Final Sprint

React, HTML, CSS, and JavaScript.

Hello all, and welcome to the final sprint. For this sprint, you can work in teams of up to three people. For team preferences, please let Nicholas know as soon as possible who your preferred teammates are in the provided excel document – if you’d prefer to be placed in a random team of three, you can also let us know, and we’ll do that for you as well. If we don’t hear from you, we’ll assume you want to do the sprint individually.

For this sprint, you have two options: firstly, you can work on a project of your own design if you’d like, or alternatively you can implement the project that we’ve designed for you. If you’d like to work on your own idea, please send a written proposal for the project to Nicholas in a private message. We will review your proposal to help you scope the project and make sure that you don’t take on too much given the time that we have available.

Note there is also an individual component of this project described at the end of this document that all students must complete on their own.

# Project Requirements

1. Have a Figma design done up for your project following the design principles that Levin taught you in UI/UX. (Generally speaking – a reasonable user experience, with some thought given to the aesthetics of the page)
2. Implement the project in React
3. Use proper Semantic HTML tags where applicable
4. Comment your code for clear representation of its purpose
5. CSS and React code that approximates the design in the Figma mockup.
6. Must use:
   1. React Router (A minimum of 4 pages/components)
   2. Units test for minimum of 3 components
   3. Either use a free API for testing (*like reqres.in, api.github.com/users etc.*) OR set up your own Json server and try to show basic CRUD operations. (*as we did in our class codes*).
   4. Create a custom hook for fetching data from Json server or any API endpoint.
7. Upload the finished project to GitHub

Beyond that, the world is your oyster.

Be creative and do what you think would be fun and interesting.

If your team can't think of a project, I have a website here with a few project ideas.  
<https://www.freecodecamp.org/news/5-react-projects-you-need-in-your-portfolio/>

Some of those projects are a little over-scoped for us, so if you want to choose one from that site, let me know what features you intend to implement ahead of time so I can help you from getting in over your head.

Replications of popular apps are always good as well if one strikes the fancy of you and your team.

# Default Project

If you’d prefer not to think of your own project, no worries, we have one designed for you here.

***[The following should serve as a reference for minimum requirements should you opt to work on your own idea]***

For this project, we're going to make a simple e-commerce website with features like product listing, product details, shopping cart, and checkout. Also, implement a fake API or use a mock server to simulate data interactions. Moreover, write tests to ensure the proper functioning of the shopping cart, product list, and product details components.

The design can follow inspiration from any ecommerce app/website of your choosing as many of them follow the same philosophies (you may find inspiration looking at amazon, eBay, Alibaba etc.). The canonical version of this app has at least four screens/pages/components, described below:

1. ProductList.jsx
   * This component will display a list of products after getting it from the mock API or Json Server (your choice)
2. ProductDetails.jsx
   * This component will display the details of a selected product again after getting it from mock API.
3. ShoppingCart.jsx
   * This component will display the items added to the shopping cart by getting it from the context file “ShoppingCartContext”
4. Checkout.jsx
   * This component will handle the checkout process by using the context file.

The following are the supportive files required to make it work properly:

1. api.js

Here is an example api.js code that you might want to use in your application:

// api.js

const products = [

{ id: 1, name: 'Product 1', description: 'Description of Product 1', price: 10 },

{ id: 2, name: 'Product 2', description: 'Description of Product 2', price: 20 },

{ id: 3, name: 'Product 3', description: 'Description of Product 3', price: 30 },

];

export const getProducts = () => products;

export const getProductById = (productId) =>

products.find((product) => product.id === productId);

1. ShoppingCartContext
   1. Create a context to manage the shopping cart state and actions. This might include returning of “cartItems, addToCart(), and removeFromCart()”.

In addition to above files, you would like to have a \_\_test\_\_ folder somewhere in your submission that has at least 3 unit tests for any of the given components.

# Submission:

1. Submit only the github link to your project, with following two extra objects:
   1. Video: include a “ONE MINUTE” video, just showing the demo run of your project – no need show or explain the code. (I will do it from github 😊)
   2. For this group project, a brief outline of the contributions made by each group member. Don’t include micro information, just a brief representation of everyone’s effort.

# Individual Project – (may or may not be a React project)

In addition to the main project, which is (optionally) group-based, all students **must** make a portfolio website for themselves that showcases your projects and skills. This website should, at a minimum, feature some dynamic JavaScript (for example, DOM manipulation, or even something more involved, like an interactive widget - one student made a little game that was embedded in their site once to fulfil this requirement – pretty much anything will work, if you’re not sure, ask Nicholas – , feature good use of semantic HTML tags, and should approximate a Figma design that must also be submitted. Most students take this portfolio as a base and grow it over time throughout the rest of their studies, so try to make it as stylish as you can, it can act as sort of a living resume for you. Many people will submit just a GitHub profile when they’re applying for jobs, but with a portfolio page, you can showcase the code on your GitHub profile – you can take pictures or videos that show it running in optimal, controlled conditions, and you can tell a story behind each project, painting your work in the best possible light. It’s really a curated experience of your work for employers. If you want some examples of portfolio websites,see:  
<https://fourandthree.com/> (Levin’s Website)

<https://www.awwwards.com/websites/portfolio/> (A collection of examples.)

# Submission:

1. Submit your individual github link to your project, with following:
   1. Video: include a “ONE MINUTE” video, just showing the demo run of your project – no need show or explain the code. (I will do it from github 😊)