FILEHUB

Secure File Sharing

NOVEMBER 20, 2025

HORIZONSEC

Submitted to: Maam Sadia

Contents

Introduction 2

  Requirements Table 2

Pre-Requisites 3

Technologies Used 3

Setup 4

Frontend (Client Side).. 5

Introduction

This is a Secure File Sharing semester project for the Secure Software Design course. In this project, as per the proposal and as cybersecurity students, we were tasked to create a file sharing webapp with security aspects implemented throughout the implementation as well as the deployment phase.

Requirements Table

Functional Requirement (FR)	**Security Non-Functional Requirement (NFR)**
FR1 — Account creation by Admin: Only Admins can create accounts for students, teachers, and staff.	**NFR1**: The account will be made with random password. Default passwords must be system-generated, strong (min 8 chars, mixed charset), and transmitted securely (OpenSSL/TLS 1.2+). Stored passwords must be hashed using a strong hashing algorithm.
FR2 — Mandatory password change: Users must change their password at first login.	**NFR2**: Passwords must meet best complexity rules (length, characters).
FR3 — Role-based access control (RBAC): Roles include **Admin**, **Teacher**, **Student**, **Staff** with different permissions (e.g., Teachers can share course files with students, Students only upload assignments). Students should not be able to see what other student upload.	**NFR3**: RBAC enforced server-side with least privilege principle; privilege escalation attempts logged; role assignments auditable.
FR4 — File upload: Teachers and students can upload files with restrictions.	**NFR4**: Only allow specific file extensions (.pdf, .docx, .pptx, .xlsx, .zip); enforce antivirus scan and file size limits to max 25MB; files encrypted at rest e.g. AES-256.
FR5 — File download: Authorized users can download shared files.	**NFR5**: All downloads over OpenSSL/TLS; files decrypted only for authorized users.
FR6 — File sharing permissions: Teachers can share files with students; Admins/Staff can share institutional documents. Students	**NFR6**: Sharing restricted by role policy; share links must be time-bound and

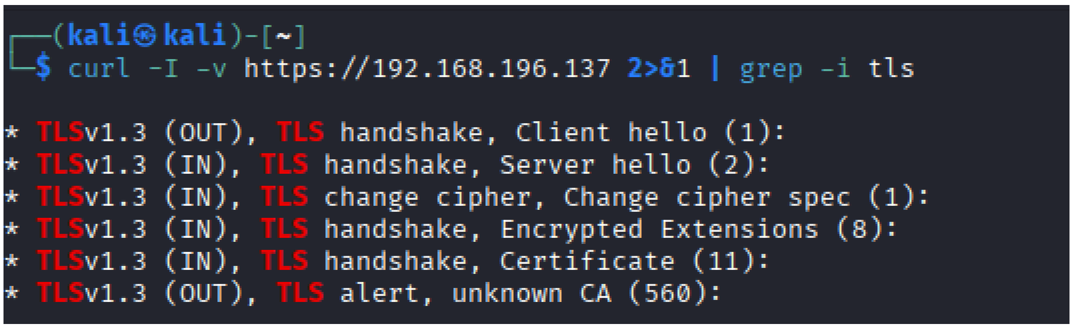
Functional Requirement (FR)	Security Non-Functional Requirement (NFR)
can only share with teachers (e.g., assignments).	revocable; actions logged with timestamp, actor, and recipient.
FR7 — Audit logging: Log all file uploads/downloads, logins, password changes, and account management actions.	**NFR7**: Logs must be immutable, timestamped, and encrypted; access to logs restricted to Admins.
FR8 — File integrity verification: Uploaded files are hashed and verified on download.	**NFR8**: SHA-256 or stronger hashing; mismatch triggers alerts; system will not allow the download of such files.
FR9 — Secure key management: Encryption keys for files are securely generated and stored.	**NFR9**: Keys managed in secure key vault; keys never stored in plaintext.
FR10 — Access to metadata and search: Users can view and search only the files they have uploaded/shared and the files explicitly shared with them. The combined file list appears in their dashboard with search and filter options.	**NFR10**: The system will only send the list of the files that are shared with certain user type from which the user can search the desired file.
FR11 — Administrative controls: Admins can deactivate/reactivate accounts, reset passwords, and enforce policies.	**NFR11**: All admin actions require MFA; sensitive changes require justification logging; Admin dashboard accessible only from trusted networks.

Pre-Requisites

- Ubuntu machine to host a web server
- Another machine, Windows/Linux, to test the client side
- Enough brain cells (x2 minimum) to do this project alone

Technologies Used

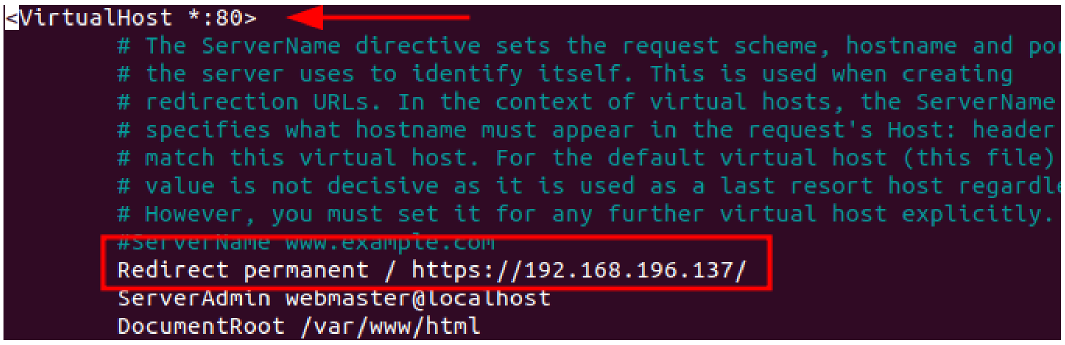
- Apache2 (apache2 -v)
    - Server version: Apache/2.4.41 (Ubuntu)
    - Server built: 2025-04-02T18:34:29

- TLS v1.3
  - 

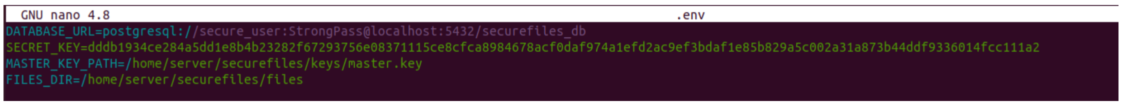
```
┌──(kali㉿kali)-[~]
└─$ curl -I -v https://192.168.196.137 2>&1 | grep -i tls

* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS change cipher, Change cipher spec (1):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (OUT), TLS alert, unknown CA (560):
```

- Python (Flask, Reverse Proxy)
- Postgresql db

Setup

- Run ubuntu in a vm
- Install apache2
- Configure TLS
- Generate a self-signed cert Digital Cert to run the website on TLSv1.3 by default
- Making sure that the *http* requests are redirected to *https* automatically
  - sudo nano /etc/apache2/sites-available/000-default.conf
  - 

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and po
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header
        # match this virtual host. For the default virtual host (this file)
        # value is not decisive as it is used as a last resort host regardle
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com
        Redirect permanent / https://192.168.196.137/
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
```

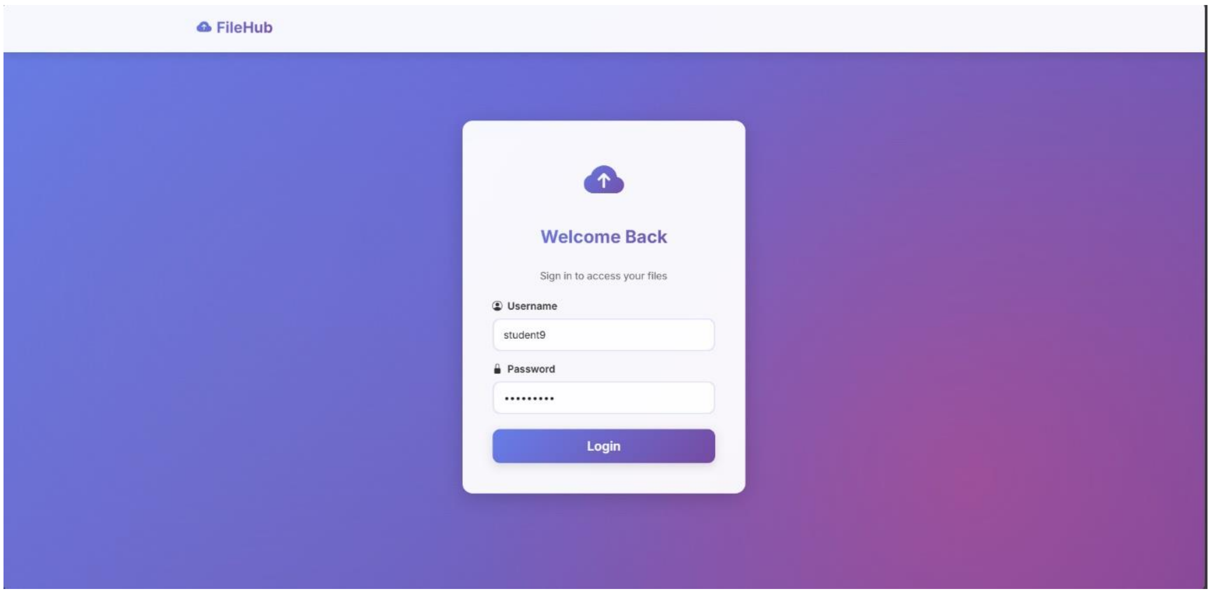
- Setting up environment variable file (.env)
  - 

```
  GNU nano 4.8                                                     .env
DATABASE_URL=postgresql://secure_user:StrongPass@localhost:5432/securefiles_db
SECRET_KEY=dddb1934ce284a5dd1e8b4b23282f67293756e08371115ce8cfca8984678acf0daf974a1efd2ac9ef3bdaf1e85b829a5c002a31a873b44ddf9336014fcc111a2
MASTER_KEY_PATH=/home/server/securefiles/keys/master.key
FILES_DIR=/home/server/securefiles/files
```

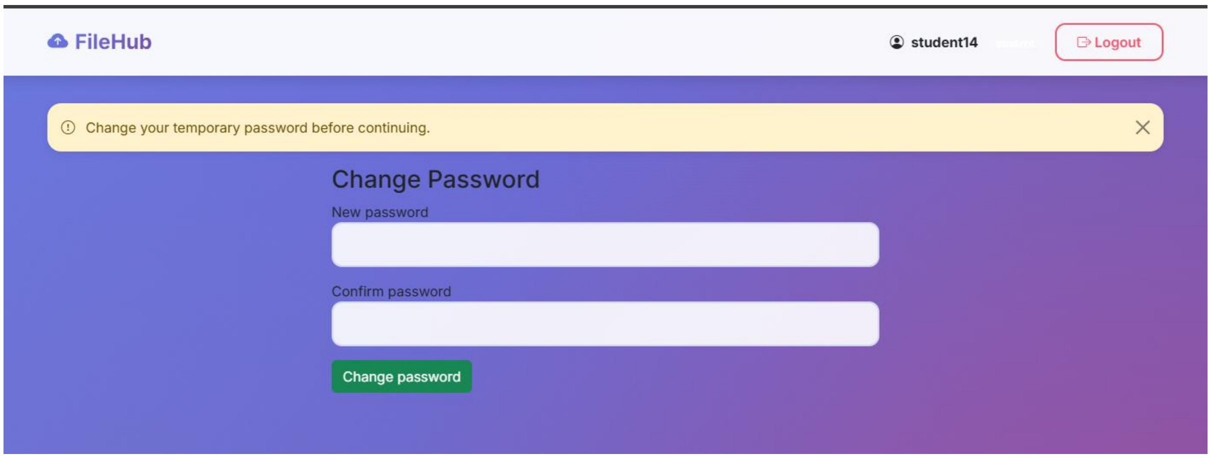
- Installing Postgresql
- Creating a db user 'db_user' with pass 1234
  - To test the creation and setup of db
    - psql -U secure_user -d securefiles_db -h localhost -W
- Using a Python-based webapp, running it locally and than using Reverse Proxy to use Apache2 webserver with TLS1.3 as front to the actual webapp

Frontend (Client Side)

- Login Page, the very first page a user sees to access the files shared with them or upload new files

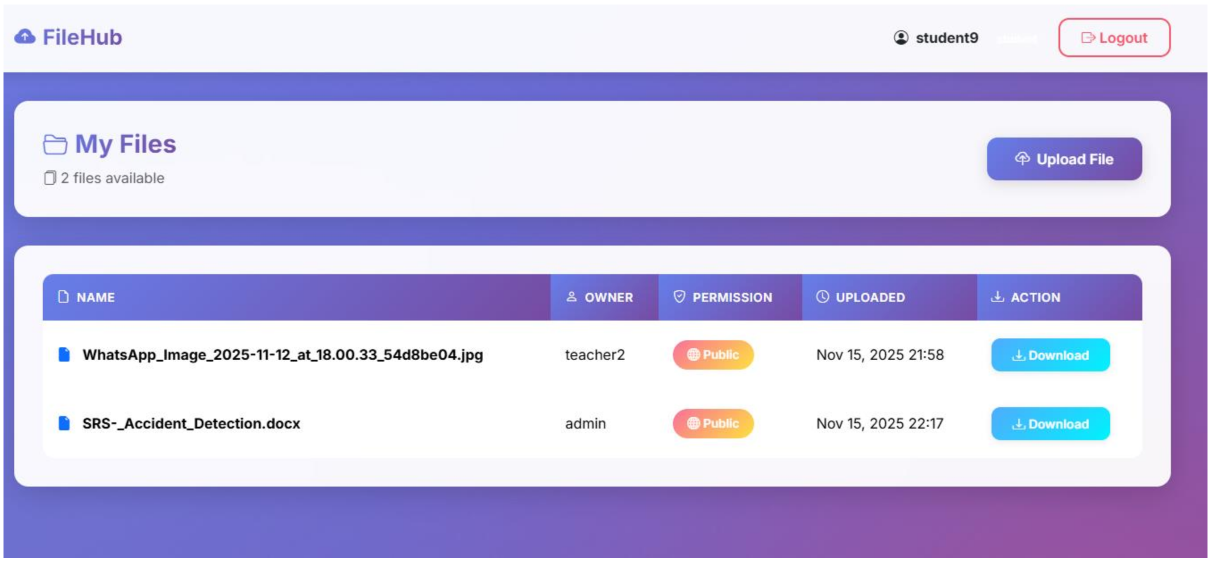


- After the **first** successful login, the user is redirected to another page to change the temporary password



Change your temporary password before continuing.

- After changing the password, the user will see the dashboard where they can see the files shared with them or a button with which they can upload a new file.

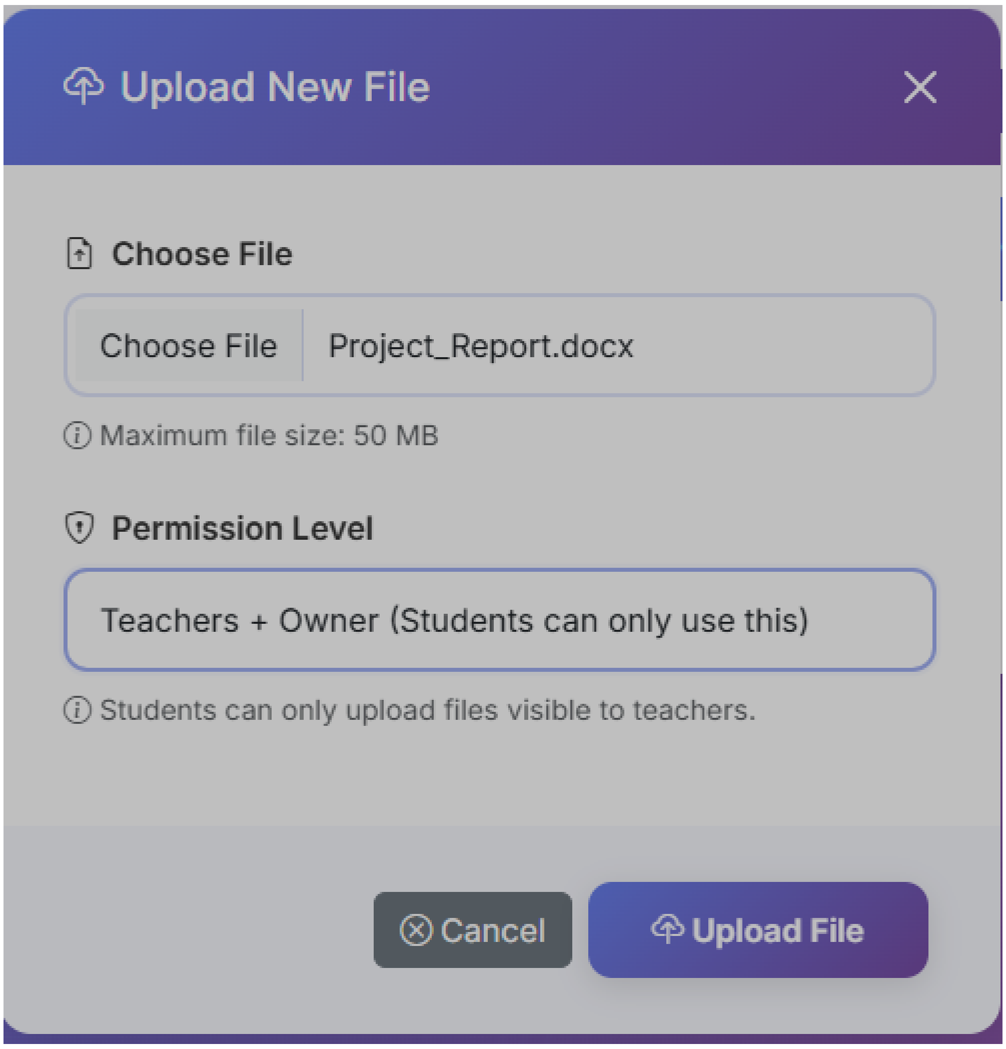


FileHub — student9 — Logout

My Files — 2 files available — Upload File

NAME	OWNER	PERMISSION	UPLOADED	ACTION
WhatsApp_Image_2025-11-12_at_18.00.33_54d8be04.jpg	teacher2	Public	Nov 15, 2025 21:58	Download
SRS-_Accident_Detection.docx	admin	Public	Nov 15, 2025 22:17	Download

- During upload the user can choose any file but with limited extensions and size and can set Access Control of the file.



Upload New File

Choose File

Choose File Project_Report.docx

Maximum file size: 50 MB

Permission Level

Teachers + Owner (Students can only use this)

Students can only upload files visible to teachers.

Cancel — Upload File

-

- After successful upload the file will be shown in their dashboard as well as the user they shared it with.