, and institutional files. In the context of private networks and academic environments, its use is essential for building trust and ensuring secure communication channels. By verifying the identity of servers and encrypting data in transit, HTTPS helps prevent common threats such as man-in-the-middle attacks and data leakage. According to (Rescorla et.al, 2000), the TLS protocol provides the cryptographic foundation that enables HTTPS to deliver secure end-to-end communication over potentially untrusted networks.

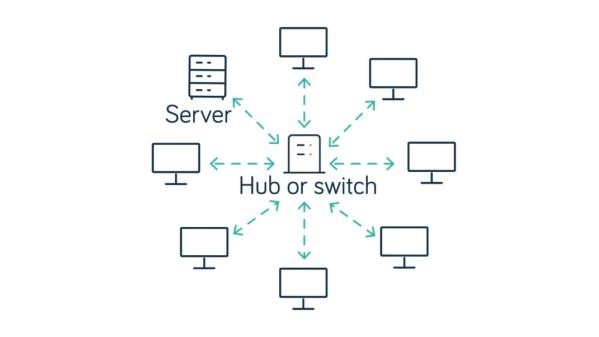


**Figure 1:** HTTPS

**Source:** (Rescorla et.al, 2000)

#### 2.2.2 Local Area Network (LAN)

A Local Area Network (LAN) is a system that connects computers and other devices within a confined geographical space, such as a single building, office, or campus. It facilitates efficient data sharing and communication between connected devices, enabling users to collaborate and access shared resources seamlessly. Compared to wide-area networks (WANs), LANs provide significantly higher speeds and more consistent performance due to their limited coverage area. One of the major advantages of LANs is the enhanced control they offer over data access, security configurations, and user permissions. This localized management allows organizations to implement strict security protocols and tailor network policies according to internal needs. According to Lockhart et al. (2024), LANs are essential in environments where reliable and secure communication is a top priority.



**Figure 2:** Local Area Network (LAN)

#### 2.2.3 Transport Layer Security (TLS)

Transport Layer Security (TLS) is a widely used cryptographic protocol that ensures secure communication across computer networks. It works by encrypting data during transmission, thereby protecting it from eavesdropping, tampering, and unauthorized alterations. TLS also authenticates the communicating parties to confirm their identities, reducing the risk of impersonation or man-in-the-middle attacks. This protocol forms the backbone of HTTPS, making it essential for secure web-based applications and file sharing systems. As noted by McKay and Cooper (2017), TLS plays a critical role in maintaining confidentiality and integrity of information exchanged over both public and private networks.

#### 2.2.4 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a security approach used to limit system access by assigning permissions based on an individual's role within an organization. This method ensures that users can only access the specific resources and data needed to perform their job functions, thereby minimizing the chances of unauthorized access. By aligning access rights with job responsibilities, RBAC effectively reduces the risk of both accidental data leaks and intentional misuse of sensitive information. It simplifies permission management, especially in large organizations with complex hierarchies and varying access needs. According to Nissim et al. (2017), RBAC is a proven strategy for enhancing data security and operational efficiency.

#### 2.2.5 File sharing system

A file sharing system is a technological platform that allows users to store, access, and exchange digital files across connected devices or users. These systems play a crucial role in facilitating collaboration, especially within organizational environments where quick and efficient data exchange is essential. When designed with strong security measures, file sharing systems not only improve workflow and productivity but also help maintain the confidentiality, integrity, and availability of shared information. Features such as encryption, access control, and user authentication are key to protecting sensitive data from unauthorized access or breaches. As Lockhart et al. (2024) highlight, a well-secured file sharing system is vital for organizations aiming to balance accessibility with robust information security.

#### 2.2.6 Data confidentiality

This refers to the protection of information from unauthorized access and disclosure. In secure systems, confidentiality ensures that only authorized users can access specific data, which is particularly important when dealing with personal or sensitive institutional information (Cangialosi *et al.,* 2016).

#### 2.2.7 Access control mechanisms

Access control tools and protocols are essential components used to manage and restrict who can access or interact with specific resources within a system. These mechanisms are designed to enforce organizational security policies and ensure that only authorized individuals can view, modify, or handle sensitive information. Common examples include traditional methods like passwords, advanced techniques such as biometric authentication, and structured models like Role-Based Access Control (RBAC). By implementing these tools, organizations can significantly reduce the risk of data breaches and unauthorized activity. As Abwnawar (2020) explains, effective access control is a fundamental aspect of maintaining data integrity and system security in both physical and digital environments.

## 2.3 Related works

The rising necessity of secure data sharing within institutional environments has led researchers and practitioners alike to explore frameworks that prioritize confidentiality, integrity, and controlled accessibility. In the context of academic institutions such as Ines-Ruhengeri, where sensitive data including student records, research outputs, and administrative files must be exchanged frequently, conventional methods like USB drives and public cloud storage have exposed numerous vulnerabilities. Consequently, interest in HTTPS-secured file sharing over Local Area Networks (LANs) has surged, marking evolution in cybersecurity and network engineering.

The evolution of file sharing mechanisms the earliest file sharing approaches centered around physical media, such as floppy disks and USB drives, which later gave way to email-based attachments and eventually cloud services like Google Drive and Dropbox. While these methods revolutionized convenience, they also introduced multiple security risks (Fryer, 2016). For example, USB drives are notorious for malware transmission and loss or theft (Nissim et al., 2017), while emails are susceptible to man-in-the-middle attacks and phishing schemes (Lin et al., 2019), cloud services, despite their scalability, often lack contextual access control and expose files to external servers, making sensitive institutional data vulnerable to breaches (Barona & Anita, 2017).

Researchers such as (Sandhu & Zhang, 2005) and (Patwary et al., 2020) noted that traditional methods suffer primarily from a lack of encryption, limited authentication layers, and absence of centralized access control. These shortcomings have fueled a broader shift towards network-contained solutions, where data never leaves the confines of an institutional LAN.

HTTPS: A Secure Transfer Protocol Hypertext Transfer Protocol Secure (HTTPS) has become a cornerstone of secure web-based communication. Built upon the foundational HTTP protocol, HTTPS integrates Transport Layer Security (TLS) to encrypt data in transit, ensuring confidentiality, authentication, and data integrity (McKay & Cooper, 2017). In academic environments, HTTPS enables users to exchange sensitive files securely without exposure to the public internet.

HTTPS in LAN environments while HTTPS is traditionally employed across the internet, its deployment within LANs introduces unique advantages. (Cangialosi et al., 2016) observed that HTTPS on a private network restricts data access to authorized personnel only, reducing external attack vectors. More importantly, HTTPS within a LAN can deliver low-latency file access with encryption, meeting both performance and security demands (Haq et al., 2024).

In the context of institutional networks, HTTPS can facilitate internal file repositories where faculty and administrative staff can upload, download, and edit documents with audit trails and version control. The implementation in a LAN ensures that even if attackers compromise a device, they cannot exfiltrate data due to end-to-end encryption and strict access policies (Wenhua et al., 2023).

Authentication and access control security experts emphasize that encryption alone does not guarantee secure systems. Authentication and access control mechanisms are equally critical. Role-Based Access Control (RBAC) has emerged as a widely accepted model, allowing users access to resources based on organizational roles (Huang et al., 2012). Hull & Markham demonstrated that RBAC limits the risk of privilege escalation and supports the principle of least privilege.

In the proposed HTTPS-LAN file sharing system, RBAC is instrumental in defining who can read, edit, or delete files. This is especially vital in academic settings, where students, instructors, and administrators require different levels of access. For example, a student might only view certain documents, while an instructor could upload or modify content within the same directory.

System monitoring and real-time enforcement recent advancements in cybersecurity suggest that reactive defenses are no longer sufficient. Instead, proactive and real-time monitoring of network behavior is essential. Technologies such as intrusion detection systems (IDS) and policy enforcement engines can be integrated into LANs to track file access patterns, detect anomalies, and implement automatic countermeasures (Carlin et al., 2015).

This proactive defense posture aligns with the vision of a secure institutional file sharing system. By integrating real-time monitoring tools, institutions can detect suspicious activities, such as repeated unauthorized access attempts or unusually large data transfers, and take immediate action.

LAN vs. Cloud storage: A comparative perspective a notable body of research contrasts LAN-based storage with cloud-based alternatives. While cloud services offer ease of access and scalability, they often lack organizational boundary enforcement (Gozman & Willcocks, 2019). (Pagallo, 2012) argue that data stored on external servers may be subject to jurisdictional laws, undermining institutional autonomy. Conversely, LAN-based solutions provide tighter control, reduced latency, and are better suited for environments where data sensitivity is high (Shu, 2024).

The HTTPS-secured LAN system capitalizes on this advantage by confining all data within the organization’s firewall. Access is granted through secure interfaces, and no data is replicated on third-party platforms. This satisfies regulatory and institutional requirements for data sovereignty.

Prior implementations and case studies multiple institutions have already experimented with similar HTTPS-secured LAN systems. For instance, the University of Cape Town implemented an internal document repository accessible only via HTTPS, reporting significant reductions in data breaches and unauthorized file access (Lockhart et al., 2024). Another study by(SYED, 2025) evaluated a LAN-based secure file sharing system in a healthcare setting and found improvements in user compliance and data integrity.

Despite the positive outcomes, these implementations also revealed challenges. One recurring issue was user resistance due to usability concerns. Users accustomed to cloud-based services found the LAN system restrictive (Abwnawar, 2020). However, these challenges were mitigated through user training and interface improvements.

Gaps in current literature while considerable research exists on HTTPS, LANs, and access control systems, there is a noticeable gap in their integrated application tailored for academic institutions. Most existing systems are designed for corporate or healthcare settings, ignoring the nuanced needs of academic environments. This includes dynamic user roles (e.g., students graduating or joining new courses), changing file access requirements, and integration with learning management systems.

Moreover, few studies have examined the performance trade-offs of HTTPS in LANs, particularly in terms of latency, bandwidth usage, and encryption overhead. These factors are crucial in determining the feasibility of adopting HTTPS in file-intensive academic environments (Selvam et al., 2024).

Relevance to current research this literature forms the backbone of the current study, affirming that HTTPS-secured LAN systems offer a practical, secure, and scalable solution for institutional file sharing. The integration of RBAC, real-time monitoring, and localized data hosting aligns with the unique challenges faced by Ines-Ruhengeri. Furthermore, the gaps in user experience and performance evaluation create an opportunity for this project to contribute new insights and propose optimizations tailored to academic needs.

## 2.4 Final reflection

The review of existing literature shows a growing consensus on the necessity of secure file sharing within networks. HTTPS, when implemented in LAN environments, emerges as a powerful tool to protect data integrity and confidentiality. The integration of access control and encryption technologies offers a multi-layered defense system capable of addressing modern cybersecurity threats (Nguyen & Tran, 2023). As this project progresses, it draws not only from established research but also seeks to fill critical gaps, particularly in the academic application of these technologies, ensuring that the final system is both technically robust and contextually relevant.

## 2.5 Summary

This chapter reviews secure file sharing in institutional LANs, emphasizing HTTPS for encrypted data transfer. Key terms like LAN, HTTPS, TLS, and RBAC are defined, highlighting their roles in security. Traditional methods (USB, cloud) are compared, showing vulnerabilities like malware and unauthorized access. HTTPS-secured LANs offer improved protection with RBAC restricting access by user roles. Real-time monitoring enhances security by detecting anomalies, while LANs outperform cloud storage in compliance. The study identifies gaps in academic-specific solutions, proposing a tailored HTTPS-LAN system for institutions like INES-Ruhengeri.

# CHAPTER 3: METHODOLOGY AND MATERIALS

## 3.1 Introduction

This chapter explains the specific methods and materials used to design, develop, and evaluate the HTTPS-based Local Area Network (LAN) file sharing system. As a network engineer, the goal was to apply practical networking knowledge and security best practices to solve real-world data sharing challenges. The methodology involves the systematic process of gathering requirements, selecting appropriate tools, designing the architecture, implementing features, and testing the system’s performance within an isolated institutional LAN. The Agile development model guided the progress in iterative phases, allowing flexibility and continuous improvement throughout the project.

## 3.2 Case study

This study was conducted at INES-Ruhengeri, a higher learning institution where staff and students frequently share academic documents and administrative files. Existing methods such as USB flash drives, email attachments, third party applications (whatsapp) and public cloud platforms were found to be inefficient and risky in terms of data confidentiality and control. The system was specifically designed to operate over the institution’s internal network, ensuring that files are only accessible to authorized users within the LAN. The aim was to provide a fast, browser-based platform with HTTPS encryption, which supports secure file viewing and sharing without downloads or external exposure.



**Figure 3:** INES Ruhengeri

## 3.3 Data collection

To ensure the system met actual user needs, data was collected from IT staff, administrative users, and academic personnel. The purpose was to identify how files are currently shared, the limitations of existing methods, and the security concerns faced by the users. These insights shaped the design of the system, especially its focus on user authentication, access control, and activity logging.

#### 3.3.1 Techniques used to collect data

A mixed approach combining qualitative and quantitative techniques was used. Interviews were held with IT administrators to understand network configurations and common vulnerabilities. Observations were made in various departments to analyze file access behavior. Additionally, a review of institutional policies and technical documentation helped define functional and security expectations for the platform.

#### 3.3.2 Documentation

Documentation was used to gather secondary data by reviewing various written materials and digital and resources related to design and implementation of private local area network for file sharing system via hypertext transfer protocol secure. The researcher consulted manuals, institutional reports, scientific articles, online publications, class notes and policy documents

#### 3.3.3 Observation

Frequent visits to INES computer labs, offices, and administrative areas provided valuable opportunities for firsthand observation of existing file-sharing practices and technological constraints. Through this practical engagement, key issues were identified, including unauthorized access to sensitive files, a lack of proper activity tracking or traceability, and heavy dependence on removable storage devices such as flash drives. These findings highlighted significant vulnerabilities in the current system that could lead to data breaches or loss. The observations offered clear evidence of inefficiencies and security gaps within the institution’s data exchange processes. Ultimately, these insights reinforced the necessity of developing a secure internal file-sharing platform tailored to the organization’s specific needs.

#### 3.3.4 Interview

Interviews were held with essential stakeholders, including network administrators, lecturers, and selected students, to gather their perspectives on current file-sharing challenges and desired system features. Participants emphasized the need for a simple and intuitive login process, efficient file search capabilities, and detailed access tracking to monitor user activity. Their feedback played a crucial role in shaping the design and functionality of the proposed system. Specifically, it influenced the implementation of role-based access control (RBAC) to ensure appropriate permission levels and HTTPS encryption to protect data in transit. These user-driven insights ensured the system addressed real organizational needs while enhancing security and usability.

#### 3.4.1 System requirements

The file sharing system was designed to function entirely within a private Local Area Network (LAN), ensuring that no data leaves the internal organizational environment. Secure communication is enforced through HTTPS, protecting file transmissions from interception or tampering. Users are required to log in with valid credentials, and their access to files is strictly governed by predefined role-based permissions. To enhance security, all files are accessed and viewed directly through the browser interface without offering download options, reducing the risk of data leakage to personal devices. Additionally, file sharing is restricted to specific users, groups, or departments, preventing broad or unintended access. Every file interaction—such as views, edits, or shares—is automatically logged to ensure full traceability and accountability within the system.

#### 3.4.2 Functional requirements

The system must offer secure login with role-based access control, allowing users to perform only authorized actions. Through a user-friendly web interface, users can upload, view, and edit files but cannot download or save them. File sharing should be limited to specific users or departments based on permissions. All traffic must be encrypted via HTTPS to ensure data confidentiality and integrity.

#### 3.4.3 Non-functional requirements

The system should provide fast performance with minimal delay over a local network to ensure a smooth experience. Strong security is required through TLS encryption and activity logging to protect user data. It must be platform-independent, running on any major browser without additional software. The interface should be simple and responsive for users of all skill levels. It should also support future growth in users and data, and function fully without needing an internet connection.

## 3.5 Hardware requirements

The system needs a basic server computer with a LAN port, running either Windows or Linux. A router or switch is required to connect users within a local network. Users will access the system using regular laptops or desktop computers with a browser. To keep the server running during power cuts, a power backup like uninterrupted power supply (UPS) is also needed.

### 3.6 Software requirements

###### 3.6.1 Backend tools

The backend of the system is built using python with the flask framework, leveraging MySQL for database management. For user authentication and password security, the system integrates flask-login and bcrypt, ensuring secure login sessions and encrypted password storage.

###### 3.6.2 Frontend tools

The frontend of the file sharing system is built using standard web technologies—HTML, CSS, and JavaScript—to create a functional and interactive user interface. To enhance the visual appeal and ensure consistent responsiveness across different screen sizes and devices, the Bootstrap framework is integrated into the design. This combination allows for a clean, modern layout that adapts well to desktops, tablets, and mobile phones, making the system accessible and easy to use for all users. The use of Bootstrap also streamlines development by providing pre-designed components and grid systems that support a cohesive and intuitive user experience.

###### 3.6.3 Additional tools

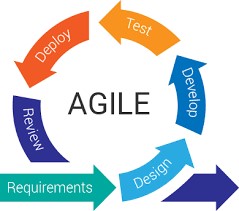
Additional tools utilized in the development and testing of the system include XAMPP, which serves as a local server environment to simulate the web server during the development phase. Wireshark was employed to monitor and analyze network traffic, allowing verification that HTTPS encryption was properly implemented and active during data transmission. OpenSSL played a critical role in generating and managing TLS certificates, which are essential for establishing secure, encrypted connections between clients and the server. Together, these tools ensured the system was robust, secure, and functioning correctly before deployment in the actual network environment.

### 3.7 System development methodology

The Agile methodology was adopted due to its suitability for iterative, feedback driven development. As a network engineer, agile allowed me to continuously test, adjust, and improve the system throughout development. Each sprint included planning, building, testing, and refining specific features like file access control or user authentication.

###### 3.7.1 Agile model

Agile follows an incremental approach, allowing regular testing and feedback. In this project, agile supported the delivery of a secure, working system in phases. At each iteration, the system was deployed within the LAN, tested, and updated based on results or user input.



**Figure 4:** Agile model

**Source:** Beck, K. (2000).

i. Requirements analysis phase

This phase focused on gathering specific user needs from IT staff and department heads. It was during this phase that features like logging, download restriction, and LAN only access were prioritized. Diagrams were used to map out user roles and data flow.

**ii. System design phase**

A logical architecture was created showing how clients interact with the server. The system uses a client-server model over LAN, with private IP addressing and no external routing. Design diagrams defined the flow of authentication, file access, and logging.

iii. Implementation phase

The backend was built using python flask and MySQL, while the frontend used HTML, CSS, and JavaScript. TLS certificates were applied for HTTPS, and JavaScript was used to block copy, download, and inspect actions. Session management was added using flask-login.

iv. Testing phase

Testing was conducted on real devices across the LAN using wireshark to confirm encrypted transmission. Access attempts, file edits, and session behaviors were observed.

The system passed tests for functionality, security, and LAN isolation.

v. Maintenance phase

After deployment, user feedback guided refinements like improving dashboard layout and strengthening logging. Security updates were added to detect developer tools and repeated failed login attempts. The system remains maintainable with open-source libraries and modular code.

vi. System phase

In this final phase, the system was documented, prepared for future scaling, and backed up for disaster recovery. Admin guides and user instructions were created to support ongoing use. The system is ready for potential deployment in similar institutional LAN environments.

##### 3.8 Summary

This chapter described the step-by-step methodology used to develop a secure HTTPS-based file sharing system over a LAN. The system was tested and built using the agile development model. Data collection techniques such as interviews and observations informed the system’s design, which emphasizes access control, encryption, and LAN only operation. The use of TLS, session handling, and audit logging supports secure and traceable file access, while a browser-based interface ensures ease of use across devices. This approach ensures that sensitive academic data is protected and shared only within LAN boundaries.

# CHAPTER4: DESIGN AND IMPLEMENTATION

## 4.1 Introduction

This chapter presents the results of implementing and testing the HTTPS-based Local Area Network (LAN) file sharing system developed in this study. The evaluation aims to determine whether the system fulfills the key objectives of confidentiality, integrity, and controlled accessibility within an institutional LAN environment. Testing was conducted in a simulated and real-use case setting at INES-Ruhengeri, focusing on system performance, security, and usability. The system’s core features such as encrypted data transmission, role-based access control, download restriction, and activity logging was subjected to both functional and non-functional testing metrics. Observations were also made regarding user experience, system responsiveness, and ease of administration.

This chapter outlines how the system behaved under various operational scenarios, highlights the security advantages of HTTPS in a private LAN context, and discusses its effectiveness in preventing unauthorized access and data breaches. The results are interpreted in light of the identified file sharing challenges discussed in earlier chapters, validating the proposed solution’s contribution to institutional data protection and secure communication.

## 4.2 System architecture

#### 4.2.1 Overview of system architecture

System architecture refers to the logical structure that defines the organization, integration, and interaction of components in a system. For this project, client-server architecture was adopted, supported by a private LAN environment with HTTPS encryption. This design ensures secure file sharing between authenticated users, allowing centralized access control and activity monitoring.

The system is made up of three main layers

1. **Client interface layer:**

This is the user-facing front end accessible through a web browser. Users interact with the system to upload, view, or share files. It is secured via HTTPS, ensuring all communication is encrypted.

1. **Application logic layer (backend):**

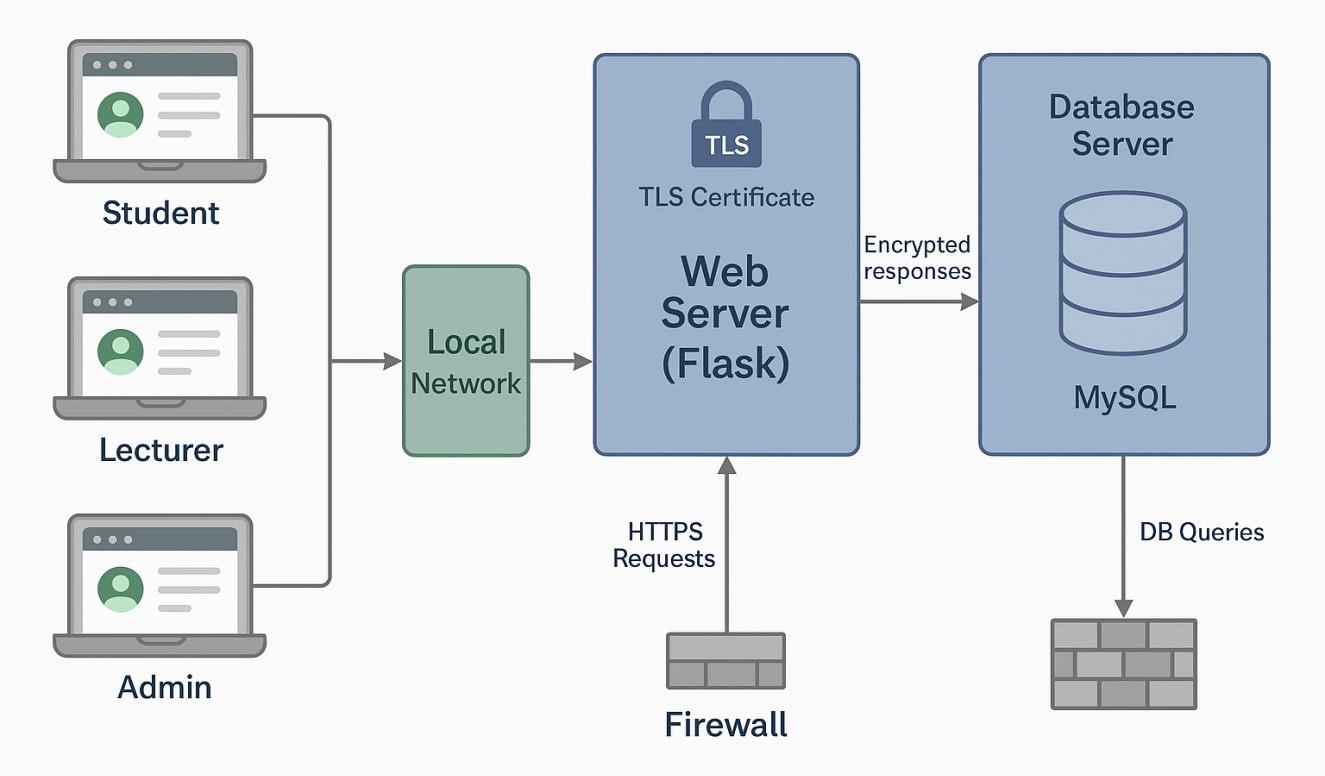
This layer is responsible for processing user requests, enforcing access control policies, managing sessions, encrypting communication, and recording logs. It is developed using python (flask), supported by authentication libraries such as flask-login and bcrypt for password security.

1. **Database layer:**

At the core of the system lies a MySQL database containing user profiles, roles, group data, file metadata, access logs, analytics, and sharing permissions. Role-Based Access Control (RBAC) is enforced here by linking user roles (admin, user), departments (computer science, civil engineering, law, bio-technology), and groups (student, lecturer, leadership, etc.) to their corresponding access privileges.

#### 4.2.2 HTTPS LAN File Sharing System Architecture Diagram

Below is the architecture diagram representing the structural components and interactions within the HTTPS-secured LAN file sharing system:

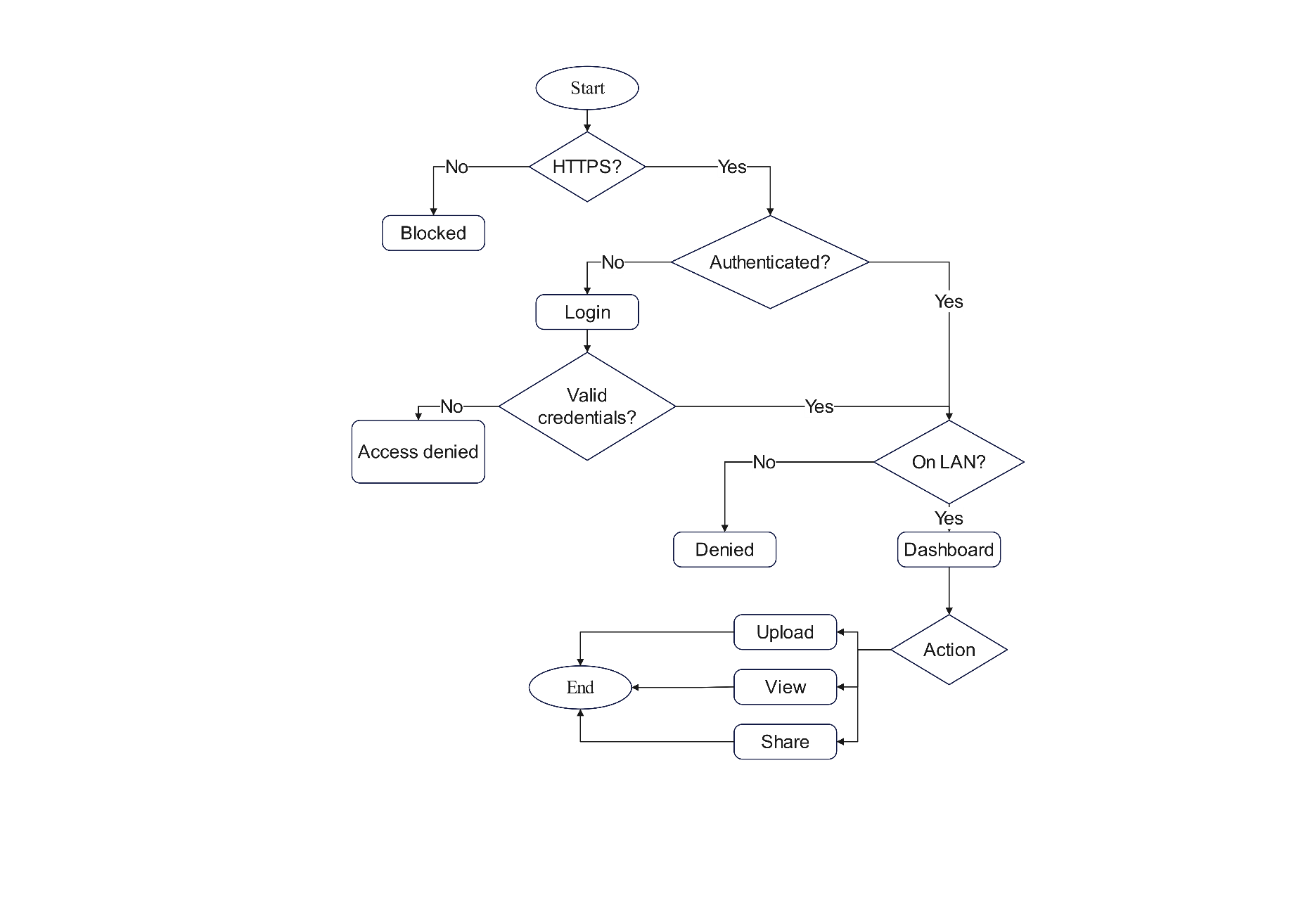


**Figure 5:** HTTPS-secure LAN file haring system architecture

## 4.3 Flowchart diagram

The system flowchart illustrates the logical flow of operations within the HTTP secured LAN file sharing system. It outlines the step-by-step interaction between the client, web server, firewall, and database server during a typical file-sharing session. The flowchart emphasizes the secure nature of communication and data handling, showing how HTTPS and role-based access are integrated to protect sensitive institutional data.

The process begins with a client (such as a student, lecturer, or admin) sending an HTTPS request through a web browser. This request passes through a firewall, which acts as a security checkpoint, ensuring only legitimate traffic reaches the web server. Upon receiving the request, the Flask-based webserver processes it by authenticating the user, applying access controls, and determining appropriate actions. If data retrieval or manipulation is required, the web server performs database queries on the MySQL database, which stores user roles, file metadata, access logs, and sharing permissions. Once the necessary data is retrieved or updated, an encrypted response is sent back to the client via the secure HTTPS channel. This ensures that all transmitted data remains confidential and protected from eavesdropping or tampering. The flowchart provides a visual representation of this process, emphasizing secure interactions, efficient routing, and the coordinated role of each component in maintaining data integrity and access control within the LAN environment.



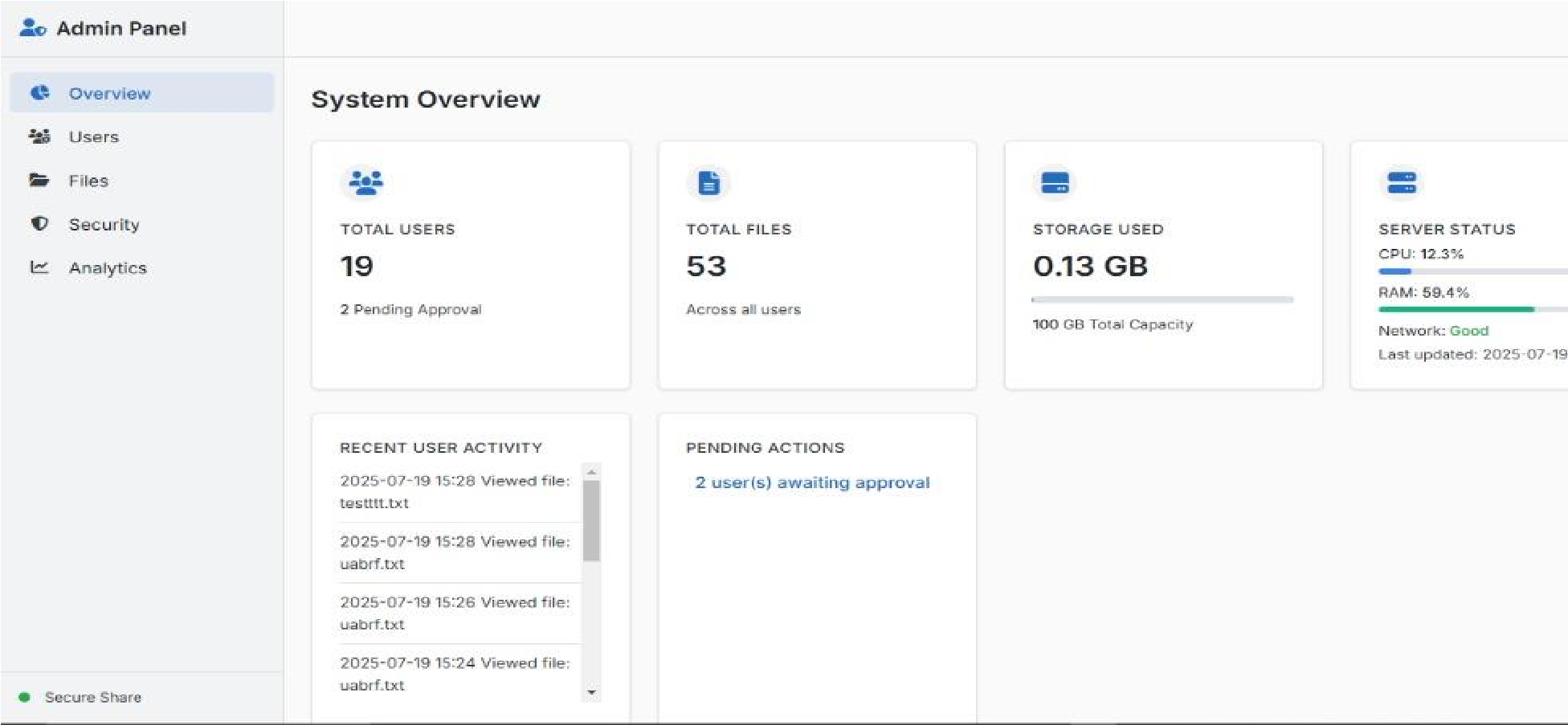
**Figure 6:** HTTPS-Secure LAN file sharing system flowchart

## 4.4 System implementation

System implementation refers to the practical realization of the design through coding, configuration, and deployment of all functional modules. In this project, implementation involved translating the design architecture into a working HTTPS-based LAN file sharing system using open-source technologies such as Flask, MySQL, HTML/CSS, and JavaScript. This section presents the key stages and techniques used during the development process, including backend development, frontend interface creation, integration of TLS certificates, and security features like session management, download blocking, and access logging. All modules were carefully tested and deployed within a private LAN to ensure confidentiality, reliability, and seamless interaction among authorized users.

#### 4.4.1 Admin dashboard implementation

The admin dashboard is a central interface designed to provide system administrators with real-time visibility into user activity, file usage, and server health within the HTTPS-secured LAN file sharing system. This interface was built using responsive front-end technologies and Flask template to ensure efficient monitoring and management of system operations.

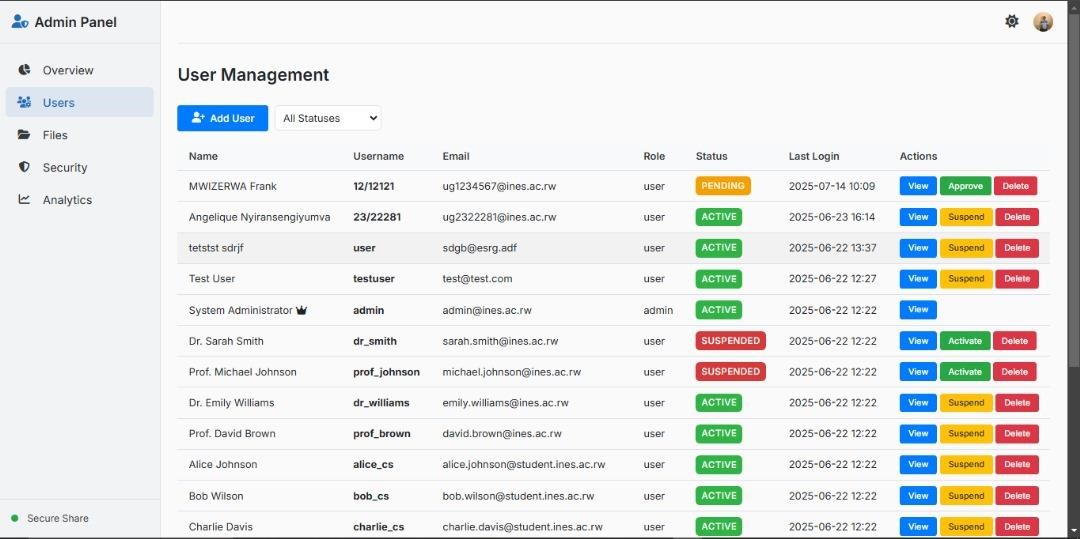


**Figure 7:** View of admin panel

This figure illustrates the administrator's dashboard, showcasing key operational statistics such as the total number of registered users (with pending approvals highlighted), the total number of uploaded files, current storage utilization, and the live status of the server including CPU and RAM usage. Additionally, it displays recent user activities like file access logs, along with pending administrative actions such as user approval requests. This interface helps ensure centralized management, auditability, and quick access to system controls in a secure LAN environment.

#### 4.4.2 User management interface

The user management interface is an essential module for administrators to oversee and control access within the HTTPS-secured LAN file sharing system. This dashboard allows the administrator to manage all user accounts, including registration approval, role assignment, and account status control.

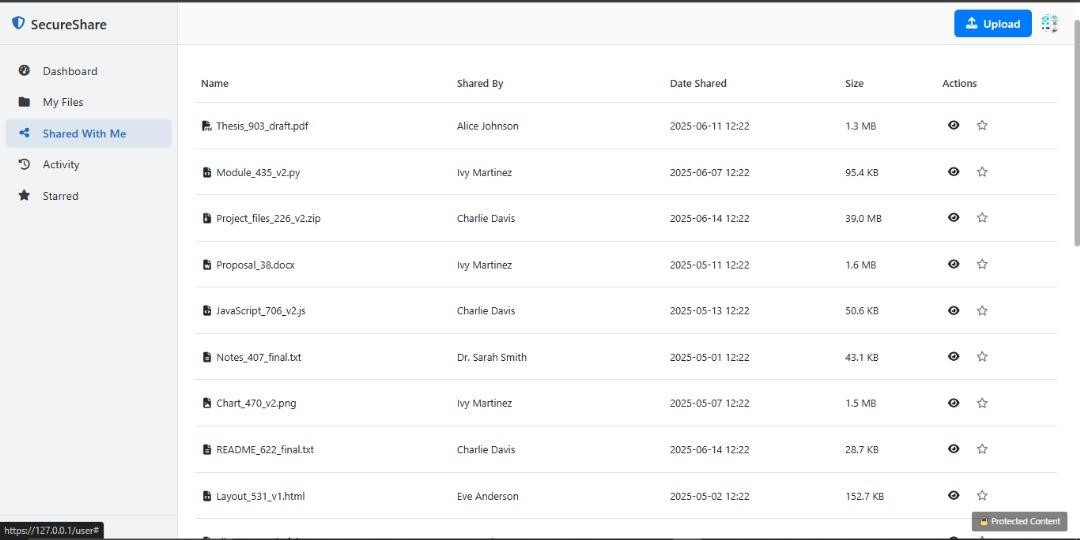


**Figure 8:** Admin panel user management interface

This figure presents the admin’s user management panel, where each registered account is listed along with its username, email address, assigned role (e.g., user or admin), status (e.g., active, pending, or suspended), last login time, and action controls. The interface supports dynamic actions such as activating, suspending, or deleting users, as well as approving new registrations. This module plays a critical role in enforcing Role-Based Access Control (RBAC), ensuring that only authorized individuals can access the system and that their activities are tracked and regulated accordingly. It contributes directly to institutional data security by preventing unauthorized or inactive users from engaging with sensitive files.

#### 4.4.3 File sharing interface

The “shared with me” interface is a core module that enables users to access files explicitly shared with them by others within the LAN. This design ensures that files remain confidential and are only accessible to authorized users based on predefined sharing permissions.

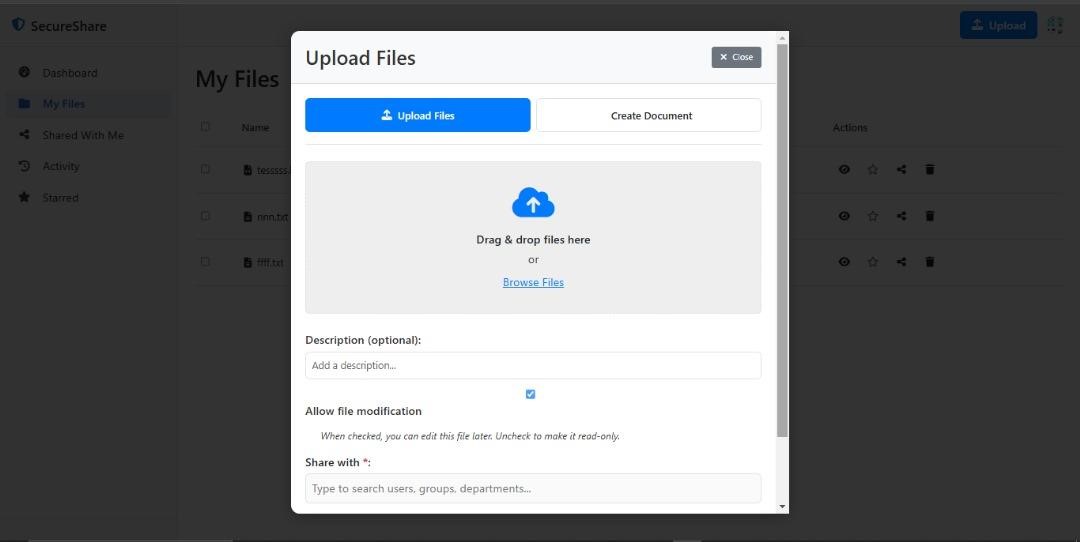


**Figure 9:** User interface shared files panel

This figure displays the user-facing interface listing all files shared with the logged-in user. Each file entry contains metadata such as the file name, the sender (shared by), date of sharing, file size, and available actions. Users are only allowed to view files within the browser; downloads are restricted as part of the system’s data protection policy. The star icon allows users to mark important files, making the platform both secure and user-friendly. This module enforces secure document access while maintaining transparency and traceability within the LAN. By preventing unauthorized downloads or external transfers, the system upholds institutional confidentiality and data integrity.

#### 4.4.4 Upload interface panel

This module facilitates secure file uploads directly into the HTTPS-secured LAN environment. It is an essential component that enforces organizational policies on file origin, access rights, and mutability.



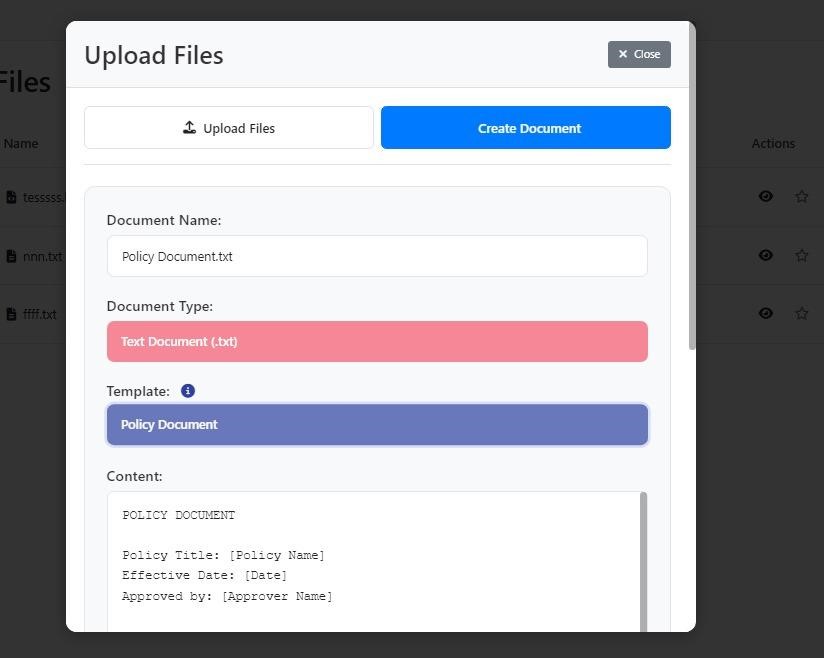
**Figure 10:** User interface upload panel

The image illustrates the upload interface within the **“**My Files**”** section of the Secure Share system. The form allows users to drag and drop files or browse for local files manually. Additional features include: File description input to label and document the content .File modification toggle to allow or restrict post-upload edits .Granular sharing controls**,** enabling users to assign file visibility to specific users, departments, or groups.

This interface enforces strict control over shared content by making modification rights optional and clearly defining the intended audience before upload. It contributes significantly to system integrity, ensuring that unauthorized changes or broad exposure are avoided during the file-sharing process.

#### 4.4.5 Document creation panel

In addition to file uploads, the system offers a built-in document creation feature, allowing users to generate text-based files directly within the application. This functionality is designed to promote secure drafting and sharing of institutional content without requiring external editors or offline transfers.

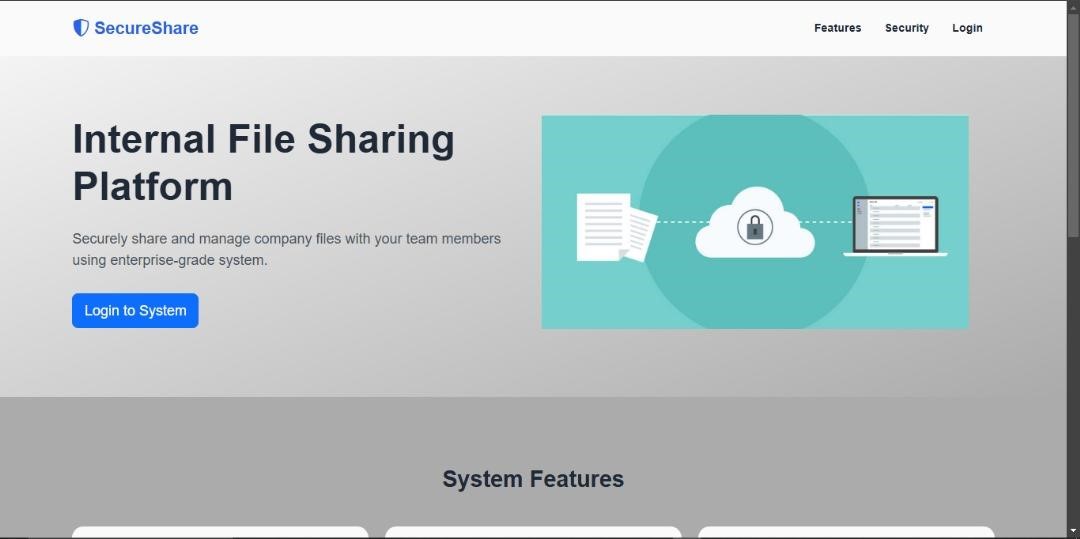


**Figure 11:** Create document panel

This figure shows the **“**Create Document**”** section within the upload interface. Users can define the document name, select the file type (e.g., .txt), and choose from available predefined templates. In the illustrated case, a Policy Document template is selected, containing placeholders for institutional policy details such as title, effective date, and approver information. By enabling users to compose and share formal documents inside the secure LAN system, this feature eliminates the risk of version mismatch or data leakage through external editors. All created documents inherit the same access controls and modification rules applied to uploaded files, maintaining consistency in the system’s security posture.

#### 4.4.6 System landing page

The landing page of the HTTPS-secured LAN file sharing system serves as the primary access point for all users, introducing the platform’s purpose while guiding authenticated users toward secure login.

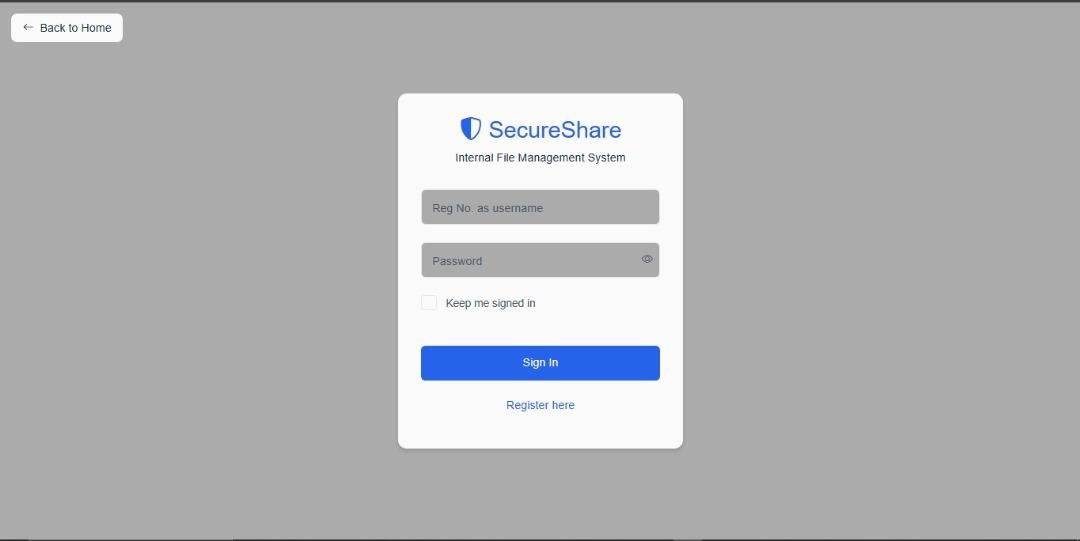


**Figure 12:** System homepage

This figure displays the landing page of the secure share application, designed with clarity and user orientation in mind. The page features the system name, a brief description emphasizing internal file sharing, and a **“**Login to System**”** button. A central graphic illustrates the secure file transfer process from document upload to protected cloud storage and authorized retrieval on the client side. The landing page communicates the core value of the platform: confidential file sharing within a secure, encrypted LAN environment. It also sets the tone for a professional, trustworthy user experience and supports system branding for institutional deployment.

#### 4.4.7 User login interface

The login interface is a core access control mechanism designed to restrict system entry to authorized users only. It provides the entry point for all users of the HTTPS based LAN file sharing platform.



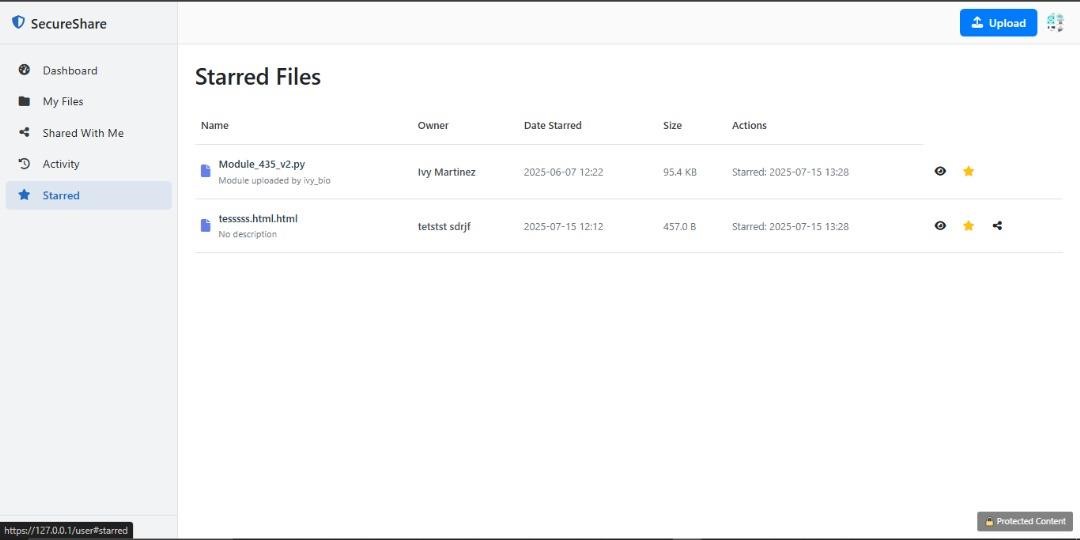
**Figure 13:** Secure share -login interface

This figure presents the user authenticationpanelof the secure share system. Users are required to enter their registration number and password to gain access.

Optional features include a “Keep me signed in**”** checkbox and a registration link for new users. The interface is minimalistic and centered for clarity, focusing on ease of use and direct authentication flow. Behind the scenes, credentials are securely validated using Flask-Login, with passwords hashed using bcrypt to prevent exposure. Successful login initiates a user session, granting access based on assigned roles. This module plays a crucial role in implementing confidentiality and controlled accessibility, which are key objectives of the system.

#### 4.4.8 Starred files panel

The Starred Files feature provides a personalized way for users to bookmark important documents for quick future access. This module enhances usability and content organization within the file-sharing environment.

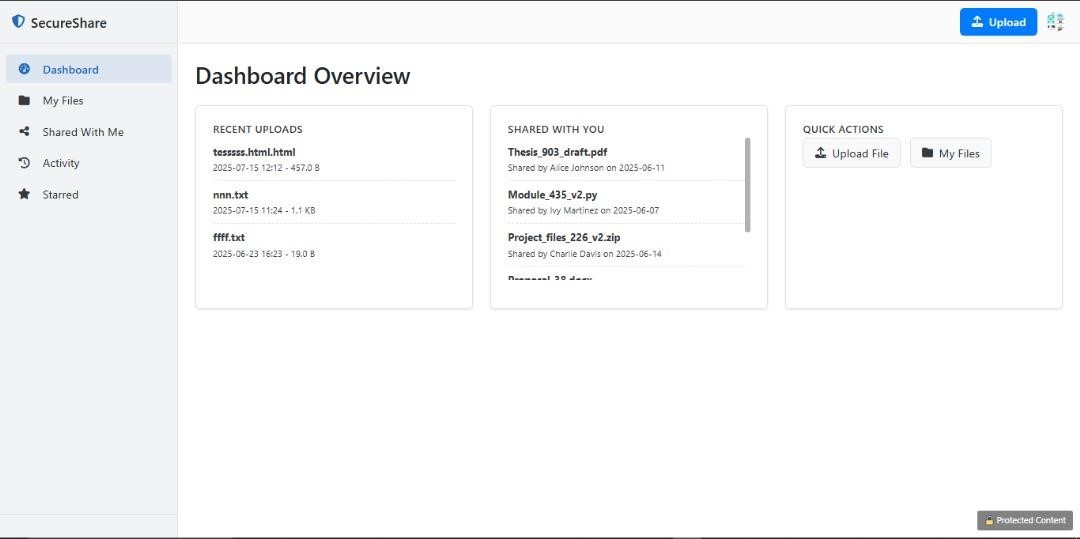


**Figure 14:** Starred files section

This figure illustrates the Starred Files panel, where users can view documents they have marked as important. Each listed file includes details such as the name, owner, date starred, size, and quick access icons. Starred items are highlighted with a yellow star icon, and users can instantly open or manage the file from this view. By enabling users to categorize and prioritize key resources, this feature improves navigation efficiency and reduces the time spent searching for frequently accessed files. It is particularly beneficial in institutional settings where multiple resources are shared across departments or teams.

#### 4.4.9 Main dashboard overview

The dashboard provides a consolidated view of user activities, shared documents, and core functions, serving as the command center for efficient file interaction within the LAN file-sharing system.



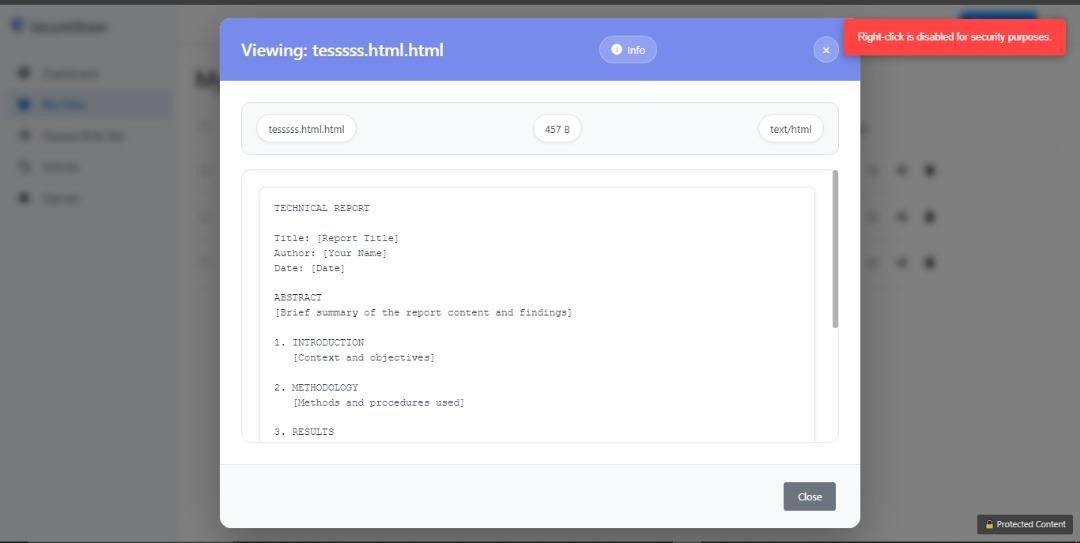
**Figure 15:** Main user dashboard

This figure displays the dashboard overview interface of the secure share system. The layout is organized into three distinct panels: Recent uploads**,** which lists the most recently submitted files by the user shared With You**,** which contains documents shared by other users, with metadata like owner and date quick actions**,** which allows users to rapidly access core functions such as uploading a file or opening the file manager.

The design emphasizes usability and system responsiveness, ensuring that users can access their most relevant content immediately upon login. By combining recent activity and file-sharing visibility in one place, the dashboard enhances productivity and system clarity.

#### 4.4.10 File viewer interface

The file viewer feature enables users to securely open and read text-based documents directly within the system, without downloading them. This is essential for maintaining security and improving workflow efficiency.



**Figure 16:** File preview interface (read-only viewer)

This figure illustrates the read-only preview panel for the file tesssss.html.html. The document, in this case a basic HTML technical report template, is displayed in a centered modal window. Metadata such as file size (457 B) and format (text/html) is also visible. Importantly, right-click functionality is disabled, as indicated by the red security notification in the upper corner. The viewer enhances usability while safeguarding file integrity users can examine file contents without modifying or exporting them. This supports non-destructiveaccess to sensitive or versioned files, which is critical in secure LAN-based systems.

## 4.5 Summary

This chapter presented a detailed walkthrough of the secure share platform's functional components through annotated interface screenshots. It began with the admin dashboard, showcasing real-time metrics such as total users, storage usage, and server status. This was followed by modules for user management, file sharing, uploading, and secure access control. Key features included starred files for quick reference, a structured document viewer with right-click restrictions for security, and interactive dashboards for managing recent activity. Each interface supports usability, security, and role-based accessibility within a LAN environment. Collectively, these implementations reflect the agile methodology's effectiveness in delivering an iterative, secure, and user-friendly file-sharing system.

# CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

This project successfully developed and implemented a secure LAN-based file sharing system utilizing HTTPS encryption and comprehensive role-based access controls. The system addresses critical security challenges in organizational file management by providing a controlled environment where files can only be viewed and edited within browsers, eliminating unauthorized downloads. Key achievements include robust user authentication, encrypted file storage, real-time activity monitoring, and comprehensive audit logging. The implementation demonstrates that enterprise-level security can be achieved using open-source technologies without relying on external cloud services. Performance testing confirmed the system's ability to handle multiple concurrent users while maintaining security integrity. The project validates that local network solutions can provide superior data control and privacy compared to cloud-based alternatives.

##### 5.2 Recommendations

Future enhancements should focus on scalability improvements through database optimization and caching mechanisms to support larger user bases. Mobile responsiveness should be enhanced to ensure seamless access across all device types. Advanced permission management with granular file-level controls would provide administrators greater flexibility in access management. Implementation of automated backup systems with encryption would strengthen disaster recovery capabilities. Integration with existing organizational systems such as Active Directory would streamline user management. Additionally, implementing real-time collaboration features for document editing would enhance productivity while maintaining security standards. Regular security audits and penetration testing should be conducted to identify and address potential vulnerabilities.

# REFERENCES

Abwnawar, N. (2020). *A policy-based management approach to security in cloud systems.* [PhD Thesis, De Montfort University]. https://www.academia.edu/download/63160233/phd20200501-24243-1y48y0a.pdf

Aldossary, S., & Allen, W. (2016). Data security, privacy, availability and integrity in cloud computing: Issues and current solutions. *International Journal of Advanced Computer Science and Applications*, *7*(4). https://pdfs.semanticscholar.org/09fd9d75103ddd6550176c10e0ba3.pdf

Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, *12*(6), 1333. https://www.mdpi.com/2079-9292/12/6/1333

Cangialosi, F., Chung, T., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., & Wilson, C. (2016). Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 628–640. https://doi.org/10.1145/2976749.2978301

Ngwenya, S., Esan, O. A., & Elegbeleye, F. A. (2024). An Evaluation of Security Issues in Cloud-Based File Sharing Technologies. *2024 4th International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 304–314. https://ieeexplore.ieee.org/abstract/document/10851167/

Nissim, N., Yahalom, R., & Elovici, Y. (2017). USB-based attacks. *Computers & Security*, *70*, 675–688. https://www.sciencedirect.com/science/article/pii/S0167404817301578

Sakthivel, S. G., & Ramalingam, S. (2025). Network security threat prevention in step based file transfer system. *AIP Conference Proceedings*, *3279*(1), 020105. https://pubs.aip.org/aip/acp/article-abstract/3279/1/020105/3341779

Santhoshkumar, S. P., Hariharasudhan, S., Prakash, S. P., Philip, J. M., Haritha, S., & Nalini, T. (2024). Security Challenges and Elucidations in Cloud Storage and File Systems: An Advanced Investigation Review. *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, 1257–1262. https://ieeexplore.ieee.org/abstract/document/10866357/

Abwnawar, N. (2020). *A policy-based management approach to security in cloud systems.* [PhD Thesis De Montfort University]. https://www.academia.edu/download/63160233/phd20200501-24243-1y48y0a.pdf

Barona, R., & Anita, E. M. (2017). A survey on data breach challenges in cloud computing security: Issues and threats. *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 1–8. https://ieeexplore.ieee.org/abstract/document/8074287/

Cangialosi, F., Chung, T., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., & Wilson, C. (2016). Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 628–640. https://doi.org/10.1145/2976749.2978301

Carlin, A., Hammoudeh, M., & Aldabbas, O. (2015). Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *International Journal of Advanced Computer Science and Applications*, *6*(6). https://e-space.mmu.ac.uk/618160/

Fryer, H. (2016). *The public health analogy in web security* [PhD Thesis, University of Southampton]. https://eprints.soton.ac.uk/412399/

Gozman, D., & Willcocks, L. (2019). The emerging Cloud Dilemma: Balancing innovation with cross-border privacy and outsourcing regulations. *Journal of Business Research*, *97*, 235–256. https://www.sciencedirect.com/science/article/pii/S0148296318302935

Haq, I. U., Ramzan, S., Ahmad, N., Ahmad, Y., & Nadeem, A. (2024). Towards Robust and Low Latency Security Framework for IEEE 802.11 Wireless Networks. *International Journal of Computing and Digital Systems*. https://www.researchgate.net/profile/Inam-Ul-Haq-14/publication/377204583\_International\_Journal\_of\_Computing\_and\_Digital\_Systems\_Towards\_Robust\_and\_Low\_Latency\_Security\_Framework\_for\_IEEE\_80211\_Wireless\_Networks/links/659967966f6e450f19cfedfe/International-Journal-of-Computing-and-Digital-Systems-Towards-Robust-and-Low-Latency-Security-Framework-for-IEEE-80211-Wireless-Networks.pdf

Huang, J., Nicol, D. M., Bobba, R., & Huh, J. H. (2012). A framework integrating attribute-based policies into role-based access control. *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*. https://doi.org/10.1145/2295136.2295170

Hull, J., & Markham, M. (2014). *RBAC Driven Least Privilege Architecture For Control Systems*. Honeywell International Inc., Golden Valley, MN (United States). https://www.osti.gov/biblio/1124080

Lin, T., Capecci, D. E., Ellis, D. M., Rocha, H. A., Dommaraju, S., Oliveira, D. S., & Ebner, N. C. (2019). Susceptibility to Spear-Phishing Emails: Effects of Internet User Demographics and Email Content. *ACM Transactions on Computer-Human Interaction*, *26*(5), 1–28. https://doi.org/10.1145/3336141

Lockhart, J., Xesi, X., & Chiware, E. R. (2024). Working towards securing and building a trusted institutional research data repository through the CoreTrustSeal process: Case of Cape Peninsula University of Technology data repository. *IASSIST Quarterly*, *48*(3). https://iassistquarterly.com/index.php/iassist/article/view/1111

McKay, K., & Cooper, D. (2017). *Guidelines for the selection, configuration, and use of transport layer security (TLS) implementations*. National Institute of Standards and Technology. https://csrc.nist.gov/CSRC/media/Publications/sp/800-52/rev-2/draft/documents/sp800-52r2-draft.pdf

Nguyen, M. T., & Tran, Q. H. (2023). An Analysis of Multi-Layered Cybersecurity Architectures: Ensuring Data Confidentiality, Integrity, and Availability Through Structured Defense Mechanisms. *Journal of Data Science, Big Data Analytics, and Predictive Modeling*, *13*(6), 1–14. https://kernpublic.com/index.php/IJDSBDAPM/article/view/2023-june-04

Nissim, N., Yahalom, R., & Elovici, Y. (2017). USB-based attacks. *Computers & Security*, *70*, 675–688. https://www.sciencedirect.com/science/article/pii/S0167404817301578

Pagallo, U. (2012). Cracking down on autonomy:challenges to design in IT Law. *Ethics and Information Technology*, *14*(4), 319–328. https://doi.org/10.1007/s10676-012-9295-9

Patwary, A. A.-N., Fu, A., Naha, R. K., Battula, S. K., Garg, S., Patwary, M. A. K., & Aghasian, E. (2020). *Authentication, Access Control, Privacy, Threats and Trust Management Towards Securing Fog Computing Environments: A Review* (No. arXiv:2003.00395). arXiv. https://doi.org/10.48550/arXiv.2003.00395

Sandhu, R., & Zhang, X. (2005). Peer-to-peer access control architecture using trusted computing technology. *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, 147–158. https://doi.org/10.1145/1063979.1064005

Selvam, H. H. S., Hanawal, M. K., & Kulkarni, S. G. (2024). Privacy Performance Trade-off in Web Services. *2024 IEEE 49th Conference on Local Computer Networks (LCN)*, 1–7. https://ieeexplore.ieee.org/abstract/document/10639729/

Shu, J. (2024). Network Storage Architectures. In J. Shu, *Data Storage Architectures and Technologies* (pp. 149–183). Springer Nature Singapore. https://doi.org/10.1007/978-981-97-3534-1\_7

SYED, S. (2025). *Blockchain Enabled Approach to Secure and Scalable Patient E-Health Data*. https://www.researchgate.net/profile/Shoeb-Ali-Syed/publication/390302985\_Blockchain\_Enabled\_Approach\_to\_Secure\_and\_Scalable\_Patient\_E-Health\_Data/links/67e8c6fe03b8d7280e11afbc/Blockchain-Enabled-Approach-to-Secure-and-Scalable-Patient-E-Health-Data.pdf

Wenhua, Z., Kamrul Hasan, M., Ismail, A. F., Yanke, Z., Razzaque, M. A., Islam, S., & Anil Kumar, B. (2023). Data security in smart devices: Advancement, constraints and future recommendations. *IET Networks*, *12*(6), 269–281. https://doi.org/10.1049/ntw2.12091

**APPENDICES**

**Appendix A:** Flask app

from app import create\_app, db from app.models import User from app import bcrypt import urllib3

app = create\_app()

app.secret\_key = 'File\_sharing\_https'

# Disable SSL warnings for self-signed certificates

urllib3.disable\_warnings(urllib3.exceptions.InsecureRequestWarning) def create\_test\_users():

"""Create test users for development""" try:

# Create admin user

admin = User.query.filter\_by(username='admin').first() if not admin: admin = User( username='admin', first\_name='Admin', last\_name='User', email='admin@test.com', intake\_year='2024-2025',

password=bcrypt.generate\_password\_hash('admin123').decode('utf-8'), group='leadership', role='admin', department='cs', status='active'

)

db.session.add(admin)

print("Created admin user: admin/admin123")

# Create test user

user = User.query.filter\_by(username='testuser').first() if not user: user = User( username='testuser', first\_name='Test', last\_name='User', email='test@test.com', intake\_year='2024-2025',

password=bcrypt.generate\_password\_hash('test123').decode('utf-8'), group='student', role='user', department='cs', status='active'

)

db.session.add(user)

print("Created test user: testuser/test123")

db.session.commit()

print("Test users created successfully!") except Exception as e: print(f"Error creating test users: {e}") db.session.rollback()

if \_\_name\_\_ == '\_\_main\_\_': with app.app\_context(): db.create\_all() create\_test\_users()

# Configure for cross-server access import socket

hostname = socket.gethostname() local\_ip = socket.gethostbyname(hostname)

# Enable request logging import logging

logging.basicConfig(level=logging.DEBUG) app.logger.setLevel(logging.DEBUG)

#app.run(host='0.0.0.0', port=5000, debug=True)

app.run(ssl\_context=('certs/cert.pem', 'certs/key.pem'), host='0.0.0.0', debug=True, port=443)

**Appendix B:** Questions for data collection

1. How do you currently share files with your colleagues or students at INES-Ruhengeri?
2. What challenges have you personally faced when sharing files using email, USB drives, or cloud services?
3. In your opinion, what are the biggest risks of sharing sensitive institutional data using current methods?
4. Can you describe any situation (real or hypothetical) where you think file sharing might lead to a data security problem?
5. What concerns do you have about using public file sharing services (like Google Drive, email, or Dropbox) in an academic setting?
6. What are your thoughts on using a secure LAN-based system (inside INES-Ruhengeri) for file sharing?
7. How do you think HTTPS encryption could improve the safety of file transfers within the institution?
8. What features would you like to see in a secure file sharing system at INES-Ruhengeri?