S. K. Somaiya College of Arts, Science, and Commerce

Project Title

Crypto Price Tracker

By

Nisarg Gandhi

Roll No . 07

as partial fulfillment for the degree of

Bachelor of Science (Computer Science)

Under the guidance of

Prof. Poonam Pandey

ACKNOWLEDGEMENT

At the onset, I would like to express deep intense thanks and my sincere gratitude to all the lecturers of **S. K. SOMAIYA OF ARTS, SCIENCE AND COMMERCE, Vidyavihar** for giving me a valuable opportunity to work on this project.

I am extremely grateful and remain indebted to my project guide **Miss. Poonam Pandey mam** for being a source of my inspiration and constant support in my work on the project. She has been a constant source of inspiration and motivation for my hard work. Her constructive advice & constant motivation have been responsible for the successful completion of the project. I came to know lots of things during this project

TABLE OF CONTENTS-

INTRODUCTION

A **cryptocurrency**, **crypto-currency**, or **crypto** is a collection of binary data which is designed to work as a medium of exchange where in individual coin ownership records are stored in a ledger which is a computerized database using strong cryptography to secure transaction records, to control the creation of additional coins, and to verify the transfer of coin ownership.

Some crypto schemes use validators to maintain the cryptocurrency. In a proof-of-stake model, owners put up their tokens as collateral. In return, they get authority over the token in proportion to the amount they stake. Generally, these token stakers get additional ownership in the token over time via network fees, newly minted tokens or other such reward mechanisms

A cryptocurrency price tracker monitors the current value of different digital currencies and tokens. Many of these services provide historical data as well, allowing users to compare current prices against older values. Some of these services even have a tool that allows users to contrast the performance of multiple cryptocurrencies against one another.

While it may seem obvious, the accuracy of the price tracker that an investor decides to use will have a significant impact on their decisions, as well as on the timing and success of their investments. A tracker that is updated consistently and that uses reliable data in calculating its values is obviously crucial.

ABSTRACT

Most of the people in the world like to invest in crypto whether it is large amount or small amount. The need for crypto price tracker is they can check cryptocurrency rate with ease and can find out which currency is best to buy. Crypto price tracker website is a dynamic website which takes data from API and gives out current rate. There is also search bar to search out users favourite cryptocurrency.

Users can log in through google account , if user don’t have account they can create new account and log in with that account. Every time user log in with a unique account , unique id will be generated and will be given to user.

User can also contact through live support or filling contact form

FUTURE SCOPE

Some of the limitations that cryptocurrencies presently face – such as the fact that one’s digital fortune can be erased by a computer crash, or that a virtual vault may be ransacked by a hacker – may be overcome in time through technological advances. What will be harder to surmount is the basic paradox that bedevils cryptocurrencies – the more popular they become, the more regulation and government scrutiny they are likely to attract, which erodes the fundamental premise for their existence.

1. Communicate with potential customers 24/7: It increases engagement level with the customer and develop more sales opportunities without giving much effort.

2. Online performance management: Performance management is a very critical process in business management, which must be done accurately.

3. Easy payment and customer support: The success of any business depends highly on its customer service. The live customer support service help most user to buy cryptocurrency

REQUIREMENT SPECIFICATIONS:-

- Hardware Requirements

Processor	1.6 Ghz or faster processor
Ram	1 Gb of Ram
Windows	7 or higher

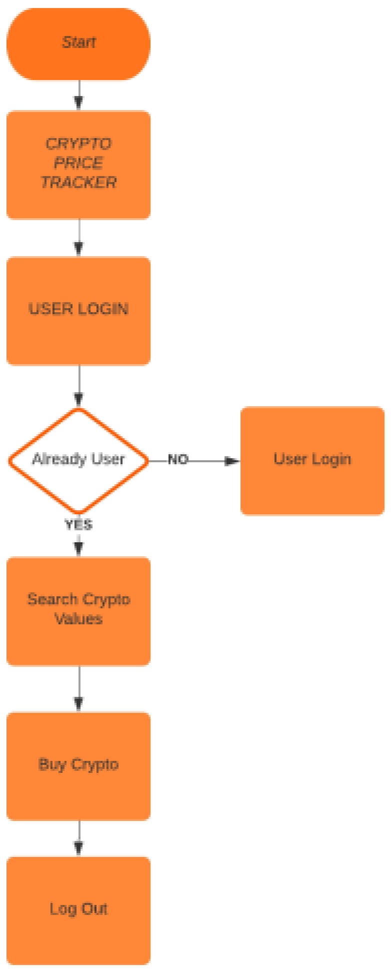
- Software Requirements

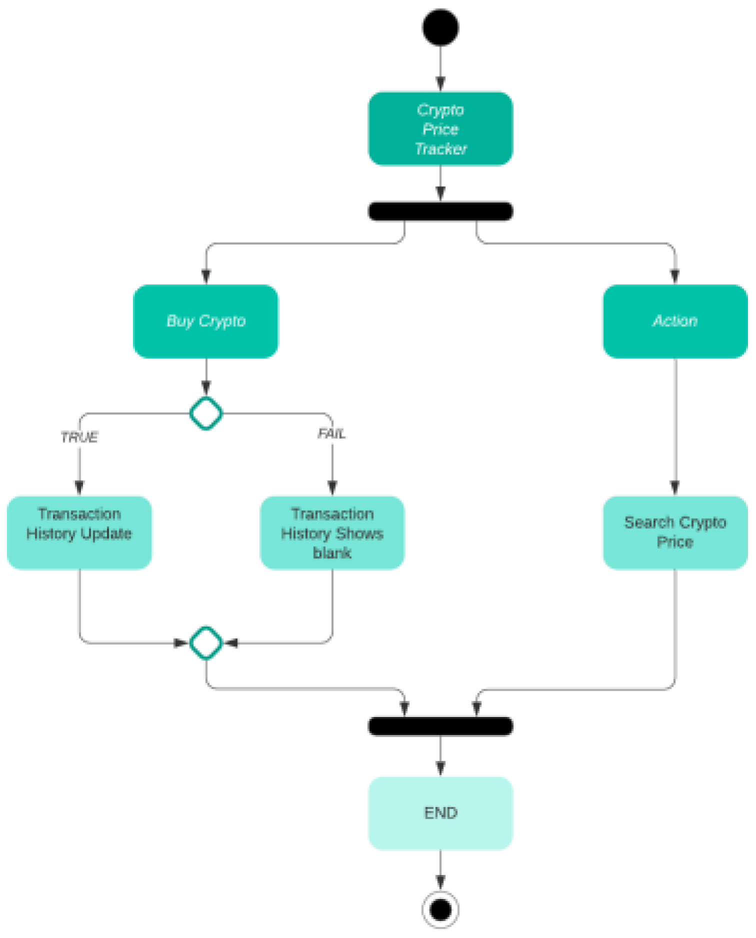
1. **Vs Code** - It is a **streamlined code editor with support for development operations like debugging, task running, and version control.** It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

2. **Firebase** - It is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides **tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.**

3. **NPM** - it is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.

SYSTEM DESIGN DETAILS





Crypto Price Tracker
Buy Crypto
Action
TRUE
FAIL
Transaction History Update
Transaction History Shows blank
Search Crypto Price
END

DATABASE DESIGN

Admin table :

SN NO	Field Name	Data Type	Description	Constraints
1	email	varchar	email address	not null
2	password	varchar	Security password	not null

CODE

App.css

```
.Nav__head,
.Nav__Link {
  color: white !important;
}

.User__name {
  margin-right: 20px;
}
```

App.js

```
import React, { useState, useEffect } from "react";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";
import { makeStyles } from "@material-ui/core/styles";

import Typography from "@material-ui/core/Typography";

import { Navbar, Container, Button, Nav } from "react-bootstrap";
import "./App.css";

import firebase from "firebase/app";
import "firebase/auth";

import Tracker from "./Tracker";
import Buynow from "./Buynow";
import Contactus from "./Contactus";

const useStyles = makeStyles((theme) => ({
  root: {
    flexGrow: 1,
  },
  title: {
    flexGrow: 1,
  },
}));

export default function App() {
```

```
const classes = useStyles();
var provider = new firebase.auth.GoogleAuthProvider();

const [user, setUser] = useState(null);

useEffect(() => {
  // FIR Auth State Observer
  firebase.auth().onAuthStateChanged((user) => {
    if (user) {
      var uid = user.uid;
      console.log(`User has signed in with UID: ${uid}`);
      setUser(uid);
    } else {
      // User is signed out
      console.log("User is not signed in.");
      setUser(null);
    }
  });
}, []);

function signInWithGooglePopUp() {
  firebase
    .auth()
    .signInWithPopup(provider)
    .then((result) => {
      console.log("User has signed in.");
    })
    .catch((error) => {
      // Handle Errors here.
```

```
        var errorCode = error.code;
        var errorMessage = error.message;
        // The email of the user's account used.
        var email = error.email;
        // The firebase.auth.AuthCredential type that was used.
        var credential = error.credential;
        console.log(
          `Errors occurred during sign in: ${errorCode}, ${errorMessage},
${email}, ${credential}`
        );
      });
  }

  return (
    <div className={classes.root}>
      <Router>
        <Navbar>
          <Container>
            <Navbar.Brand className="Nav__head">
              <Nav.Link as={Link} to={"/"} className="Nav__Link">
                Crypto Price Tracker
              </Nav.Link>
            </Navbar.Brand>
            <Nav.Link as={Link} to={"/buynow"} className="Nav__Link">
              Buy Crypto Now
            </Nav.Link>
            <Nav.Link as={Link} to={"/contactus"} className="Nav__Link">
              Contact Us
            </Nav.Link>
```

```
<Navbar.Toggle />
<Navbar.Collapse className="justify-content-end">
  <Navbar.Text className="User__name">
    {user ? (
      <div
        style={{
          display: "flex",
          alignItems: "center",
        }}
      >
        <Typography variant="body1">{user}</Typography>
        <Button
          onClick={() => {
            // signout
            firebase
              .auth()
              .signOut()
              .then(() => {
                console.log("User has signed out.");
              })
              .catch((err) => {
                console.log("Error signing out: ", err);
              });
          }}
        >
          Signout
        </Button>
      </div>
    ) : (
          <Button
            onClick={() => {
              signInWithGooglePopUp();
            }}
          >
            Login
          </Button>
        )}
      </Navbar.Text>
    </Navbar.Collapse>
  </Container>
</Navbar>
<Switch>
<Route path="/contactus">
    <Contactus />
  </Route>
  <Route path="/buynow">
    <Buynow />
  </Route>
  <Route path="/">
    <Tracker />
  </Route>
</Switch>
</Router>
</div>
);
}
```

App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Buyform.js

```
/* eslint-disable react-hooks/exhaustive-deps */
import React, { useEffect, useState, useCallback } from "react";
import InputBase from "./InputBase";

function BuyForm({ data, onPurchase }) {
  const { name, rate } = data;
  const INIT = { amount: 0, converted: 0 };
  const [exchange, setExchange] = useState(INIT);
  const [transactions, setTransactions] = useState([]);

  useEffect(() => {
    setExchange({
      ...exchange,
      converted: Number(exchange.amount / rate).toFixed(4),
    });
  }, [name]);

  useEffect(() => {
    onPurchase(transactions);
  }, [transactions]);

  const generateID = (prefix) =>
    Math.random()
      .toString(36)
      .replace("0.", prefix || "");

  const handleChange = ({ target: { value, name } }) => {
```

```
  const val = Number(value.trim());
  const converted = (val / rate).toFixed(4);
  console.log(converted);

  setExchange({
    [name]: val,
    converted,
  });
};

const makePurchase = useCallback(
  (e) => {
    e.preventDefault();

    if (!exchange.amount) {
      alert("please enter amount");
    }

    const payload = {
      ...exchange,
      name,
      id: generateID("transX-id_"),
    };

    setTransactions([...transactions, payload]);
  },
  [exchange, transactions]
);
```

```
  console.log("transX", transactions);

  return (
    <div>
      <form onSubmit={makePurchase} className="form">
        <div className="input-group mb-3">
          <InputBase name="amount" textLabel="INR"
onChange={handleChange} />
          <i className="fas fa-exchange-alt" />
          <InputBase value={exchange.converted} disabled
textLabel={name} />
        </div>
        <input className="btn btn-primary" type="submit"
value="Purchase" />
      </form>
    </div>
  );
}

export default BuyForm;
```

Buynow.css

```
.buynow {
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(5px + 2vmin);
}

.cryptocoin-icon {
  max-width: 50px;
  width: 100%;
}

.card {
  text-align: center;
  width: 170px;
  cursor: pointer;
  margin: 0 10px;
}

.card.selected {
  border: 3px solid #41cc16;
}

.form {
  max-width: 500px;
  margin: 50px 20px;
}

.form .fa-exchange-alt {
  line-height: unset;
  padding: 0 10px;
}

.TransHist {
  margin-left: 10px;
}
```

Buynow.js

```
import React, { useState, useEffect } from "react";
import "./Buynow.css";
import Cryptotile from "./Cryptotile";
import BuyForm from "./BuyForm";
import Transactions from "./Transactions";

import btc from "./Assets/btc.png";
import eth from "./Assets/eth.png";
import doge from "./Assets/doge.png";
import axios from "axios";

function Buynow() {

  const [list, setList] = useState([]);
  const [tiles, setTiles] = useState([
    { id: 1, icon: btc, name: "BTC", rate: 0 },
    { id: 2, icon: eth, name: "ETH", rate: 0 },
    { id: 3, icon: doge, name: "DOGE", rate: 0 },
  ])
  const [selectedTile, setSelectedState] = useState(tiles[0]);


  useEffect(() => {
    const getData =  async() => {
      try {
        const res = await
    axios.get('https://api.coingecko.com/api/v3/coins/markets?vs_currency=i
    nr&order=market_cap_desc&per_page=10&page=1&sparkline=false')
        const newTiles = [...tiles]
        let index = 0
        if(res.data.length > 1){
          for(const c of res.data){
            if(c.symbol.toLowerCase() ===
    tiles[index].name.toLowerCase()){
              newTiles[index].rate = c.current_price
              index++
            }
            if(index > tiles.length){
              break
            }
          }
          setTiles(newTiles)
        }


      } catch(err) {
        console.log(err)
      }
    }
    getData()
    // eslint-disable-next-line
  }, [])



  const handleSelect = (data) => {
    setSelectedState(data);
```

```
};

const buildList = (list) => {
  setList(list);
};

return (
  <div className="buynow">
    <div className="container">
      <div className="row">
        <div className="col-sm">
          <div className="d-flex">
            {
              tiles.length > 0 ?
              tiles.map((cryptocoin) =>
                <Cryptotile
                  key={cryptocoin.id}
                  data={cryptocoin}
                  onClick={handleSelect}
                  selected={cryptocoin.id === selectedTile.id}
                />
              )
              : null
            }
          </div>
          <BuyForm data={selectedTile} onPurchase={buildList} />
        </div>
        <div className="col-sm TransHist">
          <Transactions list={list} />
          </div>
        </div>
      </div>
    </div>
  );
}

export default Buynow;
```

Coin.css

```
.coin-container {
  display: flex;
  justify-content: center;
}

.coin-row {
  display: flex;
  flex-direction: row;
  /* justify-content: start; */
  align-items: center;
  height: 80px;
  border-bottom: 1px solid #d7d7d7;
  width: 900px;
}

.coin {
  display: flex;
  align-items: center;
  padding-right: 24px;

  min-width: 300px;
}

.coin h1 {
  font-size: 16px;
  width: 150px;
}
```

```css
.coin img {
  height: 30px;
  width: 30px;
  margin-right: 10px;
}

.coin-symbol {
  text-transform: uppercase;
}

.coin-data {
  display: flex;
  text-align: right;
  justify-content: space-between;
  width: 100%;
}

.coin-price {
  width: 110px;
}

.coin-volume {
  width: 155px;
}

.coin-marketcap {
  width: 200px;
}
```

```
.coin-percent {
  width: 100px;
}

.red {
  color: #f00606;
}

.green {
  color: #11d811;
}
```

Coin.js

```
import React from "react";
import "./Coin.css";

const Coin = ({
  name,
  image,
  symbol,
  price,
  volume,
  priceChange,
  marketcap,
}) => {
  return (
    <div className="coin-container">
      <div className="coin-row">
        <div className="coin">
          <img src={image} alt="crypto" />
          <h1>{name}</h1>
          <p className="coin-symbol">{symbol}</p>
        </div>
        <div className="coin-data">
          <p className="coin-price">₹{price}</p>
          <p className="coin-volume">₹{volume.toLocaleString()}</p>

          {priceChange < 0 ? (
            <p className="coin-percent
red">{priceChange.toFixed(2)}%</p>
```

```
          ) : (
            <p className="coin-percent
green">{priceChange.toFixed(2)}%</p>
          )}

          <p className="coin-marketcap">
            Mkt Cap: ${marketcap.toLocaleString()}
          </p>
        </div>
      </div>
    </div>
  );
};

export default Coin;
```

Contactus.css

```
@import
url("https://fonts.googleapis.com/css2?family=Inter:wght@100;200;300;4
00;500;600;700;800;900&display=swap");

* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}

body {
   background-color: rgb(248, 246, 253);
}

body,
input,
textarea,
button {
   font-family: "Inter", sans-serif;
}

.app {
   width: 400px;
   margin: 0 auto;
   height: 100vh;
}
```

```css
/* Contact.js */
.contactus-form {
    height: 100vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
}

.contactus-form>h1 {
    margin-bottom: 30px;
}

.contactus-form>input,
textarea {
    padding: 20px;
    border-radius: 3px;
    /* box-shadow: 0px 0px 5px rgba(0, 0, 0, 0.137); */
    margin-bottom: 20px;
    border: 1px solid lightgray;
    /* border: none; */
    background: #fff;
    font-size: 16px;
    color: rgb(0, 0, 32);
    outline: none;
}

.contactus-form>input:focus,
textarea:focus {
    border: 1px solid rgb(0, 0, 196);
```

```
}

.contactus-form>textarea {
    height: 150px;
    max-width: 400px;
    min-height: 100px;
}

.contactus-form>label {
    padding-bottom: 10px;
    color: white !important;
    font-weight: bold;
}

.contactus-form>button {
    padding: 20px;
    border: none;
    background-color: #0B5ED7;
    font-weight: 500;
    font-size: 20px;
    border-radius: 3px;
    color: #fff;
    cursor: pointer;
    transition: 0.2s ease-in-out;
    margin-top: 10px;
}

.contactus-form>button:hover {
    background-color: rgb(0, 0, 196);
```

```
}
```

Contact.js

```
import React from 'react';
import "./Contactus.css";
import emailjs from "emailjs-com"

function Contactus() {

    function sendEmail(e) {
        e.preventDefault();

        emailjs.sendForm("service_vhu6kxf",
        "template_48n98j4",
        e.target,
        "user_woAkJuiVc2znHLKbrOt1b"
        ).then(res=>{
            console.log(res);
        }).catch(err=> console.log(err));
        e.target.reset()

    }

    return (
        <div className="contactus app" onSubmit={sendEmail}>
            <form action="" className="contactus-form">
            <h1>Contact form 📞</h1>

            <label >Name</label>
            <input type="text" name="name" />
            <label >Email</label>
            <input type="email" name="user_email" />

            <label >Message</label>
            <textarea name="message" rows="4"></textarea>

            <button type="submit">Submit</button>
            </form>
        </div>
    )
}

export default Contactus
```

Cryptotile.js

```
import React from "react";

const Cryptotile = ({ data, selected, onClick }) => {
  const { name, rate, icon } = data;

  const handleClick = () => onClick(data);

  return (
    <div>
      <div className={`card ${selected && "selected"}`}
onClick={handleClick}>
        <div className="card-body">
          <img className="cryptocoin-icon" src={icon} alt="icon" />
          <div>{name}</div>
          <div> ₹ {rate}</div>
        </div>
      </div>
    </div>
  );
};

export default Cryptotile;
```

InputBase.js

```
import React from "react";

const InputBase = ({ textLabel, ...props }) => {
  return (
    <>
      <input type="number" className="form-control" {...props} />
      <span className="input-group-text">{textLabel}</span>
    </>
  );
};

export default InputBase;
```

Tracker.css

```
@import
url("https://fonts.googleapis.com/css2?family=Montserrat&display=swap"
);

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
  font-family: "Montserrat", sans-serif;
  background-color: #1a1a1c;
  color: #fff;
}

.coin-app {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 64px;
  color: #fff;
}

.coin-search {
  margin-bottom: 64px;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-content: center;
}

.coin-text {
  margin-bottom: 32px;
  text-align: center;
}

.coin-input {
  padding-left: 16px;
  width: 300px;
  height: 50px;
  border-radius: 4px;
  border: none;
  background-image: linear-gradient(
    -225deg,
    #ac32e4 0%,
    #7918f2 48%,
    #4801ff 100%
  );

  color: #e2e2e2;
}

.coin-input::placeholder {
  color: #e2e2e2;
}
```

};

Tracker.js

```
import React, { useState, useEffect } from "react";
import "./Tracker.css";
import axios from "axios";
import Coin from "./Coin";

function Tracker() {
  const [coins, setCoins] = useState([]);
  const [search, setSearch] = useState("");

  useEffect(() => {
    axios
      .get(
"https://api.coingecko.com/api/v3/coins/markets?vs_currency=inr&order
=market_cap_desc&per_page=30&page=1&sparkline=false"
      )
      .then((res) => {
        setCoins(res.data);
        console.log(res.data);
      })
      .catch((error) => console.log(error));
  }, []);

  const handleChange = (e) => {
    setSearch(e.target.value);
  };

  const filteredCoins = coins.filter((coin) =>
    coin.name.toLowerCase().includes(search.toLowerCase())
  );

  return (
    <div className="tracker">
      <div className="coin-app">
        <div className="coin-search">
          <h1 className="coin-text">Search a currency</h1>
          <form>
            <input
              type="text"
              placeholder="Search"
              className="coin-input"
              onChange={handleChange}
            />
          </form>
        </div>

        {filteredCoins.map((coin) => {
          return (
            <Coin
              key={coin.id}
              name={coin.name}
              price={coin.current_price}
              symbol={coin.symbol}
              marketcap={coin.total_volume}
              volume={coin.market_cap}
              image={coin.image}
              priceChange={coin.price_change_percentage_24h}
            />
          );
        })}
      </div>
    </div>
   );
  }

  export default Tracker;
```

Transaction.js

```
import React from "react";

const Transactions = ({ list }) => {
  return (
    <div className="list-container">
      <h5>Transaction</h5>
      <ul className="list-group">
        {list.length ? (
          list.map((item) => (
            <li key={item.id} className="list-group-item">
              <span>
                <strong style={{backgroundColor: '#fff', color:
'#000'}}>{item.name}</strong>
              </span>
              {": "}
              <span style={{backgroundColor: '#fff', color:
'#000'}}>{item.converted}</span>
            </li>
          ))
        ) : (
          <div> No Purchases</div>
        )}
      </ul>
    </div>
  );
};
```

```
export default Transactions;
```

Index.css

```
* {
  margin: 0;
}
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto",
"Oxygen",
    "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",
    monospace;
}
```

Index.js

```
import React from "react";
import ReactDOM from "react-dom";

import App from "./App";

import "bootstrap/dist/css/bootstrap.min.css";

// MUI
import "@fontsource/roboto";

// Firebase
import firebase from "firebase/app";

const firebaseConfig = {
  apiKey: "AIzaSyDMbZ-AcCZEgFMHe6h5XeXwJRkkvEZHn6w",
  authDomain: "crypto-price-tracker-285f7.firebaseapp.com",
  projectId: "crypto-price-tracker-285f7",
  storageBucket: "crypto-price-tracker-285f7.appspot.com",
  messagingSenderId: "615469203618",
  appId: "1:615469203618:web:fc1d5c0ed62dfac23f57fb",
};

if (!firebase.apps[0]) {
  firebase.initializeApp(firebaseConfig);
}

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);
```

reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB
}) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;
```

setupTests.js

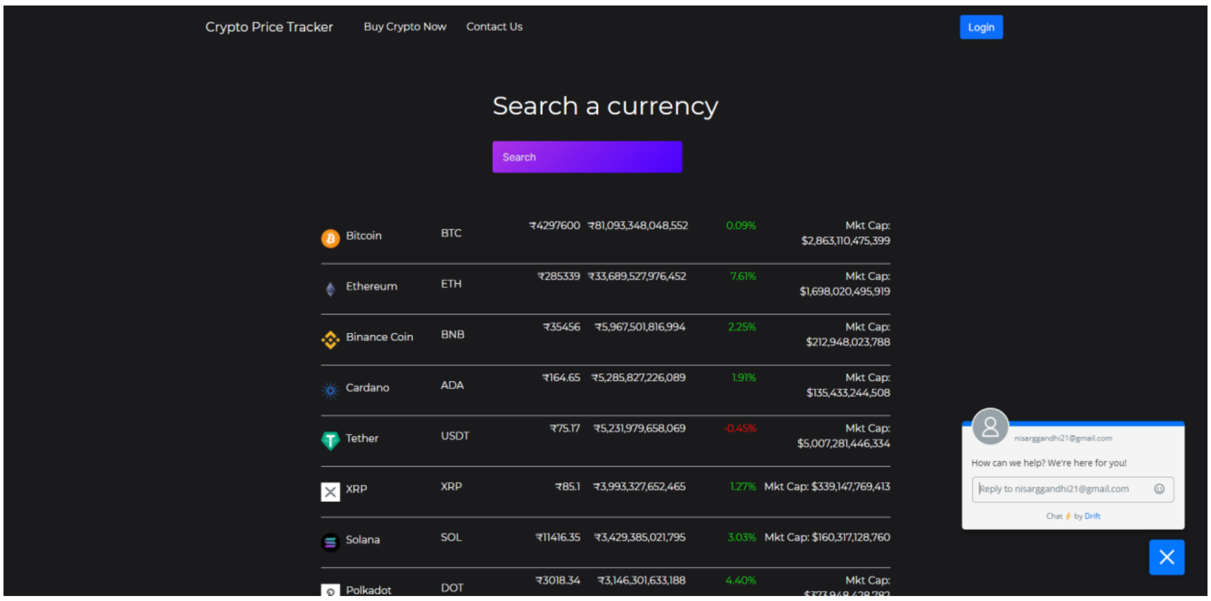
```
// jest-dom adds custom jest matchers for asserting on DOM nodes.
// allows you to do things like:
// expect(element).toHaveTextContent(/react/i)
// learn more: https://github.com/testing-library/jest-dom
import '@testing-library/jest-dom';
```

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
  <meta name="viewport" content="width=device-width, initial-scale=1"
/>
  <meta name="theme-color" content="#000000" />
  <meta name="description" content="Web site created using create-
react-app" />
  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
  <!--
      manifest.json provides metadata used when your web app is
installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta2/css/all.min.css"
    integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qpHQAEuvilg9FAn5VJUDwKZZxkJ
NuGM4XkWuk94WCrrwslk8yWNGmY1EduTA=="
    crossorigin="anonymous" referrerpolicy="no-referrer" />
  <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the
HTML.


      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico"
will

      work correctly both with client-side routing and a non-root public
URL.

      Learn how to configure a non-root public URL by running `npm run
build`.

    -->

  <title>React App</title>

  <!-- Start of Async Drift Code -->

  <script>

    "use strict";


    ! function () {

      var t = window.driftt = window.drift = window.driftt || [];

      if (!t.init) {

        if (t.invoked) return void(window.console && console.error &&
console.error("Drift snippet included twice."));

        t.invoked = !0, t.methods = ["identify", "config", "track", "reset",
"debug", "show", "ping", "page", "hide",

            "off", "on"

          ],

          t.factory = function (e) {

            return function () {

              var n = Array.prototype.slice.call(arguments);

              return n.unshift(e), t.push(n), t;

            };

          }, t.methods.forEach(function (e) {

            t[e] = t.factory(e);
          }), t.load = function (t) {
            var e = 3e5,
              n = Math.ceil(new Date() / e) * e,
              o = document.createElement("script");
            o.type = "text/javascript", o.async = !0, o.crossorigin =
"anonymous", o.src =
              "https://js.driftt.com/include/" + n + "/" + t + ".js";
            var i = document.getElementsByTagName("script")[0];
            i.parentNode.insertBefore(o, i);
          };
      }
    }();
    drift.SNIPPET_VERSION = '0.3.1';
    drift.load('rhtg357e8riv');
  </script>
  <!-- End of Async Drift Code -->
</head>

<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
</body>

</html>
```

WEBSITE SNAPSHOT

HOME PAGE



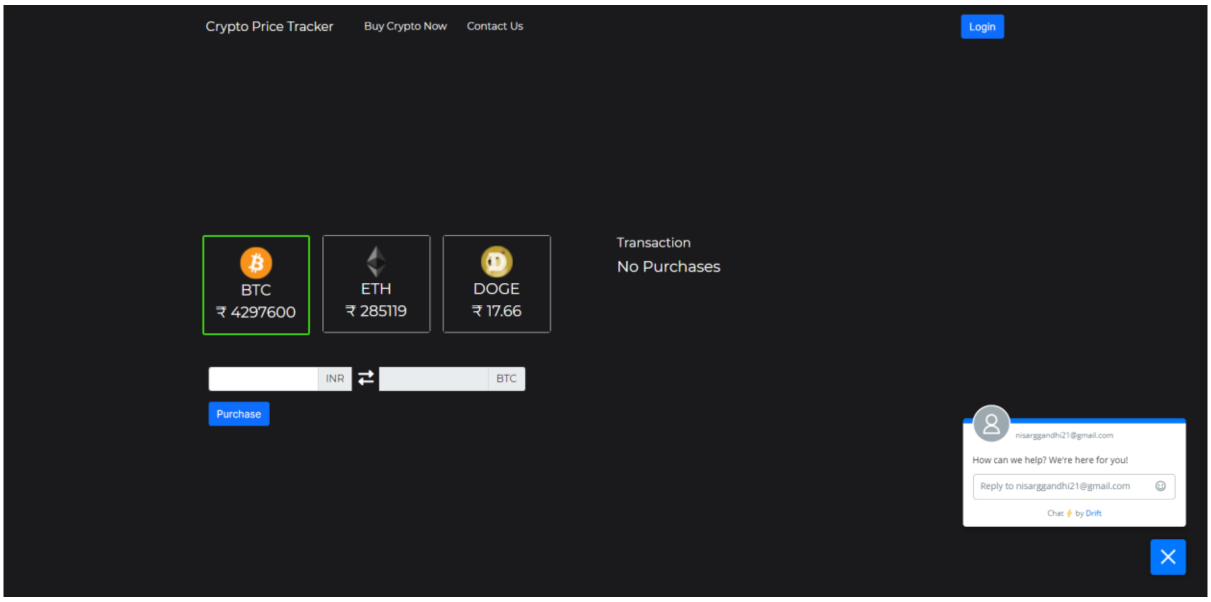
Search a currency

Bitcoin	BTC	₹4297600	₹81,093,348,048,552	0.09%	Mkt Cap: $2,863,110,475,399
Ethereum	ETH	₹285339	₹33,689,527,976,452	7.61%	Mkt Cap: $1,698,020,495,919
Binance Coin	BNB	₹35456	₹5,967,501,816,994	2.25%	Mkt Cap: $212,948,023,788
Cardano	ADA	₹164.65	₹5,285,827,226,089	1.91%	Mkt Cap: $135,433,244,508
Tether	USDT	₹75.17	₹5,231,979,658,069	-0.45%	Mkt Cap: $5,007,281,446,334
XRP	XRP	₹85.1	₹3,993,327,652,465	1.27%	Mkt Cap: $339,147,769,413
Solana	SOL	₹11416.35	₹3,429,385,021,795	3.03%	Mkt Cap: $160,317,128,760
Polkadot	DOT	₹3018.34	₹3,146,301,633,188	4.40%	Mkt Cap: $373,948,428,782

How can we help? We're here for you!

Reply to nisarggandhi21@gmail.com

BUY PAGE



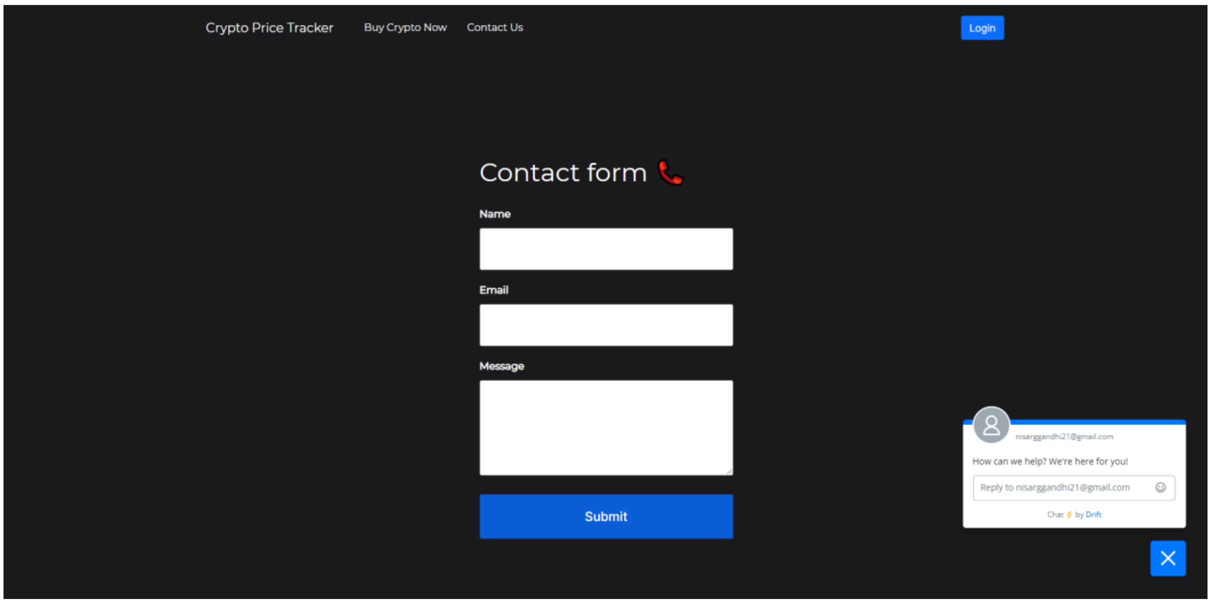
Transaction

No Purchases

How can we help? We're here for you!

Reply to nisarggandhi21@gmail.com

CONTACT US



Contact form

Name

Email

Message

Submit

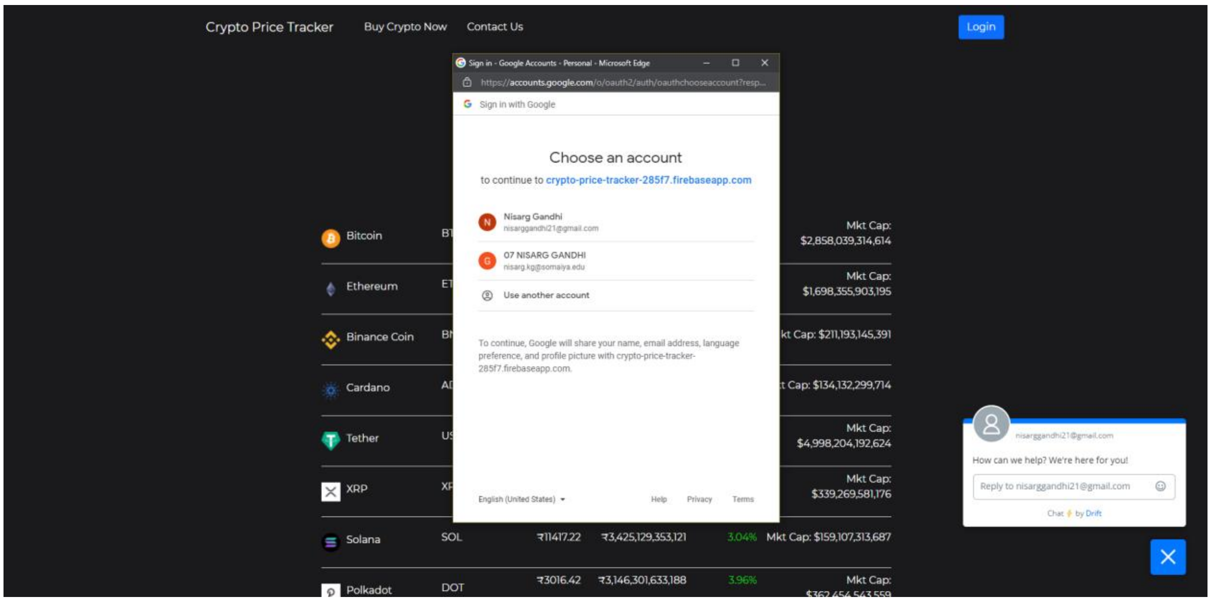
nisarggandhi21@gmail.com

How can we help? We're here for you!

Reply to nisarggandhi21@gmail.com

Chat ⚡ by Drift

SIGN IN PAGE



Sign in - Google Accounts - Personal - Microsoft Edge

https://accounts.google.com/o/oauth2/auth/oauthchooseaccount?resp...

Sign in with Google

Choose an account

to continue to crypto-price-tracker-285f7.firebaseapp.com

Nisarg Gandhi
nisarggandhi21@gmail.com

07 NISARG GANDHI
nisarg.kg@somaiya.edu

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with crypto-price-tracker-285f7.firebaseapp.com.

English (United States) ▾ Help Privacy Terms

Bitcoin	BT				Mkt Cap: $2,858,039,314,614
Ethereum	ET				Mkt Cap: $1,698,355,903,195
Binance Coin	BN				kt Cap: $211,193,145,391
Cardano	AD				t Cap: $134,132,299,714
Tether	US				Mkt Cap: $4,998,204,192,624
XRP	XR				Mkt Cap: $339,269,581,176
Solana	SOL	₹11417.22	₹3,425,129,353,121	3.04%	Mkt Cap: $159,107,313,687
Polkadot	DOT	₹3016.42	₹3,146,301,633,188	3.96%	Mkt Cap: $362,454,543,559

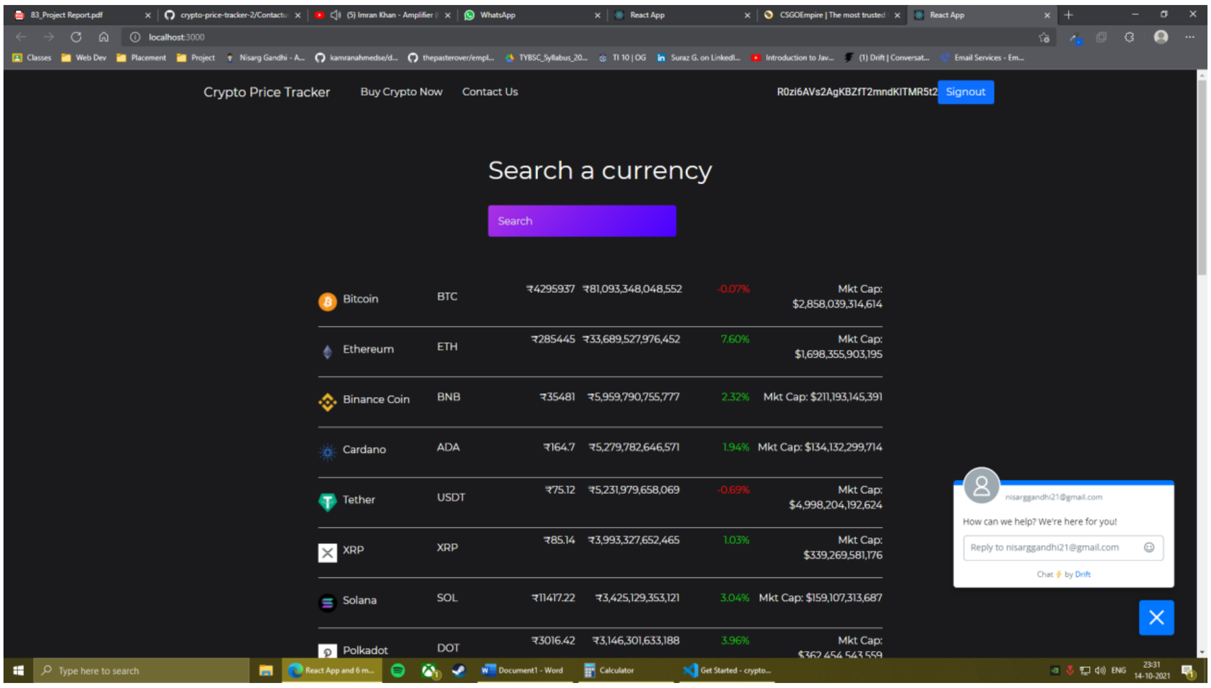
nisarggandhi21@gmail.com

How can we help? We're here for you!

Reply to nisarggandhi21@gmail.com

Chat ⚡ by Drift

AFTER SIGN IN PAGE



Crypto Price Tracker   Buy Crypto Now   Contact Us   R0zi6AVs2AgKBZfT2mndKITMR5t2 Signout

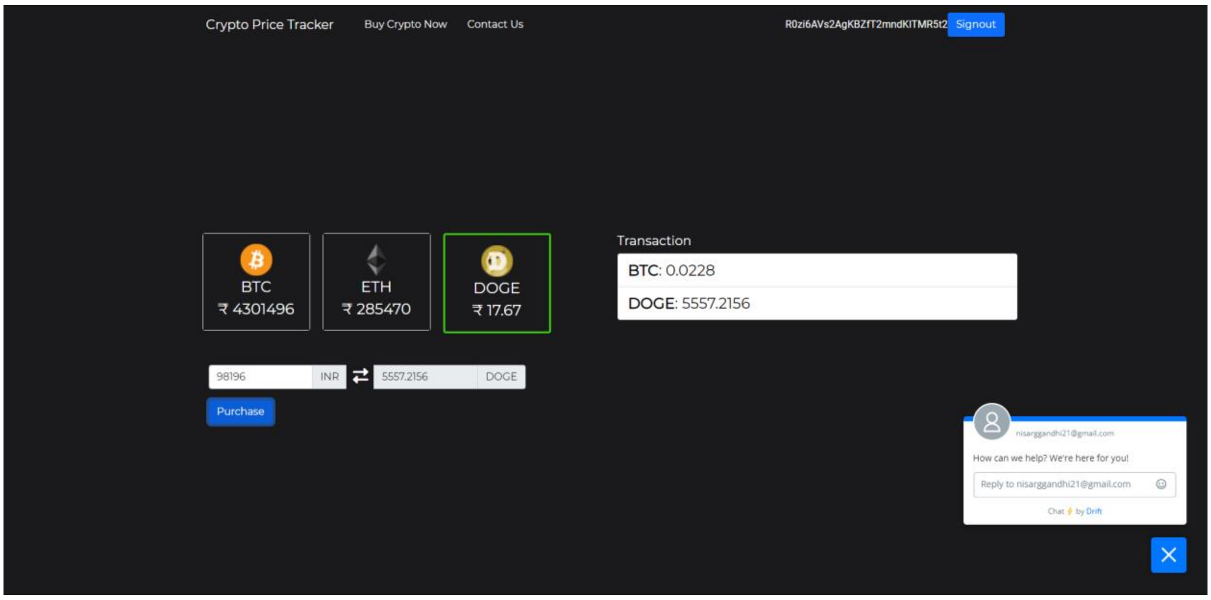
Search a currency

Search

Bitcoin	BTC	₹4295937	₹81,093,348,048,552	-0.07%	Mkt Cap: $2,858,039,314,614
Ethereum	ETH	₹285445	₹33,689,527,976,452	7.60%	Mkt Cap: $1,698,355,903,195
Binance Coin	BNB	₹35481	₹5,959,790,755,777	2.32%	Mkt Cap: $211,193,145,391
Cardano	ADA	₹164.7	₹5,279,782,646,571	1.94%	Mkt Cap: $134,132,299,714
Tether	USDT	₹75.12	₹5,231,979,658,069	-0.69%	Mkt Cap: $4,998,204,192,624
XRP	XRP	₹85.14	₹3,993,327,652,465	1.03%	Mkt Cap: $339,269,581,176
Solana	SOL	₹11417.22	₹3,425,129,353,121	3.04%	Mkt Cap: $159,107,313,687
Polkadot	DOT	₹3016.42	₹3,146,301,633,188	3.96%	Mkt Cap: $362,454,543,559

nisarggandhi21@gmail.com
How can we help? We're here for you!
Reply to nisarggandhi21@gmail.com
Chat ⚡ by Drift

AFTER TRANSACTIONS



Crypto Price Tracker   Buy Crypto Now   Contact Us   R0zi6AVs2AgKBZfT2mndKITMR5t2 Signout

BTC ₹ 4301496   ETH ₹ 285470   DOGE ₹ 17.67

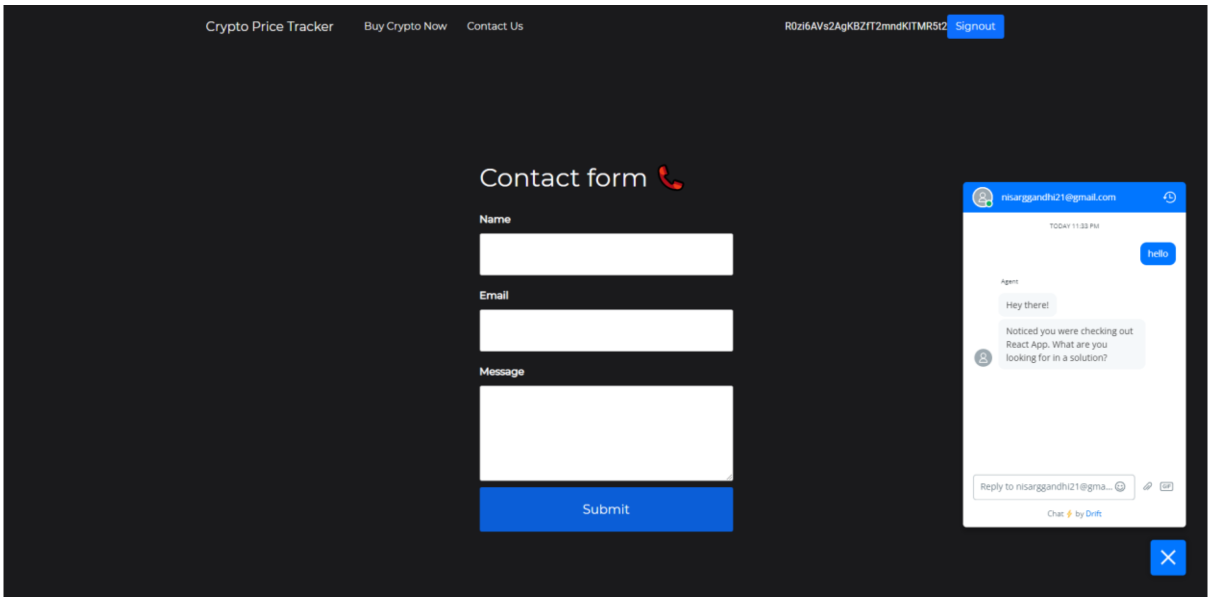
Transaction
BTC: 0.0228
DOGE: 5557.2156

98196 INR ⇄ 5557.2156 DOGE

Purchase

nisarggandhi21@gmail.com
How can we help? We're here for you!
Reply to nisarggandhi21@gmail.com
Chat ⚡ by Drift

LIVE SUPPORT



Contact form

Name

Email

Message

nisarggandhi21@gmail.com

TODAY 11:33 PM

hello

Agent

Hey there!

Noticed you were checking out React App. What are you looking for in a solution?

Reply to nisarggandhi21@gma...

CONCLUSION

In this project I tried to implement the crypto price tracker. This project is built on cryptocurrency as everything is centralized we can combine many objects in order to perform effective analysis. Users can buy cryptocurrency at the current rate and also contact using contact form or live chat.

BIBLOGRAPHY

- https://www.youtube.com/
- https://www.firebase.google.com/
- https://www.coingecko.com/
- https://www.reactjs.org/
- https://www.app.drift.com/
- https://www.emailjs.com/