CHISITO
Spin off - Università di Padova

Digital Currency Dashboard

12-06-2022

Sai Charan
Full Stack Developer.

Overview

To develop a dashboard to track the different types of digital currencies by using **PostgreSQL** for database, **Node server** for backend, **React** for front end.

Goals

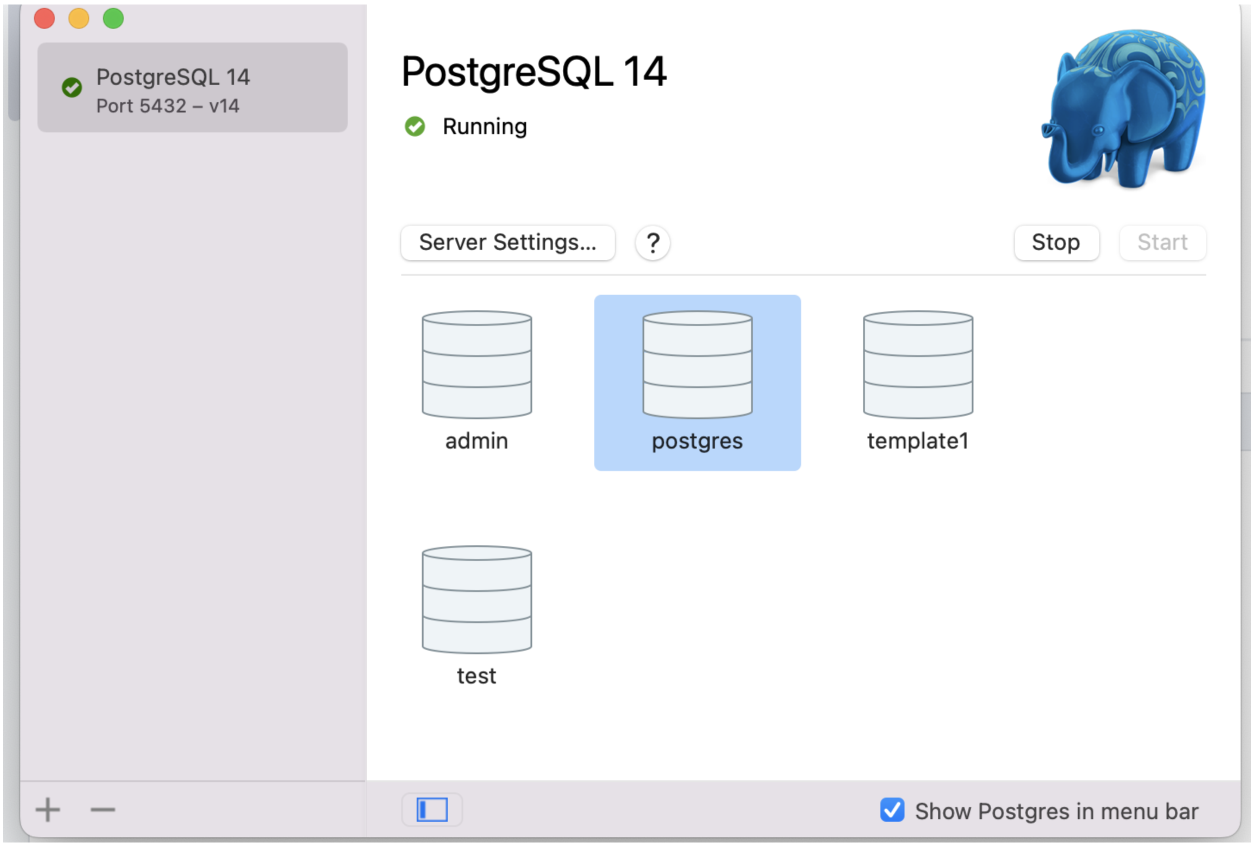
1. A simple login/register. User information: name, surname, email, password. Create Tables using PostgreSQL
2. Once the user has been logged in, the user can select a digital/cryptocurrency and see some visualizations.

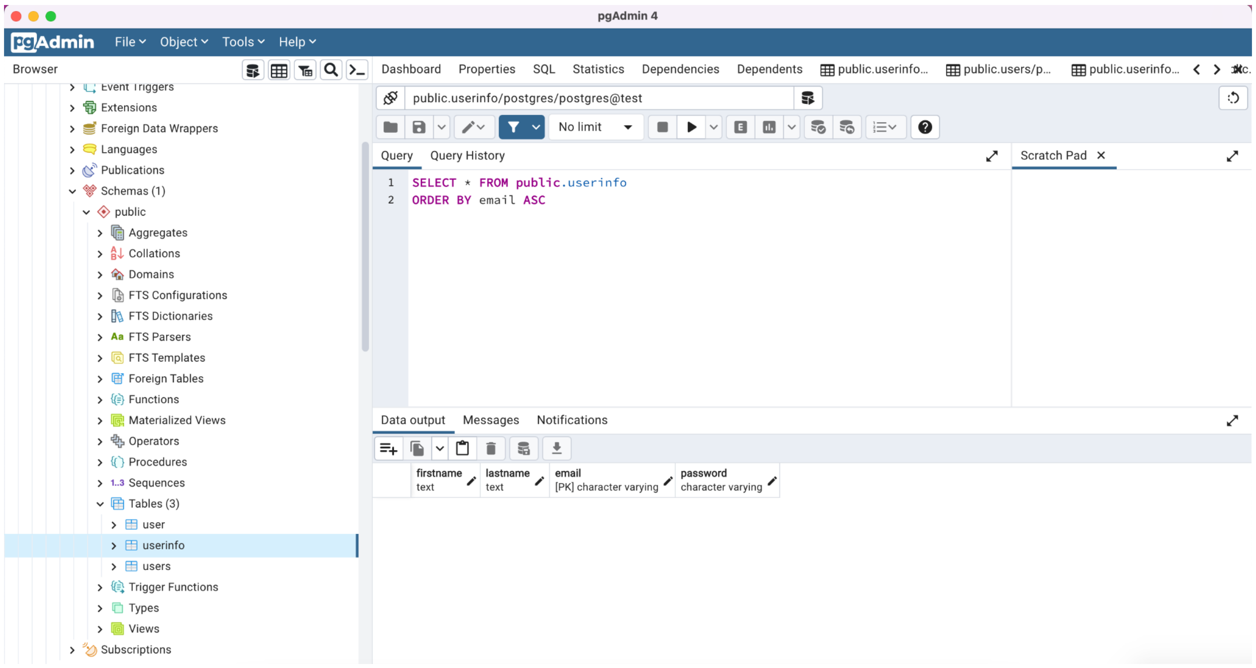
3. Then, for the currency selected by the user, the web app has to show:
    a. The current price (EUR), the variation w.r.t the previous day, and the low and high value of prices of the day.
    b. A plot of the trend for the last period (from May 1st) of the close price (EUR).
    c. A visualization (donut plot for example) that shows the percentage of how many times the volume has exceeded the average volume in the month.
4.

Development

Database : PostgreSQL

1) Installed PostgreSQL and created a table named userinfo with help of pgadmin4(tool used to access and make changes in postgres database)
2) Added the columns for the table userinfo.
    a) First Name datatype: text. Mandatory
    b) Last Name datatype: text. Mandatory
    c) Email   datatype: varying character and It will the **PRIMARY KEY** mandatory
    d) Password   datatype: varying character mandatory.
3) Start the database to be available for node servcer.



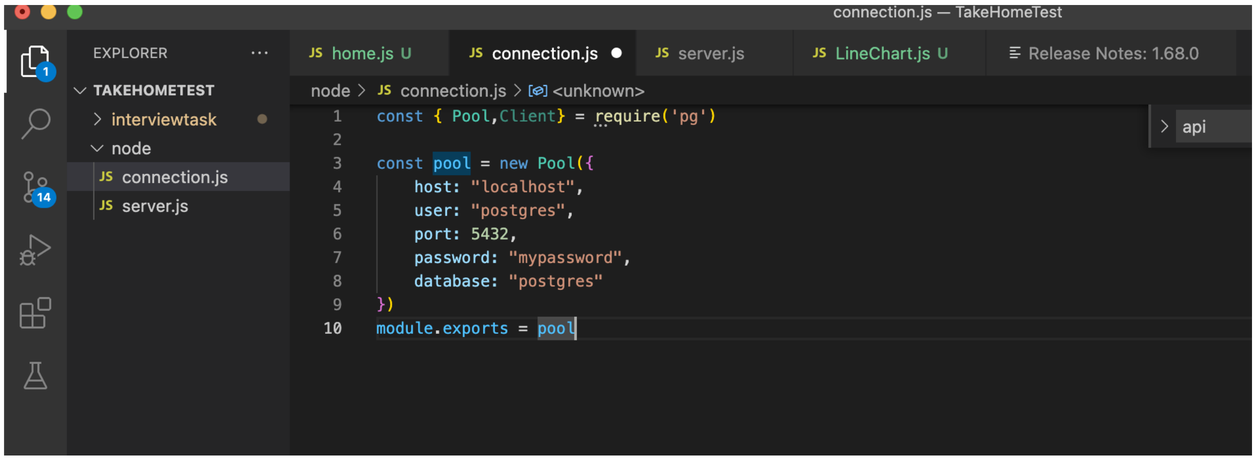


```
SELECT * FROM public.userinfo
ORDER BY email ASC
```

firstname text	lastname text	email [PK] character varying	password character varying

Node Server SetUp:

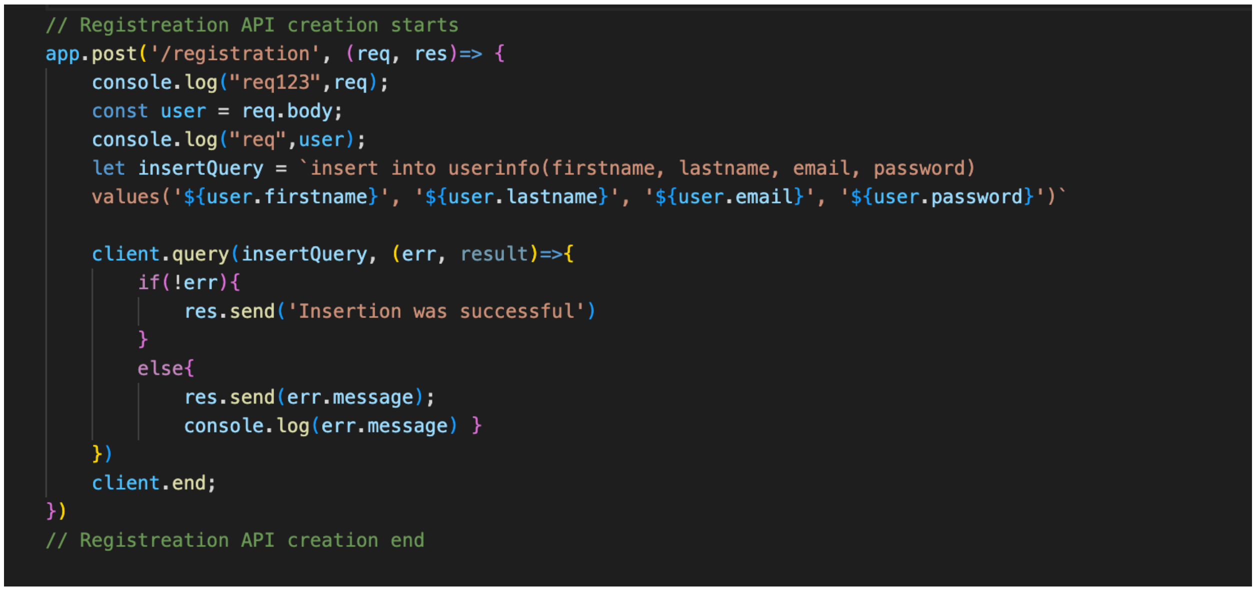
1) Installation of latest node using nvm install node and **express** js using npm.
2) Created folder with task name moved it to **visual studio code.**
3) Installed postgres npm package to folder using `npm install pg`
4) Created connection to database postgres using Pool by giving database details in connection.js file in visual studio code.



```
const { Pool,Client} = require('pg')

const pool = new Pool({
    host: "localhost",
    user: "postgres",
    port: 5432,
    password: "mypassword",
    database: "postgres"
})
module.exports = pool
```

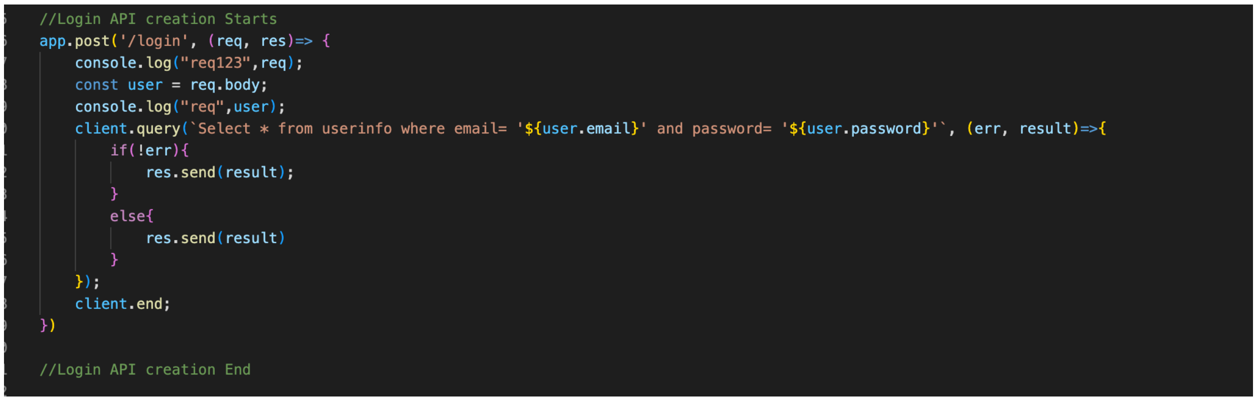
5) Using express established a connection between node and postgres in **server.js** file.
6) Created the **POST registration API** call to insert user information into postgres database using an insertion query.



```
// Registreation API creation starts
app.post('/registration', (req, res)=> {
    console.log("req123",req);
    const user = req.body;
    console.log("req",user);
    let insertQuery = `insert into userinfo(firstname, lastname, email, password)
    values('${user.firstname}', '${user.lastname}', '${user.email}', '${user.password}')`

    client.query(insertQuery, (err, result)=>{
        if(!err){
            res.send('Insertion was successful')
        }
        else{
            res.send(err.message);
            console.log(err.message) }
    })
    client.end;
})
// Registreation API creation end
```

7) Created another **POST login API** call to verify email and password exist in database or not and using a select query.



```
//Login API creation Starts
app.post('/login', (req, res)=> {
    console.log("req123",req);
    const user = req.body;
    console.log("req",user);
    client.query(`Select * from userinfo where email= '${user.email}' and password= '${user.password}'`, (err, result)=>{
        if(!err){
            res.send(result);
        }
        else{
            res.send(result)
        }
    });
    client.end;
})

//Login API creation End
```

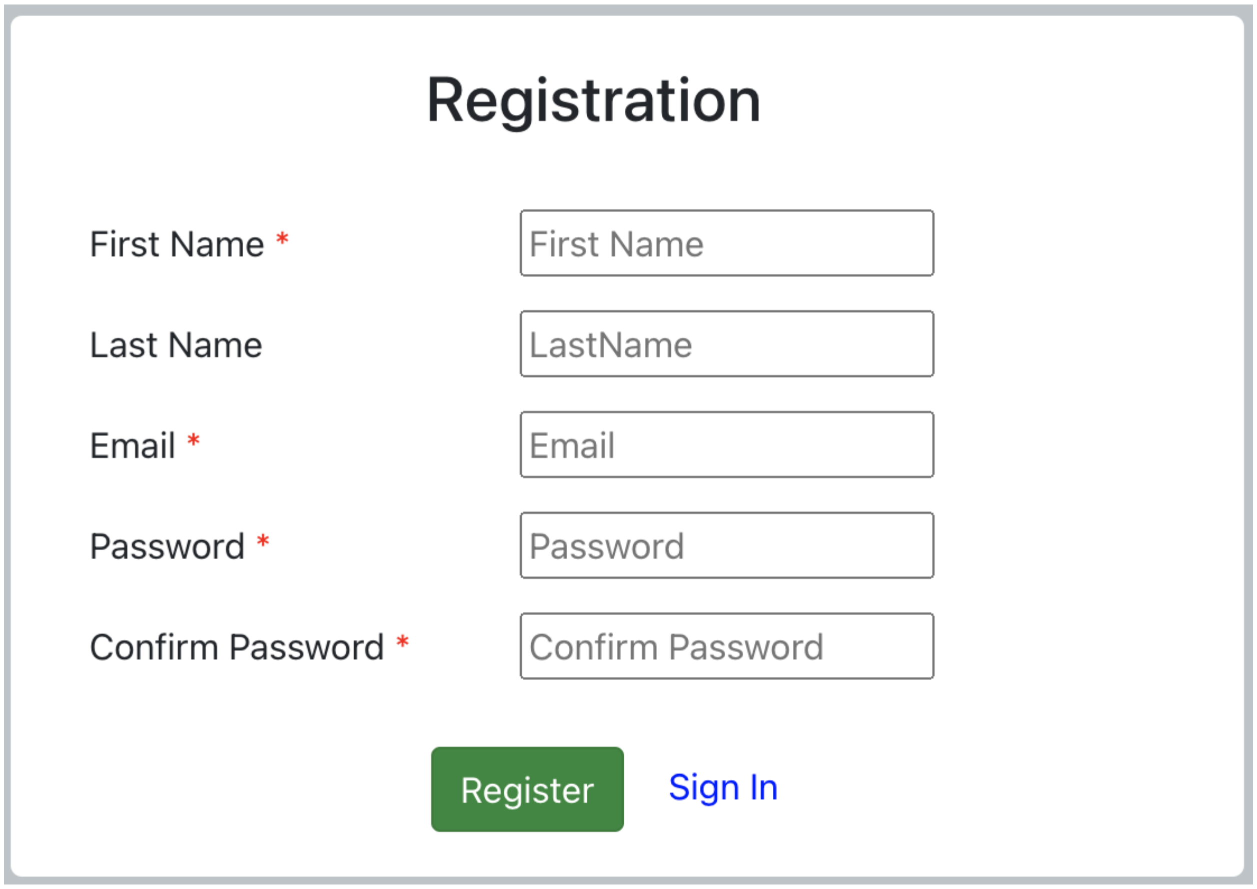
8) Node server will be run once API set is done by executing command **node servicer.js in** terminal, now **localhost:4200** server will run in system.

FrondEnd Setup

1) Installed the react in the folder by using `npm` **`install --save react react-dom`**
2) Created the project name interviewtask, using `npx create-react-app` **interviewtask**

Registration or Sign Up page:

1) Started developing the Registration page UI and functionality.
2) All the field validation has been done. Password and confirm password and checking.
3) onClick of Register button consuming node localhost registration api to save data into database.

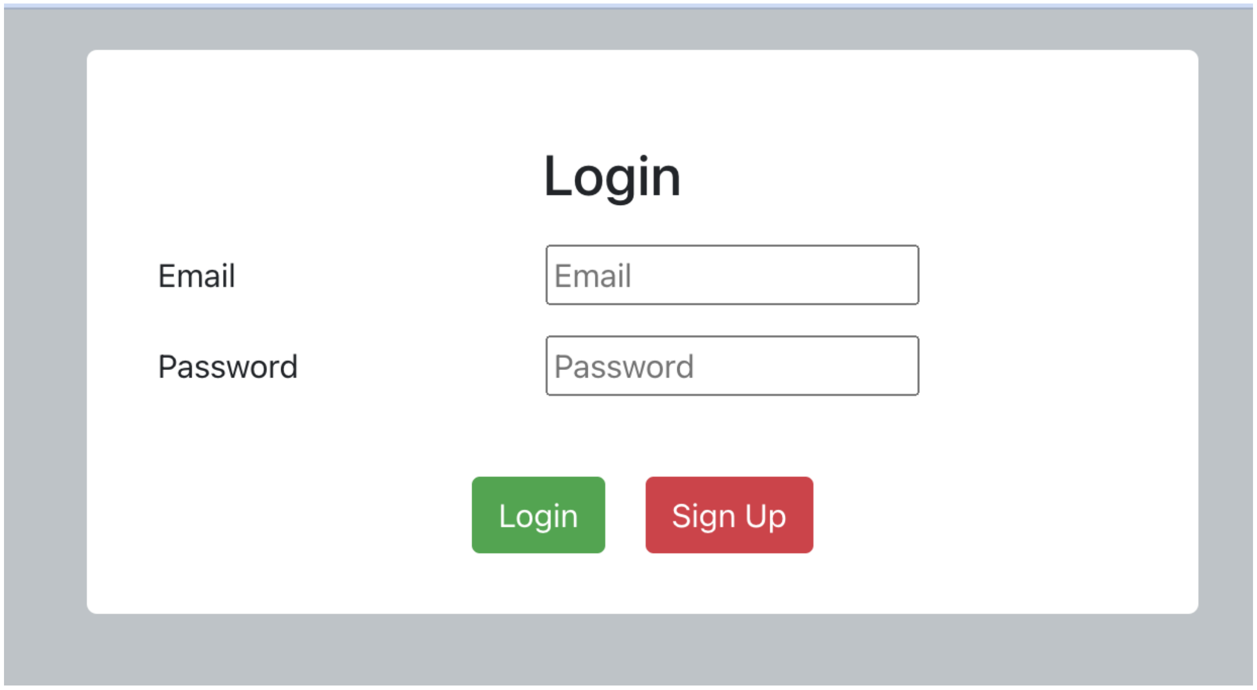


Registration

First Name *	First Name
Last Name	LastName
Email *	Email
Password *	Password
Confirm Password *	Confirm Password

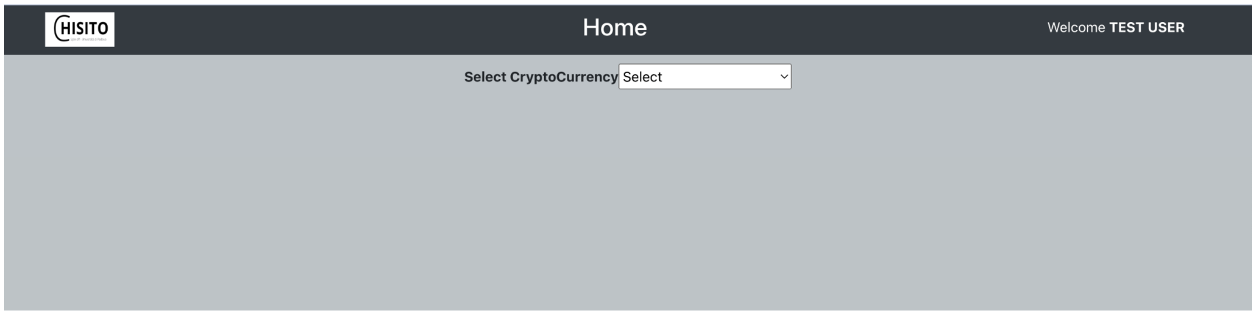
Login or Sign In page:

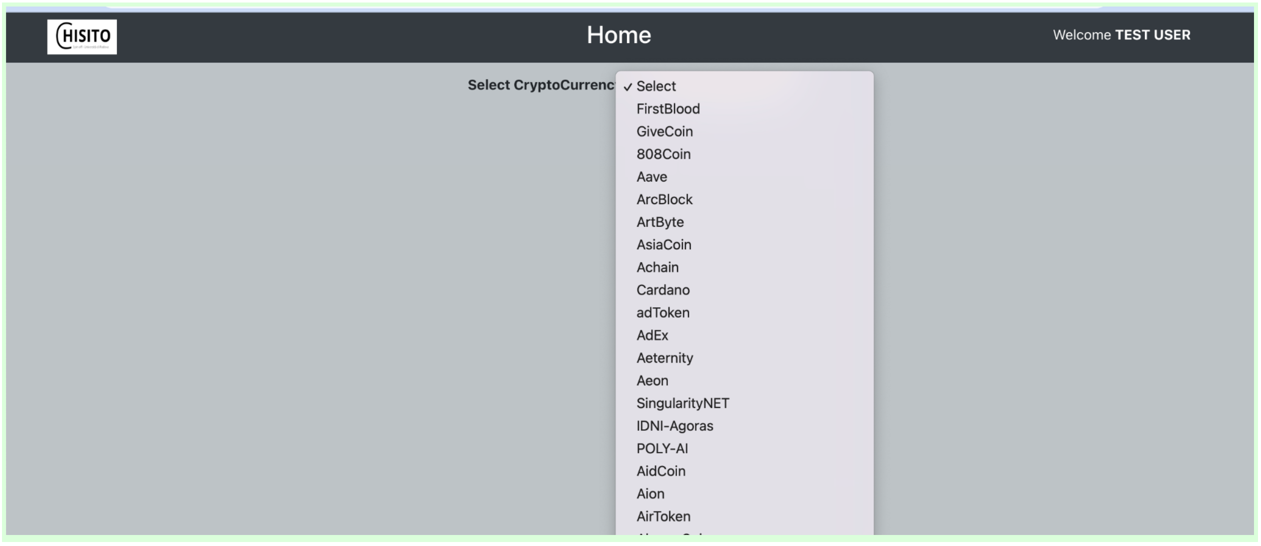
1) User has to enter a valid email and password to login.
2) onClick of Login button consuming **node localhost login api** to check whether email and password exist in the database or not.



Home Page:

1) Login API Post request will give the first name, last name, email, password. Using these details display the user name in the home page.
2) Home page will have select drop down to select the list of digital currencies.
    a) This list of digital currency will load from from the csv file download from https://www.alphavantage.co/digital_currency_list/
    b) Loaded this local csv file data into object by using npm d3 package, command to install package `npm i d3.`
    c) This digital currency data list loaded into the dropdown in the home page.

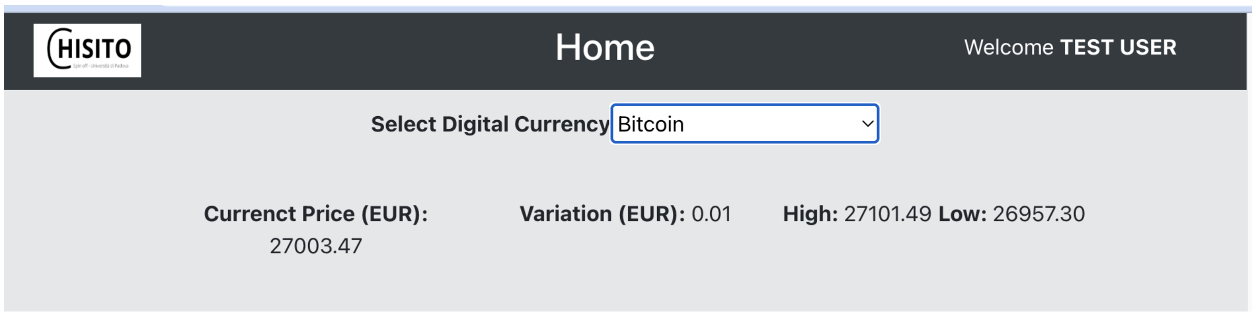




3) Once user selected the any digital currency from list, we are calling external API ***https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_DAILY&symbol=BTC&market=EUR&apikey=A4A5UWNYFSY5YRZM.***
4) Received all the data trends related to selected currency and selected market.

Displaying the Current price, Variation, High and Low:

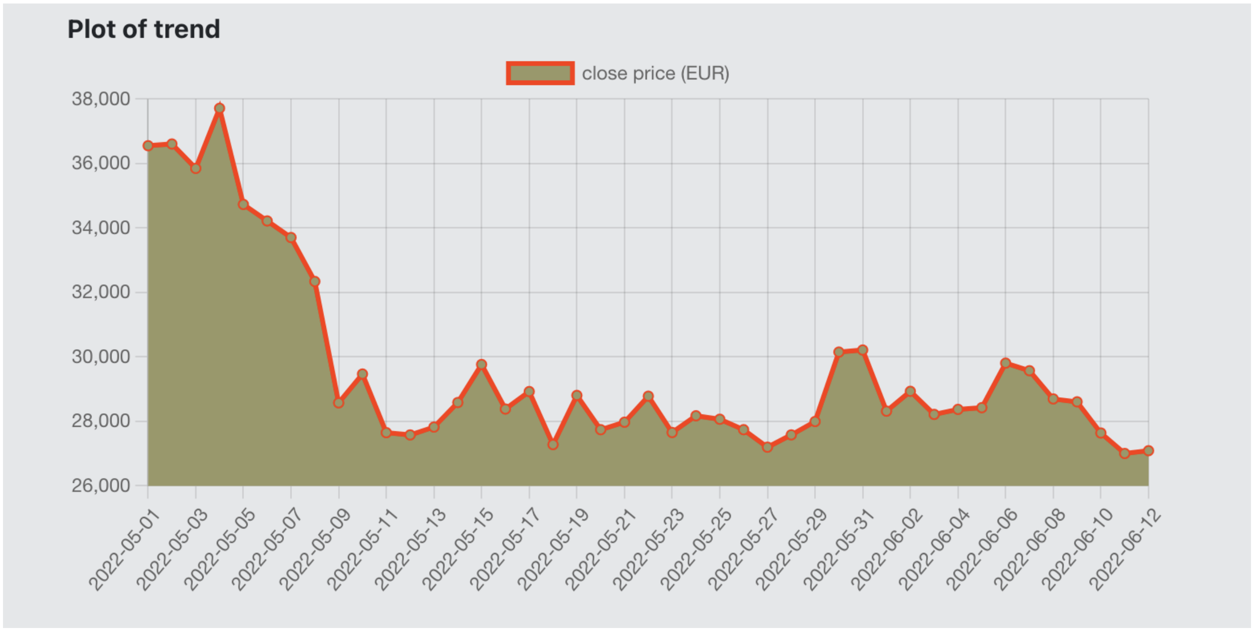
1) Created a separate component named **CurrentDayStatistics** to display all the details mentioned. Shared the data received in the external API using props.
2) Accessed the data shared using props and filter the current day details from the object. And displayed the data.
3) **Current Price** is the opening price of the day of EUR market,
4) **Variation** is diaplayed calculated this as the difference between current day Open and previous day close price.
5) **High**: Current day high price.
6) **Low**: Current day low price.



Currenct Price (EUR): 27003.47 **Variation (EUR):** 0.01 **High:** 27101.49 **Low:** 26957.30

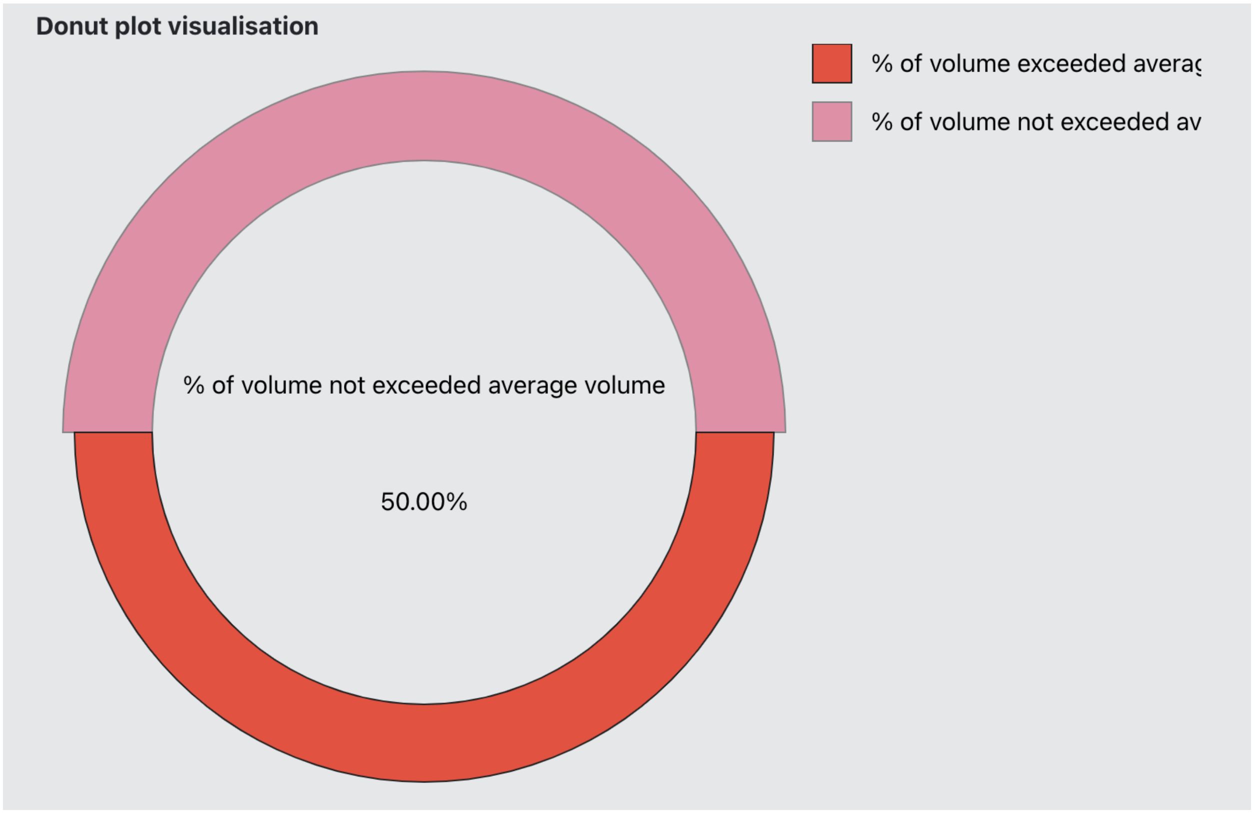
Displaying the Plot of Trend using Line chart:

1) Created a new component named LIneChart, to display the plot of trend from 1st May2022 to Current Day.
2) Filtered the data and created two Arrays, **Dates** (in between the dates **2022-05-01 to Current Date**.) and **DatesData** (closing prices of the respective dates).
3) Installed react chart2 npm package to display the line chart using `npm i react-chartjs-2. And imported into project.`
4) Send the Arrays as data input to Line Chart.



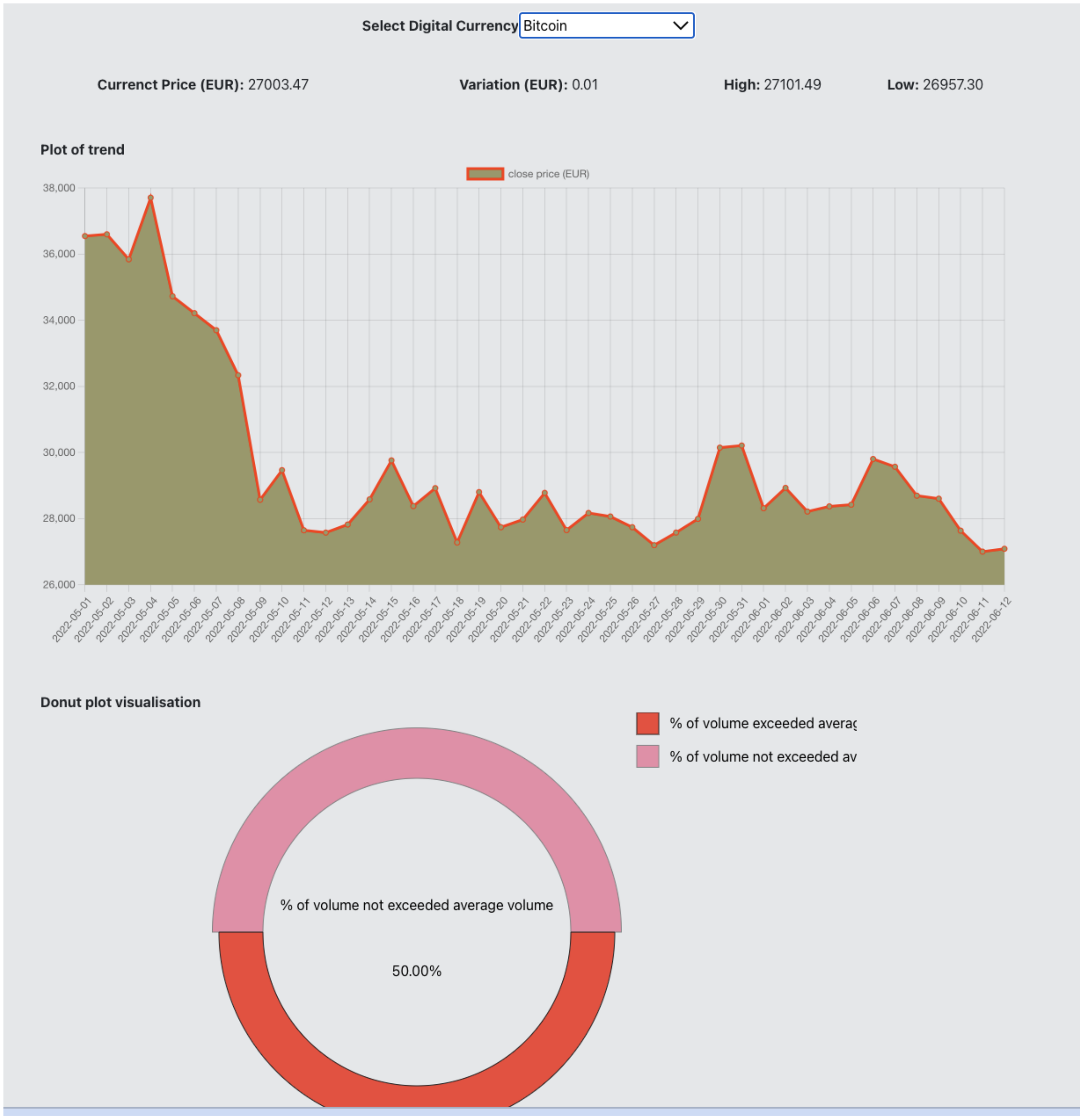
Donut Plot Visualization:

1) Created a new component named DonutPlot to show the percentage of how many times the volume has exceeded the average volume in the month.
2) Filtered the data and created two Arrays, **Dates** (in between the dates **Start of Month to Current Date**.) and **DatesData** (closing prices of the respective dates).
3) Installed `DONUT` npm package to display the Donut chart using `npm i react-donut-chart` and imported into project.
4) Send the Arrays as data input to Donut Chart.


Donut plot visualisation
% of volume exceeded averag
% of volume not exceeded av
% of volume not exceeded average volume
50.00%

Result:

Developed the Digital Currency dashboard.



Select Digital Currency Bitcoin

Currenct Price (EUR): 27003.47 **Variation (EUR):** 0.01 **High:** 27101.49 **Low:** 26957.30

Plot of trend

close price (EUR)
38,000
36,000
34,000
32,000
30,000
28,000
26,000
2022-05-01
2022-06-12

Donut plot visualisation

% of volume exceeded averaç
% of volume not exceeded av
% of volume not exceeded average volume
50.00%