

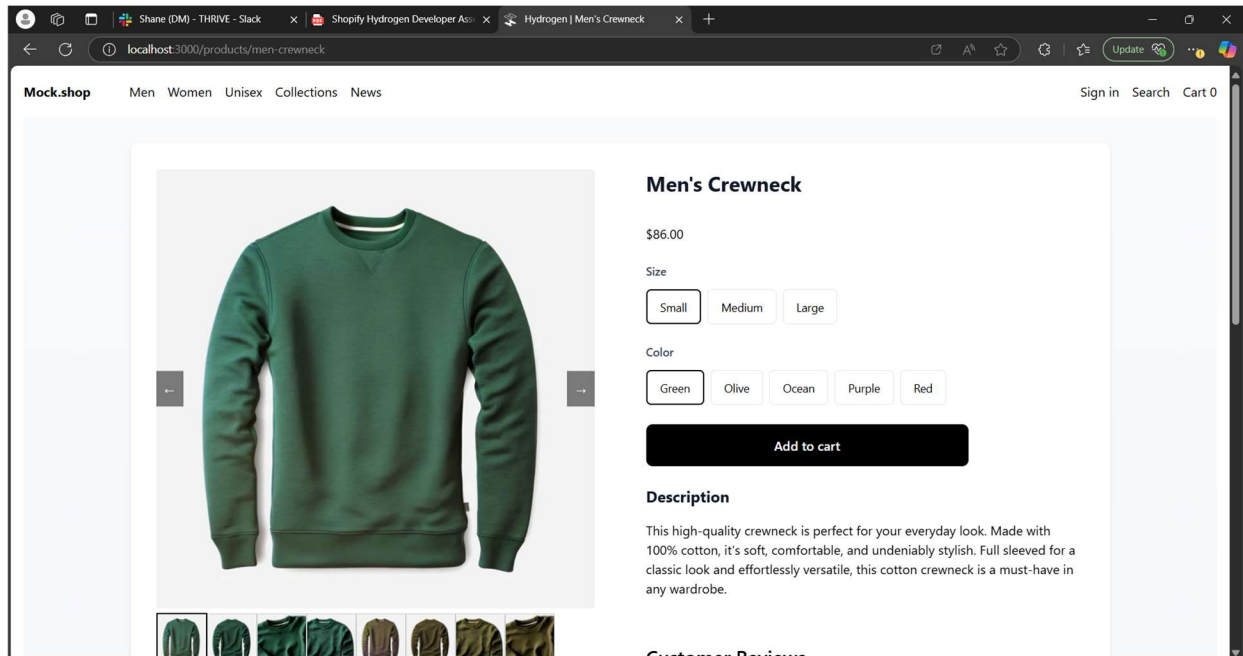
By Vishal Jaimin Vakil

GitHub: <https://github.com/VishalVakil/hydrogen-storefront>

Screenshots:

(Note: The below images are all without the theme since I was not able to link the theme to the deployed storefront due to the partner staff account access limitation on Shopify)

1. Product Details Page – Desktop:



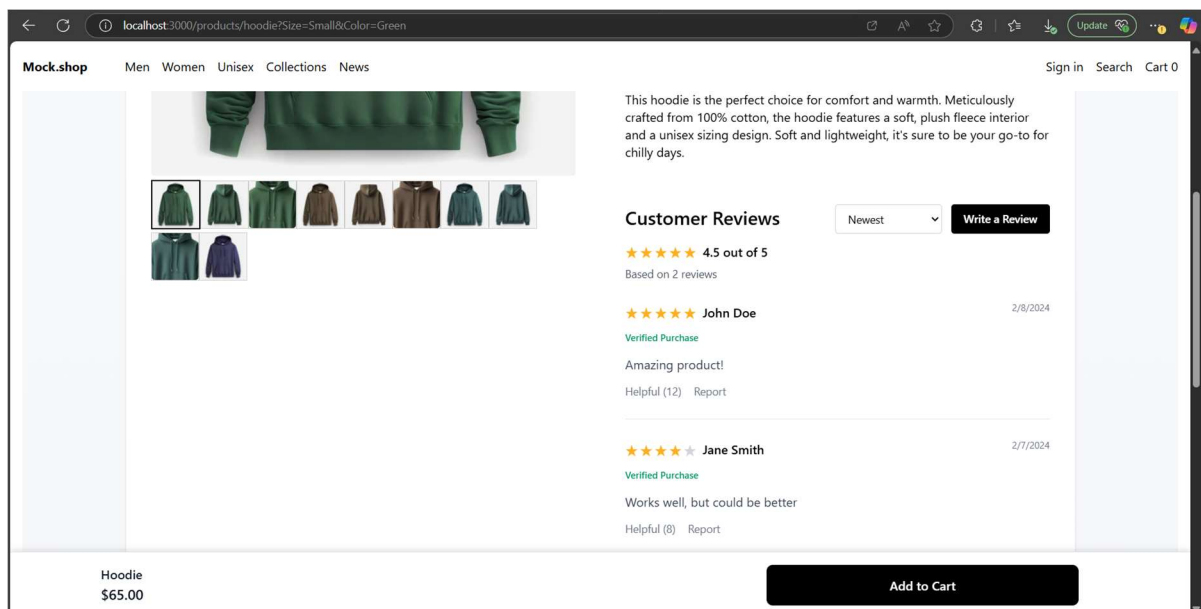
2. An interactive product image gallery with a carousel functionality – Desktop:



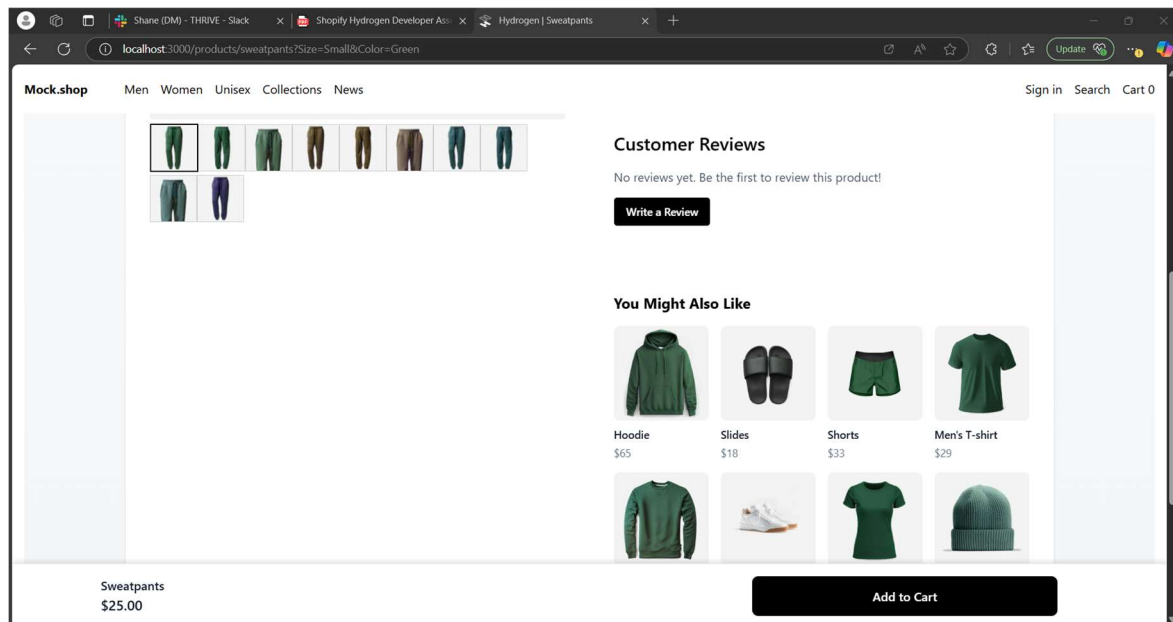
3. An interactive product image gallery with a zoom functionality on hover – Desktop:



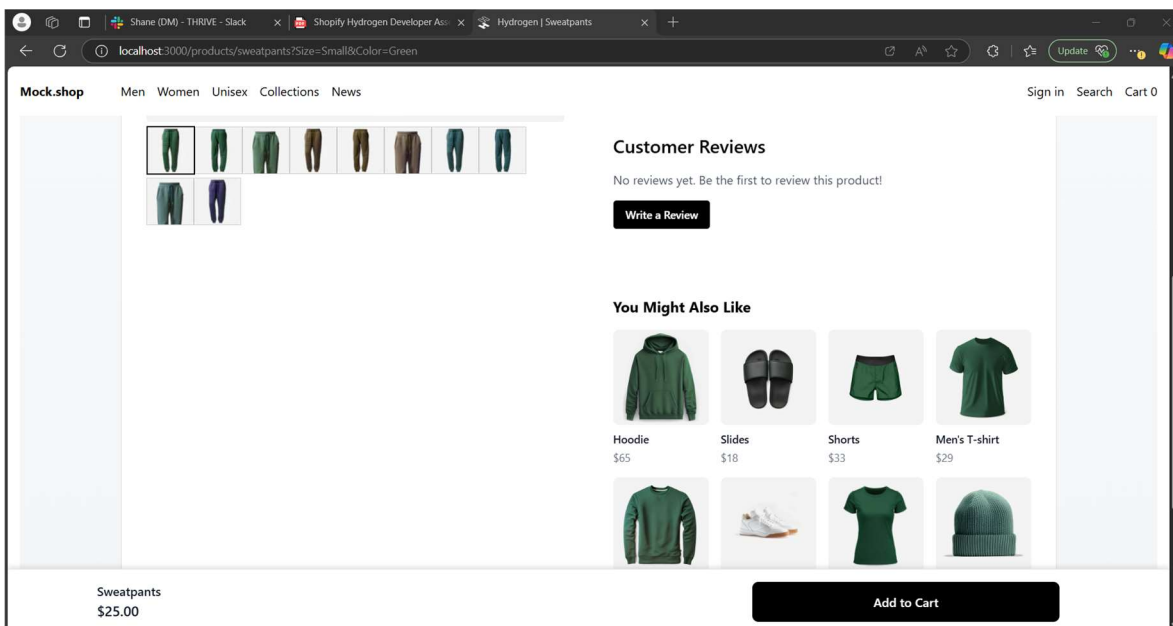
4. A Reviews section – Desktop: (Also the sticky Add to cart can be observed here)



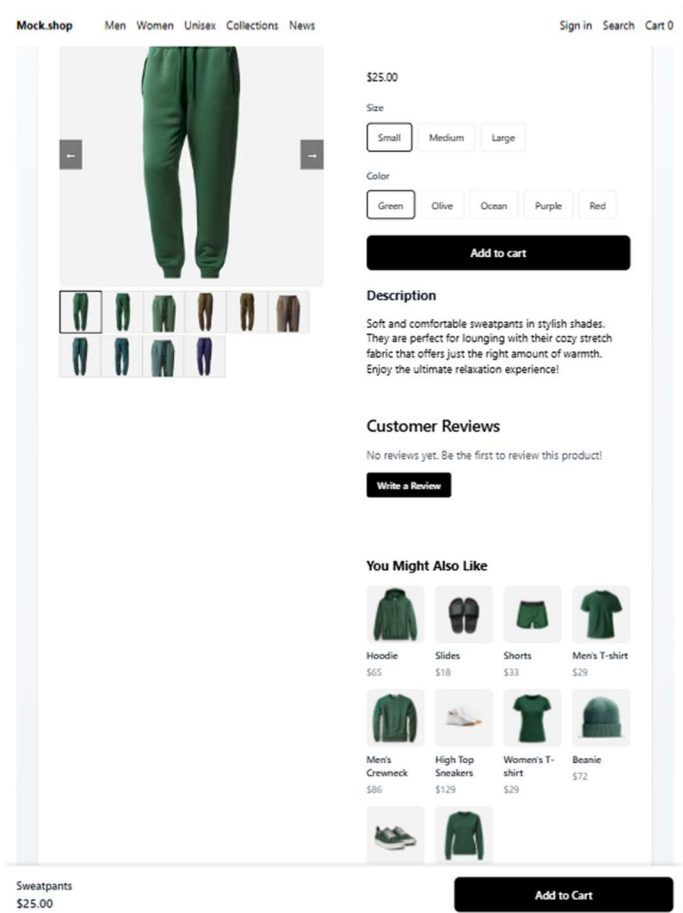
5. An AI-Powered Recommendations section – Desktop: (Wasn't able to find the API URL, so I displayed Shopify recommendations. But I created the code for AI-Powered recommendations. The current code would check for both and merge the recommendations. Since the API URL is not available, with just API KEY the AI powered recommendations would appear, so only API URL needs to be mentioned in order to fetch the AI-powered recommendations.)



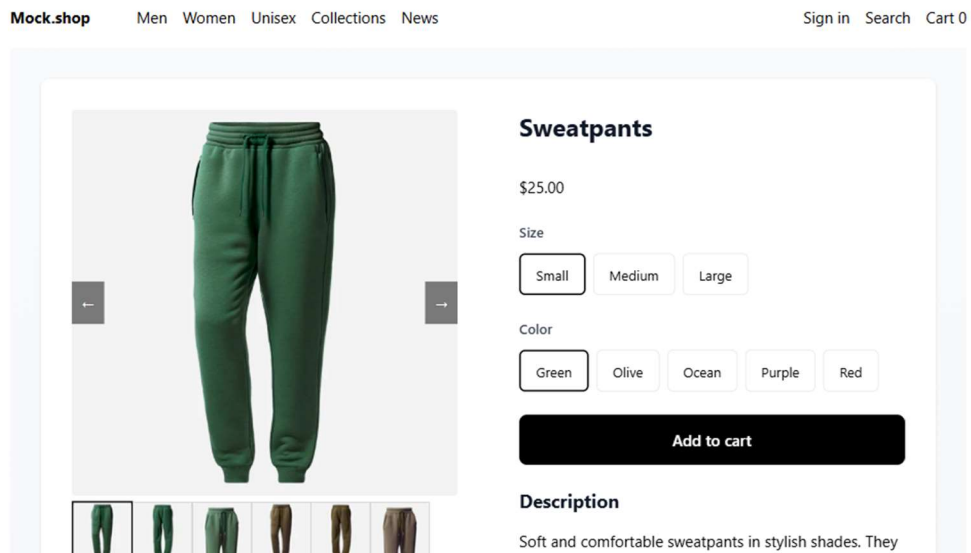
6. Sticky Add to cart – Desktop: (Bonus Task) – If scrolled more than 30px it will appear



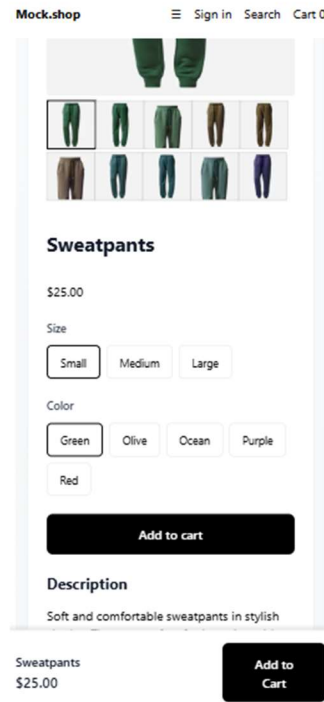
7. iPad Pro (Scrolled to show Sticky bottom bar as well):



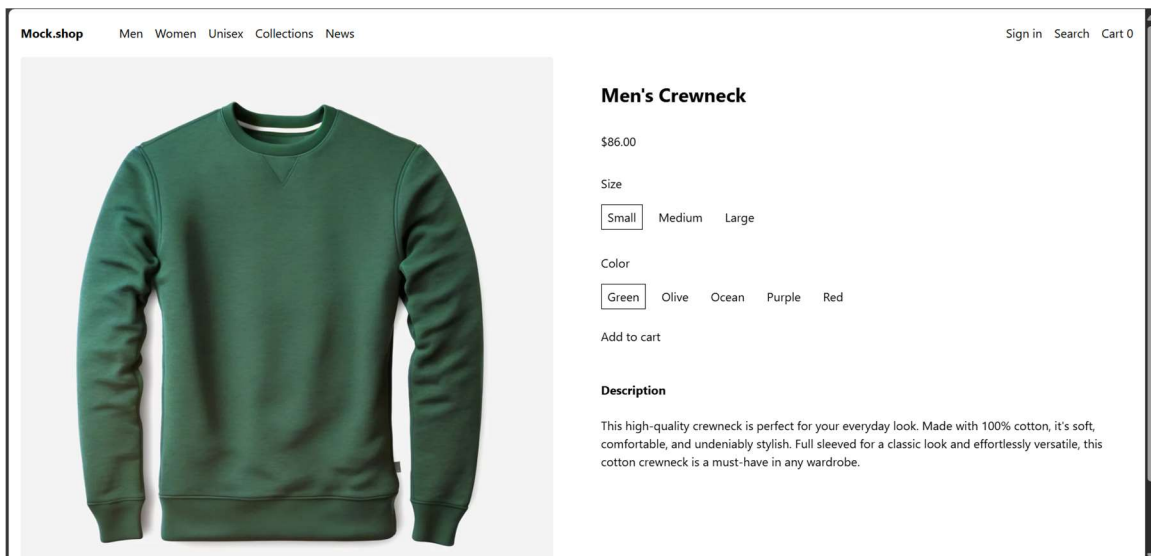
8. Nest Hub:

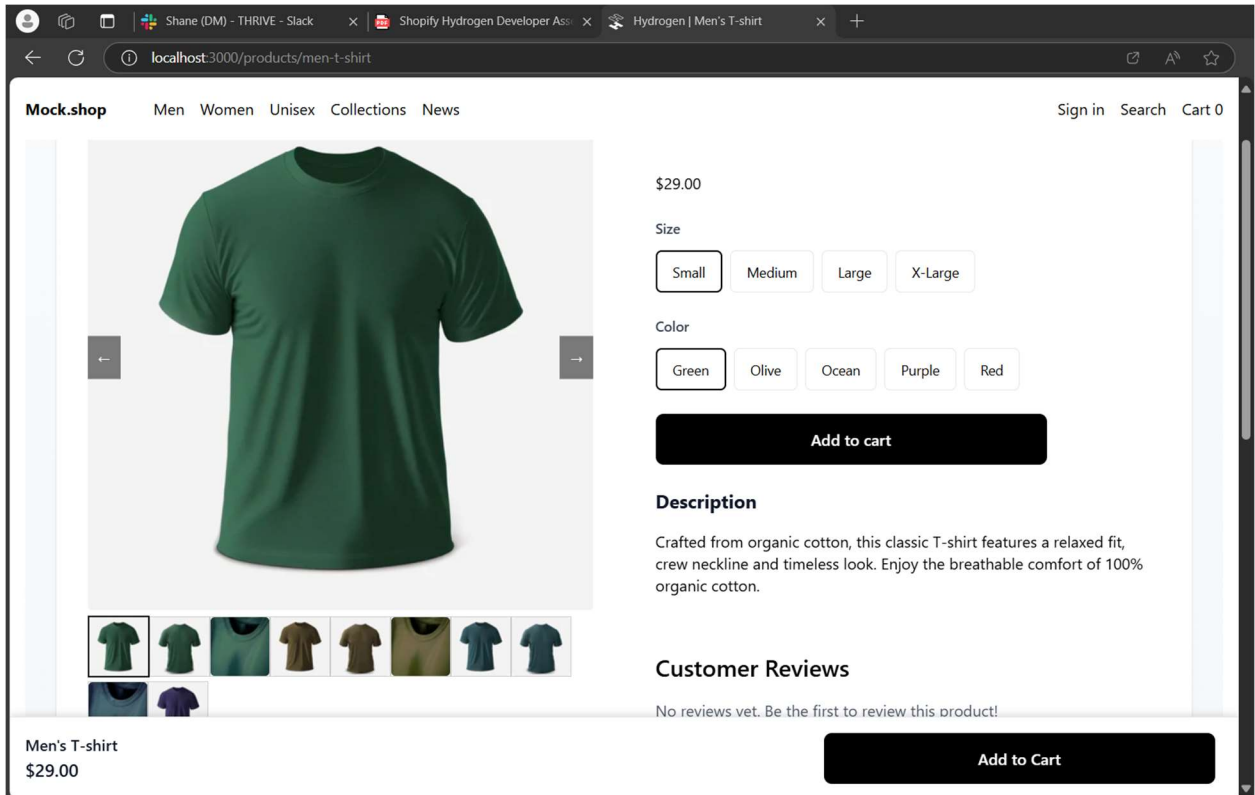


9. iPhone 14 Pro Max (Scrolled to show Stickybar):



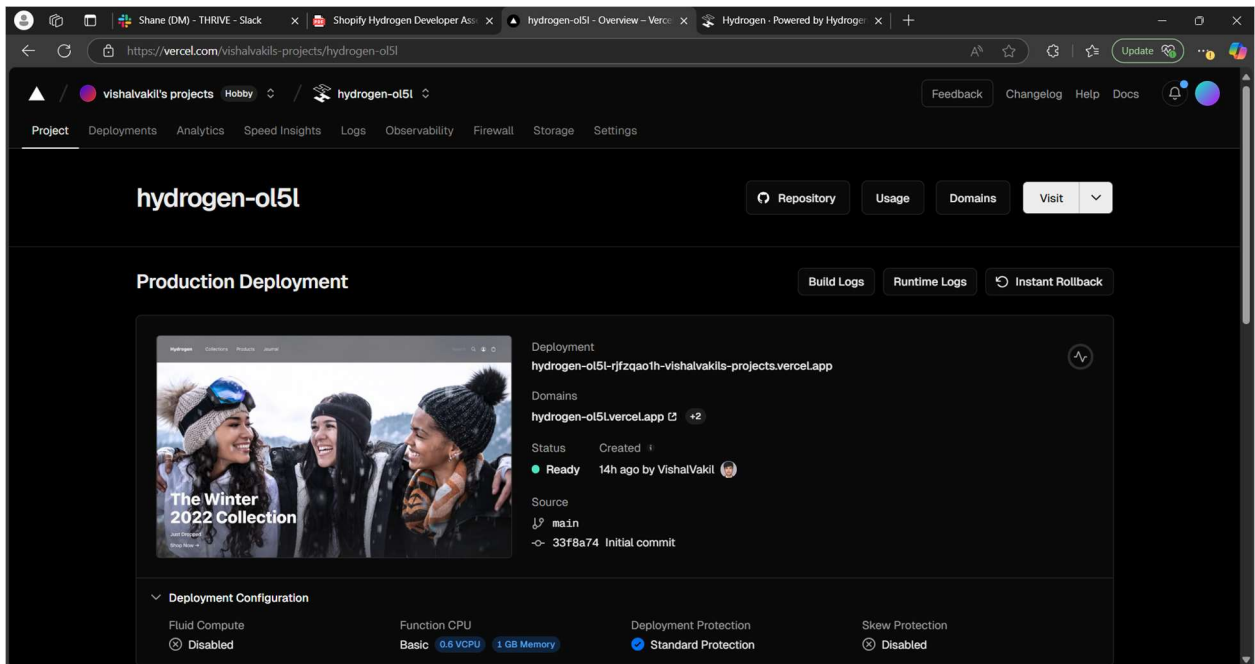
10. Before/After:



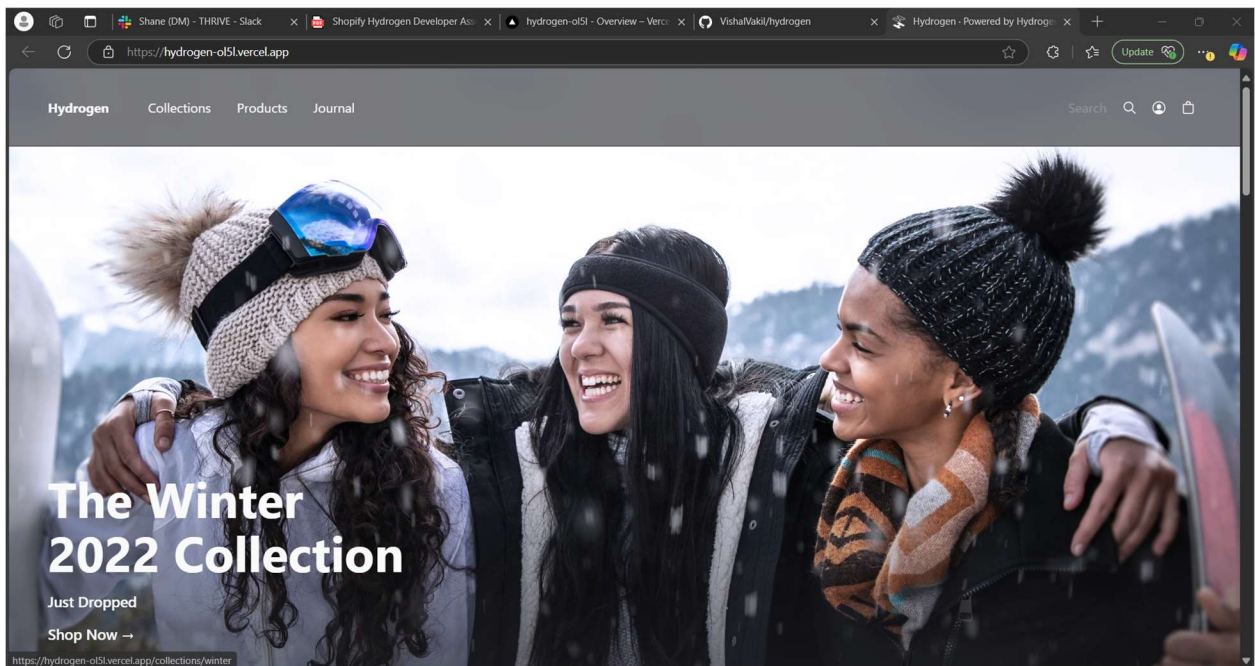


11. Deployment on Vercel:

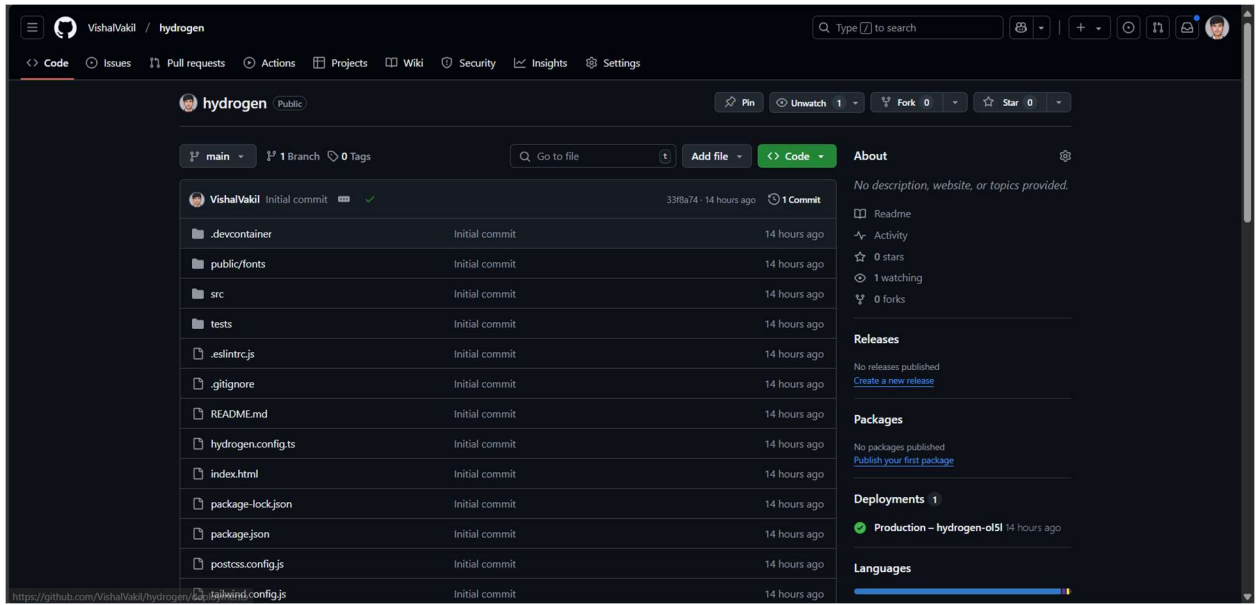
Vercel supported the deployment of hydrogen apps from the legacy version. I deploying our hydrogen version on the Vercel serverless platform but it kept failing. Upon troubleshooting, I found that Vercel's build output doesn't match with our hydrogen's output which is dist. Hence, diving deep into the issue, I found recently Vercel has rolled out zero configuration deployment for hydrogen legacy version apps. Since our version is not supported, I deployed the legacy version on Vercel to showcase the ability of deploying Hydrogen Apps.



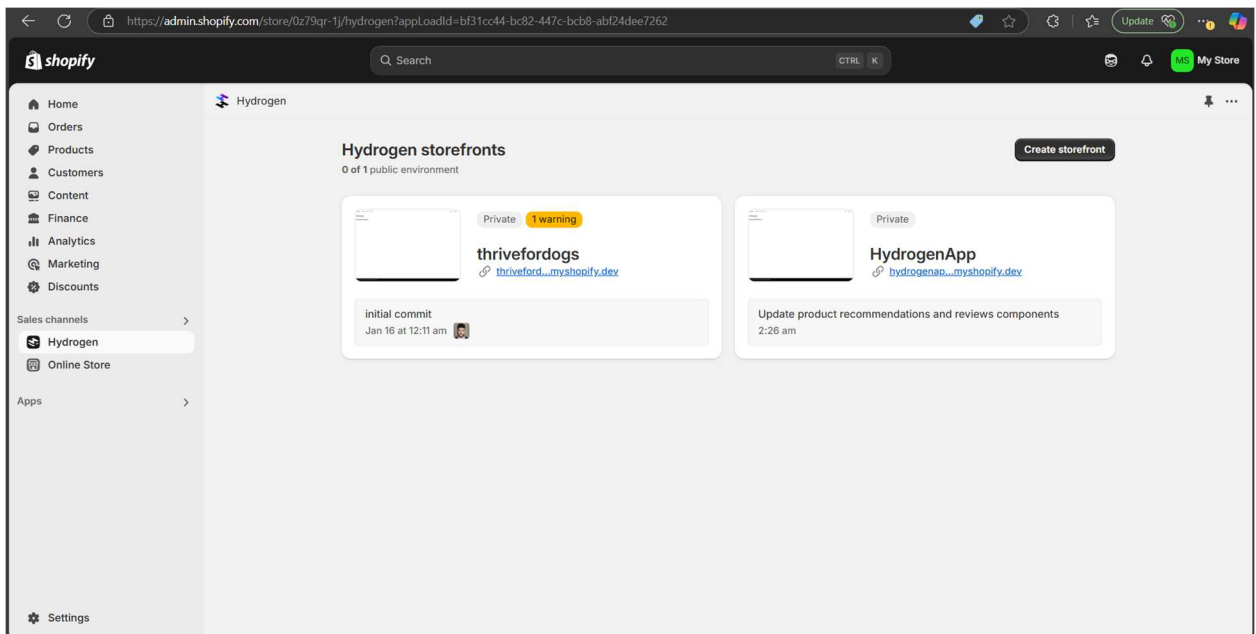
Live URL: <https://hydrogen-ol5l.vercel.app/> (Note: This is not the one that I edited since our version is different, I deployed it on Oxygen only. This is a demo deployment of a Hydrogen Legacy version which is supported by the Vercel serverless platform)

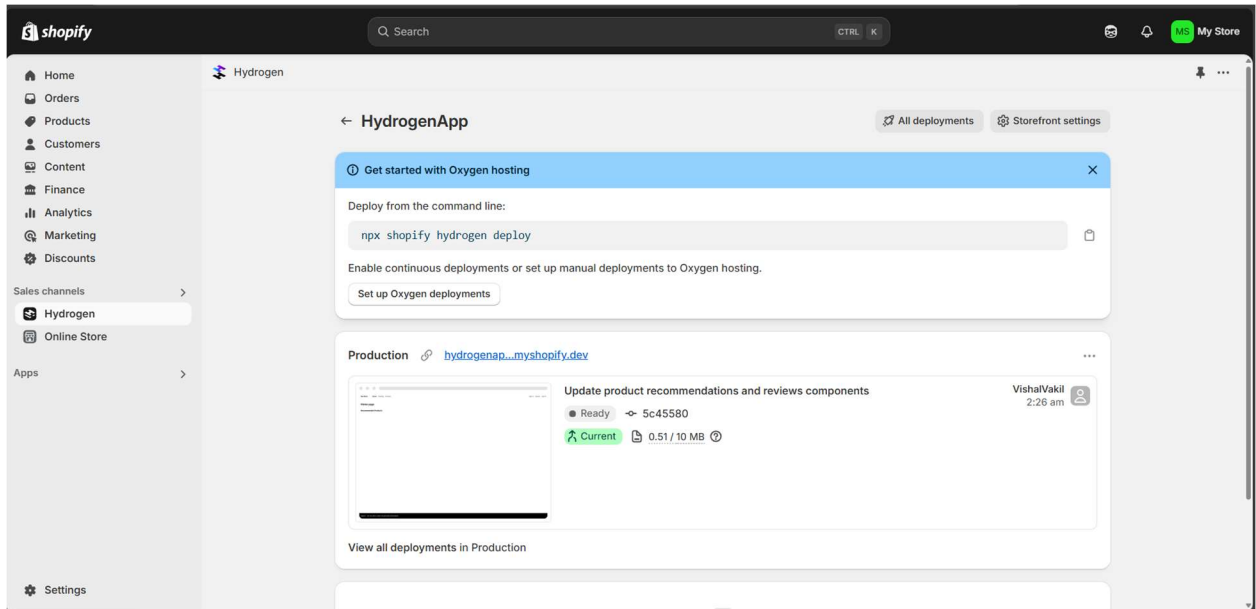


Repository I used to Deploy this Hydrogen legacy version:
<https://github.com/VishalVakil/hydrogen>



12. Deployment of Hydrogen Storefront on Oxygen:





URL: <https://hydrogenapp-b2d4aa8a7dcacf141765.o2.myshopify.dev/>

(Note: Shopify APIs are being used here so the mock data won't appear here.)

Theme Implementation:

1. On the local machine, I downloaded the Broadcast Bold theme from the Shopify store.
2. I understood structure and what kind of files are present inside the theme folder and then I created 3 files inside the below directories:

- a. sections/product-template.liquid
- b. templates/product.liquid
- c. snippets/product-form.liquid

3. Moved the folder inside the Hydrogen Storefront local folder to make sure I can track the files.

4. Navigated to the theme directory.

5. shopify theme dev --store HydrogenApp (HydrogenApp was the storefront created by me)
(Also I was logged into the Shopify Account)

6. Expected:

The theme should have been applied to the local store. But due to partner staff account access, I was unable to see the theme on local machine.

```
You are not authorized to edit themes on "HydrogenApp.myshopify.com".
```

```
You can't use Shopify CLI with development stores if you only have Partner staff member access. If you want to use Shopify CLI to work on a development store, then you should be the store owner or create a staff account on the store.
```

```
If you're the store owner, then you need to log in to the store directly using the store URL at least once (for example, using "https://HydrogenApp.myshopify.com/admin") before you log in using Shopify CLI. Logging in to the Shopify admin directly connects the development store with your Shopify login.
```

(Went on to explore the documentation and the point mentioned below is my understanding on the next steps)

7. To apply the theme to the online store:

shopify theme push --unpublished

shopify theme push

shopify theme publish

Approach:

Looking at the requirements, I listed what needs to be done inside the product details page. I understood some concepts of Hydrogen before working on it. Just like React, I figured out I need to create new components to facilitate the Product Gallery that would have carousel/zoom + lazy loading functionality.

Inside the Product details page I called the `ProductGallery.jsx` component and passed the product as the props, to display the product image and the variants. Inside the `ProductGallery.jsx`, I made the gallery for just an array of product images. I created two buttons and made a function to handle the logic of Next and Previous image based on the onclick event. To create the logic, used `currentIndex` to keep track of the index in the images array. Based on the Next or Previous buttons, the index adds or subtracts itself, which then eventually gives `currentImage` from the images array. I took care of error handling as well by first checking if an array of images is available or not with the help of `images.length`. I utilized `useState`, a Hook from React to add state to the functional component. So, whenever the button is clicked the `useState` updates the state of `currentIndex` and re-renders the entire functional component to return the desirable gallery image. Inside the `Image` tag, I added the `loading="lazy"` attribute, which helps in setting the delay in loading.

Next, I started working on the reviews section's requirements. I created a new component `Reviews.jsx` which I called inside the `products.handle.jsx` page. To my notice I found that there isn't any mock data from `mock.shop` on reviews. Hence, I customized a template reviews component to meet the requirements mentioned in the assessment. I customized it by configuring it as per our product details page. I passed the `product.id` and reviews it fetched. I created a mock data for reviews, but also created the facility to fetch it from a real API.

Similarly, I created another component for recommendations and customized it. I wasn't able to find the API URL, so I displayed Shopify recommendations. But I created the code for AI-Powered recommendations. The current code would check for both and merge the recommendations. Since the API URL is not available, with just API KEY the AI powered recommendations would appear, so only API URL needs to be mentioned in order to fetch the AI-powered recommendations. Then I created another component for `StickyAddtoCartBar`. I implemented a sticky "Add to Cart" bar that becomes visible when the user scrolls down 30px on the page. It displays the product title, price, and a dynamically updating "Add to Cart" button based on the selected variant. The bar smoothly appears and disappears, enhancing the user experience.

I also implemented the `<CartProvider>` from `@shopify/hydrogen-react` and updated the `root.jsx`, `AddtoCartButton.jsx` and `StickyAddToCart.jsx`. I used `storefront.CacheLong()` in the loader to help caching and performance. I also included a simple JSON-LD snippet for structured data.

Things done:

1. An interactive product image gallery with a carousel or zoom functionality
2. A variant selector that dynamically updates the price
3. An "Add to Cart" feature implemented with <CartProvider>
4. A reviews section
5. AI-powered product recommendations using the provided test API key as well as Shopify recommendations.
6. Deployed "Hydrogen Legacy" version to Vercel which is different than the storefront I customized. But due to the supported version of Hydrogen on Vercel I had to deploy this mock storefront.
7. Created theme files – Utilized Claude to understand and create the liquid files.
8. Utilized Claude to understand Hydrogen.
9. I went through reviewing the documentations of Hydrogen, Oxygen and Themes.
10. Deployed the customized storefront on Oxygen.
11. Implemented lazy loading for images
12. I used storefront.CacheLong() in the loader to help caching and performance.
13. I also included a simple JSON-LD snippet for structured data.

Difficulties and Trade-offs:

Debugged the Recommendations API, initially the section didn't appear as there weren't any recommendations, later on utilized Shopify recommendations and also kept the AI powered recommendations whenever they are available (Needs API URL). Initially faced difficulties in passing the prop review to the review section where I made minor mistake in comparing reviews to arrays which was Promise. So, debugged using console.log() and fixed the issue.

Did various deployment attempts on Vercel and Fly.io. Created GitHub repository and connected to Vercel. Vercel build output didn't matched to Hydrogen's output dist. To overcome, I created the vercel.json file and configured it and successfully deployed it but still there was an issue of 404 ERROR on the deployment. Due to which the content didn't appeared. Hence, looked into community forums and found to overcome the issues with Hydrogen, Vercle came up with zero config deployment for Hydrogen's legacy version. Hence, I created that version on local which is based on demo-store template and deployed it on Vercel.

I faced difficulty in linking the theme to the storefront I deployed on Oxygen because of the partner staff account access limitations, which I didn't, allow me to link the theme and create a store. Hence, I have kept the store files in the same repository for the review and also, included the customized storefront I created which runs on local. And for deployment I showed the Hydrogen's legacy version demo-store on Vercel.

Conclusion:

I understood Hydrogen, Oxygen and Themes thoroughly. I feel I can still do better to understand the liquid files, but I feel more comfortable with Hydrogen now. Through this assessment, I understood how things are put together. It won't take much time for me to learn about this new platform.