# Capstone Project Report

**Project Title:** *Secure Network File Sharing System*
**Course / Module:** Linux System Programming (LSP)
**Student Name:** Omkar Bhuyan
**Date of Submission:** 09 November 2025
**Institution / Department:** [Institute of Technical Education and Research / Computer Science and Engineering]

## A brief overview of project

The *Secure Network File Sharing System* is a Linux-based client–server application that enables users to securely share files over a network. It implements essential features such as file listing, upload, download, and secure communication through encryption. The project is built using C++ socket programming and demonstrates core concepts of the Linux System Programming (LSP) module including inter-process communication, file handling, and network programming.

# Introduction

This project aims to design and implement a simple yet secure file-sharing system using Linux socket programming. It provides a client-server model where users can authenticate, upload, or download files from a remote system. The project showcases the integration of multiple Linux concepts — system calls, socket APIs, encryption, and persistent storage  to simulate a mini network service.

**Key Objectives:**

- To understand Linux system and socket programming
- To develop a client-server file-sharing model
- To implement secure authentication and data encryption
- To enable basic file operations (upload, download, list)

## Tools & Technologies Used

| Tool / Component | Description |
|---|---|
| **Operating System** | Ubuntu (via WSL) |
| **Programming Language** | C++ |
| **Libraries Used** | `<arpa/inet.h>`, `<unistd.h>`, `<fstream>`, `<thread>`, `<dirent.h>` |
| **Encryption Used** | Simple XOR encryption |
| **Version Control** | GitHub Repository |

## System Architecture

```
+-------------+          TCP Socket           +-------------+
|   Client    | <---------------------------> |   Server    |
| (Upload/Download)                            | (File Storage & Auth)
|
+-------------+                              +-------------+
```

## Workflow Summary:

1. The client connects to the server using TCP sockets.
2. User authentication is verified via credentials in `server/users.txt`.
3. Upon successful login, users can list, upload, or download files.
4. All messages are encrypted using XOR for basic data protection.

## Implementation Details

### Key Modules:

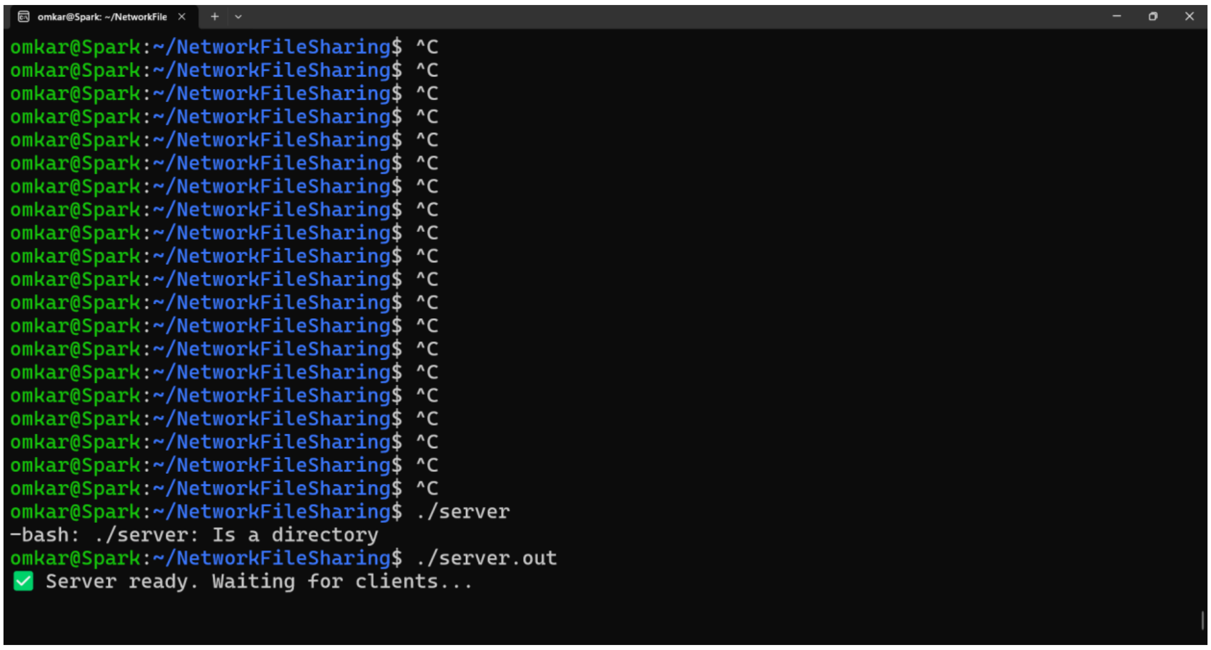
- **Server:**
  Handles authentication, file operations, and responds to client commands.
- **Client:**
  Provides menu-driven options for login, upload, and download.
- **Encryption Module:**
  Uses XOR to encrypt/decrypt communication data.
- **Persistent Login:**
  Credentials are verified from a file (`server/users.txt`) instead of hard-coded values.

## Code Structure

```
NetworkFileSharing/
│
├── server/
│   ├── server.cpp
│   ├── server_files/
│   │   ├── file1.txt
│   │   └── file2.txt
│   └── users.txt
│
├── client/
│   ├── client.cpp
│   └── client_downloads/
│
└── README.md
```

## Output Screenshots

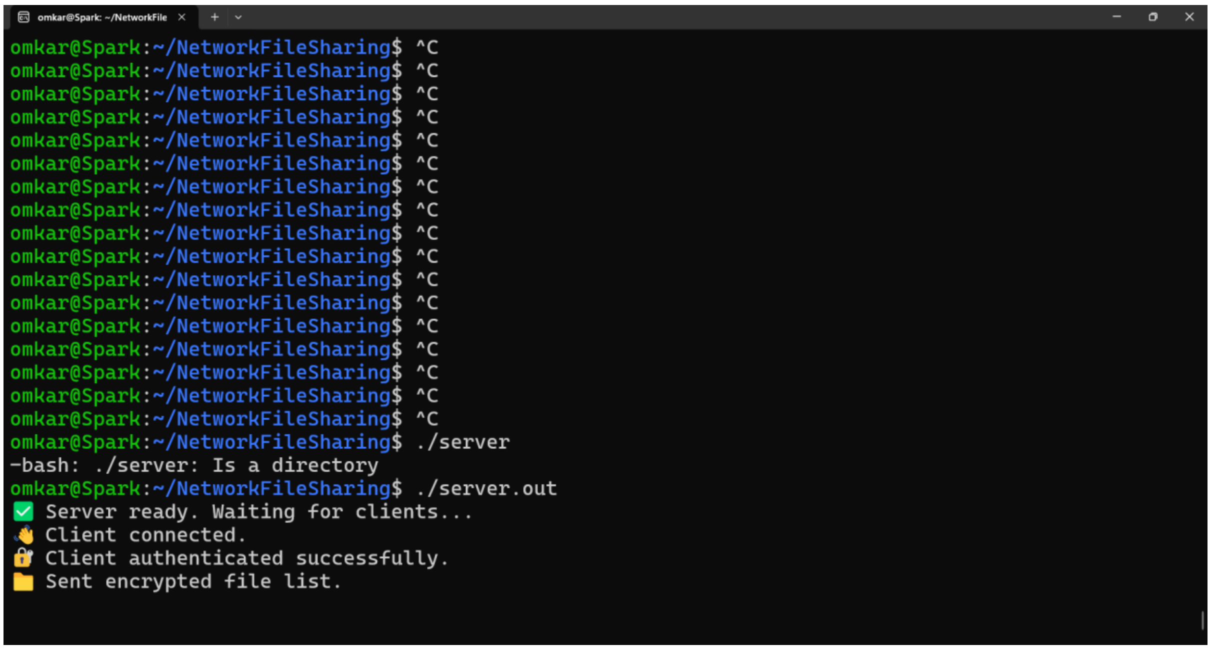
### -> Server starting up



```
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ./server
-bash: ./server: Is a directory
omkar@Spark:~/NetworkFileSharing$ ./server.out
✅ Server ready. Waiting for clients...
```

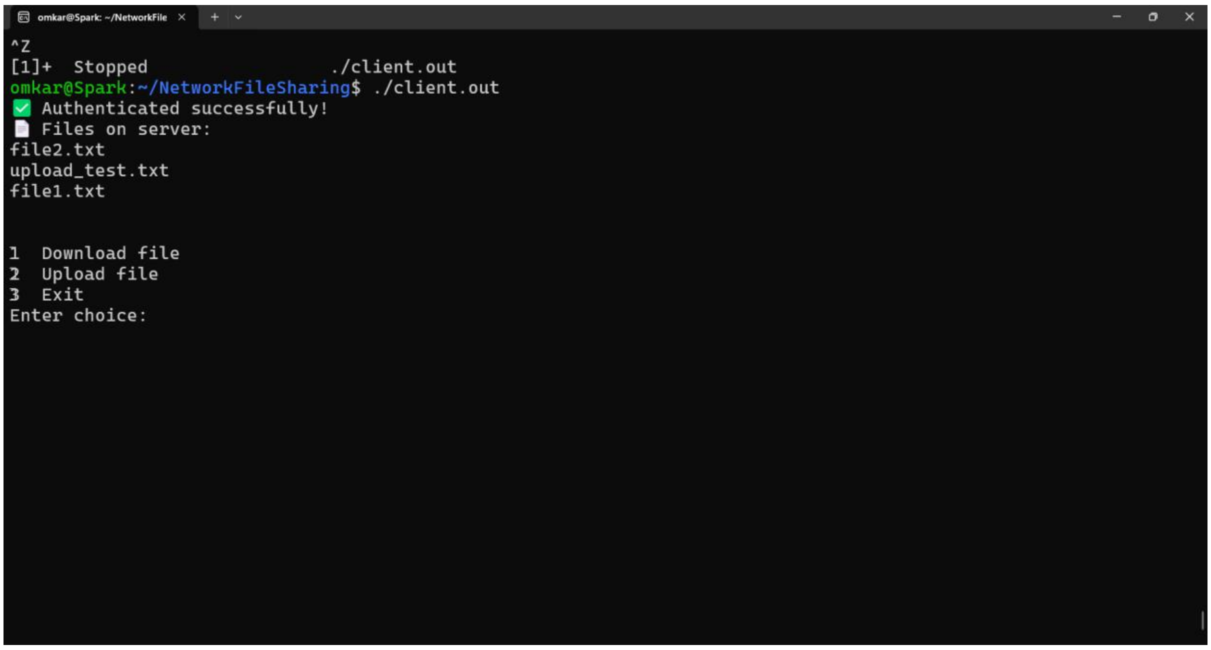
Server initialized and waiting for client connections

### -> Successful authentication



```
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ./server
-bash: ./server: Is a directory
omkar@Spark:~/NetworkFileSharing$ ./server.out
✅ Server ready. Waiting for clients...
👋 Client connected.
🔐 Client authenticated successfully.
📁 Sent encrypted file list.
```

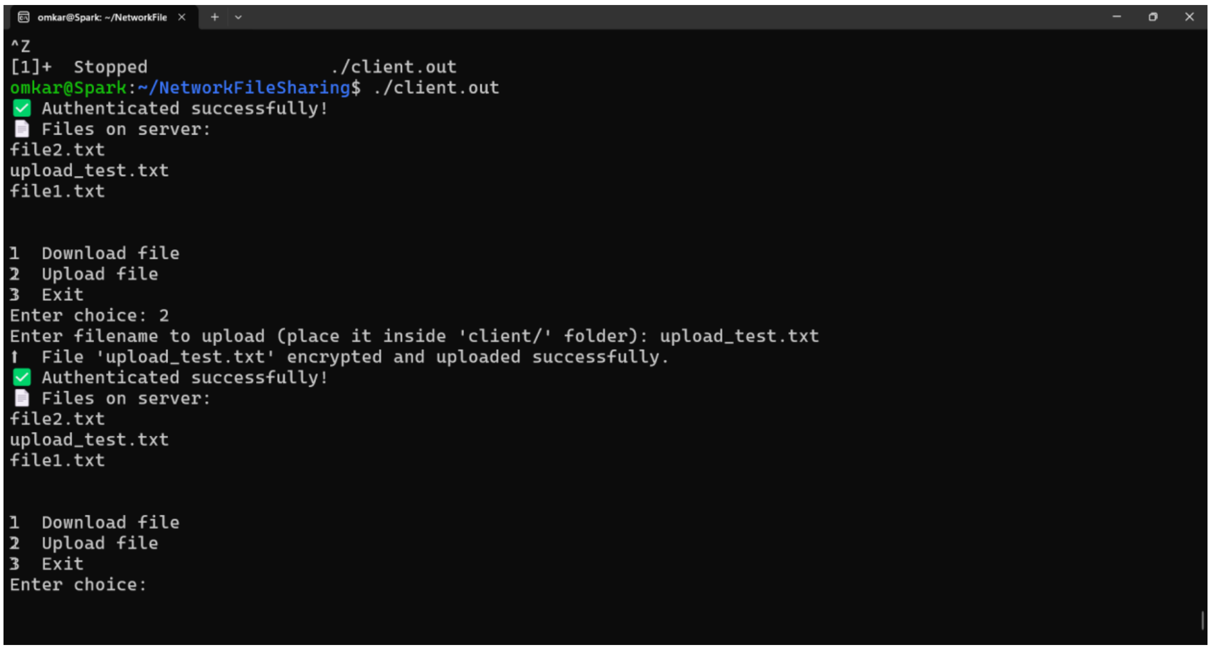
-> File listing output



```
^Z
[1]+  Stopped                 ./client.out
omkar@Spark:~/NetworkFileSharing$ ./client.out
✅ Authenticated successfully!
📄 Files on server:
file2.txt
upload_test.txt
file1.txt


1  Download file
2  Upload file
3  Exit
Enter choice:
```

-> Uploading a file

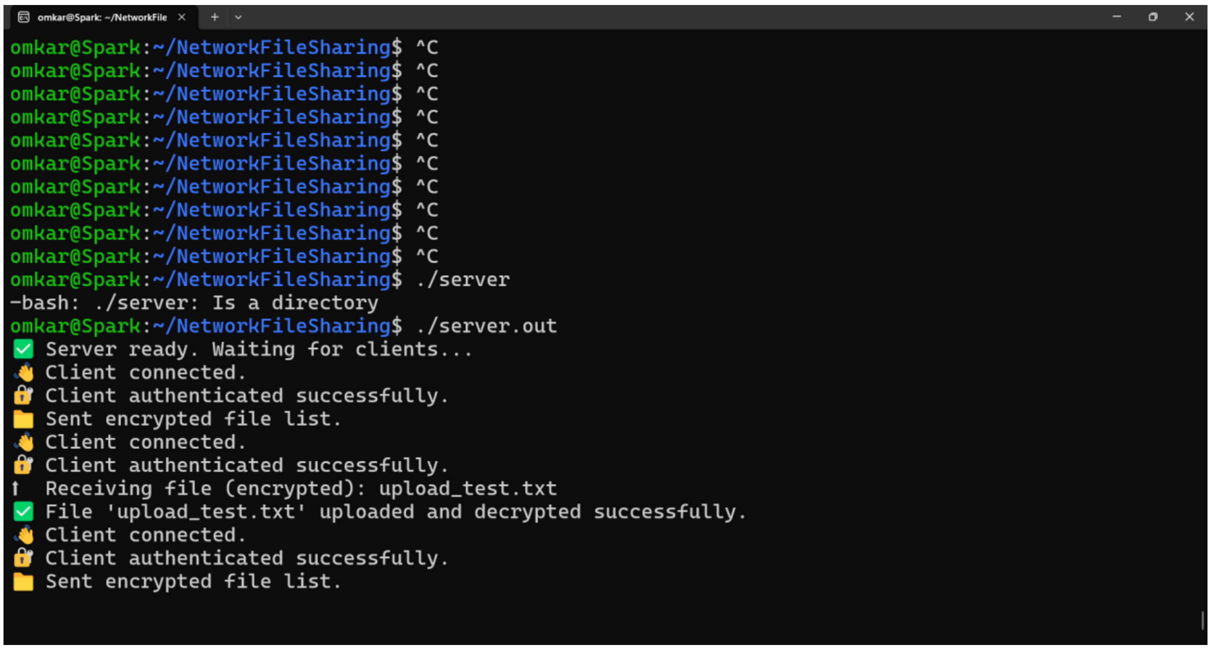


```
^Z
[1]+  Stopped                 ./client.out
omkar@Spark:~/NetworkFileSharing$ ./client.out
✅ Authenticated successfully!
📄 Files on server:
file2.txt
upload_test.txt
file1.txt


1  Download file
2  Upload file
3  Exit
Enter choice: 2
Enter filename to upload (place it inside 'client/' folder): upload_test.txt
⬆  File 'upload_test.txt' encrypted and uploaded successfully.
✅ Authenticated successfully!
📄 Files on server:
file2.txt
upload_test.txt
file1.txt


1  Download file
2  Upload file
3  Exit
Enter choice:
```

-> Downloading a file



omkar@Spark: ~/NetworkFile

```
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ^C
omkar@Spark:~/NetworkFileSharing$ ./server
-bash: ./server: Is a directory
omkar@Spark:~/NetworkFileSharing$ ./server.out
✅ Server ready. Waiting for clients...
👋 Client connected.
🔐 Client authenticated successfully.
📁 Sent encrypted file list.
👋 Client connected.
🔐 Client authenticated successfully.
⬆  Receiving file (encrypted): upload_test.txt
✅ File 'upload_test.txt' uploaded and decrypted successfully.
👋 Client connected.
🔐 Client authenticated successfully.
📁 Sent encrypted file list.
```

## Challenges & Learning

During development, challenges included handling multiple socket connections, implementing file transfer integrity, and adding simple encryption.
The project helped strengthen understanding of socket programming, Linux file handling, and modular C++ design.

## Conclusion

The *Secure Network File Sharing System* demonstrates the practical implementation of Linux networking and file management concepts. With added security and authentication, it serves as a foundational model for real-world file transfer systems.