Mariana Giraldo Kevin Penner Anastasiya Vitko Kyle Zhu

[Link to Prototype]
[Slides]

Tools Used

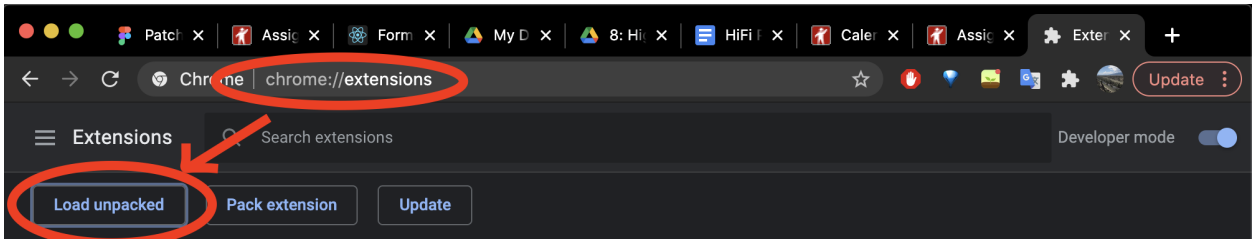
We used Figma for our medium-fidelity prototype, some of which was exported as buttons to make the interface as realistic as possible. Our target platform is Chrome, where *Patch* can be added as an extension, since we realized that this presents a great opportunity to help users interrupt unsustainable shopping habits.

To develop our extension, we used Chrome APIs and React.js (along with HTML and CSS).

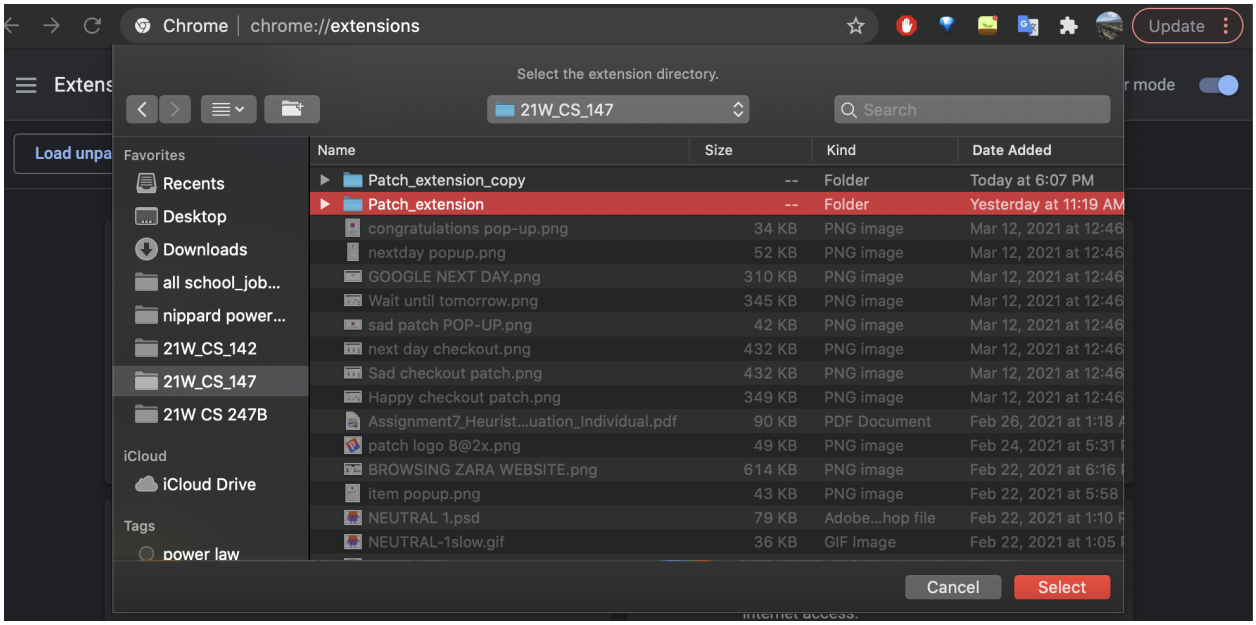
Download Instructions

Google Chrome is needed to run this, since it's a Chrome Extension

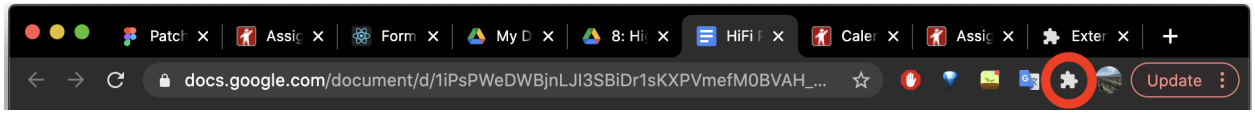
- After downloading and unzipping the folder, go to chrome://extensions



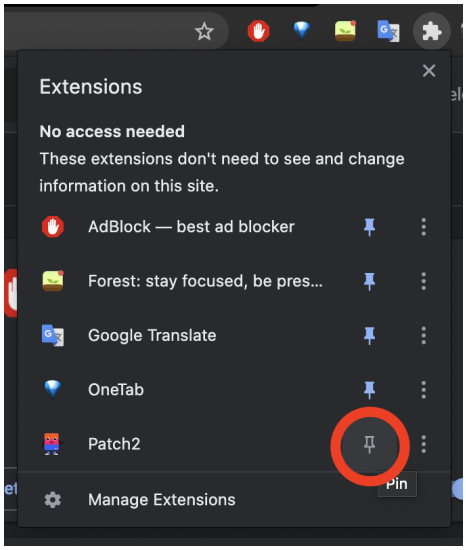
- Click “Load Unpacked” in the top left corner
- Select the folder that you just unzipped; now Patch is installed as a Chrome Extension



- Click the Puzzle piece icon right of the search bar



- Pin *Patch* to be able to use the drop-down menu for social interactions and recommendations



Limitations of Approach

We had to leave out or adjust time-based media (like waiting until the next day to check out). We also focused on the core actions for each task--listening to Patch's interventions, posting questions to friends, getting and following up on recommendations from Patch, and commenting on other users' posts.

Our Chrome extension is designed to intervene at certain points in a user's shopping journey. Since we only intervene on the Zara website at the moment, we would need to build out further domain checking--potentially against a database based on a environmental/ethical rating group like Good On you--to really see the impact of the annoying interventions--both what is effective and at what point the notifications become too much. Because we were mostly focused on the flow and feel of the app, we also don't have any real user interactions, too, as is storing changes made (like comments) or acting on user input for recommendations.

We skipped some unnecessary intermediary screens for achieving our tasks, like loading screens and task-adjacent screens, which we had designed in our final iteration.

In our medium-fi prototype we had buttons in the pop-ups (HTML inserted into the webpage) that would open the drop-down (the page that opens when the Patch icon is clicked). However, after many hours of debugging and trying to find hacky workarounds, we found out that this is impossible. To solve this issue, we would need to embed some of the drop-down functionality into the pop-ups, and then communicate between the pop-ups

and the drop-down via the background script in order for state to be consistent between the two.

Another technical difficulty we have not yet tackled is getting our desired font to work in the Chrome extension, as a result you will see some inconsistencies in font between buttons that were exported from Figma as images, and text native to the extension.

Wizard of Oz and Hard-Coded Items

Recommendation functionality: We currently don't have the ability to evaluate an image of an item and scrape clothing websites to find something similar, or take in user input to do the same.

Trading functionality: This feature requires interaction with another user over a longer time-span than several minutes of testing.

Feed: Again, this feature requires information about past actions of other users on the platform, so all of the posts are just examples.

Screen size: We hardcoded the locations and sizes of the popups, so for the best user experience please use a MacBook Pro with the screen resolution settings set to those specified above. Furthermore, we only combined the pop-up and drop-down functionality at the end of the project, at which point we realized that we were doing development using two different screen resolutions. As a result, when screen resolution is set as specified above, the drop down looks very large. (Feel free to set the resolution to one notch higher to see what we intended the drop down to look like.)

"Wait until tomorrow": We let users navigate to the next day to see interactions due to the complexity and usability in a demo of implementing a time-based reminder

Overlaid on set pages with set information about unsustainability: Even though we can now access page data, we can't intelligently analyze it to figure out why particular pieces are or aren't sustainable. As a result, we use hardcoded examples where we write in why the brand and pieces aren't sustainable.