Task 02

Name: Shrilakshmi Kakati

Group: B

Comparison Table

Blockchain Name	Ethereum	Hyperledger Fabric	Corda
Type (Public/Private/Consortium)	Public	Private	Consortium
Consensus Mechanism Used	Proof of Stake	Pluggable	Notary
Permission Model (Open/Permissioned)	Open	Permissioned	Permissioned
Speed / Throughput (TPS if available)	15–30 TPS	1,000+ TPS	170 TPS
Smart Contract Support (Y/N + Language)	Yes	Yes	Yes
Token Support (Native or not)	Native (ETH)	Not Native	Native
Typical Use Case	DApps, DeFi, NFTs	Supply chain, healthcare, enterprise	Financial agreements, enterprise
Notable Technical Feature (e.g., privacy, pluggable consensus)	Turing-complete smart contracts	Modular architecture with private channels	Point-to-point privacy by default

Compare and contrast the technical capabilities of each.

Ans: **Ethereum**, the flagship public blockchain, thrives on decentralization and openness. Powered by Proof of Stake and robust smart contract support via Solidity, it's the ideal playground for decentralized apps (DApps). However, its relatively modest throughput (~15–30 TPS) and high gas fees limit scalability for enterprise use.

Hyperledger Fabric, in contrast, is a private blockchain optimized for enterprise needs. With pluggable consensus (like Raft), private channels, and modular architecture, it offers superior throughput (~1,000+ TPS) and data privacy—ideal for a **supply chain among known partners** where trust exists but data confidentiality is crucial.

Corda, designed for consortiums, blends decentralization with privacy. Its unique notary-based consensus ensures only involved parties access data, making it perfect for **inter-bank financial applications** where transactional privacy and legal enforceability are non-negotiable.

2. Which platform would you choose for:
 a) A decentralized app?
 b) A supply chain network among known partners?
 c) An inter-bank financial application?

Ans: **Hyperledger Fabric** is the ideal choice for a **supply chain network among known partners** because it's specifically designed for **private, permissioned environments** where participants are trusted but require **data confidentiality** and **fine-grained access control**.

- **Modular Architecture**: Fabric allows organizations to customize the blockchain with pluggable consensus mechanisms (like Raft) and modular components, making it adaptable to complex supply chain workflows.

- **Private Channels**: Unique to Fabric, private channels allow subsets of participants (e.g., supplier and distributor) to transact privately without exposing data to the entire network—critical for competitive business data.

- **High Throughput**: Capable of handling over **1,000 TPS**, Fabric supports the scale and speed needed for global logistics and real-time tracking.

- **No Native Token Requirement**: Unlike public chains, Fabric doesn't use a cryptocurrency, avoiding unnecessary costs and regulatory overhead.

- **Rich Identity Management**: Through Membership Service Providers (MSPs), it offers robust identity control and role-based access—essential in regulated industries.

In short, Fabric provides **privacy, performance, and control**, making it the **go-to blockchain** for secure, efficient supply chain collaboration.

3. Justify your choice based on technical points.

Ans: Justification on my choice of hyperledger is

1. Permissioned Architecture:
   Hyperledger Fabric is designed for networks where all participants (manufacturers, suppliers, distributors, etc.) are known and trusted. This ensures greater control over who can read and write data, essential for business confidentiality.

2. Private Channels for Confidentiality:
   Fabric allows the creation of private channels—isolated ledgers between specific participants. This means a supplier and manufacturer can share sensitive order details without exposing them to others in the network.

3. High Throughput (~1,000+ TPS):
   Supply chains handle a large volume of transactions. Fabric's architecture supports parallel execution and fast consensus (like Raft), ensuring performance scales with demand.

4. No Native Cryptocurrency:
   Fabric avoids using tokens or mining, reducing operational complexity and regulatory overhead. It focuses purely on data integrity and workflow automation.

5. Modular and Flexible Design:
   Fabric's plug-and-play nature allows customization of consensus, identity management, and data storage—tailored to the specific needs of each supply chain ecosystem.

6. Smart Contracts with Chaincode:
   Business logic is implemented using general-purpose languages (Go, Java, Node.js), making it easier for existing enterprise developers to build and maintain applications.