K. J. Somaiya College of Engineering, Mumbai-77

Department of Computer Engineering

Batch: A2(BCT-1)	**Roll No.: 1911031**
Experiment No. 08	

Title: Blockchain Mini Project (Students Choice)

Objective: To understand how to create a Decentralized application using Node, React and truffle suit and how a deploy a smart contract on a local blockchain.

Expected Outcome of Experiment:

CO	Outcome
CO1	**Build your own Blockchain businesses with acquired knowledge.**
CO2	**Learn Solidity language & Multiple Technology-based developments.**
CO3	**Apply the algorithm and techniques used in Blockchain.**
CO4	**Grasp the in-depth understanding of Blockchain, Smart Contracts & how it works.**
CO5	**Describe the methods of mining.**

Books/ Journals/ Websites referred:

1. **https://trufflesuite.com/docs/truffle/how-to/create-a-project/**
2. **https://nodejs.org/en/**
3. **https://www.investopedia.com/terms/e/ethereum.asp**
4. **https://www.ibm.com/in-en/topics/smart-contracts**

Abstract:-

What Is Ethereum and How Does It Work?

What Is Ethereum?

At its core, Ethereum is a decentralized global software platform powered by blockchain technology. It is most commonly known for its native cryptocurrency, ether (ETH).

Ethereum can be used by anyone to create any secured digital technology. It has a token designed to pay for work done supporting the blockchain, but participants can also use it to pay for tangible goods and services if accepted.

Ethereum is designed to be scalable, programmable, secure, and decentralized. It is the blockchain of choice for developers and enterprises creating technology based upon it to change how many industries operate and how we go about our daily lives.

It natively supports smart contracts, an essential tool behind decentralized applications.1 Many decentralized finance (DeFi) and other applications use smart contracts in conjunction with blockchain technology.

Learn more about Ethereum, its token ETH, and how they are an integral part of non-fungible tokens, decentralized finance, decentralized autonomous organizations, and the metaverse.

Related Theory: -

KEY TAKEAWAYS

- Ethereum is a blockchain-based platform best known for its cryptocurrency, ether (ETH).
- The blockchain technology that powers Ethereum enables secure digital ledgers to be publicly created and maintained.
- Bitcoin and Ethereum have many similarities but different long-term visions and limitations.
- Ethereum changed from proof of work to proof of stake in Septemeber 2022.2
- Ethereum is the foundation for many emerging technological advances based on blockchain.

Ethereum Founder Joe Lubin Explains What It Is & Why It Matters

How Does Ethereum Work?

Vitalik Buterin, credited with conceiving Ethereum, published a white paper to introduce it in 2014.3 The Ethereum platform was launched in 2015 by Buterin and Joe Lubin, founder of the blockchain software company ConsenSys.45

The founders of Ethereum were among the first to consider the full potential of blockchain technology beyond just enabling the secure virtual payment method.

Since the launch of Ethereum, ether as a cryptocurrency has risen to become the second-largest cryptocurrency by market value. It is outranked only by Bitcoin.6

Blockchain Technology

Ethereum, like other cryptocurrencies, involves blockchain technology. Imagine a very long chain of blocks. All of the information contained in each block is added to every newly-created block with new data. Throughout the network, an identical copy of the blockchain is distributed.

This blockchain is validated by a network of automated programs that reach a consensus on the validity of transaction information. No changes can be made to the blockchain unless the network reaches a consensus. This makes it very secure.

Consensus is reached using an algorithm commonly called a consensus mechanism. Ethereum uses the proof-of-stake algorithm, where a network of participants called validators create new blocks and work together to verify the information they contain. The blocks contain information about the state of the blockchain, a list of attestations (a validator's signature and vote on the validity of the block), transactions, and much more.

In mid-September 2022, Ethereum officially switched over to a proof-of-stake algorithm, which is cheaper and more environmentally friendly than a proof-of-work model.2

Proof-of-Stake Mechanism

Proof-of-stake differs from proof-of-work in that it doesn't require the energy-intensive computing referred to as mining to validate blocks. It uses a finalization protocol called Casper-FFG and the algorithm LMD Ghost, combined into a consensus mechanism called Gasper, which monitors consensus and defines how validators receive rewards for work or are punished for dishonesty.7

Solo validators must stake 32 ETH to activate their validation ability. Individuals can stake smaller amounts of ETH, but they are required to join a validation pool and share any rewards. A validator creates a new block and attests that the information is valid in

a process called attestation, where the block is broadcast to other validators called a committee who verify it and vote for its validity.

Validators who act dishonestly are punished under proof-of-stake. Validators who attempt to attack the network are identified by Gasper, which identifies the blocks to accept and reject based on the votes of the validators.7

Dishonest validators are punished by having their staked ETH burned and being removed from the network. Burning refers to sending crypto to a wallet that has no keys, which takes them out of circulation.

Wallets

Ethereum owners use wallets to store their ether. A wallet is a digital interface that lets you access your ether stored on the blockchain. Your wallet has an address, which is similar to an email address in that it is where users send ether, much like they would an email.8

Ether is not actually stored in your wallet. Your wallet holds private keys you use as you would a password when you initiate a transaction. You receive a private key for each ether you own. This key is essential for accessing your ether. That's why you hear so much about securing keys using different storage methods.

Historic Split

One notable event in Ethereum's history is the hard fork, or split, of Ethereum and Ethereum Classic.9 In 2016, a group of network participants gained majority control of the Ethereum blockchain to steal more than $50 million worth of ether, which had been raised for a project called The DAO.1011

The raid's success was attributed to the involvement of a third-party developer for the new project. Most of the Ethereum community opted to reverse the theft by invalidating the existing Ethereum blockchain and approving a blockchain with a revised history.

However, a fraction of the community chose to maintain the original version of the Ethereum blockchain. That unaltered version of Ethereum permanently split to become the cryptocurrency Ethereum Classic (ETC).12

Ethereum vs. Bitcoin

Ethereum is often compared to Bitcoin. While the two cryptocurrencies have many similarities, there are some some important distinctions.

Ethereum is described by founders and developers as "the world's programmable blockchain," positioning itself as an electronic, programmable network with many applications.13 The Bitcoin blockchain, by contrast, was created only to support the bitcoin cryptocurrency.

The Ethereum platform was founded with broad ambitions to leverage blockchain technology for many diverse applications. Bitcoin was designed strictly as a payment method.
The maximum number of bitcoins that can enter circulation is 21 million.14 The amount of ETH that can be created is unlimited, although the time it takes to process a block of ETH limits how much ether can be minted each year.15 The number of Ethereum coins in circulation is more than 122 million.16

Another significant difference between Ethereum and Bitcoin is how the respective networks treat transaction processing fees. These fees, known as gas on the Ethereum network, are paid by the participants in Ethereum transactions. The fees associated with Bitcoin transactions are absorbed by the broader Bitcoin network.

Ethereum, as of September 2022, uses a proof-of-stake consensus mechanism. Bitcoin uses the energy-intensive proof-of-work consensus, which requires miners to compete for rewards.

The Future of Ethereum

Ethereum's transition to the proof-of-stake protocol, which enables users to validate transactions and mint new ETH based on their ether holdings, is part of a significant upgrade to the Ethereum platform. Previously called Eth2, this upgrade is now referred to only as Ethereum. However, Ethereum now has two layers. The first layer is the execution layer, where transactions and validations occur. The second layer is the consensus layer, where attestations and the consensus chain is maintained.17

The upgrade added capacity to the Ethereum network to support its growth, which will eventually help to address chronic network congestion problems that have driven up gas fees.218

To address scalability, Ethereum is continuing development of "sharding." Sharding will divide the Ethereum database amongst its network. This idea is similar to cloud computing, where many computers handle the workload to reduce computational time. These smaller database sections will be called shards, and shards will be worked on by those who have staked ETH. Shards will allow more validators to work at the same time, reducing the amount of time needed to reach consensus through a process called sharding consensus.19

Sharding is expected to be implemented sometime in 2023.

Web3

Web3 is still a concept, but it is generally theorized that it will be powered by Ethereum because many of the applications being developed use it.20

Use in Gaming

Ethereum is also being implemented into gaming and virtual reality. Decentraland is a virtual world that uses the Ethereum blockchain to secure items contained within that world. Land, avatars, wearables, buildings, and environments are all tokenized through the blockchain to create ownership.21

Axie Infinity is another game that uses blockchain technology and has its own cryptocurrency called Smooth Love Potion (SLP), used for rewards and transactions within the game.2223

Non-Fungible Tokens

Non-fungible tokens (NFTs) gained popularity in 2021. NFTs are tokenized digital items created using Ethereum.24 Generally speaking, tokenization gives one digital asset a specific digital token that identifies it and stores it on the blockchain.

This establishes ownership because the encrypted data stores the owner's wallet address. The NFT can be traded or sold and is viewed as a transaction on the blockchain. The transaction is verified by the network and ownership is transferred.

NFTs are being developed for all sorts of assets. For example, sports fans can buy a sports token—also called fan tokens—of their favorite athletes, which can be treated like trading cards. Some of these NFTs are pictures that resemble a trading card, and some of them are videos of a memorable or historic moment in the athlete's career.

The applications you may use in the metaverse, such as your wallet, a dApp, or the virtual world and buildings you visit, are likely to have been built on Ethereum.

The Development of DAOs

Decentralized Autonomous Organizations (DAOs), which are a collaborative method for making decisions across a distributed network, are being developed.25

For example, imagine that you created a venture capital fund and raised money through fund-raising, but you want decision-making to be decentralized and distributions to be automatic and transparent.

A DAO could use smart contracts and applications to gather the votes from the fund members and buy into ventures based on the majority of the group's votes, then automatically distribute any returns. The transactions could be viewed by all parties, and there would be no third-party involvement in handling any funds.

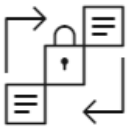
The part that cryptocurrency will play in the future is still vague. However, Ethereum appears to have a significant, upcoming role in personal and corporate finance and many aspects of our modern lives.

Benefits of Smart contracts

Speed, efficiency and accuracy
Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there's no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.

Trust and transparency
Because there's no third party involved, and because encrypted records of transactions are shared across participants, there's no need to question whether information has been altered for personal benefit.

Security
Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.

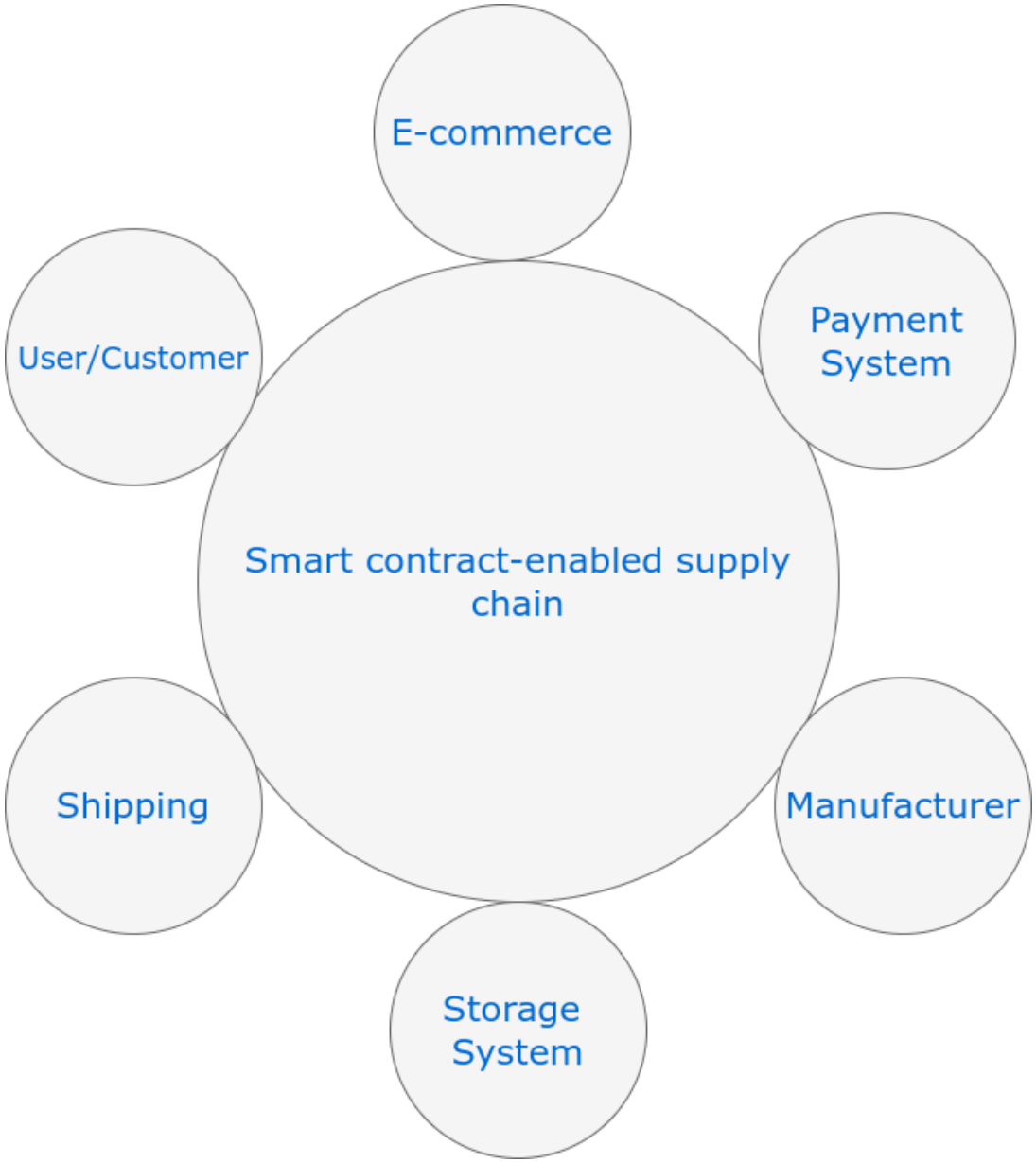


Savings
Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

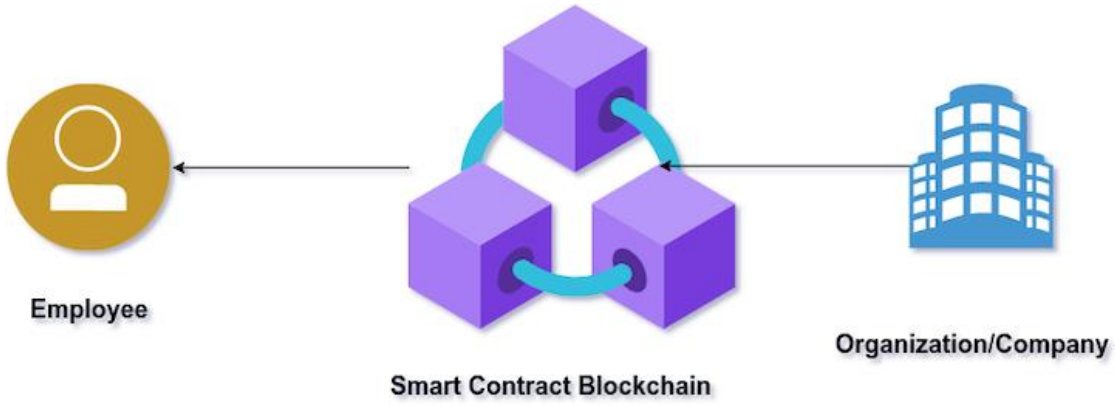
Related Theory (contd...): -

Applications of smart contract

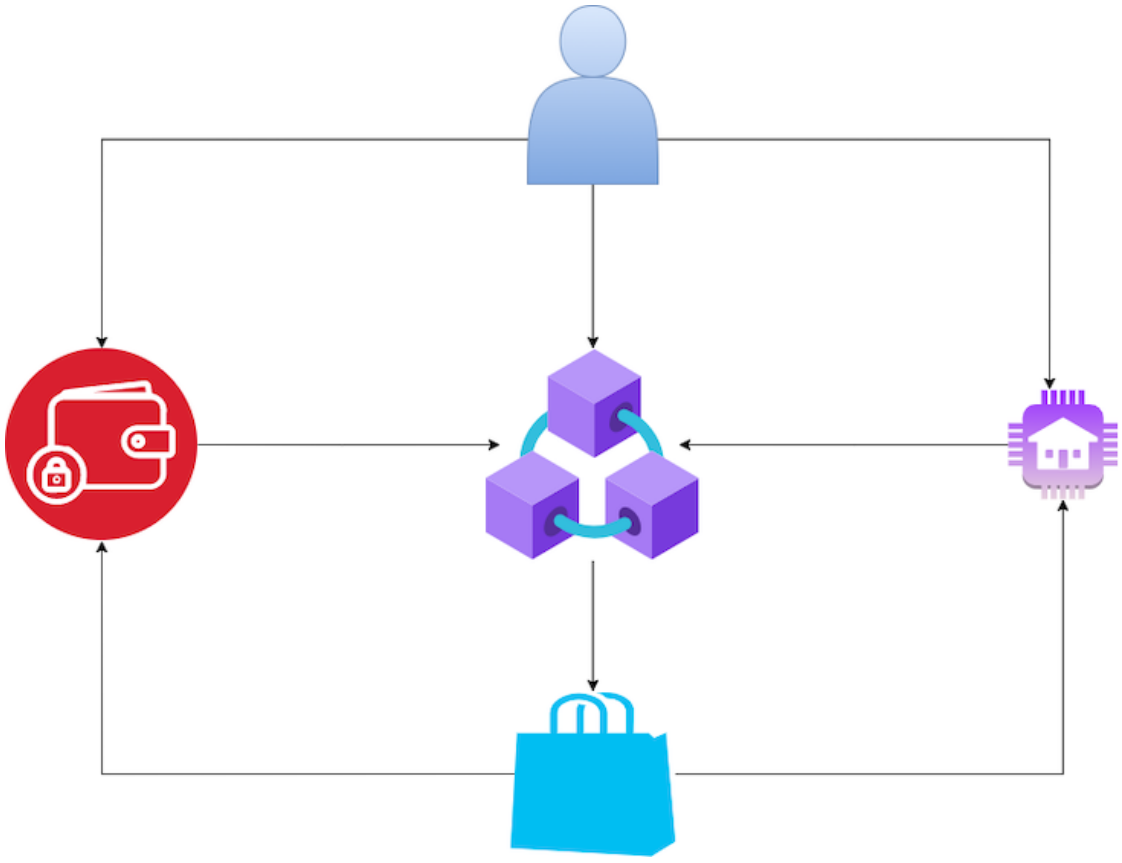
Health and agricultural supply chains



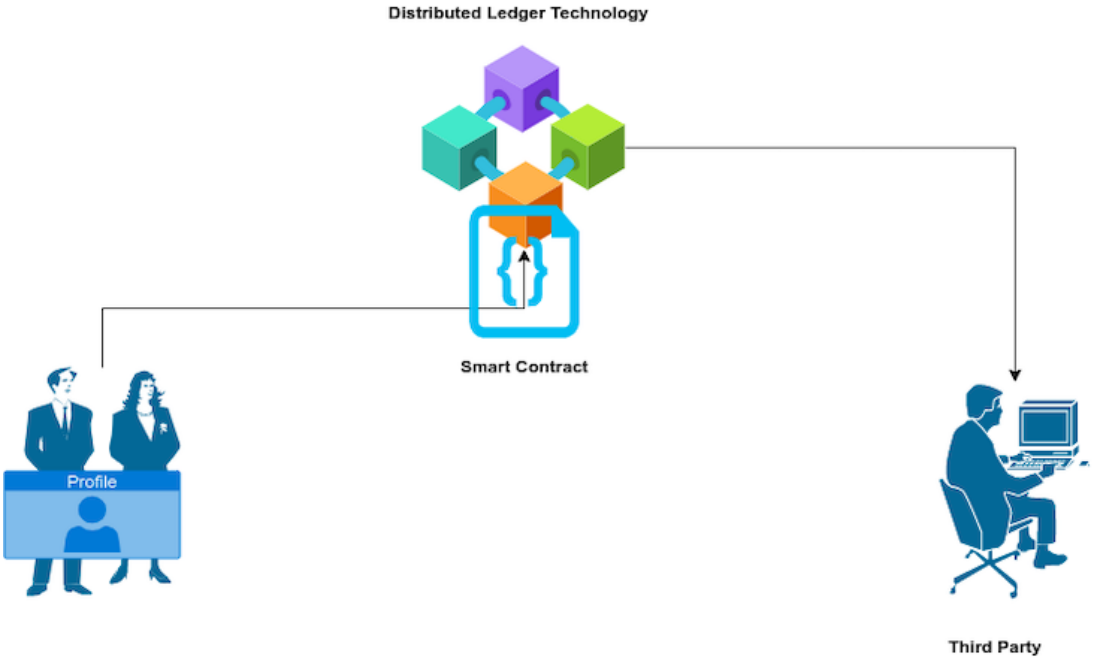
Administrative payments and billing



Real estate and crowdfunding



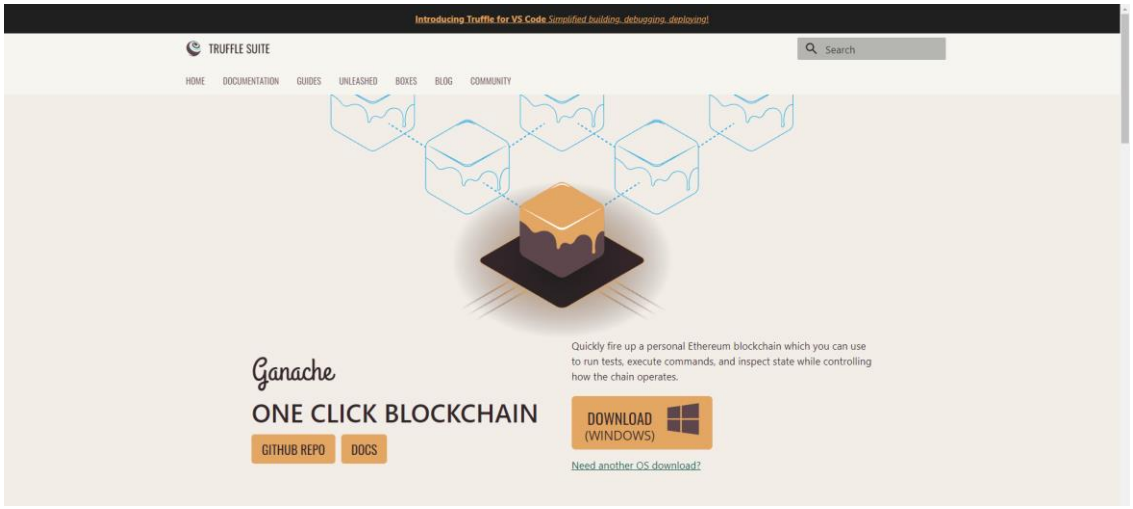
Identity management



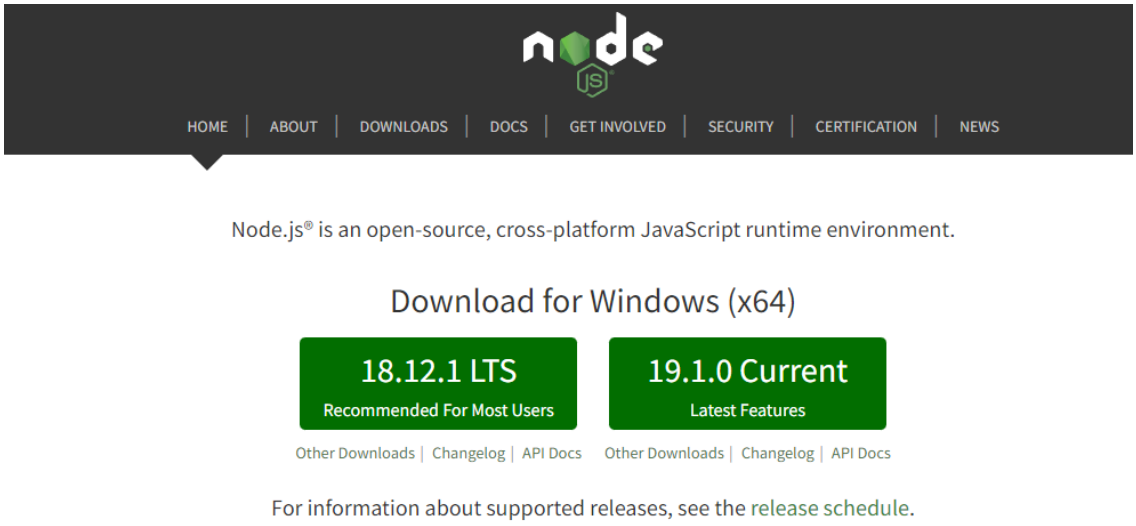
Implementation Details:

1. Enlist all the Steps followed and various options explored

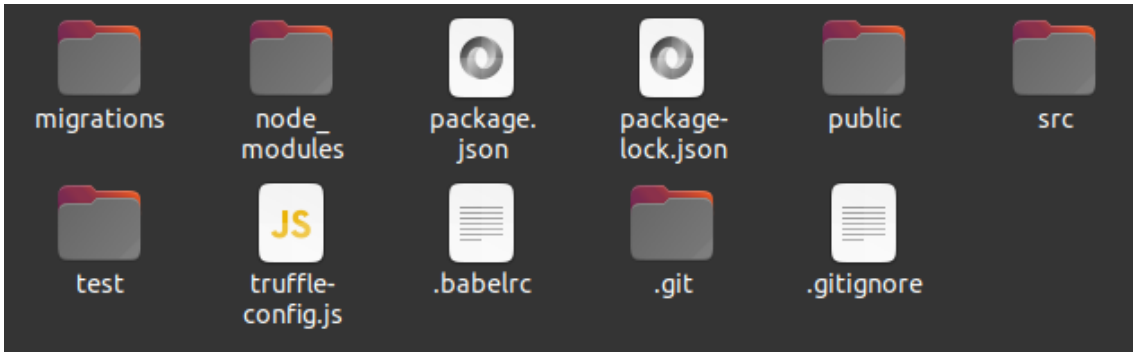
- Install ganache and truffle



- Install NodeJS

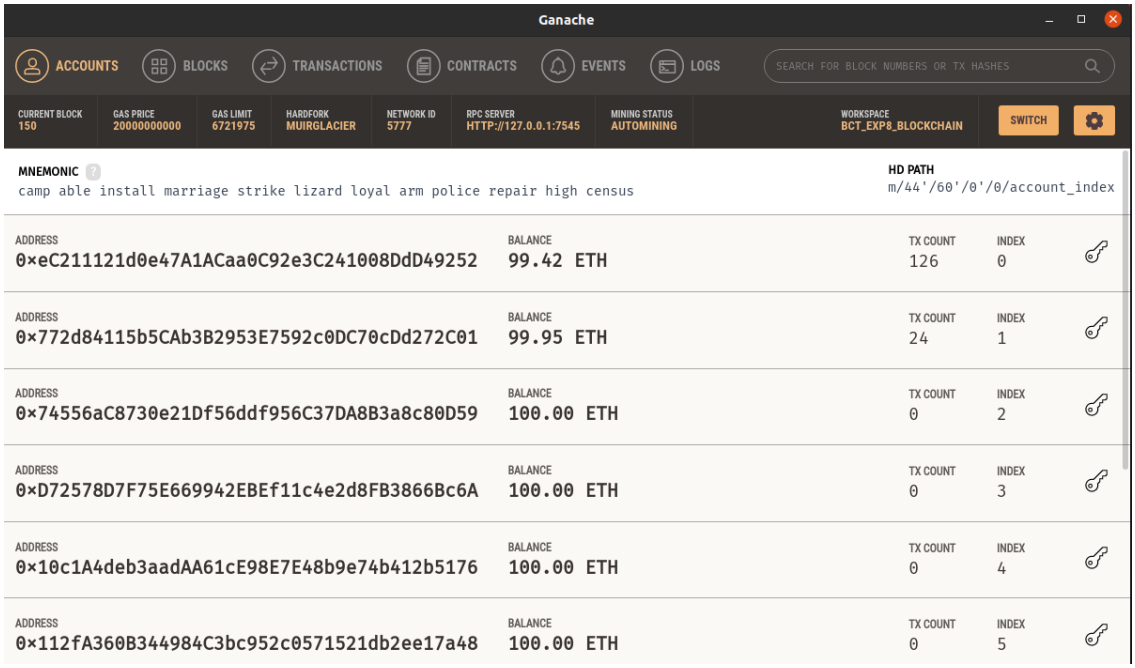


- Now create react app using the command: *npx create-react-app appname*
- Now change directory to the created react app
- Now type the following command to initiate a truffle project inside the react app
    - Command: *truffle init*
- Now the directory structure should look like this after installing all the required node dependencies

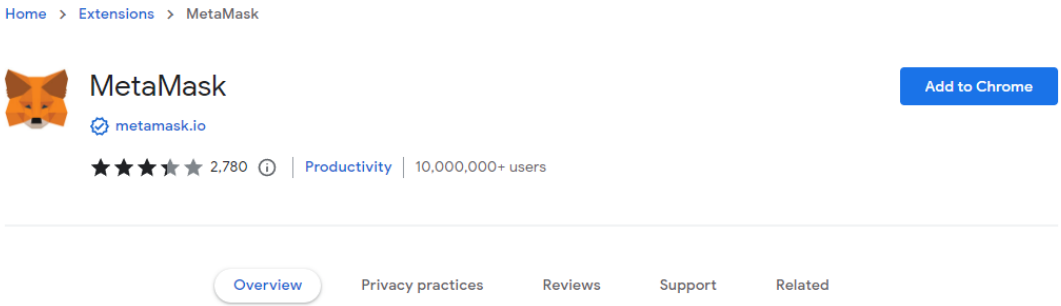



```
"@truffle/hdwallet-provider": "^1.1.0",
"babel-polyfill": "6.26.0",
"babel-preset-env": "1.7.0",
"babel-preset-es2015": "6.24.1",
"babel-preset-stage-2": "6.24.1",
"babel-preset-stage-3": "6.24.1",
"babel-register": "6.26.0",
"bootstrap": "^4.5.2",
"chai": "4.2.0",
"chai-as-promised": "7.1.1",
"chai-bignumber": "3.0.0",
"dotenv": "^8.2.0",
"identicon.js": "^2.3.3",
"ipfs-http-client": "^33.1.1",
"react": "^16.13.1",
"react-bootstrap": "^1.3.0",
"react-dom": "^16.13.1",
"react-scripts": "^3.4.3",
"truffle-hdwallet-provider-privkey": "^0.3.0",
"web3": "^1.3.0"
```

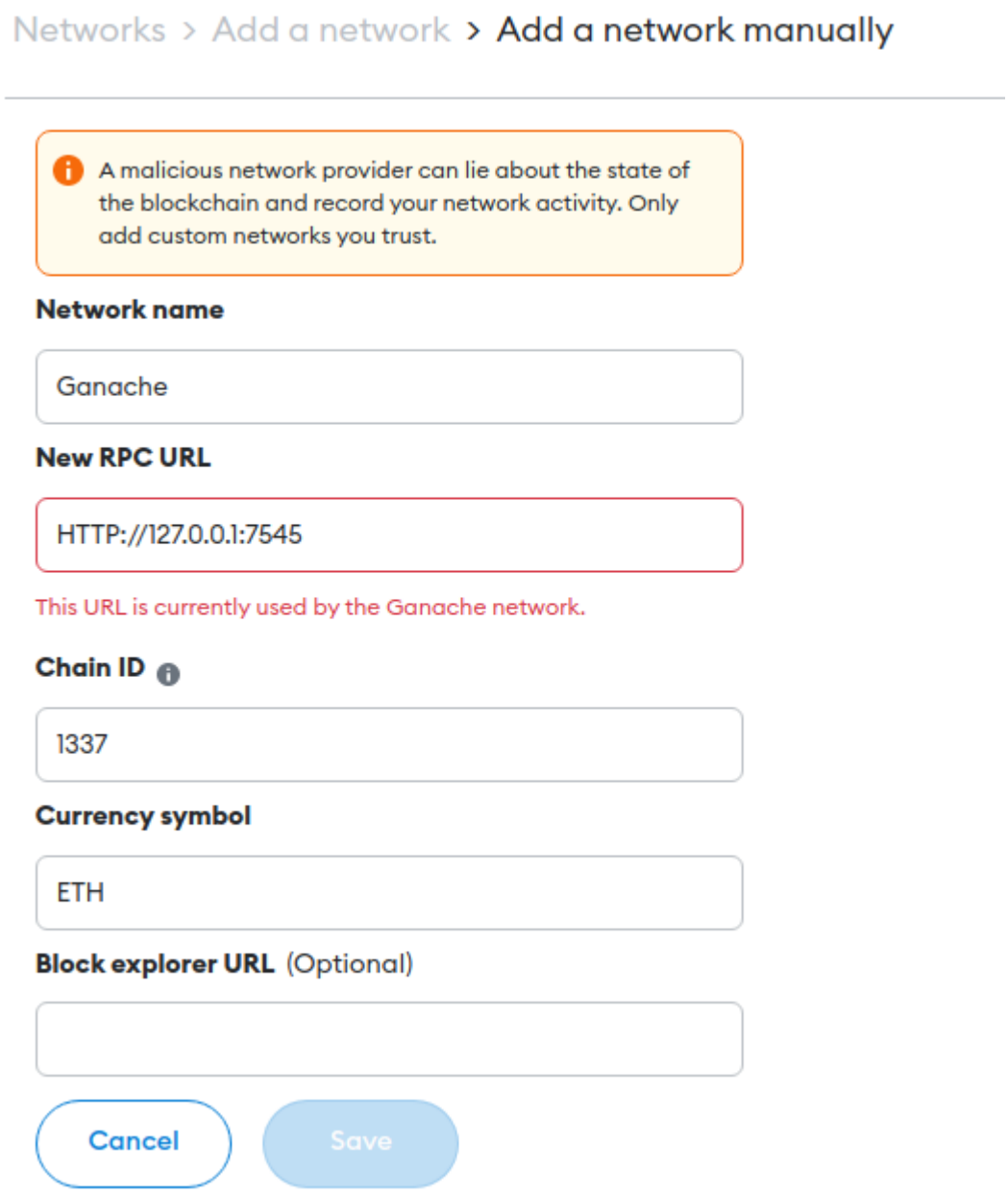
- Now we can setup ganache so that we can create a new workspace which would act as our private local blockchain which we would interact using our react application and meta mask



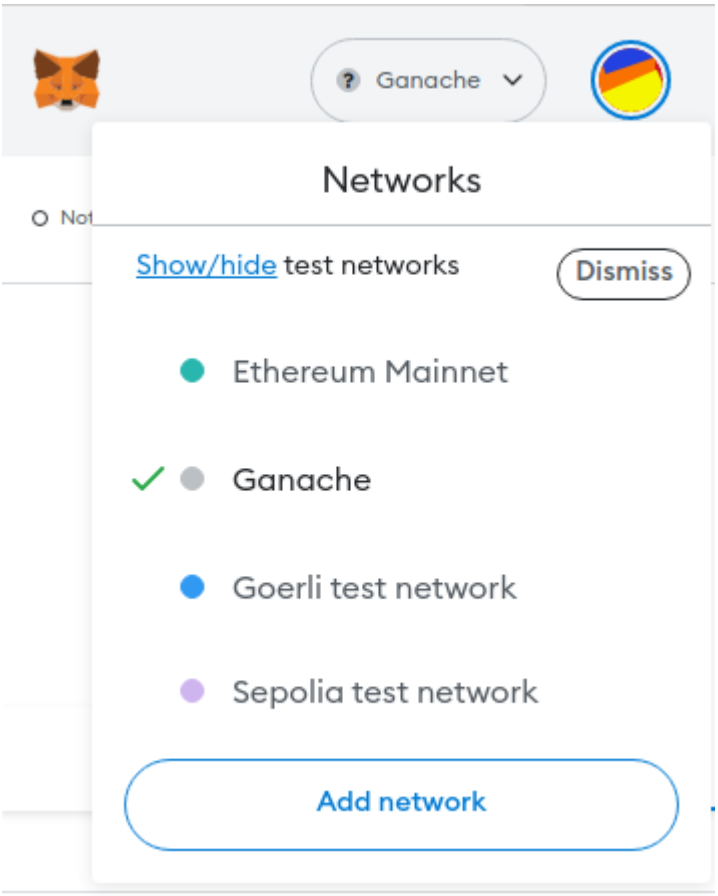
- Now we need to install metamask and create an account



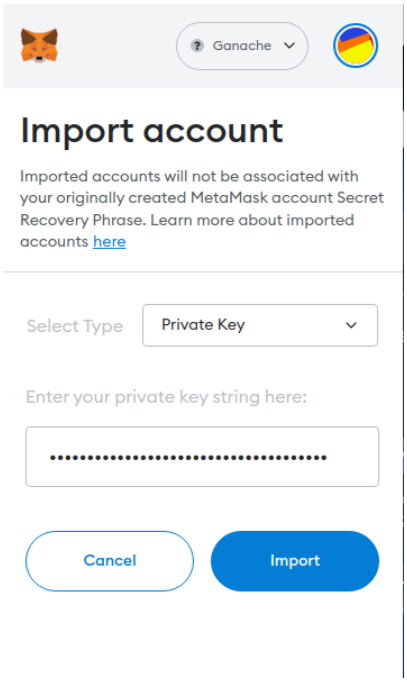
- After installing meta mask we need to create a new network which will connect our metamask wallet to the locally hosted private ganache blockchain

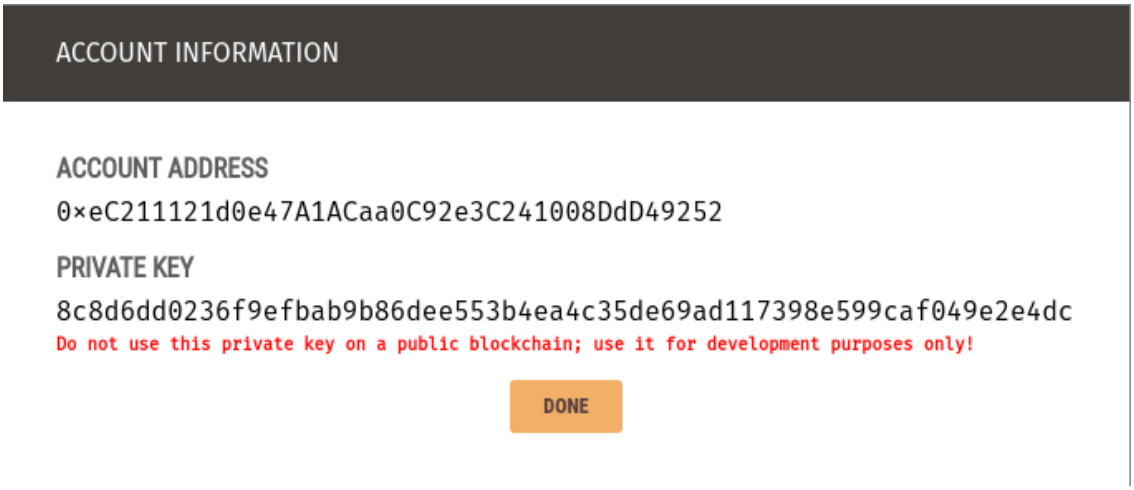


- Now click save and check for the newly created network



- Now we need to import an account from the ganache to metamask so that we can deploy and use the smart contract





- Now create the smart contract

```
pragma solidity ^0.5.0;

contract Youtube {
  uint public videoCount = 0;
  string public name = "Youtube";
  mapping(uint => Video) public videos;

  struct Video {
    uint id;
    string hash;
    string title;
    address author;
    string time;
  }

  event VideoUploaded(
    uint id,
    string hash,
    string title,
    address author,
    string time
  );

  constructor() public {
  }

  function uploadVideo(string memory _videoHash, string memory _title, string
memory _time) public {
    require(bytes(_videoHash).length > 0);
```

```
    require(bytes(_title).length > 0);
    require(msg.sender!=address(0));
    videoCount ++;
    videos[videoCount] = Video(videoCount, _videoHash, _title, msg.sender, _time);
    emit VideoUploaded(videoCount, _videoHash, _title, msg.sender, _time);
  }
}
```

- Create truffle config file to connect to ganache

```
networks: {
  development: {
    host: "127.0.0.1",
    port: 7545,
    network_id: "*" // Match any network id
  },
```

- Now we can write the logic of the application in App.js and Main.js Files

App.js

```
import React, { Component } from 'react';
import Youtube from '../abis/Youtube.json'
import Navbar from './Navbar'
import Main from './Main'
import Web3 from 'web3';
import './App.css';

class App extends Component {

  async componentWillMount() {
    await this.loadWeb3()
    await this.loadBlockchainData()
  }

  async loadWeb3() {
    if (window.ethereum) {
      window.web3 = new Web3(window.ethereum)
      await window.ethereum.enable()
    }
    else if (window.web3) {
      window.web3 = new Web3(window.web3.currentProvider)
    }
    else {
```

```
    window.alert('Non-Ethereum browser detected. You should consider trying
MetaMask!')
   }
  }

  async loadBlockchainData() {
   const web3 = window.web3
   const accounts = await web3.eth.getAccounts()
   this.setState({ account: accounts[0] })
   const networkId = await web3.eth.net.getId()
   const networkData = Youtube.networks[networkId]
   if(networkData) {
    const youtube = new web3.eth.Contract(Youtube.abi, networkData.address)
    this.setState({ youtube })
    const videosCount = await youtube.methods.videoCount().call()
    this.setState({ videosCount })

    console.log(this.state.videosCount);

    for (var i=videosCount; i>=1; i--) {
     const video = await youtube.methods.videos(i).call()
     this.setState({
      videos: [...this.state.videos, video]
     })
    }

    const latest = await youtube.methods.videos(videosCount).call()
    this.setState({
     currentHash: latest.hash,
     currentTitle: latest.title
    })
    this.setState({ loading: false})
   } else {
    window.alert('Youtube contract not deployed to detected network.')
   }
  }

  captureFile = event => {
   event.preventDefault()
   let url_link = event.target.value;
   console.log(url_link);
```

```
    let positionGoogle = url_link.search("google");
    let positionYoutube = url_link.search("youtube");
    if (positionGoogle != -1) {
      // do drive thing
      let viewPosition = url_link.search("view");

      url_link = url_link.slice(0, viewPosition) + "preview"
      this.setState({buffer: url_link})
    } else if (positionYoutube != -1) {
      // do youtube thing
      url_link = url_link.slice(0, 24) + "embed/" + url_link.slice(32, )
      this.setState({buffer: url_link})
    } else {
      alert("Wrong URL entered")
    }
    console.log(this.state.buffer);
  }


  uploadVideo = title => {
    console.log("Submitting file to IPFS...")
    let uploadTime = Date().toLocaleString()
    this.setState({currentVideoTime: uploadTime});
    this.state.youtube.methods.uploadVideo(this.state.buffer, title, uploadTime).send({
from: this.state.account })
  }

  changeVideo = (hash, title, time, id) => {
    this.setState({'currentHash': hash});
    this.setState({'currentTitle': title});
    this.setState({'currentVideoTime': time})
    this.setState({'videoNumber': id})
  }

  constructor(props) {
    super(props)
    this.state = {
      buffer: '',
      account: '',
      youtube: null,
      videos: [],
      loading: true,
```

```
      currentHash: null,
      currentTitle: null,
      currentVideoTime: null,
      videoNumber: null
    }

    this.uploadVideo = this.uploadVideo.bind(this)
    this.captureFile = this.captureFile.bind(this)
    this.changeVideo = this.changeVideo.bind(this)
  }

  render() {
    return (
      <div>
        <Navbar
          account={this.state.account}
        />
        { this.state.loading
          ? <div id="loader" className="text-center mt-5"><p>Loading...</p></div>
          : <Main
              videos={this.state.videos}
              uploadVideo={this.uploadVideo}
              captureFile={this.captureFile}
              changeVideo={this.changeVideo}
              currentHash={this.state.currentHash}
              currentTitle={this.state.currentTitle}
              currentVideoTime = {this.state.currentVideoTime}
              videoNumber = {this.state.videoNumber}
            />
        }
      </div>
    );
  }
}

export default App;
```

Main.js

```
import React, { Component } from 'react';

class Main extends Component {
```

```
 render() {
   return (
     <div className="container-fluid text-monospace">
        <br></br>
         
        <br></br>
        <div className="row">

        <div className="col-md-5 border overflow-auto text-center" style={{
maxHeight: '768px', minWidth: '175px' }}>
          <h5 style={{marginTop: '15px'}}><b>Upload Video</b></h5>
          <form onSubmit={(event) => {
            event.preventDefault()
            const title = this.videoTitle.value
            this.props.uploadVideo(title)
          }} >
             
            <div className="form-group">
              <input placeholder='Paste Video Link' id="exampleInputLink"
className="form-control" type='text' onChange={this.props.captureFile} />
            </div>
              <div className="form-group">
                <input
                  id="videoTitle"
                  type="text"
                  ref={(input) => { this.videoTitle = input }}
                  className="form-control"
                  placeholder="Enter Title..."
                  required />
              </div>
            <button type="submit" className="btn btn-success btn-block btn-
sm">Upload!</button>
             
          </form>
          <div className="row row-content">
          { this.props.videos.map((video, key) => {
            return(

              <div style={{
                display: "flex",
                flexDirection: "row",
```

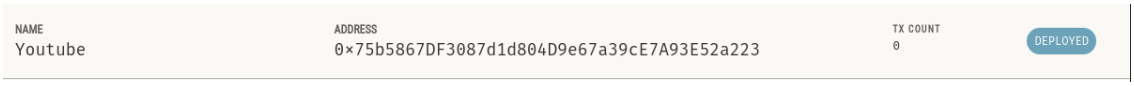
```
                flexWrap: "nowrap",
                justifyContent: "center"
                }}
                className="col-xl-6 col-lg-12 col-md-12 col-sm-12 my-4">
                <div className="card" style={{width: '20rem', borderRadius: '15px'}}
key={key} >
                  <div style={{borderTopLeftRadius: '15px', borderTopRightRadius:
'15px'}} className="card-img-top">
                   <iframe
                     style={{borderTopLeftRadius: '15px', borderTopRightRadius:
'15px'}}
                     width="100%"
                     height="200"
                     src={`${video.hash}`}
                    >
                   </iframe>
                  </div>
                  <div className="card-body">
                   <div className="card-title">
                    <p>Title: <b>{video.title}</b></p>
                    <p>Video Number: <b>{video.id}</b></p>
                    <p>Upload Time: <small>{video.time.slice(0, 25)}</small></p>
                   </div>
                   <button class="btn btn-primary" onClick={() =>
this.props.changeVideo(video.hash, video.title, video.time, video.id)}>View</button>
                  </div>
                  <div>
                  </div>
                 </div>
                </div>
              )
             })}
            </div>
           </div>
           <div className="col-md-7">
              <div className="embed-responsive embed-responsive-16by9" style={{
maxHeight: '768px'}}>
               <iframe
                style={{borderRadius: '15px'}}
                src={`${this.props.currentHash}`}
                controls
               >
```

```
            </iframe>
          </div>
          <br></br><hr></hr>
         <div>
          <h3 style={{marginTop: "2rem"}}><b>Title:
{this.props.currentTitle}</b></h3>
          <p>Video Number: {this.props.videoNumber}</p>
          <p>Upload Time: {this.props.currentVideoTime}</p>
         </div>
        </div>
       </div>
     </div>
   );
  }
}

export default Main;
```

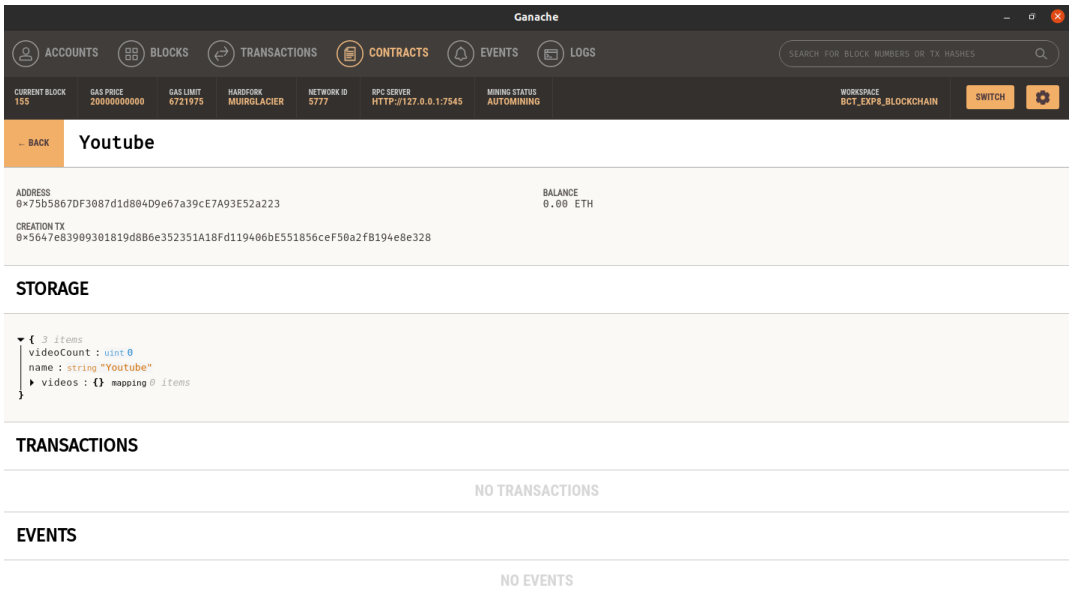
- Now we can compile the code and deploy it to the blockchain by the command: *truffle migrate –reset*

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

   Replacing 'Youtube'
   -------------------
   > transaction hash:    0x5647e83909301819d8b6e352351a18fd119406be551856cef50a2fb194e8e328
   > Blocks: 0            Seconds: 0
   > contract address:    0x75b5867DF3087d1d804D9e67a39cE7A93E52a223
   > block number:        154
   > block timestamp:     1669482171
   > account:             0xeC211121d0e47A1ACaa0C92e3C241008DdD49252
   > balance:             99.40073852
   > gas used:            703333 (0xabb65)
   > gas price:           20 gwei
   > value sent:          0 ETH
   > total cost:          0.01406666 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.01406666 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.0185714 ETH
```

- Now we can see the contract in the ganache app

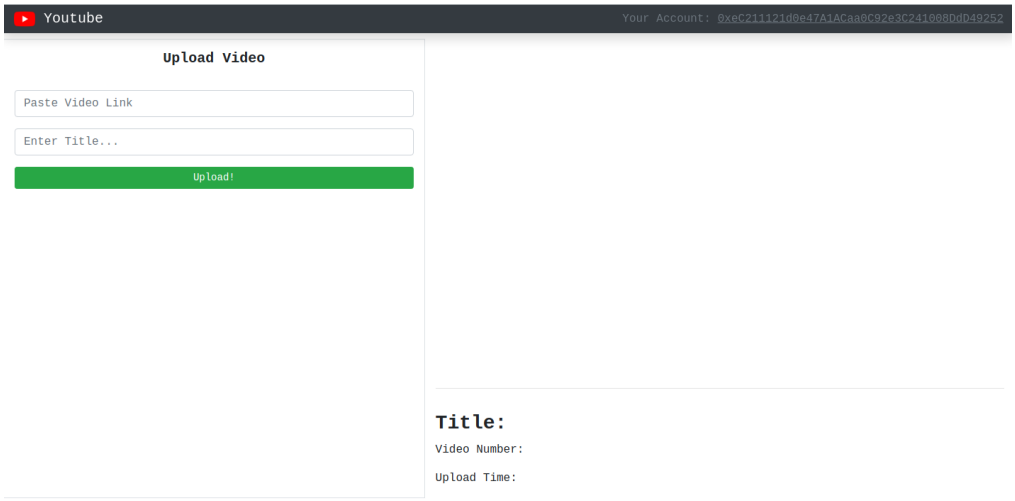


- Now we can click on it to see further information of the contract

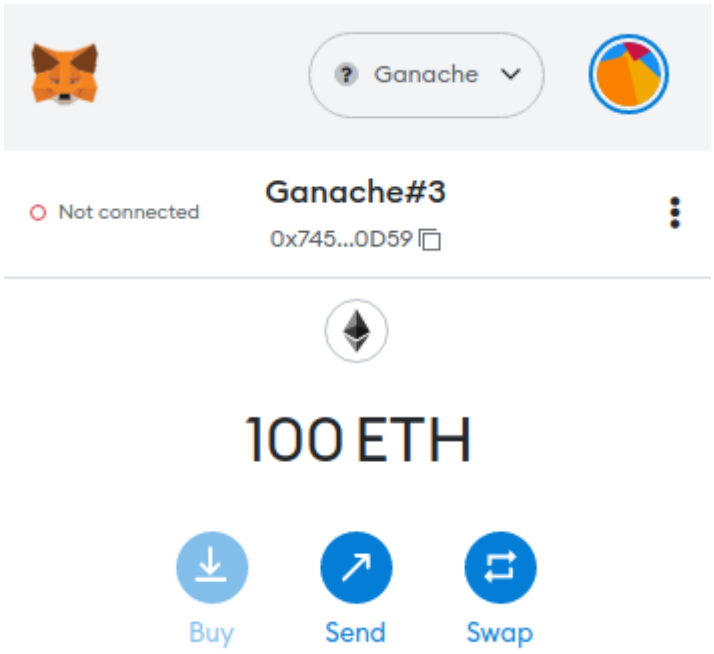


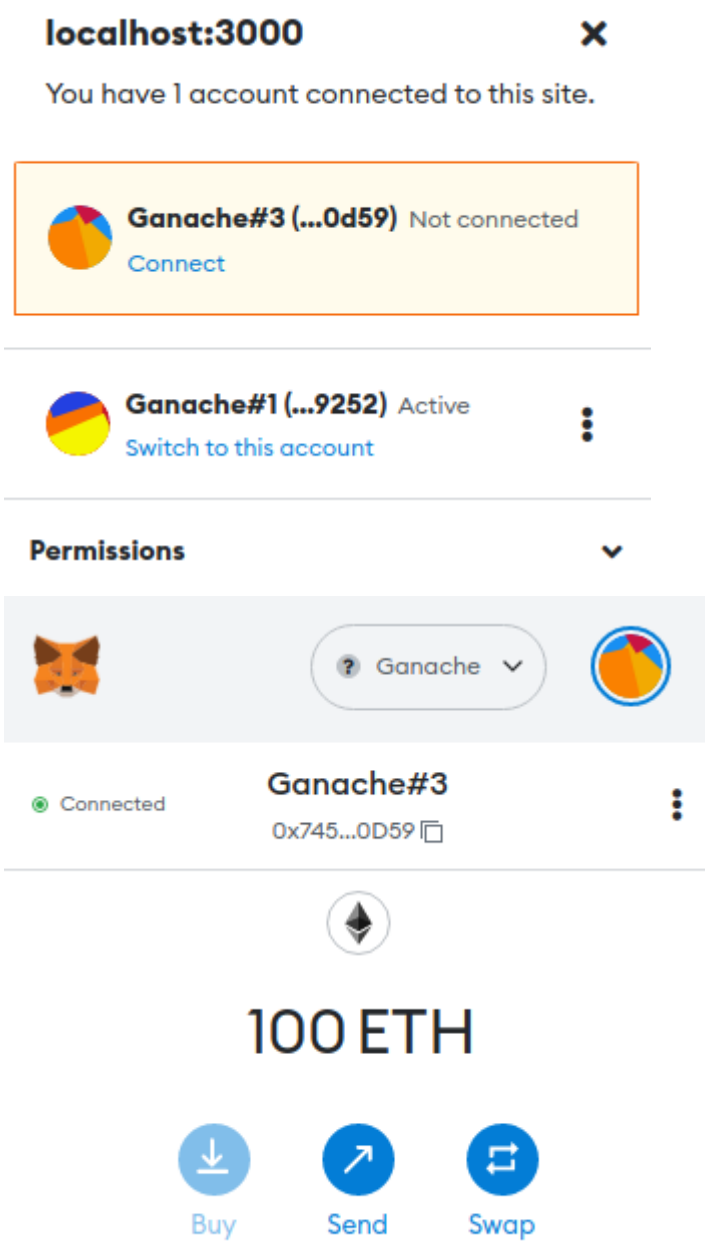
- Now we can run the project by the command: *npm start*

```
Compiled with warnings.

./src/components/Navbar.js
  Line 2:8:  'Identicon' is defined but never used  no-unused-vars

./src/components/Main.js
  Line 50:25:  <iframe> elements must have a unique title property  jsx-a11y/iframe-has-title
  Line 76:17:  <iframe> elements must have a unique title property  jsx-a11y/iframe-has-title

./src/components/App.js
  Line 67:24:  Expected '!==' and instead saw '!='  eqeqeq
  Line 73:32:  Expected '!==' and instead saw '!='  eqeqeq

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.
```



- Now lets connect a new account to the metamask wallet

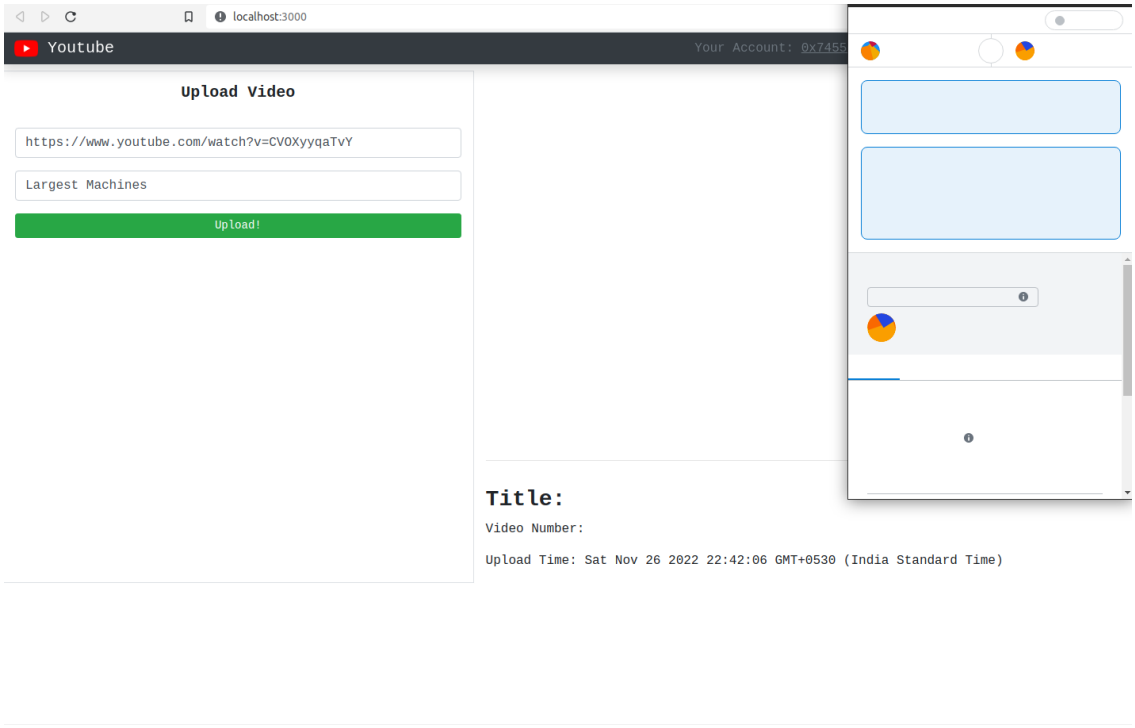


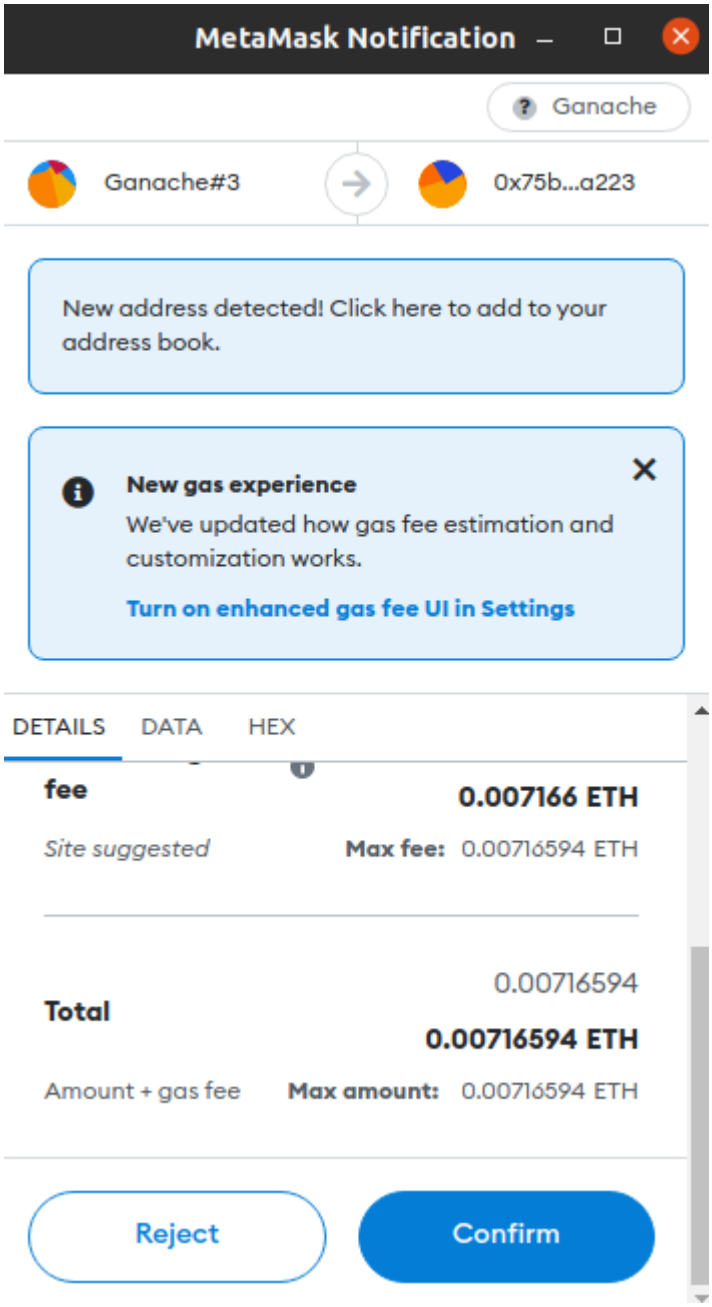


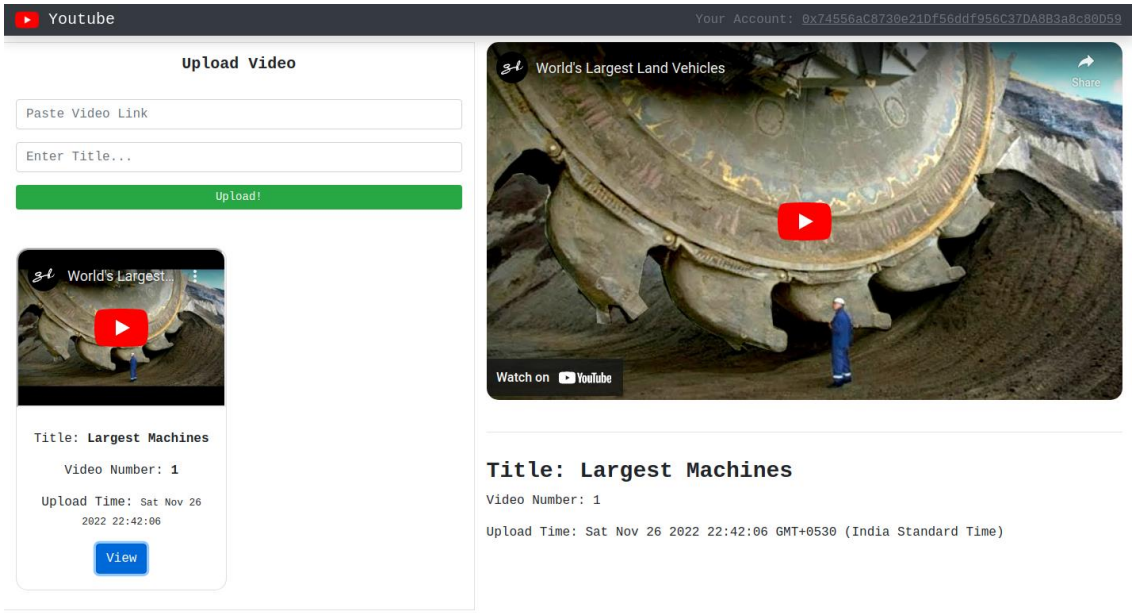
- We can see that after the new account is connected the account info changes on the website

```
Your Account: 0x74556aC8730e21Df56ddf956C37DA8B3a8c80D59
```

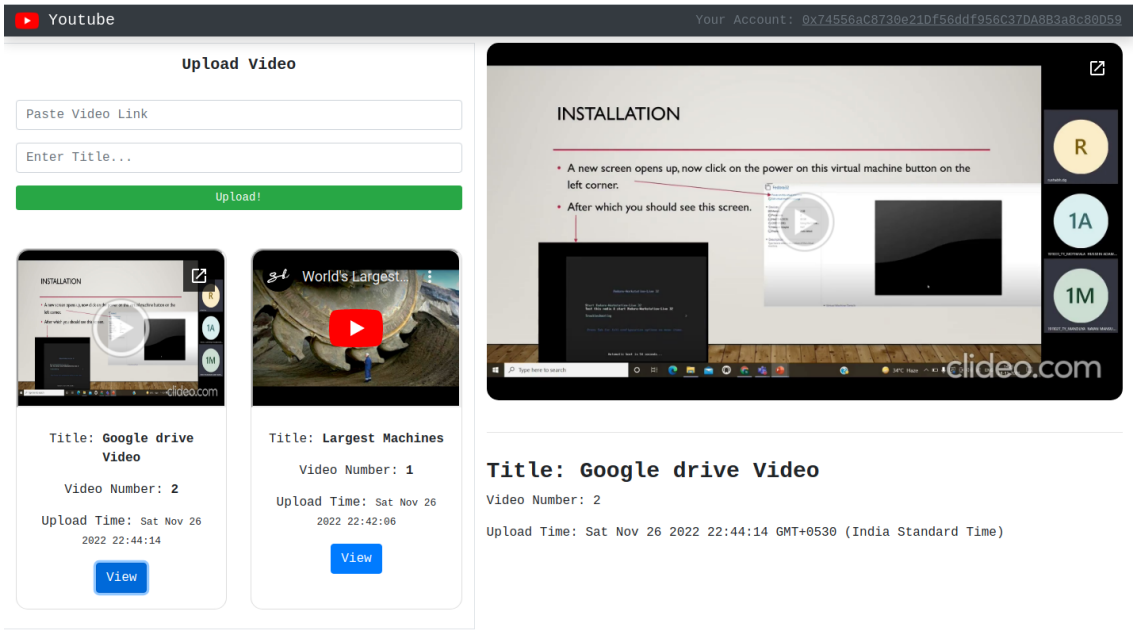
- Now lets upload a video having youtube link


localhost:3000
Youtube
Your Account: 0x7455
Upload Video
https://www.youtube.com/watch?v=CVOXyyqaTvY
Largest Machines
Upload!
Title:
Video Number:
Upload Time: Sat Nov 26 2022 22:42:06 GMT+0530 (India Standard Time)


MetaMask Notification
Ganache
Ganache#3
0x75b...a223
New address detected! Click here to add to your address book.
New gas experience
We've updated how gas fee estimation and customization works.
Turn on enhanced gas fee UI in Settings
DETAILS
DATA
HEX
fee
0.007166 ETH
Site suggested
Max fee: 0.00716594 ETH
Total
0.00716594
0.00716594 ETH
Amount + gas fee
Max amount: 0.00716594 ETH
Reject
Confirm



- Now lets upload a google drive video



- We can click on the view button of any video to see that video in the bigger area.

GitHub Link: https://github.com/hussein-hub/BCT_EXP8

2. Explain your program logic, classes and methods used.

Methods:

- uploadVideo()
- loadweb3()
- loadBlockchainData()
- captureFile()
- uploadVideo()
- changeVideo()

3. Explain the Importance of the approach followed by you

I have used truffle and Ganache because of the features and clean documentation they provide

Features of Ganache

- Displays blockchain log output
- Provides advanced mining control
- Built-in block explorer
- Ethereum blockchain environment
- Ganache has a desktop application as well as a command-line tool

Features Of Truffle Ethereum:

- Built-in support to Compile, Deploy and Link smart contracts
- Automated Contract testing
- Supports Console apps as well as Web apps
- Network Management and Package Management
- Truffle console to directly communicate with smart contracts
- Supports tight integration

Conclusion: - Understand how to create a decentralized application using ganache, truffle, Ethereum and React. How to write smart contracts and deploy them on the blockchain.