JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY  
NOIDA

ODD SEMESTER (2025-2026)  
BRANCH-IT  
DATA STRUCTURE LAB

# PROJECT TITLE:

ALLOW CLIENTS TO DIRECTLY SHARE FILES WITHOUT A CENTRAL SERVER

BATCH: F17

PRESENTED TO: Himika Verma

PRESENTED BY:

Dhruv Arora   
992401040023

Ansh Mahajan   
9924010030

Shourya Singh  
992401040036

Aaryan Didwania  
992401040034

# BRIEF EXPLANATION

This project implements a peer-to-peer (P2P) file sharing system that allows clients to exchange files directly with each other without relying on a persistent central server to store or forward file data. The system supports direct browser-to-browser transfers (WebRTC DataChannels) and a content-addressed network option (IPFS/libp2p) for wider distribution. It demonstrates how discovery, NAT traversal, chunked transfers, integrity checking, and end-to-end encryption can be combined to provide secure, resumable, and efficient file sharing between clients.

# KEY FEATURES

• Direct File Transfer between clients using WebRTC or libp2p  
• Manual / Ephemeral Signaling via QR or copy-paste  
• Chunked & Resumable Transfers  
• Content Addressing (hash-based)  
• Discovery Options (mDNS, DHT, invites)  
• End-to-end Encryption  
• Integrity & Authentication  
• UX Friendly (progress UI, QR invites, explicit accept/deny)

# FLOWCHART

A diagram of a data transfer process

AI-generated content may be incorrect.

# DATA STRUCTURES / ALGORITHMS USED

• Chunk arrays for received data  
• Queues for outgoing chunks  
• Hash maps for indexing  
• Merkle trees for integrity  
• IndexedDB/local store for persistence  
• Kademlia DHT for discovery

# SECURITY CHECKLIST

• End-to-end encryption  
• Peer identity verification  
• Signed manifests  
• Checksums  
• Permission prompts  
• Rate limiting  
• Malware warnings

# ADVANTAGES

• No central server required  
• Direct and efficient transfers  
• Content addressing supports caching  
• Browser compatibility

# LIMITATIONS / TRADEOFFS

• NAT traversal may need TURN server  
• Persistence requires pinning or always-on peers  
• Manual signaling is less user-friendly  
• Complex for large-scale distribution

# CONCLUSION

This project demonstrates how modern P2P technologies let clients share files directly without a central server. Using WebRTC for browser-to-browser transfers and libp2p/IPFS for content-addressed distribution, the system can provide private, efficient, and resilient file sharing. Challenges like NAT traversal, persistence, and peer identity are addressed with existing solutions.