



CHALLENGE

A survey company wants to make a new quiz form that rewards users with tokens for participating in the survey.

They create surveys, loading them into JSON format and they want you to develop a responsive web application using React and TypeScript for people to answer and submit the results, getting tokens in the process.

Requirements:

- Next.js (preferred), CRA or custom webpack for the front end infrastructure, taking into account for the architecture of the project hierarchy and organization of directories, routing, conventions and good practices of clean code, good design patterns for the app itself but also for the react components
- For the app state, Mobx State Tree (preferred) or React Hooks
- Ant Design for components (plus)
- Add a descriptive "readme" file, it should also include links to any resources and documentation used to solve the challenge.

The web should behave as follows:

- Connect Metamask wallet
- Ensure user is connected to Goerli and if not show a button to switch networks automatically.
- Show balance of \$QUIZ token (address below).
- Once the page is loaded, present the title of the daily trivia with its picture and a button

that allows you to begin answering.

- Once the survey starts, display the current question, which will be available for the amount
- of seconds in the lifetimeSeconds property.
- Answered or not it should move onto the next question.
- Once all the questions are finished, show an overview with all the answers.
- Show a button to submit the questions to the validator contract
- Refresh the balance of \$QUIZ



Evaluation

The challenge is simple, so we will hold the solution to the highest quality standard, considering:

- Aesthetic, clean code (use linter)
- Senseful folder structure
- Scalable solution

Keep the organization of branches and commits as if you were working in a team

Hints & Links

Survey Contract Repo Survey Sample \$QUIZ Token in Goerli AntDesign

Upon completion

Follow the steps provided by your recruiting partner to allow recruiting engineers review your code and solution.

Include in Readme file explanation on env vars required for running the project. Include also some pictures or video showing the project working. Deploy the application to vercel, netlify or any free hosting service for quick testing