


Challenge Statement

In this exercise you need to create a dynamic product page that lists some products. In a real world scenario the list would be given to you by a back-end service, but for the purpose of this exercise we provide you with a sample response in **seed.js** in the accompanying .zip file.

Here is how the page should look like:


Popular Products




▲ 55

Haught or Naught

High-minded or absent-minded? You decide.


Submitted by: 




▲ 44

Yellow Pail

On-demand sand castle construction expertise.


Submitted by: 




▲ 42

Tinfoild: Tailored tinfoil hats

We already have your measurements and shipping address.


Submitted by: 



▲ 23

Supermajority: The Fantasy Congress League

Earn points when your favorite politicians pass legislation.

Submitted by: 

The data for each product entry looks like this:

```

{
  id: 1, // Unique product ID
  title: 'Yellow Pail',
  description: 'On-demand sand castle construction expertise.',
  url: '#', // The URL is where the 'title' element points to.
  votes: generateVoteCount(), // This function is provided for you in seed.js.
  submitterAvatarUrl: 'images/avatars/daniel.jpg', // Appears on the right of
                                                    // 'Submitted by:'
  productImageUrl: 'images/products/image-aqua.png', // Appears on the left side
}

```

In your implementation:

- The layout should look as similar to the provided example as possible.
- The title of each product should be a link.
- The products are ordered from top to bottom by highest-to-lowest vote numbers.
- When one clicks the ▲ button of a product then the number of votes increases by 1. If the number of votes of the product becomes greater than the number of votes of the product above it then the product must move up so that the order is maintained on the screen.
- You may use any UI frameworks you wish (ex. React / Vue / Angular / ...)
- You may use any CSS library you wish. We don't require things to look "pretty".
- You may use any language you wish, so long as it transpiles down to JS during 'npm run build', and 'npm run test' runs as expected (see below). However if it is in Javascript or Typescript, it is preferable to us.
- Your project should build with npm (node v6+, npm v3+):
 - 'npm install' should install all required dependencies
 - 'npm run build' should build a browser-friendly sets of assets under a build/ directory with an index.html and all necessary javascript and image assets such that if one opens index.html in their browser they can see a working page.
 - 'npm run test' should test your code (We don't expect 100% code coverage. Something reasonable that demonstrates your ability to unit test your work is enough).

Assets

You are given a .zip file that contains all the assets you'll need for this exercise

seed.js: contains functions utility functions to generate vote counts and return a list of products to render.

images/ : contains all the image assets required which may be copied locally into your project.

What to return back to us

1. Include details which explain your UI component design decisions in a README file.
2. Please zip everything in a directory named `firstname.lastname/` and return via email.
3. In your email response please let us know roughly how many hours you spent for this exercise.