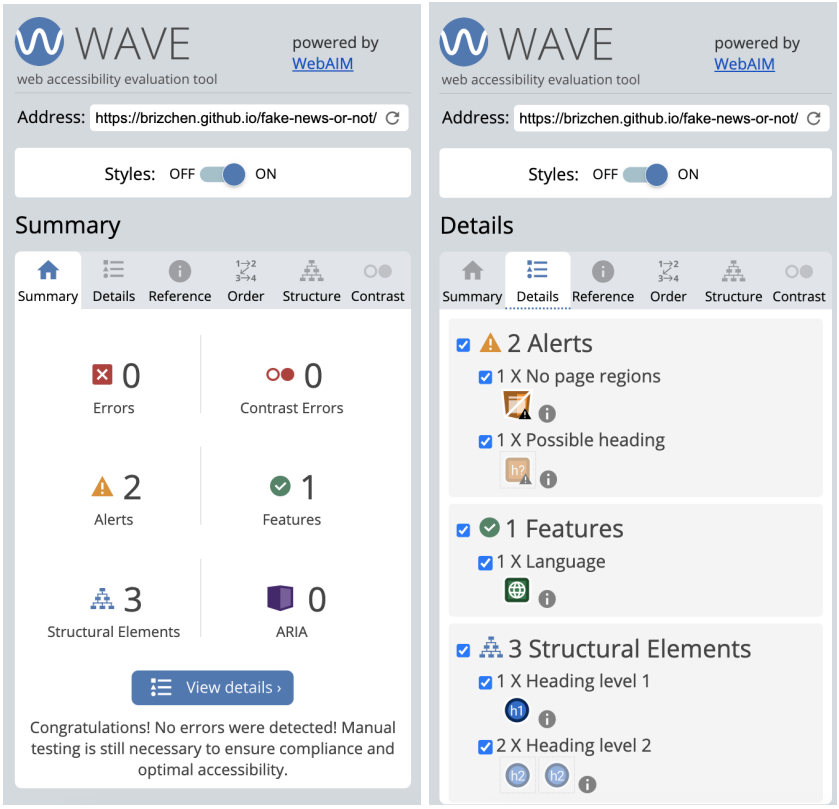
Briana Chen

05-430 Programming Usable Interfaces (Lab Section C)

Professor Alexandra Ion

FP4 - Final Project Write-up

Screen sizes: 1280 x 720 (Desktop); 1024 x 1366 (Tablet)



WAVE Tool: Screenshots of "Summary" and "Details" pages

Part 1

How easy is it to create misinformation in a world where humans are relying more on artificial intelligence? This website is an interactive game that aims to educate users about this topic. It also aims to convey that when training machine-learning algorithms, it's becoming more difficult to distinguish real between fake, making it more difficult to control the spread of misinformation in the digital news space.

The website conveys a mix of semantic feedback as well as several cues that add to the gameplay experience. It displays a series of headlines to the user as the main focal point, then affords them to engage with the buttons. On the 'game over' screen, there is a write-up about how real and fake headlines are becoming more difficult to distinguish, thus leading to difficulty in training ML algorithms and controlling the spread of misinformation. The 'next step' the user can take is to read more about the topic and how they can take charge of its impact.

Through an engaging game system, users are invited to guess whether a displayed headline is real or fake. Some headlines are real, others are fake and generated by AI. At the end, players are scored to see how they perform. This experience is enhanced through visual feedback cues such as color changes, animation, and a progress-tracking bar.

The target audience is the average American citizen who consumes digital media on a regular basis. They might be a college student who consumes news primarily on social media; they might also be an older working adult who interacts with news articles online. They are aware that we live in a digital age with misinformation in the news, but they are likely not aware of the deeper implications with regards to AI and machine learning.

Part 2

- View screen (*exploring*) - scroll on Start Screen
- Initiate game (*instructing*) - click on "Start" button on Start Screen
- Choose real or fake headlines (*instructing*) - click on "Real" or "Fake" buttons on Game Screen
- Read more in external site (*instructing/exploring*) - click on "here" link on Game Over Screen
- Start a new game (*instructing*) - click on "Play Again" button on Game Over Screen

Part 3

- Web API: GNews API (https://gnews.io/)
- I used GNews because it has a "top headlines" feature that allows users to extract real headlines across the globe. It also allows tailoring to categories and

geographic regions. The average response time is quick and CORS is enabled in the free plan for all origins.

- I incorporated it into the JSON code file to extract the database of real headlines. From this API, the game would display real-world headlines for the player to determine the authenticity of.
- It adds a crucial database (the real headlines) to the website, creating a binary where the user can either be right or wrong. Because these headlines are tagged to be true, if the user guesses wrong then their score is impacted. Therefore, this API adds the functionality and engagement aspects to the website.

Part 4

The first prototype used the NewsAPI (https://newsapi.org/) and didn't contain much context on the start and end screens. I iterated numerous times based on user feedback, including adding content to the start and end screens to provide context. Next, I added a progress bar on the game screen as feedback for the user as to how long the game would last. I also made the score bold and changed its color to provide additional feedback as to whether they guessed correctly or not. When I ran the site in GitHub, I realized the NewsAPI was only CORS-enabled for the local host, so would only run in Live Server. As a result, I changed the API to the GNews API and modified the JSON file to extract from the new source. This required a change in the "fake" headlines as well to more closely match the headlines from the new API.

Part 5

As mentioned before, one challenge was figuring out what APIs to use and how to incorporate them into the code so that they worked properly. Although the NewsAPI had the same benefits in terms of its data sources, GNews was superior in its free plan, though it formatted its headlines differently so I needed to adjust for that. It was also difficult to write the JavaScript code to implement user feedback, since JavaScript tended to fail silently. I spent a lot of time trying to figure out why a button click had no feedback, or why the image in the background wouldn't load, etc.