React-Intern assignment

ReactJS assignment Preview

Here we provide the instructions for developing a ReactJS applications, designed by the Bipolar factory team, to test the proficiency of applicants for the ReactJS Frontend Developer position(s):

Assignment 1 - Beginner: For applicants who have a good working knowledge of React and have worked on / developed at least one React application

All applicants are required to submit the assignment based on their knowledge, experience and confidence in ReactJS. We advice applicants to see the demo and instructions for the assignments before deciding which one to submit

Prerequisites

- Knowledge of ReactJS (obviously), HTML, CSS, Bootstrap (optional, but will be really helpful for the Assignment)
- Knowledge of installing npm packages (both local & global)
- Knowledge of creating React application
- Knowledge of fetching JSON data from a REST API endpoint
- A Github account in order to share his/her code with us

Sharing the code:

All applicants must share their code in one of the following ways:

- ➢ Upload their code on their Github account by creating a public repository

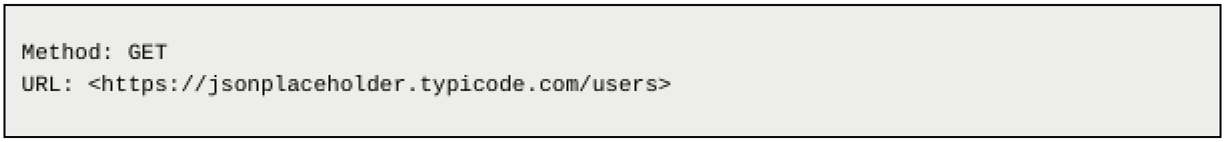
Assignment Details :

The assignment is a frontend React application with no backend development Required.

The idea of the assignments is to build a single page that displays the profile of 10 users (the data is obtained from an API endpoint). Each user's profile contains a avatar picture, name, email, phone, address, website and company name

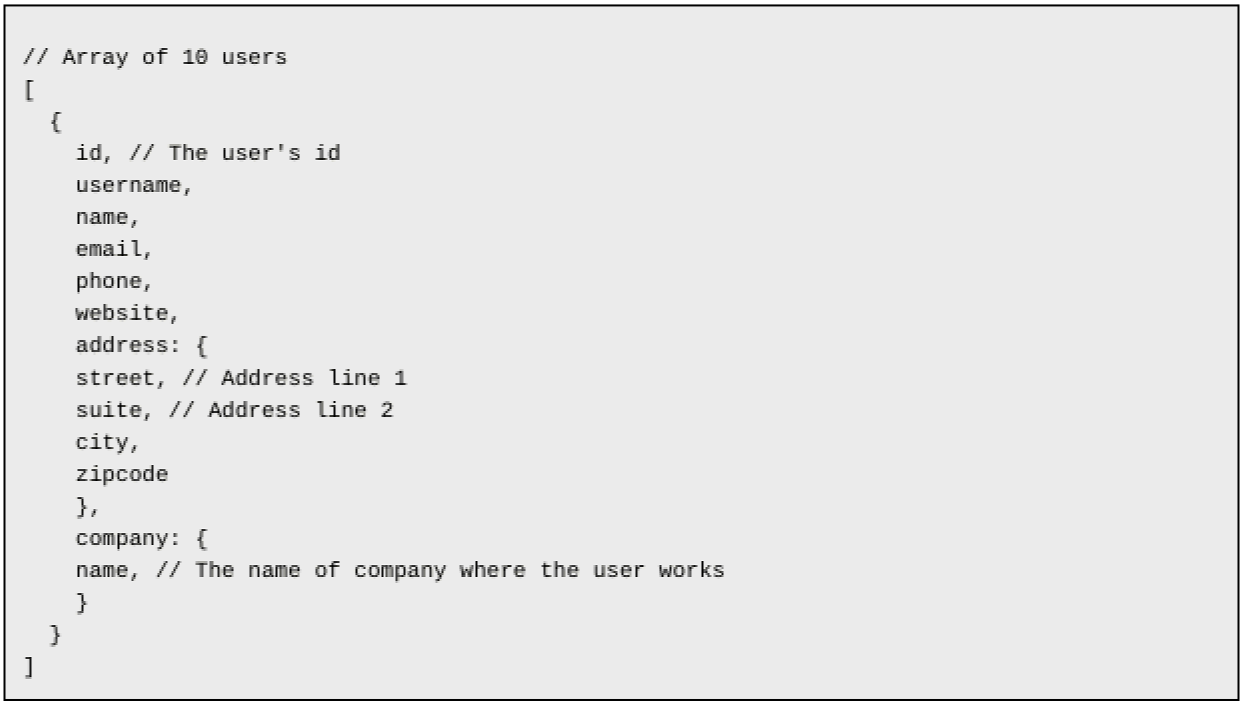
API endpoint for users data

- ❖ All 10 users profile data is to be downloaded from the following API endpoint:



```
Method: GET
URL: <https://jsonplaceholder.typicode.com/users>
```

The schema of the data received in the response is :

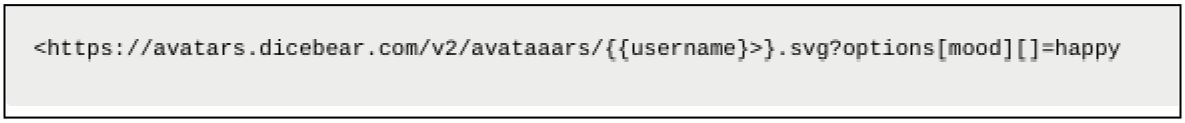


```
// Array of 10 users
[
  {
    id, // The user's id
    username,
    name,
    email,
    phone,
    website,
    address: {
    street, // Address line 1
    suite, // Address line 2
    city,
    zipcode
    },
    company: {
    name, // The name of company where the user works
    }
  }
]
```

API endpoint for users' avatar pictures :

You will be using Avatars by DiceBear. They provide a free HTTP API to create unique avatar images based on any string we send as a query parameter. Each string generates a unique image. You will use the username to generate a unique avatar for each user.

The URL for the GET endpoint is



```
<https://avatars.dicebear.com/v2/avataaars/{{username}>}.svg?options[mood][]=happy
```

The {{username}} in the URL is the placeholder for the user's username. It should be dynamically replaced by the username received from the users API endpoint. For example, if the username for one of the users is psamd then the URL for the avatar for this user will be:

https://avatars.dicebear.com/v2/avataaars/psamd.svg?options[mood][]=happy

Loading Indicator
Upon opening the app a loading indicator is displayed until the data is fetched from the API and is ready to be displayed. The source code for the loading indicator can be obtained from: http://tobiasahlin.com/spinkit/

Assignment 1 - Beginner Deadline for submission: 3 ½ hour

DEMO - https://react-advanced-assignment.psamd.now.sh/

This assignment is designed for applicants who have a good knowledge and understanding of React and have developed/worked on at least one React application.

What are we looking for?

With this assignment we would evaluate the following:

- Ability to create new react projects using vite
- Understanding of JSX
- Passing props to components
- Create React App
- Understanding of stateful and stateless Components
- Basic understanding of state management and component lifecycle methods
- Fetching data from an API endpoint
- Conditional rendering
- Working with lists
- Ability of the applicant to learn a new React UI library and use its components in
- their app
- Handling events and working with forms
- Lifting State Up

General Tips :

- Feel free to use Google, StackOverflow or any other resource
- Examine the demo apps closely to determine all the features
- Open the data API link in your browser and examine the response schema
- Try to match the UI design of the demo for each assignment as closely as possible