**Intel College Excellence Program   
Project Synopsis**

**“FPT Server”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member’s detail** | | | |
| **S. No.** | **Participant Name** | **Mobile No.** | **Email ID** |
| 1 | Chava Likhith | 9550240270 | ***chavalikhith70@gmail.com*** |
| **Faculty(college) mentor detail** | | | |
| **S. No.** | **Mentor Name** | **Mobile No.** | **Email ID** |
| 1 | Dr. Shelej Khera | 9466731632 | ***shelej.22390@lpu.co.in*** |
| **College/University Name** | | | |
| ***Lovely Professional University*** | | | |

**BACKGROUND**

FTP Server around the historical development of the File Transfer Protocol (FTP) and its purpose in enabling file transfers over a network. FTP was created during the nascent stages of network computing. Abhay Bhushan first presented it in 1971 as a component of the initial ARPANET protocols, which served as the forerunner to the internet. The objective was to develop a common protocol that would enable file transfers across a network between machines. FTP's early iterations were made to move files between computers connected to a network by using straightforward commands for uploading, downloading, renaming, deleting, and listing directory contents. RFC 765 established the protocol's standards in 1980, and RFC 959, which was later revised in 1985, is still the fundamental FTP specification to this day. The Fundamentals of FTP has a client-server architecture. The computer that houses the files is known as the FTP server, and the program that connects to the server and transfers files is known as the FTP client. By doing this, a user (client) can upload, download, or edit files that are kept on the distant server. FTP communicates via two distinct connections:

1.Control Connection (Port 21): Used to send commands (such as file requests or login credentials).

2.File transfers between the client and the server are accomplished via the data connection (Port 20 or other random ports).

Initial FTP has Security Issue Because early FTP implementations sent data in plain text, including usernames and passwords, they were vulnerable to being intercepted by hostile actors. A serious security risk resulted from this lack of encryption. More secure FTP versions were created in order to allay these security worries like SSH File Transfer Protocol, or SFTP, is a secure FTP method that encrypts data and commands and Secure file transfers via the FTP protocol are made possible by FTPS (FTP Secure), which is FTP plus SSL/TLS encryption. In order to guarantee safe, encrypted communication over public networks, businesses nowadays frequently favor SFTP or FTPS.

**PROBLEM IDENTIFICATION**

Despite being widely used for file transfers, FTP (File Transfer Protocol) has a number of documented issues and limitations, particularly in contemporary settings. By default, FTP transmits all information in plain text, including passwords and usernames. Because of this, it is extremely susceptible to man-in-the-middle (MITM) and eavesdropping attacks, which allow hackers to obtain confidential files or login passwords. Absence of data encryption Because file contents are not encrypted while being transmitted, if they are intercepted, sensitive information may be accessible to unauthorized parties.

Without extra security features like multi-factor authentication (MFA), basic FTP uses weak authentication techniques (password and username). In order to obtain unauthorized access, attackers may try a number of username/password combinations on FTP servers that are not setup correctly. The client and server establish a data connection, which firewalls may prevent, leading to problems with the connection. Firewall setups and port ranges can make this process more difficult, but the client starts the connection to a port that the server specifies. FTP needs a number of ports for data transfer (port 20 or a range of ports in passive mode) and control (port 21). It can be difficult to manage these ports and set up firewalls for FTP traffic. Users can log in without credentials to certain FTP servers that permit anonymous access. Anonymous FTP access can be a serious security issue if it is not appropriately regulated, enabling anyone to view or upload data without the required authorization. FTP does not automatically check the integrity of files that are sent. FTP does not have checksum verification or other ways to identify when a file is corrupted during transfer (for instance, because of network problems). Modern encryption methods like SSL (Secure Sockets Layer) and TLS (Transport Layer Security), which are now commonplace for safe online interactions, are not supported by the basic version of FTP. Other protocols, such as SFTP (SSH File Transfer Protocol) or FTPS (FTP Secure), must be utilized in order to secure FTP. FTP is inefficient for large file transfers when compared to more recent options like HTTP or cloud-based file-sharing services because it may function slowly over high-latency networks (such overseas connections or mobile networks).

**PROPOSED SOLUTION**

A number of fixes and best practices can be used to improve security, performance, and usability in order to solve the different issues related to FTP servers.

**Make Use of SFTP and FTP Secure:** Make use of SFTP (SSH File Transfer Protocol) to protect login credentials and file data by encrypting both the control and data connections. FTPS secures the communication channels by encrypting FTP using SSL/TLS. Additionally, it has the capability of client and server authentication. Use encryption to stop the transmission of plain text, encrypt all FTP connections.

**Turn Off Anonymous FTP Access:** Make sure that only authenticated users have access, and always turn off anonymous logins unless absolutely required. all user accounts have strong password requirements, particularly those pertaining to password complexity and expiration dates. Limit the number of log-in attempts to lessen the impact of brute-force attacks. increase security by requiring a second authentication factor to be used in order to get access, even in the event that a password is hacked.

**Implement Passive Mode Properly:** Set up the FTP server to operate in passive mode and specify a range of ports for data connections. Verify that routers and firewalls are set up to permit these ports. SFTP is firewall-friendly and eliminates the hassle of managing several ports by using a single connection via port 22, which is also used by SSH.

**Anonymous Access:** Turn off anonymous access and restrict access to the FTP server to authorized users only. Apply stringent read-only permissions and restrict access to particular directories if anonymous access is required.

**Use Checksum Verification:** Create checksums of files both before and after transfer by employing hash functions (such as MD5, SHA256, etc.) to provide external integrity checks. By doing this, users may confirm that the transferred file is identical to the original.

**Disable Plain FTP:** Make sure that all transfers are made using FTPS or SFTP and specifically turn off the server's support for plain FTP (unencrypted connections).

**Setup Essentials**

**Hardware Requirements:**

1. Server Machine.
2. Network Equipment.
3. Power Backup (**Uninterruptible Power Supply**).
4. Redundancy and Backups (**Redundant Array of Independent Disks**).

**Software Requirements:**

1. Operating System.
2. FTP Server Software.
3. Security Software.
4. User Management and Authentication.
5. Logging and Monitoring.
6. Port and Firewall Configuration.

**DESCRIPTION**

**FTP Server**

An FTP server is a software application that runs on a physical or virtual machine, allowing users to transfer files between the server and client devices over a network (local area network or internet). It uses the FTP protocol, which follows a client-server architecture. Clients connect to the FTP server using FTP client software, enabling them to download or upload files, manage directories, or perform file operations.

**Table of Contents**

1. Step 1: Installing Operating System (Ubuntu 20.04).

2. Step 2: System Update.

3. Step 3: Installing Packages Like VSFTPD.

4. Step 4: Configure the Packages.

5. Step 5: Enable and disable Few options in FTP Server.

6. Step 6: Restart Service.

7. Step 7: Allow FTP Traffic Through the Firewall.

8. Step 8: Create the FTP user and Password.

9. Step 9: Test the FTP server.

10. Step 10: Install FileZilla.

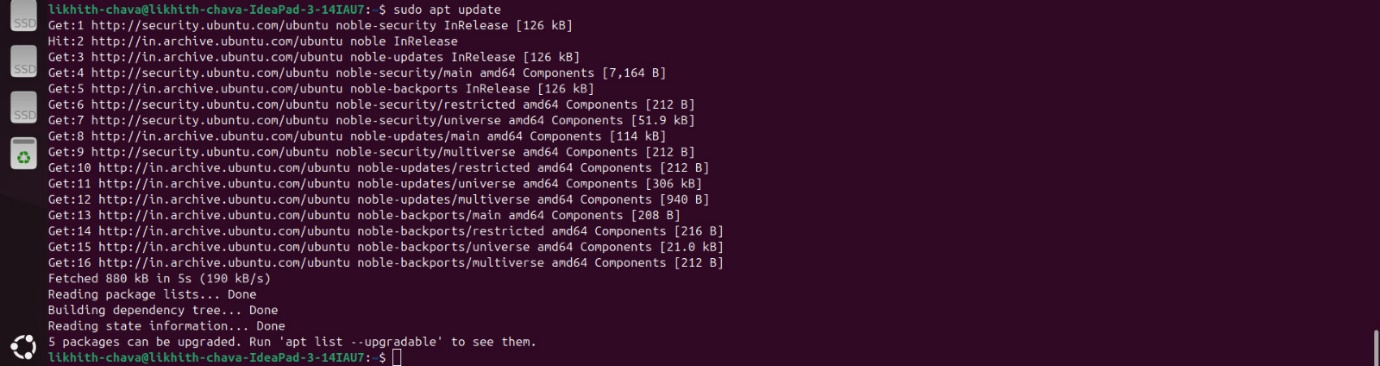
11. Step 11: Add I.P address, Host and Password.

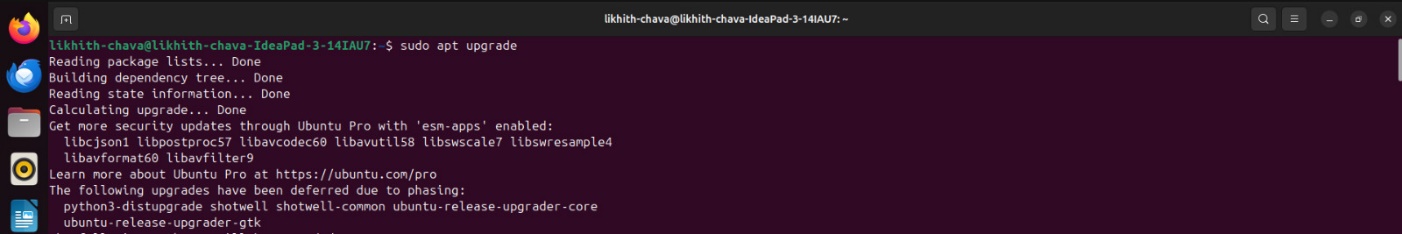
12. Conclusion.

**Step 1: System Update:**

Make sure your system is up to date before installing the VSFTPD.

* Sudo apt update.
* Sudo apt upgrade.

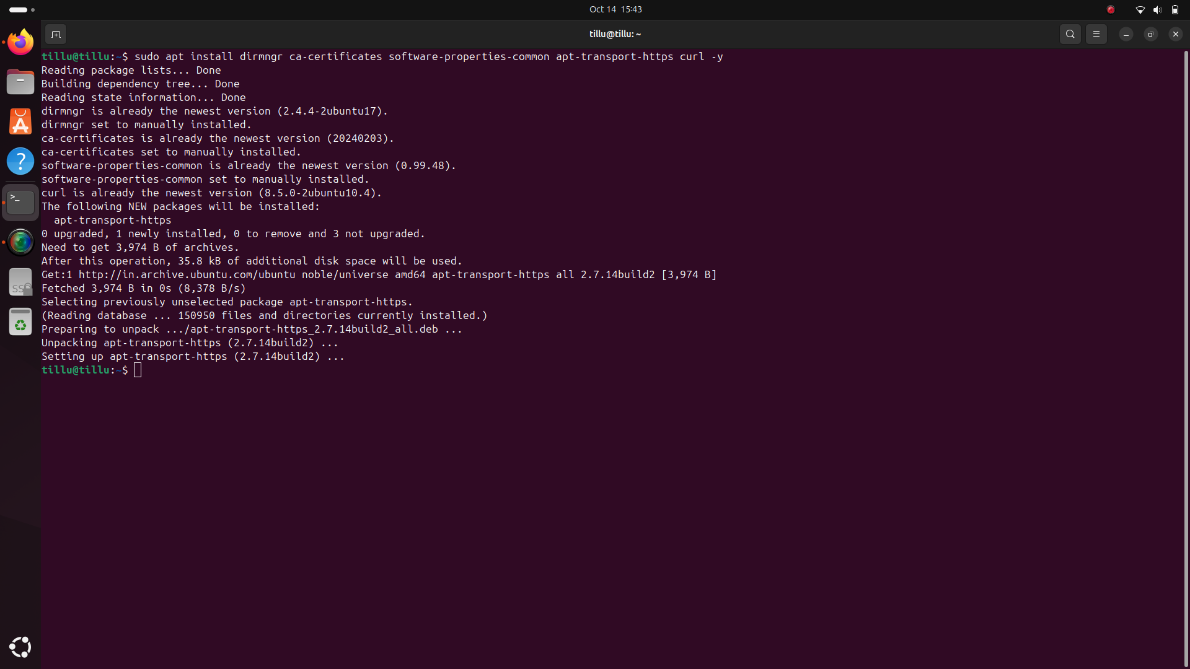


****

**Step 2: Install Required Packages:**

Install the dependencies that Plex requires:

> sudo apt install dirmngr ca-certificates software-properties-common apt-transport-https curl -y

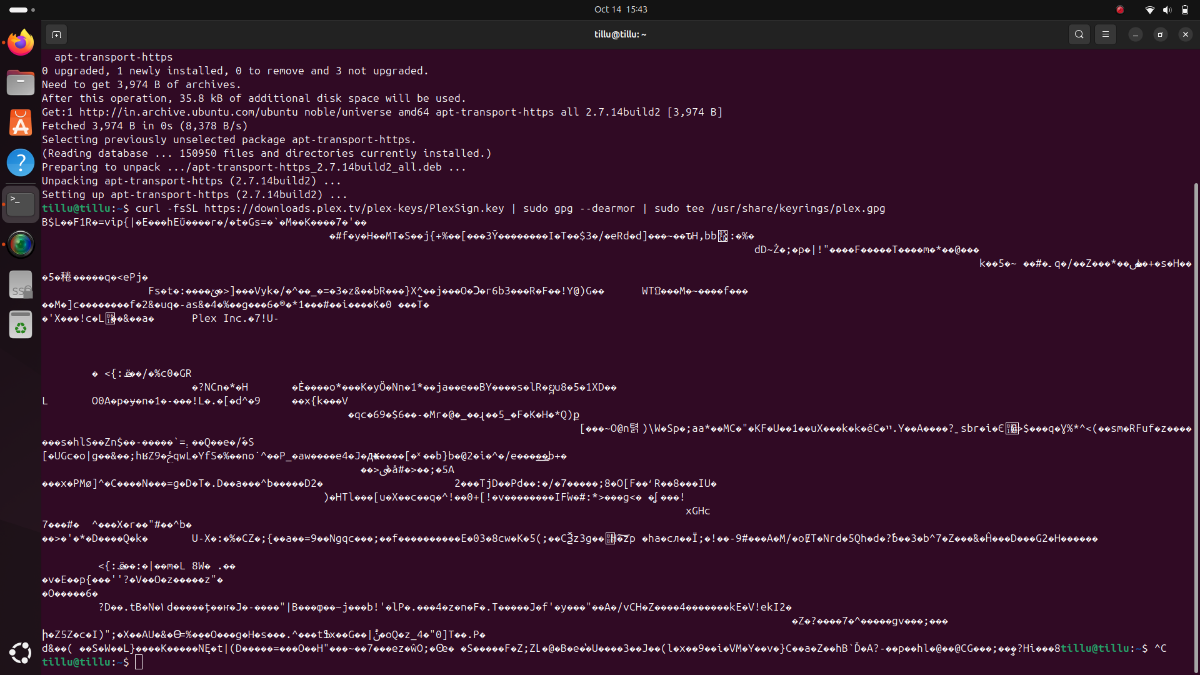
****

**Step 3: Add Plex Repository and Key:**

Add the GPG Key

To ensure that the packages from Plex are trusted, download the GPG key and add it to your keyring:

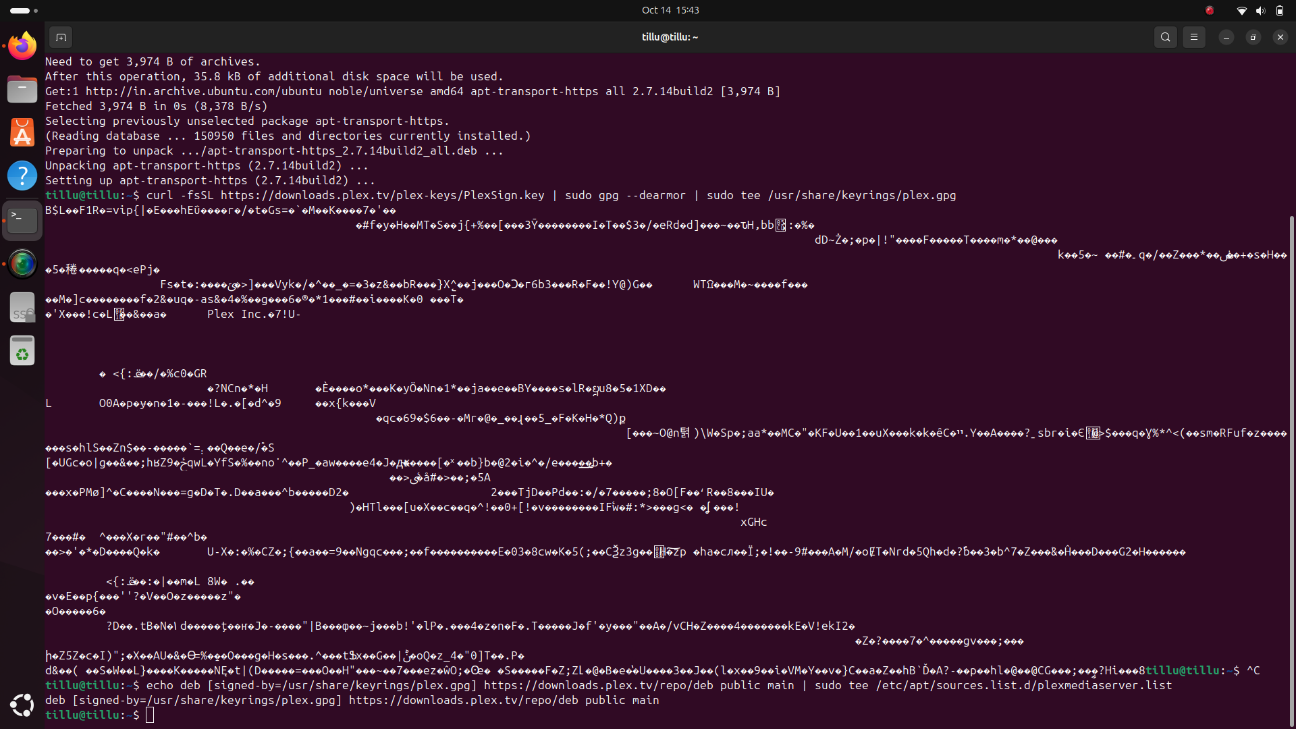
> curl -fsSL https://downloads.plex.tv/plex-keys/PlexSign.key | sudo gpg --dearmor | sudo tee /usr/share/keyrings/plex.gpg

****

Add the Plex Repository

Add the Plex repository to your APT sources list:

> echo deb [signed-by=/usr/share/keyrings/plex.gpg] https://downloads.plex.tv/repo/deb public main | sudo tee /etc/apt/sources.list.d/plexmediaserver.list

****

Update the Package List

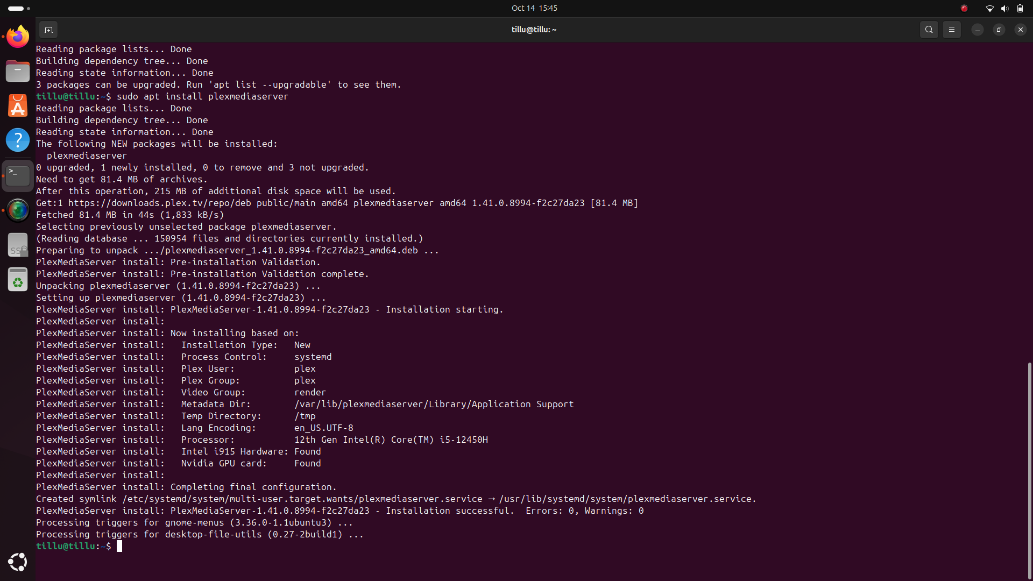
Update your package list to include the new Plex repository:

> sudo apt update

**Step 4: Install Plex Media Server**

Now that the repository is added, you can install Plex Media Server using the APT package manager:

> sudo apt install plexmediaserver

****

**Step 5: Start and Enable Plex Service**

After the installation is complete, start the Plex Media Server service and enable it to run on system boot:

Start the Plex Service:

> sudo systemctl start plexmediaserver

Enable Plex to Start on Boot:

> sudo systemctl enable plexmediaserver

**Step 6: Set Up Media Directory**

Create a directory in your home folder to store your media files for Plex:

> mkdir /home/tillu/plexmedia

**Step 7: Configure Permissions**

Set the appropriate ownership and permissions for the media directory:

Change Ownership:

> sudo chown -R plex:plex /home/tillu/plexmedia

Set Permissions:

> sudo chmod 755 /home/tillu/plexmedia

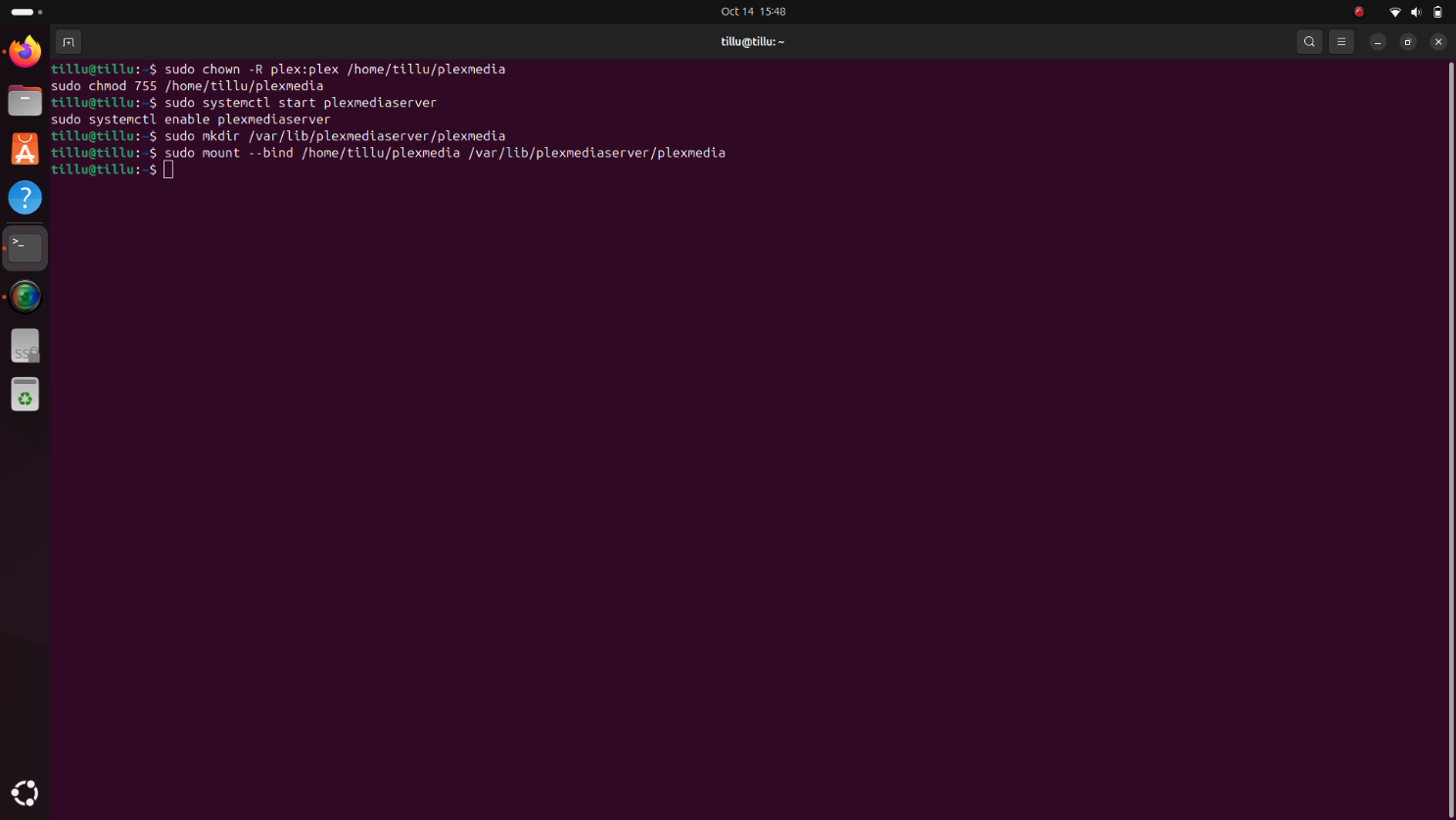
**Step 8: Mount Media Folder to Plex Directory**

To allow Plex to access your media files, mount the media folder into the Plex directory:

Create the Plex Media Directory:

> sudo mkdir /var/lib/plexmediaserver/plexmedia

Mount the Media Folder:

****> sudo mount --bind /home/tillu/plexmedia /var/lib/plexmediaserver/plexmedia

**Step 9: Make Bind Permanent**

To ensure that the bind mount persists after a reboot, edit the fstab file:

Edit the fstab File:

> sudo nano /etc/fstab

Add the Following Line:

> /home/tillu/plexmedia /var/lib/plexmediaserver/plexmedia none bind 0 0

**Step 10: Adjust Permissions for Media Access**

To ensure both the Plex service and your user can manage the media files:

Change Ownership Back to the User:

> sudo chown -R tillu:tillu /home/tillu/plexmedia

Add the User to the Plex Group:

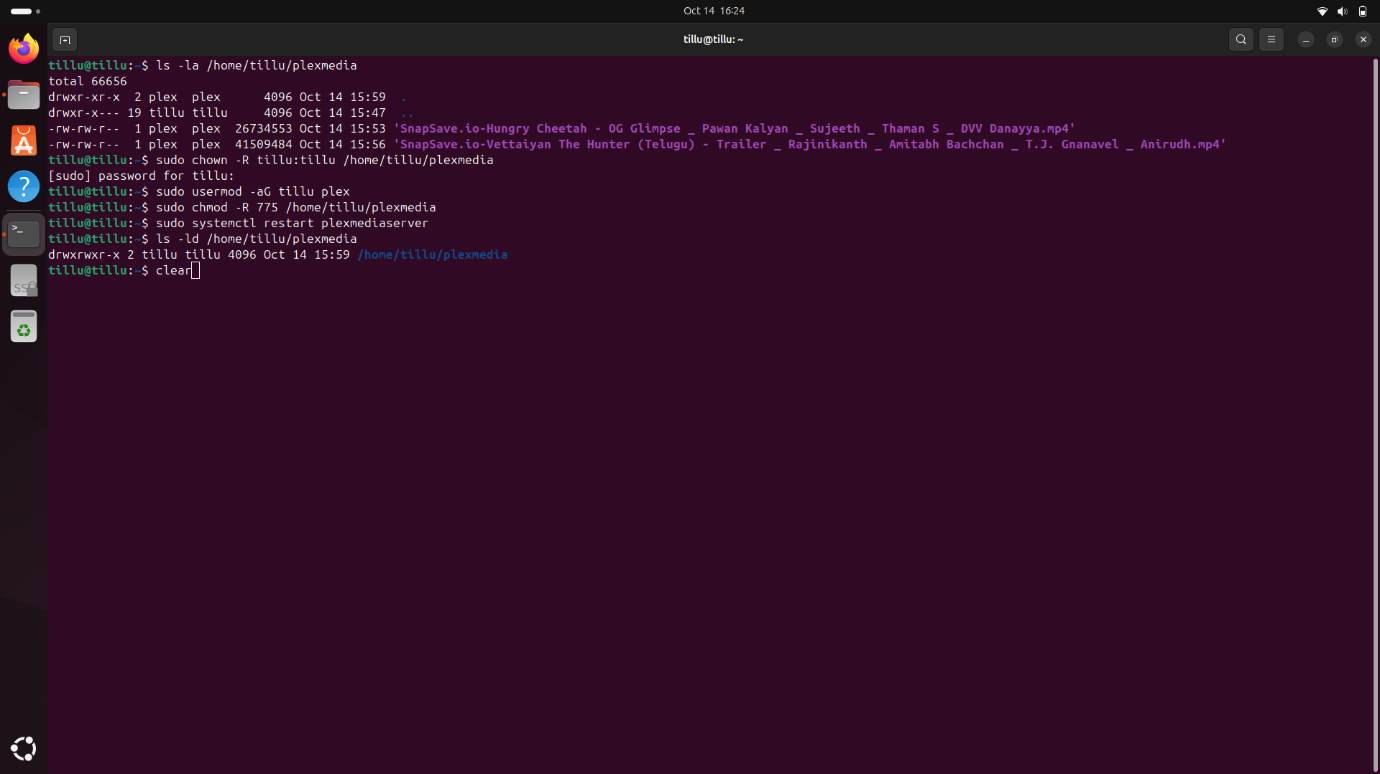
> sudo usermod -aG tillu plex

Set Directory Permissions:

> sudo chmod -R 775 /home/tillu/plexmedia

**Step 11: Finalize and Restart Plex Service**

After configuring the permissions, restart the Plex Media Server:

****> sudo systemctl restart plexmediaserver

Verify Folder Permissions:

> ls -ld /home/tillu/plexmedia

**The output should be: drwxrwxr-x 2 tillu tillu 4096 Oct 14 10:15 /home/tillu/plexmedia**

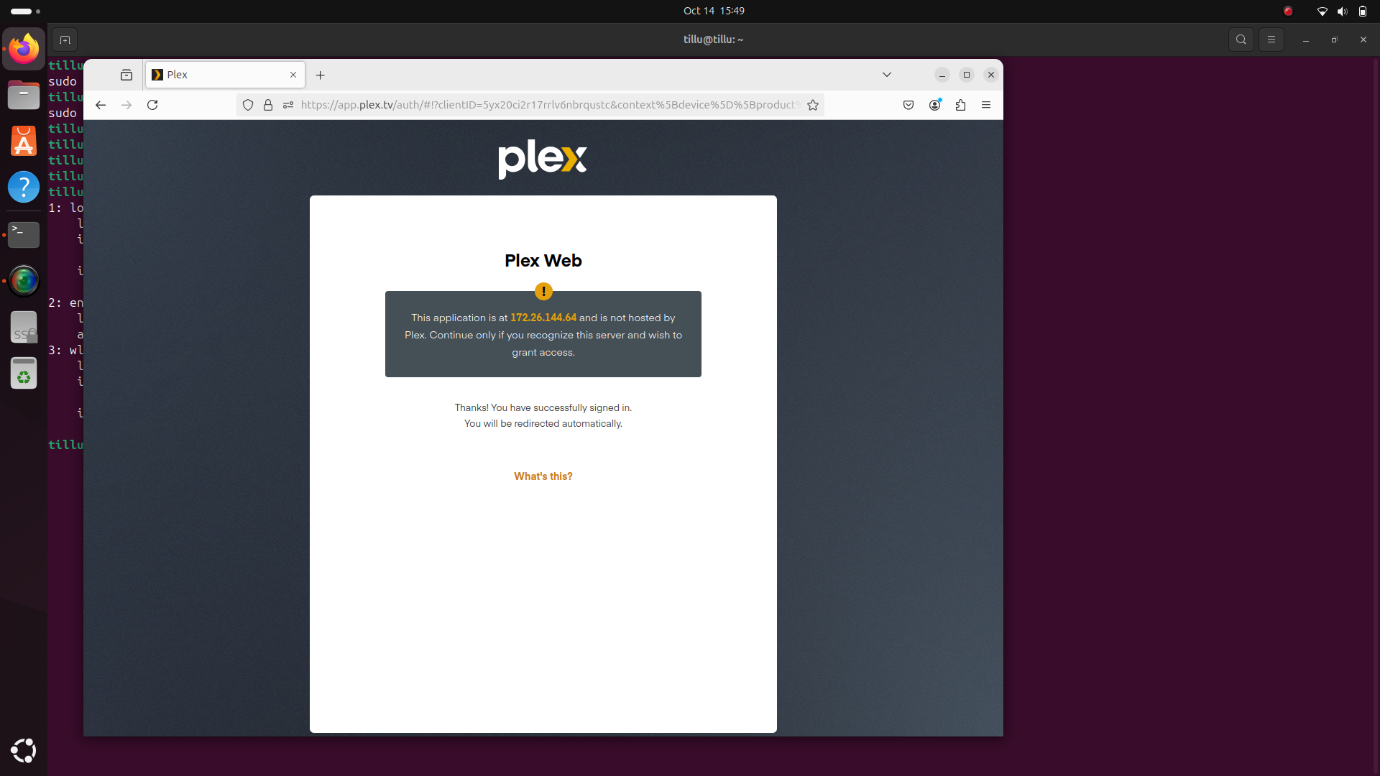
**Step 12: Access Plex Web Interface**

Once Plex Media Server is installed and running, access the Plex Web interface by navigating to:

> http://<your-server-ip>:32400/web

Replace `<your-server-ip>` with the IP address of the machine where Plex Media Server is installed (e.g.`http://172.26.144.64:32400/web`)

**Login Screen Example:**

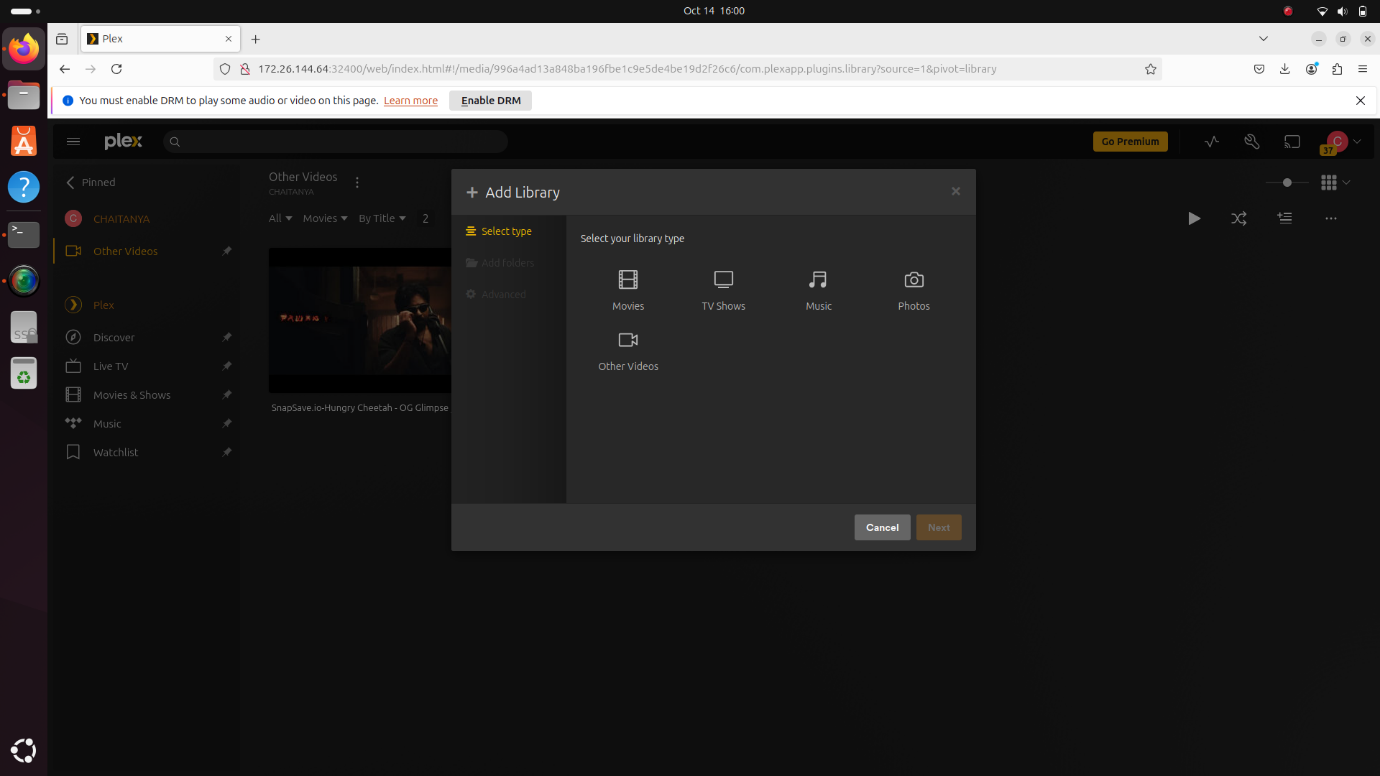
****

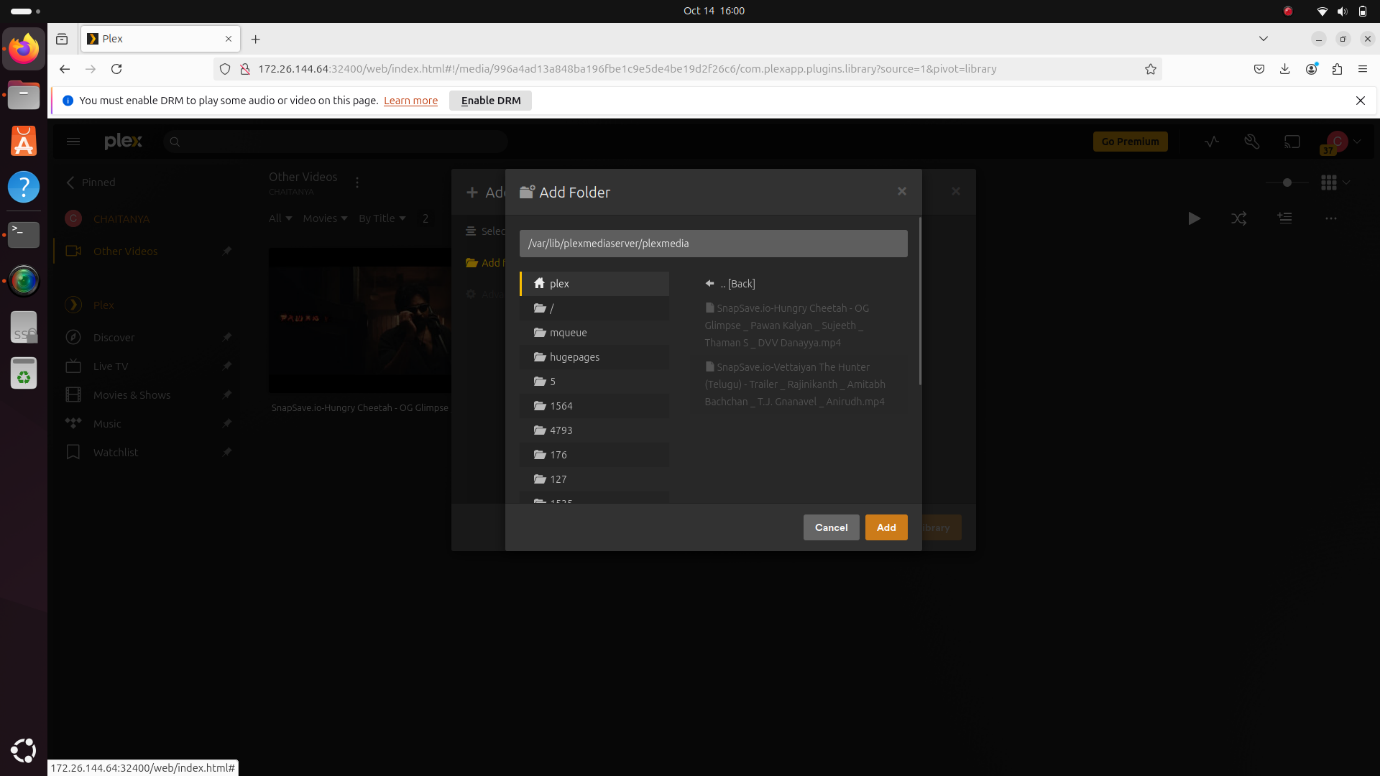
**Step 13: Plex Dashboard and Media Library Setup**

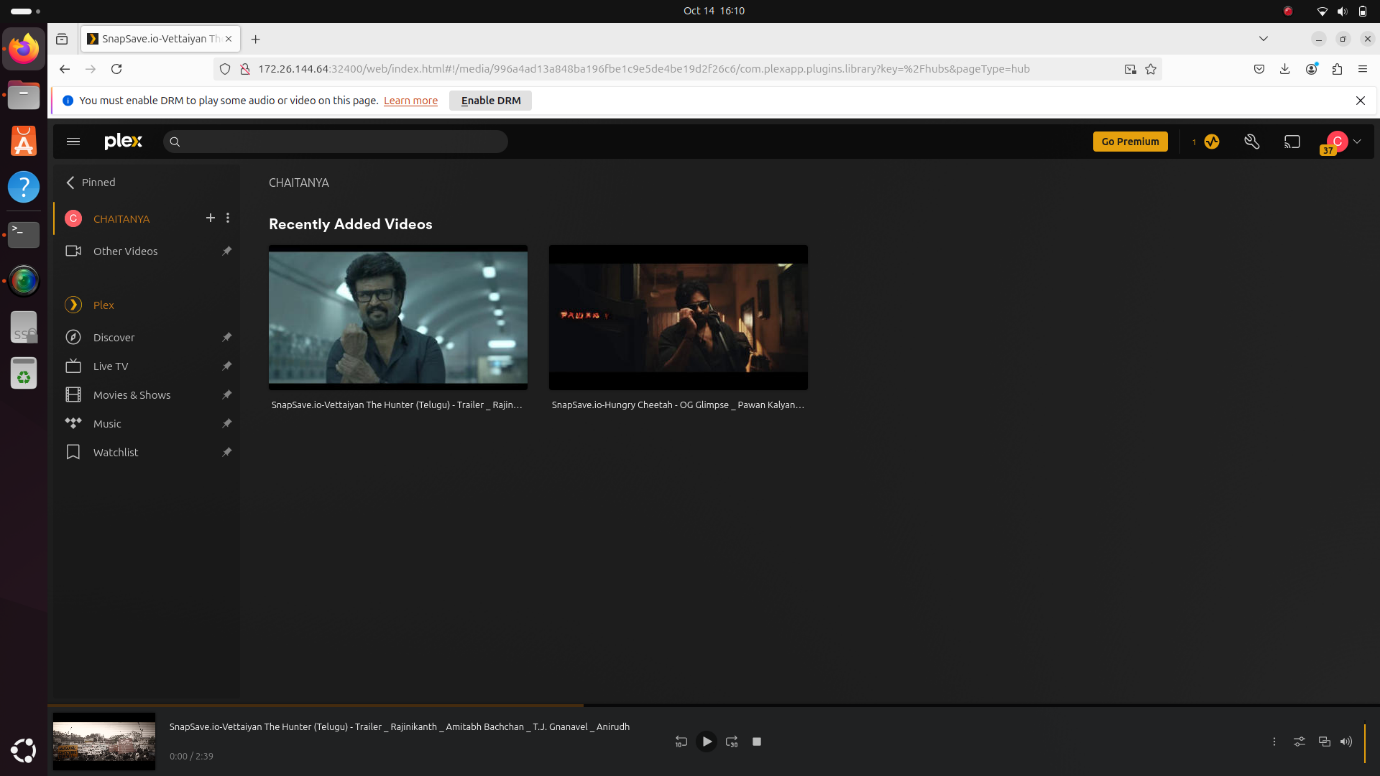
After logging in, configure your media libraries:

1. Click on “Add Library.”

2. Select the type of media (e.g., Movies, Music, TV Shows).

****

****3. Browse for the media directory (`/home/tillu/plexmedia`).

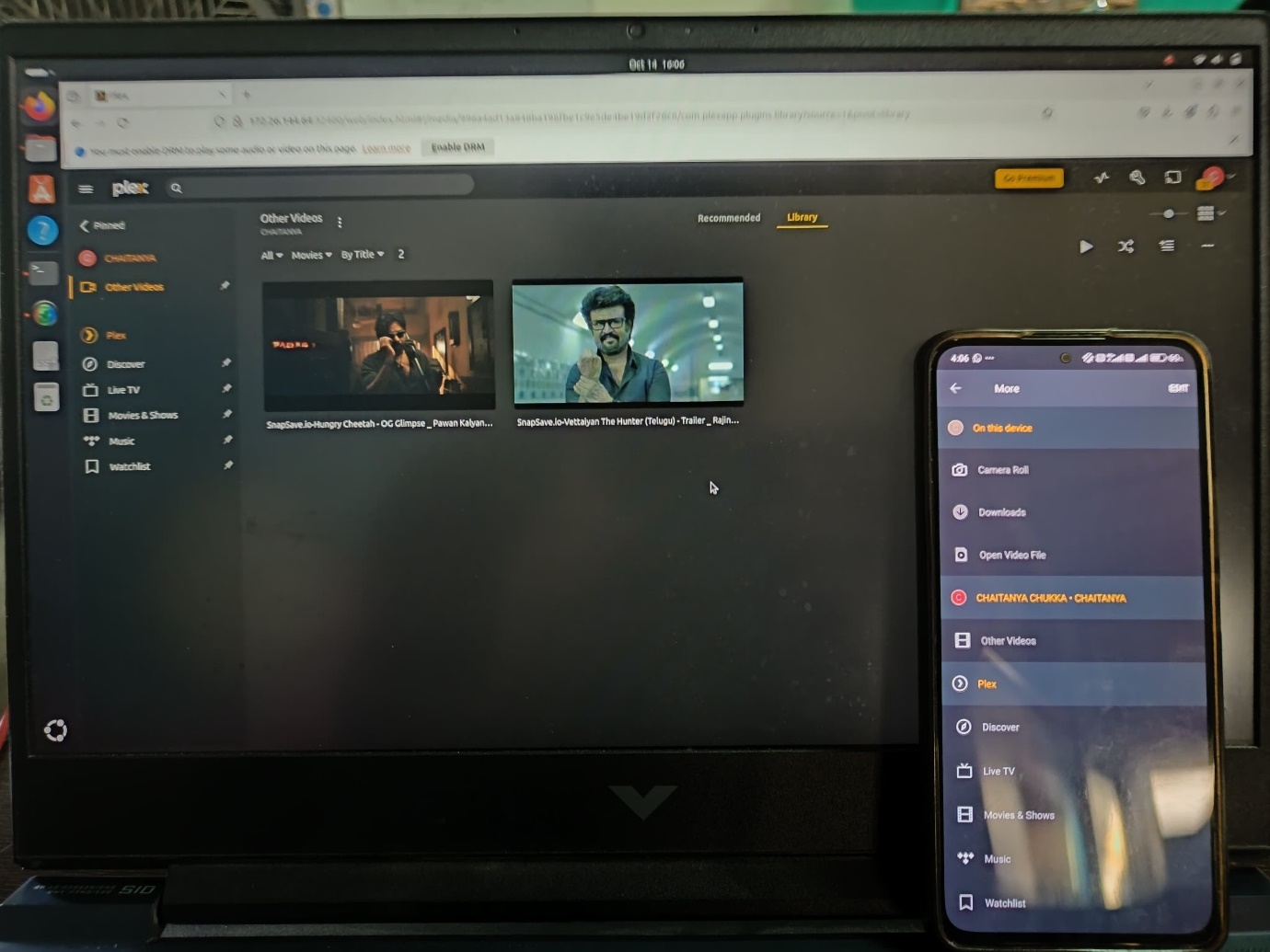
****4. Click “Add Library” to start scanning.

Plex will begin indexing your media files and fetching metadata.

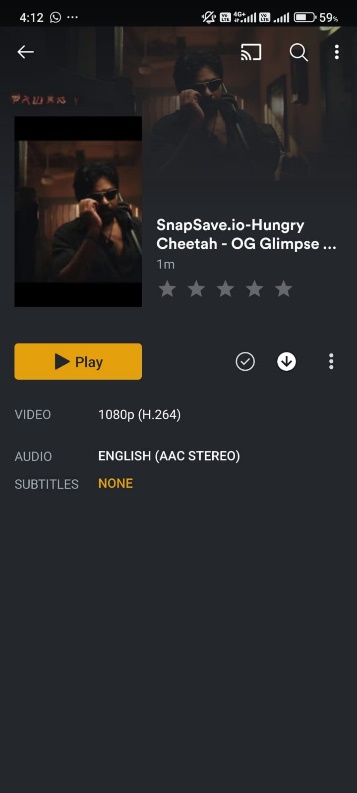
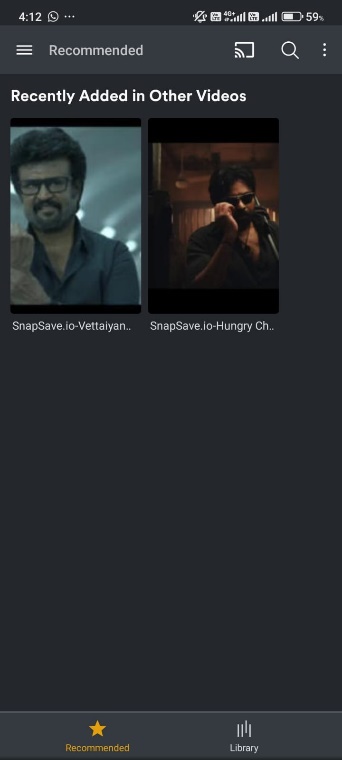
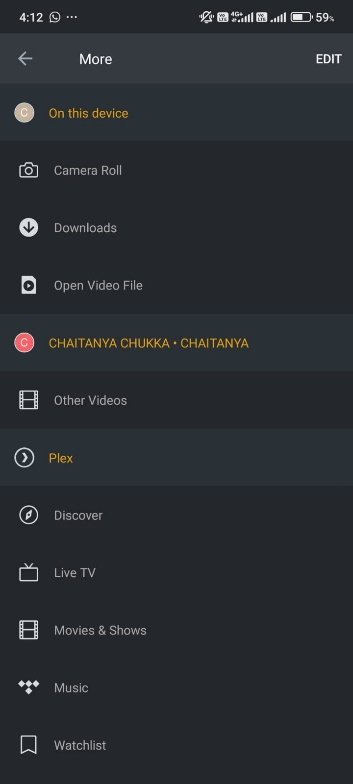
**Step 14: Share Media with Friends via Plex**

To share your media libraries with friends, follow these steps:

1. Go to your Plex Web Interface.

****2. Select a media file and click on “Watch Together.”

3. Choose a friend from the list and click “Invite.”

4. Your friend can now watch the content on their Plex account.

**FUTURE SCOPE**

In the future, Plex Media Server will evolve with advancements in artificial intelligence, cloud integration, and improved connectivity. AI will enable smarter content recommendations, personalized media experiences, and automated metadata management. Cloud solutions will offer hybrid storage, combining local and remote resources for seamless media access. As internet speeds and 5G technology improve, Plex will support effortless 4K and 8K streaming, with AI-driven transcoding optimizing playback across different devices. Enhanced security features, possibly using blockchain, will ensure secure media access and data privacy. The integration of voice assistants will make server control more intuitive, while sustainability features like energy-saving modes will reduce server power consumption. Overall, Plex will become more intelligent, efficient, and user-friendly, adapting to the growing demand for flexible and secure media streaming.

**CONCLUSION**

In conclusion, setting up Plex Media Server offers a powerful and flexible solution for managing and streaming media libraries across multiple devices. By organizing content like movies, TV shows, and music, Plex ensures an enhanced user experience with automated metadata fetching, media categorization, and seamless access from anywhere. Through proper configuration, users can leverage the platform’s ability to stream high-quality media remotely, with built-in transcoding that optimizes playback based on network conditions and device capabilities. With additional features such as user profile management, parental controls, and secure remote access, Plex offers a personalized experience for each user while maintaining strict control over content access. The ability to scale storage through NAS devices or cloud integration ensures that users can expand their libraries without compromising performance. Moreover, Plex Media Server supports multiple platforms, including smartphones, smart TVs, and gaming consoles, ensuring compatibility with various devices. By setting up permission structures and maintaining server performance through regular updates, Plex delivers a robust solution for media enthusiasts. As streaming and media consumption continue to evolve, Plex stands out as a versatile and user-friendly tool, capable of adapting to new technologies and meeting the needs of modern media consumers. With its ever-expanding features and support for advanced media formats, Plex remains a leading choice for those seeking an all-in-one media server platform that balances ease of use, performance, and security.

**REFERENCES**

Github link:

<https://github.com/Chukka003/intel_fice_5thsem>