**Blockchain Based**

**Transactional System**

**A Minor Project Work**

**Submitted In Fulfillment Of The Requirements For The Award Of Degree Of Bachelor of Technology In Information Technology By**

**Ritika Sharma**

**Enroll Number: 0201 IT 191064**

**Ritvik Shrivastava**

**Enroll Number: 0201 IT 191065**

**Varsha Pandey**

**Enroll Number: 0201IT191097**

**Under the guidance of**

**prof. Loveleen Kaur**

**Associate Professor, Department of Information Technology**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY JABALPUR ENGINEERING COLLEGE**

**JABALPUR (M.P.)**

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA BHOPAL**

**SESSION 2019-2023**

**JABALPUR ENGINEERING COLLEGE**

**JABALPUR**

****

**Approved by AICTE New Delhi & Govt. of M.P.**

**Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya**

**CERTIFICATE**

This is to certify that the dissertation work entitled “Blockchain based Transaction system” is a project work, done by Ritika Sharma , Ritvik Shrivastava,Varsha Pandey under my guidance and supervision for the degree of Bachelor of Technology in Information Technology of Rajiv Gandhi Proudyogiki Vishwavidyalaya,Bhopal (M.P.), India. The project has duly been completed and fulfills the requirement of the ordinance to the Bachelor of Technology degree of the University. To the best of my knowledge and belief the project embodies the work of the candidate himself and that the work has not been submitted earlier in part or full for the award of any other degree.

**Prof. Loveleen Kaur**

**Head of the Department**

**Date:**

**Guide Name Prof. Loveleen Kaur**

**-**

**Co-Guide Name Prof. Neelu Tiwari**

**-**

**Date:**

**DECLARATION**

I hereby declare that the work, which is being presented in the

project work, entitled “***Blockchain based Transaction system***”

fulfillment of the requirements for the award of degree of Bachelor of

Technology in Information Technology submitted in the department of

Information Technology of Jabalpur Engineering College is an authentic

record of my own work carried under the guidance of “Prof. Lovleen Kaur

”. I have not submitted the matter embodied in this report for award of any

other degrees.

**Name of the Student – Ritika Sharma**

**(Enrollment No.) –0201IT191064**

**Name of the Student – Ritvik Shrivastava**

**(Enrollment No.) - 0201IT191065**

**Name of the Student – Varsha Pandey**

**(Enrollment No.) –0201IT191097**

**Project Guide**

**JABALPUR ENGINEERING COLLEGE**

**JABALPUR**

****

**Approved by AICTE New Delhi & Govt. of M.P.**

**Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya**

**CERTIFICATE**

The Project work entitled “Blockchain based Transaction system”

submitted by -

**Ritika Sharma (0201IT191064),**

**Ritvik Shrivastava (0201IT191065),**

**Varsha Pandey (0201IT191097)**

is approved for the award of degree of

Bachelor of Technology in Information Technology.

**INTERNAL EXAMINER**

**Date:**

**EXTERNAL EXAMINER**

**Date:**

**ACKNOWLEDGMENT**

I would like to express my deep sense of gratitude and sincere thanks to my

guide Prof.Loveleen Kaur ,Associate Professor, Department of Information

Technology and Asst. Prof. Neelu Tiwari for his valuable and ever willing

precious guidance, technical support and constant encouragement during the

course of this dissertation work. It was a pleasure and unique experience for me to work under their guidance.

I am grateful to Prof. Loveleen Kaur, Head of Department of Information

Technology and other staff members of Information Technology Department

for providing the necessary facilities for the successful completion of this work.

***Ritika Sharma***

**0201IT191064**

***Ritvik Shrivastava***

**0201IT191065**

***Varsha Pandey***

**0201ITI91097**

**ABSTRACT**

The Blockchain based Transactional System is a blockchain technology based

software used to generate successful ethereum transactions between clients.

The Blockchain transactions act on the identical ledger data stored at each node.

Transactions are sent from and received by user-created Ethereum accounts. A sender must sign transactions and spend Ether, Ethereum's native cryptocurrency,

as a cost of processing transactions on the network.

Transaction records are immutable, verifiable, and securely distributed across the network, giving participants full ownership and visibility into transaction data.

Smart contracts allow participants to transact with each other without a trusted central authority.

Ethereum is a decentralized blockchain platform that establishes a peer-to-peer network that securely executes and verifies application code, called smart contracts.

This application basically provides an efficient , reliable, secure and safe way to make transactions between clients.

**LIST OF CONTENTS -**

|  |
| --- |
| 1. **INTRODUCTION** 2. **1.1 inspiration** 3. **1.2 why transactional system in blockchain** |
| **2. Literature survey** |
| **3. Methodology**   1. **Agile model** |
| **4. Implementational Details**   1. **Blockchain** 2. **Ethereum** 3. **Transactions and smart contracts.** 4. **Platforms-**   **A.1 visual studio**  **A.2 metamask** |
| **5. Software Testing**   1. **5.1 Testing principles** 2. **Types of testing** 3. **White box** 4. **Black box** 5. **Gray box** 6. **Manual testing** 7. **Testing the application.** |
| **6. Visual flow of system** |
| **7. Codes and implementations.** |
| **8. Output screens** |
| **9. Conclusion** |

**1. INTRODUCTION**

On the blockchain, the process of transaction verification and recording is immediate and permanent. The ledger is distributed across several nodes, meaning the data is replicated and stored instantaneously on each node across the system. When a transaction is recorded in the blockchain, details of the transaction such as price, asset, and ownership, are recorded, verified and settled within seconds across all nodes. A verified change registered on any one ledger is also simultaneously registered on all other copies of the ledger. Since each transaction is transparently and permanently recorded across all ledgers, open for anyone to see, there is no need for third-party verification.

**1.1 Inspiration**

Before Blockchain, traditional Banking had transfer fees, which can be both expensive and time-consuming for people. Also, sending money overseas becomes even more difficult due to the exchange rate and other hidden costs.

Transactions placed through a central authority can take up to a few days to settle. If you attempt to deposit a check on Friday evening, for example, you may not actually see funds in your account until Monday morning. Whereas financial institutions operate during business hours, usually five days a week, blockchain is working 24 hours a day, seven days a week, and 365 days a year. Transactions can be completed in as little as 10 minutes and can be considered secure after just a few hours. This is particularly useful for [cross-border](https://www.investopedia.com/articles/forex/09/currency-cross-triangulation.asp) trades, which usually take much longer because of time zone issues and the fact that all parties must confirm payment processing.

**1.2 Why a Transactional system in Blockchain**

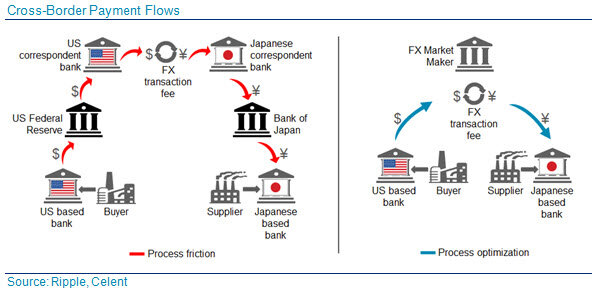
**technology–**

Blockchain eliminates the need for a middleman. Blockchain is disrupting the banking system by providing a peer-to-peer payment system with the highest security and low fees.

Blockchain technology provides instant and borderless payments across the globe Cryptocurrencies (like Ethereum, bitcoin) remove the requirement for a third party to perform transactions

Blockchain records all the transactions in a public ledger which is globally accessible by bitcoin users.

Typically, consumers pay a bank to verify a transaction, a notary to sign a document, or a minister to perform a marriage. Blockchain eliminates the need for third-party verification—and, with it, their associated costs. For example, business owners incur a small fee whenever they accept payments using credit cards, because banks and payment-processing companies have to process those transactions. Bitcoin, on the other hand, does not have a central authority and has limited transaction fees.

****

**2. LITERATURE SURVEY**

This section consists of various sources like websites, courses etc. which we used during the development of the project . From Planning to Development , these sources helped a lot

.

**Tutorialspoint** - Tutorials Point is a leading Ed Tech company striving to provide the best learning material on technical and non-technical subjects.

**StackOverflow** - Stack Overflow is a question and answer website for professional and enthusiast programmers. It is the flagship site of the Stack Exchange Network.. It features questions and answers on a wide range of topics in computer programming.

**GitHub** - GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.

**Visual Studio**- Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc.

**YouTube** - YouTube is an American online video sharing and social media platform headquartered in San Bruno, California. It is owned by Google, and is the second most visited website, after Google Search.

**GeeksForGeeks** - GeeksForGeeks The platform was founded in 2009 by Sandeep Jain (an IIT Roorkee alumnus) as a one-stop solution for CS/IT students to learn Programming Concepts ,Data Structures & Algorithms, Coding Practice, etc.

**3. METHODOLOGY**

To build this application we have used the Agile model of Software Development Life Cycle and MVVM (Model View ViewModel) architecture for the development of the project.

**3.1 Agile Model –**

The meaning of Agile is swift or versatile."Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



# **4. IMPLEMENTATIONAL DETAILS**

# ***4.1 BlockChain***

Blockchain is a type of shared database that differs from a typical database

in the way that it stores information; blockchains store data in blocks that

are then linked together via cryptography. As new data comes in, it is entered into a fresh block. Once the block is filled

with data, it is chained onto the previous block, which makes the data

chained together in chronological order.

Different types of information can be stored on a blockchain, but the most

common use so far has been as a ledger for transactions.

In Bitcoin’s case, blockchain is used in a decentralized way so that no single person or group has control—rather, all users collectively retain control.

Decentralized blockchains are immutable, which means that the data entered is irreversible. For Bitcoin, this means that transactions are permanently recorded and viewable to anyone.



# ***4.2 Ethereum***

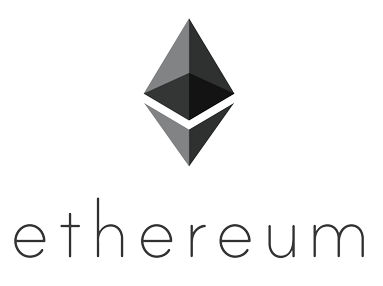
When a transaction triggers a smart contract, all nodes of the network execute every instruction. To do this, Ethereum implements an execution environment on the blockchain called the Ethereum Virtual Machine (EVM).

All nodes on the network run the EVM as part of the block verification protocol. In block verification, each node goes through the transactions listed in the block they are verifying and runs the code as triggered by the transactions in the EVM. All nodes on the network do the same calculations to keep their ledgers in sync. Every transaction must include a gas limit and a fee that the sender is willing to pay for the transaction. Miners have the choice of including the transaction and collecting the fee or not

. If the total amount of gas needed to process the transaction is less than or equal to the gas limit, the transaction is processed. If the gas expended reaches the gas limit before the transaction is completed, the transaction does not go through and the fee is still lost.

All gas not used by transaction execution is reimbursed to the sender as Ether.

This means that it's safe to send transactions with a gas limit above the estimates.



**4.3 Transactions and Smart**

**Contracts**

A transaction in Ethereum is a signed data message sent from one Ethereum

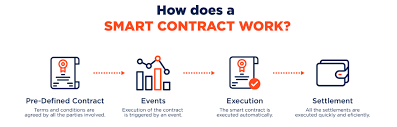
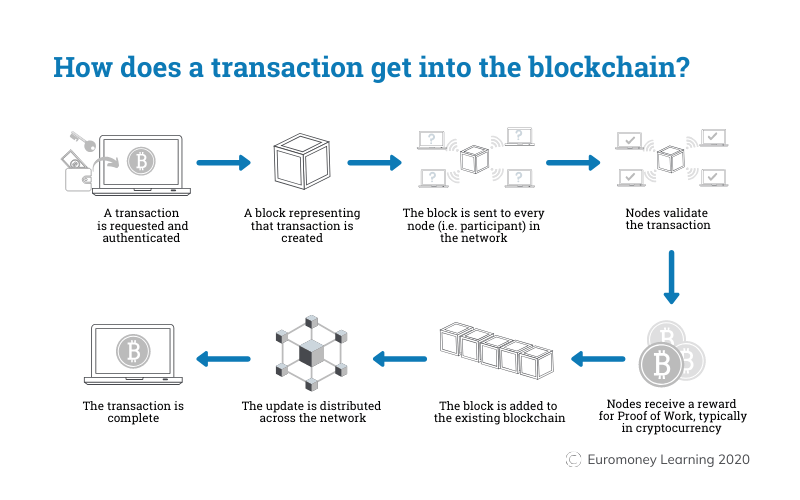
account to another. It contains the transaction sender and recipient information,

the option to include the amount of Ether to be transferred, the smart contract

bytecode, and the transaction fee the sender is willing to pay to the network

validators to have the transaction included in the blockchain, known as gas

price and limit.



**4.4 Platforms**

**Visual Studio:**

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps,

cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for

Windows as well as for macOS.

Support for different programming languages in Visual Studio is added by using a special VSPackage which is known as Language Service.

When you will install the Visual Studio then the functionality which is coded as VSPackage will be available as Service.

Visual Studio IDE provides the three different types of services known as SVsSolution, SVsUIShell, and SVsShell.

SVsSolution service is used to provide the functionality to enumerate solutions and projects in Visual Studio. SVsUIShell service is used to provide User Interface functionality like toolbars, tabs etc. SvsShell service is used to deal with the registration of VSPackages.



# ***METAMASK***

MetaMask is a software cryptocurrency wallet used to interact with the Ethereum

blockchain. It allows users to access their Ethereum wallet through a browser

extension or mobile app, which can then be used to interact with decentralized

applications.[1][2] MetaMask is developed by ConsenSys Software Inc., a

blockchain software company focusing on Ethereum-based tools and infrastructure.

MetaMask is a browser plugin that serves as an Ethereum wallet, and is

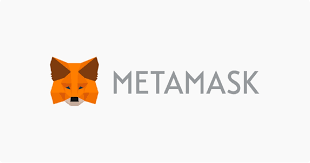
installed like any other browser plugin. Once it's installed, it allows users to

store Ether and other ERC-20 tokens, enabling them to transact with any

Ethereum address.

By connecting to MetaMask to Ethereum-based dapps, users can spend their coins in games, stake tokens in gambling applications, and trade them on

decentralized exchanges (DEXs). It also provides users with an entry point into the emerging world of decentralized finance, or DeFi, providing a way to access DeFi apps such as Compound and PoolTogether.



***5. SOFTWARE TESTING***

**5.1.Testing Principles**

Software testing is a procedure of implementing software or the application to identify the defects or bugs. For testing an application or software, we need to follow some

principles to make our product defects free, and that also helps the test engineers to

test the software with their effort and time.

Some of the basic testing principles are:

● Early Testing

● Exhaustive Testing is not possible

● Testing shows the presence of defects

● Defect Clustering

● Pesticide Paradox

● Testing is context-dependent

● Absence of errors fallacy

**5.2 Types of Testing**

**5.2.1 White-Box Testing:**

White box testing which is also known as glass box testing, structural testing, clear box testing, open box testing and transparent box testing. It tests internal coding and infrastructure of a software focused on checking predefined inputs against expected and desired outputs. It is based on the inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases.

The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the

software.

In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the Black-box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application .

**2.2 Black-Box Testing:**

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

In this method, the tester selects a function and gives input value to examine its

functionality, and checks whether the function is giving expected output or not. If the function produce correct output, then it is passed in testing, otherwise failed. The test team reports the result to the development team and then tests the next function. After completing testing of all functions if there are severe problems, then it is given back to the development team for correction.

**5.2.3 Gray-Box Testing:**

Greybox testing is a software testing method to test the software application with

partial knowledge of the internal working structure. It is a combination of black box and white box testing because it involves access to internal coding to design test cases as white box testing and testing practices are done at functionality level as black box

Testing. GreyBox testing commonly identifies context-specific errors that belong to web systems. For example; while testing, if a tester encounters any defect then he makes changes in code to resolve the defect and then tests it again in real time. It concentrates on all the layers of any complex software system to increase testing coverage. Grey box testing does not make it necessary that the tester must design test cases from source code.

To perform this testing, test cases can be designed on the base of,

knowledge of architectures, algorithms, internal states or other high - level descriptions of the program behavior. It uses all the straightforward techniques of black box testing for function testing.

The test case generation is based on requirements and preset all the conditions before testing the program by assertion method**.**

**5.2.4 Manual Testing:**

Manual testing includes testing software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the

software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

**5.2.3.1 Need for Manual Testing:**

Whenever an application comes into the market, and it is unstable or having a bug or issues or creating a problem while end-users are using it.

If we don't want to face these kinds of problems, we need to perform one round of testing to make the application bug free and stable and deliver a quality product to the client, because if the application is bug free, the end-user will use the application more conveniently.

If the test engineer does manual testing, he/she can test the application as an

end-user perspective and get more familiar with the product, which helps them to write the correct test cases of the application and give the quick feedback of the application.

**5.3 Testing the Application:**

While Testing our application, we came across many errors and bugs that we had to resolve, particularly in the coding section we debug each and every line to test it

every single time.

1. Separate unit testing of each 36 HTML pages.

2. We performed integrated testing of these connected HTML pages.

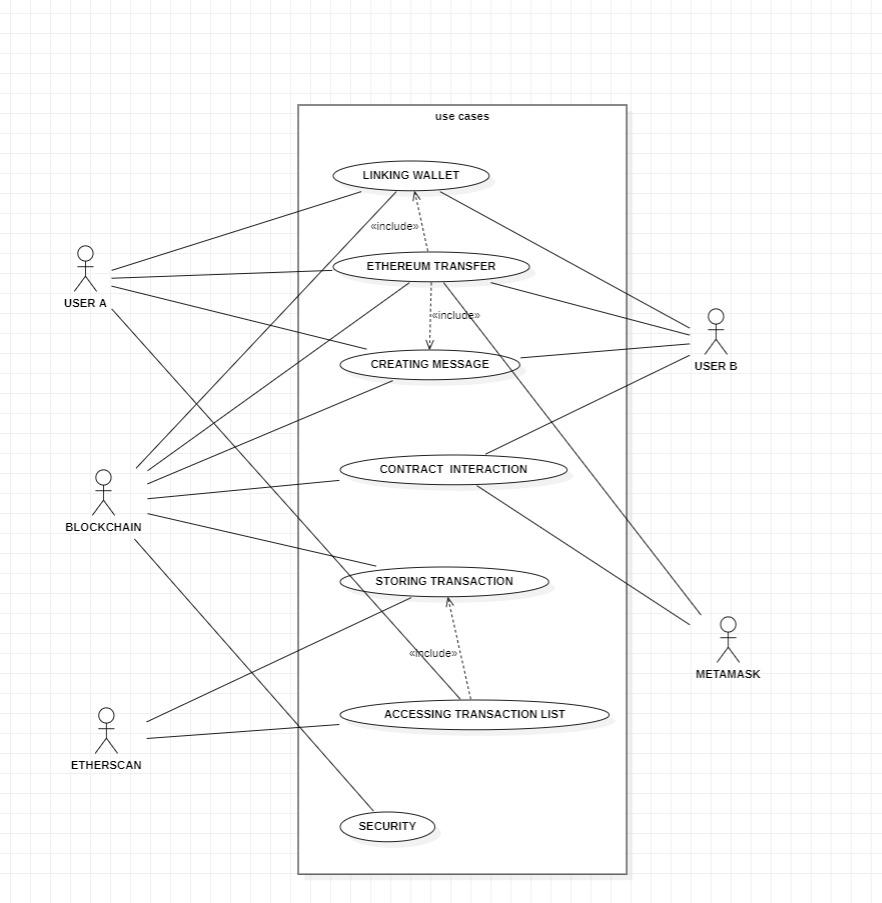
3. After that performed system testing like checking screen resolution in different devices and alignment of pages also.

4. Data was not getting added to the database after entering it.

5. We were not getting desired output.

***6. VISUAL FLOW OF SYSTEM -***

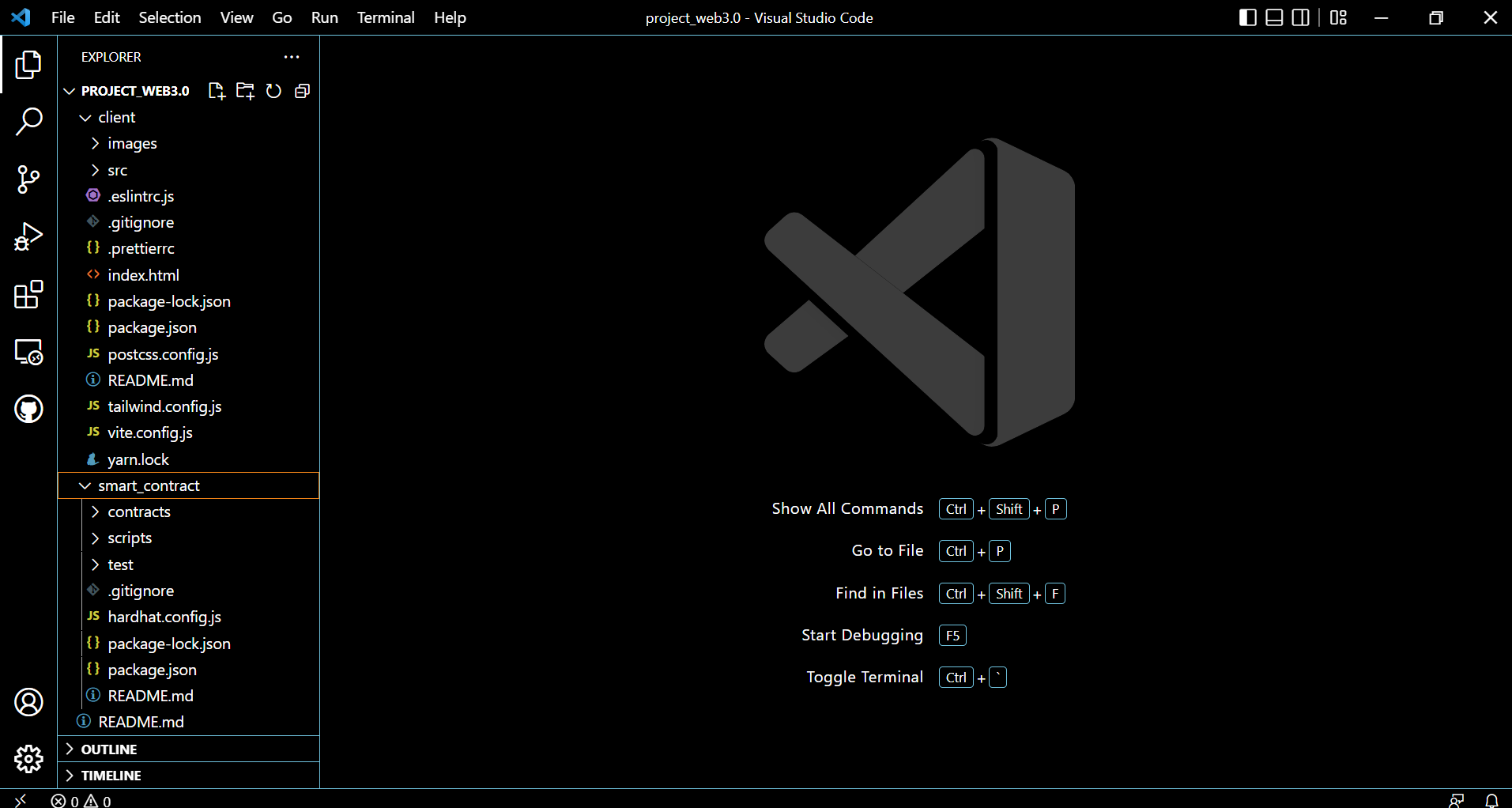
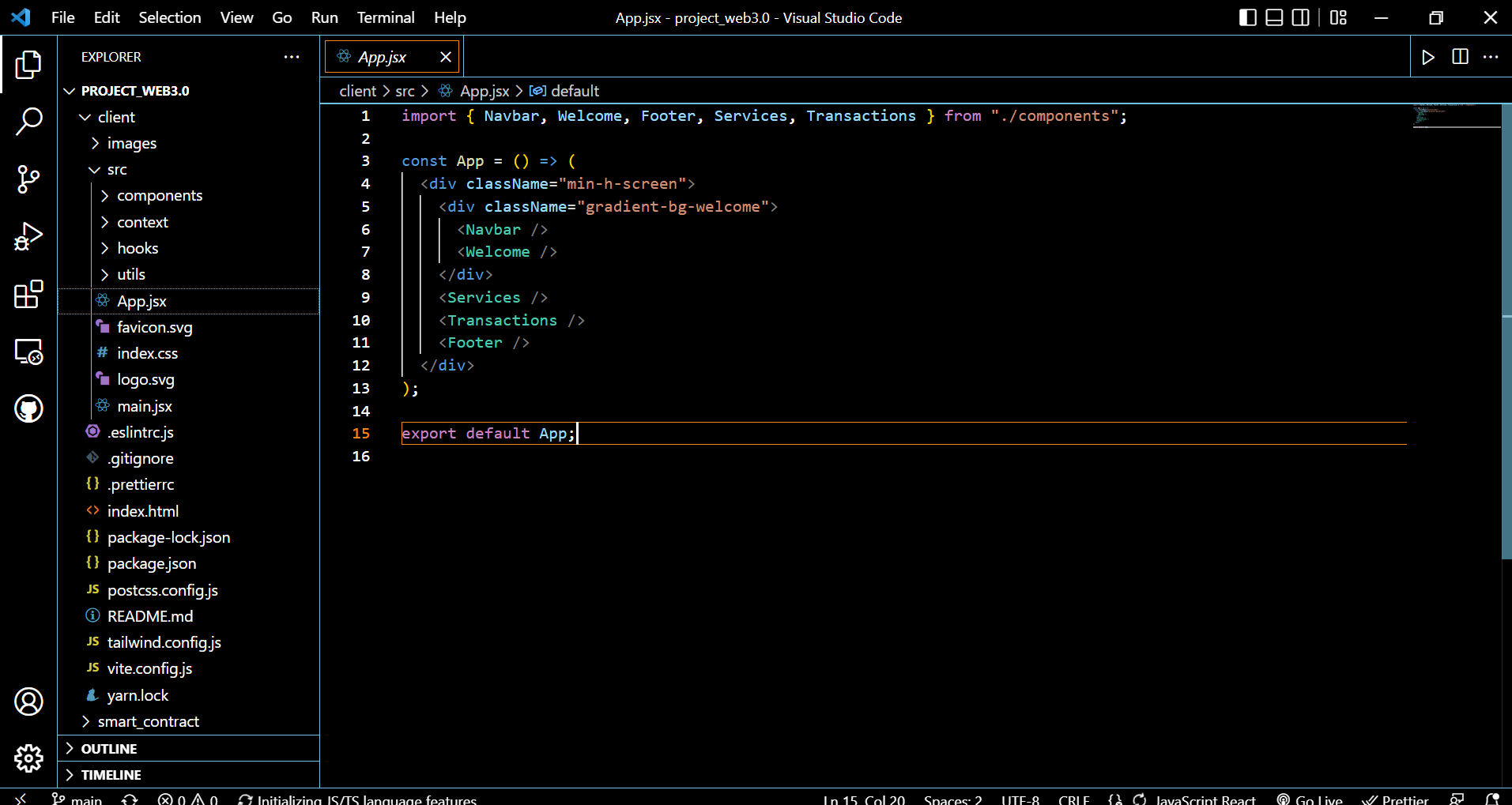
**USE CASE DIAGRAM**

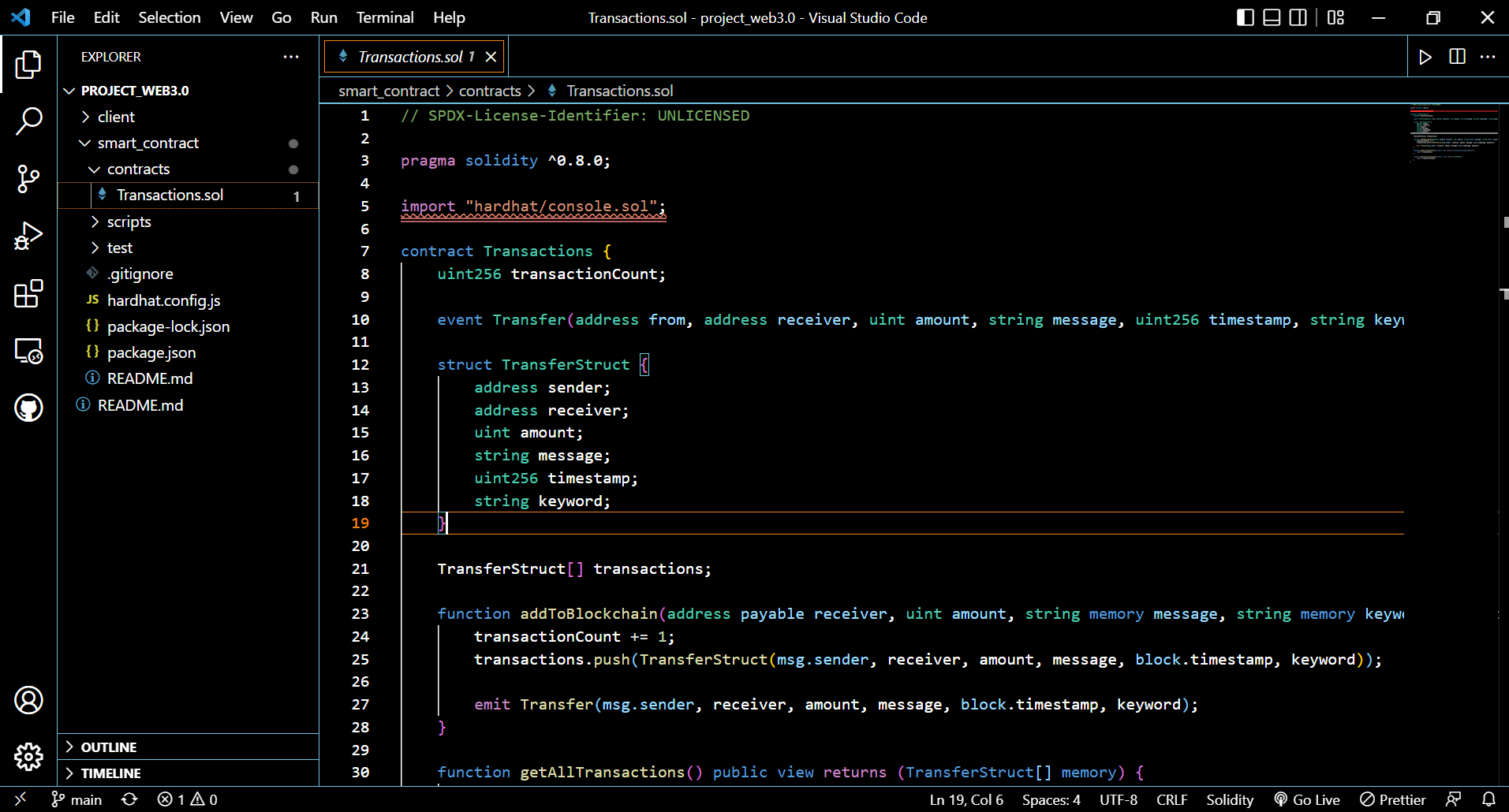


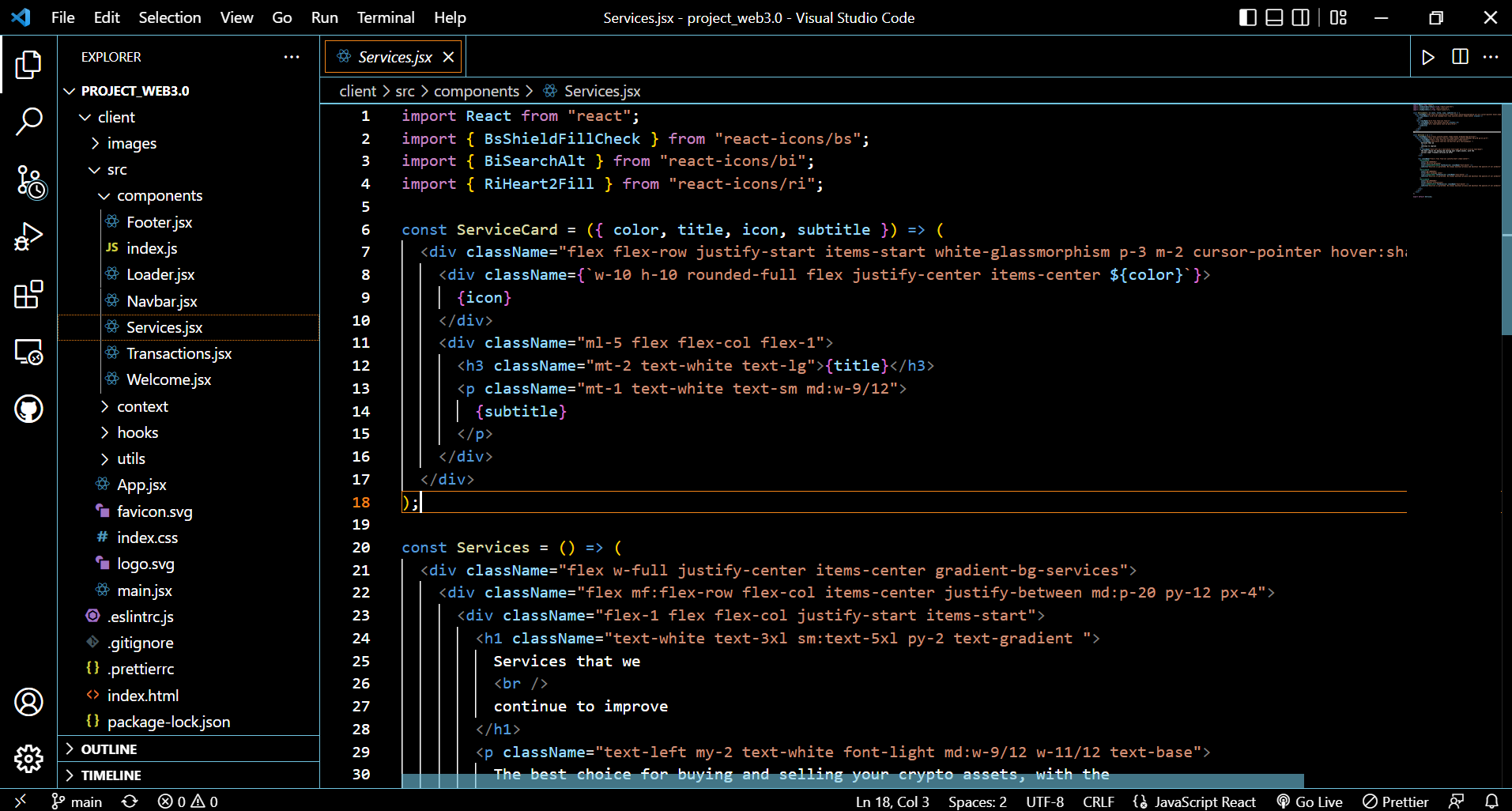
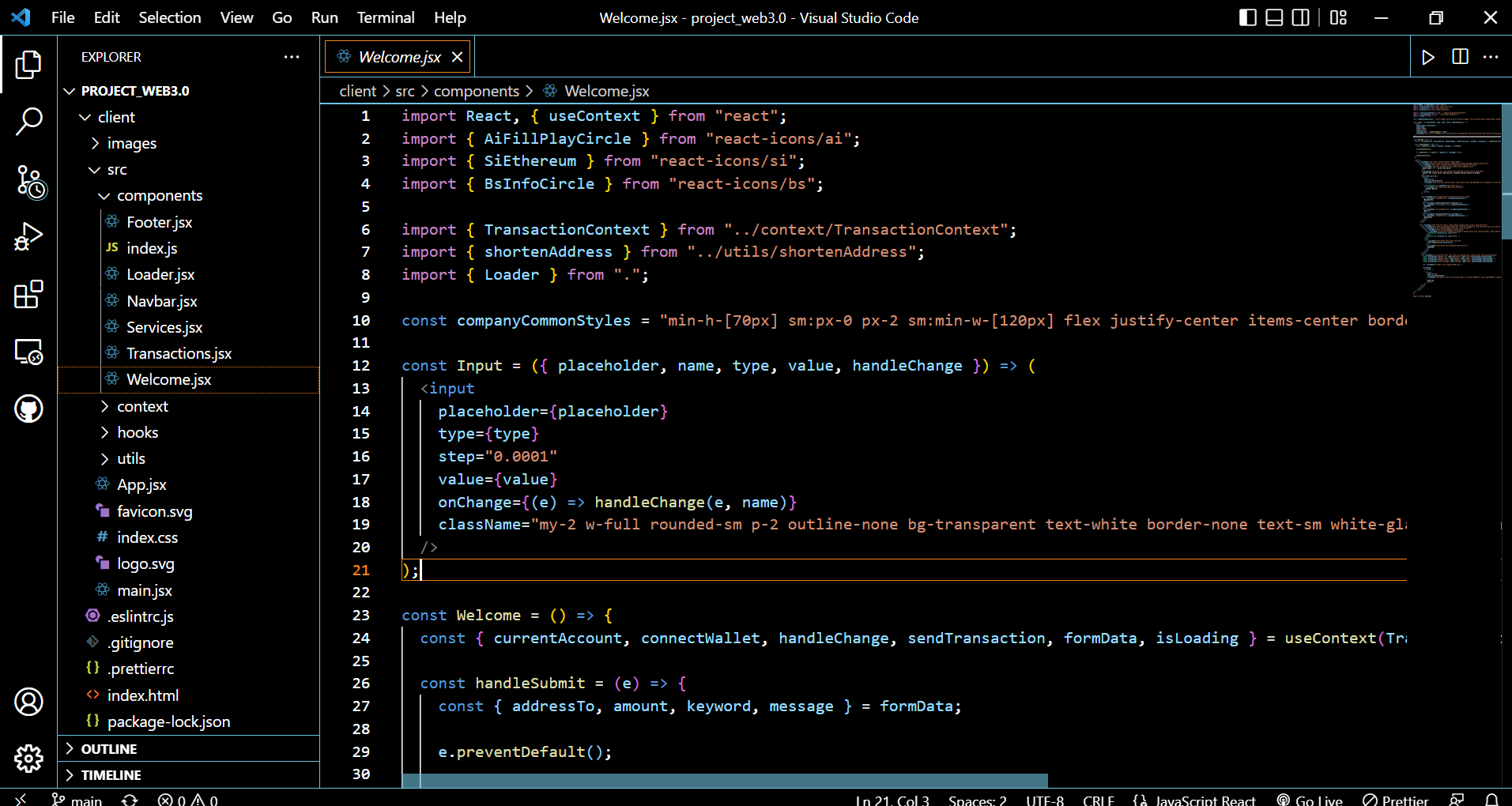
# 

# ***7. CODES AND IMPLEMENTATION***

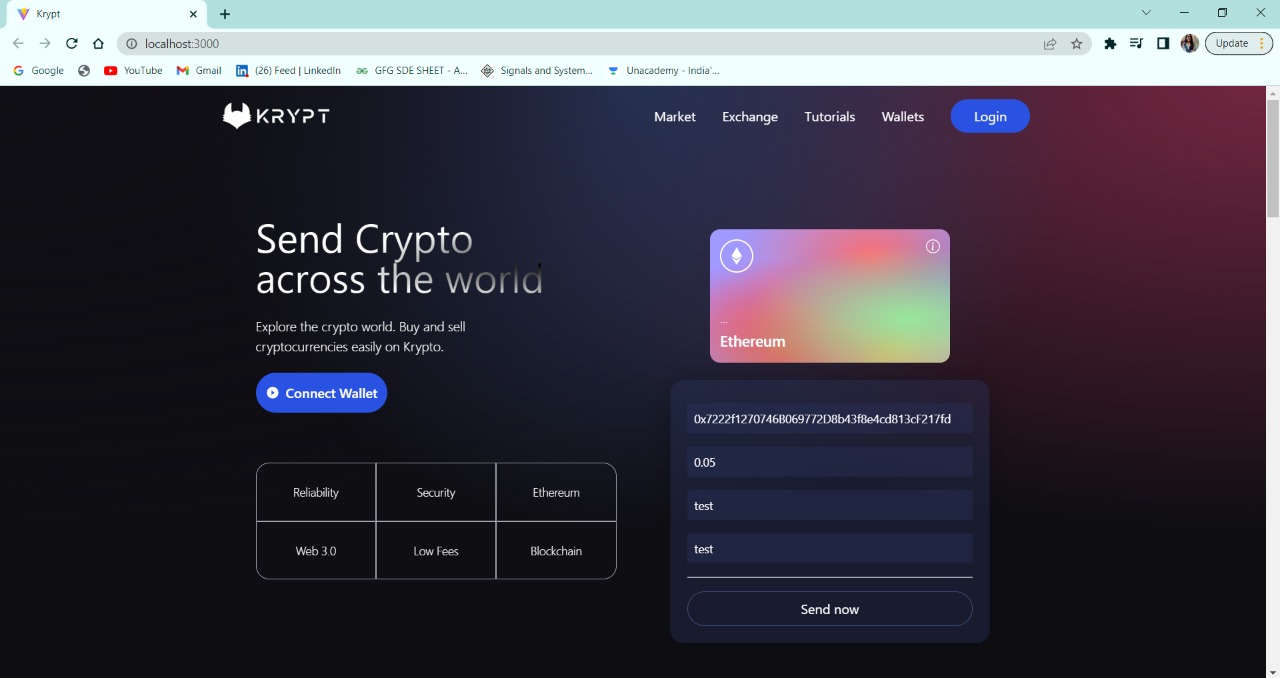
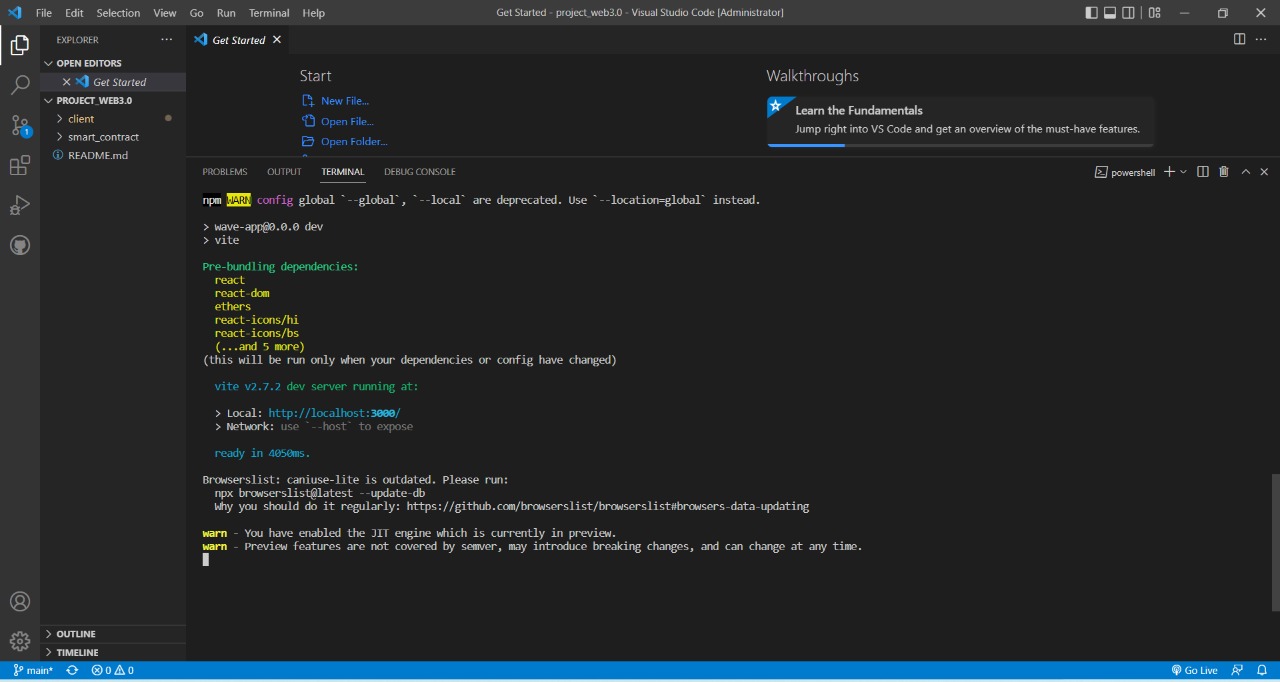
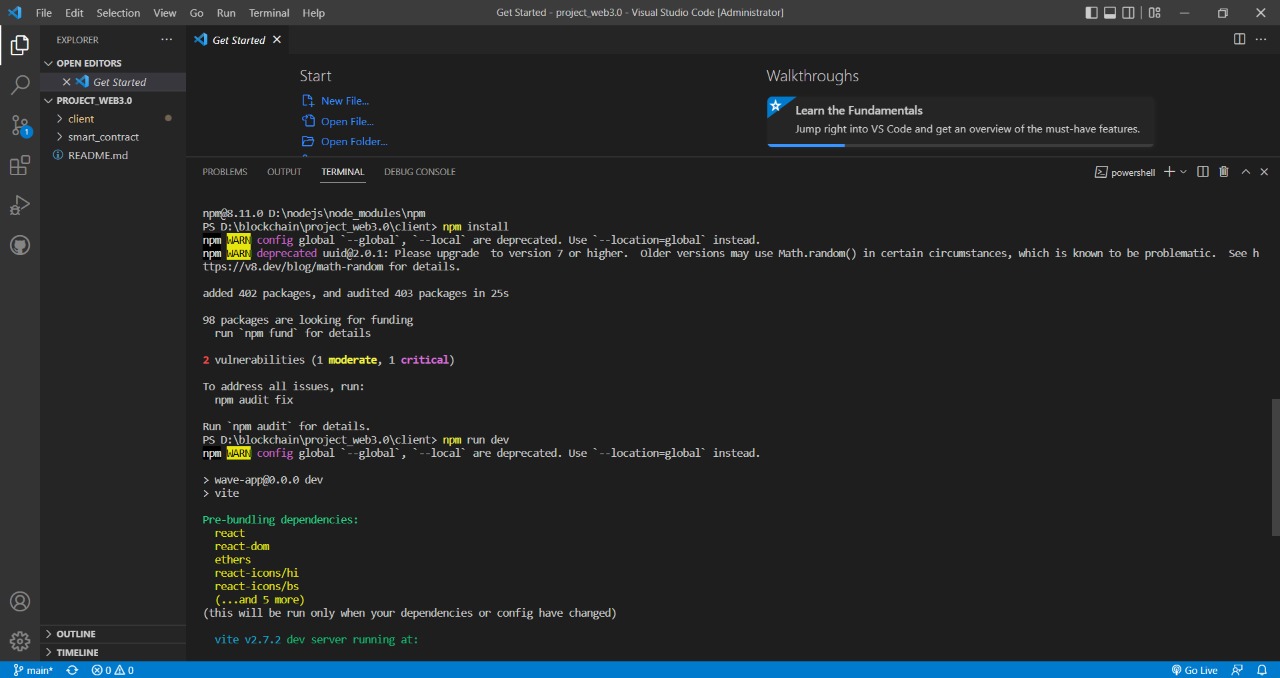
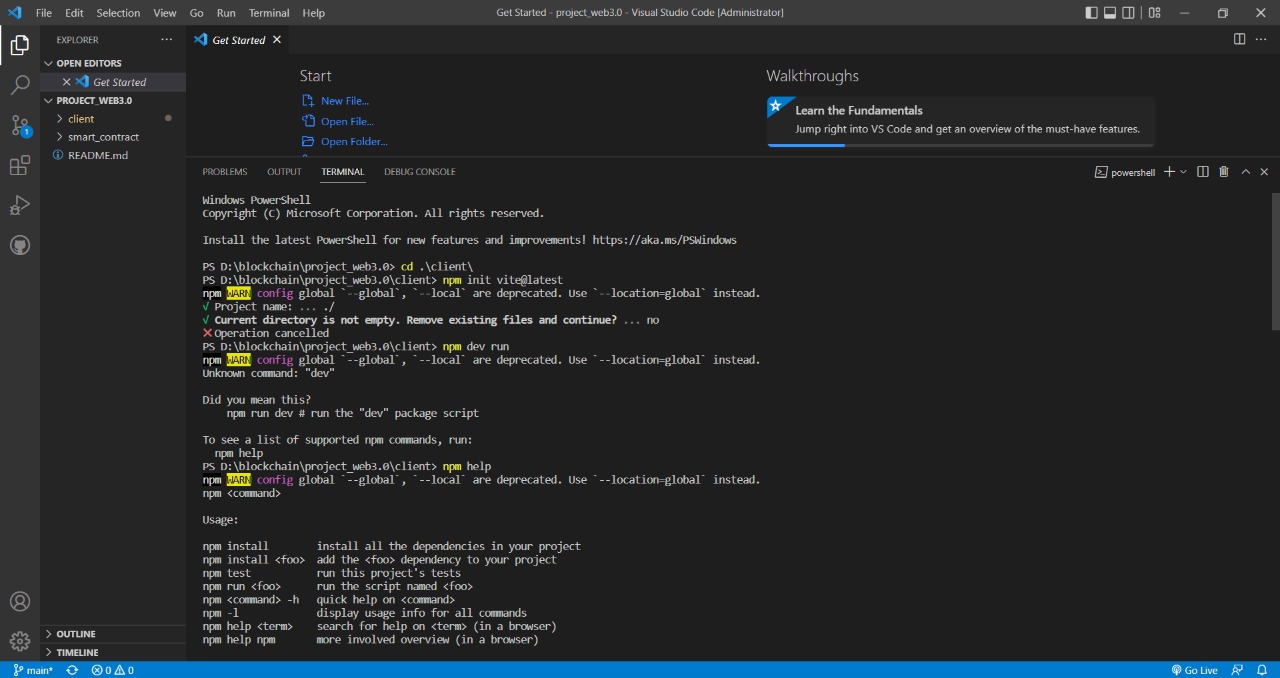
This section contains code outlets and the implementation of flow of the system.

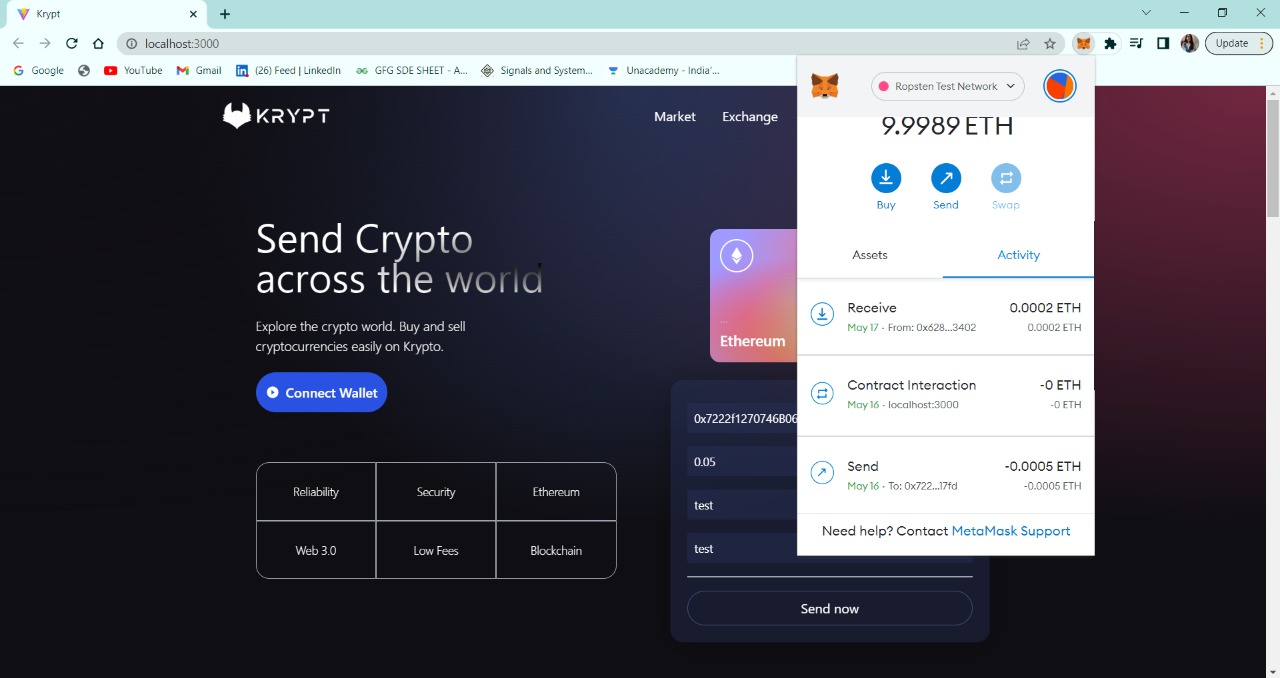
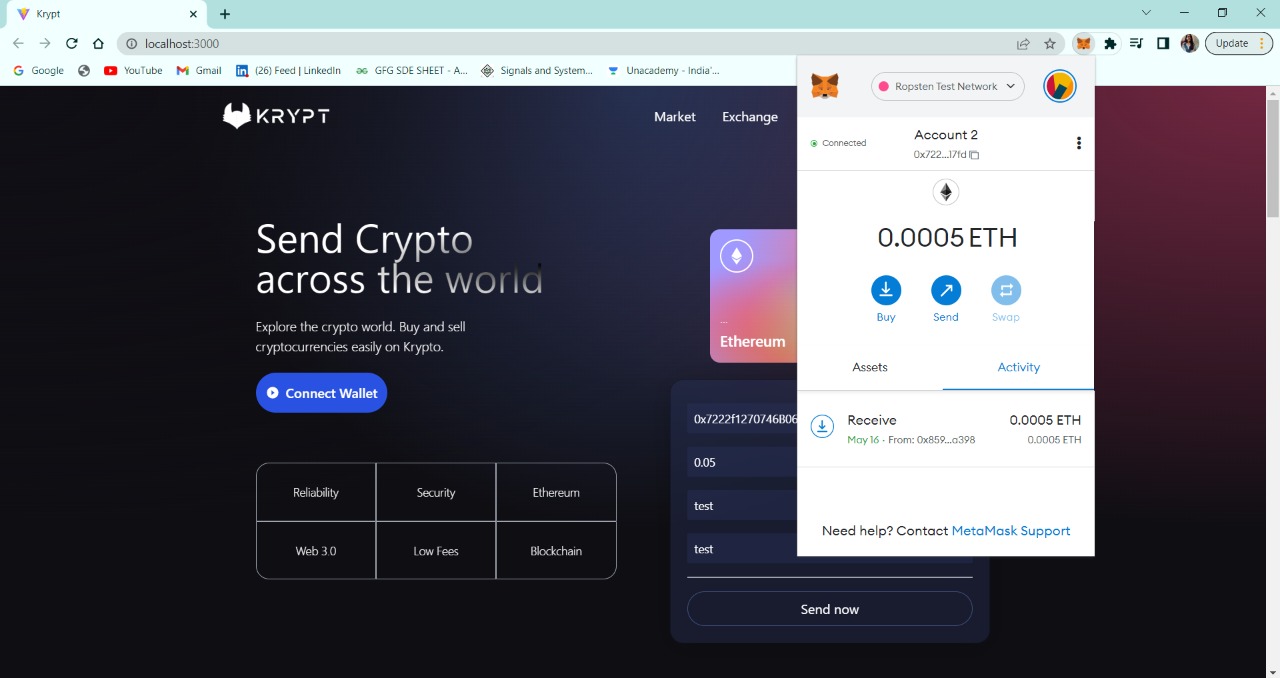
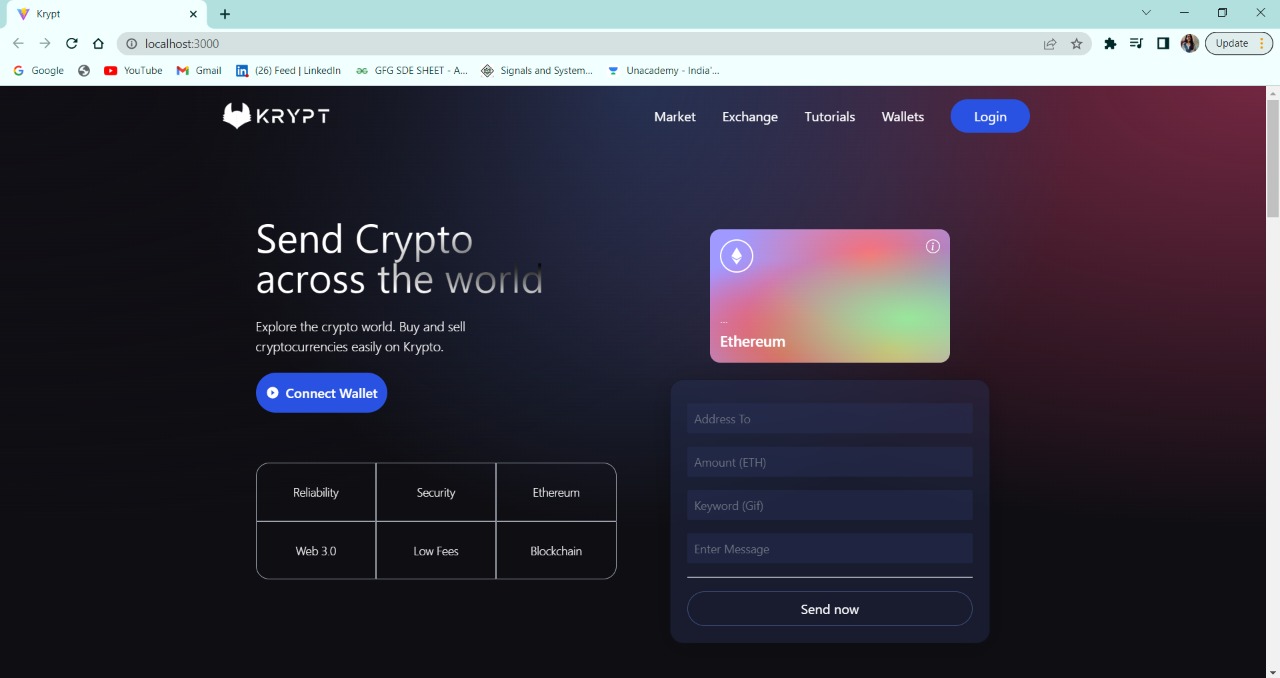






***8. OUTPUT SCREENS AND FUNCTIONING***





***9. Conclusion***

Blockchain creates an audit trail that documents the provenance of an asset at every step on its journey.

Traditional paper-heavy processes are time-consuming, prone to human error, and often require third-party mediation. By streamlining these processes with blockchain, transactions can be completed faster and more efficiently. Documentation can be stored on the blockchain along with transaction details, eliminating the need to exchange paper.

There’s no need to reconcile multiple ledgers, so clearing and settlement can be much faster.

The enhanced security offered by blockchain stems from how the technology actually works: Blockchain creates an unalterable record of transactions with end-to-end encryption, which shuts out fraud and unauthorized activity.

Thus, a blockchain based transactional system is an advancement in the day-to-day bartering of data and currency between various facilitators.