Applied and Action Learning
(Learning by Doing and Discovery)

Name of the Experiement : Build the Network – Peer-to-Peer (P2P) Simulation

Objective

- To study how a peer-to-peer (P2P) network works and understand how devices communicate directly without a central server.

- To demonstrate how blockchain networks rely on P2P connections for decentralization and data sharing among nodes.

Apparatus/Software Used:

- Node.js (using the `net` module for simulation)
- Localhost setup for network testing
- **Command-line interface (CLI)** to run multiple peer nodes
- Remix IDE (for blockchain-related conceptual understanding)
- Ganache / **Local blockchain network** for node demonstration

Theory/Concept:

A **Peer-to-Peer (P2P)** network is a type of distributed system where every connected device (known as a *peer*) can communicate directly with others. Unlike traditional client-server systems, P2P networks do not depend on a single central authority. Each peer can both send and receive data, making the network more flexible and robust.:

Main Features of P2P Networks:

1. Decentralized Structure:
   There is no central server; all peers have equal status and can independently share data.

2. Resource Distribution:
   Each node can share its files, bandwidth, or storage with other connected peers.

3. Fault Resilience:
   If one peer goes offline, the network continues to function because other peers still hold data copies.

4. High Scalability:
   The system can easily expand as more peers join the network.

5. Autonomy:
   Every node maintains its own records and can participate in tasks such as validation or data propagation.

Procedure :

- **Node Initialization:**
  Create multiple peers such as Node A, Node B, Node C, and Node D in the local environment.

- **Connection Setup:**
  Establish socket connections between all peers so that they can communicate directly.

- **Message Broadcasting:**
  Send a transaction or message from one node and ensure it reaches all connected peers through broadcasting.

- **Data Validation:**
  Receiving peers check the message integrity using hash verification to confirm accuracy.

- **Adding a New Peer:**
  Introduce a new peer into the network and allow it to synchronize the existing data from others

- **Fault Handling Test:**
  Disconnect one peer and observe how the rest continue communicating, showing fault tolerance..

Observation table:

Step	Action Performed	Expected Outcome
1	Nodes initialized	All peers created successfully
2	Connections established	Peers connected and recognized each other
3	Message broadcasted	All nodes received the message
4	Data verified	Message integrity confirmed
5	New peer joined	Peer synced with network data
6	One peer disconnected	Other peers continued functioning normally

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Interpretation Result and	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	50		

Signature of the Student:
Name :
Regn. No.

Signature of the Faculty: