FEVERTOKENS Internship Program

Technical Test

Welcome!

Thank you for your interest in our internship program. This test is designed to assess your coding skills and problem-solving abilities. Please read the following instructions carefully to ensure a smooth testing experience.

Test Overview:

This test consists of coding tasks that are relevant to the role of an engineering intern. You'll be required to write code to solve problems and demonstrate your coding proficiency.

Submission Instructions:

Create a **private** Github repository named "**fevertokens-tech-test**", where you will submit your work by adding to your Github repository to the following emails:

- lem@fevertokens.io
- toub@fevertokens.io
- maghzaz@fevertokens.io

You will have to send a notification email once your work is done. Please provide your full name and the title of the position you applied for in the subject line of the email.

GitHub Repository and Documentation:

In the **Private** GitHub repository for the test, you should follow these guidelines:

- The repository must be private and should host **task 1**.

- Include a README.md explaining how to install, build, and run the code using **pnpm** (not **npm**) as a package manager. It should contain package.json, .gitignore, .env.example, etc.
- The README.md file should also contain the codes for **task 2** in **code blocks** with the output under the code block and the answer to **task 3** with an illustration describing your proposed approach.
- Adhere to best practices for version control and code organization.
- Use proper documentation in your repository.

Evaluation Criteria:

You will be assessed on the following criteria:

- Code quality
- Problem-solving skills
- Efficiency
- Creativity

Resources:

You are allowed to use online resources, programming documentation, and any libraries or frameworks you find necessary to complete the tasks.

Honor Code:

- This is an individual assessment. Do not collaborate or share your code with others.
- Plagiarism or unauthorized assistance is strictly prohibited and will result in **automatic rejection** of your application.

Support Contact:

If you encounter any technical issues regarding the test or have questions about the tasks, please contact lem@fevertokens.io.

Feedback and Next Steps:

After you submit your test, our team will review it. If you pass this test, you will be invited for an interview. You can expect feedback within 3 working days after the submission.

Accessibility:

If you have specific accessibility needs or require accommodations, please let us know in advance, and we will do our best to accommodate them.

Test Questions:

The tasks are provided in the subsequent sections of this document.

By proceeding with this test, you acknowledge that you have read and understood these guidelines.

Task 1: Coins Catalog

Introduction

This kata is designed to assess your proficiency in **React Js** and **Next Js** by having you build a compact application.

Your main task is to develop an app showcasing cryptocurrency coins, with challenges intensifying as you proceed and optional bonus questions toward the end.

NB1: While our primary focus isn't on the web design itself (CSS), any extra effort put toward that regard will certainly be valued!

NB2: Please ensure that your project's launch process is clear!

Context

- ***What is a coin?***

In the context of cryptocurrency and blockchain technology, a "**coin**" typically refers to a digital or virtual representation of value that can be used as a medium of exchange. These coins are often decentralized and secured by cryptography, making them resistant to counterfeiting and providing a high degree of security. Here are some key points about coins in the cryptocurrency world:

- ❖ Digital Representation of Value: Cryptocurrencies are purely digital, meaning they exist only in electronic form. They don't have a physical counterpart like paper money or metal coins.

- ❖ Decentralization: Many cryptocurrencies are decentralized, meaning they are not controlled by any central authority, such as a government or a central bank. Instead, they rely on a distributed ledger technology called blockchain to record transactions and manage the issuance of new coins.

- ❖ Cryptography: Cryptocurrencies use cryptographic techniques to secure transactions and control the creation of new coins. This ensures the integrity and security of the network.

- ❖ Medium of Exchange: Cryptocurrencies are designed to be used as a medium of exchange for goods and services. They can be used for online purchases, remittances, investments, and more.

- ❖ Store of Value: Some cryptocurrencies, like Bitcoin, are considered digital stores of value, similar to gold. People may hold these cryptocurrencies as a form of investment or to preserve wealth.

- ❖ Unit of Account: Cryptocurrencies can be used as a unit of account to measure the value of goods and services. Prices are often denominated in cryptocurrencies in some online markets.

- ❖ Different Types: There are thousands of cryptocurrencies in existence, each with its own unique features and use cases. Bitcoin, Ethereum, and Ripple (XRP) are examples of well-known cryptocurrencies, but there are many others.

- ❖ Initial Coin Offerings (ICOs): Some blockchain projects raise funds by issuing their own coins or tokens through a process called an Initial Coin Offering (ICO). Investors purchase these coins with the expectation that their value will increase over time.

- ❖ Utility Tokens vs. Security Tokens: Cryptocurrencies can be categorized as utility tokens, which provide access to a specific service or product, or security tokens, which represent ownership in an underlying asset or project.

It's important to note that the cryptocurrency landscape is constantly evolving, and new coins and tokens are regularly created. The use cases, technology, and regulatory environment surrounding cryptocurrencies can vary widely, so it's essential to research and understand the specific characteristics and purposes of any cryptocurrency you may be interested in, The structure of a currency is:

- **id**
- **Attributes :**
  - **symbol (This is a shorthand symbol representing a coin, commonly used in cryptocurrency trading pairs and exchanges).**
  - **name (This is a URL pointing to an image or icon that represents a coin. It's often used in user interfaces and websites)**
  - **image (Type of coins [national, digital, virtual, metal, energy])**
  - **current_price (This attribute represents the current price of one coin in a specific currency (ex: USD), which appears to be in the currency of the data source.)**
  - **market_cap (Market capitalization is the total value of all outstanding Bitcoins in circulation. It's calculated by multiplying the current price by the circulating supply.)**
  - **market_cap_rank (This indicates that Bitcoin is currently the top-ranked cryptocurrency by market capitalization.)**
  - **fully_diluted_valuation (This represents the hypothetical market capitalization if all potential Bitcoins (max supply) were in circulation.)**
  - **total_volume (This is the total trading volume of Bitcoin in the last 24 hours, representing the total value of Bitcoin traded during that period.)**
  - **price_change_24h (This represents the change in the price of Bitcoin over the last 24 hours. In this case, it's a decrease.)**
  - **circulating_supply (The total number of Bitcoins currently in circulation.)**

- *How to get coin data?*

To obtain coin data, you can use the **CoinGecko API**, which provides comprehensive information about various cryptocurrencies. You can access the data by making an API request to the following endpoint:

https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=2&sparkline=false&locale=en

This API endpoint allows you to list all supported coins along with their price, market capitalization, trading volume, and other market-related data. The provided link will give you access to the specific data you need.

For more details on how to use the API and the available endpoints, you can refer to the CoinGecko API documentation:

https://www.coingecko.com/en/api/documentation

- *How should the application handle display?*

Create an interactive catalog with a main screen that lists coins and provides detailed information for each individual coin.

- *Environment :*

Next Js: https://nextjs.org/

Requirements:

The project must be hosted on a private GitHub repository. To launch your project we will execute the following command:

```
$ cd coins-catalog
$ pnpm install
$ pnpm run build
$ pnpm run start -- --port=8000
```

All the commands must work and should run a web server with your application available on port 8000.

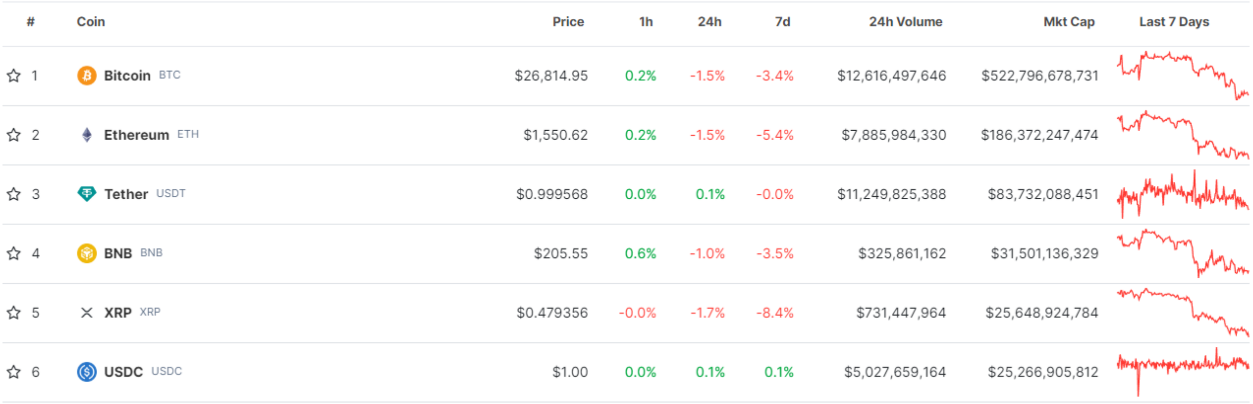
Steps :

- ## *Coins Display*

Design a page to showcase all available coins with a focus on essential details, including the identifier, type, and symbol.

Acceptance criteria:

- *A title Available coins should be displayed*
- *All coins must be displayed on one page :*
    - *Each currency must display its name, image, and symbol.*
    - *You can add a border to distinguish each currency.*



	#	Coin	Price	1h	24h	7d	24h Volume	Mkt Cap	Last 7 Days
☆	1	Bitcoin BTC	$26,814.95	0.2%	-1.5%	-3.4%	$12,616,497,646	$522,796,678,731	
☆	2	Ethereum ETH	$1,550.62	0.2%	-1.5%	-5.4%	$7,885,984,330	$186,372,247,474	
☆	3	Tether USDT	$0.999568	0.0%	0.1%	-0.0%	$11,249,825,388	$83,732,088,451	
☆	4	BNB BNB	$205.55	0.6%	-1.0%	-3.5%	$325,861,162	$31,501,136,329	
☆	5	XRP XRP	$0.479356	-0.0%	-1.7%	-8.4%	$731,447,964	$25,648,924,784	
☆	6	USDC USDC	$1.00	0.0%	0.1%	0.1%	$5,027,659,164	$25,266,905,812	

NB: This example is for illustrative purposes only, you can be as creative about the design as you want!

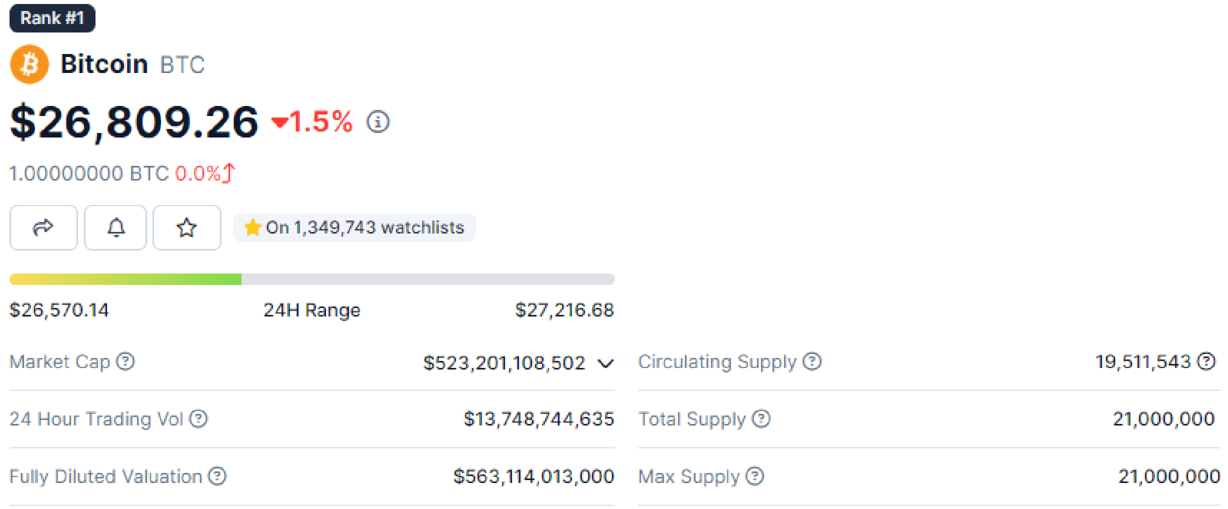
- ## *Display currency details*

Upon selecting a coin from the main page, ensure that the user is redirected to the currency detail page, where all currency attributes are presented. Include a back button to enable users to return to the currency list page.

Acceptance criteria:

- *Create a dedicated page for displaying all currency data.*
- *This page should be accessible via the following route: "**http://localhost:8000/coins/{id}**," where "**{id}**" will be replaced by the currency identifier (e.g., "http://localhost:8000/coins/bitcoin").*

- *When a currency on the main page is clicked, users should be automatically redirected to the detailed page, where all currency attributes, including the identifier, are visible.*



Rank #1

Bitcoin BTC

$26,809.26 ▾1.5%

1.00000000 BTC 0.0%

On 1,349,743 watchlists

$26,570.14 24H Range $27,216.68

Market Cap	$523,201,108,502	Circulating Supply	19,511,543
24 Hour Trading Vol	$13,748,744,635	Total Supply	21,000,000
Fully Diluted Valuation	$563,114,013,000	Max Supply	21,000,000

NB: This example is for illustrative purposes only, you can be as creative about the design as you want!

In this step, you will be tasked with retrieving data from the Coingecko API, specifically by utilizing the resources provided by **/v3/coins/markets**. On the main page, your objective is to present a comprehensive list of all the available coins. When a user selects a particular currency, your application should promptly fetch and display detailed information about that specific currency.

Acceptance criteria:

- *The main page should display the first 100 coins*
- *When selecting a currency from the main page, the detailed page should display all data for any of the available currency*

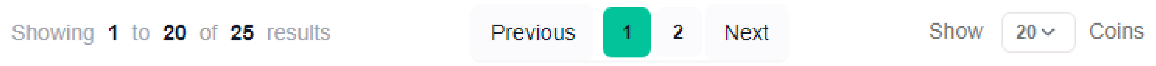
Bonus

Bonus 1: Pagination

To enhance the user-friendliness of the interface due to the large number of coins, we will implement a pagination feature. By default, the application should display **10 coins per page**, but it should also provide users with the option to select how many coins they want to see per page.

Acceptance criteria:

- *Pagination should be displayed at the bottom of the list of coins*
- *The Select box should allow to display of 10 (default value), 50, or 100 coins per page*
- *We should have a previous and next button to go from page to page*
- *If we are on the first page, the previous button should be disabled*
- *If we are on the last page, the next button should be disabled*
- *We should have a button to rewind to the first page or to forward to the last page*



Showing 1 to 20 of 25 results

NB: This example is for illustrative purposes only, you can be as creative about the design as you want!

Bonus 2: Filter Options

Incorporate filter options to enable users to locate coins based on the following criteria:

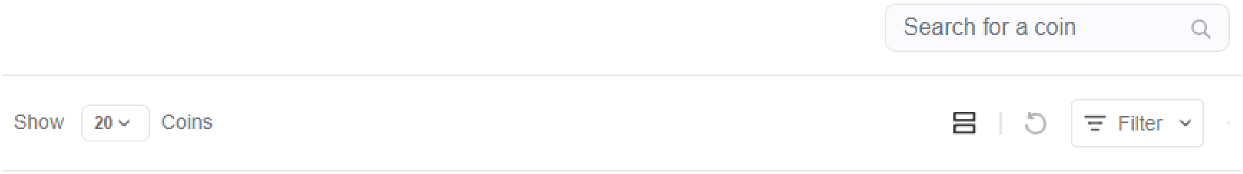
- By ID: Users can search for coins by their unique identifiers.
- By Code: Users can search for coins using their specific codes.
- By Name: Users can search for coins by their names.
- By Type: Users can filter coins by their types.

These filtering options will improve the search and navigation capabilities within the application.

Acceptance criteria:

- *Provide a select box to let users specify which field they want to filter on.*

- *Include a search functionality with a text box that allows users to search for coins based on any field or attribute.*



NB: This example is for illustrative purposes only, you can be as creative about the design as you want!

Bonus 3: Responsive UI

Revise the existing design to ensure compatibility with mobile and tablet devices. You can utilize a responsive web design framework or library of your choice to achieve this. Responsive design will adapt the layout and elements to provide a seamless and user-friendly experience across various screen sizes and devices.

Acceptance criteria:

- *On mobile, we should display 2 coins per row*
- *On a tablet, we should display 4 coins per row*
- *On the desktop, we should display 6 coins per row*

Task 2: Algorithmic

Write a code in **two** programming languages of your choice that prints the numbers from 1 to 100, however, for every multiple of ‘3’ it should print “Hello” instead of the number and for the multiples of ‘5’ it prints “World” and for the multiples of ‘7’ it prints “Yoo”.

Example of desired output:

1, 2, Hello, 4, World, Hello, Yoo, 8, Hello, World, 11, Hello, 13, Yoo, Hello World, 16, 17, Hello, 19, World, Hello Yoo, 22, 23, Hello, World, 26, Hello, Yoo, 29, Hello World, 31, 32, Hello, 34, World, Hello, 37, ...

Task 3: Logic

Scenario:

You are on the highway in your green car, and your friend contacts you to inform you that his red car has broken down. The challenge is that neither of you knows which direction he is in.

Explain in the most clear way, how would you proceed to find your friend's car in a **finite amount of time**. Remember, all you can do is drive.

Assumptions & limitations:

- You can’t leave the highway.
- You can’t call your friend.
- The highway is **bidirectional**, meaning you can move with your car in any direction you want.
- The highway is infinite, meaning if you move in the same direction, you will never get to an end.
- Your car has infinite gas to run until you find your friend.