

'WEB3-GAMING PROJECTS EXPLORER'

PROGRAMMING CHALLENGE IN NODE+REACT

last update: 2024-01-02

INDEX

Description	3
Requirements	3
Optional Enhancements	4
Evaluation Criteria	4
Time Delivery	4

DESCRIPTION

Create a React application allowing users to explore Web3 Gaming Projects . The web-app should use the [CoinMarketCap RESTful JSON API](#) (CMC) for fetching data.

Note: Only Gaming projects should be provided to the user

REQUIREMENTS

Home Screen

- Display a fixed list (no horizontal scroll) of the top 5 most popular crypto gaming projects retrieved from the CoinMarketCap API - top should be ordered by the ranking provided by CMC;
- Each project item should show the project's icon, name, symbol, and price;
- Display a scrollable list (horizontal scroll) of the favorite projects, marked by the user, also showing icon, name, symbol and price.

Detail Screen

- When a user taps on a project item on the Home Screen, navigate to a detailed view of that project;
- Show a detailed view about the selected project, such as, icon, name, symbol, price, url (clickable), description and other info (please choose the most appropriate and content rich data);
- Include a button to mark/unmark the project as favorite. The favorite status should be stored locally on the device.

Local Storage

- Implement a local storage solution to persist the user's favorite project across app sessions.
- Use React Hooks or another suitable method for storing and retrieving favorite projects.

Error Handling

- Implement robust error handling for API requests. Display a user-friendly message when there is a network error or if the API request fails.

UX/UI

- Develop the UI so it is easily navigated by users using a desktop computer, but also possible to navigate using mobile devices. It will be a PLUS if the Web App is developed according to the Responsive Web Design principles
- Implement a navigation header with appropriate titles and navigation controls.

OPTIONAL ENHANCEMENTS

- Implement an evolution arrow to show price variation between last and current user's browsing session (green arrow up for price increase, dash for same price, red arrow down for price decrease). Arrow should be positioned right after project's token/coin price and have the previous price under braces. Here's an output example for this mechanism using Bitcoin price evolution:

BTC 42.352 \$ ↑ (41.683 \$)

- Implement a search feature on the Home Screen that allows users to search for projects by title, which should query the CMC's API for the effect.
- Implement pagination for the projects list to load more projects as the user scrolls.

EVALUATION CRITERIA

- **Code Quality:** Evaluate the cleanliness, organization, and structure of the code.
- **API Integration:** Verify that the app successfully fetches data from the TMDb API and handles errors gracefully.
- **Data Storage:** Check if the local storage solution effectively persists and retrieves favorite projects.
- **Quality Assurance:** Make sure the project as a whole works bug-free and, if you still have time, develop unit/automated tests
- **Completeness:** Try to develop according to the specification, as much as possible.
- **UI/UX Design:** Assess the user interface for a visually appealing and intuitive design.

Note: the criteria is ordered in terms of importance, from most (top) to least (down) important. All topics should be taken into consideration though.

This project will test the developer's skills in Node, React, API integration, state management, and Web Design, making it a comprehensive and challenging task.

TIME DELIVERY

Send the *github* project link to jobs@blockbastards.io with the subject: "BB Programming Challenge - Your Name" The time frame to perform this challenge should be done within a day after you receive it.

IMPORTANT NOTE: Don't panic if you cannot accomplish all your goals. Just go feature by feature and do your best! :)