

Blockchain Project Showcase

Analysis

Introduction

After cloning of the repository, the next step was to analyze carefully the project showcase problem and it involved 5 main objectives :-

1. Display of relevant data
2. UI/UX design
3. Implementation of the design
4. Unit test for the components
5. Integration and more testing

Content

Display of relevant data

According to the task we have to display data about blocks for a public block chain. Sample data was given but the API for the public data was not given so the first step was to research for a suitable API that could provide information a public blockchain

After hours of screen and research finally got a public api provided for ethereum data.

The public api was a GraphQL API which was fun as I could query exactly what type of data required and have all the perks of GraphQL

The link the API can be found [here](#) link, and like all public APIs it has its limitations and benefits. The benefits of the API was it had a playground for the queries which allowed me to experiment with the data it provided and the data which it didn't. After lots of time playing it was time for UI/UX as I was confident I had all the data required for the project.

UI/UX design and implementation

This step requires the showcasing of our design capabilities and the implementation of the design.

Design is no easy fit as it requires a lot of processes and steps to bring out a good design. But due to the time limit I had to look at various designs already available in Pinterest, Dribbble and other platforms to meet up with the short time requirement

After much research and searching I agreed on a design found in this link. The design had to adhere to one of the most fundamental aspects of a block chain platform and that is a dark theme, this is fundamental because users will have to stare at screen for long hours which can have effects on eye strain.

Even though I had access to design pics, I didn't have access to the design files and purchasing one will not be cost effective for the task, so I foiled in the missing parts of the design with my skills. Like all designs it will never be the final design and there will always be ways to update this the Agile development was required for this project. It was time to get to implementation.

Implementation of design

Story book is great for the development process but we are limited by time so for my implementation process I went for the next best thing, that's is dump all my components on a test page where it can be accessed by every developer and easy for them to use.

For the implementation process I started with bootstrapping my application with **create-next-app** coupled with **tailwind css**. From here I proceeded to structuring the code base for development. I used a structure I find suitable for navigating and expanding in future

Next I started the development with small components like **stamps, buttons, cards**, and more. Confident with the components I had I proceeded to block components such as **transactions tables, blocks tables, dash transaction table, transaction cards**, and more

Next after development of components I proceeded to the development of the layout the **side-nav** and **top-bar**.

While working on the components and screen I simultaneously worked in the dark theme which was an essential part of the project.

Development of the various parts are always great and easy to achieve next was to develop the screen required for the project and I had three main screens

- dashboard screen
- Transaction screen
- Blocks screen

Started with the dashboard screen which gives a brief overview of most of the data for the block chain.

Next was the transactions screen which gives information about recent transactions of the block chain followed by the development of the blocks page giving information about the recent blocks in the blockchain.

Unit test for the components

There's no excuse I could give as to why I didn't do unit test in parallel with the development of my components. But test are important so I dedicated a whole working session to write test for my components, even though typescript ensures a level of correctness of code, test are still important so I wrote test for all my components and ensured it worked accordingly to the way it was designed

I used jest for testing in this project.

Integration and more testing

For integration I always use state management and axios. Installed the packages and set up redux and axios for the integration process

During the integration process, the data returned was not exactly formatted to the way it was to be used on the platform so I wrote functions to clean the data received, functions to shorten long string of data like the hash, bloom and others.

Used environment variables to store my keys and other confidential information. The design lacked means of refreshing the data and loading more data so I add more components and test for refresh button and loading more button for the data

Used redux to implement filtering of data using the search input components, so users will be able to search for wallets, transactions, hash and more.

Conclusion

There were lots of drawbacks and cutting of corners due to the limitations of time but the project ever remains to be improved. Some limitations and drawbacks I face are:-

1. I didn't have time to implement a tour guide for the application so that users can be able to know how to properly navigate the platform

2. The public API could not handle stress of constantly being refreshed so I implemented a notification to tell users when an error occurred
3. I was obsessed with responsive design and I couldn't help myself but implement the responsive design for mobile users and it cost me lots of time.
4. I used tailwind in the css files to avoid clustering my tsx files. I know it is not recommended by the tailwind docs but I prefer using it that way

Overall, the project was great and I hoped I was able to accomplish the task required. Below are some screenshots of the project and a comparison between the image design selected and the final design I made.

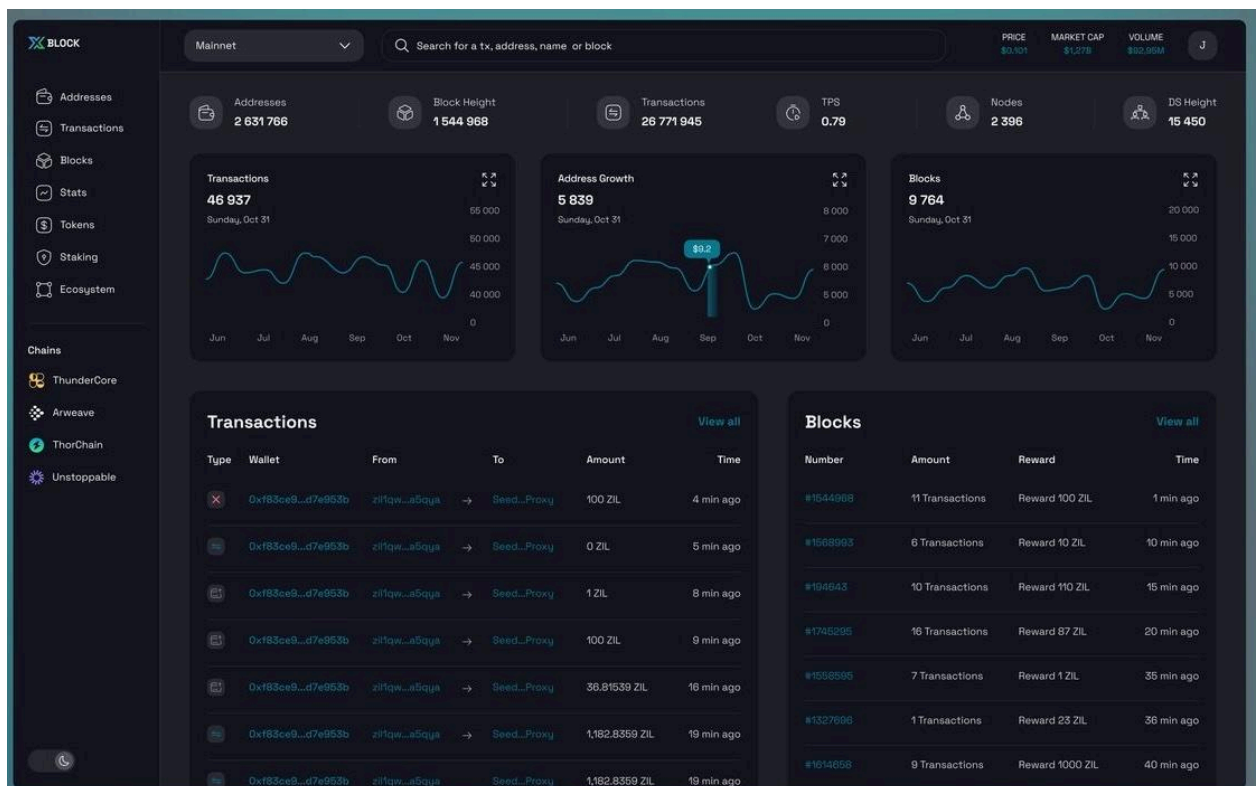


Figure 1: Picture of selected UI design

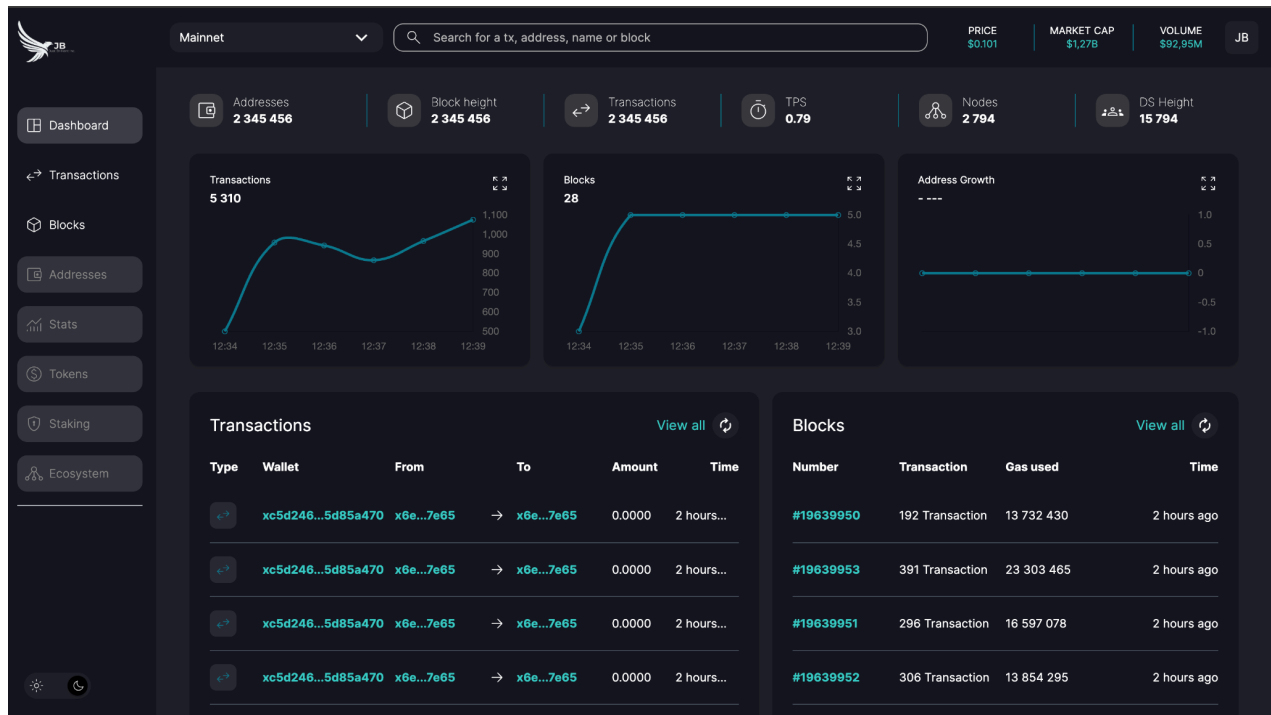


Figure 2: Developed UI

Some tools








Transactions						View all	
Type	Wallet	From	To	Amount	Time		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		
	xc5d246...5d85a470	x6e...7e65	→ x6e...7e65	0.0000	2 hours...		

Figure 3: Dashboard Transaction table

The dashboard transaction table has at the top right corner: -

- **View all** link, for the user to navigate to the transaction page to view more information and load more
- **Refresh button**, for the user to refresh the data for the transactions.

These are also found on the dashboard Blocks table. I choose a method between hybrid refresh and an onload page refresh due to the limitation of the API link and limited number credits for a free token.

This means that the API has an amount of credits that are allocated for a token generated by a user account. So for each API call, depending on a lot of factors, a certain amount of credit is deducted for it. For this reason, the use manually refreshes and each time the user leaves the page and comes back it is automatically refreshed to save on credits.

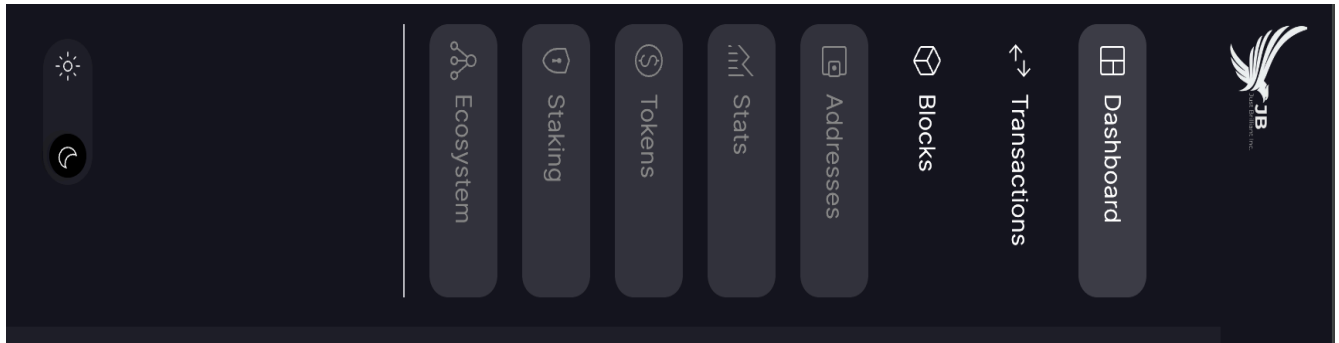


Figure 4: Side navigation

For the side navigation, at the bottom we have:-

- **The theme toggler** to switch between light and dark mode
- **Navlinks** to help the user navigate the platform.

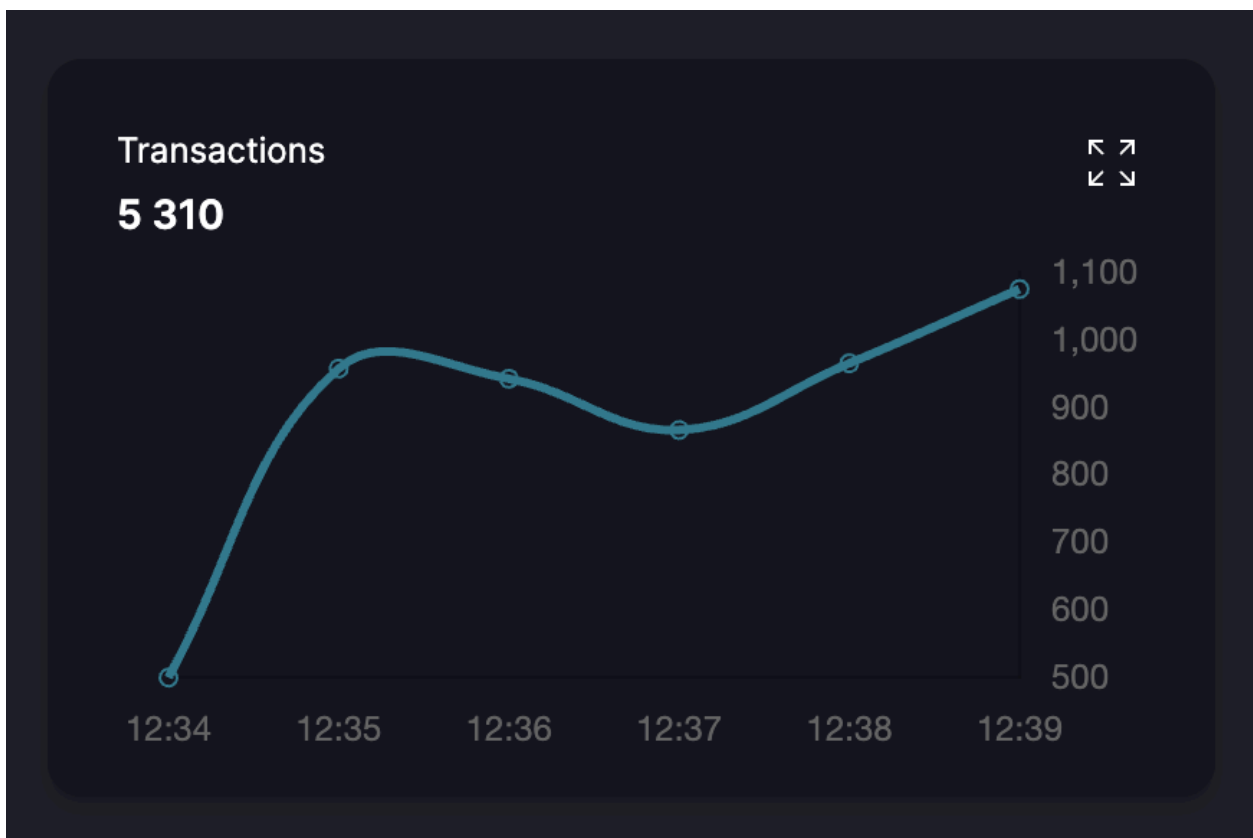


Figure 5: Chart card

For the chart card we have: -

- **Expand button**, to expand the card for the user to better see the details on the card
- Information title and total for the user to understand the chart

- For the x axis we have the time span and the y axis the amount



Figure 6: Search Input

The main use of the search input is for the user to search data, based on the tx (transactions number) address (hash, wallet, bloom) name and block.