Write-up

Part 1

The purpose of my website is to serve as a resource where Pennsylvania residents can learn about logistical voting information for upcoming elections. My target audience is broad: the 13 million residents of Pennsylvania, particularly those new to or unfamiliar with the process. The design is simple, and the title and tagline make it clear that it's geared toward Pennsylvania voters. It includes basic and essential information including how to register to vote, how to vote in person and by mail, how to vote early and on Election Day, and where to find information if you have disabilities and need help. I linked some sections to other pages, including registration forms, but I tried to keep most information on the webpage itself. This choice was in response to feedback I received from classmates in lab.

Pennsylvania is a crucial swing state in national elections. In 2016, only 70% of registered voters actually voted. There are many accessibility issues surrounding voting in the USA, including barriers to obtaining information, registering, and traveling to the polls. Therefore, I wanted to ensure that my site is as interesting, engaging, and accessible as possible. Website design can be solved more quickly and easily than bigger policy and systemic issues like voter suppression, and we must do whatever we can to alleviate unjust barriers to voting. I designed the color palette on Adobe to be accessible for visually-impaired and colorblind folks, and I translated everything into Spanish for extra accessibility. I distinguished each section visually by alternating between white and light blue backgrounds to keep things visually interesting. I ordered the sections logically to minimize confusion and keep viewers engaged.

Part 2

- **Change languages**: to toggle back and forth between English and Spanish, click the default "English" in the menu and then select "Español". You should now see all of the content in Spanish. You can repeat the process and select "English" to change back to English
- **Interactive menu**: Click any menu item at the top of the page to scroll down to that section. For instance, you can click "Vote in Person" on the menu, and you'll be taken to that part of the page
- **Open and close expandable content**: to open expandable content, click the down arrow at the right of any informational heading (e.g., Voting Eligibility). Expand "Voting Eligibility" by clicking the right down arrow, which will subsequently turn into an up arrow. Then, click on the up arrow to collapse the box.

Part 3

- Name of tools: I used React (JavaScript library for building user interfaces) and MUI (Material-UI).
- Why I used them: I used React because I wanted to use state to switch the website between English and Spanish. It has a bunch of pre-built components, so I didn't have to code the components from scratch. The MUI React component library implements Google's Material Design, making the layout process way more efficient and better looking.
- How I used them: With Material UI installed, I could easily implement great responsive accordions throughout my React application. The accordions are super user-friendly, visually appealing, and accessible. React, once I learned it, facilitated building out my website structure.
- What they add to the website: I liked the design and structure MUI added to my site, and it has a good accordion component that I implemented (for the text to be able to drop down and come back up). React helped me write a codebase that is flexible and relatively easy to edit, which would be way more complex using plain HTML, CSS, and JS.

Part 4

I ended up changing the style of my website, though the initial design structure I chose in FP2 remained relatively the same. The changes I made were mostly to bolster visual accessibility and make the website more visually appealing. However, I did have to change how I envisioned the top-level navigation at the top; React took a really long time for me to learn from nothing, and I couldn't figure out how to make the menu items stretch and retract. So as an alternative, I kept it responsive by dropping menu items down a line as the screen width shrinks.

Part 5

Among all the React-related challenges, I found it particularly challenging to implement the language-switching option, which I'd originally added to make my language accessible to Spanish speakers. Complying with the WAVE tool was a headache because it couldn't detect my original MUI labels, thereby incorrectly marking incorrectly that I had a "Missing form label" error. To fix this error and pass the WAVE test, I consulted with 2 TAs and some CS friends to resolve this, but ended up having to set up a decoy label (you can see the decoy label on line 19 of LanguageSelector.js, and the correct label that WAVE couldn't register on line 20 –– this fix makes the "Language" label look weird to comply with WAVE).

Screen sizes to should test the interface: 1440px (laptop/Macbook Pro) and 1024 (tablet/iPad)

"Summary" and "Details" sections from the WAVE tool: