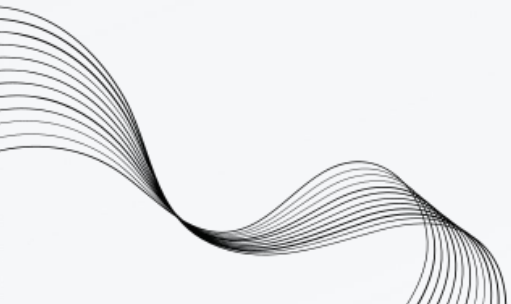
DeFi App

TEAM MEMBERS

Name	Email
Mahaan N Bhat	mahaan321@gmail.com
Manish K Shetty	manishshetty1234567@gmail.com
Rao Chaitanya	raochaitu13@gmail.com
Nidhisha Naik	nidhishanaik@gmail.com

INTRODUCTION

To develop a decentralized banking system leveraging blockchain technology, allowing users to interact with financial services such as loans, transactions, deposits, and withdrawals in a secure, transparent, and trustless environment.



TECH STACK

Frontend:

- **React:** JavaScript library for building user interfaces.
- **React Router:** For handling navigation and routing between different pages.
- **Web3.js:** JavaScript library to interact with Ethereum blockchain and smart contracts.
- **MetaMask:** Browser extension for managing Ethereum wallet and connecting to the blockchain.

Smart Contracts:

- **Solidity:** Programming language for writing smart contracts on Ethereum.
- **Remix IDE:** Online IDE for developing, deploying, and testing smart contracts.
  -

TECH STACK

Development Environment:

- **Truffle:** Development framework for Ethereum, used for smart contract compilation, deployment, and testing.
- **Ganache:** Personal blockchain for Ethereum development, used for local testing of smart contracts.
- **Node.js:** JavaScript runtime environment for executing server-side code.
- **npm:** Package manager for JavaScript, used for managing project dependencies.

Backend Infrastructure:

- **Ethereum Blockchain:** Decentralized platform for running smart contracts.
- **Infura (optional):** API to connect to the Ethereum network without running a full node.
  -

TECH STACK

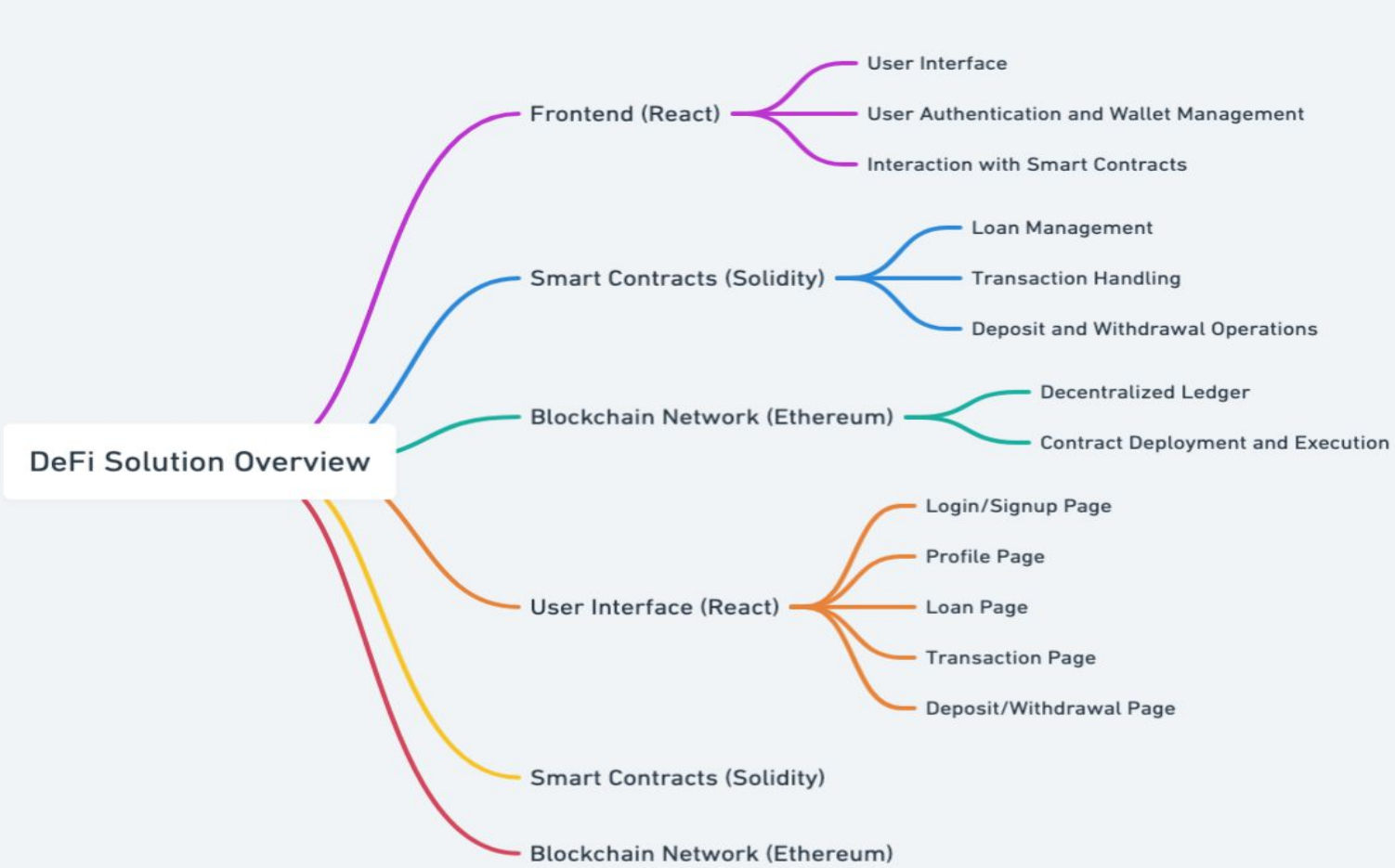
Libraries and Tools:

- **axios:** Promise-based HTTP client for making API requests.
- **dotenv:** Module to load environment variables from a `.env` file into `process.env`.

APIs:

- **MetaMask API:** For user authentication and wallet management.
- **Web3.js API:**
  - **web3.eth.Contract:** For creating contract instances and interacting with deployed contracts.
  - **web3.eth.getAccounts:** To retrieve user accounts connected to MetaMask.
  - **web3.eth.getBalance:** To fetch the balance of a user account.
  - **web3.eth.sendTransaction:** To send transactions to the Ethereum blockchain.

SYSTEM DESIGN OR LOGIC



SYSTEM DESIGN OR LOGIC

Logic Behind the Project

1. User Interface (React)

Login/Signup Page:

- Users log in or sign up using MetaMask, which manages their Ethereum accounts.
- Authentication is handled through MetaMask, fetching the user's wallet address.

Profile Page:

- Displays user details such as account balance and recent transactions.
- Users can view loan details and other financial information.

Loan Page:

- Users apply for loans by providing details like loan type, amount, interest rate, and repayment date.
- Smart contracts handle loan issuance and store loan details on the blockchain.

SYSTEM DESIGN OR LOGIC

Transaction Page:

- Displays the user's transaction history, including loan transactions, deposits, and withdrawals.

Deposit/Withdrawal Page:

- Users can deposit and withdraw funds directly to and from the smart contract.
- The smart contract updates the user's balance and records the transaction on the blockchain.

2. Smart Contracts (Solidity)

DeFiBank.sol:

- **Loan Management:**
  - `issueLoan`: Allows users to request loans with specified parameters.
  - `repayLoan`: Users repay loans, and the contract calculates the remaining balance and updates the loan status.

SYSTEM DESIGN OR LOGIC

- **Transaction Handling:**
  - Tracks all user transactions, ensuring transparency and immutability.
- **Deposit and Withdrawal:**
  - Users can deposit funds to their account and withdraw available funds.
  - The smart contract maintains user balances and securely handles fund transfers.

3. Blockchain Network (Ethereum)

Smart Contract Deployment:

- Smart contracts are deployed on the Ethereum network using Remix IDE.
- Deployment details, including the ABI and contract address, are retrieved for frontend integration.

Transaction Execution:

- All interactions with the smart contract (loan issuance, repayment, deposits, withdrawals) are executed as blockchain transactions.
- The Ethereum network ensures all transactions are securely recorded and immutable.

USE CASES

Personal Loans:

- Users can apply for personal loans with specified terms.
- Loan issuance and repayment are managed by smart contracts.

Microfinance:

- Small business owners can obtain microloans.
- Loans are issued and repaid in small installments via smart contracts.

Investment Opportunities:

- Users can invest in DeFi products through the platform.
- Smart contracts manage investments and payouts transparently.

Savings Accounts:

- Users can deposit funds into high-interest savings accounts.
- Smart contracts handle interest accrual and withdrawals.

USE CASES

Automated Loan Repayment:

- Users can schedule automatic loan repayments.
- Smart contracts deduct repayment amounts and update loan statuses.

Transparent Financial Transactions:

- All transactions are recorded on the blockchain.
- Users have an immutable ledger of their financial activities.

Peer-to-Peer Lending:

- Users can lend money to others with predefined terms.
- Smart contracts facilitate loan agreements and repayments.

Credit History Building:

- Users can build credit history by taking and repaying loans.
- Blockchain records provide transparent creditworthiness.

THANK YOU