Mini Task 1: Build & Explain a Simple Blockchain

Mini Task 1: Build & Explain a Simple Blockchain

1. Blockchain Basics
    - Define blockchain in your own words (100–150 words).
    - List 2 real-life use cases (e.g., supply chain, digital identity).

Answer: Blockchain is a decentralized digital ledger that records transactions in "blocks" linked together in a chronological chain. Each block contains a batch of validated transactions and a unique cryptographic hash referencing the previous block. Because it's shared across many computers (nodes), no single person or organization controls it. Once a transaction is added, it's nearly impossible to change without the network noticing, making the system secure, transparent, and trustworthy. The result is a distributed database where entries are permanent and verifiable—ideal for applications that need reliable records without relying on intermediaries .

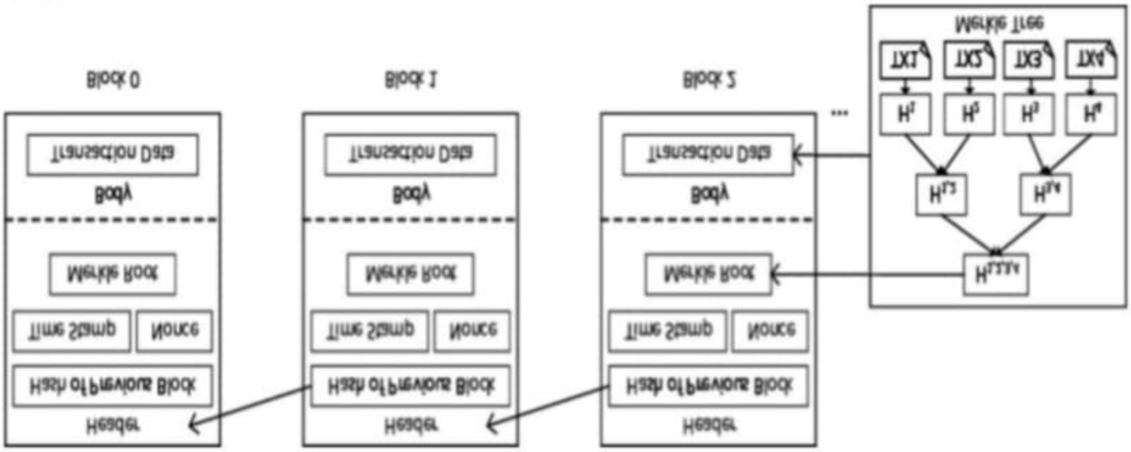
REAL LIFE USE CASES:

1. Supply Chain Transparency
Companies like Walmart and Bumble Bee Seafoods use blockchain to track food from the farm to your plate. By scanning a QR code, you can see where your mango or tuna came from and that it was handled safely. This helps prevent fraud and makes recalls faster.

2. Digital Identity & Personal Data Security
Apps like Civic use blockchain to help you store your personal information securely. You stay in control—only you decide when to share details like age or ID. It keeps your data safe from hackers and unnecessary sharing.

2. Block Anatomy
    - Draw a block showing: data, previous hash, timestamp, nonce, and Merkle root.
    - Briefly explain with an example how the Merkle root helps verify data integrity.

Answer:



A Merkle Root is like a summary fingerprint of the block's transactions. It lets you verify that a transaction is included and unchanged without checking every single one.

Example:

Say a block has 4 transactions:
T1, T2, T3, T4.

1. **Hash each transaction:**
   H1 = hash(T1), H2 = hash(T2), etc.
2. **Pair and hash:**
   H12 = hash(H1 + H2), H34 = hash(H3 + H4).
3. **Merkle Root = hash(H12 + H34).**

To check T2 without downloading all data:

- **Get H1 (the hash of T1) and H34 (hash of T3+T4 pair).**
- **Recompute H12 = hash(H1 + hash(T2)), then the root = hash(H12 + H34).**
- **If that matches the Merkle Root in the header, T2 is valid and untampere**

3: Consensus Conceptualization

- **Explain in brief (4–5 sentences each):**
  - **What is Proof of Work and why does it require energy?**
  - **What is Proof of Stake and how does it differ?**
  - **What is Delegated Proof of Stake and how are validators selected?**

Answer: Proof of Work (PoW):

Proof of Work is a method where powerful computers (miners) race to solve a tough puzzle. The first one to solve it gets to add the next block to the blockchain and wins a reward. It uses lots of electricity because these computers run puzzle-solving algorithms millions of times per second, like running marathons in the computer world . This big energy usage keeps attackers out—it's too expensive to cheat the system.

Proof of Stake (PoS):

Proof of Stake replaces energy-heavy puzzles with a lottery-like system. Instead of mining, people "stake" (lock up) their coins. The system picks a validator at random, but chances are higher for those who stake more. Validators earn from transaction fees and risk losing their staked coins if they misbehave. It uses far less energy, making it greener and cheaper

Delegated Proof of Stake (DPoS):

In DPoS, coin-holders vote to elect a small team of validators (delegates). Only these selected delegates take turns creating new blocks. Votes are based on how many coins each person stakes. If delegates act maliciously or perform poorly, voters can remove them and choose new ones. This setup speeds up transactions and adds a democratic layer to blockchain security.