

Group Members:

Archna Sobti: archnas2

Anjali Kumar: anjalik4

Aryan Vakharia: aryannv2

Manu Pillai: mpilla6

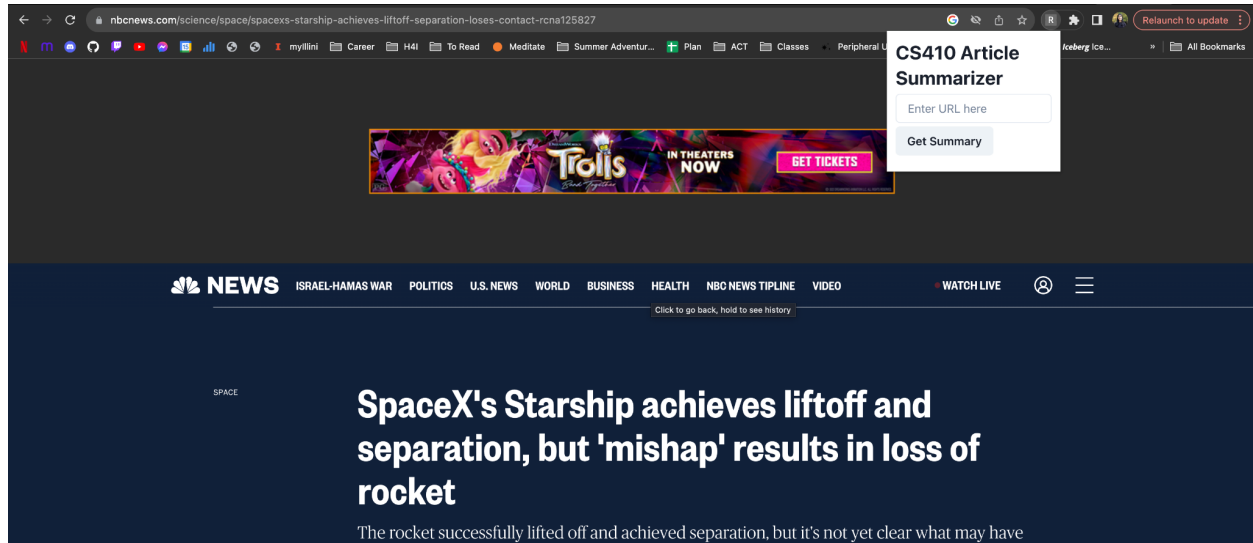
Progress Made

Backend:

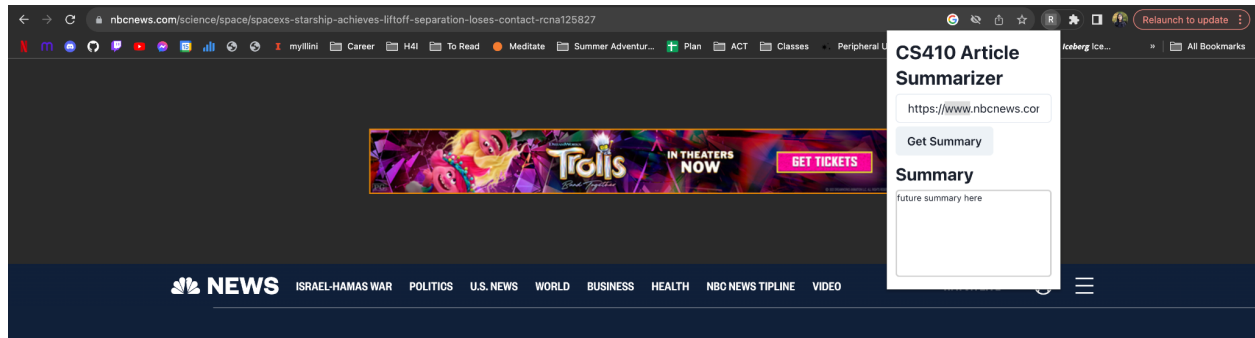
We created a server backend using node.js and express.js that so far takes a url input, and retrieves all the text from that webpage using express.js and node.js as part of the request body. We also used python and the meta toolkit to create an initial version function that summarizes the text (uses inverted index and BM25) into a default minimum of 3 sentences. The parameter for the number of sentences of the summary is subject to change. Furthermore, we tested this URL parsing functionality using .ejs files which created a visual testing method for the URL parsing route.

Frontend:

We have built out the frontend for the chrome extension, adding an input for the user to enter a URL to a page we'll summarize. We are also considering functionality that automatically retrieves the URL of the current page a user is viewing.



Whenever the user presses the get summary button, they will see the summary of the article show up like the example below. This will be placed within a text box that will only appear once a summary has been generated.



General:

We have also worked to integrate the frontend and backend, so the frontend can make API calls to the backend, requesting a summary. This then populates the summary field on the frontend.

Remaining Tasks

Backend:

We need to clean the text, because right now every piece of text on the webpage, including the page's backend html tags and everything are also outputted as text. In order to do this, we will use pattern matching and regex to match only the type of content that we want to display. We then need to save that text as a txt file, and save that file as a toml file to use in our summarizing function since the input of the function is required to be a toml file.

Frontend:

We want to clean up the extension to look more professional. This includes making the summary box look better, potentially adding some color to the extension, and changing the overall extensions shape to not be so boxy. We will also consider ways to format the summary neatly within the box (perhaps using bullet points, indentations, etc).

General:

We need to deploy our extension, which we expect to take some time since we need to run our backend on a server.

Challenges/Issues

Backend:

Currently, our output also parses the HTML code of the website along with unwanted characters, so a separate function using regular expressions or NLTK to clean the website to have just words is needed to solve this issue. We also faced issues with installing the MeTA toolkit due to it being compatible with Python 2.7 only. This was resolved by using a Conda virtual environment.

Frontend:

We were using ChakraUI as our UI library, and we ran into some issues with getting the styling the way we wanted it. We played around with a few different components, and finally decided to hardcode some styling for now to make it look the way we wanted.

General:

While integrating the frontend and backend, there were some issues with getting the response from the API. For example, while calling for the summary, we ran into an issue with accidentally trying to send a response twice, resulting in `err_connection_refused`. After some research, we figured out that we did not need to call `res.render()`, which was initially used for testing the API call output and was left in the code.