



Centurion
UNIVERSITY
*Shaping Lives...
Empowering Communities...*

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 Experiment :

*** Coding Phase: Pseudo Code / Flow Chart / Algorithm**

Algorithm:

1.Introduction to Web3:

Web3 represents a decentralized internet powered by blockchain, where users own data and interactions via cryptographic identities like wallets and smart contracts.

2.Role of AI in Web3:

AI enhances the intelligence of decentralized applications (DApps) through prediction, personalization, and real-time automation.

AI models can be hosted and queried in decentralized compute systems (e.g. Fetch.ai, Ocean Protocol).

3.AI + Blockchain Integration Models:

On-chain AI: Smart contracts interacting with lightweight ML models (limited due to gas constraints).

Off-chain AI Oracles: AI models hosted off-chain provide results to smart contracts via oracles like Chainlink.

4.Decentralized AI Marketplaces: Networks like SingularityNET let users buy/sell AI services using cryptocurrency.

Decentralized Data Layer for AI: Data lakes on Filecoin/IPFS allow secure and permissioned data training.

5.Use Cases of AI + Web3:

- Fraud detection in decentralized finance (DeFi)
- Autonomous economic agents in IoT
- Decentralized social media content moderation
- Personalized AI agents for identity and credentials

*** Softwares used**

No coding or implementation – purely research and presentation based (AI models, Web3 concepts, decentralized data frameworks).

* Implementation Phase: Final Output (no error)

During the practical session, students formed groups and prepared a 5–7 slide presentation demonstrating the connection between AI and blockchain in real-world applications. Each group selected a specific AI + Web3 use case and presented a visual flow with architecture diagrams.

1. Use Case Demonstration:

- Example chosen: Autonomous Fraud Detection in DeFi using AI Agents
- AI model trained off-chain reads real-time blockchain transaction data via oracles

2. The AI sends fraud alert score back to smart contract for blocking suspicious transfers

- Architecture Diagram Shared (Sample Flow):
- User → Blockchain Network → Oracle → AI Engine → Smart Contract → Action on-chain
- Diagram illustrated how off-chain AI interacts with on-chain logic

3. Live Demo (Optional):

- Simulated decentralized sentiment analysis model (mock predictions) interacting with a governance smart contract
- When sentiment score < threshold → Contract triggers automated proposal vote

4. Tools Used:

- Blockchain: Ethereum Testnet (Goerli/Sepolia)
- AI Model Hosting: Local Python-based ML script (mock deployment)

5. Wallets: MetaMask for user interaction

- Diagramming Tool: Draw.io / Figma for architecture visualization

6. Presentation Summary:

- Highlighted the hybrid nature of AI + Web3 systems
- Discussed challenges: Gas cost, model verification, privacy of training data

7. Emphasized future opportunities like on-chain AI agents, DAO-powered AI funding, decentralized compute networks

* Implementation Phase: Final Output (no error)

Applied and Action Learning

* Observations

- 1.AI and Web3 combined enable distributed intelligence, which is not controlled by any single entity.
- 2.Decentralized storage (IPFS, Filecoin) and smart contracts enable fully transparent AI workflows.
- 3.AI agents in decentralized environments may enhance privacy and decision-making through encrypted computation.
- 4.High costs and slow execution on-chain limit AI deployment directly on blockchain — hybrid systems are preferred.

ASSESSMENT

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

Signature of the Student:

Name :

Regn. No. :

Signature of the Faculty:

Page No.....

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