NFT – PLATFORM

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

BY
Tanishq Porwal
EN18CS301280

Under the Guidance of
Mr. Preetesh Purohit
Mr. Snehal Agrawal

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

May 2022

NFT – PLATFORM

A Project-II Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

BY

Tanishq Porwal
EN18CS301280

Under the Guidance of

Mr. Preetesh Purohit
Mr. Snehal Agrawal

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

May 2022

Report Approval

The project work **“NFT – Platform”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner
Name:
Designation:
Affiliation:

External Examiner
Name:
Designation:
Affiliation:

Declaration

I hereby declare that the project entitled **"NFT – Platform"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of **Mr. Preetesh Purohit,** Faculty of Engineering, Medi-Caps University Indore and **Mr. Snehal Agrawal, Technical Lead, Fourty Seven Billion Information Technologies Pvt. Ltd.** is an authentic work.

Further, I declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Tanishq Porwal

EN18CS301280

Date: 19/05/2022

Certificate

We, **Mr. Preetesh Purohit** and **Mr. Snehal Agrawal** certify that the project entitled **“NFT – Platform”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Tanishq Porwal** is the record carried out by him under our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Snehal

Mr. Preetesh Purohit

Computer Science and Engineering

Medi-Caps University, Indore

Mr. Snehal Agrawal

Technical Lead

Fourty Seven Billion Information Technologies Pvt. Ltd.

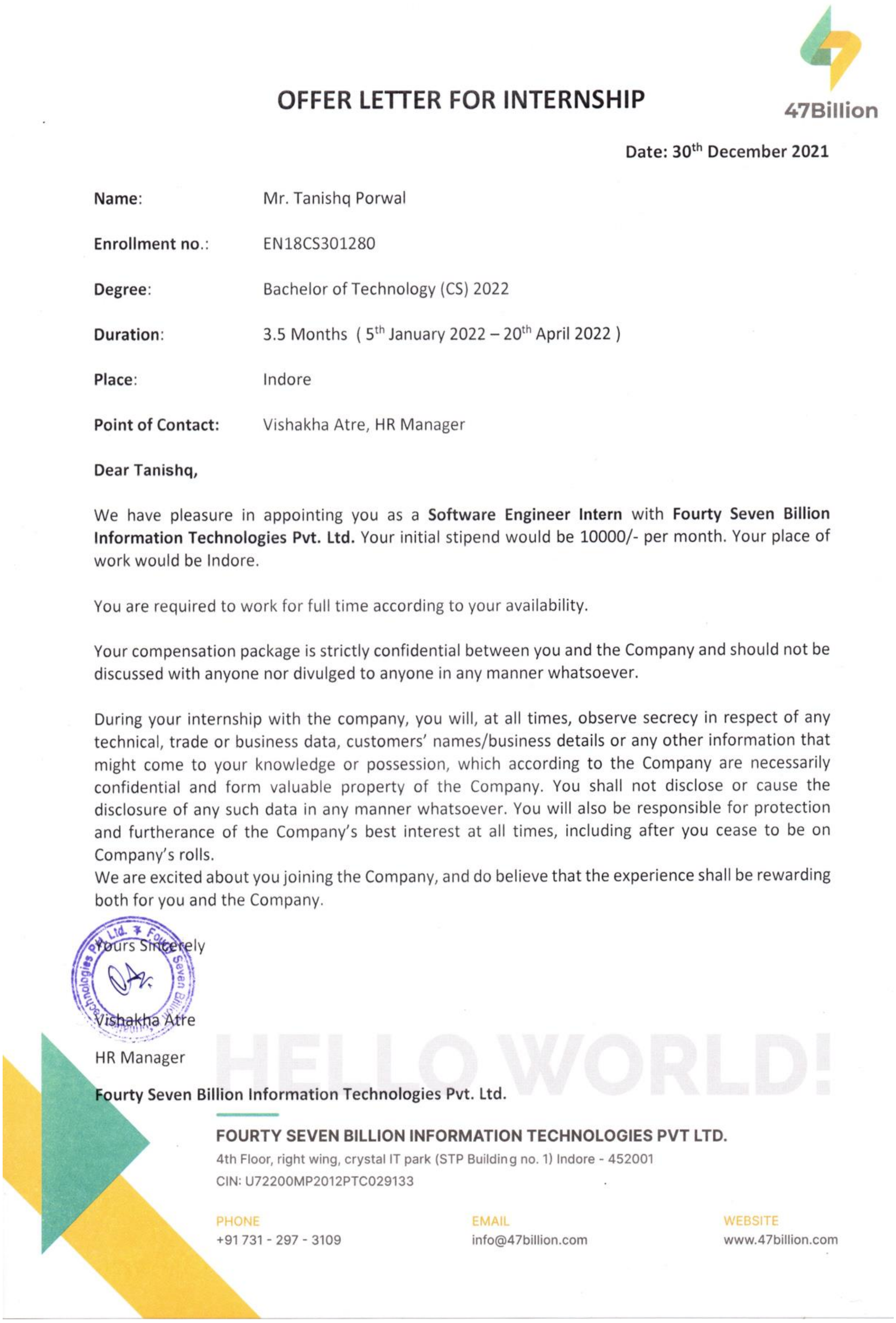
Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

<u>**Offer Letter of the Project work-II/Internship**</u>



47Billion

OFFER LETTER FOR INTERNSHIP

Date: 30th December 2021

Name:	Mr. Tanishq Porwal
Enrollment no.:	EN18CS301280
Degree:	Bachelor of Technology (CS) 2022
Duration:	3.5 Months (5th January 2022 – 20th April 2022)
Place:	Indore
Point of Contact:	Vishakha Atre, HR Manager

Dear Tanishq,

We have pleasure in appointing you as a **Software Engineer Intern** with **Fourty Seven Billion Information Technologies Pvt. Ltd.** Your initial stipend would be 10000/- per month. Your place of work would be Indore.

You are required to work for full time according to your availability.

Your compensation package is strictly confidential between you and the Company and should not be discussed with anyone nor divulged to anyone in any manner whatsoever.

During your internship with the company, you will, at all times, observe secrecy in respect of any technical, trade or business data, customers' names/business details or any other information that might come to your knowledge or possession, which according to the Company are necessarily confidential and form valuable property of the Company. You shall not disclose or cause the disclosure of any such data in any manner whatsoever. You will also be responsible for protection and furtherance of the Company's best interest at all times, including after you cease to be on Company's rolls.
We are excited about you joining the Company, and do believe that the experience shall be rewarding both for you and the Company.

Yours Sincerely

Fourty Seven Billion Technologies Pvt. Ltd.

[illegible]

Vishakha Atre

HR Manager

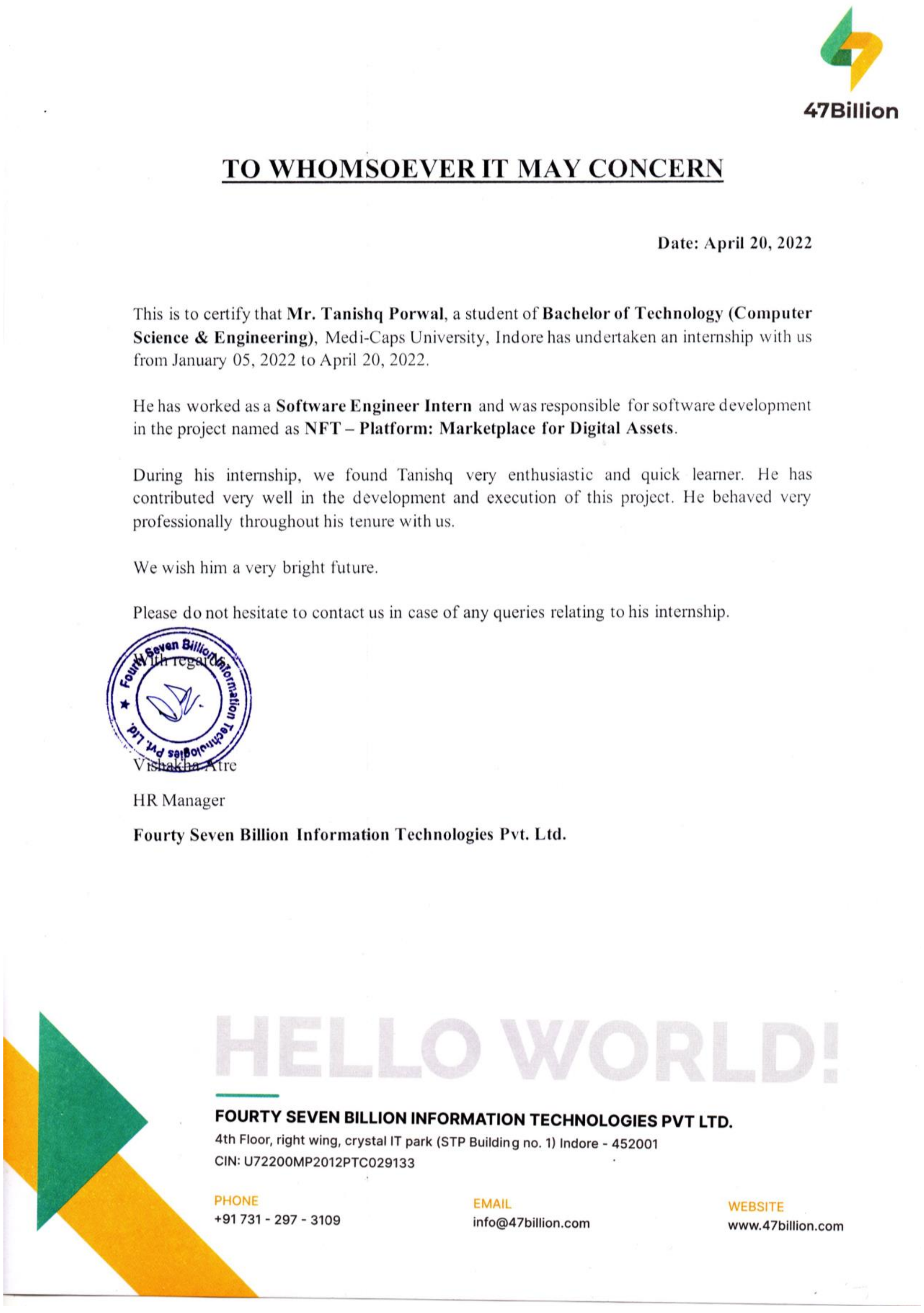
Fourty Seven Billion Information Technologies Pvt. Ltd.

HELLO WORLD!

FOURTY SEVEN BILLION INFORMATION TECHNOLOGIES PVT LTD.
4th Floor, right wing, crystal IT park (STP Building no. 1) Indore - 452001
CIN: U72200MP2012PTC029133

PHONE	EMAIL	WEBSITE
+91 731 - 297 - 3109	info@47billion.com	www.47billion.com

Completion certificate/Letter



47Billion

TO WHOMSOEVER IT MAY CONCERN

Date: April 20, 2022

This is to certify that **Mr. Tanishq Porwal**, a student of **Bachelor of Technology (Computer Science & Engineering)**, Medi-Caps University, Indore has undertaken an internship with us from January 05, 2022 to April 20, 2022.

He has worked as a **Software Engineer Intern** and was responsible for software development in the project named as **NFT – Platform: Marketplace for Digital Assets**.

During his internship, we found Tanishq very enthusiastic and quick learner. He has contributed very well in the development and execution of this project. He behaved very professionally throughout his tenure with us.

We wish him a very bright future.

Please do not hesitate to contact us in case of any queries relating to his internship.

With regards,

Fourty Seven Billion Information Technologies Pvt. Ltd. ★

Vishakha Atre

HR Manager

Fourty Seven Billion Information Technologies Pvt. Ltd.

HELLO WORLD!

FOURTY SEVEN BILLION INFORMATION TECHNOLOGIES PVT LTD.

4th Floor, right wing, crystal IT park (STP Building no. 1) Indore - 452001

CIN: U72200MP2012PTC029133

PHONE
+91 731 - 297 - 3109

EMAIL
info@47billion.com

WEBSITE
www.47billion.com

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R. C. Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dilip K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D. K. Panda,** Pro Vice Chancellor, **Dr. Suresh Jain,** Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my External Guide, **Mr. Snehal Agrawal**, Technical Lead, Fourty Seven Billion Information Technologies Pvt. Ltd. as well as to my Internal Guide, **Mr. Preetesh Purohit**, Faculty, Department of Computer Science and Engineering, MU, without whose continuous help and support, this project would ever have reached to the completion.

I would also like to thank to my team at Fourty Seven Billion Information Technologies Pvt. Ltd. who extended their kind support and help towards the completion of this project.

It is their help and support, due to which I became able to complete the design and technical report. Without their support this report would not have been possible.

Tanishq Porwal
B.Tech. IV Year
Department of Computer Science & Engineering
Faculty of Engineering
Medi-Caps University, Indore

Abstract

As a part of my undergraduate program, I joined Fourty Seven Billion Information Technologies Pvt. Ltd. on 05th January 2022 as a Software Engineer Intern. I was a part of the backend team and worked on Blockchain technology. The project was to create an NFT – Platform which will help users to get ownership of any digital art, collectibles, etc as an NFT. The ownership can be transferred in replacement of an Ethereum token (ETH) and complete transaction details will be available on the Ethereum blockchain.

Fourty Seven Billion Information Technologies Pvt. Ltd. is a product engineering services company that specializes in end-to-end product development. They have experts who understand your business domain, they will define your product requirements, visualize the product from all the touch-points of your end customers, design intuitive user interfaces, develop a solid and scalable architecture, suggest an appropriate infrastructure stack, and develop a high-quality, scalable product that is ready to go live with a large number of your customers. They are handling projects from all over the globe from the delivery centers in Santa Clara, California, Bangalore, and Indore in India.

Table of Contents

List of Figures

Abbreviations

Abbreviations	**Full Form**
NFT	Non-Fungible Token
ETH	Ether
EVM	Ethereum Virtual Machine
ERC	Ethereum Request for Comments
IPFS	InterPlanetary File System
CID	Content Identifier
URI	Uniform Resource Identifier
NVM	Node Version Manager
NPM	Node Package Manager
IDE	Integrated Development Environment

Chapter-1

Introduction

1.1 Introduction

The project NFT – Platform is a web application based on Blockchain Technology. Main business logic is written in Smart Contracts using Solidity Programming Language, which will be stored on Blockchain. Smart Contracts were created and tested in Remix IDE. The main aim of the project is to get ownership of any digital art, collectibles, etc as an NFT. The ownership can be transferred in replacement of an Ethereum token (ETH) and complete transaction detail will be available on the Ethereum blockchain.

1.2 Organization Profile

Fourty Seven Billion Information Technologies Pvt. Ltd. was incorporated on 10th September 2012. It is a product engineering services company that specializes in end-to-end product development. We have experts who understand your business domain, they will define your product requirements, visualize the product from all the touch-points of your end customers, design intuitive user interfaces, develop a solid and scalable architecture, suggest an appropriate infrastructure stack, and develop a high-quality, scalable product that is ready to go live with a large number of your customers.

We have technical expertise in various areas like big data analytics, cloud computing, IoT, mobile and web applications, User Interface, Incubation Centre, Machine Learning, and Intelligent Automation. We have successfully deployed many projects in telecom, logistics, agriculture, ad-tech, tourism, education, healthcare, ERP for SMEs, IoT, and Digital signages and Kiosks.

We are handling projects from all over the globe from our delivery centers in Santa Clara, California, Bangalore, and Indore in India. We are product partners with many well-established companies and start-ups. Some partners we are working with include Personagraph, L-Squared Networks, Cisco, Agrilife Technologies and a big game publisher and telecom provider in India. Apart from our services, we have developed in-house products in logistics, smart tourism, video conferencing and workplace collaboration. These can be deployed as they are or can be customized as per the requirements of customers.

- **Alerio:** A school bus tracking platform. Parent has a mobile app that shows child's bus in real-time.
- **Conflame:** A video conferencing and collaboration tool.
- **LSquared Networks:** Centralized cloud-based management software for digital signage and touch screen kiosk.
- **Notalytics:** Real-time analytics and Campaign Management Server.
- **72Fit:** A doctor search and appointment platform.

Website: www.47billion.com

Company Size: 100 – 200 employees

Major Clients: Cisco, aPlaceforMom, visp.net, Nazara, Apsalar, CAVO eD, L SQUARED, AGRI BOLO, Sodexo, jio and many more.

1.3 Internship Profile

My profile at 47Billion was Software Engineer Intern. I was a part of the backend team and mainly worked on Blockchain technology. I had contributed in:

- Writing smart contracts
- Working with web3.js
- Working with IPFS
- Deploying and testing smart contracts on Rinkeby Test Network

1.4 Initial Training

Initial training was given to me for a better understanding of the tools and technologies. The tools and technologies learned by me during the training period are:

- Git and GitLab
- Remix IDE
- Solidity
- IPFS
- MetaMask
- Ganache
- Truffle
- Hardhat
- JavaScript
- Web3.js
- ReactJs

Chapter-2

Blockchain

2.1 Introduction to Blockchain

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. It is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved.

Some main features of blockchain are: It runs on a decentralized system, transaction details are stored in containers called blocks, immutable and cryptographically secure. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT). Blockchain is a type of DLT in which transactions are recorded with an immutable cryptographic signature called a hash.

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.

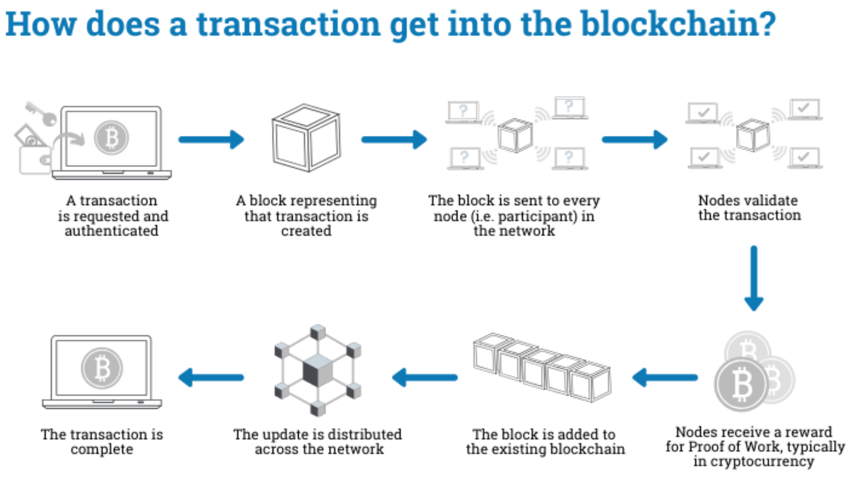
2.2 How Blockchain is different than Regular Database

One key difference between a typical database and a blockchain is how the data is structured. A blockchain collects information together in groups known as blocks, that hold sets of information. Blocks have certain storage capacities and when filled are closed and linked to the previously filled block, forming a chain of data known as the blockchain. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

A database usually structures its data into tables, whereas a blockchain as its name implies, structures its data into chunks (blocks) that are strung together. This data structure inherently makes an irreversible timeline of data when implemented in a decentralized nature. When a block is filled, it is set in the chain and becomes a part of this timeline. Each block in the chain is given an exact timestamp when it is added to the chain.

2.3 Working of Blockchain

The goal of blockchain is to allow digital information to be recorded and distributed, but not edited. In this way, blockchain is a foundation for immutable ledgers or records of transactions that cannot be altered, deleted, or destroyed.


How does a transaction get into the blockchain?
A transaction is requested and authenticated
A block representing that transaction is created
The block is sent to every node (i.e. participant) in the network
Nodes validate the transaction
Nodes receive a reward for Proof of Work, typically in cryptocurrency
The block is added to the existing blockchain
The update is distributed across the network
The transaction is complete

The use of blockchains has exploded via the creation of various cryptocurrencies, decentralized finance (DeFi) applications, non-fungible tokens (NFTs), and smart contracts.

Blockchain works via a multistep process, which in simple terms happens as follows:

1. An authorized participant inputs a transaction, which must be authenticated by the technology.
2. That action creates a block that represents that specific transaction or data.
3. The block is sent to every computer node in the network.
4. Authorized nodes verify the transaction and add the block to the existing blockchain. (Nodes in public blockchain networks are referred to as miners, they're typically paid for this task in the form of cryptocurrency.)
5. The update is distributed across the network, which finalizes the transaction.

Chapter-3

Version Control System

3.1 Introduction to Version Control System

Version control also known as source control, is the practice of tracking and managing changes to software code. They are a class of software tools that help a software team manage changes to source code over a period of time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistakes while minimizing disruption to all team members.

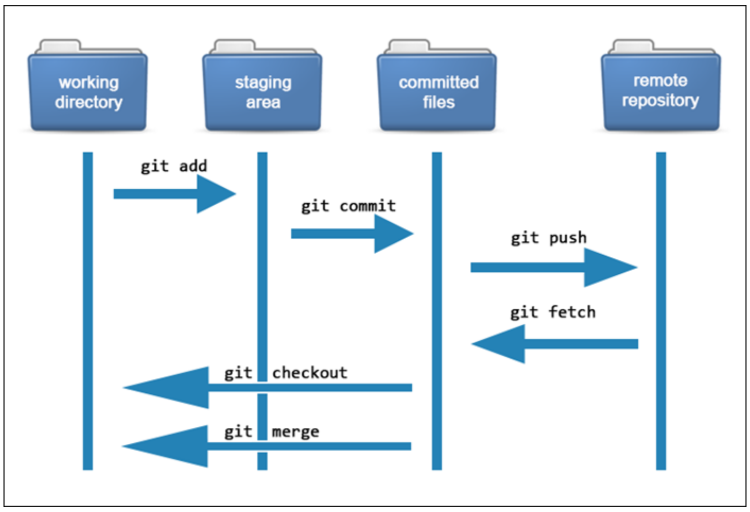
Benefits of Version Control System:

Developing software without using version control is risky, like not having backups. It enables developers to move faster and it allows software teams to preserve efficiency and agility as the team scales to include more developers. Some primary benefits are:

- A complete long-term change history of every file
- Branching and Merging
- Traceability

3.2 Git

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. A staggering number of software projects rely on Git for version control, including commercial projects as well as open-source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).



Git is the best choice for most software teams today. While every team is different and should do their own analysis, here are the main reasons why version control with Git is preferred over alternatives:

- Git has the functionality, performance, security and flexibility that most teams and individual developers need.
- Git has been designed with the integrity of the managed source code as a top priority.
- Git repositories are secured with a cryptographically secure hashing algorithm called SHA1. This protects the code and the change history against both accidental and malicious change and ensures that the history is fully traceable.
- The raw performance characteristics of Git are very strong when compared to many alternatives. Committing new changes, branching, merging and comparing past versions are all optimized for performance.
- It synchronizes code between different people and test changes to code without losing original.
- Git is a very well-supported open-source project with over a decade of solid stewardship.

Some of the useful git commands are: git init, git clone, git add, git commit, git status, git push, git pull, git fetch, git branch, git checkout, etc.

3.3 GitLab

GitLab is an open-source code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

It is The DevOps Platform that empowers organizations to maximize the overall return on software development by delivering software faster and efficiently while strengthening security and compliance. With GitLab, every team in the organization can collaboratively plan, build, secure, and deploy software to drive business outcomes faster with complete transparency, consistency and traceability across the DevOps lifecycle. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuously integration and wikis for every project.

Chapter-4

Remix IDE and NPM

4.1 What is Integrated Development Environment

An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface. An IDE typically consists of:

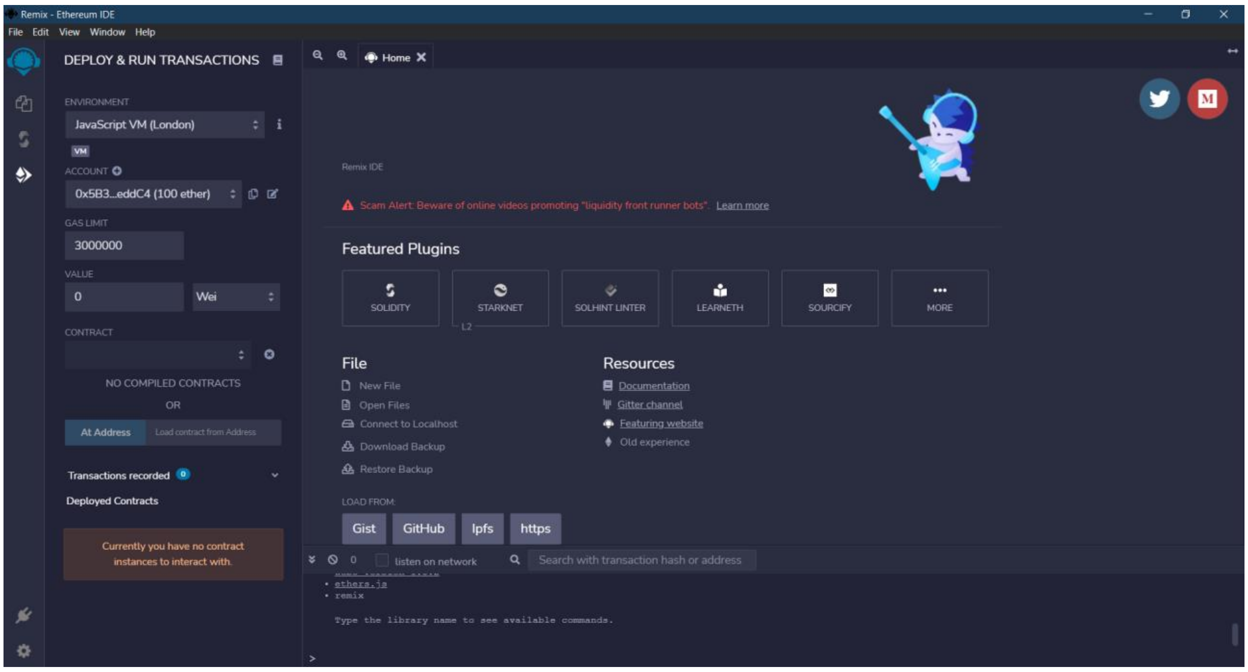
- **Source code editor**: A text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language-specific auto-completion, and checking for bugs as code is being written.
- **Local build automation**: Utilities that automate simple, repeatable tasks as part of creating a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests.
- **Debugger**: A program for testing other programs that can graphically display the location of a bug in the original code.

An IDE allows developers to start programming new applications quickly because multiple utilities don't need to be manually configured and integrated as part of the setup process. Developers also don't need to spend hours individually learning how to use different tools when every utility is represented in the same workbench. Most features of IDEs are meant to save time, like intelligent code completion and automated code generation, which removes the need to type out full character sequences.

4.2 Remix IDE

Remix IDE is an open-source web and desktop application. It fosters a fast development cycle and has a rich set of plugins with intuitive GUIs. Remix IDE allows developing, deploying and administering smart contracts for Ethereum like blockchains.

Remix IDE is a powerful open-source tool that helps us to write Solidity contracts straight from the browser. It is written in JavaScript. Remix IDE has modules for testing, debugging and deploying of smart contracts and much more.



ENVIRONMENT: JavaScript VM (London)

ACCOUNT: 0x5B3...eddC4 (100 ether)

GAS LIMIT: 3000000

VALUE: 0 Wei

CONTRACT: NO COMPILED CONTRACTS

Currently you have no contract instances to interact with.

Scam Alert: Beware of online videos promoting "liquidity front runner bots". Learn more

```
• ethers.js
• remix

Type the library name to see available commands.
```

4.3 NVM and NPM

NVM also known as Node Version Manager, is a command line tool that allows you to manage different versions of NodeJS. It provides easy installation, switching between versions and retains globally installed packages for each version.

NPM is a package manager, which helps you install libraries, plugins, frameworks and applications.

Once NVM is installed we can use this command **nvm install version** to install different versions of npm.

4.4 Project Setup

Doing a project setup and following a proper structure is very important because it allows us to follow a proper standard so that we can easily debug the code and if required in the future we can easily make changes in the code.

Follow these setups to setup a project:

1. Create a new directory with the project name using the command **mkdir project_name**.
2. Go to the directory with the help of Terminal (You must have nvm and npm installed).
3. Run the command **npm init,** It will create a package.json and package-lock.json file.
4. To install the dependencies/packages require in the project, use this command **npm install package_name --save.** It will automatically update the package.json and package-lock.json file.
5. You will also notice that it will automatically create a **node_modules** folder in the directory which will contain all the packages/dependencies required for the project.
6. After installing packages run the command **truffle init** which will create a truffle-config.js file which will help in deploying smart contracts.
7. Now you are good to go, you can start working on project.

Chapter-5

Solidity and Smart Contracts

5.1 Solidity

Solidity is an object-oriented, high-level language for implementing smart contracts. It is a curly-bracket language designed to target the Ethereum Virtual Machine. It is influenced by C++, Python and JavaScript.

It is statically typed, supports inheritance, libraries and complex user-defined types among other features. With Solidity, you can create contracts for uses such as voting, crowdfunding, blind auctions, multi-signature wallets, etc.

While deploying contracts, you should use the latest released version of Solidity. Apart from exceptional cases, only the latest version receives security fixes. There are several integration platforms (to compile, execute and run code) that implement Solidity, including Remix, which is an easily available browser and desktop-based IDE. Moreover, it also supports various type-safe functions, which means the compiler will validate types and produce an error if the wrong type is assigned to a variable.

5.1.1 Ethereum Virtual Machine

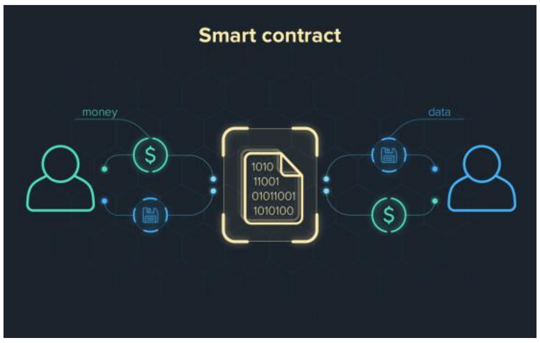
Ethereum is a decentralized blockchain platform that runs smart contracts, applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third-party interference.

EVM is the runtime environment for smart contracts in Ethereum. EVM focuses on providing security and executing untrusted code by computers all over the world. The EVM specialised in preventing Denial-of-service attacks and ensures that programs do not have access to each other's state, ensuring communication can be established without any potential interference.

It is not only sandboxed but completely isolated, which means that code running inside the EVM has no access to the network, filesystem or other processes. Smart contracts even have limited access to other smart contracts.

5.2 Smart Contracts

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. These transactions are trackable and irreversible. The concept of smart contracts was first proposed by Nick Szabo in 1994.



Smart Contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss.

A contract in the sense of Solidity is a collection of code (its functions) and data (its state) that resides at a specific address on the Ethereum blockchain. When a contract is deployed, its constructor (a function declared with the constructor keyword) is executed once. Once deployed, there is no way to invoke the constructor again. It is not necessary to have a constructor in the smart contract, as it is optional. Only one constructor is allowed, which means overloading is not supported.

After the constructor has executed, the final code of the contract is stored on the blockchain. This code includes all public and external functions and all functions that are reachable from there through function calls. The deployed code does not include the constructor code or internal functions only called from the constructor.

Chapter-6

IPFS

6.1 What is IPFS

IPFS (InterPlanetary File System) is a distributed system for storing and accessing files, websites, applications, and data. It is basically a peer-to-peer (p2p) storage network. Content is accessible through peers located anywhere in the world, that might relay information, store it, or do both.

IPFS knows how to find what you ask for using its content address rather than its location.

6.2 Importance and working of IPFS

IPFS helps to keep the data decentralized. It is decentralized because it loads the content from thousands of peers instead of one centralized server.

Every piece of data is cryptographically hashed, resulting in a safe unique content identifier **CID**. The integrity of IPFS content can be cryptographically verified and the IPFS content is de-duplicated. If we try to store the same file again in the IPFS node, they would be stored only once, eliminating the duplication, because their hash would produce an identical CID.

There are three fundamental principles to understanding IPFS:

1. Unique identification via content addressing
2. Content linking via directed acyclic graphs (DAGs)

3. Content discovery via distributed hash tables (DHTs)

These three principles build upon each other to enable the IPFS ecosystem.

IPFS uses **content addressing** to identify content by what's in it rather than by where it's located.

IPFS uses a **Merkle DAG** that is optimized for representing directories and files. To build a Merkle DAG representation of your content, IPFS often first splits it into blocks. Splitting it into blocks means that different parts of the file can come from different sources and get authenticated quickly.

To find which peers are hosting the content you want, IPFS uses a **distributed hash table**. A hash table is a database of keys to values. A distributed hash table is one where the table is split across all the peers in a distributed network.

IPFS works by connecting all devices on the network to the same file structure. This file structure is a Merkle DAG, which combines Merkle trees (used in blockchains to ensure immutability), and Directed Acyclic Graphs (used in Git version control, which also allows users to see the versions of content on IPFS). Think of it as a large BitTorrent swarm. Imagine you want to read the IPFS whitepaper. What you would normally do is type in a URL, which can be resolved to an IP address, which provides information about the location of the file (would be the IPFS servers, if they had those), which then allows your client to make a connection with the host and get the file. Now, imagine accessing it from the IPFS network. The file and all of its blocks, are identified by a unique cryptographic hash of the content itself. The whole system is based around a key-value data store. This is what allows for content addressing, anyone can host the key no matter the origin of the information. So, you would connect to the swarm and request to the network that file. It would first look to the peers closest to you, because chances are they have a copy of that file. If they don’t however, you will connect with the node that originally uploaded the file, since he’s the one that hosts it.

6.3 IPFS CID

CIDs uniquely identify a piece of content. A CID can be stored and sent over the network in a compact binary form, but they're represented as strings of random-seeming characters when displayed to users. If we try to upload the same content on IPFS, we will receive the same CID.

Here's an example of IPFS CID:

QmXHDtRs1hjqEBLniwKk5ZxMheadsXUMngwudpwDmo8qsu

6.4 IPFS URI

A Uniform Resource Identifier or URI, is used to specify a particular piece of content in a given context. The context is determined by the URI scheme (appended to the URI as a prefix, followed by ://). The URI scheme for IPFS is simply ipfs.

Here's an example of a full IPFS URI:

ipfs://QmXHDtRs1hjqEBLniwKk5ZxMheadsXUMngwudpwDmo8qsu

IPFS URIs are the canonical representation for an IPFS link, since the ipfs scheme makes it clear and unambiguous that the CID refers to content on IPFS and not some other system. To produce an IPFS URI, simply prefix a CID string with the static string ipfs://

Chapter-7

JavaScript and Web3.js

7.1 Introduction to JavaScript

JavaScript is a lightweight, cross-platform, object-oriented and interpreted scripting language which is used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc). It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments.

Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks**,** form input and page navigation. Useful libraries for the client-side are AngularJS, ReactJS, VueJS and so many others.

Server-side: It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database and provide continuity of information from one invocation to another of the application or perform file manipulations on a server. The useful framework which is the most famous these days is Node.js which allows us to add more functionality to a website than downloading files (such as real-time collaboration between multiple computers).

Inside a host environment (for example: a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them. JavaScript can be added to your HTML file in two ways:

- **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.

- **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

7.2 Callback Function

A callback is a function that is passed as an argument to another function. This technique allows a function to call another function. A callback function can run after another function has finished.

JavaScript runs code sequentially in top-down order. However, there are some cases where code runs (or must run) after something else happens and also not sequentially. This is called asynchronous programming.

Callbacks make sure that a function is not going to run before a task is completed but will run right after the task has been completed. It helps us develop asynchronous JavaScript code and keeps us safe from problems and errors.

In JavaScript, the way to create a callback function is to pass it as a parameter to another function, and then call it back right after something has happened or some task is completed.

Example:

```
function test (name)
{
        console.log ("Hello " + name + ", Welcome to Medi-Caps University");
}

function callCallback (callback)
{
        console.log("Inside callCallback Function");
        const name = "Tanishq";
        callback(name);
}

callCallback(test);
```

7.3 Async & Await Function

An async function is a function declared with the async keyword and the await keyword is permitted within it. The async and await keywords enable asynchronous, promise-based behavior to be written in a cleaner style, avoiding the need to explicitly configure promise chains.

Async, simply allows us to write promises based code as if it was synchronous and it checks that we are not breaking the execution thread. It operates asynchronously via the event-loop. Async functions always return a value. It makes sure that a promise is returned and if it is not returned then javascript automatically wraps it in a promise which is resolved with its value.

Await function is used to wait for the promise. It can be used within the async block only. It makes the code wait until the promise returns a result.

Example:

```
const test = async () =>
{
   console.log("My name is Tanishq");
   await test2();
   console.log("Hi");
}

test2 = () =>
{
   console.log("Hello");
}

test();
```

}

7.4 Web3.js

Web3.js is a collection of libraries that allow you to interact with a local or remote ethereum node using HTTP, IPC or WebSocket. It is a collection of JS libraries and helps us to communicate with Smart Contract through MetaMask.

It provides us with an API to use so we can easily work with the blockchain. Web3 works as a wrapper for JSON RPC to connect to a remote or local Ethereum node with either an HTTP or IPC connection. Web3 is basically a connection between the Ethereum blockchain and smart contract.

There are various methods provided by web3. Some of them are:

- **web3.eth.getAccounts:** It returns a list of accounts the node controls in the form of an array.
- **web3.eth.getBalance:** It returns the current balance for the given wallet address in wei.
- **web3.eth.getTransactionReceipt:** It returns the receipt of a transaction by transaction hash. The receipt is not available for pending transactions and returns null.

Example:

```
const Web3 = require('web3');
const web3=new Web3(Web3.givenProvider||"http://localhost:8545")
console.log(web3);
async function test()
{
        let showAccountBalance = await
        web3.eth.getBalance("0x0726e3E7FfA7118465aF0c734858E69B35500F95");
        console.log(showAccountBalance);
}
test();
```

Chapter-8

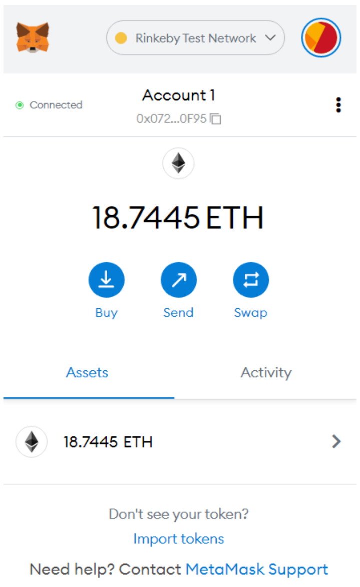
Tools

8.1 MetaMask

MetaMask is a crypto wallet and gateway to blockchain apps. It allows users to manage accounts and their keys in a variety of ways, including hardware wallets while isolating them from the site context. It is available as a browser extension and as a mobile app. MetaMask equips you with a key vault, secure login, token wallet, and token exchange—everything you need to manage your digital assets.

MetaMask provides the simplest yet most secure way to connect to blockchain-based applications. You are always in control when interacting on the new decentralized web. It generates passwords and keys on your device, so only you have access to your accounts and data.

It comes pre-loaded with fast connections to the Ethereum blockchain and several test networks such as Rinkeby (An Ethereum test network that allows for blockchain development testing before deployment on Mainnet, the main Ethereum network). This allows you to get started without synchronizing a full node, while still providing the option to upgrade your security and use the blockchain provider of your choice.



Today, MetaMask is compatible with any blockchain that exposes an Ethereum-compatible JSON RPC API, including custom and private blockchains. For development, we recommend running a test blockchain like Ganache.

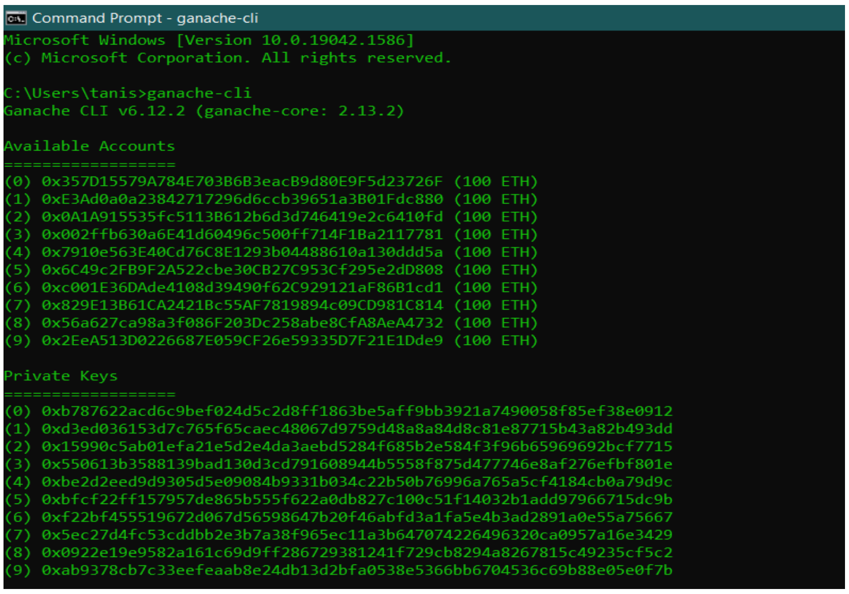
8.2 Ganache

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle: enabling you to develop, deploy, and test your dApps in a safe and deterministic environment.

Ganache comes in two types: a UI and CLI. Ganache UI is a desktop application supporting both Ethereum and Corda technology. The command-line tool, ganache-cli (formerly known as the TestRPC) is available for Ethereum development.

In short, ganache is your local blockchain simulator which quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.

After installing ganache, open the terminal and run the command **ganache-cli** and you will get some accounts with their private key which you can import it into MetaMask.

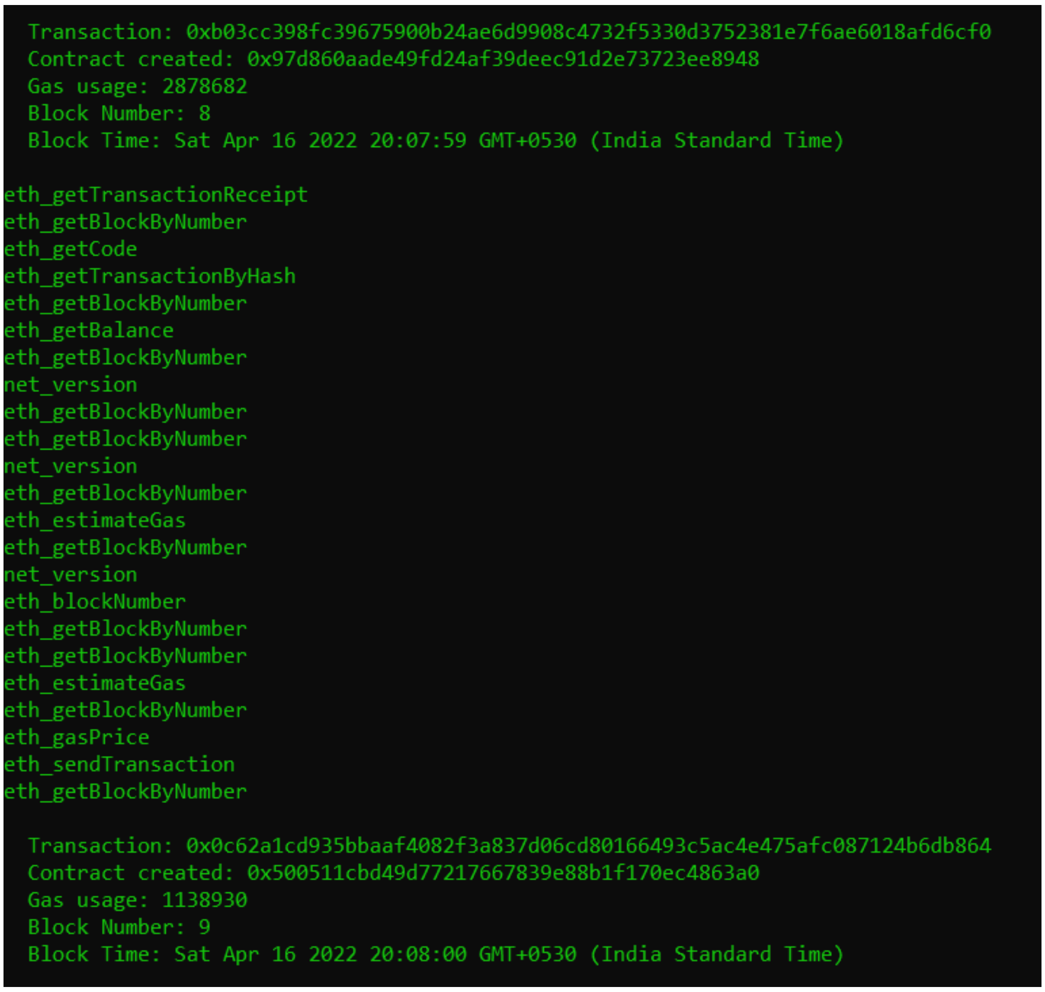

```
Command Prompt - ganache-cli
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tanis>ganache-cli
Ganache CLI v6.12.2 (ganache-core: 2.13.2)

Available Accounts
==================
(0) 0x357D15579A784E703B6B3eacB9d80E9F5d23726F (100 ETH)
(1) 0xE3Ad0a0a23842717296d6ccb39651a3B01Fdc880 (100 ETH)
(2) 0x0A1A915535fc5113B612b6d3d746419e2c6410fd (100 ETH)
(3) 0x002ffb630a6E41d60496c500ff714F1Ba2117781 (100 ETH)
(4) 0x7910e563E40Cd76C8E1293b04488610a130ddd5a (100 ETH)
(5) 0x6C49c2FB9F2A522cbe30CB27C953Cf295e2dD808 (100 ETH)
(6) 0xc001E36DAde4108d39490f62C929121aF86B1cd1 (100 ETH)
(7) 0x829E13B61CA2421Bc55AF7819894c09CD981C814 (100 ETH)
(8) 0x56a627ca98a3f086F203Dc258abe8CfA8AeA4732 (100 ETH)
(9) 0x2EeA513D0226687E059CF26e59335D7F21E1Dde9 (100 ETH)

Private Keys
==================
(0) 0xb787622acd6c9bef024d5c2d8ff1863be5aff9bb3921a7490058f85ef38e0912
(1) 0xd3ed036153d7c765f65caec48067d9759d48a8a84d8c81e87715b43a82b493dd
(2) 0x15990c5ab01efa21e5d2e4da3aebd5284f685b2e584f3f96b65969692bcf7715
(3) 0x550613b3588139bad130d3cd791608944b5558f875d477746e8af276efbf801e
(4) 0xbe2d2eed9d9305d5e09084b9331b034c22b50b76996a765a5cf4184cb0a79d9c
(5) 0xbfcf22ff157957de865b555f622a0db827c100c51f14032b1add97966715dc9b
(6) 0xf22bf455519672d067d56598647b20f46abfd3a1fa5e4b3ad2891a0e55a75667
(7) 0x5ec27d4fc53cddbb2e3b7a38f965ec11a3b647074226496320ca0957a16e3429
(8) 0x0922e19e9582a161c69d9ff286729381241f729cb8294a8267815c49235cf5c2
(9) 0xab9378cb7c33eefeaab8e24db13d2bfa0538e5366bb6704536c69b88e05e0f7b
```

Figure 8.1: Ganache Accounts with Private Keys



```
  Transaction: 0xb03cc398fc39675900b24ae6d9908c4732f5330d3752381e7f6ae6018afd6cf0
  Contract created: 0x97d860aade49fd24af39deec91d2e73723ee8948
  Gas usage: 2878682
  Block Number: 8
  Block Time: Sat Apr 16 2022 20:07:59 GMT+0530 (India Standard Time)

eth_getTransactionReceipt
eth_getBlockByNumber
eth_getCode
eth_getTransactionByHash
eth_getBlockByNumber
eth_getBalance
eth_getBlockByNumber
net_version
eth_getBlockByNumber
eth_getBlockByNumber
net_version
eth_getBlockByNumber
eth_estimateGas
eth_getBlockByNumber
net_version
eth_blockNumber
eth_getBlockByNumber
eth_getBlockByNumber
eth_estimateGas
eth_getBlockByNumber
eth_gasPrice
eth_sendTransaction
eth_getBlockByNumber

  Transaction: 0x0c62a1cd935bbaaf4082f3a837d06cd80166493c5ac4e475afc087124b6db864
  Contract created: 0x500511cbd49d77217667839e88b1f170ec4863a0
  Gas usage: 1138930
  Block Number: 9
  Block Time: Sat Apr 16 2022 20:08:00 GMT+0530 (India Standard Time)
```

Figure 8.2: Transaction Details on Ganache

8.3 Truffle

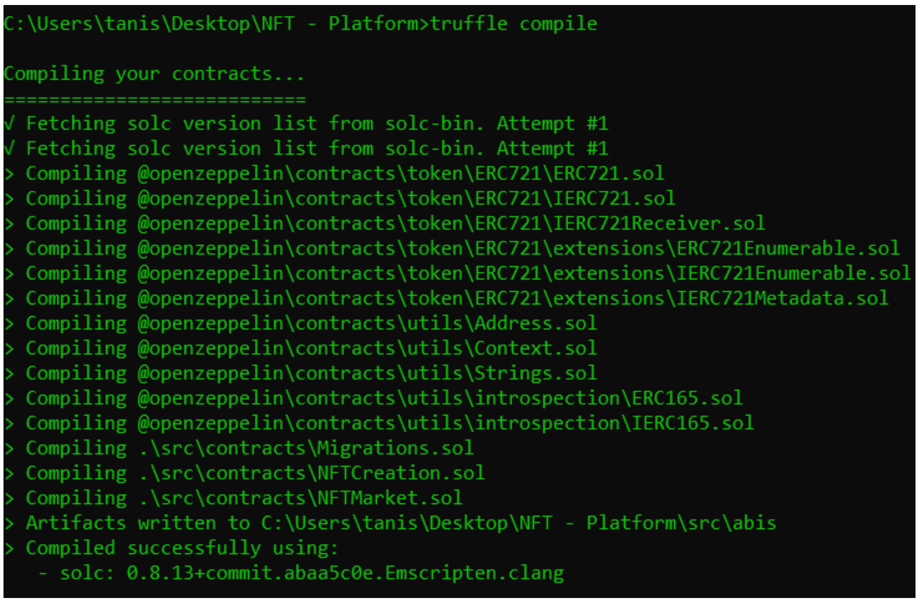
Truffle is the most popular development tooling for Ethereum programmers. With truffle, we can easily deploy smart contracts and communicate with their underlying state without heavy client-side programming. It is an especially useful library for the testing and iteration of Ethereum smart contracts.

truffle config.js: It is a file that is used to configure the truffle project. It is seeded with some common settings for different networks and features like migration, compilation and testing.

We uncomment the ones we need or modify it according to the project. This file is located at the root of project directory.

Some basic commands of Truffle are:

- **truffle compile:** Used to compile the smart contracts.

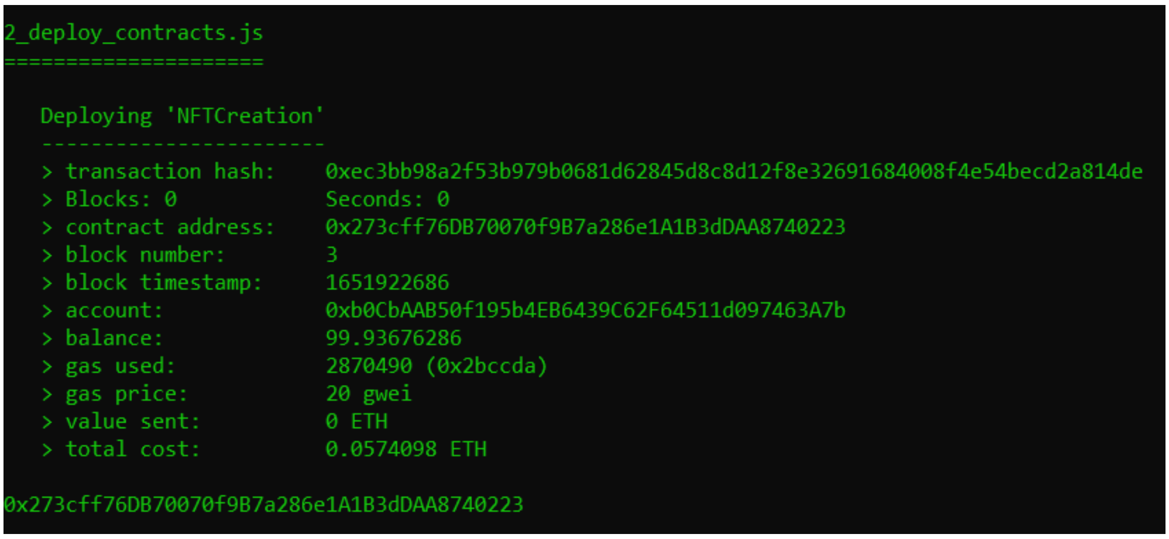


```
C:\Users\tanis\Desktop\NFT - Platform>truffle compile

Compiling your contracts...
===========================
√ Fetching solc version list from solc-bin. Attempt #1
√ Fetching solc version list from solc-bin. Attempt #1
> Compiling @openzeppelin\contracts\token\ERC721\ERC721.sol
> Compiling @openzeppelin\contracts\token\ERC721\IERC721.sol
> Compiling @openzeppelin\contracts\token\ERC721\IERC721Receiver.sol
> Compiling @openzeppelin\contracts\token\ERC721\extensions\ERC721Enumerable.sol
> Compiling @openzeppelin\contracts\token\ERC721\extensions\IERC721Enumerable.sol
> Compiling @openzeppelin\contracts\token\ERC721\extensions\IERC721Metadata.sol
> Compiling @openzeppelin\contracts\utils\Address.sol
> Compiling @openzeppelin\contracts\utils\Context.sol
> Compiling @openzeppelin\contracts\utils\Strings.sol
> Compiling @openzeppelin\contracts\utils\introspection\ERC165.sol
> Compiling @openzeppelin\contracts\utils\introspection\IERC165.sol
> Compiling .\src\contracts\Migrations.sol
> Compiling .\src\contracts\NFTCreation.sol
> Compiling .\src\contracts\NFTMarket.sol
> Artifacts written to C:\Users\tanis\Desktop\NFT - Platform\src\abis
> Compiled successfully using:
   - solc: 0.8.13+commit.abaa5c0e.Emscripten.clang
```

Figure 8.3: Compiling Smart Contracts Using Truffle

- **truffle migrate:** Used to deploy the smart contracts on the local blockchain network.
- **truffle migrate --reset:** It runs all migrations from the beginning instead of running from the last completed migration.



```
2_deploy_contracts.js
=====================

   Deploying 'NFTCreation'
   -----------------------
   > transaction hash:    0xec3bb98a2f53b979b0681d62845d8c8d12f8e32691684008f4e54becd2a814de
   > Blocks: 0            Seconds: 0
   > contract address:    0x273cff76DB70070f9B7a286e1A1B3dDAA8740223
   > block number:        3
   > block timestamp:     1651922686
   > account:             0xb0CbAAB50f195b4EB6439C62F64511d097463A7b
   > balance:             99.93676286
   > gas used:            2870490 (0x2bccda)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.0574098 ETH

0x273cff76DB70070f9B7a286e1A1B3dDAA8740223
```

Figure 8.4: Deploying Smart Contracts Using Truffle

8.4 Hardhat

Hardhat is a development environment to compile, deploy, test, and debug your Ethereum software. It helps developers manage and automate the recurring tasks that are inherent to the process of building smart contracts and dApps, as well as easily introducing more functionality around this workflow. This means compiling, running and testing smart contracts at the very core. It is built by the Nomic Foundation for the Ethereum Community.

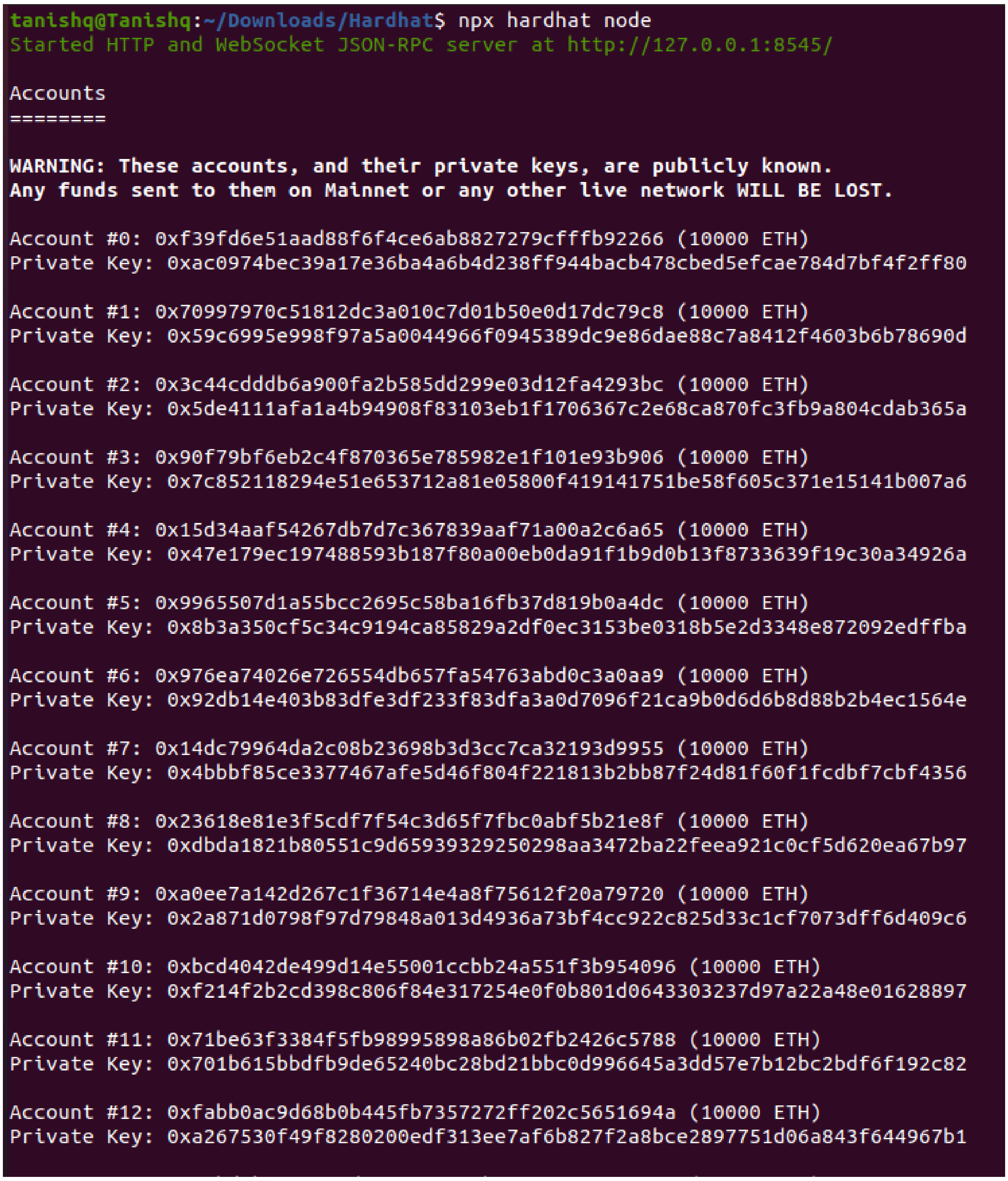
Hardhat comes built-in with Hardhat Network, a local Ethereum network designed for development. Its functionality focuses around Solidity debugging, featuring stack traces, console.log () and explicit error messages when transactions fail.

Hardhat Runner, the CLI command to interact with Hardhat is an extensible task runner. It's designed around the concepts of tasks and plugins. Every time you're running Hardhat from the CLI you're running a task. E.g. npx hardhat compile is running the built-in compile task. Tasks can call other tasks, allowing complex workflows to be defined. Users and plugins can override existing tasks, making those workflows customizable and extendable.

A lot of Hardhat's functionality comes from plugins and as a developer, you're free to choose which ones you want to use. Hardhat is unopinionated in terms of what tools you end up using, but it does come with some built-in defaults all of which can be overridden.

Hardhat is used through a local installation in your project. This way your environment will be reproducible, and you will avoid future version conflicts. To install it, you need to create an npm project by going to an empty folder, running npm init, and following its instructions.

Once your project is ready, you should run **npm install --save dev hardhat** and then run the command **npx hardhat node** which will give you some accounts (which will have some Ethers for testing) with their private keys and will also create a local blockchain network.



```
tanishq@Tanishq:~/Downloads/Hardhat$ npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
========

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970c51812dc3a010c7d01b50e0d17dc79c8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3c44cdddb6a900fa2b585dd299e03d12fa4293bc (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90f79bf6eb2c4f870365e785982e1f101e93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34aaf54267db7d7c367839aaf71a00a2c6a65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507d1a55bcc2695c58ba16fb37d819b0a4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba

Account #6: 0x976ea74026e726554db657fa54763abd0c3a0aa9 (10000 ETH)
Private Key: 0x92db14e403b83dfe3df233f83dfa3a0d7096f21ca9b0d6d6b8d88b2b4ec1564e

Account #7: 0x14dc79964da2c08b23698b3d3cc7ca32193d9955 (10000 ETH)
Private Key: 0x4bbbf85ce3377467afe5d46f804f221813b2bb87f24d81f60f1fcdbf7cbf4356

Account #8: 0x23618e81e3f5cdf7f54c3d65f7fbc0abf5b21e8f (10000 ETH)
Private Key: 0xdbda1821b80551c9d65939329250298aa3472ba22feea921c0cf5d620ea67b97

Account #9: 0xa0ee7a142d267c1f36714e4a8f75612f20a79720 (10000 ETH)
Private Key: 0x2a871d0798f97d79848a013d4936a73bf4cc922c825d33c1cf7073dff6d409c6

Account #10: 0xbcd4042de499d14e55001ccbb24a551f3b954096 (10000 ETH)
Private Key: 0xf214f2b2cd398c806f84e317254e0f0b801d0643303237d97a22a48e01628897

Account #11: 0x71be63f3384f5fb98995898a86b02fb2426c5788 (10000 ETH)
Private Key: 0x701b615bbdfb9de65240bc28bd21bbc0d996645a3dd57e7b12bc2bdf6f192c82

Account #12: 0xfabb0ac9d68b0b445fb7357272ff202c5651694a (10000 ETH)
Private Key: 0xa267530f49f8280200edf313ee7af6b827f2a8bce2897751d06a843f644967b1
```

Figure 8.5: Hardhat Accounts with Private Keys

Chapter-9

ReactJS

9.1 Introduction to ReactJS

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application. A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications.

9.2 Why use ReactJS?

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the applications. The JavaScript virtual DOM is faster than the regular DOM. It uses component and data patterns that improve readability and helps to maintain larger applications.

9.3 React create-react-app

Starting a new React project is very complicated, with so many build tools. It uses many dependencies, configuration files, and other requirements such as Babel, Webpack, ESLint before writing a single line of React code. **create-react-app** CLI tool removes all that complexities and makes React app simple. For this, we need to install the package using NPM, and then run a few simple commands to get a new React project.

The **create-react-app** is an excellent tool, which allows us to create and run React project very quickly. It does not take any configuration manually. This tool is wrapping all the required dependencies like Webpack, Babel, etc for React project itself. This tool sets up the development environment, provides an excellent developer experience, and optimizes the application for production.

Chapter-10

Project

10.1 Project Description

The Project is to create an NFT – Platform where:

- User will first connect with the MetaMask Wallet.
- Upload an NFT.
- NFT and its Metadata will get uploaded on the decentralized platform IPFS.
- If the user wants to sell an NFT, he/she will enter the price (in Ether).
- The one who likes NFT can buy it. 20% service fees will be charged that is 20% of the price of NFT, Ether will be transferred to the organization's wallet that provides the NFT – Platform service and the remaining Ether will be transferred to the seller's wallet.
- If the user doesn't want to sell his/her NFT, he/she can remove it from sell.
- Moreover, the user has also an option to Burn NFT. If, he/she doesn't want to sell NFT or doesn't want to be the owner of NFT.

This project is based on blockchain which will help users to get ownership of any digital art, collectibles, etc as an NFT. The ownership can be transferred in replacement of Ethereum token (ETH) and complete transaction detail will be available on the Ethereum blockchain.

10.2 Problem Statement

Blockchain has become an emerging technology and with blockchain, the concept of decentralization has also emerged which allows users to store the data decentralized. In order to maintain transparency and making it difficult or impossible to change or hack the ownership of NFTs. This NFT – Platform is created where people can create, sell and buy digital assets (NFT) whose complete data will be stored in the Ethereum Blockchain and complete transaction details can be easily viewed on the blockchain.

10.3 Objectives

The objectives of the project are:

- To provide a platform where people can create/mint digital assets as NFT like art, gifs, etc.
- To provide a platform where people can see the NFTs and buy them if they like.
- To maintain the ownership of NFT decentralized so that it becomes difficult or impossible to change, hack, or cheat the system.
- All the operations performed on the platform will be stored in Blockchain to maintain proper transparency.

10.4 What is NFT

NFT stands for a non-fungible token, which means it can neither be replaced nor interchanged because it has unique properties. It is a digital asset that represents Internet collectibles/real-world objects like art, music, and games with an authentic certificate created by blockchain technology. No one can modify the record of ownership.

Tokenizing these real-world tangible assets makes buying, selling and trading more efficient while reducing the probability of fraud. Furthermore, NFTs can also function to represent individuals' identities, property rights, and many more.

Anyone can view the images (NFTs) online for free. So why are people willing to spend millions on something they could easily screenshot or download? Because NFT allows the buyer to own the original item. Not only that, it contains built-in authentication, which serves as proof of ownership.

10.5 Tools and Technologies Used

The tools and technologies used in the project are as follows:

- **Programming Language:**	Solidity
- **Decentralized Platform:**	IPFS
- **Library:**	web3.js

- **Tools:**	Ganache, Truffle
- **IDE:**	Remix, Visual Code Studio
- **Crypto Wallet:**	MetaMask
- **Git Repository:**	GitLab

10.6 Project Architecture

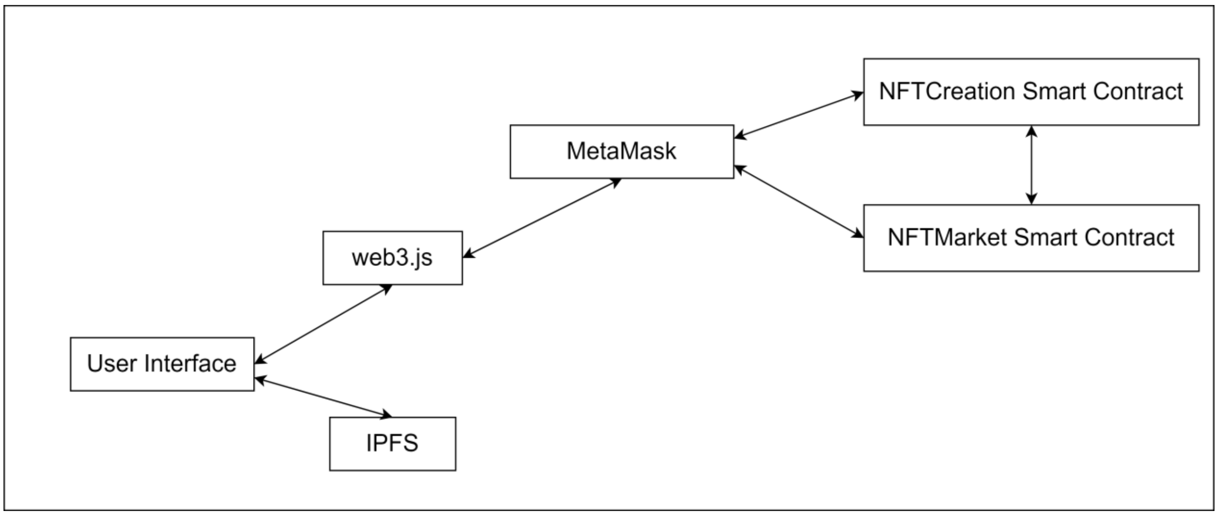


Figure 10.1

The user opens a website and connects to his MetaMask Wallet. Once the wallet his connected successfully, user can mint, sell and buy NFT. If a user's NFT is sold then the amount at which NFT sold, 20% of that amount will be transferred to Organization's Wallet and the remaining amount will be transferred to the Seller's Wallet.

Web3.js, a JavaScript library will help to communicate with smart contract through MetaMask. All the NFTs will be stored and fetched from the Ethereum Blockchain. The operations performed by a user will require a MetaMask for confirmation and to pay a gas fee as it will create a transaction on Ethereum Blockchain. Once the transaction is successful, the data will be permanently stored in the NFTCreation and NFTMarket smart contract.

Furthermore, when the user mint an NFT, NFT and its metadata will also be uploaded on IPFS (Decentralized Platform) which will create a permanent CID. This CID will be stored on blockchain and will be used to prove ownership.

NFTCreation Smart Contract for creation (minting) of NFT, Maintaining the NFTs which follows ERC 721 standard, Transfer of Ownership, etc.

NFTMarket Smart Contract for performing various operations on NFT such as Buy, Sell, Transferring of Ethers, etc.

10.7 Ethereum ERC Standards

ERC is an acronym that stands for Ethereum Request for Comments. ERCs are application-level standards for Ethereum and can include token standards, name registries, library/package formats, and more. Anyone can create an ERC, but it is the author's responsibility to clearly explain their standard and foster support for it within the community.

Common ERC standards define a required set of functions for a token type, allowing applications and smart contracts to interact with them in a predictable way. The most common ERC standard is ERC-20, a type of standard that makes the creation, use, and exchange of Ethereum-based tokens quite simple.

Crypto-enthusiasts should be aware of the issues that may arise when different projects adopt limited functionality from each standard, making their smart contracts different from a proper ERC standard.

10.7.1 ERC-721 Standard

ERC-721 is a standard for representing ownership of non-fungible tokens, that is where each token is unique. It provides functionalities like to transfer tokens from one account to another, to get the current token balance of an account, to get the owner of a specific token and also the total supply of the token available on the network. Besides these it also has some other functionalities like to approve that an amount of token from an account can be moved by a third party account.

ERC721

If a Smart Contract implements the following methods and events it can be called an ERC-721 Non-Fungible Token Contract and, once deployed, it will be responsible to keep track of the created tokens on Ethereum.

All NFTs have a uint256 variable called tokenId, so for any ERC-721 Contract, the pair contract address, uint256 tokenId must be globally unique. ERC-721 is a more complex standard than ERC-20, with multiple optional extensions, and is split across a number of contracts.

10.8 Use Case Diagram

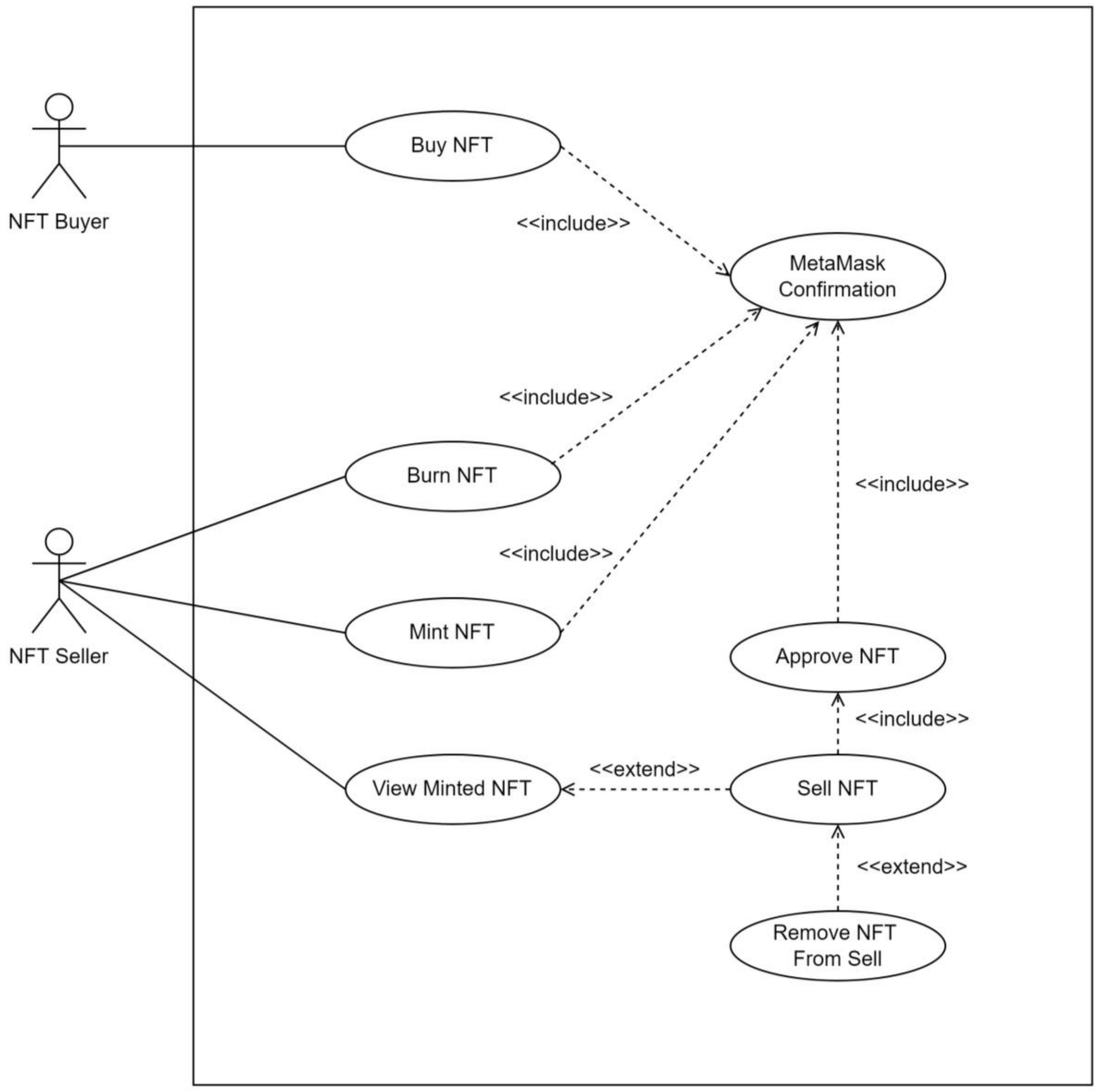


Figure 10.2

10.9 Activity Diagram

a. Mint NFT

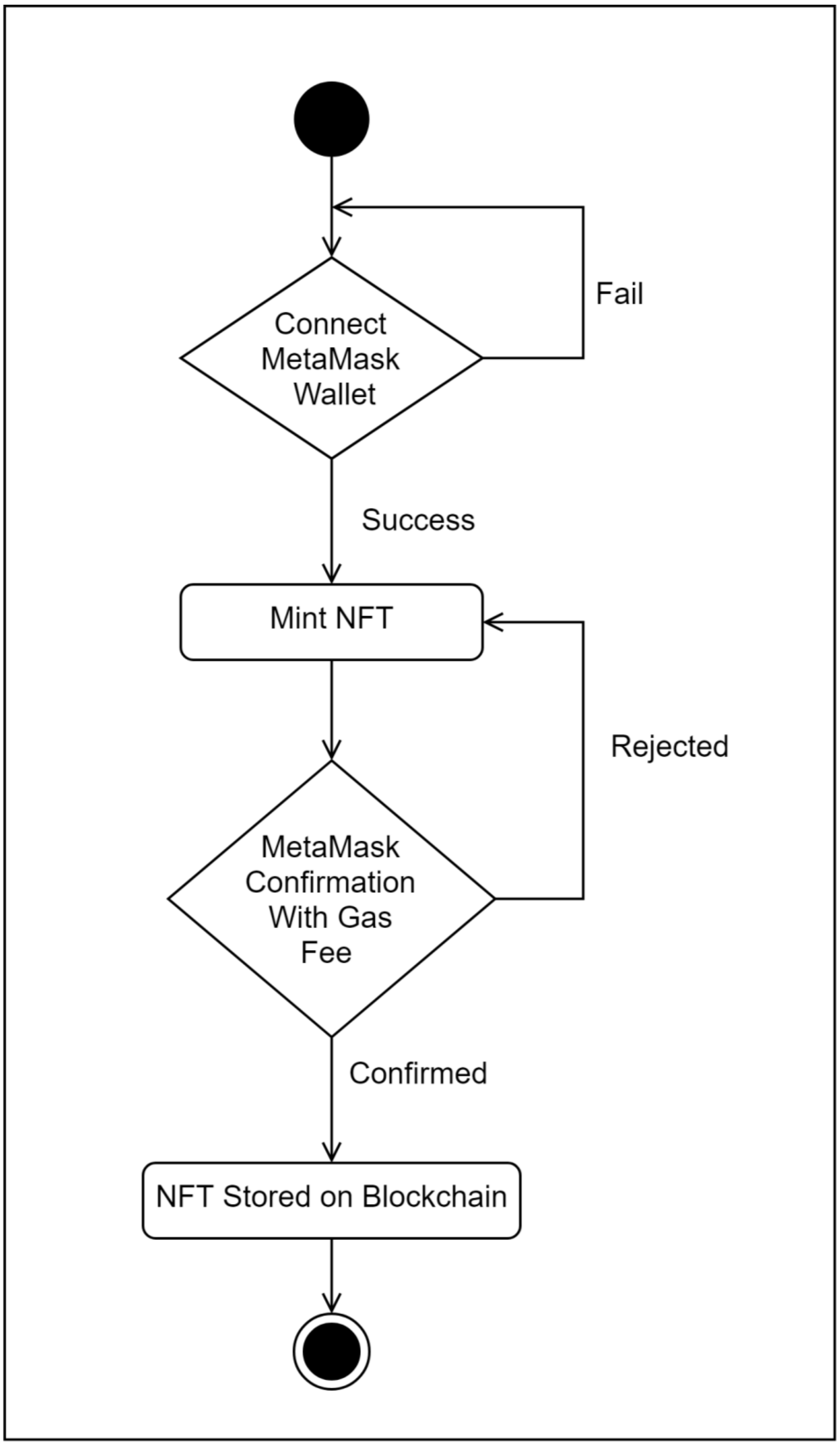


Figure 10.3

b. Sell/Burn NFT

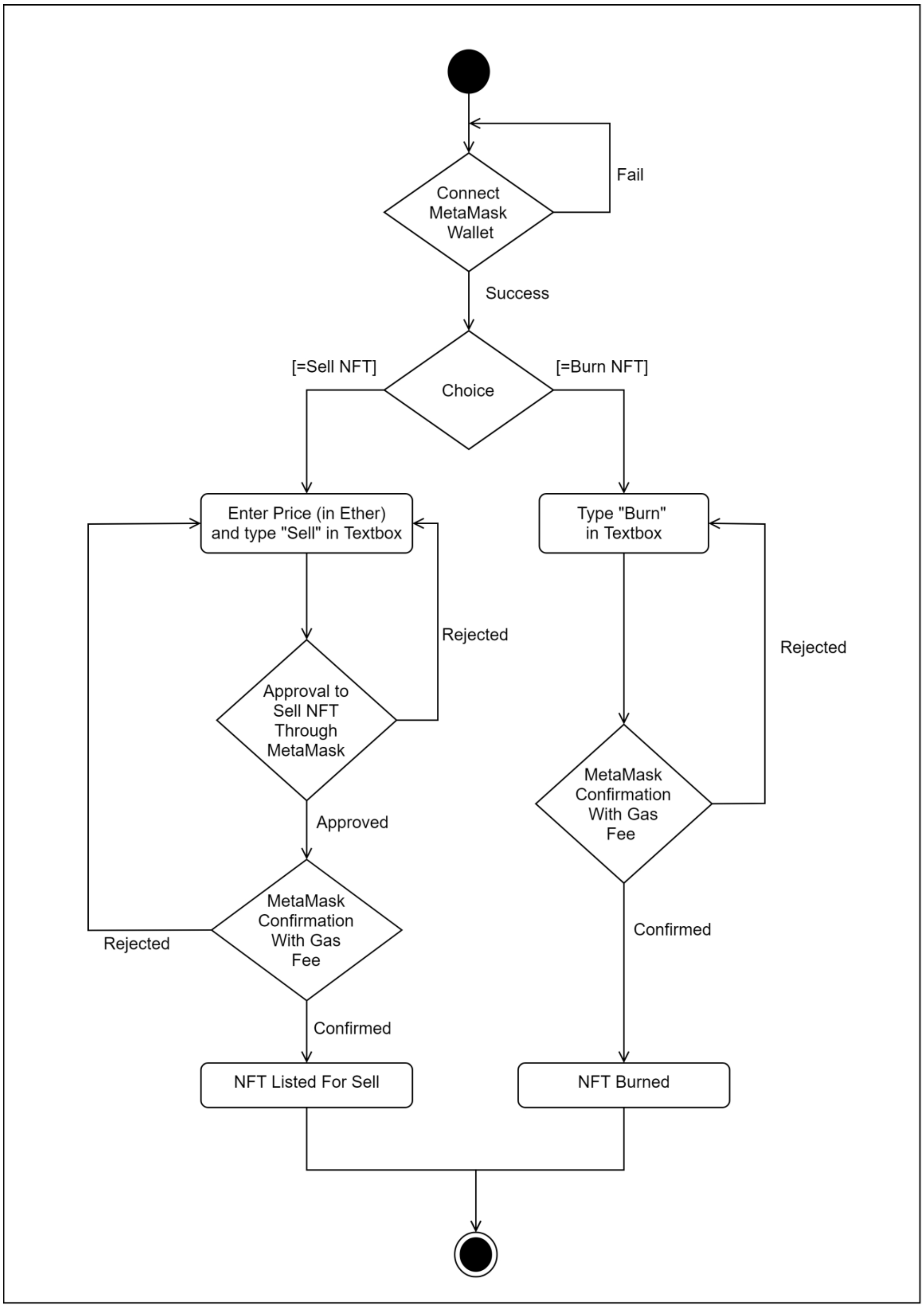
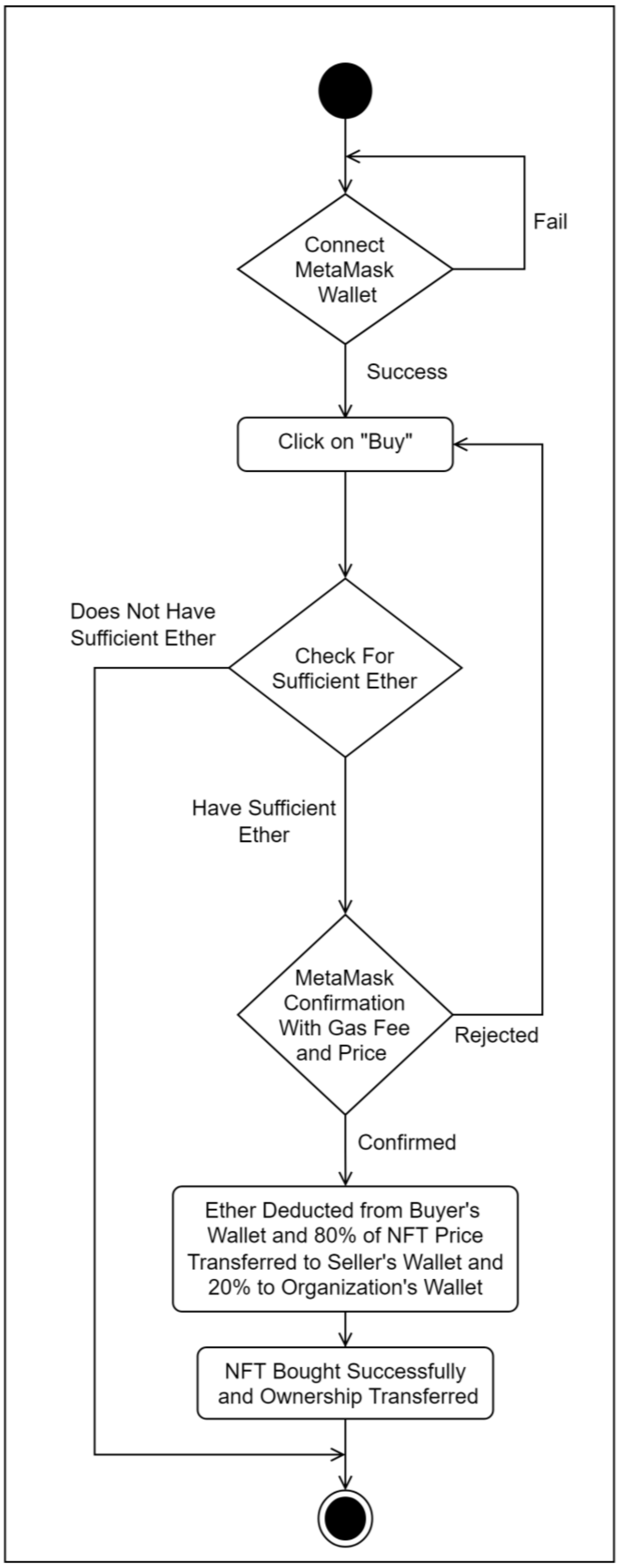


Figure 10.4

c. Buy NFT



Connect MetaMask Wallet
Fail
Success
Click on "Buy"
Does Not Have Sufficient Ether
Check For Sufficient Ether
Have Sufficient Ether
MetaMask Confirmation With Gas Fee and Price
Rejected
Confirmed
Ether Deducted from Buyer's Wallet and 80% of NFT Price Transferred to Seller's Wallet and 20% to Organization's Wallet
NFT Bought Successfully and Ownership Transferred

Figure 10.5

10.10 IPFS Workflow

IPFS is a great fit for storing and addressing data for NFTs. Since an NFT can't be easily changed after it's been created, it's a good idea to think about how the data for your NFTs is stored, addressed, and made persistent over time.

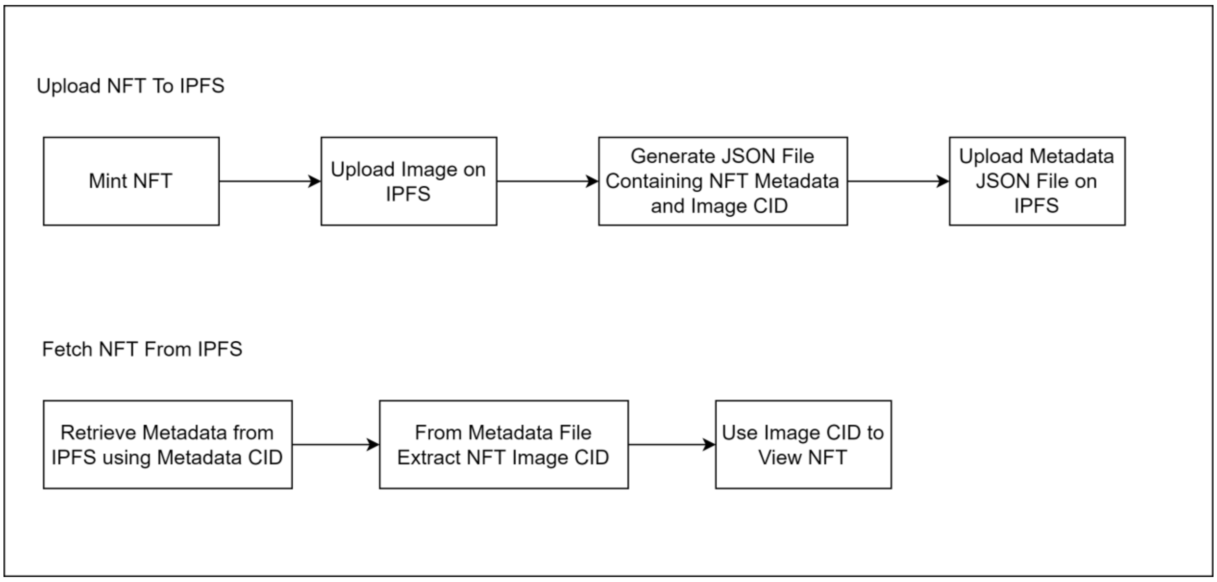

Upload NFT To IPFS
Mint NFT
Upload Image on IPFS
Generate JSON File Containing NFT Metadata and Image CID
Upload Metadata JSON File on IPFS
Fetch NFT From IPFS
Retrieve Metadata from IPFS using Metadata CID
From Metadata File Extract NFT Image CID
Use Image CID to View NFT

Figure 10.6

Uploading NFT to IPFS:

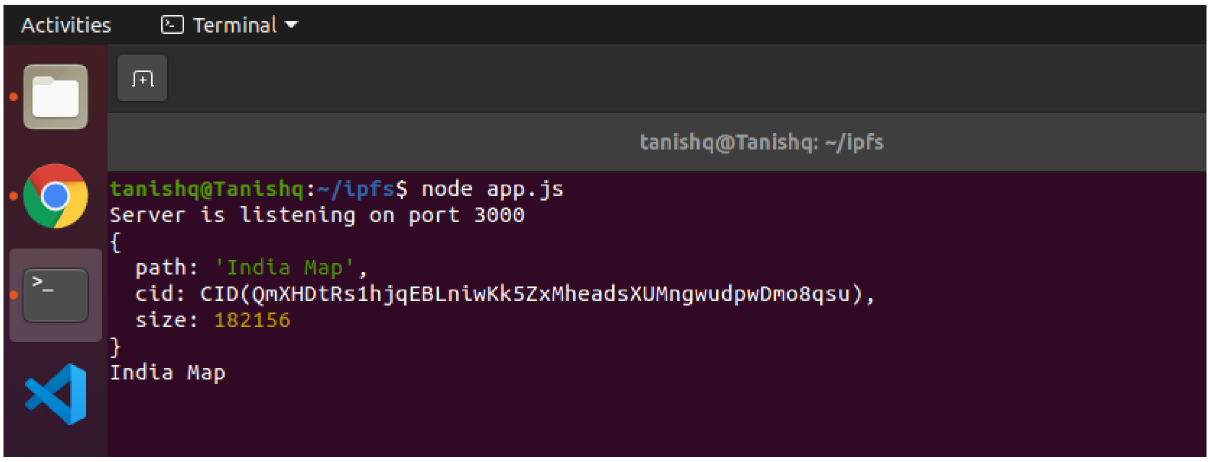
When an NFT is created, its image will be first uploaded on IPFS which will return a unique CID. Once the NFT image is uploaded, NFT Metadata File will be created which is basically a JSON file in which NFT name, description and NFT image CID will be present. Now, the JSON file is uploaded on IPFS which will also return a unique CID which will be stored in the blockchain.

Let’s take an example: Consider the image given below as NFT.

INDIA
States and Union Territories
Map not to Scale
Copyright © 2019 www.mapsofworld.com

We will upload this image on IPFS and a unique CID will be returned

QmXHDtRs1hjqEBLniwKk5ZxMheadsXUMngwudpwDmo8qsu



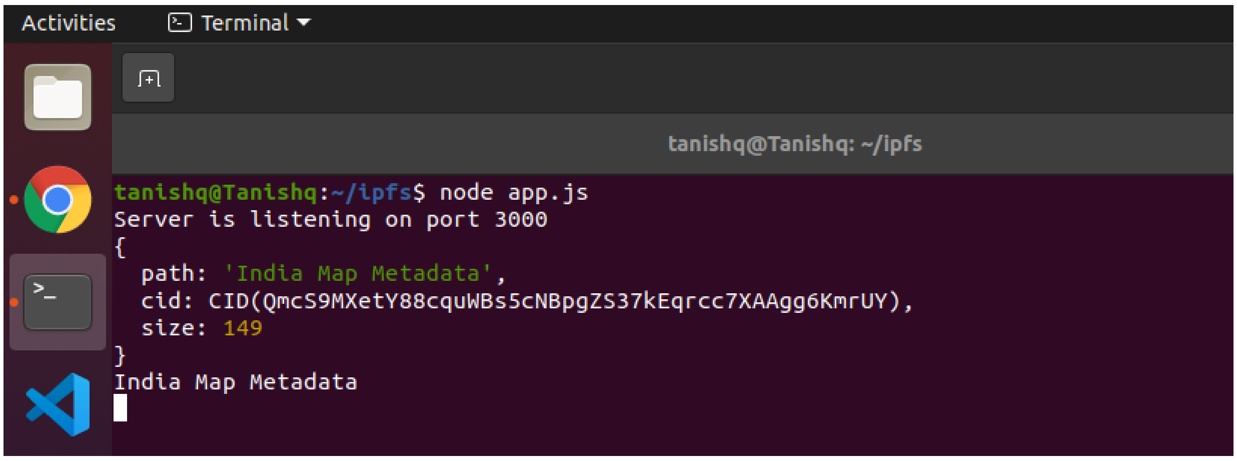
```
tanishq@Tanishq:~/ipfs$ node app.js
Server is listening on port 3000
{
  path: 'India Map',
  cid: CID(QmXHDtRs1hjqEBLniwKk5ZxMheadsXUMngwudpwDmo8qsu),
  size: 182156
}
India Map
```

Now, we will create a JSON File in which CID of an image will be stored along with the metadata.

```
{
   "name": "India Map",
   "description": "This is the Map of India",
   "image": "QmXHDtRs1hjqEBLniwKk5ZxMheadsXUMngwudpwDmo8qsu"
}
```

Upload this metadata file on IPFS and a unique CID will be returned

QmcS9MXetY88cquWBs5cNBpgZS37kEqrcc7XAAgg6KmrUY



```
tanishq@Tanishq:~/ipfs$ node app.js
Server is listening on port 3000
{
  path: 'India Map Metadata',
  cid: CID(QmcS9MXetY88cquWBs5cNBpgZS37kEqrcc7XAAgg6KmrUY),
  size: 149
}
India Map Metadata
```

This CID will be stored in blockchain.

Fetching NFT from IPFS:

1. Use the CID of NFT Metadata File and view the data **https://ipfs.io/ipfs/CID**.
2. Extract the Image CID from JSON File.
3. Use the extracted CID to view the NFT, **https://ipfs.io/ipfs/CID**.

Chapter-11

Project Testing

Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The testing is important since it discovers defects/bugs before the delivery to the client, which guarantees the quality of the software and ensures reliability and high performance. It makes the software more reliable and easy to use.

Each smart contract has been tested individually first in Remix IDE on Rinkeby Test Network and then it got tested with the web application.

11.1 Testing Methods Applied

The various types of testing that have been applied to the project are:

1. **Unit Testing:**
   After development, each component has been tested individually whether it is working with proper functionality or not. Are there any bugs or problems in an individual component?

2. **Integration Testing:**
   After developing some components of the project, those components are integrated together logically and tested as a group to check after combining whether the project functionalities are working properly or not.

3. **Black Box Testing:**
   After developing and integrating the components together. Black box testing has been applied to check whether all the functionalities are working correctly or not. The users who don't have technical/programming knowledge can also perform this testing.

4. **White Box Testing:**

   White Box Testing is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software. It is mostly done by software developers. Knowledge of the implementation of the project is very important.

5. **Regression Testing:**

   Regression testing has been applied at places where some changes has been made in the smart contracts/components or where new functionality has been added. This testing has been applied to check that whether by making these changes, the project old functionalities are working properly or not. **“Are there any issues in the project old functionalities after making the changes in the project?”** If there is any issue present or by making changes in one component other components are also getting affected, then that issue has been tried to resolve.

11.2 Test Cases

1. Mint NFT

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Mint an NFT	Name, Description, Image	MetaMask pop up for confirmation to pay gas fee, after confirming NFT should be minted successfully	NFT Minted Successfully	Pass
2.	To Mint an NFT after	Name, Description,	NFT not Minted	NFT not Minted	Pass

	rejecting MetaMask confirmation	Image			

2. Sell NFT

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	Sell NFT	Price (in Ether), Type “Sell” in textbox and click on Enter Button	MetaMask pop up for confirmation to pay gas fee and for approval, after confirming NFT should be Listed for Sale	NFT Listed for Sell Successfully. Now, Buyer can buy this NFT if he/she likes it.	Pass
2.	Sell NFT after rejecting MetaMask confirmation	Price (in Ether), Type “Sell” in textbox and click on Enter Button	NFT should not be Listed for Sale	NFT not Listed for Sale	Pass

3. Remove NFT from Sell

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	Remove NFT from Sell so, that Buyer cannot Buy	Click on Cancel Button	MetaMask pop up for confirmation to pay gas fee, after confirming NFT should be removed from Sell	NFT Removed from Sell	Pass
2.	Remove NFT from Sell after rejecting MetaMask confirmation	Click on Cancel Button	NFT should not be Removed from Sell	NFT not Removed from Sell	Pass

4. Buy NFT

Test Case ID	Test	Expected Input	Expected Output	Actual Result	Status
1.	To Buy NFT	Click on Buy Button	MetaMask pop up for confirmation to pay gas fee and price of NFT, after confirming	NFT Ownership Transferred Successfully and 20% of NFT Price Transferred to	Pass

			NFT Ownership should be transferred to Buyer and 20% of NFT Price should be transferred to Organization’s Wallet	Organization’s Wallet	
2.	Insufficient Funds to Buy NFT	Click on Buy Button	Not able to buy NFT due to Insufficient Funds	Not able to buy NFT due to Insufficient Funds	Pass

5. Burn NFT

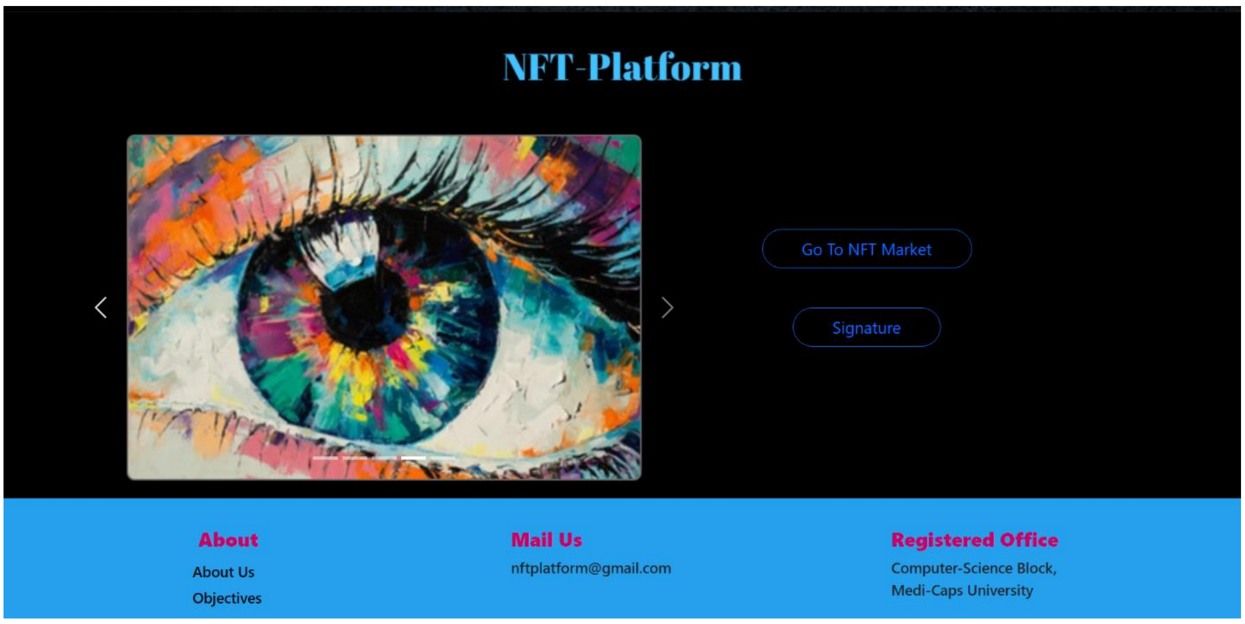
Test Case ID	**Test**	**Expected Input**	**Expected Output**	**Actual Result**	**Status**
1.	To Burn NFT	Type “Burn” in textbox and click on Enter Button	MetaMask pop up for confirmation to pay gas fee, after confirming NFT Ownership should be transferred to Organization’s Wallet	NFT Burned Successfully	Pass

2.	Not able to Burn NFT, after rejecting MetaMask confirmation	Type “Burn” in textbox and click on Enter Button	Not able to Burn NFT	Not able to Burn NFT	Pass

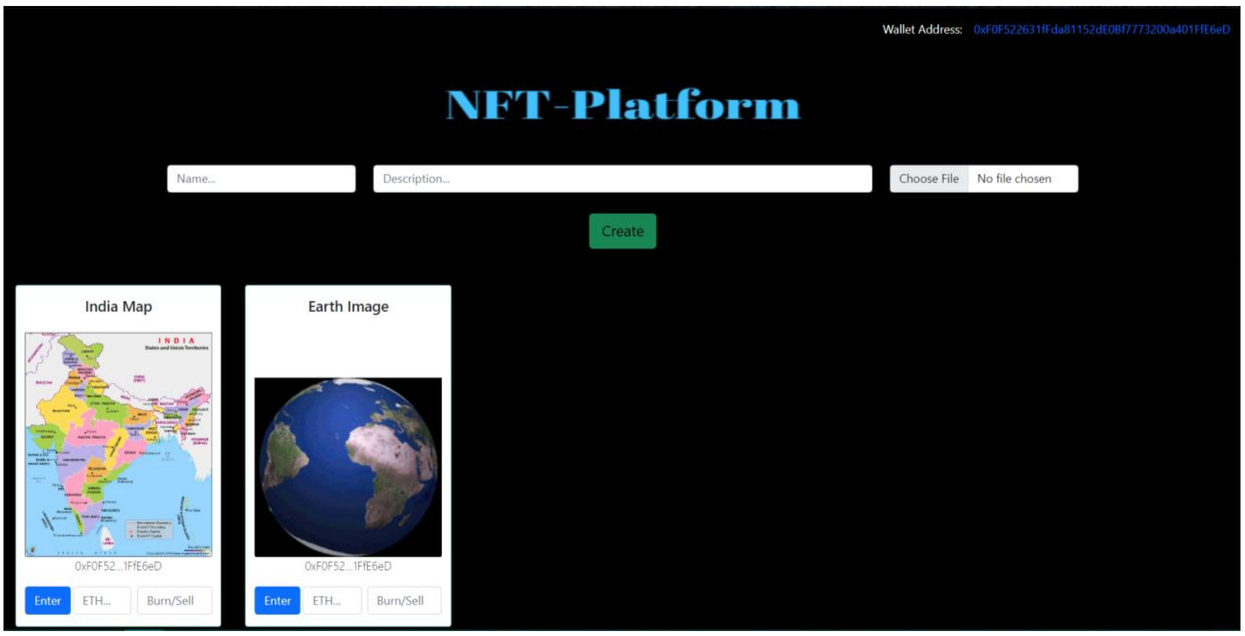
Chapter-12

Snapshots of Project

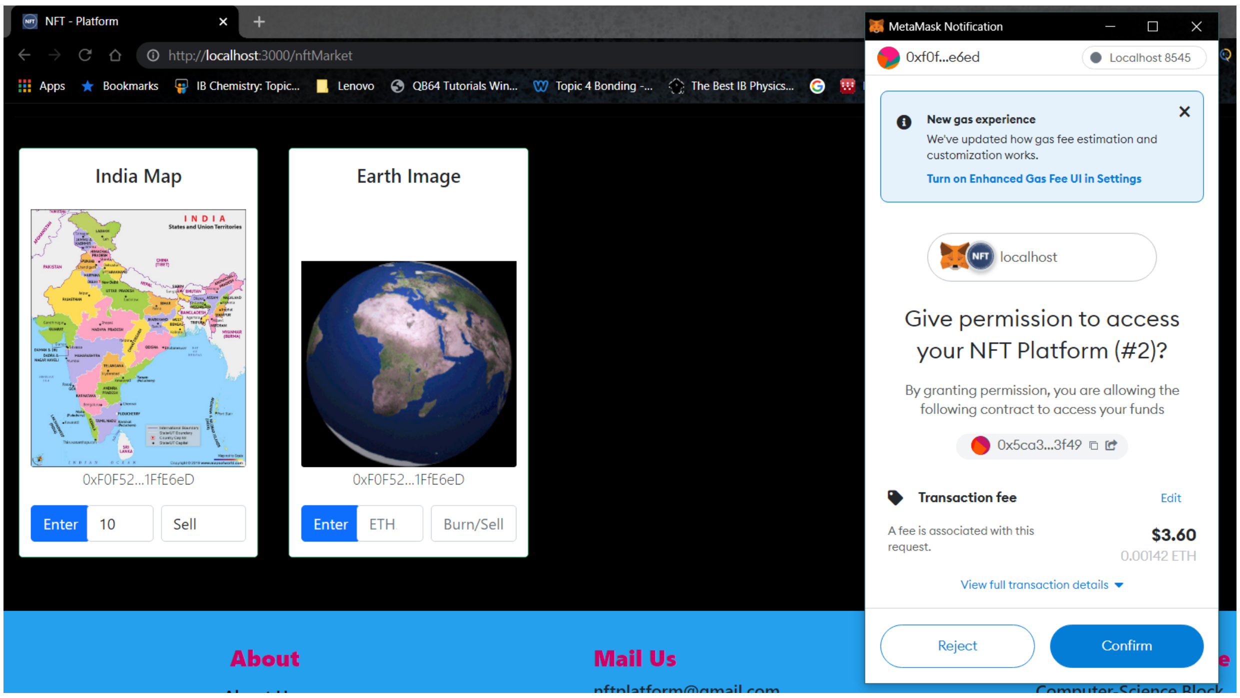
Home Page



NFT Market



Approving NFT to Sell on the Marketplace



New gas experience
We've updated how gas fee estimation and customization works.
Turn on Enhanced Gas Fee UI in Settings

Give permission to access your NFT Platform (#2)?

By granting permission, you are allowing the following contract to access your funds

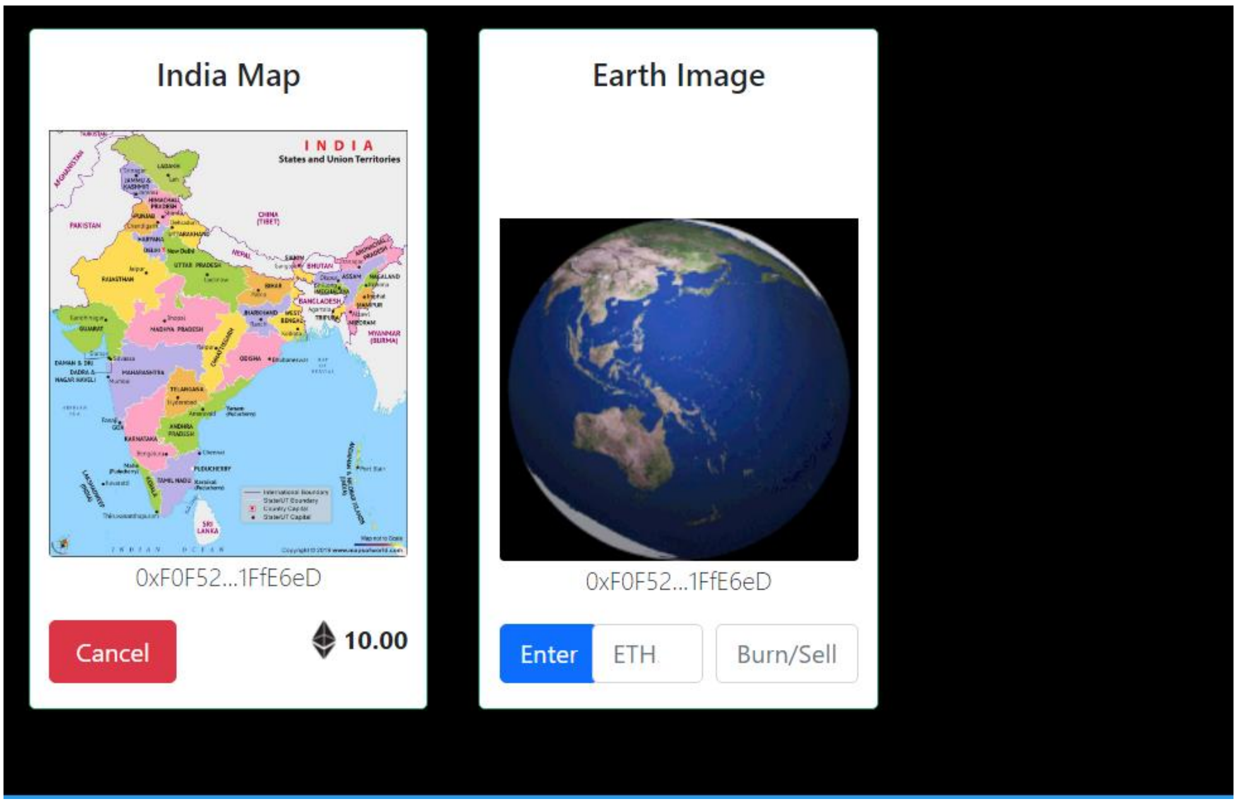
0x5ca3...3f49

Transaction fee — Edit

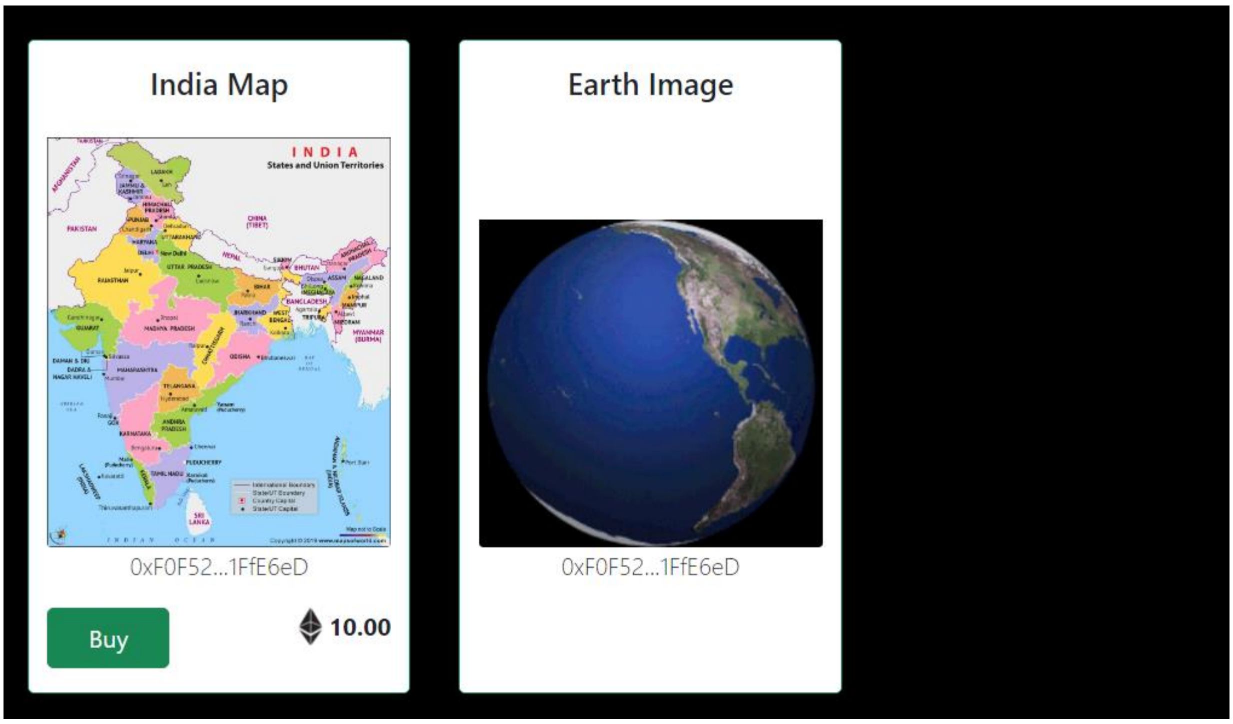
A fee is associated with this request. $3.60 0.00142 ETH

View full transaction details

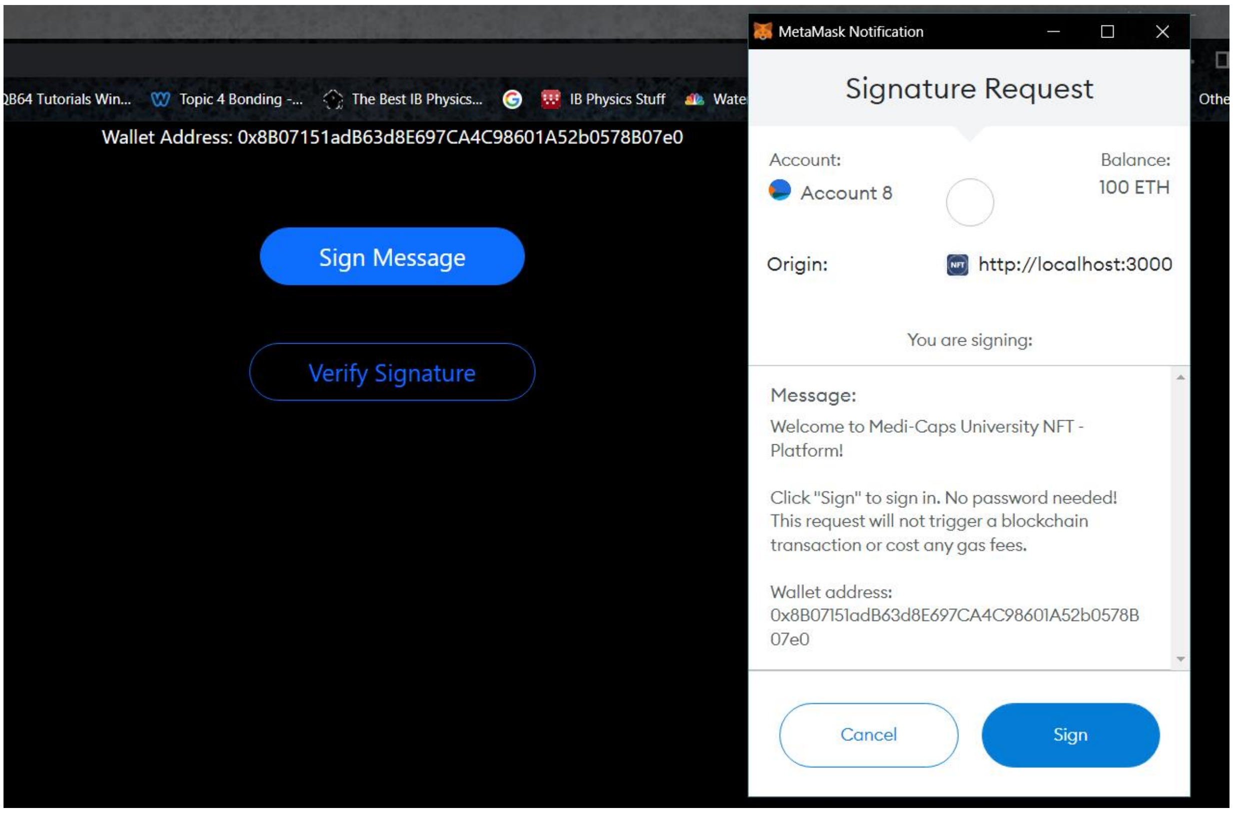
NFT Listed for Sell



Buy NFT



Signature



Wallet Address: 0x8B07151adB63d8E697CA4C98601A52b0578B07e0

Signature Request

Account: Account 8

Balance: 100 ETH

Origin: http://localhost:3000

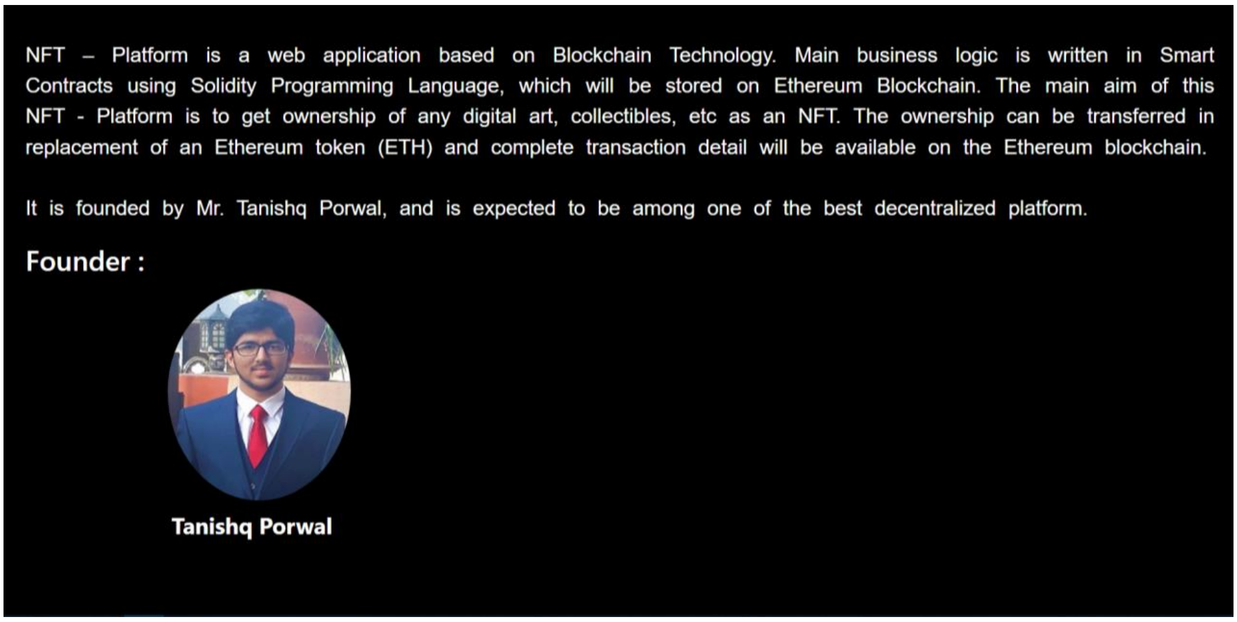
You are signing:

Message:
Welcome to Medi-Caps University NFT - Platform!

Click "Sign" to sign in. No password needed! This request will not trigger a blockchain transaction or cost any gas fees.

Wallet address:
0x8B07151adB63d8E697CA4C98601A52b0578B07e0

About Us

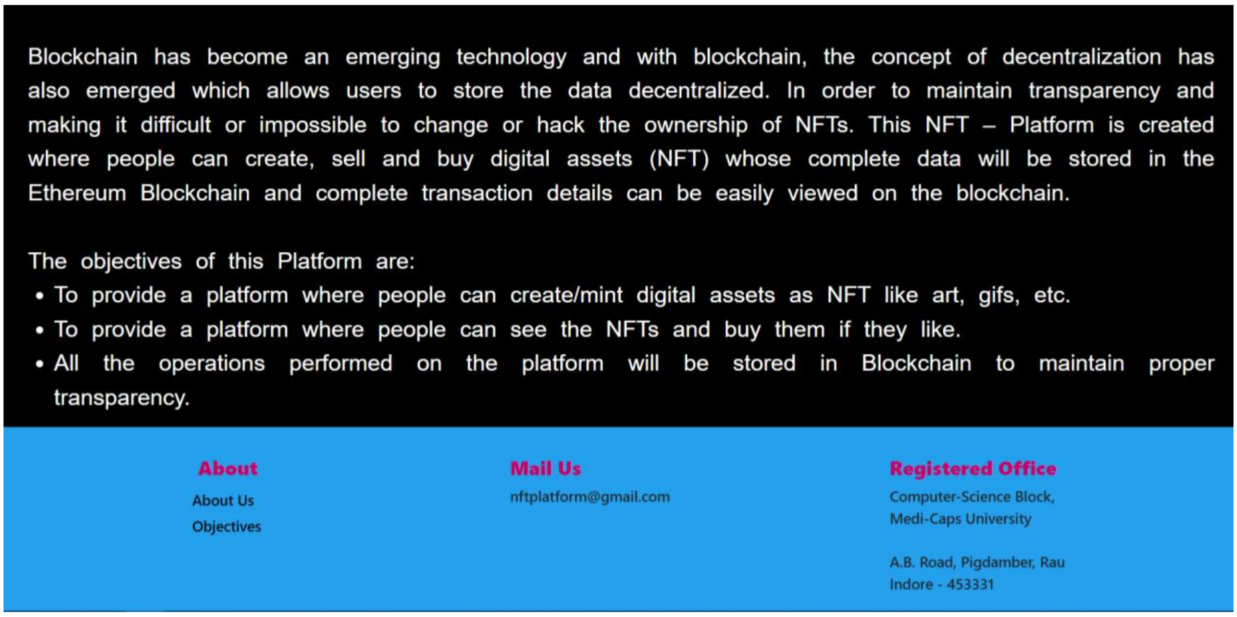

NFT – Platform is a web application based on Blockchain Technology. Main business logic is written in Smart Contracts using Solidity Programming Language, which will be stored on Ethereum Blockchain. The main aim of this NFT - Platform is to get ownership of any digital art, collectibles, etc as an NFT. The ownership can be transferred in replacement of an Ethereum token (ETH) and complete transaction detail will be available on the Ethereum blockchain.

It is founded by Mr. Tanishq Porwal, and is expected to be among one of the best decentralized platform.

Founder :

Tanishq Porwal

Objectives


Blockchain has become an emerging technology and with blockchain, the concept of decentralization has also emerged which allows users to store the data decentralized. In order to maintain transparency and making it difficult or impossible to change or hack the ownership of NFTs. This NFT – Platform is created where people can create, sell and buy digital assets (NFT) whose complete data will be stored in the Ethereum Blockchain and complete transaction details can be easily viewed on the blockchain.

The objectives of this Platform are:

- To provide a platform where people can create/mint digital assets as NFT like art, gifs, etc.
- To provide a platform where people can see the NFTs and buy them if they like.
- All the operations performed on the platform will be stored in Blockchain to maintain proper transparency.

About
About Us
Objectives

Mail Us
nftplatform@gmail.com

Registered Office
Computer-Science Block,
Medi-Caps University

A.B. Road, Pigdamber, Rau
Indore - 453331

Chapter-13

Summary and Conclusion

Internship at Fourty Seven Billion Information Technologies Pvt. Ltd. as a Software Engineer Intern was a wonderful experience. Initially training was given, which was very helpful. Training and development are considered as a strategy for growth in the field of Computer Science. These training programs enhance skills, improve efficiency, productivity and growth opportunities.

This internship has been a very useful experience for me. I have gained new knowledge, skills and met many new people. During my internship, I realize that observation is the main element to find out the root cause of a problem.

The project I worked on is NFT – Platform: Marketplace for Digital Assets. During the project, I cooperated with my colleagues to determine and solve the problems. The project has helped me to discipline myself, be patient, self-trust, and take initiatives. Besides, my communication, time management and critical and analytical thinking skills got strengthened. Furthermore, I got to know how companies manage such big projects. This Internship has helped me to enhance my knowledge and practical skills.

As a result of this initial training and internship, I am now aware of all these technologies mentioned, and also ready to use that knowledge in the future for building various projects.

Bibliography

- HAYES, "Blockchain Explained", *Investopedia*, 05-Mar-2022. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed: 10- Jan- 2022].
- "Solidity — Solidity 0.8.13 documentation", *Docs.soliditylang.org*, 2016. [Online]. Available: https://docs.soliditylang.org/en/v0.8.13/. [Accessed: 21- Jan- 2022].
- "Welcome to Remix's documentation! — Remix - Ethereum IDE 1 documentation", *Remix-ide.readthedocs.io*, 2019. [Online]. Available: https://remix-ide.readthedocs.io/en/latest/. [Accessed: 01- Feb- 2022].
- S. Richards, "Introduction to smart contracts", *ethereum.org*, 12-Apr-2022. [Online]. Available: https://ethereum.org/en/developers/docs/smart-contracts/. [Accessed: 07- Feb- 2022].
- "What are smart contracts on blockchain?", *IBM*. [Online]. Available: https://www.ibm.com/topics/smart-contracts#:~:text=Smart%20contracts%20are%20simply%20programs,intermediary's%20involvement%20or%20time%20loss.f. [Accessed: 07- Feb- 2022].
- "web3.js - Ethereum JavaScript API — web3.js 1.0.0 documentation", *Web3js.readthedocs.io*, 2016. [Online]. Available: https://web3js.readthedocs.io/en/v1.7.3/. [Accessed: 18- Feb- 2022].
- "IPFS Powers the Distributed Web", *IPFS powers the distributed web*. [Online]. Available: https://ipfs.io/. [Accessed: 24- Feb- 2022].
- "The crypto wallet & gateway to Web3 blockchain apps", *metamask.io*. [Online]. Available: https://metamask.io/. [Accessed: 02- Mar- 2022].
- "Ganache - Truffle Suite", *trufflesuite.com*. [Online]. Available: https://trufflesuite.com/ganache/. [Accessed: 03- Mar- 2022].
- "Sweet Tools for Smart Contracts", *Home - Truffle Suite*. [Online]. Available: https://trufflesuite.com/. [Accessed: 03- Mar- 2022].

- "Ethereum development environment for professionals by Nomic Foundation", *hardhat.org*. [Online]. Available: https://hardhat.org/. [Accessed: 08- Mar- 2022].
- "W3Schools Free Online Web Tutorials", *w3schools.com*, 2022. [Online]. Available: https://www.w3schools.com/. [Accessed: 15- Mar- 2022].
- "What is version control | Atlassian Git Tutorial", *Atlassian*. [Online]. Available: https://www.atlassian.com/git/tutorials/what-is-version-control. [Accessed: 01- Apr- 2022].
- "Git", *Git-scm.com*. [Online]. Available: https://git-scm.com/. [Accessed: 01- Apr- 2022].
- "The One DevOps platform", *GitLab*. [Online]. Available: https://about.gitlab.com/. [Accessed: 01- Apr- 2022].
- "Tutorials - Javatpoint", *www.javatpoint.com*, 2022. [Online]. Available: https://www.javatpoint.com/. [Accessed: 02- Apr- 2022].
- "Free Online Tutorials and Courses", *tutorialspoint.com*, 2022. [Online]. Available: https://www.tutorialspoint.com/tutorialslibrary.htm. [Accessed: 03- Apr- 2022].
- "Flowchart Maker & Online Diagram Software", *app.diagrams.net*, 2022. [Online]. Available: https://app.diagrams.net/. [Accessed: 10- Apr- 2022].