Capstone Project II
Academic Year- 2021-22

SRS

BlockPay: A blockchain-based transaction system

1.	Tripti Lamba	BT18GCS193
2.	Kanishya Mohan	BT18GCS005
3.	Varshith Sharma	BT18GCS302
4.	Yuvraj Singh	BT18GCS168
5.	Sahil Singh	BT18GCS146

Contents:

1. Introduction

This platform eliminates the hassle of exchanging money and making the process of transactions faster. It ensures security by encryption to ensure transaction blocks are indisputable and accurate using blockchain technology. This fulfills transparent transactions for the user and specifies the basic transaction details. Also, end-users are saved from the transaction charges and exchange rate loss that are part of international transactions.

1.1 Document Purpose

BlockPay is a system that enables all users to perform easy transparent transactions. The purpose is to enable users to perform transactions using blockchain similar to the bitcoin framework. Through this software, the hassle of managing money comes to an online platform. Since this software is on a blockchain platform, it's secure, immutable, and cannot be manipulated.

1.2 Product Scope

When it comes to matters of money, users seek a trusted medium through which their financial demands are fulfilled therefore, building a simple web application is not preferable for this purpose because the user's money can be stolen, transactions can be manipulated, balance can be altered, etc. For these reasons, we are revolutionizing the transaction process. The most important factor of this application is the safety that it provides users. It is easy to log into, feasible and easy to use, transparent, audible. The SRS will help in building the application with the flow and also provide a clear vision of the application before we actually start making the application.

1.3 Definitions, Acronyms, and Abbreviations

The SRS is designed in a way that is comprehensive to whom it concerns. Here is a little detail of the few words used in the SRS

1. GUI: Graphical User Interface
2. HTML: Hyper Text Markup Language
3. CSS: Cascading stylesheet
4. URL: Uniform Resource Locator
5. RSA: Rivest–Shamir–Adleman public key cryptosystem
6. Ledger is an open, distributed file that can record transactions between two parties efficiently and in a verifiable and permanent way.
7. Dapp: Decentralised application

1.4 Document Conventions

The document strictly follows the IEEE standard format and no other formatting is done in the document.

1.5 References and Acknowledgments

1. Blockchain Database System Concepts by Mastering Etherium.
2. A. Popova and N. G. Butakova, Research of a Possibility of Using Blockchain Technology without Tokens to Protect Banking Transactions 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St.Petersburg and Moscow, Russia, 2019, pp. 1764-1768.
3. Yadav N. and S. V. "Venturing Crowdfunding using Smart Contracts in Blockchain," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020.p. 192–197.
4. Nakamoto S. Bitcoin: A Peer-to-Peer electronic cash system.2008[cited 2021 May 19].Available from: https://www.bitcoincash.org/bitcoin.pdf.
5. Yadav N. and S. V. "Venturing Crowdfunding using Smart Contracts in Blockchain," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020.p. 192–197.

2. Overall Description

2.1 Product Perspective

The product is made with the perspective that it will provide users with a safe, transparent, and fast translation experience. We'll have a traditional front-end client that is written in HTML, CSS, and JavaScript. For connecting to the blockchain on the backend we are using Python Flask. The user can create their wallet with a set of public and private keys. Public keys for other wallets can be used as the destination address for the transactions.

2.2 Product Function

This application will allow users to create a wallet and make transactions. There is also a dashboard for miners where they can mine the transactions to add in the blockchain and add nodes to the blockchain network. At any given point, a user can see the entire blockchain and all the transactions made. It is based on a peer-to-peer system that eliminates the need for a middle man. The transactions will be transparent and secure because of blockchain technology.

2.3 Users and Characteristics

There will be two types of interfaces, one for users to create a wallet, perform transactions and check blockchain networks. The other dashboard would be for nodes to maintain the blockchain network wherein you will be able to create nodes and mine transactions.

2.4 Operating Environment

The application can be run on any computer (machine), since it is a web application, it requires an internet connection. The application is supported by both Windows and Linux. No other hardware component is required other than a computer to access the application. Working internet connection is a must.

2.5 Design and Implementation Constraints

The application has been designed for large scale public use therefore it was required to keep the design and implementation constraints as low as possible, however, the few of them could be:

- The customer's organization that is responsible to maintain the project must be familiar with how blockchain works.
- The budget could be a major constraint, for example, if the customer wants more options and is low on budget then it becomes a major design constraint.
- Any changes in the design demanded by the client in the latter stages of application development could bring great implementation troubles.
- GUI is in English only.
- Users should have basic knowledge of computers.

2.6 Assumptions and Dependencies

- The end-user knows English and has basic knowledge of computers.
- Roles and tasks are pre-defined.
- The administrator is created in the system already.

3. External Interface Requirements

3.1 Hardware Interfaces

- The only interfaces are through a computer system.
- The operating system can be Windows/Linux or MAC.

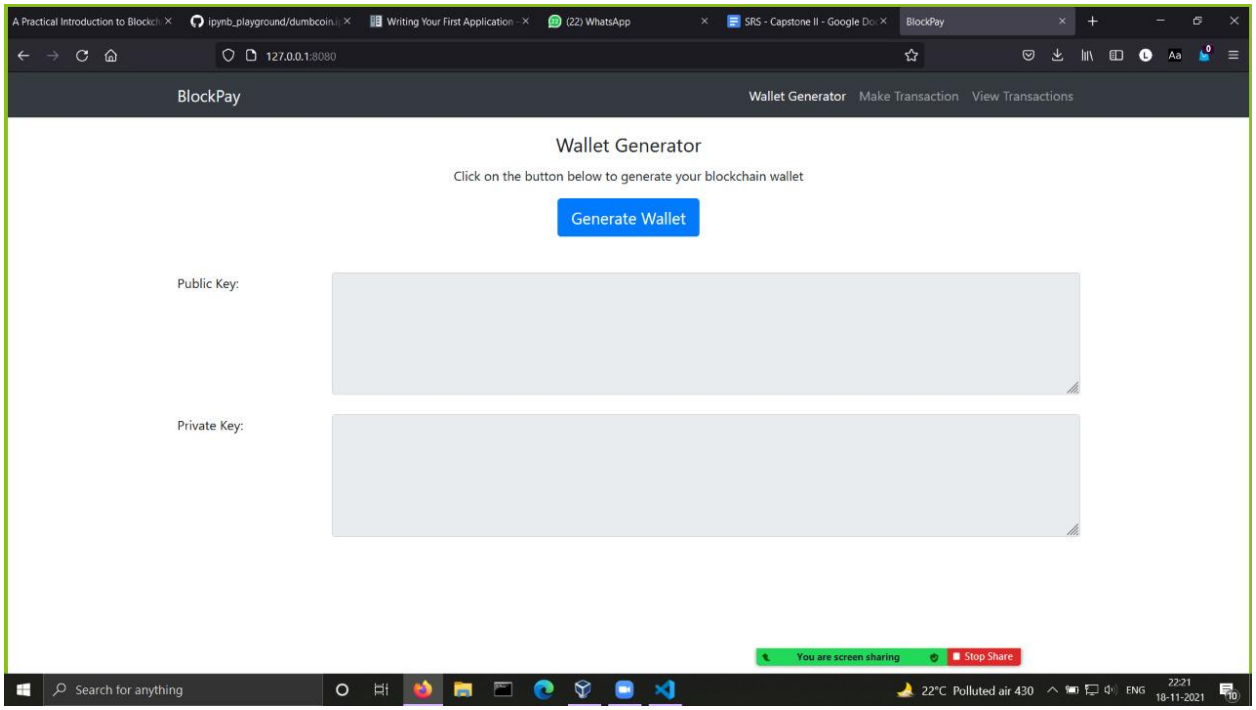
3.2 Software Interfaces

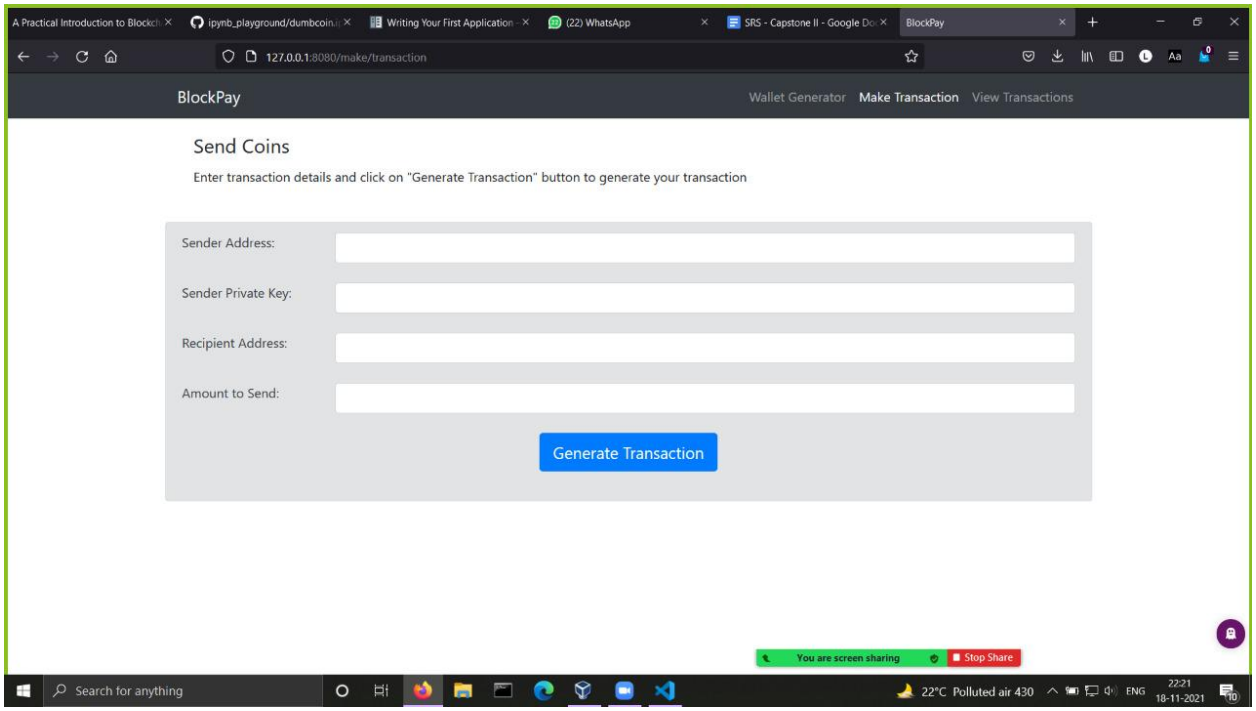
To create this decentralized web app, we will use python, and flask. This project is unique as it uses blockchain to authenticate transactions by checking if the prev hash is equal to the current hash. For the front-end we use HTML, CSS and JavaScript.

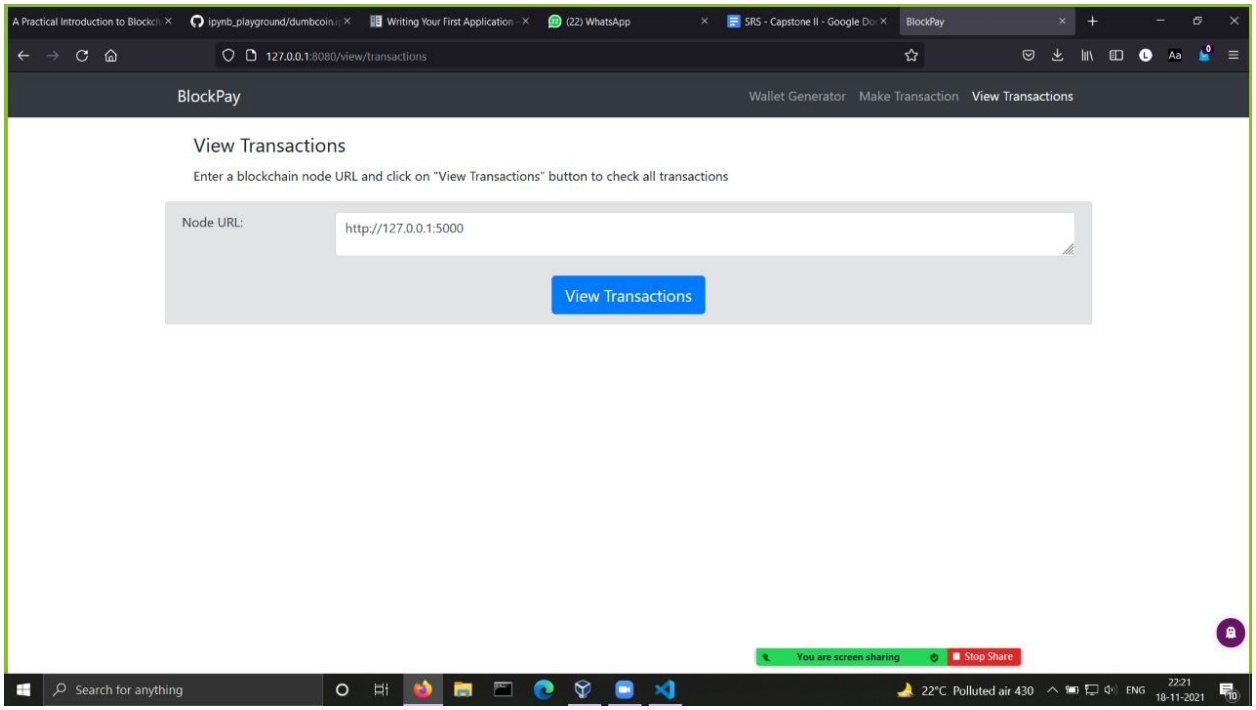
3.3 Communication Interfaces

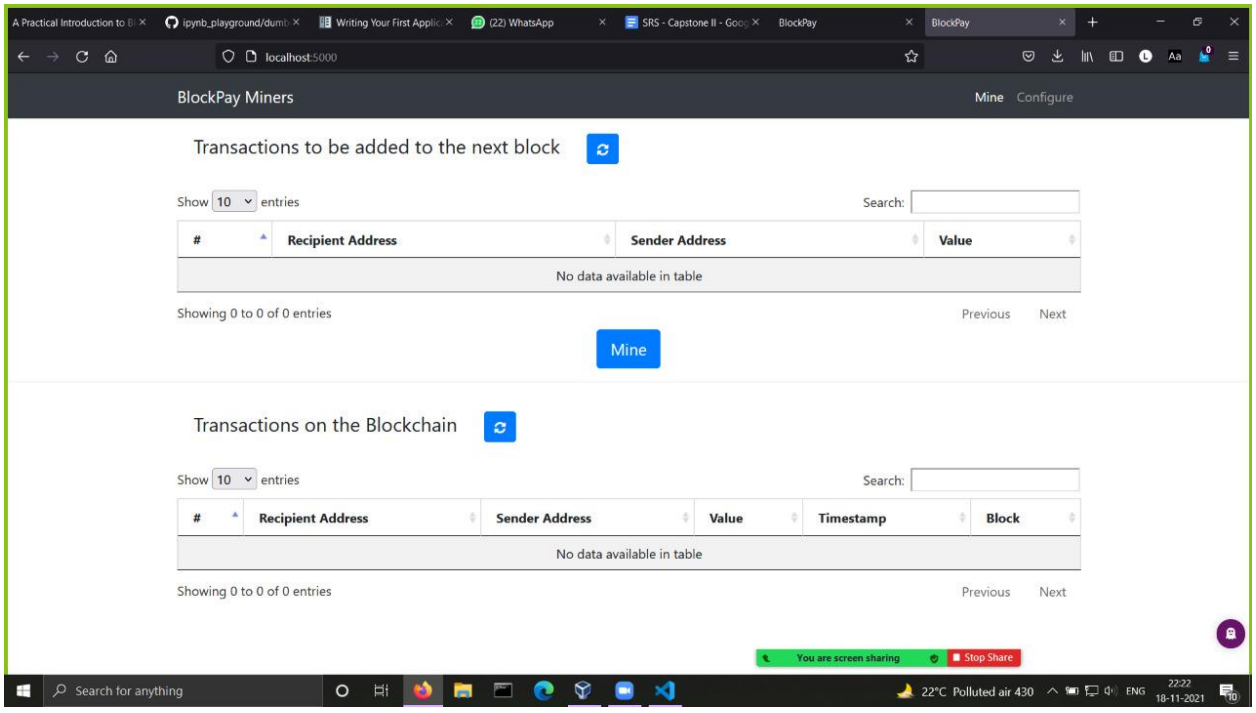
- A router is connected through the internet.
- User logs in and enters the dashboard
- Data is fetched from Blockchain.

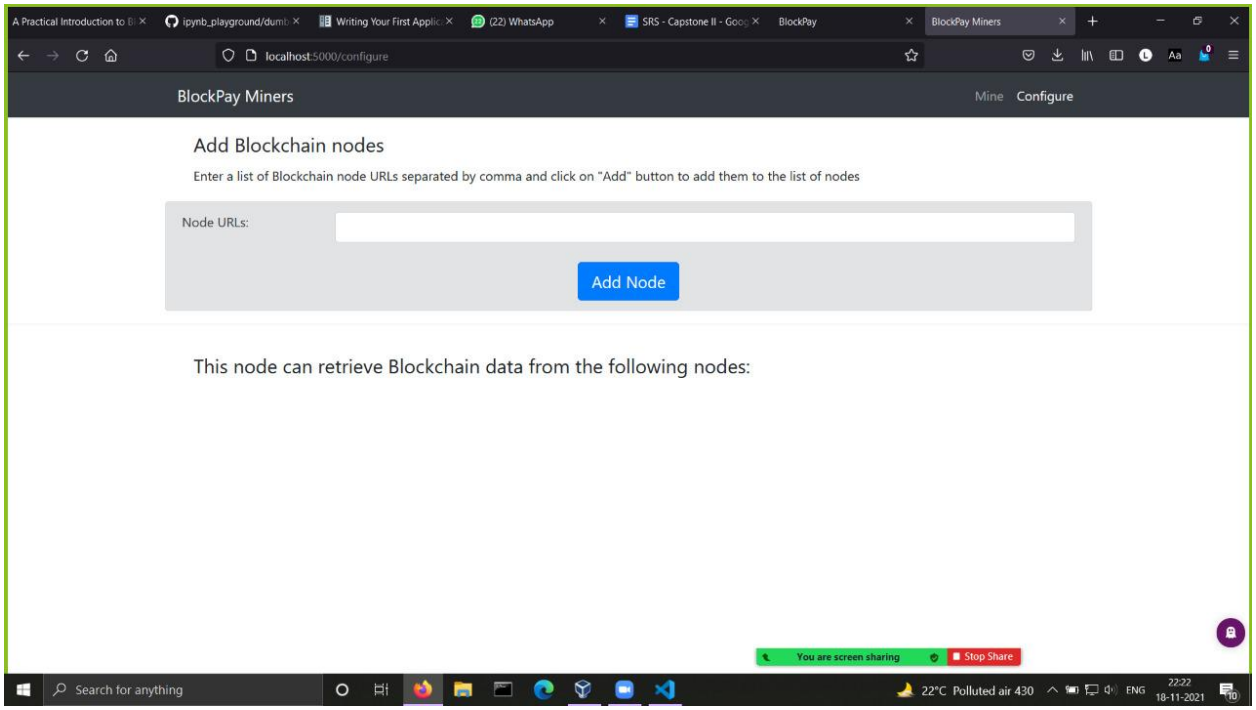
3.4 User Interface


BlockPay
Wallet Generator
Make Transaction
View Transactions
Wallet Generator
Click on the button below to generate your blockchain wallet
Generate Wallet
Public Key:
Private Key:
You are screen sharing
Stop Share
Search for anything
22°C Polluted air 430
ENG
22:21
18-11-2021


127.0.0.1:8080/make/transaction
BlockPay
Wallet Generator
Make Transaction
View Transactions
Send Coins
Enter transaction details and click on "Generate Transaction" button to generate your transaction
Sender Address:
Sender Private Key:
Recipient Address:
Amount to Send:
Generate Transaction
You are screen sharing
Stop Share
Search for anything
22°C Polluted air 430
ENG
22:21
18-11-2021


BlockPay
Wallet Generator
Make Transaction
View Transactions
View Transactions
Enter a blockchain node URL and click on "View Transactions" button to check all transactions
Node URL:
http://127.0.0.1:5000
View Transactions


BlockPay Miners
Mine
Configure
Transactions to be added to the next block
Show 10 entries
Search:
Recipient Address Sender Address Value
No data available in table
Showing 0 to 0 of 0 entries
Previous
Next
Mine
Transactions on the Blockchain
Show 10 entries
Search:
Recipient Address Sender Address Value Timestamp Block
No data available in table
Showing 0 to 0 of 0 entries
Previous
Next



4. Functional Requirements

4.1 Interface for Users

4.1.1 Create Wallet

Wallet is generated based on the RSA algorithm, it will print two keys: a public key and a private key. The public would act as the user address and private key would be used to authorise transactions.

4.1.2 Make Transactions

Users will be able to make transactions by providing their public key and private key, a sender address in the form of public key and the amount to send. This will generate a signature for the transaction. Miners would be able to see that transaction and if it is mined then it is added to the blockchain network.

4.1.3 Check transactions on the blockchain network

Users will be able to check transactions by providing a node URL. The result would be displayed in tabular form.

4.2 Interface for Nodes

4.2.1 Mining

Nodes on the network would be able to access this dashboard and mine transactions. The transactions waiting to be mined and transactions on the blockchain would be shown here.

4.2.2 Add nodes to the network

Existing nodes on the network would have the authority to add other nodes on the network by providing their URL. Then that node would be able to be a part of the network and perform all the functionalities.

8. Non-Functional Requirements

5.1 Performance Requirements:

1. The response time of a transaction should be less than 5 seconds most of the time. Response time refers to the time that the user should wait for before getting a response from the system after querying it.
2. Transactions will be updated on to the platform within 10 seconds of entering and conforming to the user and confirmation shall be provided within the next minute.

5.2 Security Requirements:

Data is decentralized!

Since our BlockPay system is based on blockchain, it provides an exceptional level of security since the technology is decentralized in nature and therefore does not rely on one central point of control. It is a digital ledger of transactions with every device having a complete copy of the data. A lack of a single authority makes the BlockPay system fairer and considerably more secure. Instead of depending on a central authority to securely transact with other users, our E-voting system utilizes innovative consensus protocols across a network of nodes, to validate votes and record them in a manner that is incorruptible. Since the data is saved on multiple devices, it is extremely secured even if one or two devices malfunction. Unfeasibly hard to hack Since the data is decentralized and distributed ledgers across peer-to-peer networks are continuously updated and kept in sync. Each ‘node’ is connected to all the other ‘nodes’ before and after it. While hackers can break into traditional networks and find all the data in a single repository and exfiltrate it or corrupt it, the blockchain makes this unfeasibly hard.

5.3 Software Quality Attributes

5.3.1 Correctness

BlockPay will perform as per the previously mentioned functional and non-functional requirements correctly and accurately.

5.3.2 Reusability

Yes, the component of our BlockPay system can be used for other block-chain applications (Dapps).

5.3.3 Portability

BlockPay is highly portable since it is an online application, therefore any device can access BlockPay using a browser and an internet connection.

5.3.4 Reliability

Since BlockPay is based on Blockchain, it is highly reliable because full copies of the blockchain ledgers are maintained by all active nodes. Thus, if one node goes offline, the ledger is still readily available to all other participants in the network and therefore lacks a single point of failure.

5.3.5 Usability

Our online app is very user-friendly, with a simple and easy user interface so that our voters may easily place and check the votes and have a satisfactory level of experience with BlockPay.