Centurion
UNIVERSITY

School: .... Campus: ....

Academic Year: .... Subject Name: .... Subject Code: ....

Semester: .... Program: .... Branch: .... Specialization: ....

Date: ....

Applied and Action Learning

(Learning by Doing and Discovery)

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

* Coding Phase: Pseudo Code / Flow Chart / Algorithm

Initialize Network

Define the number of nodes (peers) participating in the network.
Assign each node a unique ID and empty ledger/memory.

Create Connections

Each node connects to a few other nodes randomly (simulating a mesh network).
Maintain a list of peers each node can send/receive messages from.

Message Broadcast

One node (the initiator) sends a transaction or message to its peers.
Each peer forwards the message to its connected nodes (except the sender).

Verification Process

Each node validates the message (e.g., checks if it's new and not already received).
Invalid or duplicate messages are ignored.

Ledger Update

If the message (transaction/block) is valid, it's added to the node's ledger.

Consensus (Optional)

If simulating blockchain consensus, include a simple rule like:
The first valid message received by all nodes is accepted.
Nodes reject conflicting data.

End Simulation

Display how many nodes successfully received and accepted the message.
Show that the network achieved synchronization without a central server.

Software used

1. MetaMask Wallet
2. VS Code.
3. MS Word.
4. Brave for researching.

*As applicable according to the experiment.
Two sheets per experiment (10-20) to be used.

* Implementation Phase: Final Output (no error)

Input: Number of peers (e.g., 6)

Establish peer connections.

Node 1 broadcasts a message.

Nodes 2–6 receive and validate the message.

Ledger updated in each node.

Output:

```
Message broadcast from Node 1
Node 2 received message from Node 1
Node 3 received message from Node 2
Node 4 received message from Node 3
All nodes synchronized ✅
```

* Observations:

- The message reaches all nodes without any central server.
- Duplicate messages are automatically avoided using validation.
- The network demonstrates **decentralization**, **fault tolerance**, and **equal node importance**.
- Communication delay or missing peers can affect synchronization — similar to real blockchain networks.
- When consensus rules are added, the simulation behaves like a mini blockchain environment.

**As applicable according to the experiment. Two sheets per experiment (10-20) to be used.*