****

**PROJECT**

**NETWORK FILE SYSTEM**

**COURSE:**

Operating System

**SUBMITTED TO:**

Prof. Hassan Rauf

**SUBMITTED BY:**

Name : Laiba

Roll no. : 229

Section : D

BSCS(2022-2026)

**SUBMITTED ON**:

3-MAR-2025

**File Management System - User Manual**

**1. Introduction**

The File Management System is a web-based application that allows users to upload, list, view, download, and delete files from a server. It follows a client-server architecture using Flask as the backend and a socket-based communication protocol for file management.

**2. Features**

* Upload Files: Users can upload files through a web interface.
* List Files: Displays all available files with timestamps.
* View Files: Open files in a new tab for quick preview.
* Download Files: Download stored files with their original names.
* Delete Files: Remove unwanted files from the server.
* Log System: Tracks all file-related activities.

**3. System Requirements**

* Python 3.x
* Flask framework
* A web browser (Chrome, Firefox, Edge, etc.)

**Step 1: System Configuration & Code Setup**

Before running the project, ensure that your system is correctly configured.

**1.1 Prerequisites**

* Operating System: Windows
* Python Version: 3.x (Ensure Python is installed)
* Required Software: VS Code (or any code editor), Terminal (VS code or command prompt)

**1.2 Setting Up Project Directory**

Create a project folder and place all files accordingly:

OS project/

│── server.py

│── myclient.py

│── interface.py

│── templates/ (for HTML files)

│── static/ (for CSS, JS if needed)

│── downloads/ (stores downloaded files)

│── server\_files/ (stores uploaded files)

│── server\_logs.txt (logs all activities)

│── active\_users.txt (tracks active users)

**Step 2: Install Required Dependencies**

Since the project uses Flask, you need to install the necessary Python libraries.

**2.1 Install pip (if not installed)**

python -m ensurepip --default-pip

**2.2 Install Flask and Other Modules**

Run the following command in the terminal:

pip install flask

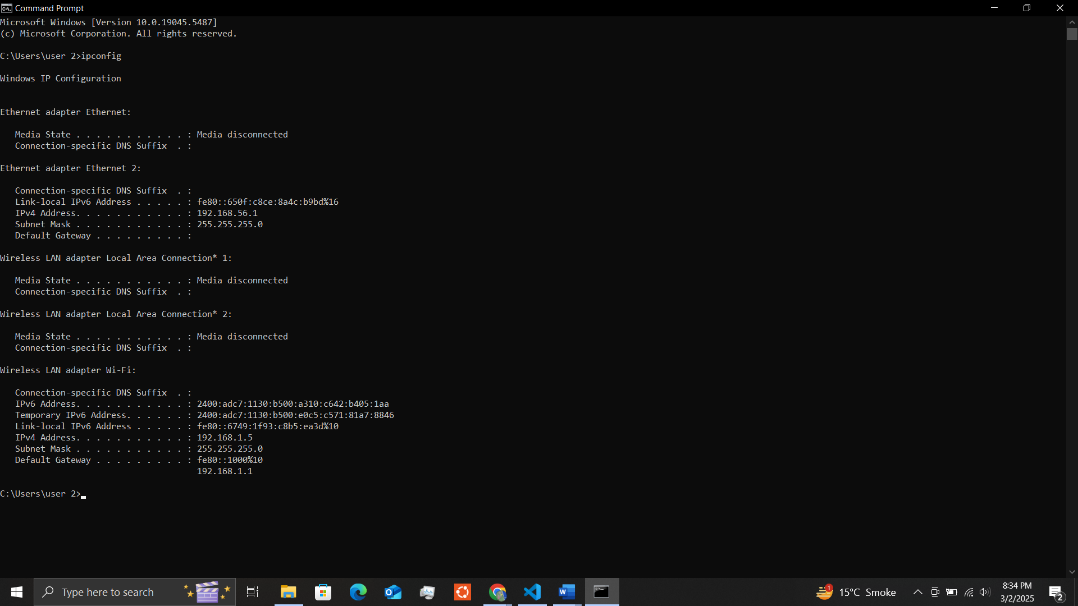
This ensures that Flask is available to run your application.

**Step 3: Change the configuration**

The configuration is applied according to the system according to your system.

**3.1 Open the command prompt and write the following command**

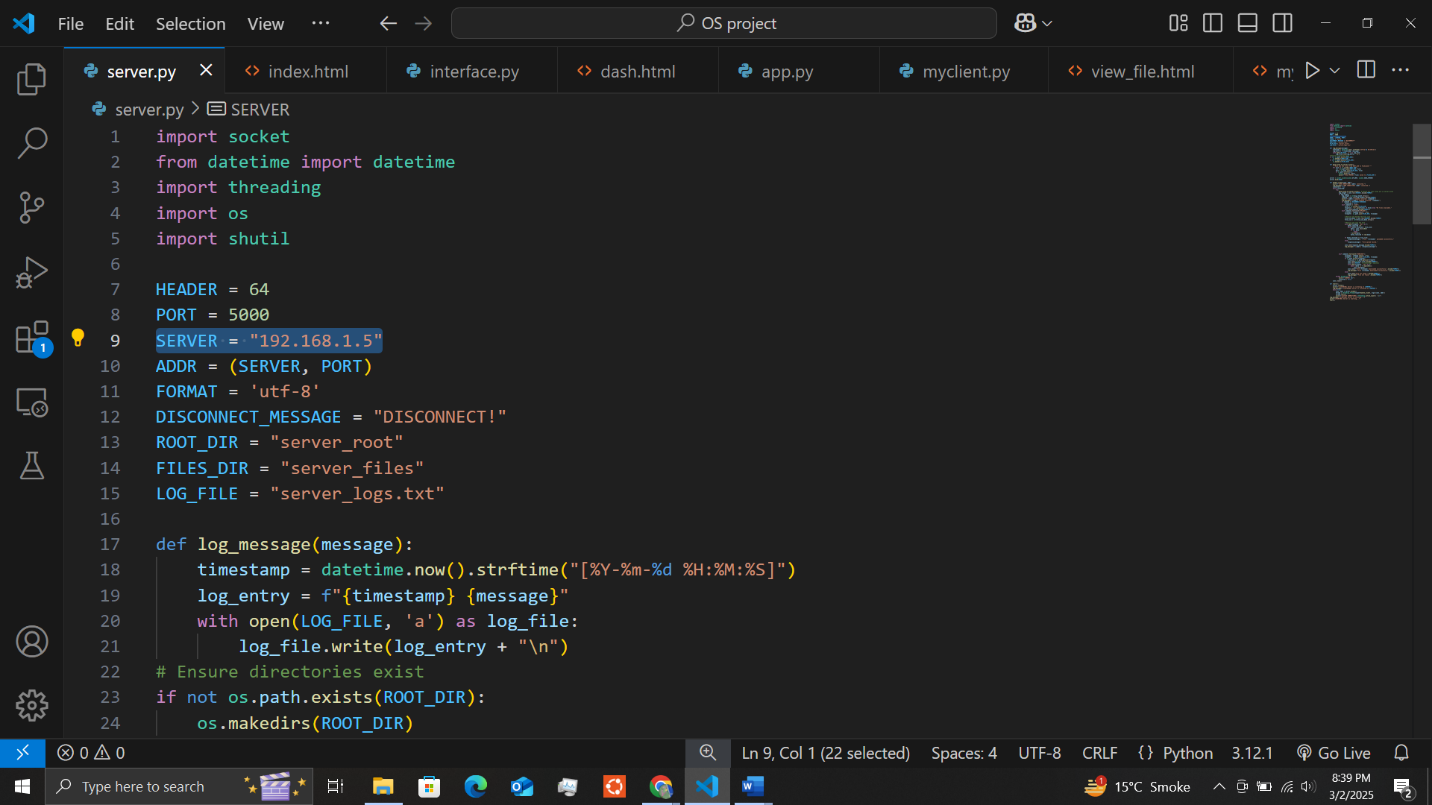
ipconfig



The IPv4 address in the wireless section will be changed on both server (server.py) and client (myclient.py).

**Sequence of running the files:**

**server.py---> interface.py---> myclient.py**



**Step 4: Running the Server (server.py)**

The server handles requests from the client and manages file storage.

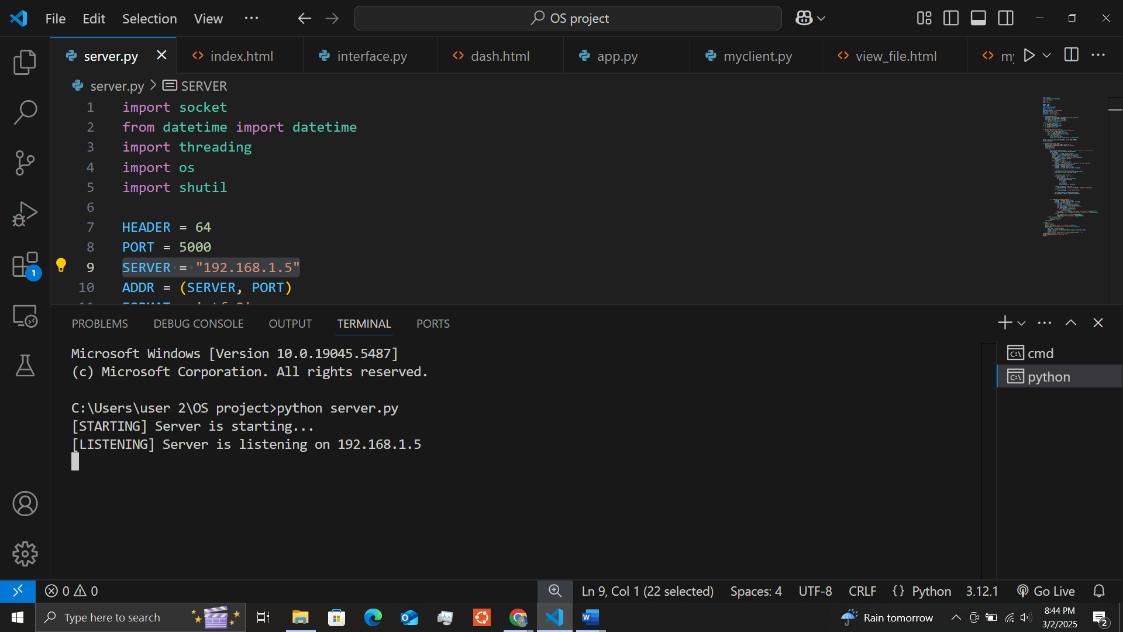
**4.1 Open a new terminal and navigate to the project folder**

cd path\to\ Project

**4.2 Run the server script**

python server.py

**This will start the server, which will listen for incoming connections.  
You should see:**



**Step 5: Running the Interface (interface.py)**

The web interface provides a dashboard for users to manage files.

**5.1 Open a new terminal and navigate to the project folder**

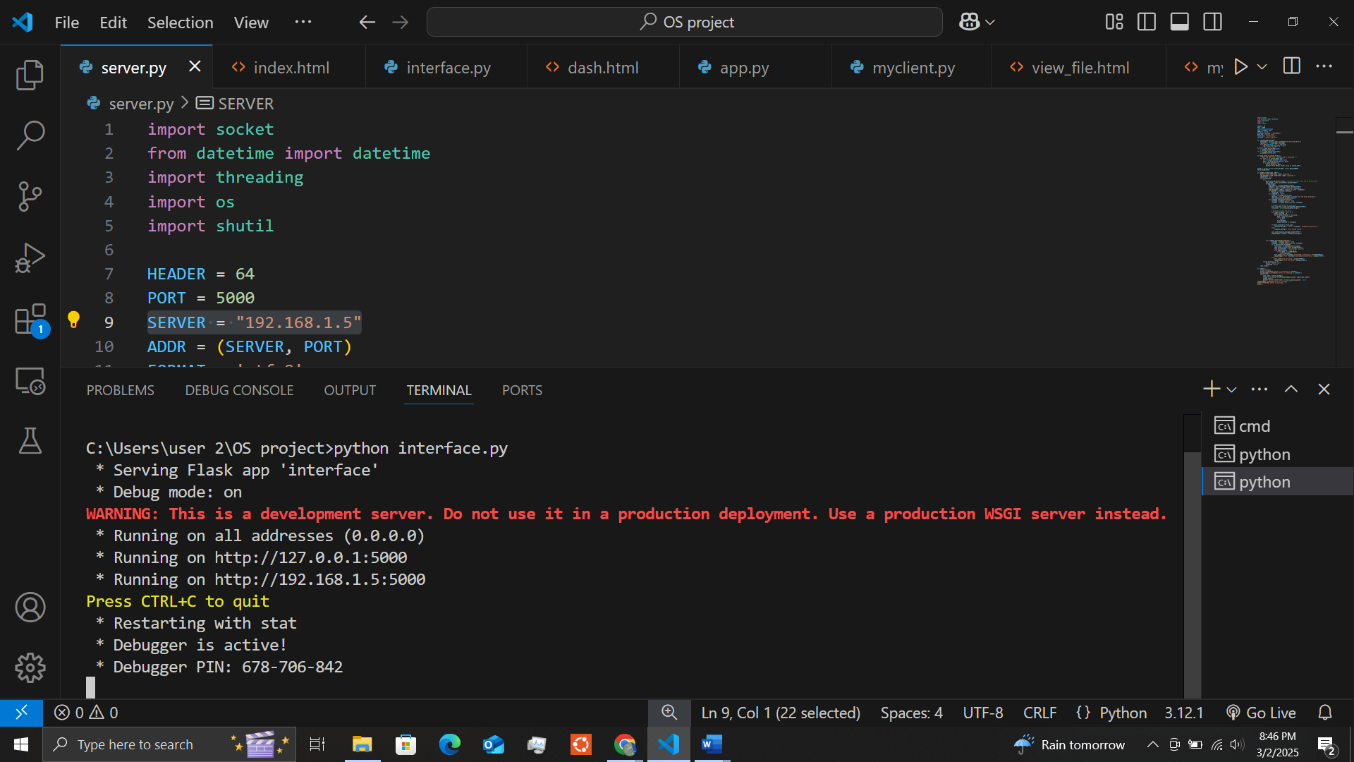
cd path\to\ Project

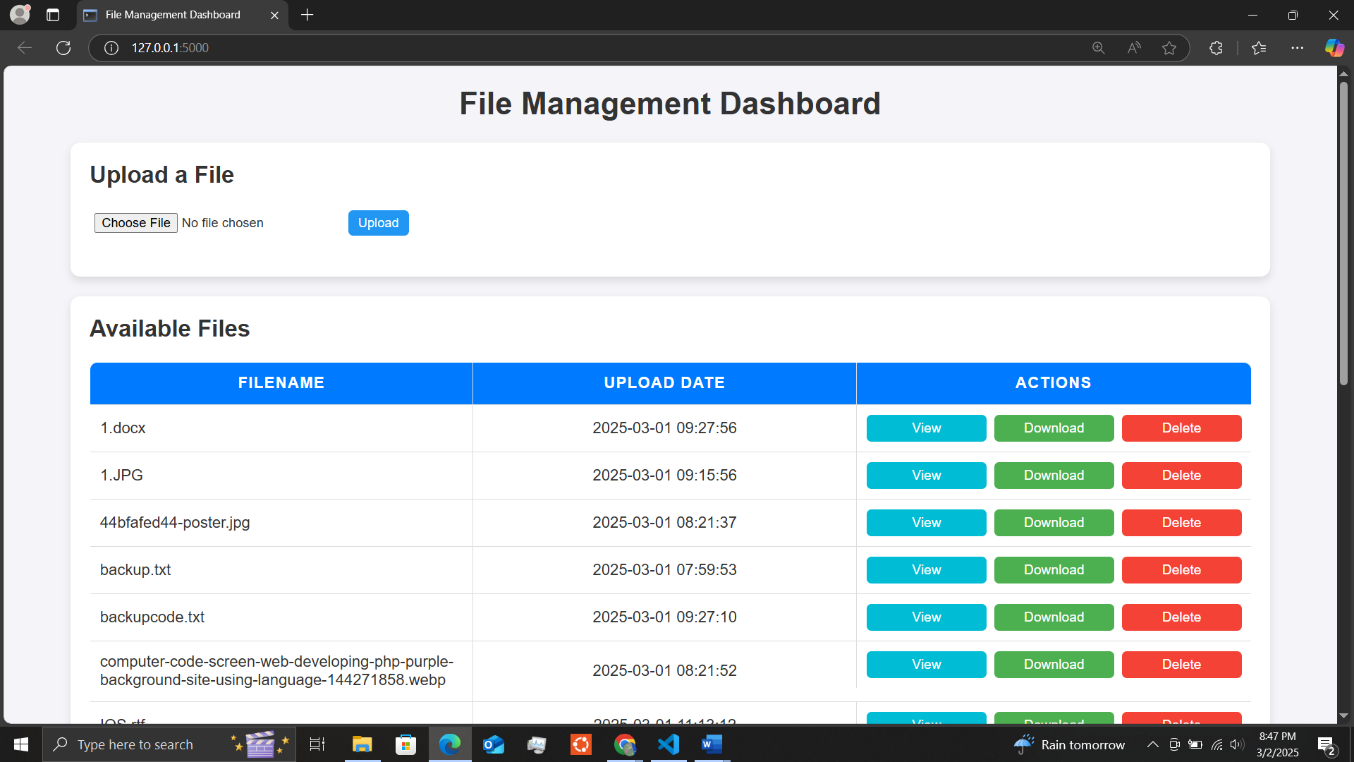
**5.2 Run the interface script**

python interface.py

**The interface will now be available at:**

[**http://localhost:5000**](http://localhost:5000)

****

****

**Step 6: Running the Client (myclient.py)**

The client connects to the server and performs file operations.

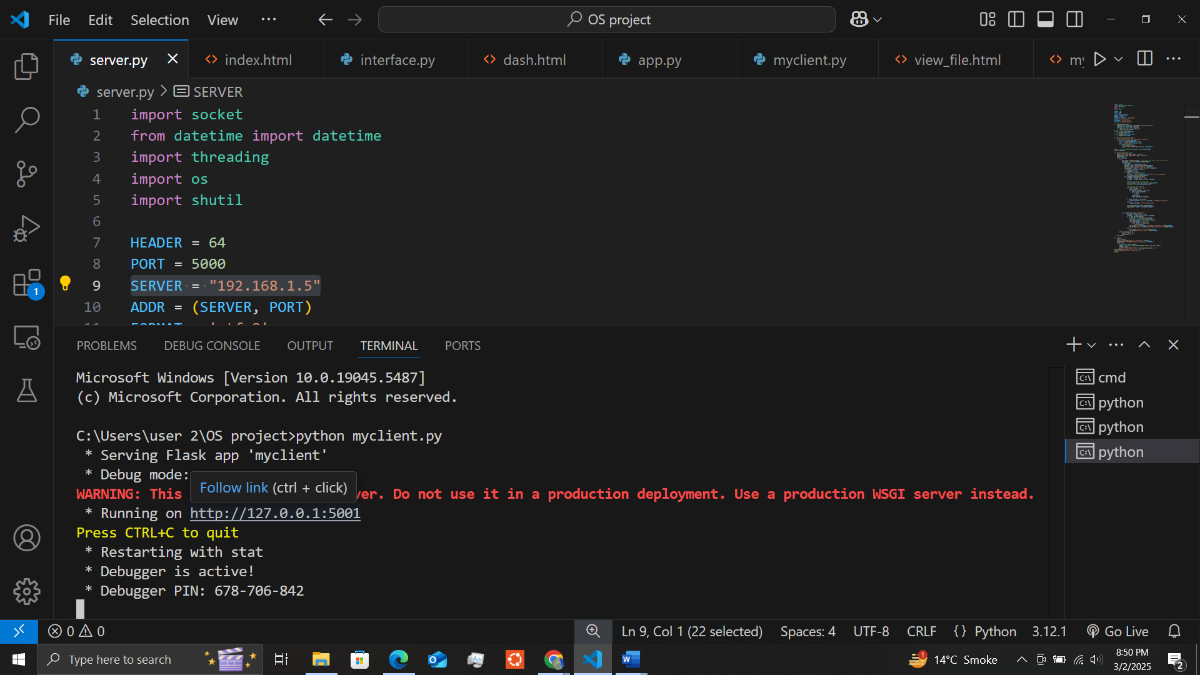
**6.1 Open a new terminal and navigate to the project folder**

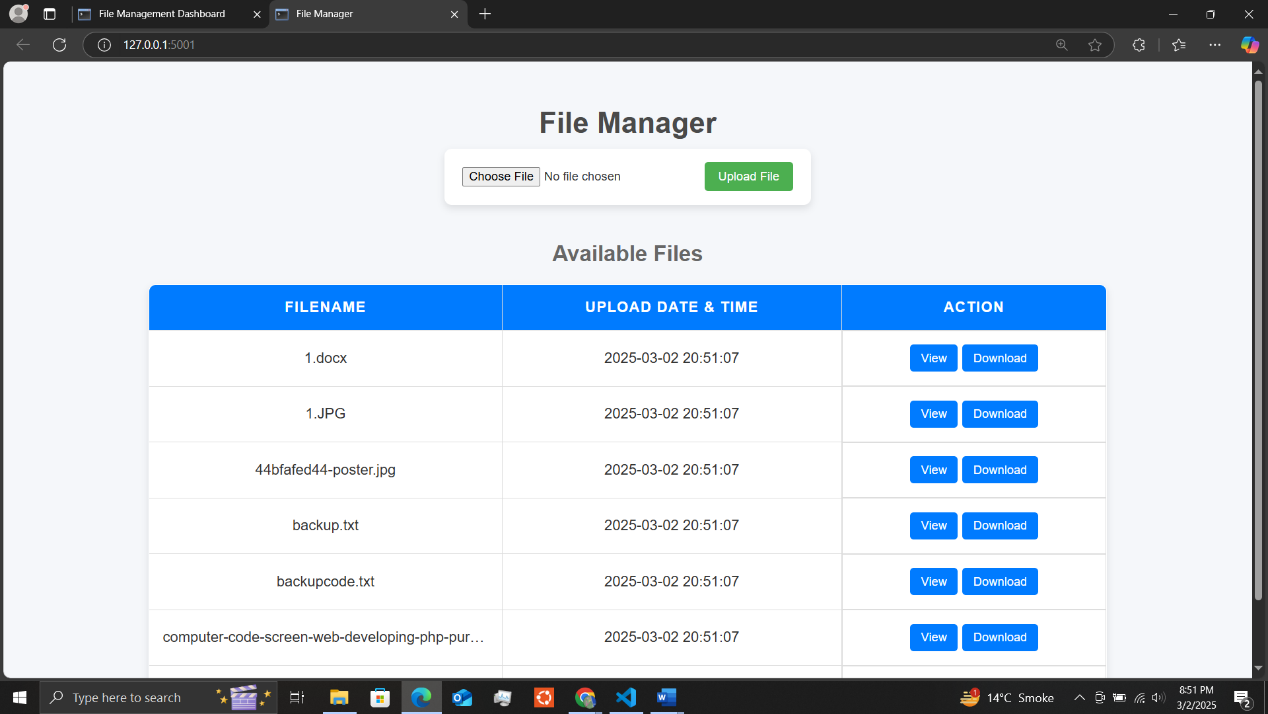
cd path\to\ Project

**5.2 Run the client script**

python myclient.py

The client will establish a connection with the server.



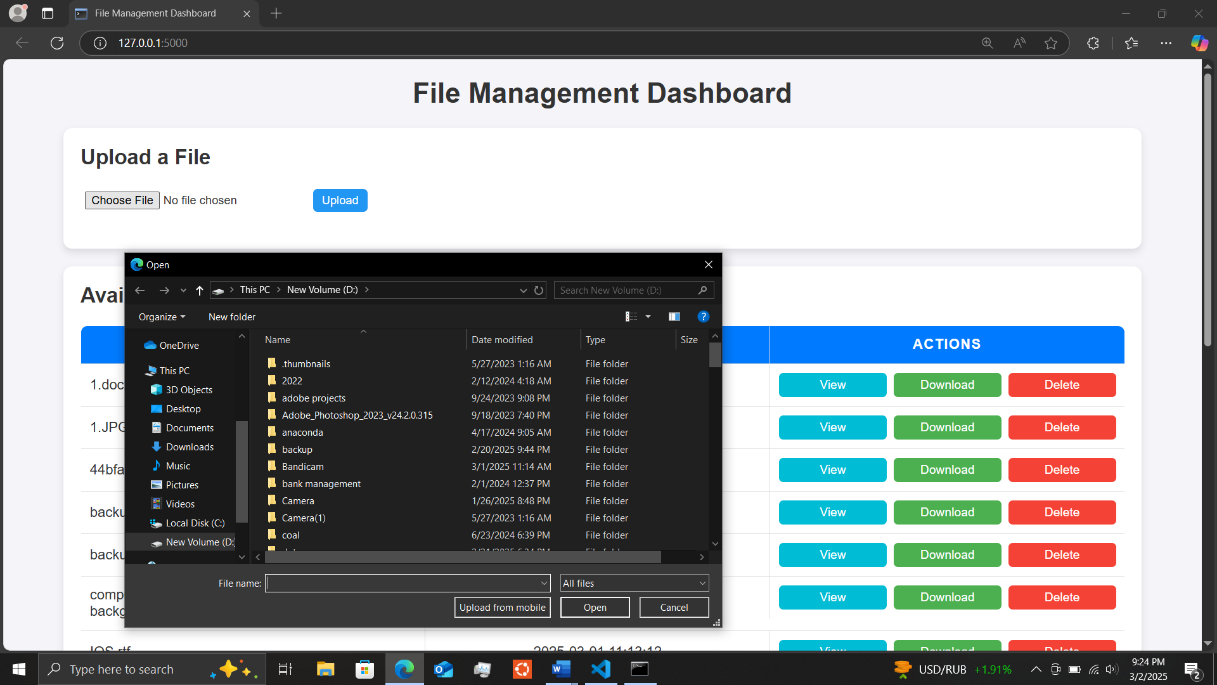


**Step 7: Using the System**

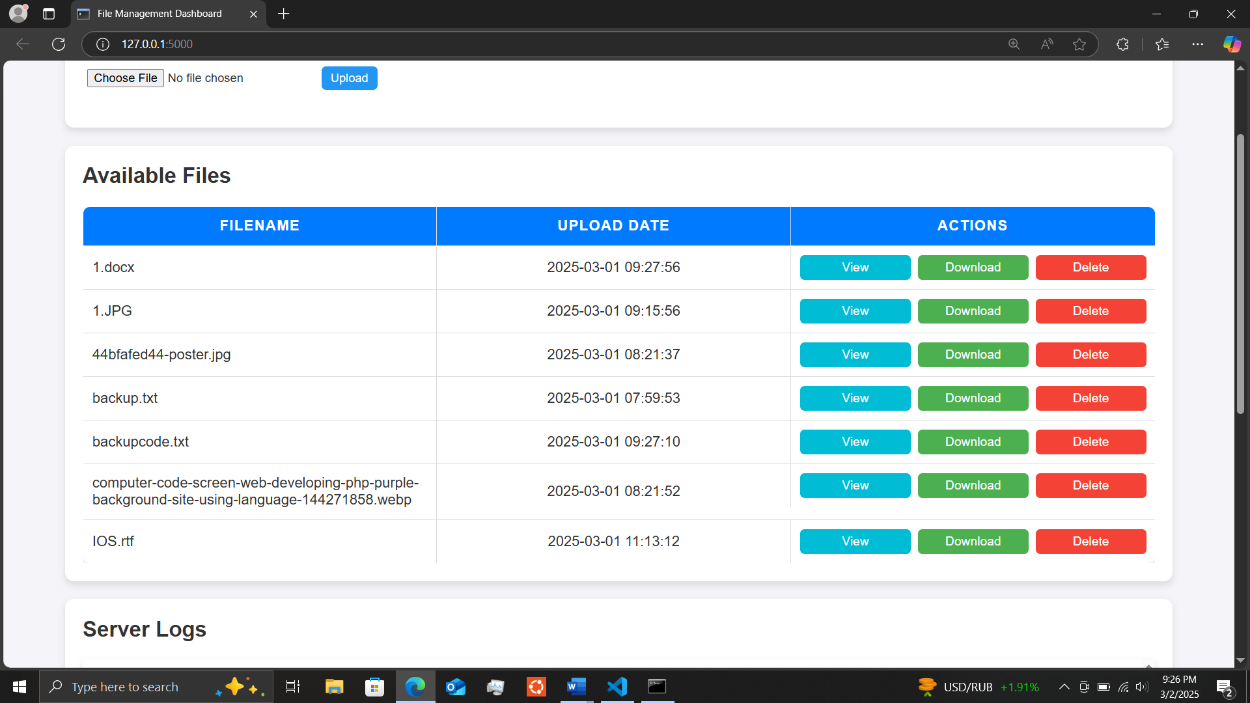
Once all three components (server, interface, and client) are running, you can:

**On Server:**

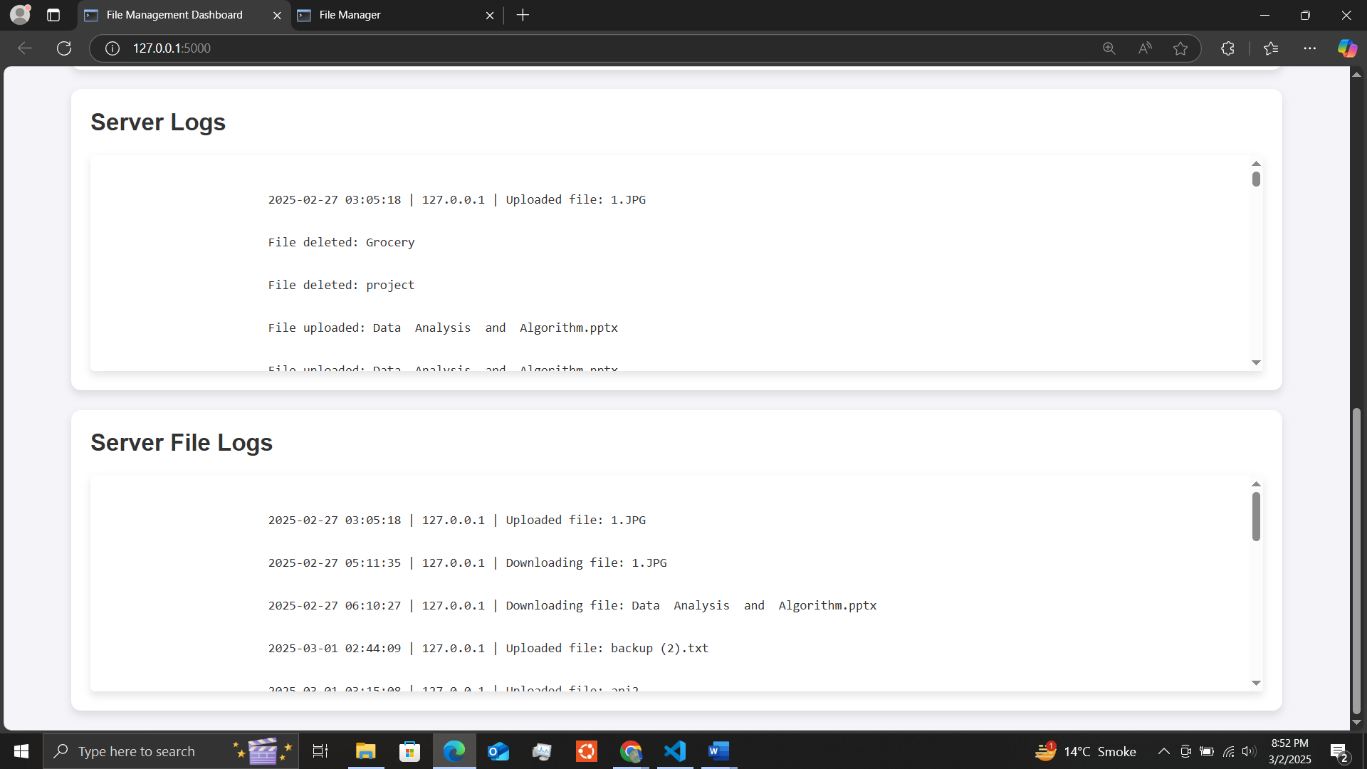
* Upload Files: Use the interface (http://localhost:5000) to upload files.



* Delete Files: Delete files from the available files.

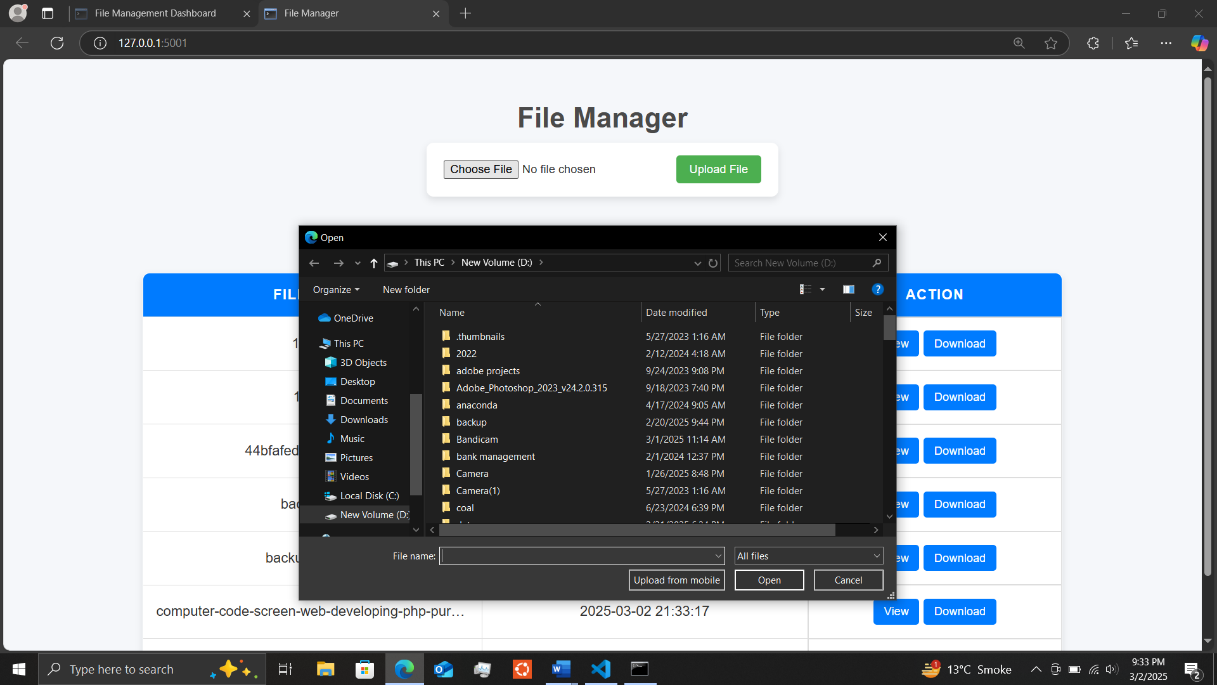


* Monitor Logs: Check server\_logs.txt for system activity.

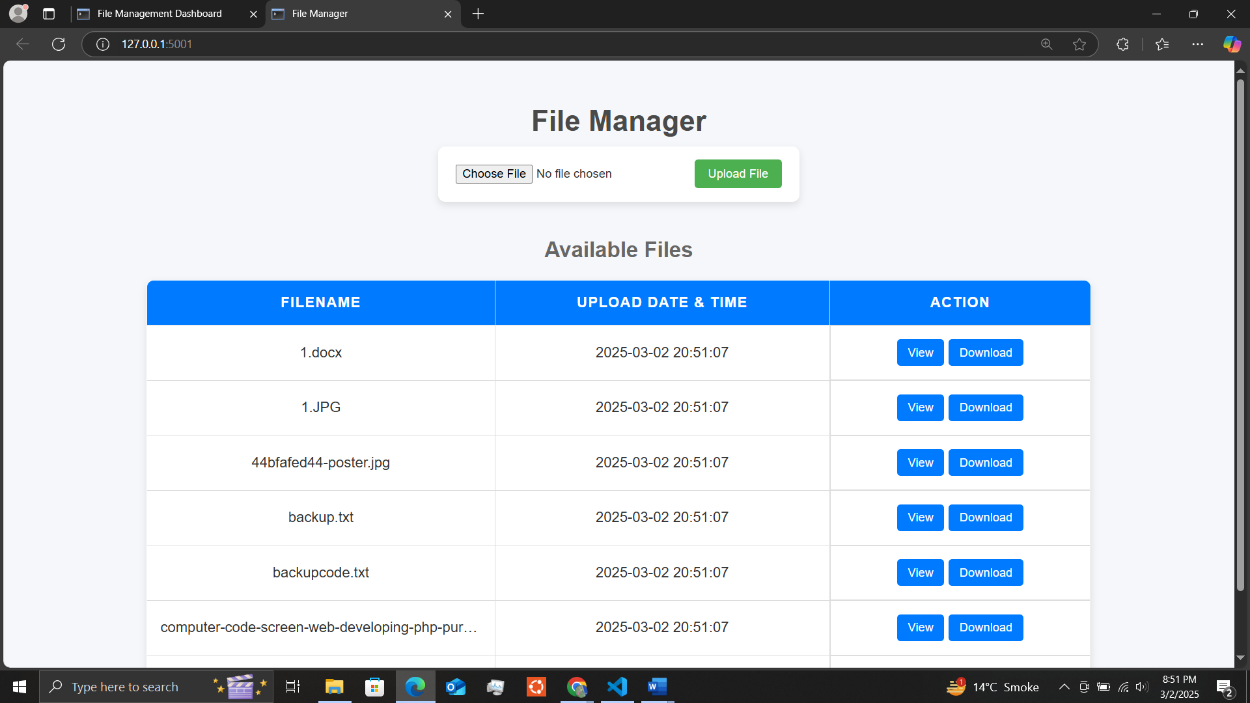


**On Client:**

* Upload Files: Use the interface to upload files.



* Download/View Files: Download and view the files from the available files.



**Step 8: Running All Components in Separate Terminals**

Each script should be running in a separate terminal window:

It can be either on one laptop or on two different laptops with one system acting as a client and the other as a server.

|  |  |  |
| --- | --- | --- |
| **Component** | **Command** | **Expected Output** |
| **Server** | **python server.py** | **Server is listening on 192.168.1.5** |
| **Interface** | **python interface.py** | **Access at** [**http://localhost:5000**](http://localhost:5000) |
| **Client** | **python myclient.py** | **Connected to the server** |