🎯 Roadmap for Privacy-First P2P File Sharing Network

Phase 1: Build the Core P2P Network (10 Days)

Goal: Create a decentralized, trackerless P2P network.

- **Tasks:**
  - Set up node communication using Distributed Hash Table (DHT).
  - Implement basic file sharing between peers.
  - Ensure node discovery works without a central server.

Tech Stack:

- **Language:** Python / C++
- **Libraries:** Kademlia DHT (Python or C++ version)
- **Networking:** Sockets (TCP/UDP)
- **Testing:** Wireshark for packet analysis

Phase 2: Implement Chunk-Level Encryption (10 Days)

Goal: Secure file transfers with data split into encrypted chunks.

- **Tasks:**
  - Split files into fixed-size chunks.
  - Encrypt each chunk individually using modern encryption algorithms.
  - Ensure only authorized peers can reassemble the file.

Tech Stack:

- **Encryption Library:** liboqs (Open Quantum Safe library — or other strong, efficient encryption)
- **Hashing:** SHA-256 / Blake3 for file integrity
- **Data Handling:** Python/C++ file handling and byte manipulation

---

Phase 3: Metadata Obfuscation & Privacy Boost (10 Days)

Goal: Hide file metadata and disguise traffic patterns.

- **Tasks:**
  - Obfuscate metadata within the DHT (e.g., fake keys or mixed content tags).

    - Implement basic traffic masking (e.g., padding or random noise injection).
    - Ensure performance remains stable.

Tech Stack:

- **Data Obfuscation:** Custom metadata handling in Python/C++
- **Traffic Analysis:** Wireshark (for testing how hidden your traffic looks)
- **Optional UI:** Flask/Django (if you want a basic front-end for peers to connect)

🚀 Final Touches & Presentation (5 Days)

- **Documentation:** Clean up code, comment thoroughly, and prepare a technical report.
- **Performance Testing:** Test on different networks (LAN, VPN, throttled connections).
- **Presentation:** Create a visual walkthrough of the system (Diagrams + Demo).
