

FrontEnd developer task

Basic instructions

1. Figma design is [here](#).
Password: **tesotask22**

Tip: Login to Figma to see color codes, spacings and font sizes.
2. Assets can be downloaded [here](#).
Password: **tesotask22**
3. Do your best with the task
4. Prepare a repository in github, or any other similar platform, and let us know that you are done (feel free to add comment about the task)

Requirements

Design

- Design should be recreated as per provided design. We aren't talking about pixel perfect, we know there are gazillion screen sizes these days, just follow best UI/UX patterns, don't invent new icons, colors or spacing and you're all good!
- Design must be mobile-friendly and responsive.
- Use SVG's where possible
- For CSS you can use whatever you need and feel comfortable with (vanilla CSS is an option too), but we strongly recommend these:
 - pre-processor ([SCSS](#))
 - [CSS-in-JS](#) library ([styled-components](#) , [JSS](#) , [emotion](#) , [aphrodite](#))
 - [BEM](#) methodology

App

- Use ES6+ features where applicable
- Use [react.js](#)
- Use state management solution ([redux](#) preferably). We know it is possible to do without, but we are curious - can you do with?
- This must be a single page application. Use routing library([react-router-dom](#) or [@reach/router](#))
- Implement login by sending an authorization request (POST) to <https://playground.tesonet.it/v1/tokens> to generate a token (don't forget to pass Content-Type):

```
{ "username": "tesonet", "password": "partyanimal" }
```

- Use browser storage solution to persist token between sessions
- Use the token to retrieve the server list from <https://playground.tesonet.it/v1/servers>,
- Order the results of servers list by `distance` and `name`.
- Implement logout (don't forget about the token in the storage)
- Your app must work on all modern browsers and IE11+

Bonus

- It is all good to use [create-react-app](#) as a starter, but if you have time and want showcase your skill - use JS bundler ([Webpack](#) preferred)
- We highly recommend following [TDD](#) patterns and showcasing your skills at writing tests([unit](#) , [integration](#) , [e2e](#) - all are good)
- Use [npm scripts](#) for running tasks, i.e. for development, clening build or etc.
- Do validation of login fields and provide user friendly error messages if needed **#UXmatters**
- Indicate loading state for the user whenever requests are in action and user might not understand they needs to wait

Few tips

- Imagine this as a production level product at scale
- Structure! With great structure, comes great reusability!
- Maybe You have an idea that is not in the task? Do it! It's on you!
- Have fun!