BLOCKCHAIN PROJECT

Decentralized payment System for Ecommerce website

SYSTEM MANUAL

Presented by

Sandeep Chanda

14427033

Under the Guidance of

A S M Touhidul Hasan

At

UMKC

UNIVERSITY OF MISSOURI-KANSAS CITY

System Manual:

Decentralized payment System for Ecommerce website

1. Overview:

This project provides a decentralized payment page for E-Commerce website where user can buy tokens and use it for shopping here, I used solidity for smart contract development, Hardhat for testing and development and next.js for budling frontend and Polygon Amoy as a blockchain platform ICO. This decentralized payment method make customer feel more secure and safe for payments and data privacy for companies it provides full control over token economy it can bring potential customers across the globe

System Workflow:

1. **ERC20 Token Creation**: The ERC20 token is created using a smart contract deployed on the Ethereum blockchain. The contract is designed to manage token minting, transfers, and balances.
2. **ICO Token Sale**: The token sale contract is used to sell tokens to investors in exchange for POL. The sale has a fixed rate, and users can buy tokens by sending POL
3. **E-Commerce Payment**: After buying tokens, users can use them to pay for goods and services within the integrated e-commerce store.

Core Features

1. User Friendly Interface:

- Simple and accessible user Interface
- Access to global audience

2. Smart Contract Integration:

- Selling Tokens
- Check balance
- Initial Coin Offering

3. Transparent Transactions:

- All transactions are recorded on block chain

4. Security:

- Full control on token economy
- Measures to safe guard user funds

2. Software Requirements

Hardware requirements:

- 4 GB of above Ram
- At least 20 GB Free Storage

Software Requirement

- **Operating System:** windows 10+, macOS, Linux
- **Node.js**: Version 14.x or later
- **Git: Version** Control Software
- **Ethereum wallet:** MetaMask Wallet
- **Vs Code:** IDE **(**latest version**)**

Technologies Used

1. **Solidity:** solidity for smart contract development
2. **Hardhat**: Hardhat for smart contract testing and deployment
3. **Next.js:** for building the frontend web application.
4. **Polygon Amoy:** For deploying and interacting with smart contracts.

Libraries and Framework:

1. **Solidity:** Smart Contract programming language
2. **hardhat:** development framework
3. **React.js:** frontend framework for UI
4. **Web3.js:** blockchain interface library

3. Development Environment Setup:

Installation

1.Clone the Repository

2.Install Hardhat Dependencies

3.Compile Smart Contracts

4.Deploy the Smart Contract

5. Application

Once dependencies are installed, start the React application:

Bash

Configuration

1. Set Up Environment Variables

You will need an API key for Pinata (a service to interact with IPFS). Sign up for an account on [Pinata](https://www.pinata.cloud/) and obtain your API keys.

- In the project, navigate to the React component `FileUpload.js` and update the code with your Pinata API keys (replace the placeholders with your actual keys).

2. Update Smart Contract Address in React App

After deploying your smart contract to the Ethereum network, obtain the contract address. Then, in the App.js file within the React client directory, update the contract address with the correct value:

javascript

```
const contractAddress = "<YOUR_CONTRACT_ADDRESS>";
```

5. Usage

Selling Cryptocurrency

1. Navigating to the sell crypto section:

- After connecting your wallet, locate the token sell section on the DApp's

2. Enter the Amount to sell:

- Specify the amount of cryptocurrency you want to sell
- The DApp will show the equivalent amount in ETH or your preferred currency

3. Confirm transaction

- Click sell to initiate the sale
- MetaMask will prompt you to confirm the transaction. Review the gas fees and confirm the transaction to proceed

4. **Transaction Competition:**

    - Once the Transaction is completed on the blockchain you will receive a transaction complete notification

    - Check MetaMask wallet to verify funds have been updated

    - In a few situations application gets a transaction declined notification and declined in MetaMask wallet to too but when you check in account on the blockchain page the transaction was successful

6. Check your Balance

- Go to the balance section on DApp to view your current balance

- Check the sales section to view the total no of sales completed it also gives a percentage of sales

Troubleshooting

Wallet not connecting;

- Ensure MetaMask is installed and unlocked if not refresh the page and connect again
- always make sure to have your secret keywords saved in a safe location so that you don't lose your accounts if you forget your password

High Gas Fees:
Gas fees can fluctuate so try selling them during low-traffic

Maintenance and Updates:

- **Bug fixes:** address any issues
- **Security patches**: Regularly update libraries and check security vulnerabilities
- **Feature Enhancements**: based on feedback, add features to improve the DApp

Security Considerations

- Private Keys: Never share your private keys or seed phrases for your Ethereum wallet. Always use MetaMask for handling transactions securely.

\