A3: App Concept, Framework, and Problem Statement

Our group decided to revamp the ZARA Desktop Website. The mindmap for the development plan can be viewed at FigJam Brainstorm Board

Framework and Build Plan

1. The tool we chose

- We chose the Next.js JavaScript framework to build our project
- Supporting libraries and tools will be used along with next.js are:
    - API Handling: axios for HTTP requests
    - Styling: Tailwind CSS
    - State Management (to manage cart, auth, UI state, etc.): Redux Toolkit and built-in React state
    - Pre-built UI Components: Material UI

2. Why we choose it

- E-commerce websites like Zara.com require a web app that prioritizes performance, scalability, SEO, and a responsive, user-centric shopping interface. Next.js meets the standard, and thus, we decided to use this framework.

- Benefits of using Next.js for our project:
    - **Built-in API Routes**: enables lightweight backend logic such as cart management and product filtering without needing a separate backend server
    - **Developer Efficiency**: file-based routing simplifies page organization, and the server-side rendering feature of Next.js allows us to fetch real-time product data (stock, pricing, availability) before the page loads.
    - **CI/CD and hosting support:** with Next.js, we can use Vercel's built-in CI/CD functionality, which automatically builds, tests, and deploys our app when we push to GitHub
    - **Faster initial load speed**: we can fetch and render data on the server, then send a fully hydrated page to the browser. This is also good for SEO as it allows search engines to crawl the page's details effectively.
    - **Image Optimization**: heavy image rendering for an e-commerce website can reduce performance. To handle this, we use the Next.js built-in image component to automatically resize and compress images when the screen size changes, and to lazy-load images for faster page loads. Images are loaded only when they enter the viewport.
    - **Internationalization Support**: Next.js includes built-in features for multi-language routing, which is ideal for global brands like Zara.

- For the other mentioned libraries and tools, we will use them to enhance the project's efficiency and scalability:
    - **Axios:** simplifies HTTP requests with cleaner syntax.
    - **Tailwind CSS:** allows responsive UI development.
    - **Redux Toolkit:** helps manage global states like cart and auth
    - **Material UI:** provides pre-styled, customizable components that speed up development.

3. Challenges and obstacles we might encounter while using the tool

- Next.js is full-stack so we must manage both frontend and backend (API routes, middleware, server-side logic). This is gonna be a moderate learning curve, since we usually have the front-end and back-end handled separately.
- Choosing between SSR, SSG, CSR, and ISR in Next.js project can be tricky, since each has trade-offs in SEO, performance, and complexity. UI inconsistencies between server and client render can occur, and to avoid this issue, we'll try to follow these rules:
    - Use SSG (static site generation) when the data rarely changes.
    - Use SSR (server-side rendering) for dynamic, user-specific or real time data.
    - Use CSR (client-side rendering) for highly interactive pages
    - Use ISR (incremental static regeneration): Mix of SSG + SSR, to update static pages on demand (e.g., product pages updated every 60s)
- Deep routing and maintaining state (e.g., cart/auth) across server and client render is challenging and needs a structured design.
- Next.js doesn't come with a built-in testing framework, so we'll need to use other testing frameworks or do this manually.

4. Backup plan in case we run into project-ending problems

In case of project-ending challenges such as integration failures, server-side rendering issues, or deployment blockers, the following strategies will be used:

- Declare all components as client-side components, as in a basic React project, to reduce complexity
- If third-party services or database connections fail, replace them with mock data or simplified UI components to preserve the user flow for demonstration purposes.
- If dynamic features cannot be supported, present a static version using Tailwind CSS and Material UI.
- Manually test the functionality of our project if setting up a testing framework involves too many challenges.

App Concept and Problem Statement

Overview

Our project aims to reimagine the desktop version of the Zara website to create a more accessible and inclusive online shopping experience while preserving the brand's signature minimalism and photo-centric aesthetic. We are developing a more responsive e-commerce website using Next.js, a React-based framework optimized for speed, scalability, and SEO capabilities, which are key features for global retail platforms. Support tools include Axios for clean API handling, Tailwind CSS for modular, responsive design, Redux Toolkit and React state for managing interactions such as shopping cart and accessibility toggles, and Material UI for efficient UI development. Our goal is to build a website that not only visually aligns with Zara's editorial brand identity but also meets accessibility standards, ensuring the experience is equitable and intuitive for all users.

Problem Statement

Zara presents accessibility options for their desktop website, but their implementation falls short, causing usability barriers such as insufficient text-to-background contrast, broken layout responsiveness, and truncated text. In an e-commerce setting where quick navigation and visual clarity are vital, we need to create an intuitive and fully responsive shopping platform that meets accessibility standards through optimized imagery, seamless navigation, and device flexibility, while maintaining the elegance and simplicity of Zara's brand.

Requirements

- **Navigation Bar:** We will redesign this in order to make it so that it does not follow the user when they are scrolling down the page and blocking their view when looking at the clothing on the page.
    - Decrease the "Zara" logo size on the navigation bar to be in line with the rest of the navigation bar. This way, there is a clear distinction between the header and the main body of the website.

    - Add a "shopping cart" or "shopping bag" symbol next to the "SHOPPING BAG" tab on the navigation bar to make it easier for users to locate their cart. Additionally, remove the "[ ]" next to the "SHOPPING BAG" and display the number of items in the cart by overlaying it on the symbol.
        - Example:

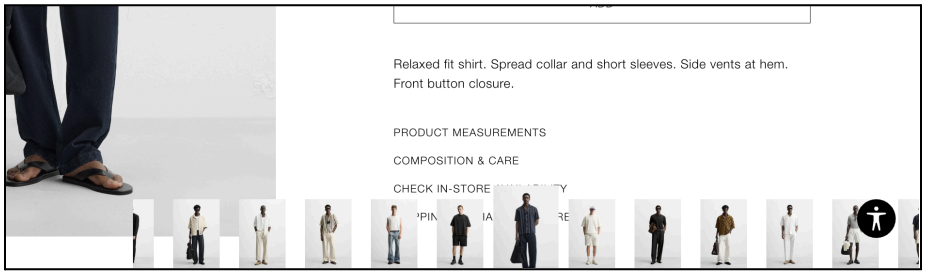
- Make the navigation bar static – when selecting an option from the navigation bar, the option should remain on the navigation bar once the new page loads (it should not disappear). Instead, navbar selection will be indicated by highlighting the selected option (underlining, boldening, etc.). This way, consistency is maintained, and users will not be confused by frequently changing elements.
    - For example: After selecting the “SHOPPING BAG” option from the navigation bar, the navigation bar changes, and the original options are replaced.

      Before:

      After:

- **Fonts**: We intend to improve readability and fix the wording on the website by increasing font size and choosing a better font that is more accessible to the audience.
- **Accessibility**: We plan to fix the accessibility issues that are present with the layout of the page, along with making the accessibility features work properly with the page. Using the W3C guidelines for accessibility, we will implement a Level AA of conformance.
    - Follow W3C recommendations for color contrast, text/wording, and images.
- **Layout**: We want to redesign how the products and the product images are displayed to the user, as the current layout is not consistent with different products and is not easy to navigate through to look at different options. Images are placed in different locations and are of different sizes.
    - Product Listing: Product images will be presented in a slide format, rather than separate images at the bottom of the listing.

        - The carousel at the bottom of the page will be removed/relocated to prevent clutter and blockage of product details.



    - Product Browsing: We will standardize the layout of each page to maintain consistency across the entire site. Product pages will all follow the same format, with the same number of listings presented in each row/section. When choosing the number of listings per section, we will ensure that listing images and descriptions are easily visible to users.

- **Color scheme**: We plan on fixing the color contrast on the website to make it easier to read and see the text on images and with the background of the photos to the clothes as they are inconsistent with each other and blend into the overall white background.
- **Sidebar Menu**: We plan to make changes to the sidebar menu to be more intuitive and user-friendly. We will rearrange the menu options to adopt an intuitive hierarchy, adjusting fonts and headers to highlight different categories. Options will be consolidated to prevent information overload, as well as relocating irrelevant options to other parts of the website (For example: “Careers” and “Stores” will be moved to the footer).