

# Tune Code Challenge: Front End Engineer

---

This is a project to show off your coding abilities for the Tune front end team.

What we expect from you during this:

- An app with a great user experience
- Clean, readable code. (CSS Optional)

The instructions below describe what is required for a minimum viable product. Feel free to add any features you want. You will not be graded poorly for a failure to add additional features, but this is a chance for you to stand out.

And yes, we intentionally made the project look like a sharepoint/dreamweaver site from 2004. It's called retro.. deal with it.

## The Project

A small e-commerce front end that consists of a product list page, a product detail page, and a shopping cart review page.

You will be provided with the following:

- An html with example markup for the interface
- A stylesheet that will make things look purty.
- A simple REST API with access to the products library.
  - NOTE: The API will return random data after each restart.

You may complete the project using any framework that you would like. The only stipulation is that the application is expected to be a single page javascript application that runs in the newest versions of Firefox and Chrome.

## Getting started

---

### You will need

- Node.js and npm installed.

### Instructions

- Unzip the project
- run `npm install` in the project root.

### Running it

- run `npm start` from the command line. Note: This will generate a new data set each time you start the app.
- 

## Files

In `./site_files/` you are provided with HTML and CSS for the basic look and feel of the site. You can use these directly or copy as you need.

The data is generated from `./server_data/data_generator.js` and uses the `faker.js` api. Feel free to add any additional columns or fields you might want.

## The API

The API will expose the following routes on GET requests:

`/products/` => A list of all products.

`/products/{id}` => The detail for a single product

Further API documentation can be found at <https://www.npmjs.com/package/json-server>

## The Cart

The cart is not persisted to any remote source. You may do this if you would like to. In memory works fine for this project.

## Submitting your results

Please zip up your results and send them to your recruiting contact.

Do NOT commit your results to any public repository such as GitHub or BitBucket.