Implementation of Blockchain in Financial Sector to Improve Scalability

A Main project report submitted in the partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

by

NUKALA SRIKAR	Regd.No.19131A05G1
POKALA DIVYAGANDH	Regd.No.19131A05H7
PEELA HEMA HARSHA VARDHAN	Regd.No.19131A05H4
NITIN ANCHA	Regd.No.19131A05G0

Under the esteemed guidance of

Dr. H Parthasarathi Patra

Associate Professor

Department of Computer Science and Engineering

COLLEGE OF ENGINEERING
(AUTONOMOUS)

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING (AUTONOMOUS)

(Affiliated to JNTU-K, Kakinada)

Vishakhapatnam

2022-2023

COLLEGE OF ENGINEERING
(AUTONOMOUS)

CERTIFICATE

This is to certify that the main project report entitled "**Implementation of Blockchain in Financial Sector to Improve Scalability,** being submitted by

NUKALA SRIKAR	Regd.No.19131A05G1
POKALA DIVYAGANDH	Regd.No.19131A05H7
PEELA HEMA HARSHA VARDHAN	Regd.No.19131A05H4
NITIN ANCHA	Regd.No.19131A05G0

In partial fulfilment of the requirements for the Award of Degree of **Bachelor of Technology in Computer Science and Engineering** to the Jawaharlal Nehru Technological University, Kakinada in record of bonafide work done under my guidance and supervision during VIII semester during the academic year 2022-2023

Dr. H Parthasarathi Patra
Associate Professor
Project Guide
GVPCE (A)

Dr. D.N.D. Harini
Head of the Department
Computer Science and Engineering
GVPCE (A)

DECLARATION

We hereby declare that this main project entitled "**Implementation of Blockchain in Financial Sector to Improve Scalability**" is a bonafide work done by us and submitted to the Department of **Computer Science and Engineering, Gayatri Vidya Parishad college of engineering (Autonomous) Visakhapatnam**, in partial fulfilment for the award of the degree of B. Tech is of own and it is not submitted to any other university or has been published any time before.

Place: Visakhapatnam

Date:

N.Srikar(19131A05G1)

P.Divyagandh(19131A05H7)

P.Hema HarshaVardhan(19131A05H4)

Nitin Ancha(19131A05G0)

ACKNOWLEDGEMENT

We Would like to express our deep sense of gratitude to our esteemed institute **Gayatri Vidya Parishad College of Engineering (Autonomous)** which has provided us an opportunity to fulfil our cherished desire.

We thank **Dr. A. B. KOTESWARA RAO** principal, **Gayatri Vidya Parishad College of Engineering (Autonomous)** for extending his utmost support and cooperation in providing all the provisions for the successful completion of the project.

We consider it our privilege to express our deepest gratitude to **Dr. D.N.D. HARINI, Associate professor and Head of the Department of Computer Science and Engineering**, for her valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We are extremely thankful to **Dr. H Parthasarathi Patra, Associate Professor, Computer Science and Engineering** for giving us an opportunity to do this project and providing us support and guidance which helped us to complete the project on time.

We also thank our coordinator, **Dr. CH. SITA KUMARI, Associate Professor, Department of Computer Science and Engineering**, for the kind suggestions and guidance for the successful completion of our project work.

We also thank all the members of the staff in Computer Science and Engineering for their sustained help in our pursuits. We thank all those who contributed directly or indirectly in successfully carrying out his work.

NUKALA SRIKAR (19131A05G1)
POKALA DIVYAGANDH (19131A05H7)
PEELA HEMA HARSHA VARDHAN (19131A05H4)
NITIN ANCHA (19131A05G0)

ABSTRACT

The Blockchain is an encrypted database that stores information statistics, or in different words, it is a virtual ledger of any transactions, contracts - that needs to be independently recorded. One of the key capabilities of Blockchain is that this virtual ledger is out there throughout several masses and heaps of computers and isn't always sure to be stored in a single place. Blockchain chain has already commenced disrupting the financial offerings area, and it's this technology which underpins the virtual currency- Bitcoin transaction. The aim of the paper is to conduct research on the effect of Blockchain technology on the financial sector. There is no doubt that the world is curious to see how this promising technology will influence or shape the future of banking. Blockchain enhances safety in data storage and transmutation, avails a decentralized and transparent network infrastructure and significantly reduces the costs in operations. These remarkable attributes make Blockchain a very promising and in-demand solution even in an industry as restricted as the banking sector.

KEYWORDS:

- Encrypted database, Virtual ledger, Virtual currency, Bitcoin, Transmutation

TABLE OF CONTENTS

1. INTRODUCTION **8**
1.1 EXISTING SYSTEM **9**
1.2 PROPOSED SYSTEM **13**

2. REQUIREMENT SPECIFICATIONS **18**
2.1 HARDWARE DESCRIPTION **18**
2.2 SOFTWARE DESCRIPTION **18**

3. SYSTEM FEASIBILITY STUDY **19**
3.1 ECONOMIC FEASIBILITY **19**
3.2 TECHNICAL FEASIBILITY **19**
3.3 SOCIAL FEASIBILITY **20**

4. SYSTEM DESIGN **21**
4.1 UML DIAGRAM **21**
4.2 USE CASE DIAGRAM **22**
4.3 CLASS DIAGRAM **23**
4.4 SEQUENCE DIAGRAM **23**
4.5 COLLABORATION DIAGRAM **24**

5. SRS DOCUMENT **25**
5.1 FUNCTIONAL REQUIREMENT **26**
5.2 NON-FUNCTIONAL REQUIREMENT **26**

6. SOFTWARE REQUIREMENTS **27**
6.1 PYTHON **27**
6.2 MODULES **30**
6.2.1 TENSORFLOW **30**
6.2.2 NUMPY **30**
6.2.3 PANDAS **31**
6.2.4 MATPLOTLIB **31**
6.2.5 SCIKIT-LEARN **31**
6.2.6 OPENCV-PY **32**
6.2.7 KERAS **32**
6.2.8 H5PY **33**
6.2.9 PILLOW **33**
6.2.10 SKLEARN **33**
6.2.11 DJANGO **34**
6.2.12 PYMYSQL **34**

7. SYSTEM IMPLEMENTATION **35**
7.1 INSTALLATIONS **35**
7.2 DJANGO FRAMEWORK **42**
7.3 SOURCE CODE **44**
7.4 OUTPUT **51**

8. SYSTEM TEST **59**
8.1 UNIT TESTING **59**
8.2 INTEGRATION TESTING **59**
8.3 FUNCTIONAL TESTING **59**
8.4 SYSTEM TESTING **60**
8.5 WHITE BOX TESTING **60**
8.6 BLACK BOX TESTING **60**

9. CONCLUSION **62**

10. REFERENCES **63**

1.INTRODUCTION

A blockchain is a distributed digital ledger where transactions can be recorded and checked electronically over a network of computers in the absence of a central System. Cryptography is used to protect the data from deception or hackers and any other malicious attacks. Blockchain is being called “the new internet”, and is expected to transform businesses across various sectors.

It was invented by “Satoshi Nakamoto” in 2008. A blockchain helps to record all the transactions made so that no alterations can be made later on so as to maintain the security of the data. Today, entities maintain records in their own traditional ledgers for transactions between them. This sometimes leads to transfer or exchange of a considerable amount of data between entities, resulting in an increase in time and cost for them. It also makes the process of any asset transfers inefficient, costly and vulnerable. The duplicated shared ledger concept in blockchain technology can overcome those limitations. The use of smart contracts, an application of blockchain technology, can enhance efficiency through event-triggered mechanisms. Most credit and budgetary organizations can't do their work without various go-betweens, while their interest makes the administrations of these establishments substantially more costly. The execution of blockchain will empower pointless arbiters to be relinquished and give clients and banks less expensive administrations. The fundamental zones in which banks and other budgetary organizations will probably actualize blockchain innovation: Payment, Client Identification framework, Loans, and Credits protection [4].

1.1 EXISTING SYSTEM:

A. Working of blockchain:

Blockchain not only changes how we transfer value but could dramatically shift our systems of trade identity and governance and one of the aspects that are most interesting is how it can make these systems more transparent.

There this record book shows every transaction that has ever happened in chronological order and we each hold an exact copy of it because of the way the pages are bound together trying to go back and change past trades would be practically impossible and because we each hold a copy of it, we would know if something went wrong, this shared record book is essentially a blockchain [6]

A blockchain is a distributed, immutable ledger or record of distributed transactions. It means that it's shared and stored in multiple locations removing a single point of vulnerability and providing perceptibility across massive participants. Here, Immutable tells that it's nearly impossible to go back and change the past records once they have been agreed by and attached using cryptography.

Even as a logo-new transaction or an edit to a current transaction is available into a blockchain, usually a majority of the nodes inside a blockchain implementation should execute algorithms to assess and verify the information of the user blockchain block that is proposed [8]. If a majority of the nodes come to consent that the facts and signature are valid, the brand-new block is common and is introduced to the chain of transactions. If a majority does not approve of the addition or modification of the ledger, then it isn't always delivered to the chain. The distributed consent version is what lets in blockchain to run as an allotted ledger without the need for a few important, unifying authority announcing which transactions are legitimate and which ones aren't legitimate [2]

With blockchain innovation, each page in a record of exchanges structures a square. That square affects the following square through cryptographic hashing. At the end of the day, when a square is finished, it makes an extraordinary secure code, which integrates with the following page or square, making a chain of squares, or blockchain [7]. It takes trust in the information before a square can be added to the chain; a couple of things need to occur. Initially, a cryptographic riddle must be comprehended, in this manner making the block. The personal computer that illuminates the riddle shares the answer for the majority of different Personal computers on the system; this is called confirmation-of-work. The system will at that point confirm this evidence of-work and, if right, the square will be added to the chain. The blend of this perplexing math confounds, and confirmation of numerous Personal computers guarantees that we can confide in every single square on the chain. Since the

system does the trust working for us, we presently have the chance to collaborate legitimately with information in genuine time [2].

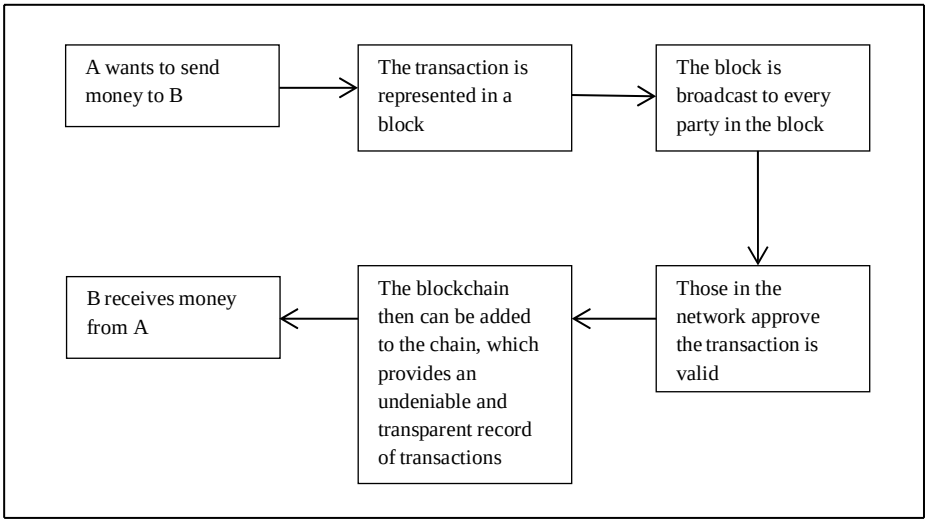


Fig 1.1: Working of Blockchain

The blockchain technology is categorized in two parts:

1. Public Blockchain- A public blockchain network or permission-less blockchain network is completely open ended and anyone inclined to participate in this kind of network can take part without anyone's permission . This is the important and only difference between public and private blockchain networks. Anyone can take part in the unauthorized network, perform the consent protocol and keep the shared open public ledger.

2. Private Blockchain- A non-public blockchain community calls for an invitation to participate in the community. The invitation must be established either via network starter or by using the regulations/conditions placed by the community starter. Permission blockchain community places a limit to the access of participants and allows most effective the kind of participant that is required in the network.

B. Blockchain is NOT Crypto-Currency:

Blockchain is the platform which brings crypto currencies into play. The blockchain is the era; this serves because of the distributed ledger that paperwork the network. This technology creates the way for transacting and allows moving of price and records.

Crypto-currencies are the tokens used within those networks to ship fees and pay for these transactions. We can see them as a tool on blockchain technology, in a few instances serving as an aid or application feature. Other instances may be used to digitize the cost of an asset.

Blockchains serve as the idea generation, wherein crypto currencies are a part of the ecosystem [10]. They pass hand in hand, and crypto is regularly important to transact on a blockchain. But without the blockchain, we might not have a means for those transactions to be recorded and transferred.

S.No.	Blockchain	Crypto-currency
1.	Blockchain does not require crypto-currency	A crypto-currency is an application of crypto-technology, allowing the transfer of cash or value via transactions recorded on a blockchain.
2.	The program can be constructed to handle a varying set of rules and configurations	There are various existing crypto-currencies, most famous is bitcoin.
3.	Related technologies such as smart contracts can seriously improve process efficiency transparency, reliability and reduce risk.	Specific to crypto-currencies a key benefit includes preventing extra spending.

TABLE 1.1: DIFFERENCE BETWEEN BLOCKCHAIN AND CRYPTO-CURRENCY

Blockchain is certainly one of the most mentioned technologies inside the financial region these days. The shift from a centralized technical infrastructure to distributed, atmosphere-enabling systems is building the rules for a new business model in bills, virtual banking, and financial transaction technology. Monetary services enterprise is presently the chief in experimenting with the generation. Blockchain holds the ability for all the individuals in a commercial enterprise community to percentage a gadget of records so as to offer consensus, provenances, immutability, and finality across the switch of belongings in the business network [11]. In spite of the efforts to reduce the complexities and boost the interconnectedness of participants' transaction information, the commercial enterprise network is still typically replacing statistics or messages between them to conclude transactions. Blockchain technology has the capability to deal with positive limitations of the cutting-edge approaches by means of modelling, streamlining and simplifying the conventional design of the monetary enterprise infrastructure with a shared material of common place information [10]

The gain brought through blockchain technology may be extensively categorized into fee-saving, efficiency, and transparency.

(i).Cost saving:

a. Fraud Prevention: As Blockchain technology is constructed on the concept of sharing statistics throughout events and consensus throughout transactions; it saves on reconciliation price between banks and forestall losses due to documentary frauds.

b. Saves costs on forex volatility: Blockchain generation is used in go border bills can assist the consumers and banks to take gain of the foreign exchange marketplace to get the first-class deal transparently from the marketplace players[8].

c. Save costs over delayed settlements: In case of a distributed fee community, Blockchain technology guarantees the transaction agreement facts are also strategies simultaneously along with the feed message.

(ii). Efficiency:

a. Resilience through redundancy: Being a dispensed structure by using layout, Blockchain technology permits the network to be operated by way of all permission nodes within the atmosphere. All the essential contributors of the price surroundings- banks, monetary institutions shall efficiently become the participating nodes inside the Blockchain generation network [12].

b. Reduced time for processing: In the Blockchain era, the transaction is discovered to all the approving nodes, right away. As a result, Blockchain generation can assist in enhancing the rate of processing transactions by way of discount in choice making a time across the agencies resulting in decreased prices of processing and stronger transparency of choices to all the participating nodes.

c. Faster settlements: Blockchain also can assist to deal with recognizing your customer and identity management challenges as a variety of the records to show identification is already in virtual shape and Blockchain technology should immediately verify.

(iii) Transparency:

a. Immutable Transactions: Maintaining an immutable report of transaction occasions in a chronological order being a major pillar of its structure, Blockchain era guarantees a great deal of desired attributes to banking and financial transactions which includes immutability and finality[13].

1.2 PROPOSED SYSTEM:

Blockchain is an era that strengthens an awesome manner to have huge-undertaking implications so that it will now not genuinely transform financial offerings, but many other commercial enterprises and industries. Billions of humans and groups are served and trillions of bucks are moved around the previous worldwide financial device each and every day. Nevertheless, closely reliant and dependent on paper, despite the fact that dressed up with a virtual appearance, there are various problems with this era. Motive brought price and delays as well as make it much less complicated for crime and fraud to cripple it. In spite of the monetary employer's resistance to trade, blockchain and its anticipated benefits make it worthwhile. Blockchain, not like traditional structures, is dynamic enough to come to be a pacesetter in implementation in a chargeable market situation. In a blockchain, the best advantage it guarantees is that every celebration has a report that is maintained in a ledger to be handed to everyone. It is extensively surpassed between special users developing a shared database that is replicated to those users and who can get right of entry to its simplest when they have the get admission to the right for it

A. Things blockchain can do for the financial sector

a). On-chain settlement:

Blockchain is a pioneering technology based on a distributed ledger. It has a capability to lower the fraud rates in the international bank system and it is also capable of providing On-chain settlement. Blockchain can be used in the financial sector specifically in the banking sector providing a platform for banks to reduce fraud as well as On-chain settlement to the users that also helps in reducing the processing time. DLT is capable of providing a platform on Ethereum blockchain. The user will not have to rely on the centralized system for the confirmation of the transaction.

b.) Low transfer fees:

The user will have a transparent cost model for sending a certain amount of money for overseas transactions. The traditional system has a number of intermediates which results in the high transfer fees. The banks have to rely on the centralized system for verifying the transactions. The process is complicated and takes a lot of time to verify the transactions. The platform proposed will have a transparent cost model for sending the money across borders that will provide ease to the user and they have to pay only the negligible cost for sending money.

c). Round the clock Availability:

The platform is accessible anytime from anywhere from the world. The nodes in the distributed network will verify the transaction and if more than 75% verifies the transactions, the process will be completed and the user on

the side will receive the funds. The nodes will have a certain amount of price to verify and block creation

d). Transparency:

The bank system presently changes the conversion rate without informing the users which results in high transaction cost. The platform proposed will have a transparent conversion rate that will be visible to the user for sending the money overseas with ease. This will also allow the user to seek in his ledger and see the transaction history and conversion rate.

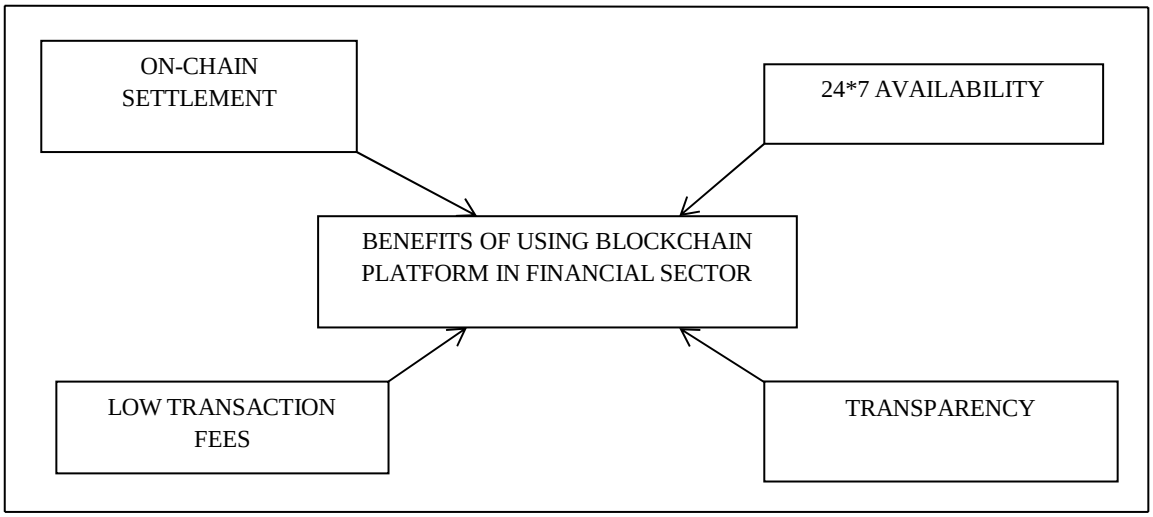


Fig 1.2: Benefits of Blockchain

B. Proposed working of platform :

Every bank registered on this blockchain platform will have to update the ledger by uploading customer data in encrypted form which allows security to the user's data. By registering to this platform every bank will have the same ledger for the customer data and transaction history. DLT will provide a full transparency model to the user to send money overseas along with (24*7) availability. This will also reduce the time for the transaction to process as every node present in the network will verify the transaction and store the transaction history in the blockchain database. This distributed ledger will eliminate the double spending problem present in the centralized system. This platform also provides on-chain settlement with the negligible cost for a transaction.

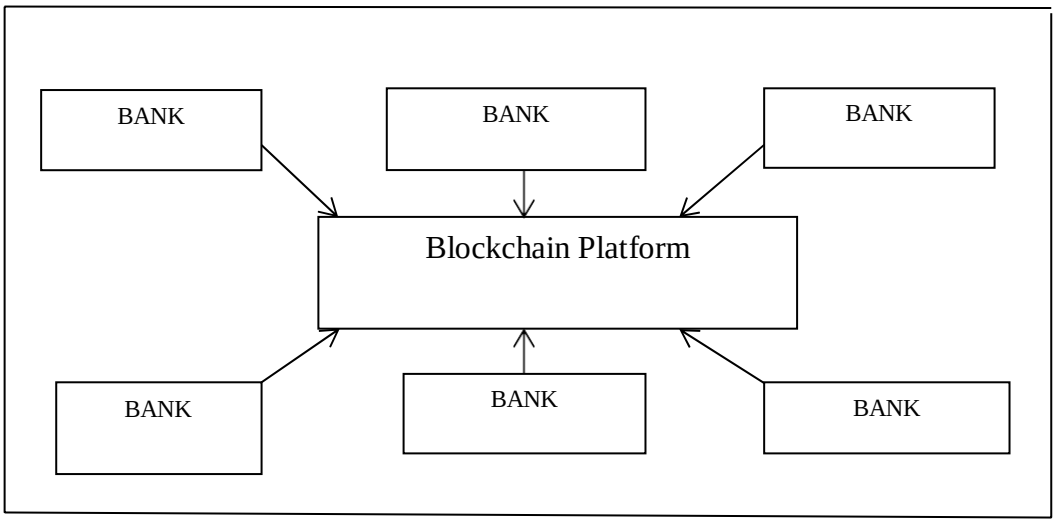


Fig 1.3: Banks on the blockchain platform

If a user wants to send money to the other user, the platform will provide some features over the centralized system. The banks registered within the blockchain platform will interact with each other in the ledger. The user can only send money to the banks registered on the platform that will carry all the information of other users in the distributed ledger.

The user sending the money via this platform will have transparency for the transaction made with the negligible fees. The user on the receiver side will get the funds after more than 75% nodes verify the transaction. However, the availability of the platform is easily accessible and they don't have to visit banks to send money overseas. The transaction made has to undergo a consensus mechanism that will be carried out by nodes in the network.

The platform will have a certain advantage over the centralized system as the transactions made will be immutable which reduces the rate of fraud conversions.

C. Benefit of the proposed platform :

The benefit of this platform is that it will have only a single database for the user information which reduces the process of KYC of the user for every bank. The timeline of the transaction will be reduced as there will be an automation process of transferring money which will benefit the user sending money in any corner of the world.

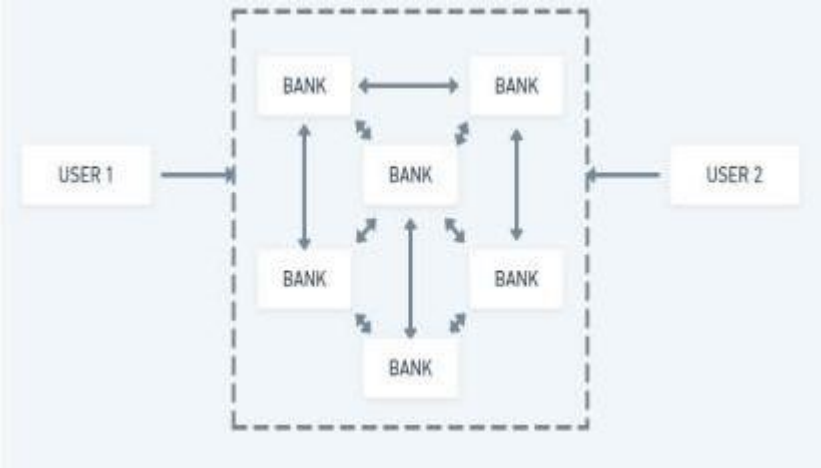


Fig 1.4: Blockchain System

SHA ALGORITHM:

SHA algorithm is used in the encryption of data and crypto technologies. The Secure Hash Algorithms (SHA) is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS). SHA stands for Secure Hash Algorithm, which is a modified version of MD5 used for hashing data and certificates**,** shortening the input data into a smaller form that cannot be understood by using bitwise operations, modular additions, and compression functions

Block chain uses Sha algorithm to create hashing functions for the blocks, based on the previous inputs and the timestamp values. The block is divided into 2 parts the block chain header and the data part. The blockchain header consists of

- Version number
- Timestamp
- Previous blockchain address
- Nonce
- Merkle root
- Hash value of the current block

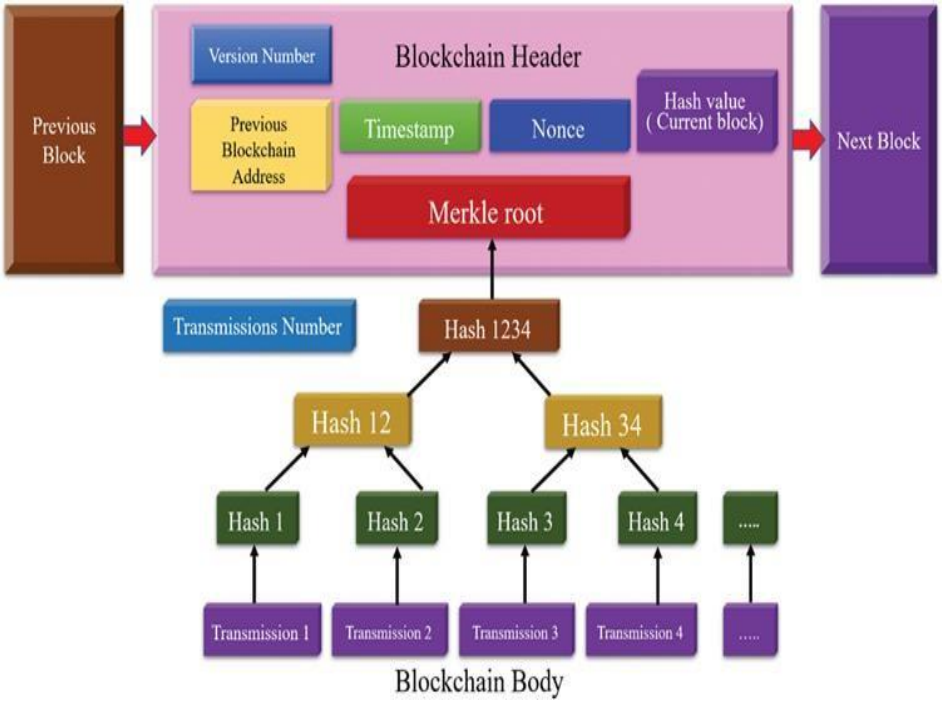


Fig 1.5: Blockchain header

S.No.	**Current Pain Points**	**How Blockchain can Help**
1.	Laboring Documentation	Automated Documentation
2.	Time consuming process	Real time settlement of transaction
3.	Lack of mechanism to track throughout the process	Real time-tracking of transaction
4.	Potential of Fraud	Fraud Proof

TABLE 1.2: EASY PAYMENT THROUGH BLOCKCHAIN

2. REQUIREMENTS SPECIFICATIONS:

2.1 HARD REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Monitor : 15 VGA Color.
- Ram : 512 MB.

2.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 8Professional and Above
- Coding Language : python

3. SYSTEM FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM DESIGN

4.1 UML DIAGRAMS:

UML stands for Unified Modelling Language. UML is a standardized general- purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have provensuccessful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they candevelop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of the OO tools market.
6. Support higher level development concepts such as collaborations, frameworks,patterns and components.
7. Integrate best practices.

4.2 USE CASE DIAGRAM:

A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

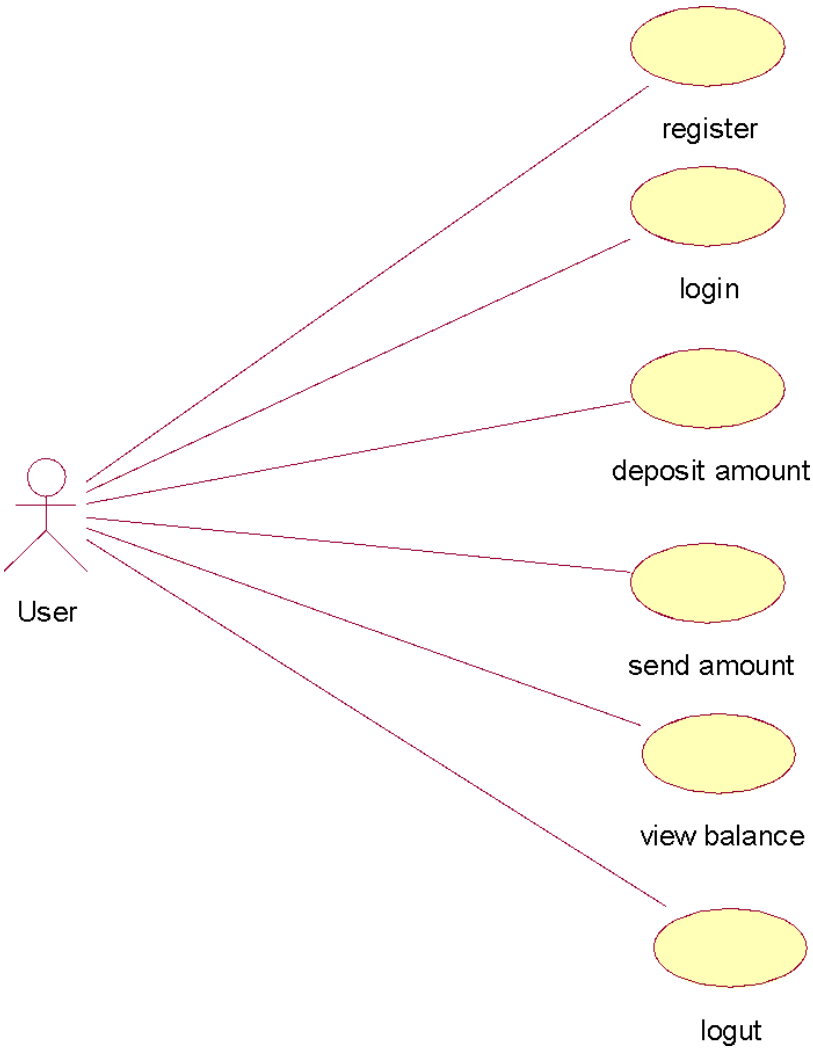


Fig:4.1:Use Case Diagram

4.3 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

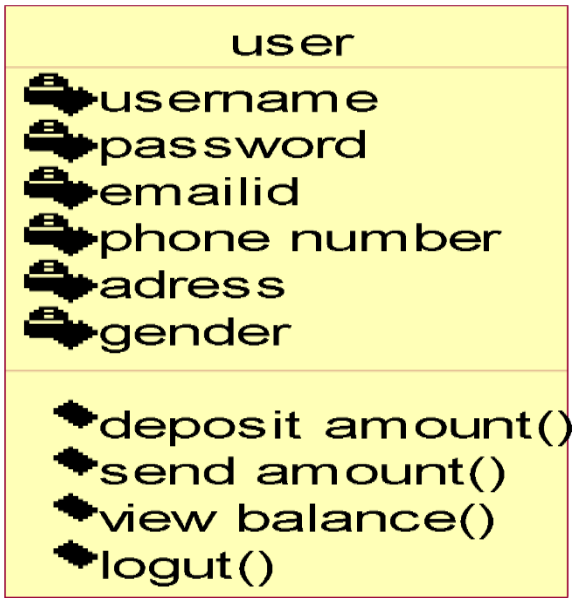


Fig:4.2:Class Diagram

4.4 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

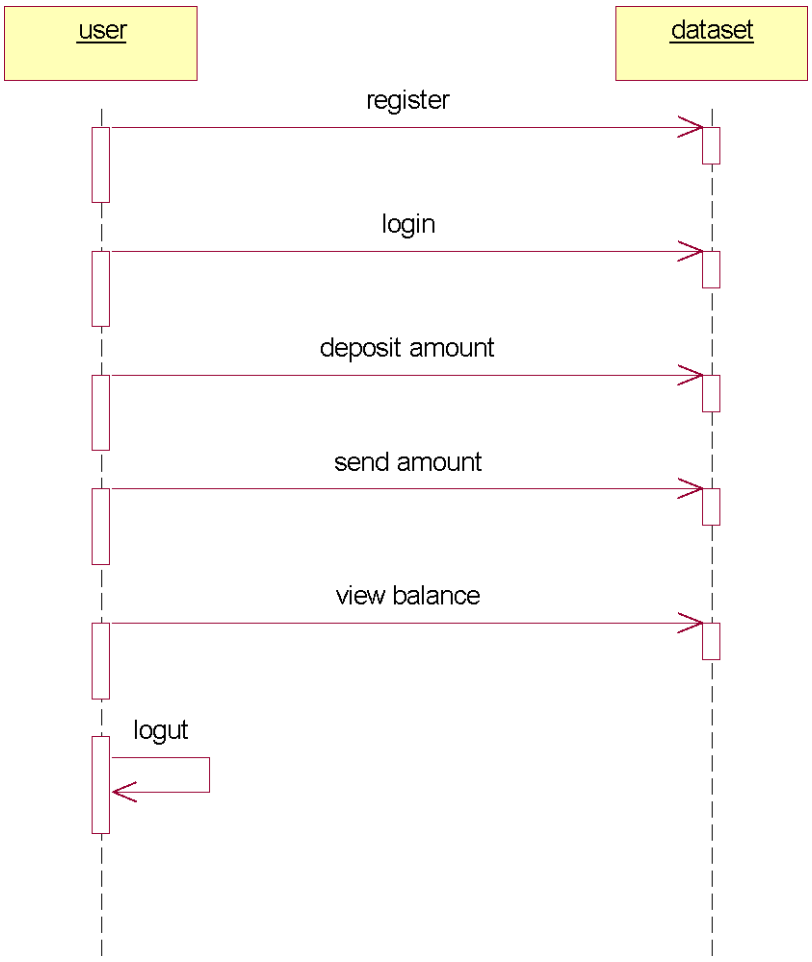


Fig:4.3:Sequence Diagram

4.5 COLLABORATION DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

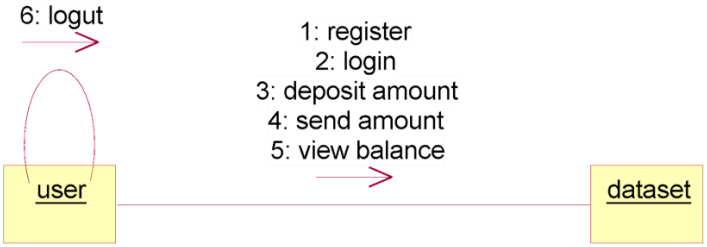


Fig:4.4:Collabration Diagram

5. SRS DOCUMENT

SRS is a document created by a system analyst after the requirements are collected from various stakeholders. SRS defines how the intended software will Interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing,Security, Quality, Limitations etc. The requirements received from clients are written in natural language. It is the responsibility of the system analyst to document the requirements in technical language so that they can be comprehended and useful by the software development team. The introduction of the software requirement specification states the goals and objectives of the software, describing it in the context of the computer-base system. The SRS includes an information description, functional description, behavioral description, validation criteria. The purpose of this document is to present the software requirements in a precise and easily understood manner. This document provides the functional, performance, design and verification requirements of the software to be developed. After requirement specifications are developed, the requirements mentioned in this document are validated. Users might ask for illegal, impractical solutions or experts may interpret the requirements incorrectly. This results in a huge increase in cost if not nipped in the bud.

5.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a system or its component. A function can be described as set of inputs, the behavior, and outputs. It also depends upon the type of software, expected users and the type of system where the software is used. Functional requirements of our project are:

User should be able to implement:

Sign up: should be able to register the details of user into the server or database.

Deposit Amount: link to add amount to his account

View Balance: link to view his balance, link again to view his current balance

Send Amount: link to get below screen

5.2 NON-FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security and Portability. Non-functional requirements are called qualities of a system, there are as follows:

- Performance-The average response time of the system is less.
- Reliability - The system is highly reliable.
- Operability - The interface of the system will be consistent.
- Efficiency - Once user has learned about the system through his interaction, he can perform the task easily
- Understandability-Because of user friendly interfaces, it is more understandable to the users

6. SOFTWARE ENVIRONMENT

6.1 Python

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and the indentation requirement of the language makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, and Uber etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter and PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python:-
Let's see how Python dominates over other languages.

1. Extensive Libraries
Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more.* So, we don't have to write the complete code for that manually.

2. Extensible
As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable
Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

4. Improved Productivity
The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. It provides increased readability and simplicity to the implementation.

5. IOT Opportunities
Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **'Hello World'**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python; they have a hard time adjusting to other more verbose languages like Java.

7. Readable
Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** These further aids the readability of the code.

8. Object-Oriented
This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

9. Free and Open-Source
Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and

even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now on the advantages of Python? Mention in the comment section.

Purpose:-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

6.2 Modules Used in Project:-

6.2.1 TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.
TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

6.2.2 NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

6.2.3 Pandas

Pandas' module is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data mugging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

6.2.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can

generate plots, histograms, power spectra, bar charts, error charts and scatter plots etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object-oriented interface or viaa set of functions familiar to MATLAB users.

6.2.5 Scikit – learn

Scikit-learn provide a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

6.2.6 OPENCV-PYTHON

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries,

such as NumPy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e., whatever operations one cando in NumPy can be combined with OpenCV.
This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

6.2.7 KERAS

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

Up until version 2.3, Keras supported multiple backend, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML. As of version 2.4, only TensorFlow is supported. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet is also the author of the Xception deep neural network model.

6.2.8 H5PY

The h5py package is a Pythonic interface to the HDF5 binary data format. It lets you store huge amounts of numerical data, and easily manipulate that data from NumPy. For example, you can slice into multi-terabyte datasets stored on disk, as if they were real NumPy arrays. Thousands of datasets can be stored in a single file, categorized and tagged however you want.

H5py uses straightforward NumPy and Python metaphors, like dictionary and NumPy array syntax. For example, you can iterate over datasets in a file, or check out the .shape or .dtype attributes of datasets. You don't need to know anything special about HDF5 to get started.

In addition to the easy-to-use high-level interface, h5py rests on an object-oriented Python wrapping of the HDF5 C API. Almost anything you can do from C in HDF5, you can do from h5py.

6.2.9 PILLOW

Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. As of 2019, Pillow development

is supported by Tidelift. The Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

6.2.10 SKLEARN

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

6.2.11 SWARM PACKAGE PY

PySwarms is an extensible research toolkit for particle swarm optimization (PSO) in Python. It is intended for swarm intelligence researchers, practitioners, and students who prefer a high-level declarative interface for implementing PSO in their problems. PySwarms enables basic optimization with PSO and interaction with swarm optimizations. The pyswarm package is a gradient-free, evolutionary optimization package for python that supports constraints and is both Python2 and Python3 compatible.

6.2.12 DJANGO

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It is a Python package that is specifically intended for use in a Django project. An application may use common Django conventions, such as having models, tests, urls, and views sub modules.

6.2.13 PYMYSQL

PyMySQL is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2. 0 and contains a pure-Python MySQL client library. The goal of PyMySQL is to be a drop-in replacement for MySQLdb. Besides reading data from MySQL databases, we can also use PyMySQL to perform other CRUD (Create, Read, Update, and Delete) functions to manage the database. For example, we can execute a query that inserts a data record into the customer's table.

7.SYSTEM IMPLEMENTATIONS:

7.1 INSTALLATIONS

Python, a versatile programming language, doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.
The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac:

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest versionof Python is version 3.7.4 or in other words, it is Python 3.
Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before starting the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. For a Windows 64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7device or to install Python 3. Download the Python Cheat Sheet here.The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to helpunderstand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

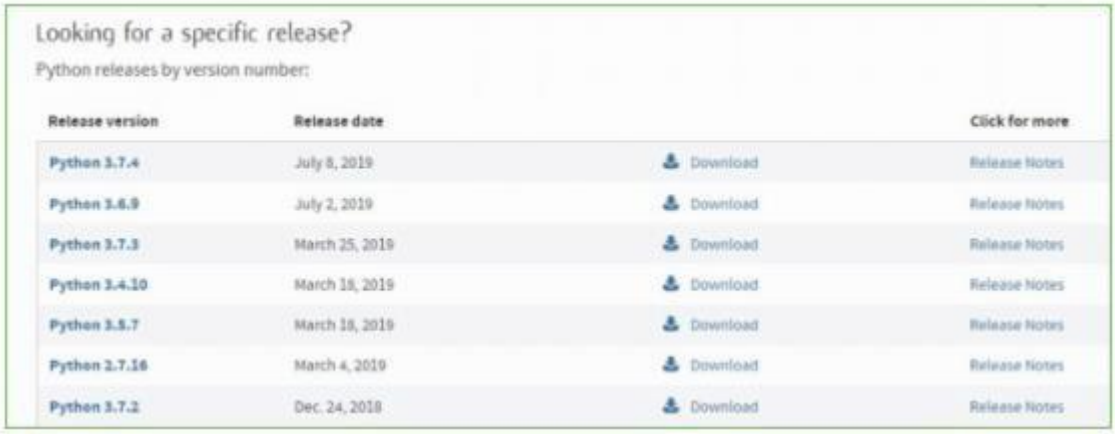
Fig 7..1: Python.org website

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Fig 7.2: Python for Windows

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Fig 7.3: Files in python

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b243f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c3cf6d9ec0b9abe83184a4f729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76de6db3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c3c1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab38d1f8841879fda9413357413 9d8	6741626	SIG
Windows x86 executable installer	Windows		33cc802942a54446a3d64514763947 89	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig 7.4 Windows x86Web-based Installer

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation **Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

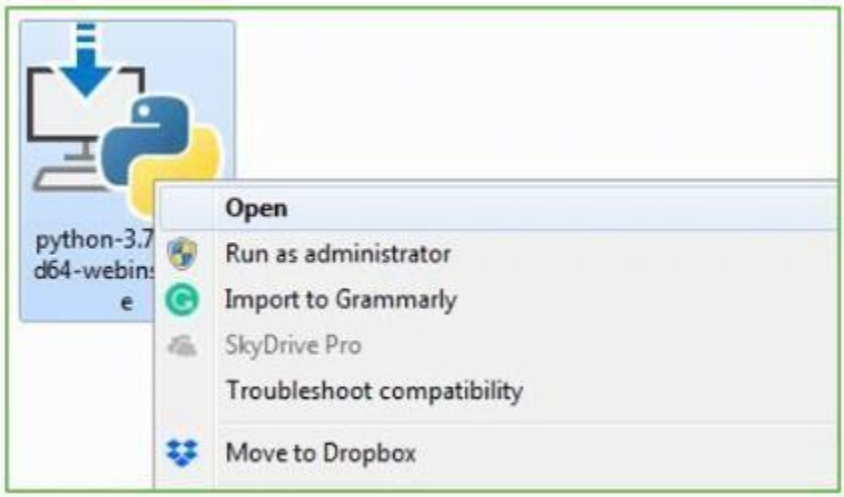


Fig 7.5: Installation steps

Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.

Fig 7.6: Installations

Step 3: Click on Install NOW After the installation is successful.

Fig 7.7: Setup

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.
Note: The installation process might take a couple of minutes.

Verify the Python Installation
Step 1: Click on Start
Step 2: In the Windows Run Command, type “cod”.

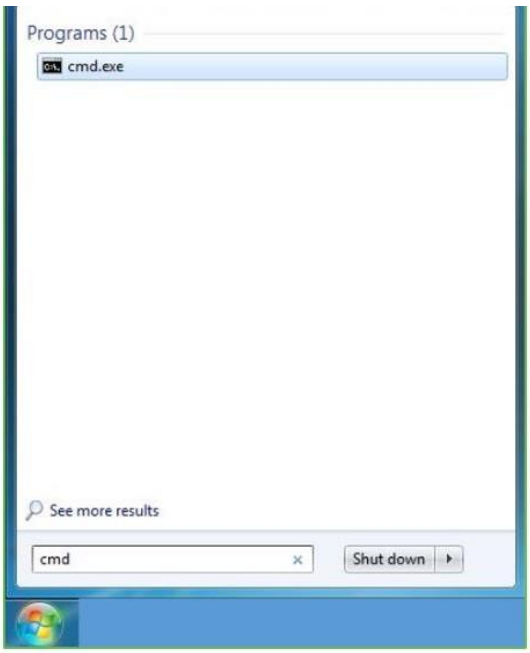


Fig 7.8: Command prompt

Step 3: Open the Command prompt option.
Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.

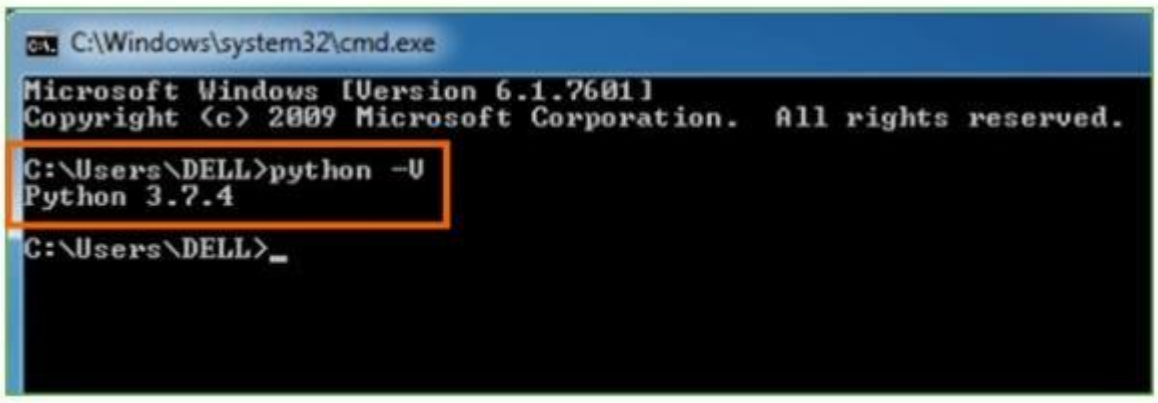


Fig 7.9: Commands

Step 5: You will get the answer as 3.7.4
Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works
Step 1: Click on Start
Step 2: In the Windows Run command, type “python idle”.

Fig 7.10: IDLE

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program
Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**

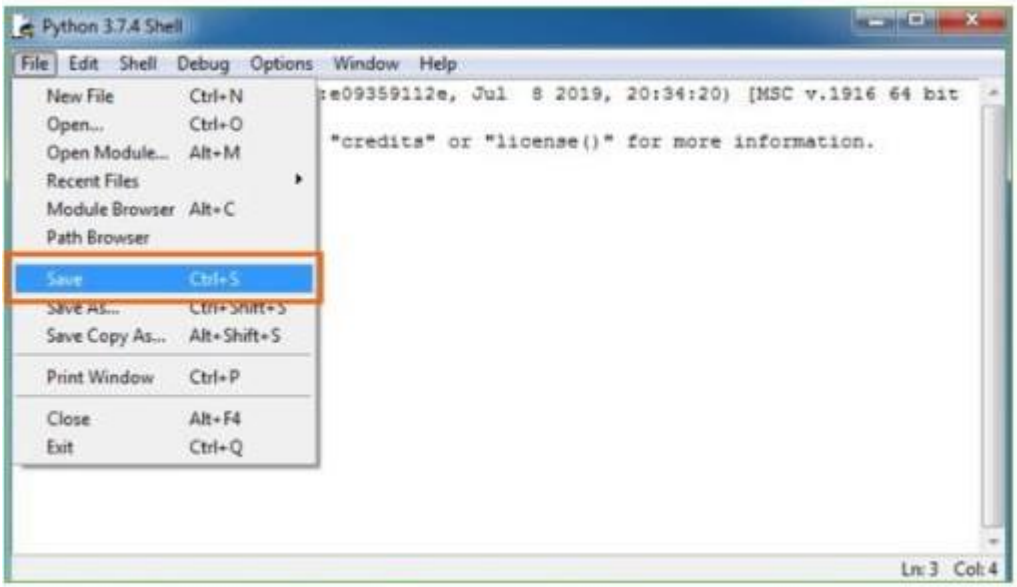


Fig 7.11: Shell

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.
Step 6: Now for e.g. **enter print**

7.2 DJANGO FRAMEWORK:

Django was developed in a fast-paced newsroom environment, it was designed to make common web development tasks fast and easy. Django helps to create a database driven web app

Installing Django

Step-1: Open the command prompt and firstly install python

Sterp-2: Run the Django command-line utilities to create the database tables automatically

Step-3: $ Python manage.py make migrations

$ python manage.py migrate

The make migrations command looks at all your available models and creates migrations for whichever tables don't already exist. Migrate runs the migrations and creates tables in your database, as well as optionally providing much richer schema control.

Django Provides Free Api to link with the database and the API is called python API. The API is created on the fly, no code generation necessary

Designing a model in Django:

Although you can use Django without a database, it comes with an object-relational mapper in which you describe your database layout in Python code.

The data-model syntax offers many rich ways of representing your models – so far; it's been solving many years' worth of database-schema problems.

Example Code:

```
from django.db import models
class Reporter(models.Model):
    full_name = models.CharField(max_length=70)
    def__str_(self):
        return self.full_name
class Article(models.Model):
    pub_date = models.DateField()
    headline = models.CharField(max_length=200)
    content = models.TextField()
    reporter = models.ForeignKey(Reporter, on_delete=models.CASCADE)
    def__str_(self):
        return self.headline
```

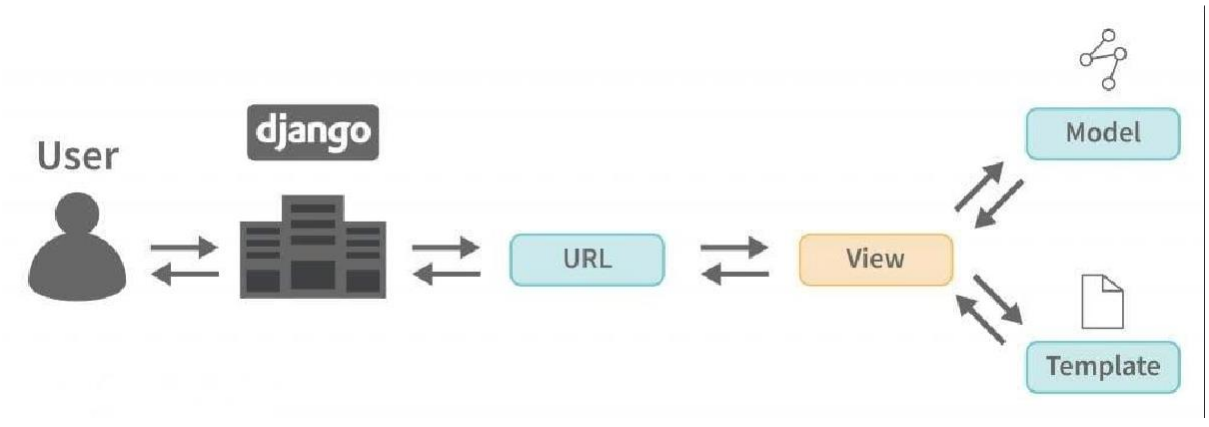


Fig 7.12 : Working of DJANGO

7.3 Source code:

BankContract.sol

```
pragma solidity >= 0.8.11 <= 0.8.11;

contract BankContract {
    string public users;
    string public bankaccount;

    constructor() {
        users = "empty";
        bankaccount = "empty";
    }

    function addUsers(string memory u) public {
        users = u;
    }

    function getUsers() public view returns (string memory) {
        return users;
    }

    function bankAccount(string memory ba) public {
        bankaccount = ba;
    }

    function getBankAccount() public view returns (string memory) {
        return bankaccount;
    }
}
```

Fig:7.13:BankContract Json File

Migrations.sol

```
// SPDX-License-Identifier: MIT
pragma solidity >= 0.8.11 <= 0.8.11;

contract Migrations {
  address public owner = msg.sender;
  uint public last_completed_migration;

  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
    );
    _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

Fig:7.14: Migrations Sol File

manage.py

```python
#!/usr/bin/env python
import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Bank.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

Fig:7.15:Management File

Views.py:

```python
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponse
import os
import json
from web3 import Web3, HTTPProvider
from datetime import datetime


global details
= '' global
user_id

def readDetails(contract_type):
    global details
    details = ""
    print(contract_type+"======================")
    blockchain_address = 'http://127.0.0.1:9545' #Blokchain connection IP
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'BankContract.json' #bank contract code
    deployed_contract_address = '0xE8BD0f2334E60f470FBD569a0318461F775163e9'
#hash address to access bank contract
    with open(compiled_contract_path) as file:
        contract_json = json.load(file) # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary
to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address,
abi=contract_abi) #now calling contract to access data
    if contract_type == 'adduser':
        details = contract.functions.getUsers().call()
    if contract_type == 'account':
        details = contract.functions.getBankAccount().call()
```

```
    if len(details) > 0:
        if 'empty' in details:
            details = details[5:len(details)]
    print(details)

def saveDataBlockChain(currentData, contract_type):
    global details
    global contract
    details = ""
    blockchain_address = 'http://127.0.0.1:9545'
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    compiled_contract_path = 'BankContract.json' #bank contract file
    deployed_contract_address = '0xE8BD0f2334E60f470FBD569a0318461F775163e9'
#bank contract address
    with open(compiled_contract_path) as file:
        contract_json = json.load(file) # load contract info as JSON
        contract_abi = contract_json['abi']  # fetch contract's abi - necessary
to call its functions
    file.close()
    contract = web3.eth.contract(address=deployed_contract_address,
abi=contract_abi)
    readDetails(contract_type)
    if contract_type == 'adduser':
        details+=currentData
        msg = contract.functions.addUsers(details).transact()
        tx_receipt = web3.eth.waitForTransactionReceipt(msg)
    if contract_type == 'account':
        details+=currentData
        msg = contract.functions.bankAccount(details).transact()
        tx_receipt = web3.eth.waitForTransactionReceipt(msg)

def SendAmountAction(request):
    global details
    if request.method == 'POST':
        sender = request.POST.get('t1', False)
        balance = request.POST.get('t2', False)
        receiver = request.POST.get('t3', False)
        amount = request.POST.get('t4', False)
        amount = float(amount)
        balance = float(balance)
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        if balance > amount:
            data = sender+"#"+str(amount)+"#"+str(timestamp)+"#Sent To
"+receiver+"\n"
            saveDataBlockChain(data,"account")
            data = receiver+"#"+str(amount)+"#"+str(timestamp)+"#Received From
"+sender+"\n"
            saveDataBlockChain(data,"account")
            context= {'msg':'Money sent to '+receiver}
            return render(request, 'UserScreen.html', context)
        else:
            context= {'msg':'insufficient balance'}
            return render(request, 'UserScreen.html', context)



def SendAmount(request):
    if request.method == 'GET':
        global user_id
        readDetails("account")
        deposit = 0
        wd = 0
        rows = details.split("\n")
```

```
        for i in range(len(rows)-1):
            arr = rows[i].split("#")
            if arr[0] == user_id:
                if arr[3] == 'Self Deposit' or "Received From " in arr[3]:
                    deposit = deposit + float(arr[1])
                else:
                    wd = wd + float(arr[1])
        deposit = deposit - wd
        output = '<tr><td><font size="3" color="black">Username</td><td><input
type="text" name="t1" size="20" value='+user_id+' readonly/></td></tr>'
        output += '<tr><td><font size="3"
color="black">Available Balance</td><td><input type="text" name="t2"
size="20" value='+str(deposit)+' readonly/></td></tr>'
        output += '<tr><td><font size="3"
color="black">Choose Receiver Name</td><td><select name="t3">'
        readDetails("adduser")
        rows = details.split("\n")
        for i in range(len(rows)-1):
            arr = rows[i].split("#")
            if arr[0] == "adduser":
                if arr[1] != user_id:
                    output += '<option value="'+arr[1]+'">'+arr[1]+'</option>'
        output += "</select></td></tr>"
        context= {'msg1':output}
        return render(request, 'SendAmount.html', context)


def ViewBalance(request):
    if request.method == 'GET':
        global user_id
        output = '<table border=1 align=center width=100%>'
        font = '<font size="3" color="black">'
        arr = ['Username','Amount','Transaction Date',"Transaction Status"]
        output += "<tr>"
        for i in range(len(arr)):
            output += "<th>"+font+arr[i]+"</th>"
        readDetails("account")
        rows = details.split("\n")
        deposit = 0
        wd = 0
        for i in range(len(rows)-1):
            arr = rows[i].split("#")
            if arr[0] == user_id:
                output += "<tr><td>"+font+arr[0]+"</td>"
                output += "<td>"+font+arr[1]+"</td>"
                output += "<td>"+font+arr[2]+"</td>"
                output += "<td>"+font+arr[3]+"</td>"
                if arr[3] == 'Self Deposit' or "Received From " in arr[3]:
                    deposit = deposit + float(arr[1])
                else:
                    wd = wd + float(arr[1])
        deposit = deposit - wd
        output += "<tr><td>"+font+"Current Balance : "+str(deposit)+"</td>"
        context= {'msg':output}
        return render(request, 'ViewBalance.html', context)


def LoginAction(request):
    global  details
    global user_id
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        status = 'none'
```

```
            readDetails("adduser")
            rows = details.split("\n")
            for i in range(len(rows)-1):
                arr = rows[i].split("#")
                if arr[0] == "adduser":
                    if arr[1] == username and arr[2] == password:
                        status = 'success'
                        user_id = username
                        break
            if status == 'success':
                file = open('session.txt','w')
                file.write(username)
                file.close()
                context= {'msg':"Welcome "+username}
                return render(request, 'UserScreen.html', context)
            else:
                context= {'msg':'Invalid login details'}
                return render(request, 'Login.html', context)


    def DepositAction(request):
        global details
        if request.method == 'POST':
            username = request.POST.get('t1', False)
            amount = request.POST.get('t2', False)
            timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            data = username+"#"+amount+"#"+str(timestamp)+"#Self Deposit\n"
            saveDataBlockChain(data,"account")
            context= {'msg':'Money added to user account '+username}
            return render(request, 'Deposit.html', context)

    def SignupAction(request):
        global details
        if request.method == 'POST':
            username = request.POST.get('t1', False)
            password = request.POST.get('t2', False)
            contact = request.POST.get('t3', False)
            email = request.POST.get('t4', False)
            address = request.POST.get('t5', False)
            gender = request.POST.get('t6', False)
            record = 'none'
            readDetails("adduser")
            rows = details.split("\n")
            for i in range(len(rows)-1):
                arr = rows[i].split("#")
                if arr[0] == "adduser":
                    if arr[1] == username:
                        record = "exists"
                        break
            if record == 'none':
                data =
    "adduser#"+username+"#"+password+"#"+contact+"#"+email+"#"+address+"#"+gender+"
    \n"
                saveDataBlockChain(data,"adduser")
                context= {'msg':'Signup process completd and record saved in
    Blockchain'}
                return render(request, 'Signup.html', context)
            else:
                context= {'msg':username+'Username already exists'}
                return render(request, 'Signup.html', context)

    def Signup(request):
        if request.method == 'GET':
           return render(request, 'Signup.html', {})
```

```
def Deposit(request):
    if request.method == 'GET':
        global user_id
        output = '<tr><td><font size="3" color="black">Username</td><td><input
type="text" name="t1" size="20" value='+user_id+' readonly/></td></tr>'
        context= {'msg1':output}
        return render(request, 'Deposit.html', context)

def Logout(request):
    if request.method == 'GET':
       return render(request, 'index.html', {})

def index(request):
    if request.method == 'GET':
       return render(request, 'index.html', {})

def Login(request):
    if request.method == 'GET':
       return render(request, 'Login.html', {})
```

Similarly you can create N users and send money which running code TRUFFLE ETHEREUM tool and DJANGO SERVER must be running and this server I am showing below screen

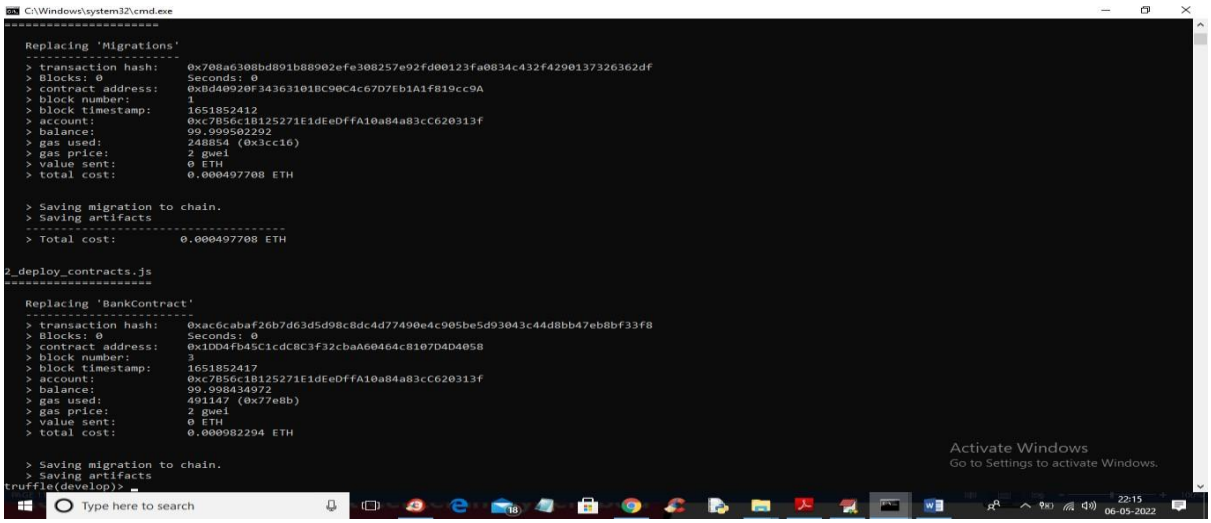


Fig:7.16 : Truffle Migrations

Double click on ‘run.bat’ file to start ‘DJANGO’ web server like below screen

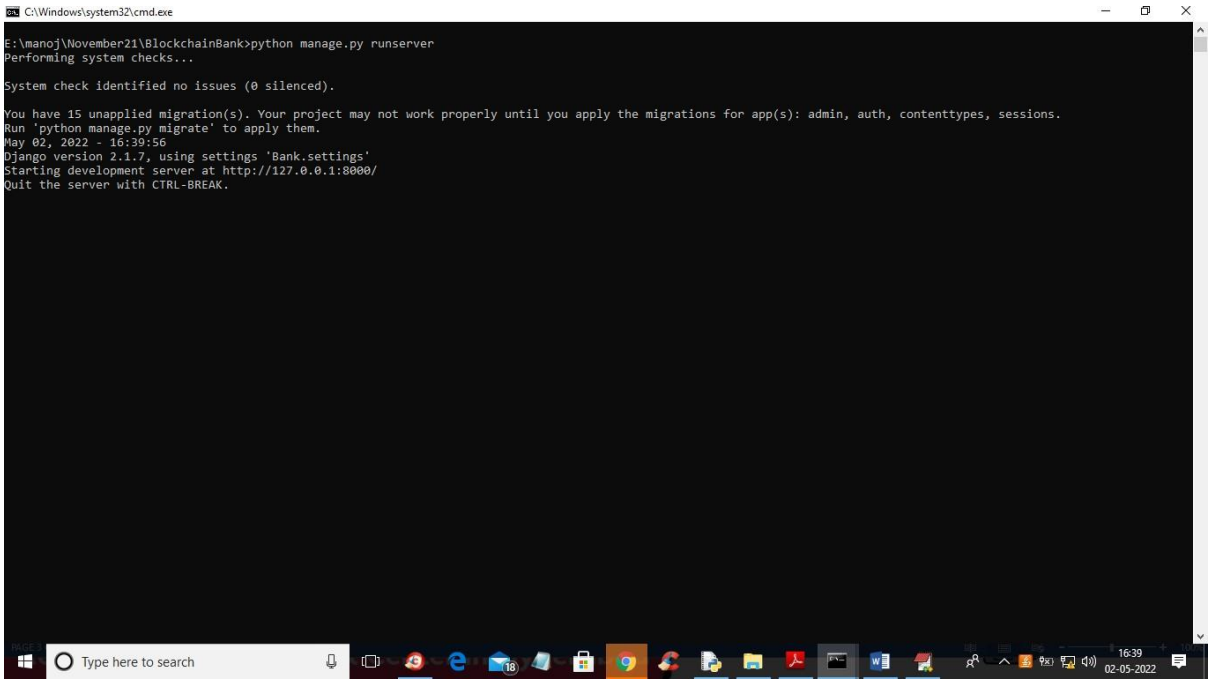


Fig:7.17:Local Host Access

In above screen DJANGO server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below page

7.4 Output screens:

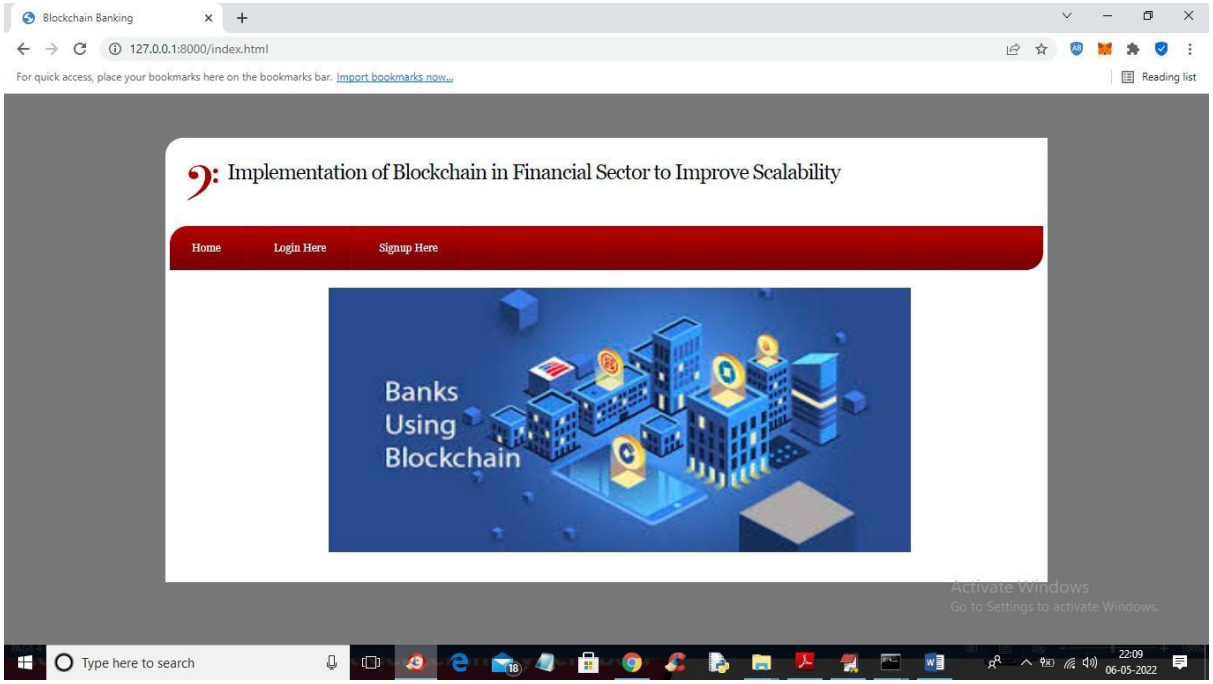


Fig:7.18: Home Screen Interface

In above screen click on 'Signup here' link to get below screen

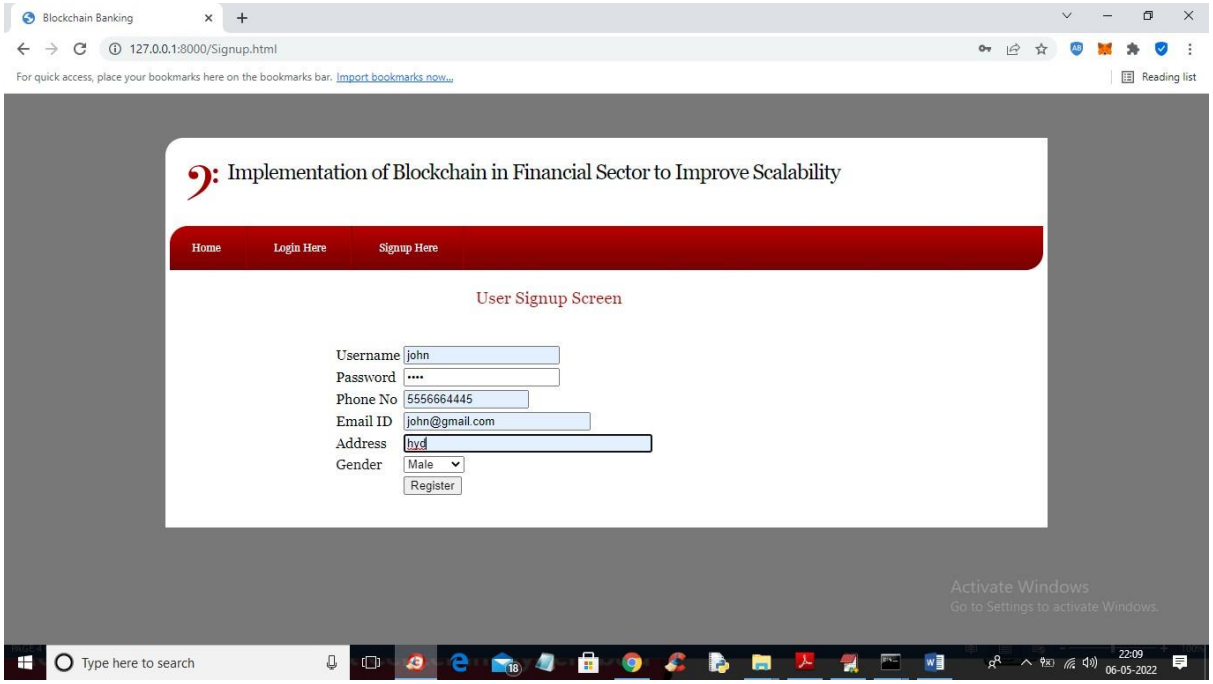


Fig 7.19: SignUp Screen

In above screen user is entering signup details and then press button to get below output

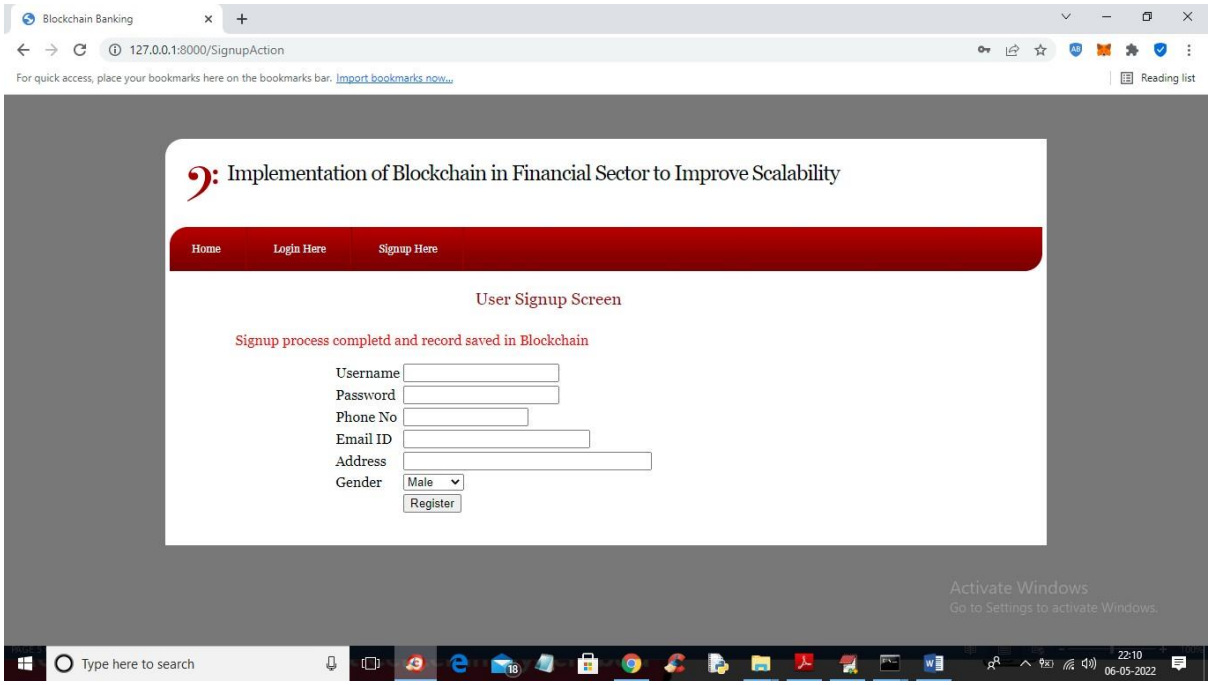


Fig:7.20: Registered User Screen

In above screen user signup details saved in Blockchain and now click on 'Login Here' link to get below login screen

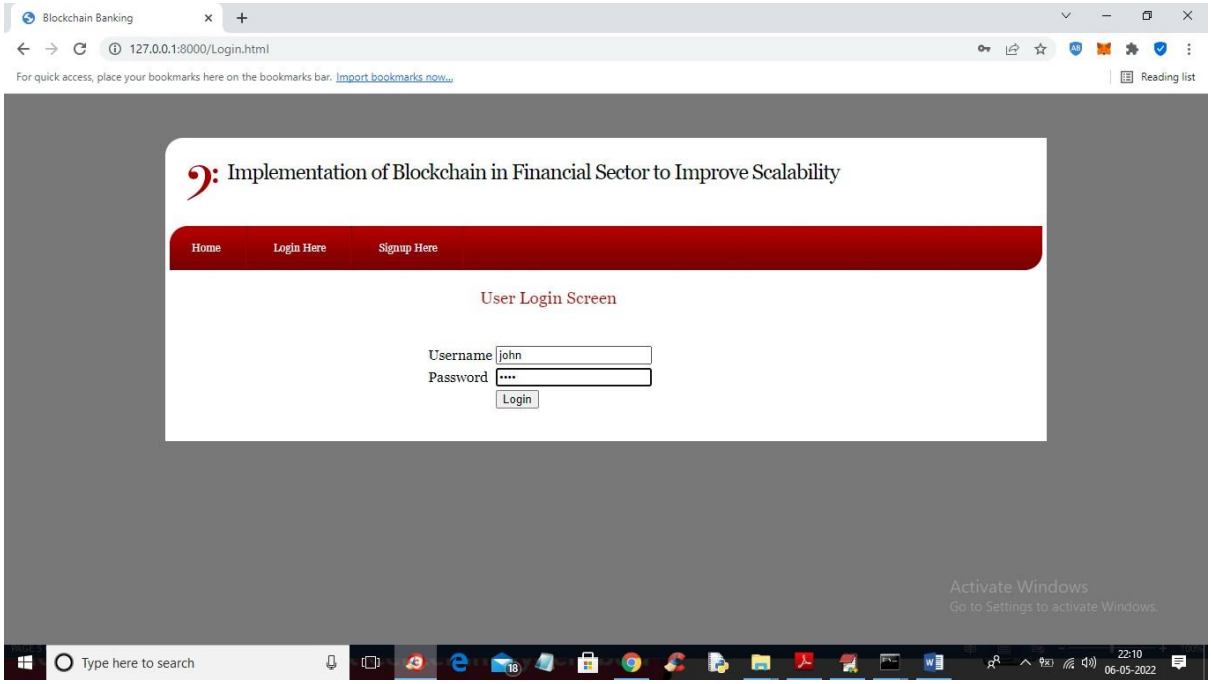


Fig 7.21:Login Screen

In above screen user is login and after login will get below output

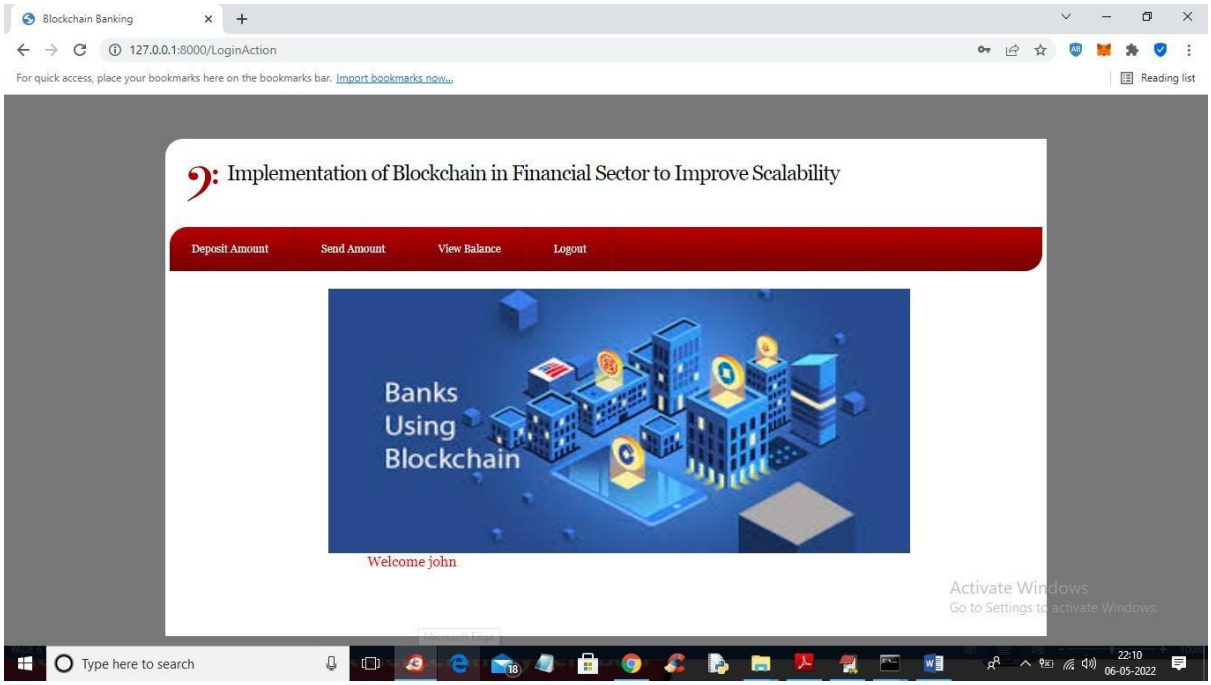


Fig:7.22:Login Home Screen

In above screen user can click on 'Deposit Amount' link to add amount to his account

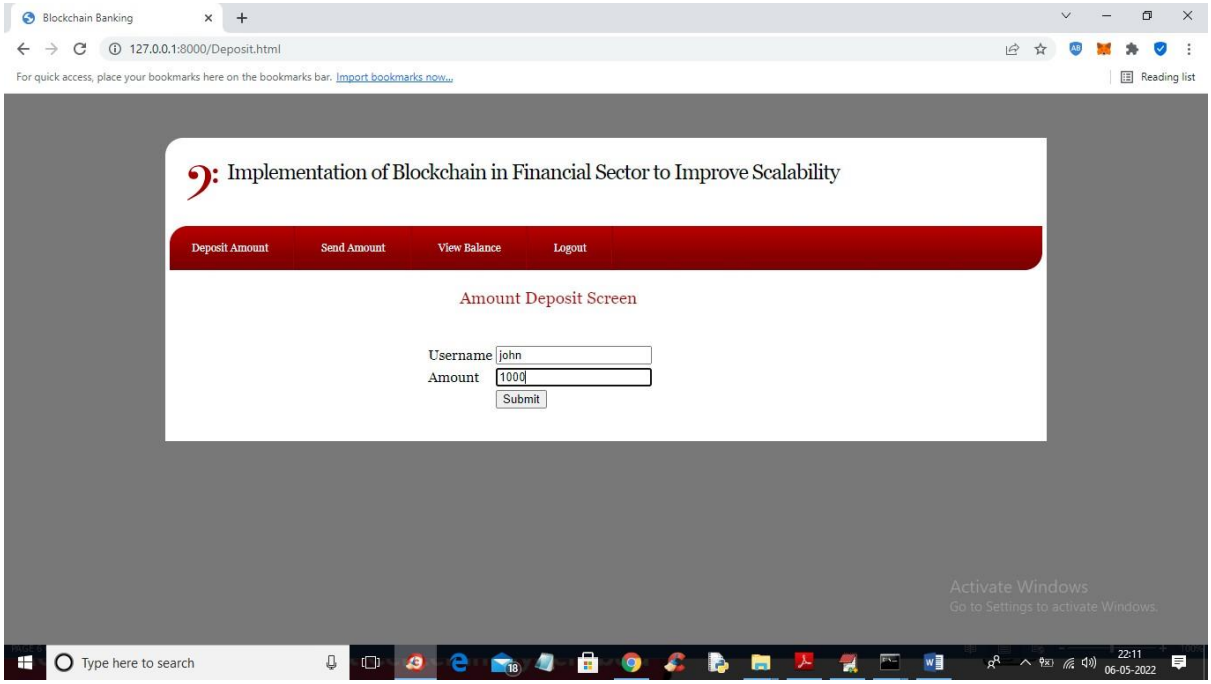


Fig:7.23: Deposit Screen

In above screen user will enter amount and press button to add amount to his account

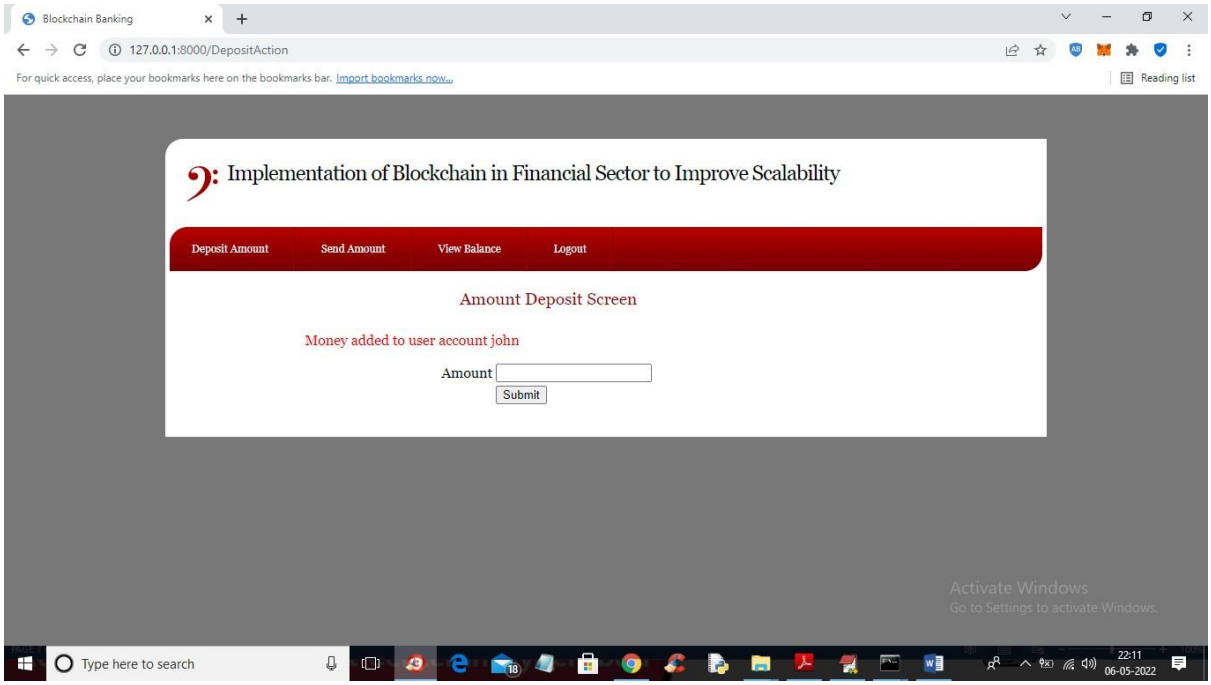


Fig:7.24: Deposited Screen

In above screen money added to user account and now click on 'View Balance' link to view his balance

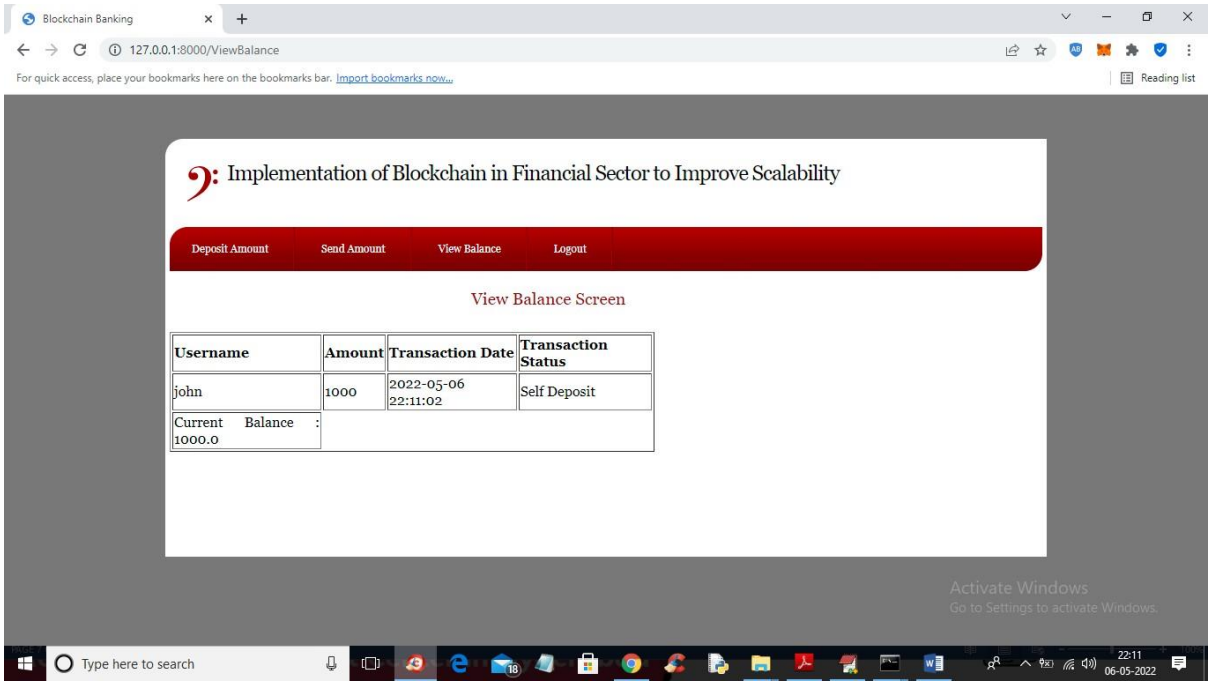


Fig:7.25:Balance Sheet

In above screen user has a current balance 1000 and now clicks on 'Send Amount' link to get below screen

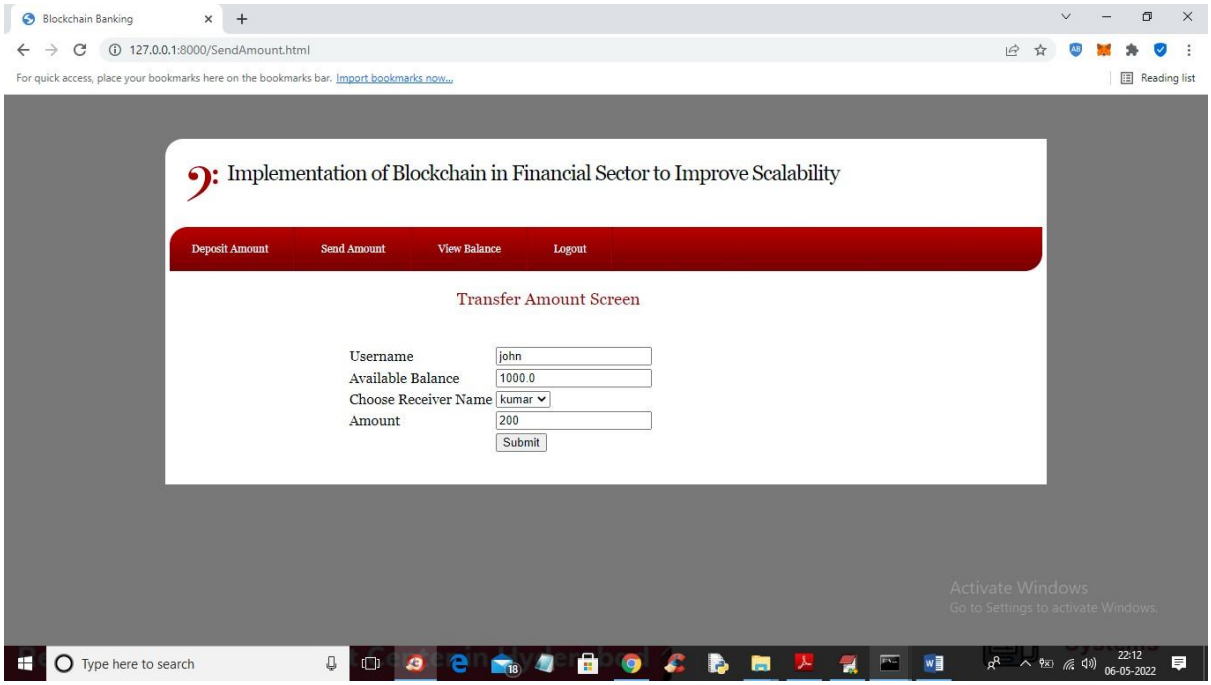


Fig:7.26: Amount Transfer Screen

In above screen user 'John' selecting receiver name 'kumar' to send amount 200 and his current balance is 1000 and now press button to send amount and get below output

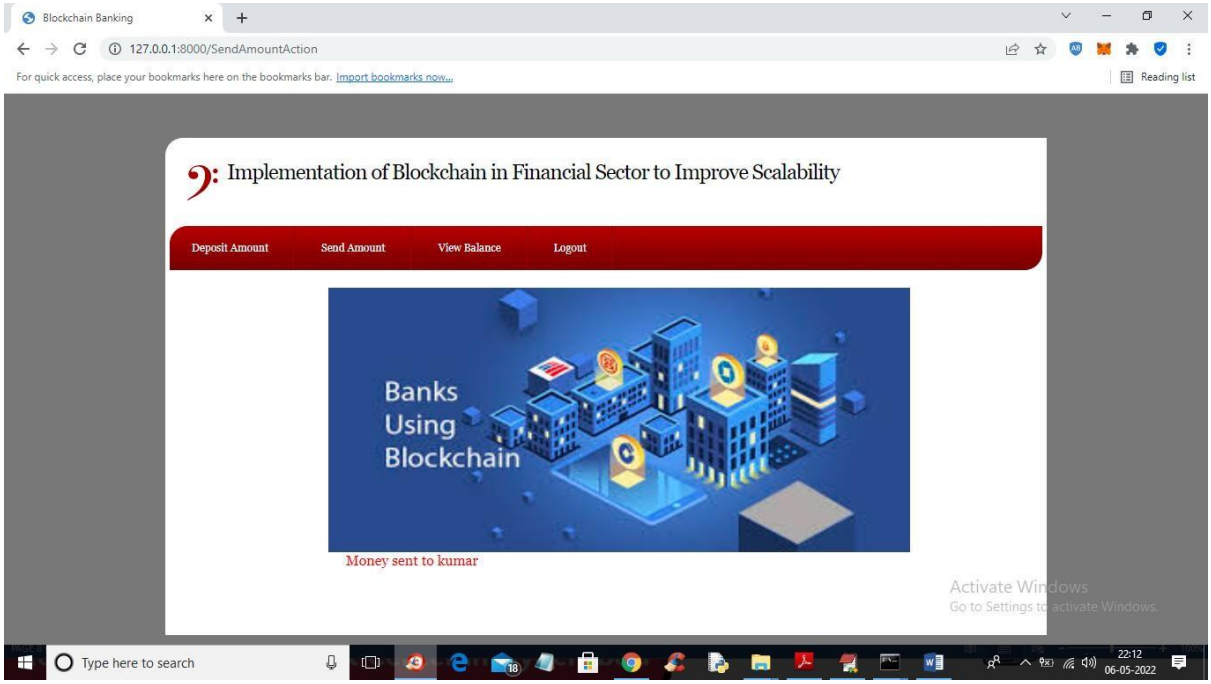


Fig:7.27: User-2 Home Screen

In above screen in red colour text we can see money sent and now click on 'View Balance' link again to view his current balance

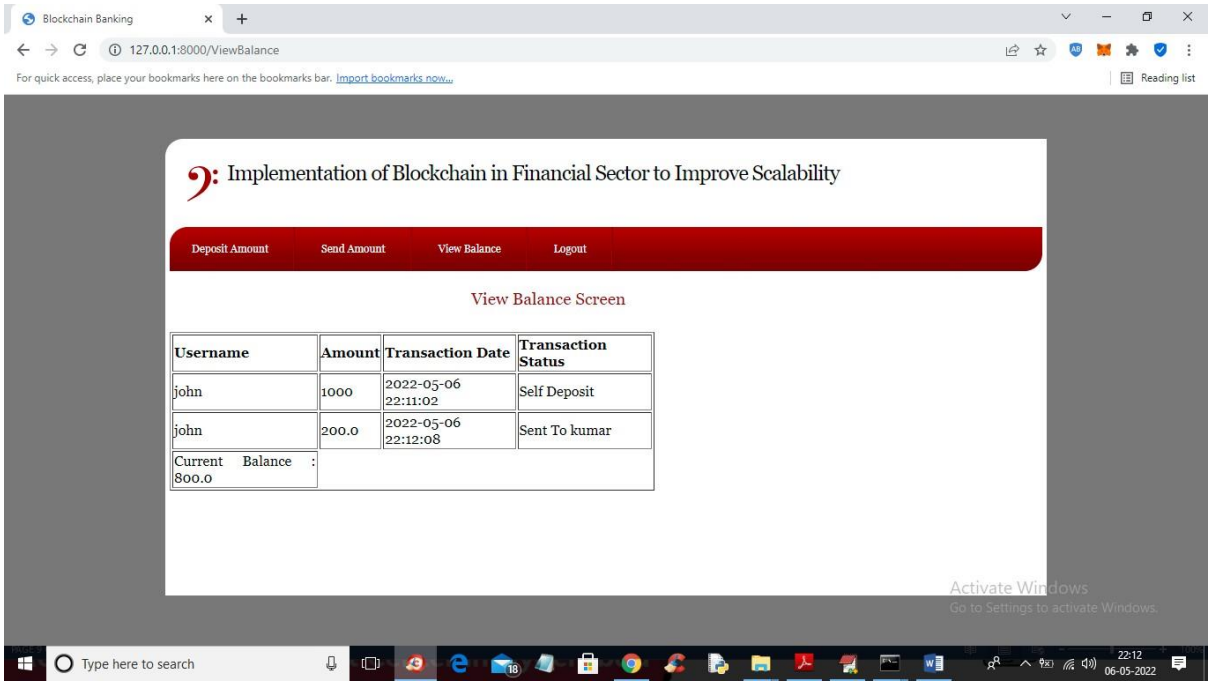


Fig:7.28:Reciver's Balance Screen

In above screen user balance reduced to 800 after sending 200 to kumar and similarly you can create any number of accounts and make transaction using Blockchain Bank accounts. In below user kumar screen we can see 200 credited from user John

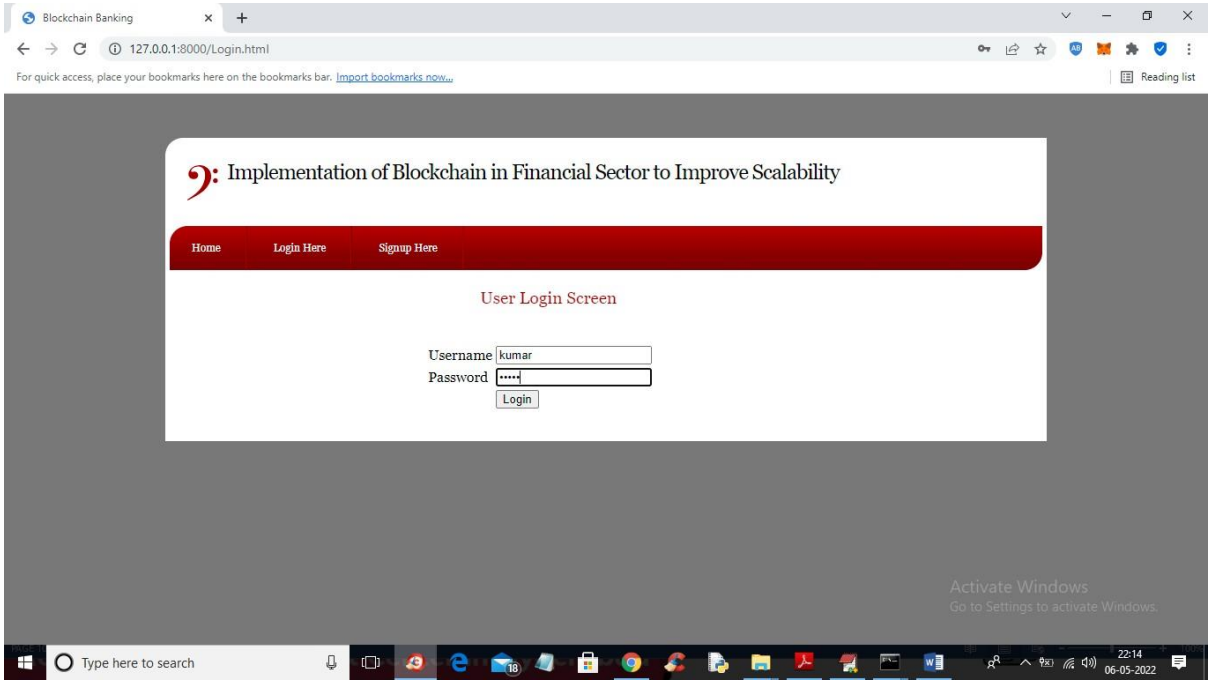


Fig7.29: Login Screen

In above screen kumar user is login and after login click on 'View Balance' link to get below output

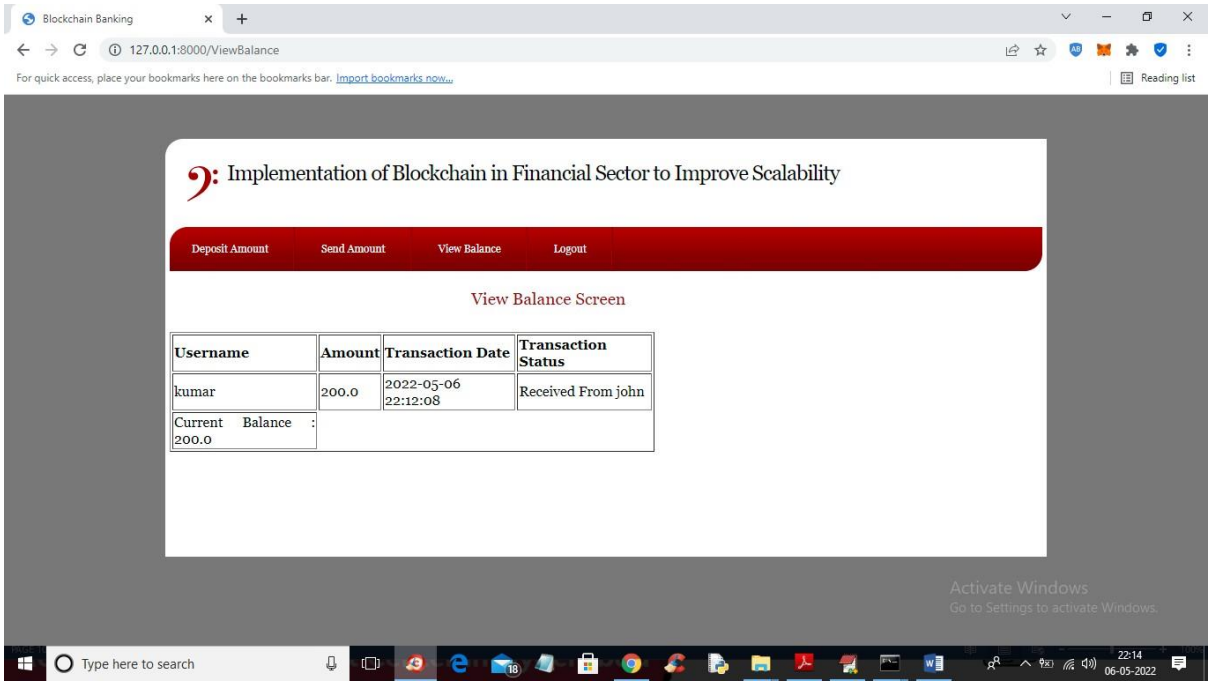


Fig:7.30:Receiver's Balance screen

Json updated:

Fig:7.31:Json File With Transaction Hash

In above screen we can see user kumar received 200 from user John

Similarly you can create N users and send money which running code TRUFFLE ETHEREUM tool and DJANGO SERVER must be running and this server I am showing below screen

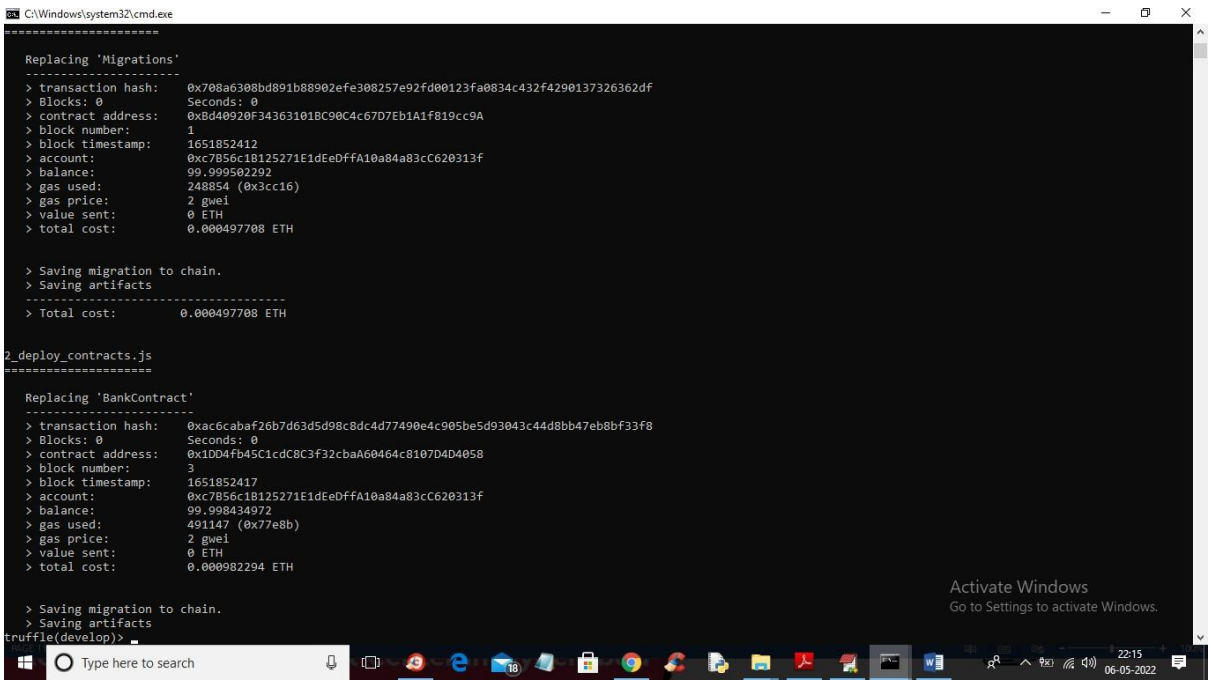


Fig:7.32: Command and Prompt Screen with Transaction Hash

8.SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

TYPES OF TESTS

8.1Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .It is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.5 White Box Testing

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It has a purpose. It is used to test areas that cannot be reached from a black box level.

8.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a test in which the software under test is treated as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed

- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.CONCLUSION:

Although the potential of blockchain is widely claimed to be at par with early commercial interest, banking firms need to understand the key features of the technology and how it can solve the current business issues as on one hand, the internet enabled the exchange of data while on other, the blockchain can involve the exchange of value. Banks need to identify opportunities, determine feasibility and impact and test proof of concepts. However, the questions around emulations will have to be resolved through focused discussions with competent regulatory authoritiesand incorporation of their thought-process.

Further we will research how we can provide off-chain settlement for the banks whichare not listed on the platform, one of the alternate to do it is to access its database with the permission, due to which further transaction can be taken place (between listed and non-listed banks) so that both can have equal ledger maintained.

This increasing demand deems sustainability in the long run and improvised security system. The Blockchain is implemented for handling a huge amount of data without compensating any vulnerability. It provides a distributed ledger system making it equally accessible and authorized for all the users. The longer and stronger the chain is, the more it is efficient.

It is a rapidly developing interface in the contemporary world seeking new opportunities in the market and for business.

10. REFERENCES:

[1] Tejal Saha, Shalilak Jani, "Applications of Blockchain Technology in banking and finance", Parul CUniversity, Vadodara, India, February 2018 DOI: 10.13140/RG.2.2.35237.96489

[2] DUSKO KNEZEVIC, "Impact of blockchain technology platform in changing the financial sector and other industries. University Union Belgrade, Serbia, Montenegrin Journal Of Economics, Vol. 14, No. 1(2018), pp.(109-120).

[3] Lin William Cong Zhiguo He Working Paper 24399 http://www.nber.org/papers/w24399 NATIONAL BUREAU OF ECONOMIC RESEARCH 1050 Massachusetts Avenue Cambridge, MA 02138 March 2018, Revised April 2018

[4] Soonduck Yoo, (2017) "Blockchain based financial case analysis and its implications", Asia Pacific Journal of Innovation and Entrepreneurship, Vol. 11 Issue: 3, pp.312-321 https://doi.org/10.1108/APJIE-12-2017-036

[5] C. Miguel and L. Barbara, "Practical byzantine fault tolerance," in Proceedings of the Third Symposium on Operating Systems Design and Implementation, vol. 99, New Orleans, USA, 1999, pp. 173–186.

[6] Underwood, S. (2016), Blockchain beyond Bitcoin, Commun. ACM, Vol. 59, No. 11, pp. 15–17. https://doi.org/10.1145/2994581

[7] "Digital currencies", IEEE Commun. Survey Tutorials, Vol. 18, No. 3, pp. 2084–2123. https://doi.org/10.1109/COMST.2016.2535718. Greenspan, G. (2015), "MultiChain Private Blockchain", White Paper Founder and CEO, Coin Sci-ences Ltd, https://www.multichain.com

[8] M. Vukoli´c, "The quest for scalable blockchain fabric: Proof-of work vs. bft replication," in International Workshop on Open Problems in Network Security, Zurich, Switzerland, 2015, pp. 112– 125.

[9] D. Kraft, "Difficulty control for blockchain-based consensus systems," Peer-to-Peer Networking and Applications, vol. 9, no. 2, pp. 397–413, 2016.

[10] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoinng: A scalable blockchain protocol," in Proceedings of 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), Santa Clara, CA, USA, 2016, pp. 45–59.

[11] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in Proceedings of 13th USENIXSymposium on Networked Systems Design and Implementation (NSDI16), Santa Clara, CA, USA, 2016, pp. 45–59.

[12] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W.Felten, “Mixcoin: Anonymity for bitcoin with accountable mixes,” in Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg, 2014, pp. 486– 504.

[13] I. Miers, C. Garman, M. Green, and A. D. Rubin, “Zerocoin: Anonymous” Distributed E-Cash from Bitcoin. IEEE Sympouism on Security and privacy 2013.