C3 DEV TEST (React)

> PLEASE CLONE/FORK THIS REPOSITORY AND COMMIT WORK TO YOUR OWN VERSION.

Start here ...

> `npm install` to install all the packages

> `npm test` to ensure the application is working as expected

> `npm run dev` - will run application and watch for developer changes, then you're good to go

Commands

- `dev` - runs the web application and watch for file changes

```
npm run dev
```

- `test` - runs the available component tests

```
npm test
```

- `test:watch` - runs the available unit tests, file changes will trigger a re-run npm test:watch

```
npm run test:watch
```

What are we looking for?

- We want to see your understanding of JavaScript (including ES6 and ES7) so please stick to VanillaJS.
- We love to see testable components and code, see the testing section below
- we want to see clean class naming BEM or OOCSS
- Do all your styling within `pages/index.scss`
- You can create any number of components within the `components` folder

Testing

> Please take time to familiarise yourself with the unit testing tools/examples by reading the documentation. Showing good examples of testing will impressing us.

Our tests use Mocha and Chai assertions. additional helper libraries are available to make testing easier:

- enzyme - provide simple API to test React components
- sinon - testing utility

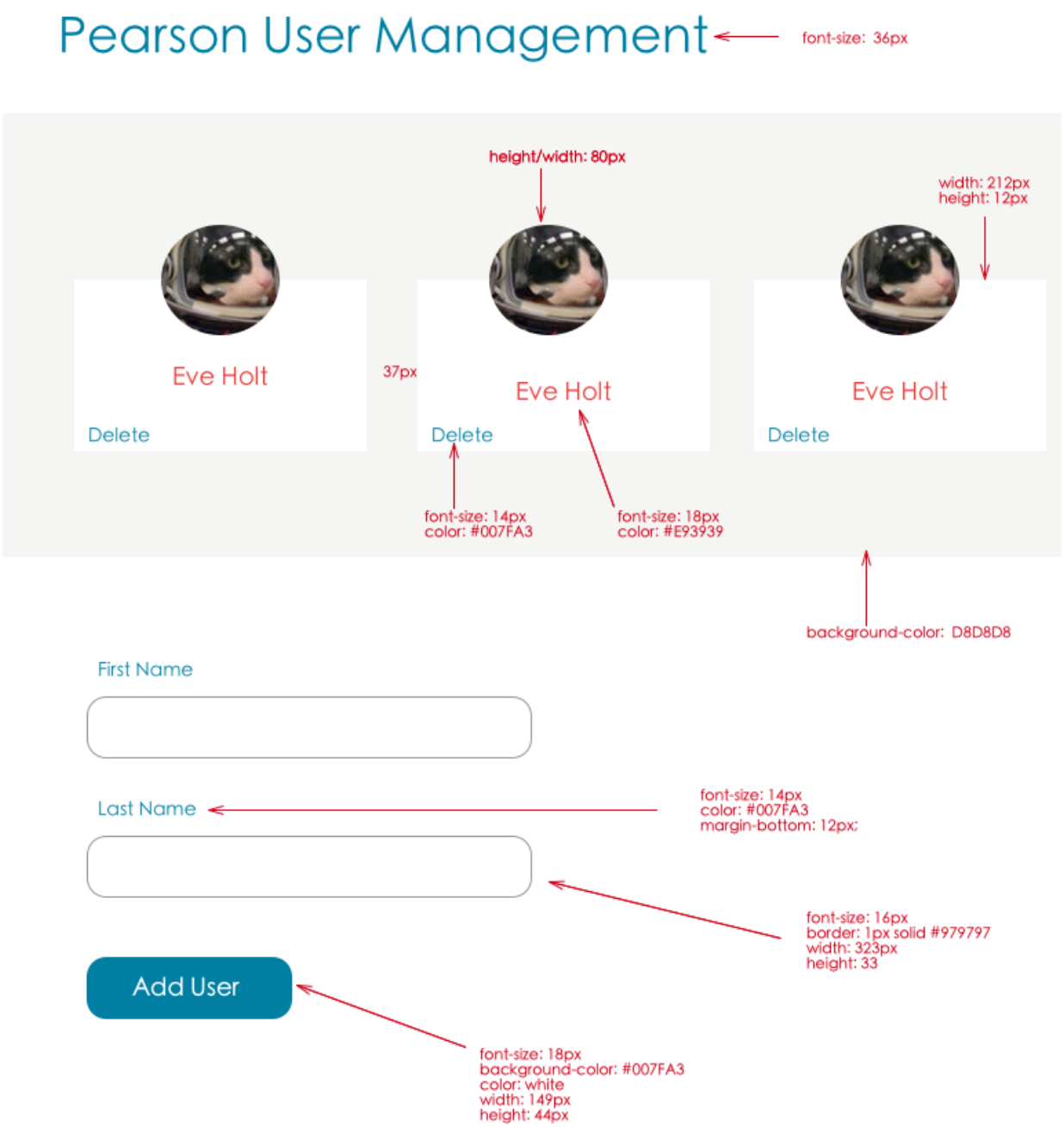
> See example tests for usages.

Task

The task aims to highlight to us your experience, and understanding of the three main components within our team; JavaScript, ReactJS and CSS/SCSS.

Phase 1

Build the following user interface using ReactJS (see path: pages/index.js and components folder), CSS (see path: pages/index.scss)



> ^ click to see full image

things we would like to see:

- Valid HTML 5, use correct HTML tags
- Small testable components (any file with `.test.js` suffix will run when `npm test` is triggered.)
- Example such Button, Input, User avatar etc

- BEM for CSS classNames

Phase 2

`pages/index.js` has the following state:

```
this.state = {
  users: [
    {
      id: 4,
      first_name: "Eve",
      last_name: "Holt",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/marcoramires/128.jpg"
    },
    {
      id: 5,
      first_name: "Charles",
      last_name: "Morris",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/stephenmoon/128.jpg"
    },
    {
      id: 6,
      first_name: "Tracey",
      last_name: "Ramos",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
    },
    {
      id: 50,
      first_name: "Anish",
      last_name: "Patel",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
    },
    {
      id: 10,
      first_name: "Jack",
      last_name: "Bravo",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
    },
    {
      id: 12,
      first_name: "Ahmed",
      last_name: "Ali Khan",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
    }
  ]
};
```

- Extend phase 1 to include data from the state
- Implement `add user` functionality to support capturing user information(first name, last name, avatar url, generate a random id) from the form
- Implement `delete user` functionality, this is done by removing a user from the state
- Focus on writting clear/clean production quality code and tests

FAQ

1. How to run the application?

> All you need to run is `npm run dev` and then open the browser to `http://localhost:3001`

Any file changes will trigger the application to refresh on the browser, and any errors will also be shown on the browser.

2. How do you want the code deliveried ?

Please zip the folder and email it back to us.