

Assignment: Simple Currency Converter

Overview

Develop a small application that converts amounts between different crypto currencies. The application will involve calling a public currency conversion API, performing some calculations, and displaying the results.

Technical Requirements

Backend (Node.js & Express.js):

- Set up an API endpoint to fetch the top 100 cryptocurrencies and supported currencies(USD, EUR, etc).
- Set up an API endpoint that takes a source cryptocurrency, amount, and target currency(EUR, USD, etc) as parameters.
- Use a public API (like [Coingecko](#) or [Coinmarketcap](#)) to get real-time exchange rates of cryptocurrencies and top 100 cryptocurrencies.
- Perform the currency conversion(between crypto and selected currency(USD, EUR, etc) on the server and return the result.

Frontend (React):

- Design a simple form where users can select a source cryptocurrency(fetch the list from api), input the amount, target currency(USD, EUR, etc), by default USD should be selected, and submit the request.
- Display the converted amount to the user.
- Ensure basic error handling and validation of user inputs.
- Impress us with your UI creativity.

Code Quality and Architecture:

- Write clean, well-organized(proper folder structuring), and documented code.
- Structure the application in a way that demonstrates good understanding of software architecture.

Testing and Deployment:

- Include basic unit tests for your backend logic - Good to have
- Deploy the frontend on Vercel and the backend on a cloud platform of your choice.

GitHub Repository:

- Use Git for version control with clear commit messages.
- Organize the repository to reflect professional coding standards.

Deliverables

- Link to the GitHub repository.
- Live URLs for the backend and frontend.

Evaluation Criteria

- **Functionality:** Does the application perform the required task?
- **Code Quality:** Is the code clean, efficient, and well-organized?
- **Logical Implementation:** How is the currency conversion logic implemented?
- **Testing:** Are there basic tests, and do they cover key functionalities?
- **Architectural Understanding:** Does the structure of the project reflect good software architecture principles?