Software Design

Software : Python 3, Flask Library.

BlockChain Design

Store the data into blocks.

→ Each block will have Unique ID.
→ Multiple Blocks.
→ Data: Name, Age, Biometrics, etc.

Making blocks immutable

→ Hash f^n will be used
→ SHA 256
→ Library: hashlib.

Start chaining the Blocks.

→ Hash of previous block into the current block
→ If content of previous Block changes, mismatch will occur.
→ 1st Block → Genesis Block.

Implementing Proof of Work (POW)

→ If previous Block content changes, hash can be recomputed.
→ TO prevent this, hash will be calculated in different way
→ Constraints eg: 2 leading Zeros
→ NONCE : A number that will keep changing until we get hash that fulfills our constraints.

Adding Blocks to chain

→ Verify POW
→ Add the blocks

Mining (confirm)

→ The data will be stored in a pool of unconfirmed block.
→ Putting the block & verifying POW will be mining.

Interface (GUI)

REST-API to Interact with our node.

→ Flask Library will be used.
→ An endpoint to submit new transaction
→ A command to mine.

Establishing consensus & decentralization

→ Creating multiple nodes across servers.
→ Selecting Longest valid chain.
→ Everyone can update & move on to mine other data.

Building the application

→ Basic HTML, CSS, Json
→ POST request to a connected node, to add the data in the unconfirmed pool
→ Followed by mining, the data → Refresh.

Cloud

→ IBH → AWS → GCP.

→ Creating Accounts using public-private key Cryptography.
→ New user needs a public key & private key to POST.
→ Key will be 'Digital Signature'.
→ Public key can decode the content encrypted by corresponding Private key
→ Public key will verify the transaction.