

Frontend Assignment (React js)

Some ground rules:

Please read these instructions carefully

1. We expect all your solution components - Code, configuration files, readme, documentations, diagrams if any/applicable - push to a public Git repository (e.g Github) where we can access it.
2. We expect that we can run your code easily and you have provided us with kind simple instructions to run your code. If we are not able to run your code then we won't be able to evaluate it.

Problem statement

We'd like you to write a single-page application using React and [GitHub API](#) for showing the pull requests data for the [neovim](#) repository in a table/list format. These are the following things which we would like to see in the application:

1. Table/List should be infinitely scrollable so that when we reach the end of the current number of Pull Requests it should fetch the next number of Pull Requests on the same page.
2. We should be able to see a loader when it will be fetching the data from the API. You can get as creative with the loader as you want.
3. The table/list should have these mandatory fields:
 - a. Title
 - b. Default Branch
 - c. Author
 - d. Created On
 - e. Reviewers
 - f. Labels
4. We should be able to click on a Pull Request and go to its respective detail page in Github.
5. Keep in mind that the Pull Request title can be long so you need to think of showing that completely in the list for every Pull Request. Basically, every field data should be clearly visible to the user.
6. Use only functional components.

Additional Information

1. You might need to create an access token for GitHub API which will give you enough API hit rate limit. You can do that by creating an env file that can store your token as it is not suggested to push your access token to any public repo.
2. We will be evaluating your solution on the basis of these criteria:
 - a. Maintainability
 - b. Reusability
 - c. User Experience
3. Be creative with the design for the table/list. This exercise is not meant to test whether you can program at all, but instead, you should think of it as a software design test.

Here are some extra pointers that might help:

As always start with a simple solution noting down the limitations and slowly start adding the complexity.

4. For implementing an Infinite scroll you can use [Virtuoso](#) which is an out-of-the-box library that handles everything you need to implement in this task.

Bonus Points:

1. Use of Custom Hooks
2. Responsive UI

Please push your code to Github along with README consisting of instructions on how to run your code and send the link to us.

Looking forward to discussing your solution! All the best