**Project Title:**

**Decentralized Identity System Using Blockchain and Zero-Knowledge Proofs (ZKP)**

**Objective:**

The objective of this project is to develop a decentralized identity management system that enables **secure and private identity verification** using **blockchain** and **Zero-Knowledge Proofs (ZKP)**. This system ensures **user control, privacy, and protection against identity theft**, allowing individuals to verify credentials without revealing sensitive information.

**Problem Statement:**

Traditional identity systems rely on centralized authorities (governments, banks, tech companies), creating serious security and privacy risks:

1️. Data Breaches: Centralized databases (e.g., Equifax, Aadhaar) are prime targets for hackers, exposing millions of users.

2️. Privacy Issues: Users are forced to share excessive personal details

3️. Lack of User Control: Organizations own and control identities, not the individuals themselves.

4️. Inefficient Verification: Users must repeatedly verify their identity with different services, causing inconvenience.

Thus, a decentralized and privacy-preserving identity system is needed to address these challenges.

**Solution:**

This project proposes a Decentralized Identity System (DID) that leverages:

1. Blockchain (Ethereum/Polygon/Hyperledger) – Stores decentralized identity records securely.
2. Zero-Knowledge Proofs (ZKP) – Allows identity verification without revealing personal data.
3. IPFS (InterPlanetary File System) – Stores encrypted identity documents in a decentralized manner.
4. Smart Contracts (Solidity) – Automates identity issuance and verification.

**Programming Languages & Frameworks:**

1. Blockchain Development: Solidity (Ethereum Smart Contracts)
2. Cryptography & ZKP: Circom, SnarkJS, ZoKrates
3. Decentralized Storage: IPFS (InterPlanetary File System)
4. Backend & Web App: Node.js, Express.js, React.js

**Methodology:**

1. Define System Components: Identify key entities (Issuer, Holder, Verifier) and set up DID structure on the blockchain.
2. Identity Creation & Storage: Users generate DIDs, store them on Ethereum/Polygon, and encrypt identity documents on IPFS.
3. Zero-Knowledge Proof (ZKP) Verification: Use zk-SNARKs to verify identity without revealing personal details.
4. Smart Contracts for Verification: Deploy smart contracts to validate ZKP-based identity proofs on-chain.
5. Decentralized Login & Authentication: Users log in via DIDs, and websites verify identity through blockchain proofs.
6. User Interface & Integration: Develop a React.js web app with Metamask/WalletConnect, integrating backend APIs (Node.js) for seamless interaction with the blockchain.