Week 7

Final Project Abstract: First Person Account

Abstract:

Many people experience sensory overload when using the modern web. Autistic users, people with ADHD, individuals with migraines, people with vestibular disorders, and those with chronic fatigue often browse websites filled with autoplay videos, flashing ads, animated transitions, pop-ups, and constantly updating content. A student may be trying to read an article for class in a quiet library. An employee may be reviewing important policy information at work. A person with a migraine may simply be trying to check their email. In each case, their goal is to focus, read, and process information. Instead, they encounter motion, clutter, and unexpected interruptions that increase stress and cognitive load.

In the first-person account video “Ask An Autistic: Sensory Processing Disorder,” Amythest explains how sensory processing disorder affects the way the brain handles certain input such as light, sound, and movement. She explains that even normal environments can become overwhelming. She explicitly introduces the idea of “brain energy,” saying that sensory triggers throughout the day drain mental energy that would otherwise go toward learning or socializing. Constant stimulation can build up and lead to sensory overload or meltdown. She emphasizes that these reactions are neurological. Her perspective shows that reducing sensory input and having control over the environment is essential for focus and well-being, which directly connects to the purpose of a low-stimulation browser extension.

In response, we propose building a low-stimulation Chrome browser extension that reduces motion, visual clutter, and dynamic distractions across websites. The extension will disable non-essential animations, pause autoplay media, reduce visual noise, simplify layouts, and allow users to toggle sensory-reduction levels. The goal is to create a browser layer that supports sensory regulation and sustained focus, giving users control over their digital environment rather than forcing them to adapt to overstimulating design.

First Person Source:

Amythest Schaber:
Ask an Autistic #9 - What is Sensory Processing Disorder?

First Person Reflection:

- What first person account did you find and does it meet the requirements for a first person account described above
- What are the barriers and opportunities the person described?
- What technology did they describe using?
- How might what you learned extend beyond this specific person, disability and/or technology?

The account we found comes from Amythest Schaber youtube series Ask an Autistic, the episode titled “what is sensory processing disorder”. Amythest is an autistic person speaking from direct lived experience about their sensory processing differences. We thought this met the requirement of a first person account because they are describing how sensory processing affects their daily life and how it feels in their body and mind, as well as what they personally do to cope with it.
One of the most important barriers Amythest talks about is how overwhelming the world can feel when the sensory stimuli is too much like it's too loud, too bright, too fast or even too constant. They explain that the world is “so loud and so fast and so bright” and that negative sensory stimuli is not just annoying but often can feel physically painful. They also talked about how background noise makes it hard to process information and repeated sensory triggers drain “brain energy” throughout the day. Without accommodations the overload can lead to meltdowns, exhaustion and difficulty focusing or communication. Amythest also highlights opportunities through accommodation and self regulation. They talked about carrying earplugs, sunglasses, gum and other items to help reduce negative sensory stimuli and regain control. This shows that small environmental adjustments can make a big difference.

The technology Amythest is mostly low tech and physical rather than digital. Amythest personal sensory kit includes earplugs to help with overwhelming auditory input, sunglasses and hat to manage flickering lights and bright environments, wet wipes to address unexpected discomfort, and gum for deep pressure sensory. None of these are high end assistive technology, they are more simple, easily findable items created to manage environments that are not designed to help.
What stood out is that the “brain energy” draining and need for sensory control is not unique but a universal experience for most people. Because of ADHD, migraines, anxiety or even being tired, many people experience cognitive overload when they are in environments that are too loud, bright, even distracting. So sometimes when people are trying to conserve their last brain energy in an overwhelming space it can apply to anyone. This suggests that devices that can help minimize sensory stimuli, including digital tools like a low stimulation browser extension can help a number of people try to regain control.

Additional Resources:

Feedback:

- **What does it look like more than reading? What is important to change?**
- **Centring the text only on the page**
- **Removing ads/ change color**
- **Updating the css of any page**
- **What does it look like letting a person dictate what is removed and what is helpful and make it accessible?**
-

Week 8

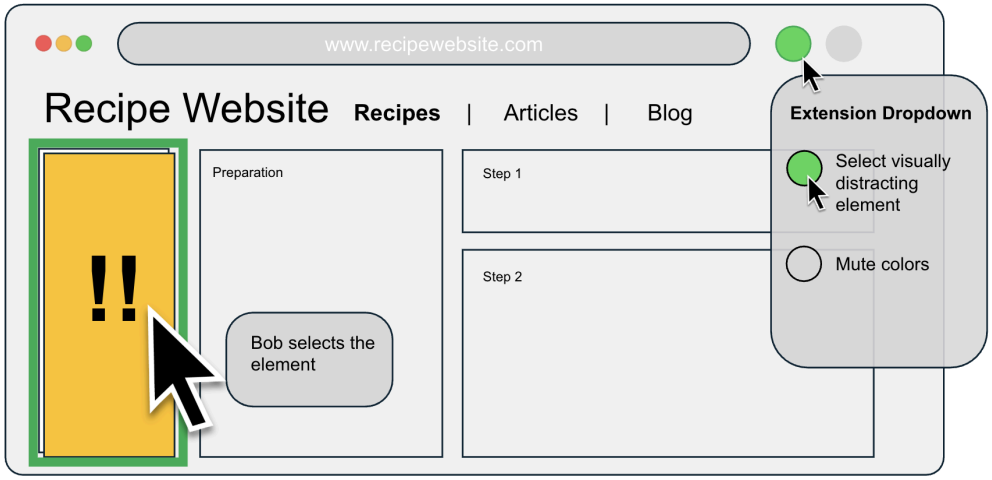
(Week: 8) Storyboard / Timeline

Storyboard

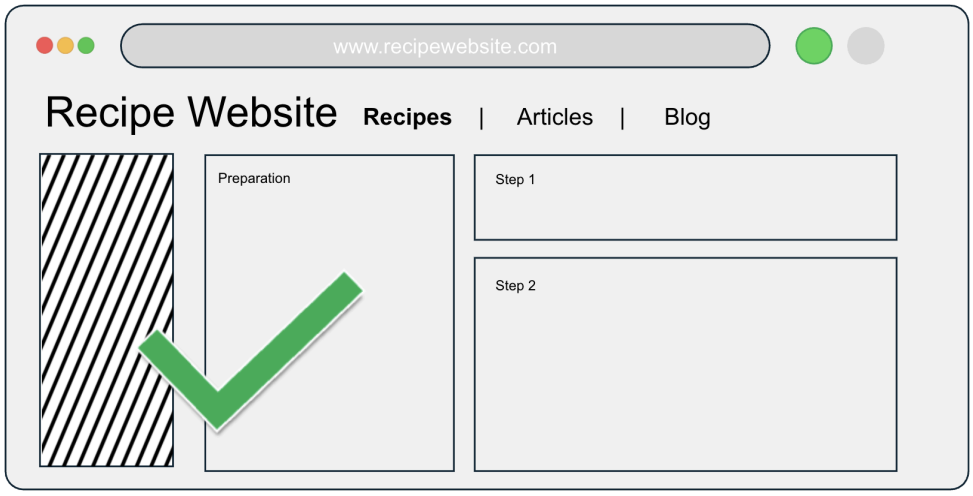
Link to pptx: 443 - Storyboard.pptx

Storyboard 1

Bob is trying to follow a recipe, but a flashing sidebar ad keeps pulling his attention away.

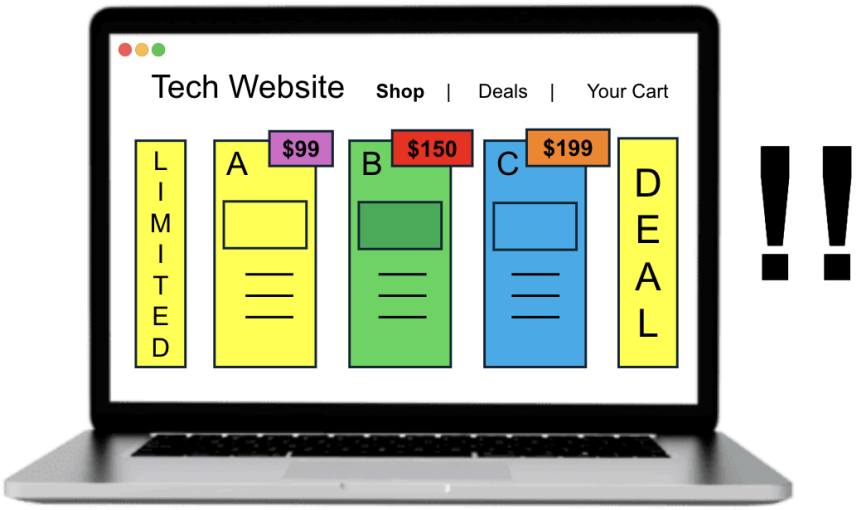


Bob activates extension and selects visually distracting element



Visually distracting element gets covered/removed when page is re-rendered

Storyboard 2



Bob is trying to compare products, but the page is full of bright sale badges, neon banners, and flashy category colors.

Bob opens the extension and turns on “Mute Colors” to reduce visual intensity.

The page stays readable but less intense, so Bob can compare specs and choose the right product.

Disability Analysis (Group)

Conduct a disability model analysis

- Identify at least two disability justice principles that your proposed system does or does not meet
- Consider what ableist assumptions it makes or doesn't make. For example, does it increase control and agency?
- Consider disability leadership. What are you missing by only using first person accounts?
- Who are you including? Who are you not? Identify at least one group that you are including, and one you are leaving out or not considering.

Timeline (Group)

Your timeline should include *what* you will do and *who* will do each task.

- What will you implement by milestone one (week 9)?
- How will you validate your work (week 10)? This should include
    - an accessibility audit
    - a correctness analysis, which is going to be specific to what you are proposing.
- What will you do by the final presentation (include required work such as poster preparation)

The goal we have for week 9 is to have a working prototype of the Chrome extension with most of the important features implemented. We want to include the setting up the extension structure with either a manifest or package file and content script implementing a motion reduction, such as disabling CSS animation and auto play media. Adding a basic layout simplification such as removing ads, centraling text, reducing visual clutter. Building a simple popup UI that lets users toggle between features on and off. We will divide this work across the four of us, one person will own the extension setup and architecture, one will handle motion/animation reduction, one will work on layout and CSS modifications, and one will build the popup toggle interface.

For our accessibility audit we will run the extension against WCAG guidelines using tools like axe DevTools, Lighthouse or WAVE to make sure our own interface is accessible and manually test with keyboard navigation and a screen reader. We will also test the extension on a range of commonly used websites like news sites, google or social media to check that our css modifications do not break page functionality or readability.

For our correctness analysis we want to try on the success based on whether the extension works as its intended without breaking the page usability. If autoplay media is paused and CSS animations and transitions are turned off when motion suppression is turned on, the functionality will be considered functional. If ads and other visual elements are removed but the main text is still readable and centered, layout simplification will be considered good. Also we will make sure that toggle controls work smoothly, which means that each feature may be turned on and off consistently and that modifications are made correctly.At least five different websites will be used to evaluate these requirements, and we will record what works, what doesn't and any edge cases or layout conflicts we come across.

For the final week, we will focus on fixing any bugs found during the testing, fixing and doing major edits on the UI, writing the documentation. We will also be working on the poster, talking about the problem we wanted to address, first person sources, and disability analysis, demo/ screen shots of the extension. If we end up needing to do a major fix, we will split into 2 groups, one group doing the poster design and writing and the other group doing the final fixes and clean up.

Feasibility Analysis (Group)

Reflect on feasibility: Is the idea viable within the constraints of the class. This is hard, because scope may still be uncertain, and your approach is still being defined. You can revise, but it's important to go in with a plan. Try not to downplay the risks, it's better to have something do-able in the end!

- Technically (can be built within 3 weeks)
    - What technologies will you use
    - How well does your group know them
    - What is your fallback plan?

  Timeline
    - How many hours per week can each person put in? Does this match the expectations of the timeline?

For each aspect of feasibility: What will you need? What challenges do you foresee?

Response

We believe this project is feasible within our timeframe, but only if we clearly define which features we are implementing and how deep each one goes. Instead of redesigning the entire web, we are building a browser layer that modifies CSS, media behavior, and layout structure in controlled ways using the Chrome Extensions API.

Core Features and Technical Feasibility

Motion Reduction
• What it means: Disable or reduce CSS animations, transitions, parallax effects, and autoplay media that create unexpected movement.
• How we implement it:
– Inject a content script that overrides CSS using * { animation: none !important; transition: none !important; }
– Detect and pause <video> and <audio> elements with autoplay enabled
– Optionally apply prefers-reduced-motion style overrides at runtime
• Tools that support this: Chrome Extensions content scripts, DOM querying,

MutationObserver to catch dynamically loaded content

• Feasibility: High. This is technically straightforward and well documented. The main challenge is handling dynamically injected content, which can be addressed with observers.

Autoplay and Dynamic Content Suppression

• What it means: Stop videos, animated ads, and constantly refreshing content from updating unexpectedly.
• How we implement it:
– Programmatically pause media elements
– Remove or hide elements that contain autoplay attributes
– Optionally freeze GIFs using CSS filters or replacement techniques
• Tools that support this: DOM manipulation, querySelectorAll, Chrome permissions for activeTab
• Feasibility: Moderate to high. Autoplay pausing is simple. Ad and refresh detection is more variable depending on site structure.

Layout Simplification and Visual Noise Reduction

• What it means: Remove sidebars, pop ups, banner ads, and non essential visual clutter while keeping the main text readable.
• How we implement it:
– Identify common semantic tags such as <aside>, <nav>, and known ad container patterns
– Hide elements with high z index popups
– Recenter <main> or primary content container and increase line spacing
• Tools that support this: DOM inspection, CSS overrides, browser developer tools for testing site structures
• Feasibility: Moderate. Websites vary widely in structure, so we will focus on rule based simplification rather than perfect redesign. We will test on a fixed set of common sites to keep scope manageable.

Color and Contrast Adjustments

• What it means: Allow users to reduce harsh contrast, lower brightness, or switch to a softer color palette.
• How we implement it:
– Inject CSS variables that override background and text colors
– Offer preset themes such as low contrast mode or sepia mode
• Tools that support this: CSS filters, root variable overrides, popup UI state storage
• Feasibility: High. This is controlled styling and does not depend heavily on site structure.

User Controlled Toggles

• What it means: Let users decide what gets removed or reduced instead of enforcing one fixed low stimulation mode.
• How we implement it:
– Build a popup interface using HTML and CSS
– Use Chrome storage API to persist user preferences
– Dynamically enable or disable features per site
• Tools that support this: Chrome storage.sync or storage.local, popup scripts, message passing between popup and content script
• Feasibility: High. This is standard Chrome extension architecture.

Accessibility of Our Own Interface

• What it means: Our popup must be keyboard navigable, screen reader compatible, and WCAG aligned.
• How we implement it:
– Use semantic HTML
– Ensure focus states are visible
– Test with axe DevTools and Lighthouse
• Feasibility: High. The popup UI is small in scope.

Scope Control and Fallback Plan

If layout rewriting across many websites becomes too unpredictable, our fallback plan is to focus on:

• Motion reduction
• Autoplay suppression
• Color and contrast presets
• A clean reading mode that isolates article text using document.body.innerText extraction or a simplified container clone

These features alone still meet the goal of sensory reduction and are realistic within three weeks.

Time and Risk Considerations

Each group member contributing 5 to 10 hours per week gives us roughly 60 to 120 total hours across three weeks. That is enough time to:

• Build a working prototype in week one
• Integrate and test across 5 to 7 real world sites in week two
• Debug inconsistencies and prepare presentation materials in week three

The biggest technical risk is inconsistency across websites. Different DOM structures, dynamic frameworks like React, and aggressive ad injection may cause layout conflicts. To manage this, we will define a fixed test set of sites early and build for those cases first. That ensures we have a stable, demonstrable product rather than chasing edge cases.