# About our interview

Meeting time: 03.17 03:30 ~ 04:00

Interviewer: **Dan**

# 

# About My Home Assignment

<https://github.com/MasterVentures/takehome-nextjs-test>

This is the initial home test project.

As you can see, this project is initially built with Next.js.

Next.js is an open-source web development framework built on top of Node.js enabling React based web applications functionalities such as server-side rendering and generating static websites.

And this project uses Next.js which supports typescript.

Typescript is a strict syntactical superset of JavaScript and adds optional static typing to the language. TypeScript is designed for the development of large applications and transcompiles to JavaScript.

By doing so, I can develop a full-featured react application which is easy to read and understand components and supports better static type checking and better intellisense and so on.

These are the requirements you specified.

So as a start of my home test, I have cloned this repository and made my own repository in my account and pushed initial project to there as you specified.

There are mainly 5 requirements.

## Step 1 (Requirement 1)

Here requirement 1 is all navigation should happen in the same page.

To do so, you should define client-side router in this app.

Next.js has a file-system based router built on the concept of pages.

When a file is added to the ‘pages’ directory, it’s automatically available as a route.

Additionally, I defined redirect route to make this app user-friendly. Defined in next.config.js

\*\*React's Strict Mode is **a development mode only feature for highlighting potential problems in an application**. It helps to identify unsafe lifecycles, legacy API usage, and a number of other features.\*\*

By coding and defining redirects() function, redirecting to /awards path is possible.

redirects is an async function that expects an array to be returned holding objects with source, destination, and permanent properties:

By this, I have done requirement 1 and committed and pushed to my repository as first commit.

## Step 2 (Requirement 3 & 5)

As a next step, I structured my project by splitting pages into multi components called component modularization.

So I created “awards” page component as our main page component like this.

This page includes many child components. I have defined these components in “components/awards” directory of root.

By this, I have done requirement 3 and 5 at once and pushed.

## Step 3 (Requirement 4)

Requirement 4 is to utilize css to create the layout of the page and add hover styles to the items that the user is interacting with.

To do this, first I defined style which will be applied to all pages and components globally and imported it in \_app component.

Here, I defined constant color variables by defining them in root block.

I get Google Roboto font by connecting to google roboto font link in \_document component.

\*\*A custom Document can update the <html> and <body> tags used to render a [Page](https://nextjs.org/docs/basic-features/pages). This file is only rendered on the server, so event handlers like onClick cannot be used in \_document.

To override the default Document, you can create the file pages/\_document.js.

<Html>, <Head />, <Main /> and <NextScript /> are required for the page to be properly rendered.

\*\*

I defined background color and font color and applied to body tag so that background color of page and font color are changed to our expected color.

And also defined many css classes to make layout of page. By defining container and row and justify-content-center and so on.

I have simulated css classnames as sames as bootstrap css classes and applied them.

By doing this, I have done creating layout.

And I also created each component css modules which will be applied to matched component and imported it.

\*\*We can style React Component in mainly four ways, which are given below:

* Inline Styling
* CSS Stylesheet
* CSS Module
* Styled Components

\*\*

To apply hover styles, I defined nominee hover style in this nominee css module.

By doing this, I have completed requirement 4.

## Step 4 (Requirement 2)

I think requirement 2 is the main feature of our application.

By using react hooks, we can implement the feature which selects one nominee per each category and submit it.

I used react hooks i.e. useState and useEffect at “awards” page component.

First, I used useEffect hook to define the function matched to componentDidMount lifecycle function in this function component.

The useEffect Hook **allows you to perform side effects in your components**.

To call the function only when it is mounted, you have to pass an empty array as the second parameter to useEffect hook.

In this function, I get fake ballots data using api that is already defined in this project.

After getting ballots data, I used useState hook to save that in the state of this component.

And outputted the saved ballots state data in render function using map function.

Noticing point here is I have passed select item function as props to Nominee component.

This select item function is called when user click one nominee card and accepts ballotId as ballots category id and itemId as each item id and state as its current selected state.

In this function, I did appropriate select or unselect operation due to parameter values.

And resaved in the state.

Like this, I saved selected nominee id in nominee category object.

Due to saved selected nominee id, in render function, I determined if the nominee is selected or not and changed button text and applied selected style due to the determination.

Additionally, I used redux to show the selected results in the modal when user submits.

Of course, there was no requirement of using redux and this can be easily done by using props.

But I always used to consider extending an application and so I decided to use Redux.

I used reduxjs toolkit to implement redux in our application.

\*\*Redux **Toolkit** is Simple. Includes utilities to simplify common use cases like store setup, creating reducers, immutable update logic, and more.

The recommended way to start new apps with **React** and Redux is by using that.

\*\*

So I defined store as a whole state management object and defined award reducer in it’s slice.

In award slice, I create it’s reducer which pushes awarded nominee information called submitAwards and exported its action.

In awards page, I dispatched submitAwards action by using useDispatch hook and opened the result modal.

In result modal component, it can get awarded result which is saved in store by using useSelector hook like this and output it.

Like this, I have done most of the requirements.

And after that, I made some additional requirements like mobile responsive feature and unit testing.

Lastly, updated Readme due to your requirements and did last checked and finally submitted my home test.

# About Company

# **About Master Ventures**

Master Ventures is a blockchain-focused venture studio helping to build the next generation of blockchain-based Web 3.0 system innovations within the crypto industry. Launched in 2018 by Founder and CEO Kyle Chassé, the company’s ethos can best be summarized in the acronym #BeBOLD: Benevolent, Open, Love, Decentralized.

Master Ventures co-creates with entrepreneurs and businesses worldwide to turn the best ideas into innovative and disruptive products. They do this by investing as strategic partners through offering advisory services to the projects they believe in. To date, Master Ventures has invested in over 40 crypto projects, including the likes of Kraken, Coinbase, Bitfinex, Reef, DAO Maker, Mantra DAO, Thorchain, and Elrond.

Master Ventures co-creates with entrepreneurs and businesses worldwide to turn the best ideas into innovative and disruptive products. They do this by investing as strategic partners through offering advisory services to the projects they believe in. To date, Master Ventures has invested in over 40 crypto projects, including the likes of Kraken, Coinbase, Bitfinex, Reef, DAO Maker, Mantra DAO, Thorchain, and Elrond.

## About Team

Master Disruptor & CEO: Kyle Chasse

Managing Partner: Dane Hoy

Head of Operations: Marta Vitvitska

Head of Marketing: Jessica Gonzales

CTO: Ben Stahlhood

Business Analyst: Wojciech Gtowacz

[Michael White](https://sg.linkedin.com/in/michael-white-4a4610185): Senior Blockchain Engineer

full-stack web **developer** within a dedicated, product-based team. Worked with React, Backbone.js, Bootstrap, Foundation or similar **front-end** web application

[Edgar Sucre](https://pa.linkedin.com/in/edgar-sucre-96308736): Senior Software Engineer

- Work with project managers to setup roadmaps. - Design solutions considering business requirements and the chief architect vision. - Help the **development**

## About ongoing projects