Full Stack Tech Test 2021

⚠️ We understand that "at home" tech tests can take longer than pair programming tasks.
We are advocates of a healthy work/life balance and appreciate you taking the time to do this.

With this in mind, please refrain from any temptation to spend every evening of the week or the whole weekend on this. A couple of hours is just fine - we will provide you with the opportunity to talk through any incomplete features with us.

Build a Trello clone

We'd like you to build a small and simple <u>Trello</u> clone.

Required Functionality

💡 This is the core functionality we're looking for.

At a minimum, the Trello **interface** should consist of lists and cards:

- A list has a name and a collection of cards
- A card has a title and a description

The application should **work by**:

- Reading an API endpoint to populate the lists and cards
- Persisting new lists or cards created by the user
  - Considering time constraints, please prioritise the endpoint creation over a working UI here!

Optional functionality

⚠️ If you have time - you won't be marked on these.

- Allow users to delete cards

- Allow users to move cards between lists

- Add a due date to cards

- Add a visual indicator to cards that are overdue

What we're looking for:

> We will discuss this tech test with you in your second interview.

- A logical and maintainable approach to solving the problem

- Sensible use of testing

- Documentation where necessary

What we're not looking for:

- An entire UI design - if you have time to create a UI you can use a framework for this.
  - Material Design, Semantic UI etc
- User authentication / management
  - Assume the user is logged in, or hard code any details
- Absolute perfection
  - We really just want to see how you approach it
- Lots of time spent on this
  - Limit yourself to a couple of hours
  - You can talk us through any ideas / plans when we review the project.

Some guidance on the tech

Where possible, we like to use JavaScript both client side and server side. With that in mind we'd like to see:

- API endpoints written in Node.js - we'll leave any frameworks and libraries up to you

- We're fans of React.js so feel free to use this on the frontend but a library or framework isn't mandatory

- You are free to choose a data storage solution and we'd be interested in your point of view in this area

Submitting the test

- Please commit the project to a GitHub repository.

- Please provide instructions for starting the app and running any tests.