# NCS Ai Lead Intelligent Platform – Frontend Test

About the challenge

This is a basic test, in which you should be able to write a React or .NET Core app (depending on the Job Description), using the provided screens as reference. You’ll need to get some JSON content, filter, sort and search the data and print it on view.

Although this is a basic exercise, we’ll be looking for simple, well-designed and tested code in the submission. Please be prepared to walk through the code and demonstrate your thought processes in an interview.

Estimated time to complete: 24 Hours

Please attempt the questions to the best of your ability within 24 hours and send back to riley.green@au.ncs-i.com. If you take longer than 24 hours your attempt will be void.

Task

Using the provided screen as only a reference, you’ll need to build a set of components to render the app. You’ll also need to request a JSON feed, filter that data, and use the relevant fields.

We have provided mock movie data as a JSON file. While requesting JSON feed, we also ask you to use Redux (or similar) for storing incoming data. So this stored data will be used when you build the app and the components.

Use the returned data to display a page of results that matches something like the given design.

Please include a README with setup instructions, and any tests or other documentation you created as part of your solution.

We also recommend you to use create-react-app in order not to make basic configuration yourself.

You can use frameworks or packages as long as you can explain to us why you chose them.

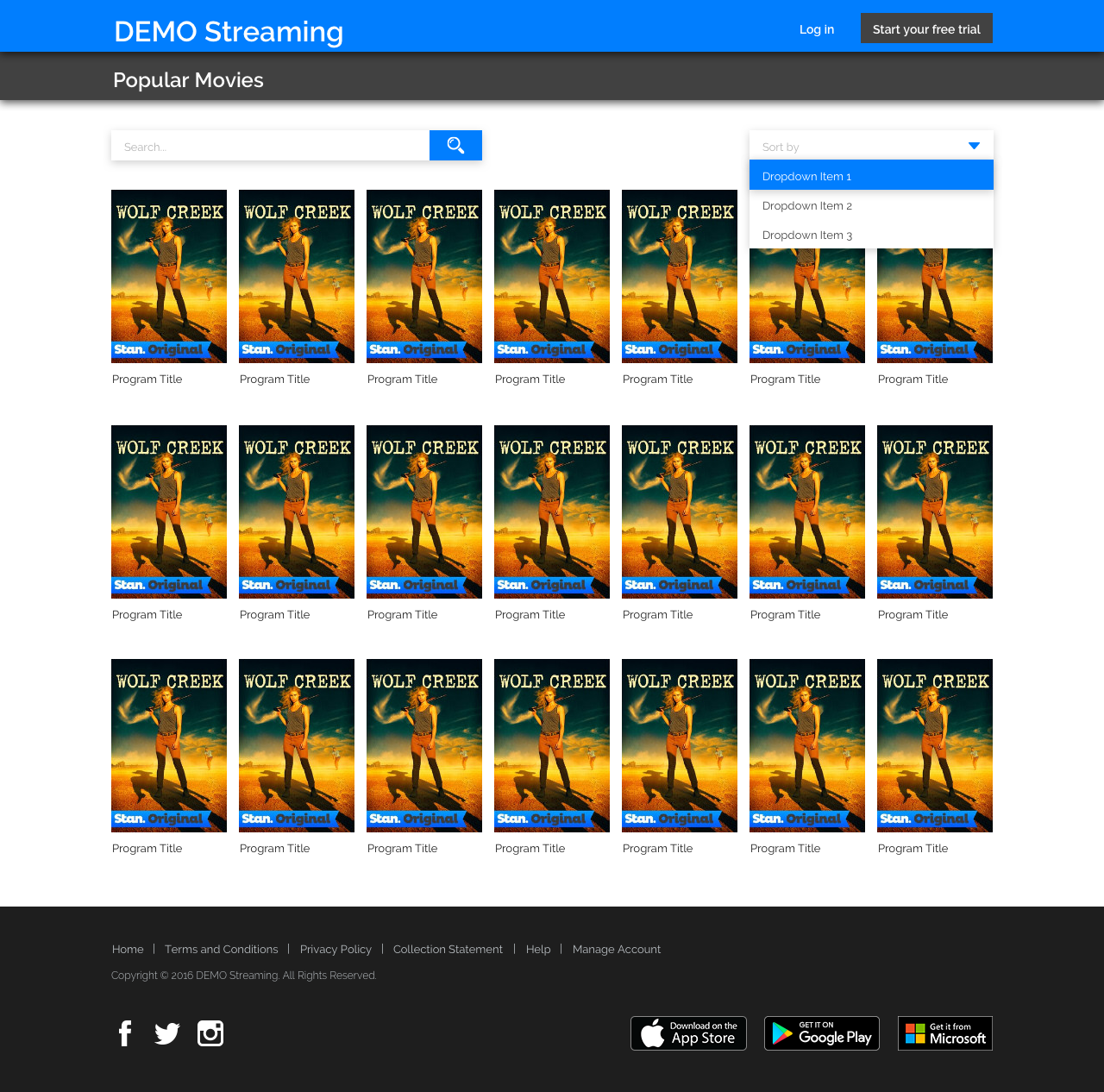
Design Example

Figure 1: Sample Design

Please use this as an example, however don’t feel the need to copy it completely. Get Creative!

There's no screens designed for the mobile version, but if you build a responsive app it's a big plus.

We encourage you to use latest CSS techniques (like flex-box) and responsive design approach.

Details

You will need to build the Home page, similar pages for only Series and Movies are optional.

Home Page

This will be your index.html screen. For each page you will need to fetch JSON feed in feed/sample.json, then:

* Display the first XX entries.
* Where the entry has a releaseYear attribute value >= 2010.
* Sorted by the title attribute value in ascending alphanumeric order.

For the Search function:

* It should filter the title attribute value of movies and series
* It should start to filter after typing 3 characters.

For Movies and Series pages there will be a dropdown list which has options of:

* Sort by year in descending order.
* Sort by year in ascending order.
* Sort by title in descending order.
* Sort by title in ascending order.

For the movie type filter:

* Where the entry has a programType attribute value of series
* Where the entry has a programType attribute value of movie

The attributes you should use to display the entries are:

* title
* images → Poster Art → url
* Anything else you desire.

You will also need to handle the loading and error states, of fetching the JSON feed:

* Check the design file for loading and error screens.
* Please create components for each part of the page (eg. header, content, footer, etc).

You can assume that you do not have to support legacy browsers without features such as fetch or flexbox.

Nice to have

These are not obligatory but big plus if you implement. Here's few suggestions of things we'd like to see:

* Implement solution in TypeScript.
* Responsive design for mobile device.
* Showing your work through your Git commit history.
* Production grade code (clean, maintainable, reusable code)
* Implement unit tests with Jest & Enzyme or react-rest-renderer.
* Prefer to use a css methodology, such as BEM.

Submission Guidelines

Please submit your code by sending a Github repository link.

Project evaluation

We are evaluating project according to the features which specified table at the below.

Base requirements

1. HTML Mark-up
2. React & State Management
3. Production grade & Best practices
4. Project folder structure

You will get extra points if have implement bonus features.

Bonus Points

1. Typescript
2. Next.js
3. Git commit history
4. Unit Tests