Test Assignment for Senior FullStack Developer

Overview

You are tasked with creating a simple web application allowing users to view current cryptocurrency prices for the TON/USDT pair and USDT/TON.

The back end should provide an API that interacts with a cryptocurrency pricing service to handle current prices. In contrast, the front end should present a user interface to select a cryptocurrency.

Scalability Considerations:

You should consider the potential growth of the application. While you are not required to implement these features in this assignment, the solution should allow for future expansion, including:

- Supporting more trading pairs and tokens.

- Displaying historical cryptocurrency data.

- Integrating with multiple data providers.

Requirements Backend

- Develop a REST or GraphQL API using Node.js (e.g., Express, NestJS, or any other framework you choose; please explain your choice in the README file).

- The API should provide current cryptocurrency prices using CoinGecko, CoinMarketCap, or any other provider, with a maximum delay of no more than 30 minutes.

- Implement at least one unit test for backend functionality.

- Data caching should be implemented to minimize calls to the external pricing service.

Frontend

- Create a simple user interface using a modern JavaScript framework (e.g., React, Vue, Angular).

- The UI should allow the user to view the current price of selected cryptocurrencies.

- Aesthetic design is not required, but the application should focus on functionality and usability.

- Implement at least one unit test for frontend functionality. Database - You can use any database you choose (e.g., PostgreSQL, MongoDB).

- The database configuration must be provided via a Dockerfile or docker-compose file for easy setup. Please include instructions in the README on how to run the Docker commands to start the database.

General Requirements

- Include a README file in the repository with the following: - Instructions on how to set up and run the project.

- Recommendations for potential improvements or additional features that could be added.

- Code quality, structure, and best practices will be evaluated.

- The solution should demonstrate thoughtful architectural decisions, code modularity, and clean external data handling.

Deliverables

- Git repository containing:

- Backend and frontend code.

- Docker configuration files for the database and other dependencies

- README with setup and run instructions and suggestions for future enhancements

Expectations

- The solution should reflect senior-level experience, showcasing the best back-end and front-end development practices.

- Tests are mandatory (at least one backend and one frontend test).

We look forward to seeing how you approach this assignment and how you demonstrate your skills!