UNIT-IV **(15 Hours)**

Probabilistic versus quantum algorithms, Quantum parallelism, Deutsch algorithm, Deutsch-Jozsa algorithm, Simon's algorithm, Shor's algorithm

References:

1. *Quantum Computation and Quantum Information, M A Nielsen and I L Chuang.*
2. *An Introduction to Quantum Computing, P Kaye, R Laflamme and M Mosca.*

Practicals:

1. Use Qiskit or IBMQ to create a qubit in superposition and visualize it on the Bloch Sphere.
2. Build a quantum circuit to generate an entangled Bell state and verify it using measurement results.
3. Simulate quantum teleportation of a qubit's state using a quantum circuit with entanglement and classical bits.
4. Create a quantum circuit for Deutsch's algorithm and run it to determine if a function is constant or balanced.
5. Use IBMQ or Qiskit to implement basic quantum gates (X, H, Z, CNOT) and observe their effect on quantum states.

DSE – 24
Blockchain and its Applications

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Blockchain and its Applications	**4**	**3**	**0**	**1**	**Class XII pass with Mathematics**	**NIL**

Learning Objectives:

1. *To cover the basic concepts behind blockchain and present Bitcoin and other cryptocurrencies as the motivation for blockchain technologies.*
2. *To provides a substantive discussion about different technologies behind blockchain and cryptocurrencies.*

Learning Outcomes:

1. *Understand the fundamental principles and structure of blockchain technology, including cryptographic components and decentralized systems.*
2. *Analyze and compare consensus mechanisms such as Proof of Work and Proof of Stake in real-world blockchain networks.*
3. *Demonstrate practical knowledge of cryptocurrencies, transactions, wallets, and major platforms like Bitcoin and Ethereum.*
4. *Design and deploy basic smart contracts using Ethereum and Solidity for decentralized applications.*

UNIT-I **(10 Hours)**

Introduction: History and evolution of money to digital currencies, Introduction to Blockchain: concept, purpose, key features, Cryptographic foundations: Hashing, Digital Signatures, Public vs Private vs Consortium Blockchains, Blockchain structure: Blocks, Hash functions, Block headers, Merkle Trees.

UNIT-II **(10 Hours)**

Consensus and Decentralized Systems: Peer-to-peer networks and decentralization principles, Consensus algorithms: Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Forks and governance in blockchain systems, Blockchain scalability and energy concerns.

UNIT-III **(10 Hours)**

Cryptocurrencies and Transactions: Bitcoin architecture: Addressing and UTXO model, Ethereum basics: Accounts, Gas and Ether, Transactions and Wallets, Overview of major cryptocurrencies: Bitcoin, Ethereum, Stablecoins (USDT, USDC).

UNIT-IV **(15 Hours)**

Smart Contracts and Applications: Smart Contracts: Concept and use-cases, Introduction to Ethereum and Solidity, DeFi overview: Lending, DEX, Staking, Real-world applications: Supply Chain, NFTs, Digital Identity, Legal and ethical considerations in smart contract deployment.

References:

1. Mastering blockchain Distributed ledger technology, decentralization, and smart contracts explained by Imran Bashir, 2nd edition (2018), Packt Publication.
2. Mastering Blockchain Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications by Lorne Lantz and Daniel Cawrey, 1 st Edition (2020), O'Reilly Publication.
3. Introducing Ethereum and Solidity Foundations of Cryptocurrency and Blockchain Programming for Beginners by Chris Dannen, 1st Edition (2017), Apress Publication.

Practicals:

1. Using SHA256, obtain the message digest of string "Blockchain Developer".

2. Write a program to encrypt and decrypt the message “Hello World” using
SHA256. 3. Implement a simple chain of 5 nodes using linked list.
4. Implement RSA cryptographic algorithm.
5. Create a simple blockchain using Proof of Work (PoW).
6. Demonstrate sending of a digitally signed document.
7. Create a hash table that has ‘8’ number of buckets and insert the keys 20, 1, 7, 15, 25,16, 8, 20 using hash function h(k) = k mod 3.
8. Create a blockchain having 5 nodes and print the hash values of each
block. 9. Create a blockchain having 5 nodes and check its validity.
10. Use flask to deploy blockchain.