Summer Project On

CrypTrade

By

Abhishek Bawachkar 2021510004

Under the guidance of

Internal Supervisor

Prof.Arti Karande

Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University

2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Abhishek Bawachkar 2021510004

Have satisfactorily carried out work on the project entitled

"CrypTrade"

Towards the fulfilment of project, as laid down by Sardar Patel Institute of Technology during year 2022-23.

Project Guide:
Prof. Arti Karande

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Abhishek Bawachkar 2021510004

Have successfully completed the Project report on

"CrypTrade",

which is found to be satisfactory and is approved

at

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI

INTERNAL EXAMINER EXTERNAL EXAMINER

HEAD OF DEPARTMENT PRINCIPAL

Contents

Abstract

CrypTrade webapp is a website/webapp which can be used to trade crypto currencies via a decentralised method which keeps the transaction safe and secure. All the latest crypto currencies will be displayed.

The user can login into the site keep a track of all the currencies which will be updated real time via api call. User can buy and trade currencies as it is DAPP (Decentralised App) it is the most secure way of trading.

This site will provide a gateway for people to trade/exchange/buy crypto currencies.

Objectives

The web based app "CrypTrade" is used

- To provide a platform to trade/buy/exchange Crypto Currencies.
- To provide a convenient medium to keep track of all the Currencies.
- To provide a facility to keep an eye on top 10 crypto currencies which will be updated real time via API call.

List of Figures

List of Tables

1 Introduction

1.1 Problem Definition

To eliminate online scam regarding Crypto trading which leads into huge losses and manipulation of the market. To make it easy for user to choose from large variety of coins.

1.2 Objectives and Scope

1.2.1 Objectives

The web based app "CrypTrade" is

- To provide a platform to trade/buy/exchange Crypto Currencies.
- To provide a convenient medium to keep track of all the Currencies.
- To provide a facility to keep an eye on top 10 crypto currencies which will be updated real time via API call.

1.2.2 Scope

A user can provide his/her wallet details in Metamask and can login into their account which store all the info regarding their account and currencies.

In this web app a user can surf the site there is no need to login to use the base functionality of keeping a track of all the coins. So if someone wants to learn and keep a watch on the prices which keeps on changing user can use this website.

This site is being made for reducing efforts of trading and buying a crypto on top of managing a wallet. As all the data is stored in a database(Moralis).

1.3 Proposed System

The user is a person seeking a platform to trade it crypto currencies who will register on this site via MetaMask Wallet. MetaMask is a wallet to store all your coins which is safe and secure. This System will connect users with each other so that they can trade coins with each other. This platform will provide a top 10 views of coins which will constantly change their prices real time.

The user will create its own id and it will be stored in moralis database which is considered as very functional and important database for DAPP.

User can also create multiple account and can trade anytime, anywhere as it is a web app user get's the flexibility of platform.As the

Some of the advantages of our system are as follows :

- Instant Trade
  Users can instantly trade coins with ease as the site is based on smart contracts which directly sends crypto coins after triggering a contract.
- Top 10 Coins at one place
  User can view list of top 10 coins as per their prices all together.
- Privacy and Security
  As this app follows the rule set of DAPP's it is more secure thanks to smart contracts and blockchains/nodes.

1.4 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Android Device with Touch Screen minimum 5” inch Display or any desktop
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	Moralis

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Operating System	Android/IOS Smartphone/Desktop
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of our project is to develop an web based UI application that can help user to easily access their wallet and can trade/exchange/buy crypto coins at ease..
This can save lots of efforts of user's and trader and it will be easier to store the each and every single and small detail regarding the transaction in the database (moralis). This web app will keep track of all the information regarding the history of trade including user wallet id, transaction amount and the price updates.

2.2 Definition

To build a web based Crypto Trading Application so the user can have an easy trading experience.

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. Authentication: Users are required to Sign-up and Log-in using Meta Mask Wallet. Users will get a verification email for successful registration.
2. Profile: This will contain information regarding wallet id and all the recent purchased crypto.
3. Top 10 : This will contain information regarding top 10 crytpo currencies.
4. View Coins: It allows user to check information of each and every coin.

2.3.2 User Characteristics

There are two types of users:

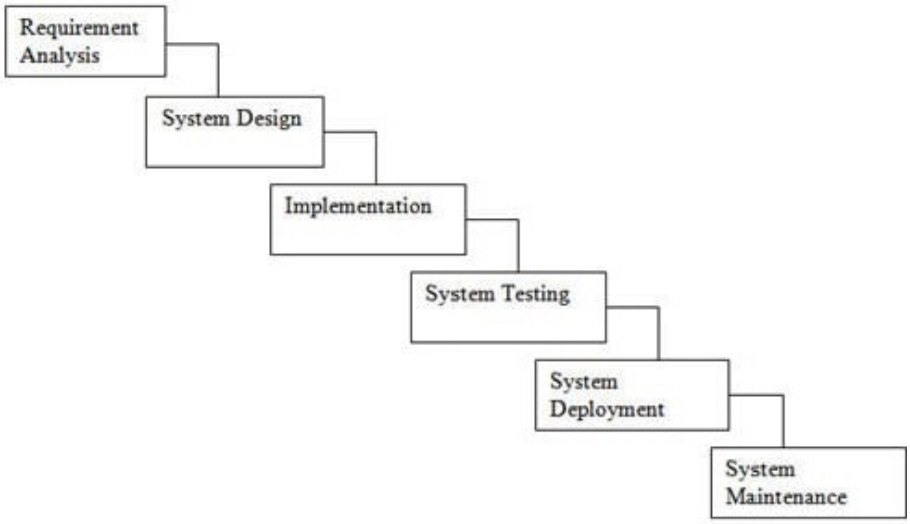
- Surfer: A surfer will be a type of a user where user can only surf the site user doesn't need to login or create an account in order to access. User can visit this website anytime and can check the volatile prices of crypto currencies.
- Customers: Customers are the type of users which can login into site or can register and can use all the functionalites of the site ie buy/trade/exchange.

3 Project Analysis and Design

3.1 Methodologies Adapted

In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

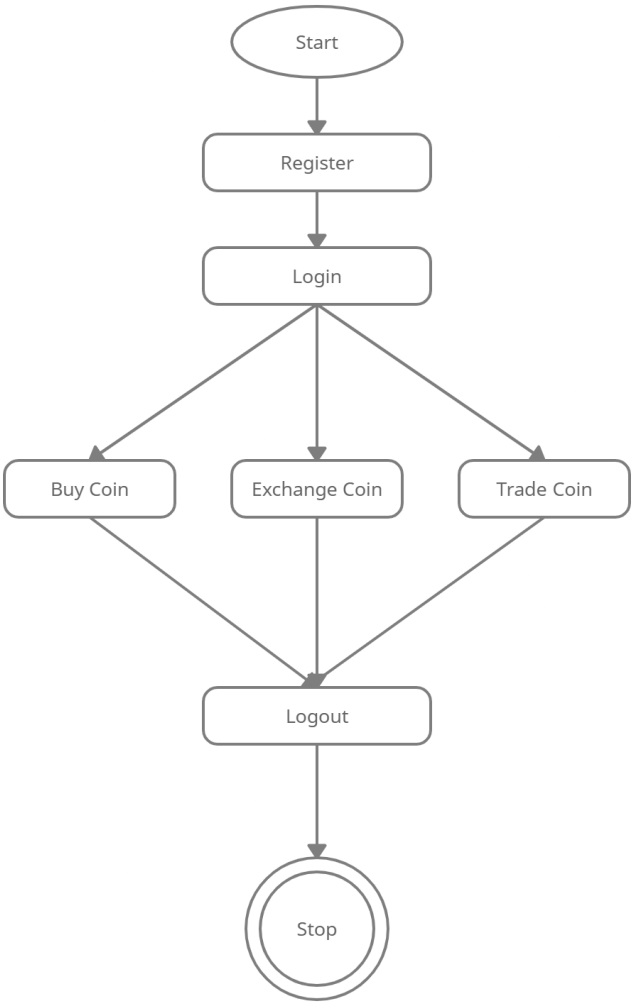
Once the product is developed and if any failure occurs then the cost of such issues is very high, because we need to update everything from document till the logic.



3.1.1: Diagrammatic Representation of Waterfall Model

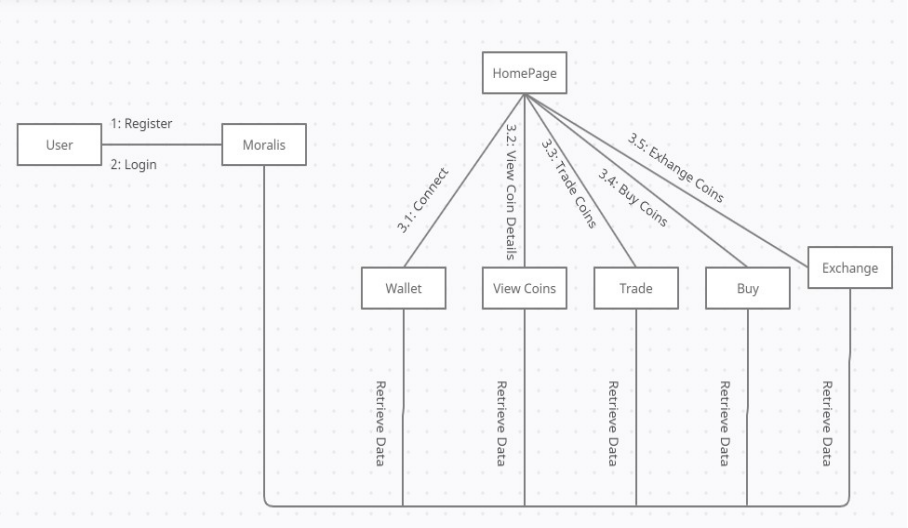
3.2 Modules

3.2.1 Activity diagram



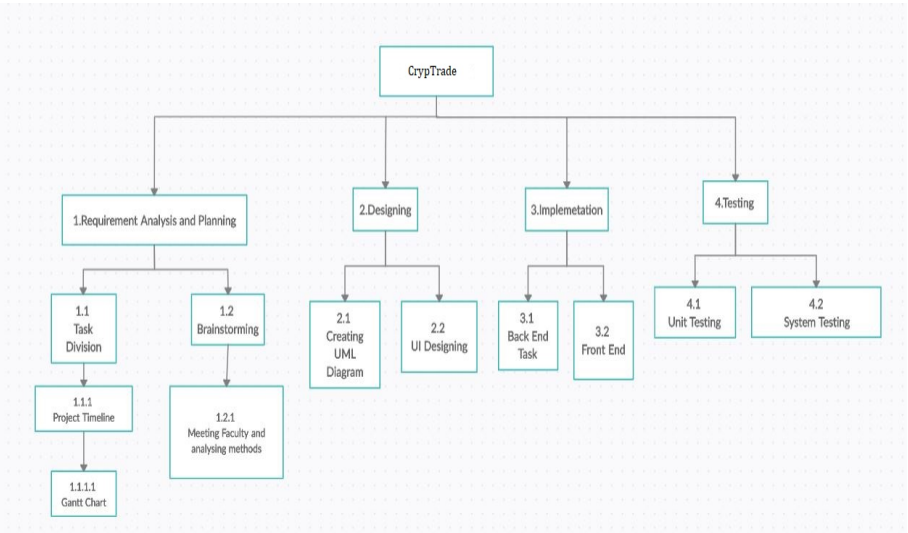
3.2.1: Activity Diagram

3.2.2 Communication Design



3.2.2: Communication Diagram

3.2.3 Work Breakdown Structure



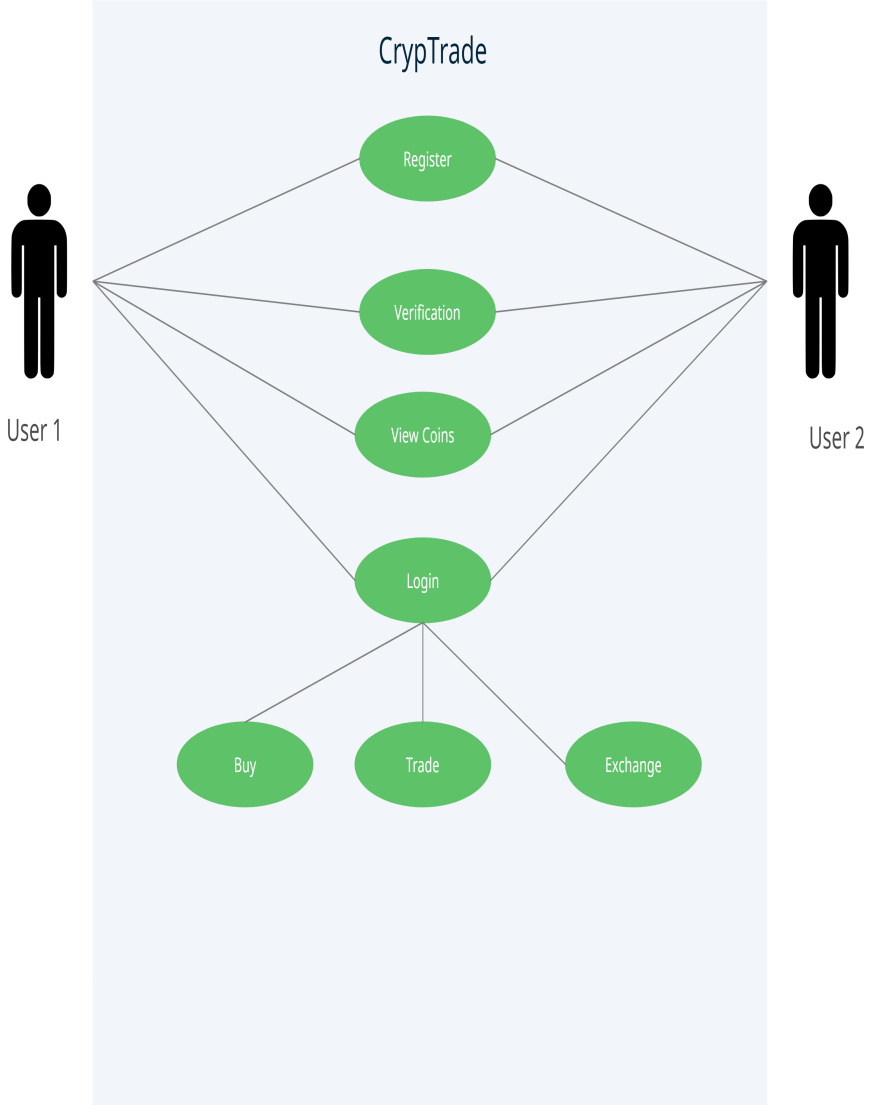
3.2.3: Work Breakdown Structure

3.2.4 Gantt Chart

	Name	Assigned To	Start	End	Dur	%	2022
							May Jun Jul Aug Sep
	CrypTrade		24/5/22	24/9/22	89		
1	Problem Statement		24/5/22	30/5/22	5		
2	Brainstorming		31/5/22	13/6/22	10		
3	Requirement Gathering		13/6/22	23/6/22	9		
4	Project Planning		24/6/22	29/6/22	4		
5	Analysis		29/6/22	12/7/22	10		
6	Design		12/7/22	25/7/22	10		
7	Implementation		25/7/22	16/9/22	40		
8	Testing		16/9/22	20/9/22	3		
9	Documentation		20/9/22	24/9/22	4		

3.2.4: Gantt Chart

3.2.5 Use-Case



3.2.5: Use-Case Diagram

Use Cases:

1. Register
2. Login
3. Update Profile
4. View Coins
5. Buy Coins
6. Trade Coins
7. Exchange Coins

Table 4.2.1: Use Case Table - Register

Use Case ID	1
Use Case Name	Register
Actor	User and Customer
Pre-Condition	They must register themselves first
Post-Condition	Customer can login
Flow of events	Login,Register or Edit details and view or buy/trade/exchange coins

Table 4.2.2: Use Case Table - Login

Use Case ID	2
Use Case Name	Login
Actor	Customers
Pre-Condition	They must register themselves first
Post-Condition	Customer can view details of coins and can purchase/buy/trade coins
Flow of events	Login,Register or Edit details and Customer can view details of coins and can purchase/buy/trade coins

Table 4.2.3: Use Case Table - Update Profile

Use Case ID	3
Use Case Name	Update Profile
Actor	Customers
Pre-Condition	Login
Post-Condition	User can view or edit the profile

Table 4.2.4: Use Case Table - View Coins

Use Case ID	4
Use Case Name	View Coins
Actor	User and Customer
Pre-Condition	None
Post-Condition	User and customer can view coins and their details

Table 4.2.5: Use Case Table - Buy Coins

Use Case ID	5
Use Case Name	Buy Coins
Actor	Customer
Pre-Condition	Login
Post-Condition	Can buy coins

Table 4.2.6: Use Case Table - Trade Coins

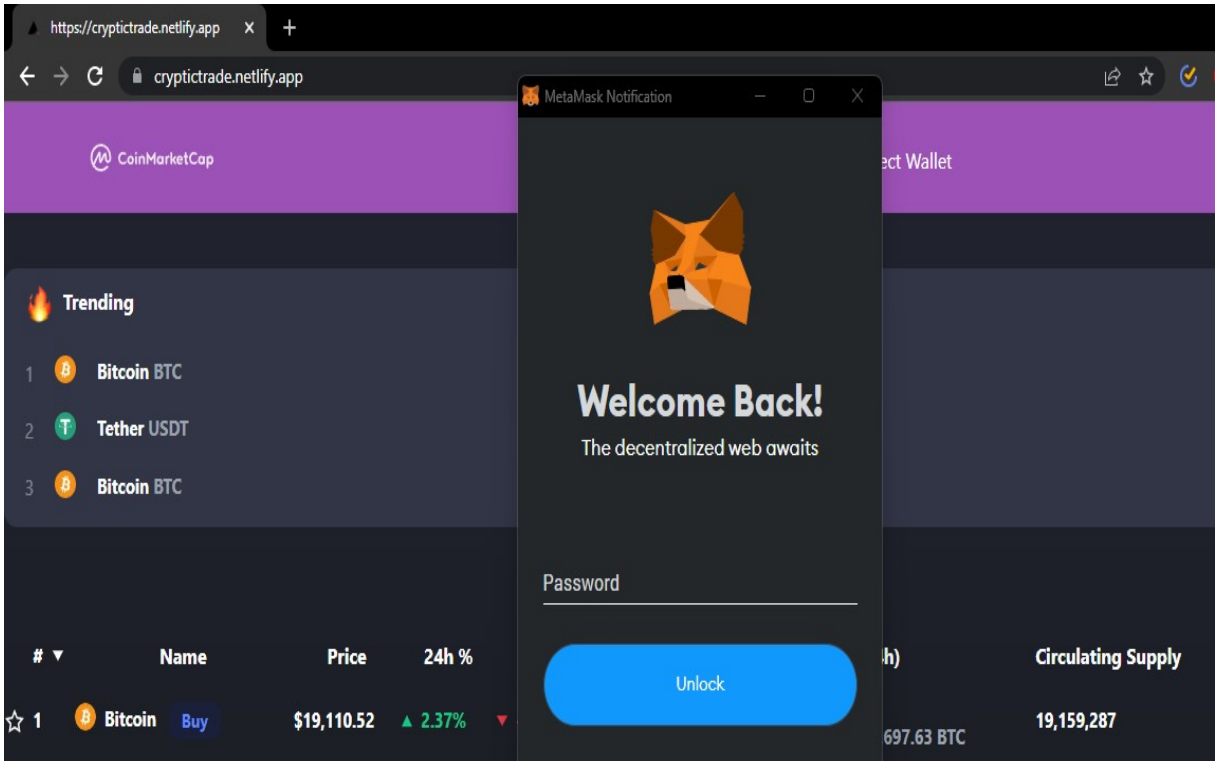
Use Case ID	6
Use Case Name	Trade Coins
Actor	TPC
Pre-Condition	Login
Post-Condition	Can Trade Coins

Table 4.2.7: Use Case Table - Exchange Coins

Use Case ID	7
Use Case Name	Exchange Coins
Actor	TPC
Pre-Condition	Login
Post-Condition	Can exchange coins online with other coins.

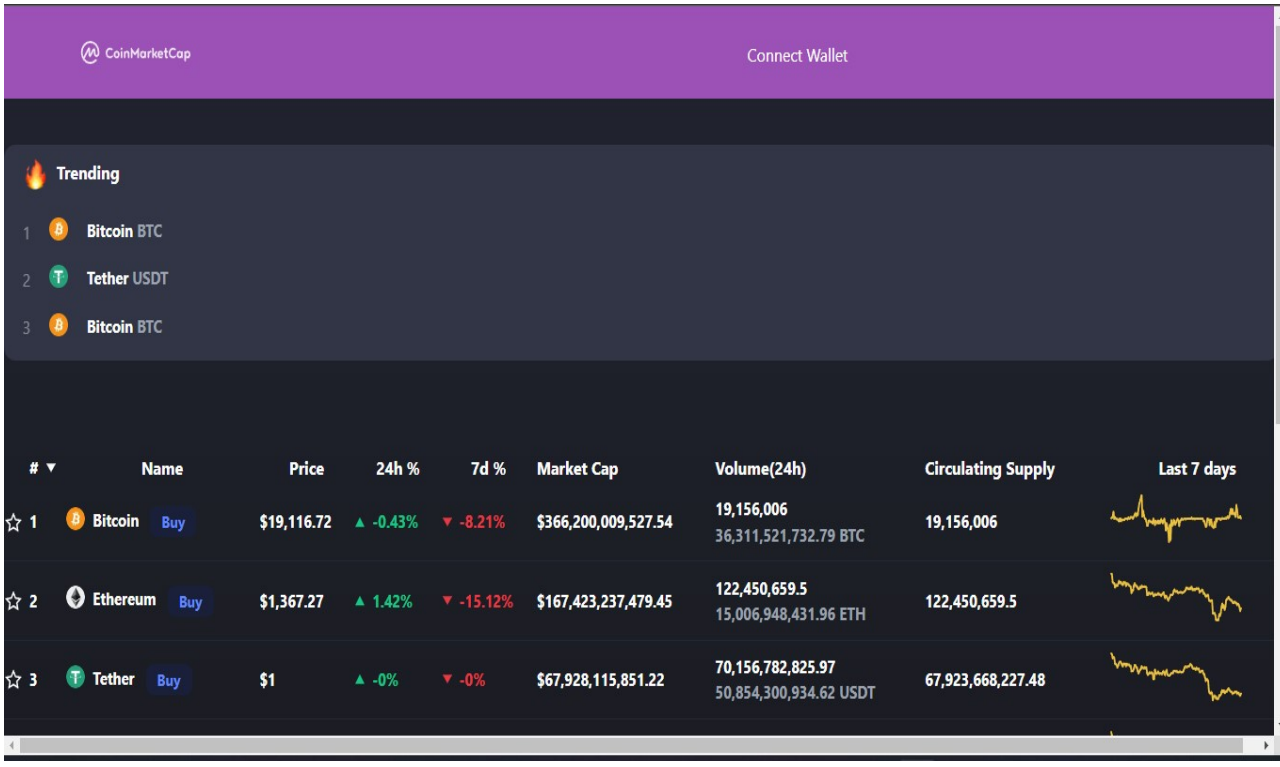
4 Project Implementation and Testing

4.1 Login and Register



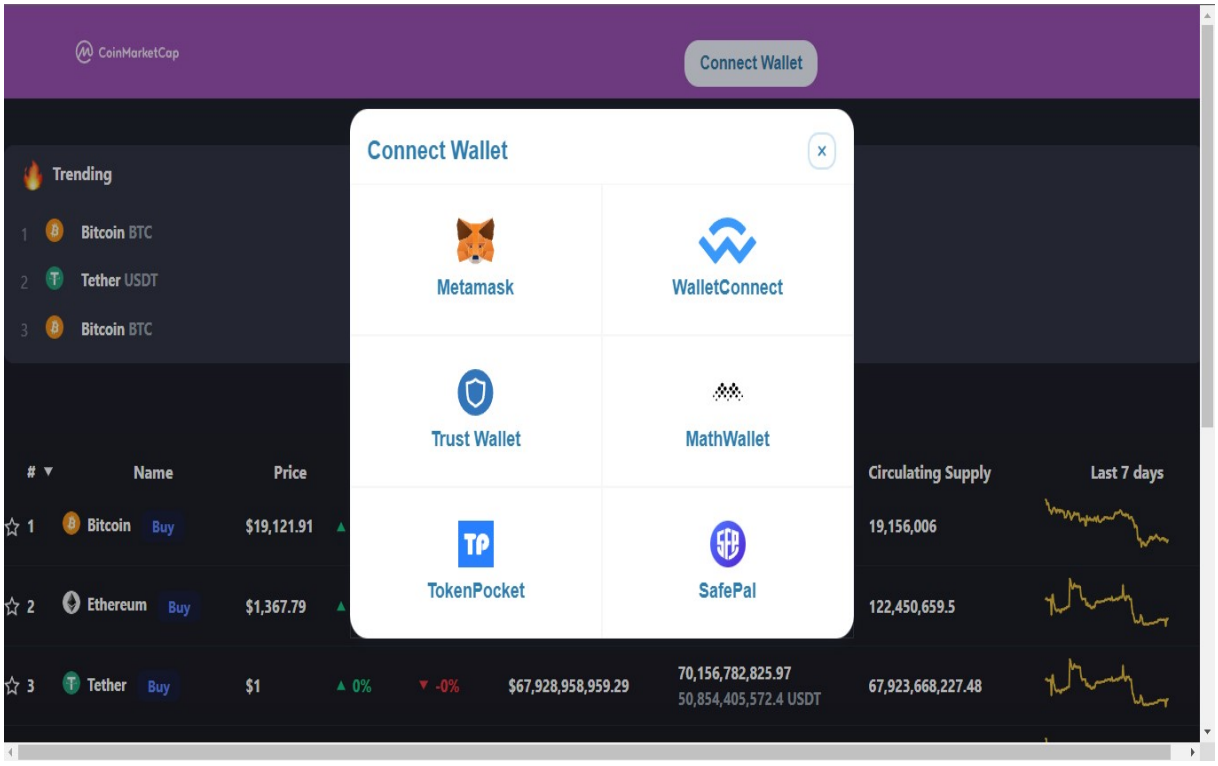
4.1.1: Login and Register

4.2 HomePage View



4.2.1: HomePage View

4.3 Connect wallet



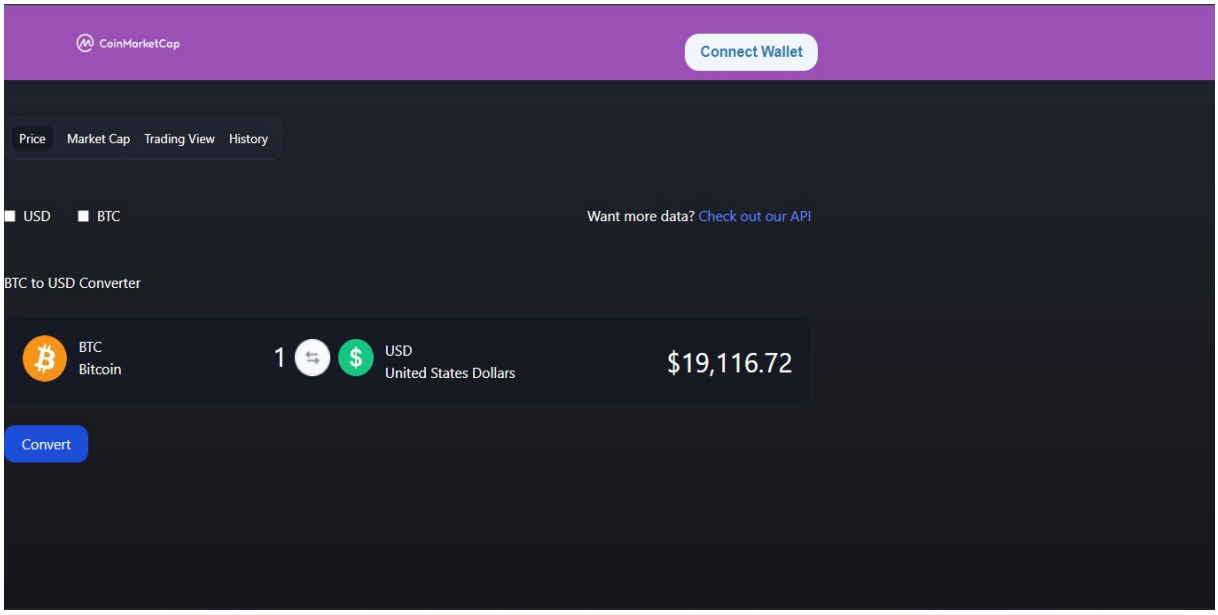
4.3.1: Connect Wallet

4.4 CoinList

#	Name	Price	24h %	7d %	Market Cap	Volume(24h)	Circulating Supply	Last 7 days
1	Bitcoin Buy	$19,121.91	▲ -0.5%	▼ -8.15%	$366,299,477,706.3	19,156,006 36,326,959,455.42 BTC	19,156,006	
2	Ethereum Buy	$1,367.79	▲ 1.35%	▼ -15.06%	$167,486,850,654.27	122,450,659.5 15,006,777,552.35 ETH	122,450,659.5	
3	Tether Buy	$1	▲ 0%	▼ -0%	$67,928,958,959.29	70,156,782,825.97 50,854,405,572.4 USDT	67,923,668,227.48	
4	USD Coin	$1	▲ 0%	▼ -0%	$50,192,263,755.38	50,192,523,446.7 4,913,820,361.94 USDC	50,192,523,446.7	
5	BNB	$269.65	▲ 1.06%	▼ -4.84%	$43,505,003,892.74	161,337,261.09 884,783,080.9 BNB	161,337,261.09	
6	XRP	$0.41	▲ 15.36%	▼ 21.79%	$20,613,515,019.17	99,989,294,935 3,793,377,448.09 XRP	49,848,747,475	
7	Binance USD	$1	▲ -0.09%	▼ -0.03%	$20,510,466,375.84	20,517,253,084.59 8,211,029,694.44 BUSD	20,517,253,084.59	
8	Cardano	$0.45	▲ -0.62%	▼ -5.64%	$15,249,292,188.76	34,854,947,575.48 562,922,645.96 ADA	34,182,044,152.63	
9	Solana	$32.29	▲ 0.7%	▼ -7.82%	$11,429,584,942.95	511,616,946.14 857,260,799.14 SOL	353,927,589.41	

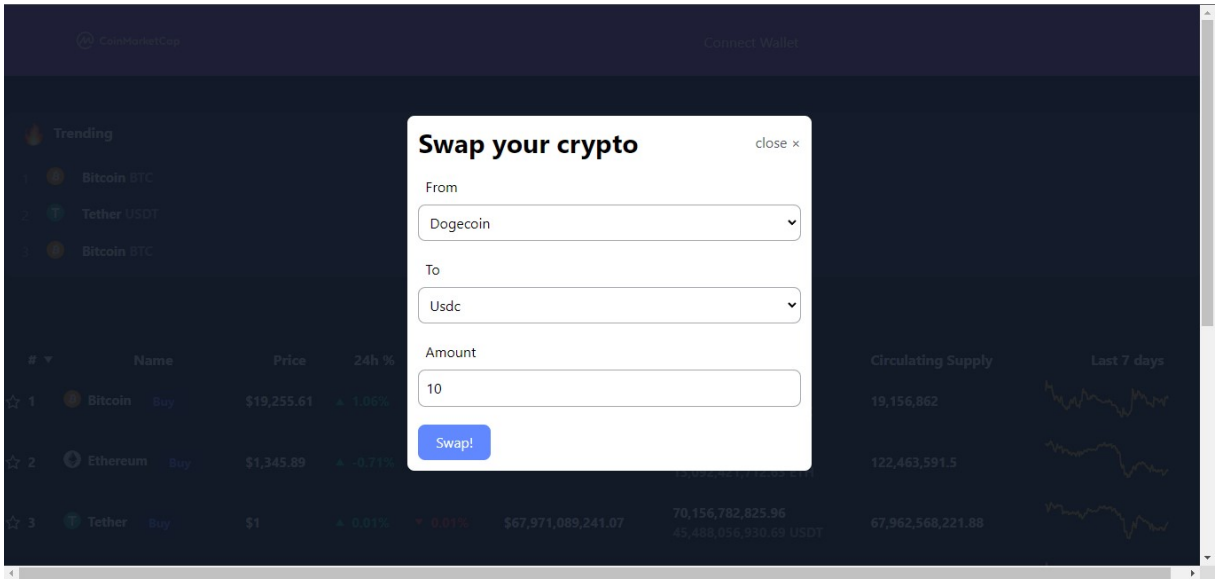
4.4.1: Coin List

4.5 Coin Details



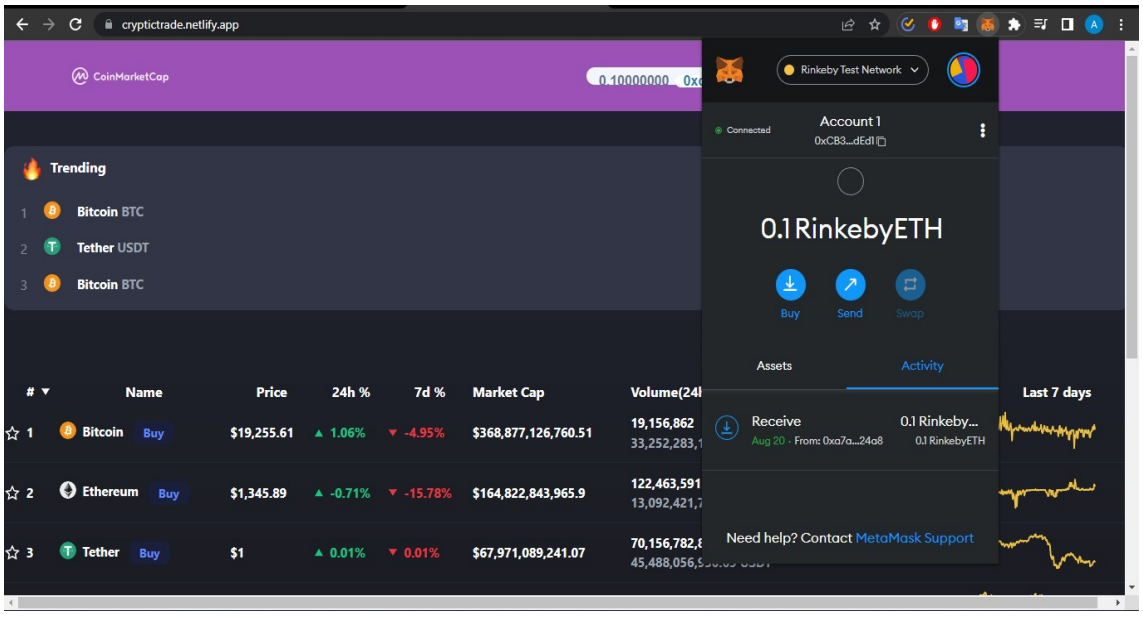
4.5.1: Coin Details

4.6 Swap Coin



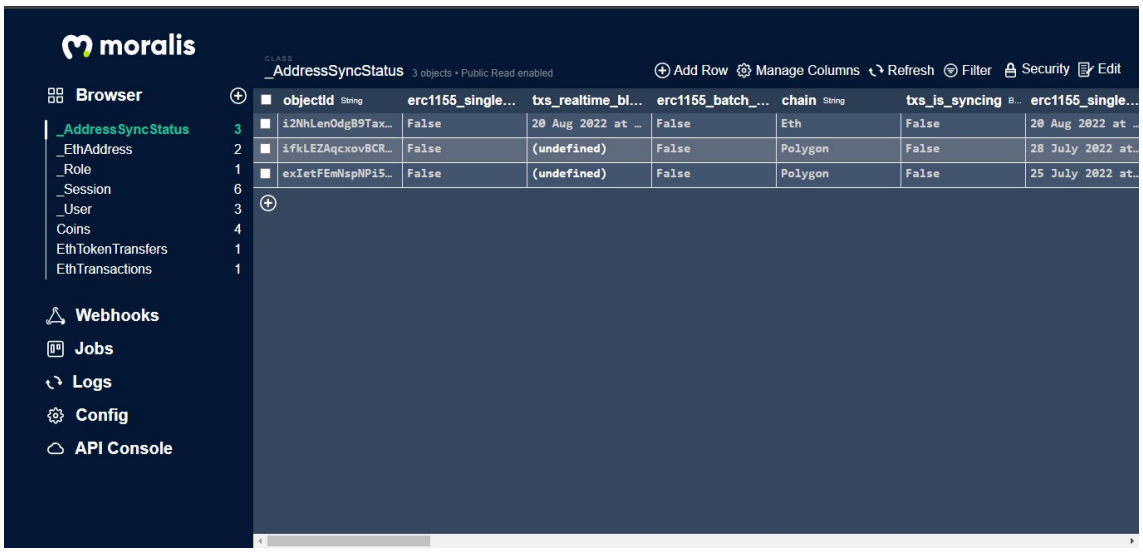
4.6.1: Swap Coin

4.7 Complete Ui



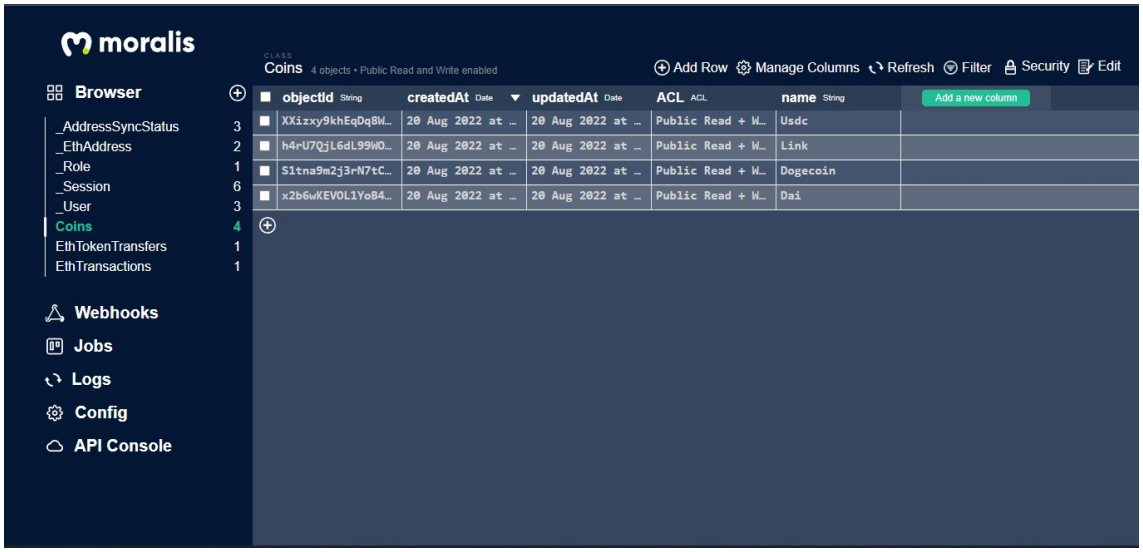
4.7.1: Complete UI

4.8 Database



4.8.1: Database

4.9 Coin Database



4.9.1: Coin Database

4.10 Code 1

components > cmc-table > CMCTable.js > ...

```
import { useContext, useEffect, useState, useCallback } from 'react'
import btc from '../../assets/btc.png'
import { CoinMarketContext } from '../../context/context'
import CMCtableHeader from './CMCtableHeader'
import CMCtableRow from './CMCtableRow'

const CMCtable = () => {
  let { getTopTenCoins } = useContext(CoinMarketContext)
  let [coinData, setCoinData] = useState(null)

  useEffect(() => {
    setData()
  }, [])

  const setData = useCallback(async () => {
    try {
      let apiResponse = await getTopTenCoins()
      let filteredResponse = []

      for (let i = 0; i < apiResponse.length; i++) {
        const element = apiResponse[i]
        if (element.cmc_rank <=10) filteredResponse.push(element)
      }

      setCoinData(filteredResponse)
    } catch (e) {
      console.log(e.message)
    }
  }, [getTopTenCoins])

  return (
    <div className='text-white font-bold'>
```

4.11 Code 2

pages > currencies > price.js > Price

```
import { useState, useEffect } from "react"
import Header from "../../components/header"

const Price = () => {
    const [coinName, setCoinName] = useState('')
    const [coinSymbol, setCoinSymbol] = useState('')
    const [price, setPrice] = useState('')

    useEffect(() => {
        getData()
    }, [])

    const getData = async () => {
        const queryString = window.location.search;
        const urlParams = new URLSearchParams(queryString);

        setCoinName(urlParams.get('coin'));
        setPrice(Number(urlParams.get('price')).toLocaleString(0));
        setCoinSymbol(urlParams.get('symbol'));
    }

    return <div>
        <Header />
    </div>
}

export default Price
```

4.12 Code 3

pages > api > getTopTen.js > handler

```
export default function handler(req, res) {
    const getData = async () => {
      const response = await fetch(
        `https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest?CMC_PRO_API_KEY=${process.env.CMC_API_KEY}`,
        {
          method: 'GET',
          headers: {
            Accept: '*/*',
          },
        },
      )

      const data = await response.json()

      res.status(200).json({ data })
    }

    getData()
  }
```

4.13 Code 4

lib > constants.js > usdcAddress

```
import doge from './Dogecoin.json'
import dai from './Dai.json'
import link from './Link.json'
import usdc from './Usdc.json'

export const dogeAbi = doge.abi
export const daiAbi = dai.abi
export const linkAbi = link.abi
export const usdcAbi = usdc.abi

export const dogeAddress = '0x1DF8B43f64Eb0C8296fb3451CBF344ABc5c53029'
export const linkAddress = '0x309707740c96fd92dC541c86a639C0083a2AD4Be'
export const daiAddress = '0x934eEA3f90a868F4DB5E110Ae1f31991B56D1Bbd'
export const usdcAddress = '0x17E7bF33E89624F257604AE7febb3B0f3751B05a'
```

5 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enter user id and password	Enters the correct user id and password	Log in Successful	Home Page	Pass
2	User enter user id and password	Enters the user id and password	Prompt error	Prompt error	Pass
3	User enter user id and password	Valid user id and password which doesn't exist in Database	Registered Successfully	Login Page	Pass
4	User enter user id and password	Invalid user id and password which contains in Database	Prompt error	Prompt error	Pass

Table 6.2: Test Case - Others

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User tries to buy coins with less tokens	Less Token amount	Prompt Message showing error	Successful Updated	Fail
2	User enters valid wallet	Valid Wallet Details	Connected Successfully	Connected	Pass

6 Limitations

- It needs internet to be accessed.
- Due to unavailability (company started charging for the node service) of services user cannot trade/buy coins.
- Only top 10 coins can be displayed.
- Only few Crypto Coins can be purchased/trade/exchange.

7 Future Enhancements

- Automatic report generation of all the trade carried out.
- Feature of Buying/Selling/Trading of Crypto Coins
- Automatic buying of coin at a certain price.
- Feature to stop trading at particular point after a coin reaches to a particular price.

8 Prerequisite

What is MetaMask?
MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications.

9 User Manual

Part 1 – Register/Login
In order to access the site a user does not have to register or login. They can just surf the site and can keep a watch on Coins and their prices.
But if a user wants to trade/buy/exchange they need to register their account via their MetaMask wallet.

Part 2 – Login
User needs to connect thier metamask wallet in order to buy coins.

Part 3 – Coin Details
User can see the coin details (price, name) all the data is updated in real time.

Part 4 – Swap
User can swap their coins with other coins via swap modal.

10 Bibliography

10.1 Web References

[1.] https://docs.metamask.io/guide/
[2.] https://docs.moralis.io/reference
[3.] https://docs.soliditylang.org/en/v0.8.17/
[4.] https://www.youtube.com/watch?v=M576WGiDBdQ
[5.] https://www.youtube.com/watch?v=mT5siI19gtc
[6.] https://stackoverflow.com/
[7.] https://www.draw.io/
[8.] https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/
[9.] https://www.geeksforgeeks.org/