

# Proposal

<b>Project Name</b>	NewsFlex for WaPo: Your Personalized News Companion
<b>Team Members</b>	Brandon Gong, Chris Xu, Sam Shi, Yuyan Lei
<b>Team Roles &amp; Responsibilities</b>	<ul style="list-style-type: none"><li>• User Interface dev + options page(1 person)</li><li>• OpenAI API Dev (1 person)</li><li>• Feature Handlers (4 features for 2 persons)</li></ul>
<b>Repository Link</b>	POC: <a href="#">WaPoExtension</a>
<b>Slides Link</b>	

## Project Overview

NewsFlex for WaPo is a Chrome extension that provides additional website customization of accessibility settings for the WaPo official website, while incorporating the capabilities of OpenAI to provide content-related features.

### Key Features

#### a. For Young or Educationally Disadvantaged Users:

Replace the article content with a version that uses simpler vocabulary.

#### b. For ADHD Users:

Change the font to a specific type suitable for individuals with ADHD.

Offering short-read (200-word summary) and medium-read (500-word summary) options at the beginning of each article for users to choose and read.

#### c. For Non-Native English Speakers:

Display the original text on one line and the translation in another language on the following line. This display format would be more suitable for quickly reading in other languages while cross-referencing details with the English version. (The browser's built-in Google Translate provides global translation and hides the original text, which is not ideal for this purpose.)

### Goals

Through this project, we aim to provide additional accessibility customization features for the Wapo official website, allowing young users, educationally disadvantaged users, ADHD users, and non-native English speakers to access and understand the content more easily. The implementation of this project will contribute to improving the accessibility of the Wapo official website and provide a better reading experience for specific user groups.

Moreover, these features are not limited to specific user groups; the descriptions provided earlier are merely introductions for potential user segments. In the actual extension development, we won't explicitly specify which features are suitable for particular groups. Instead, we will openly offer all four functionalities, allowing any user to use them individually or in combination to enhance their reading experience on The Washington Post website.

For instance:

- A busy professional who works long hours and wants to quickly browse through a large number of daily news articles after a whole day of work might find the summary feature particularly useful. This way, they do not need to spend too much time reading lengthy articles,

making it convenient and time-saving for their busy schedule.

- A parent looking to share quality time with their child may opt to activate the summary feature to quickly identify articles suitable for their child to read, while also enabling the simpler content feature to read the articles to their child.
- An English learner may find it helpful to enable the font replacement feature to speed up English reading and also enable the translation feature to cross-reference translations as a reference.

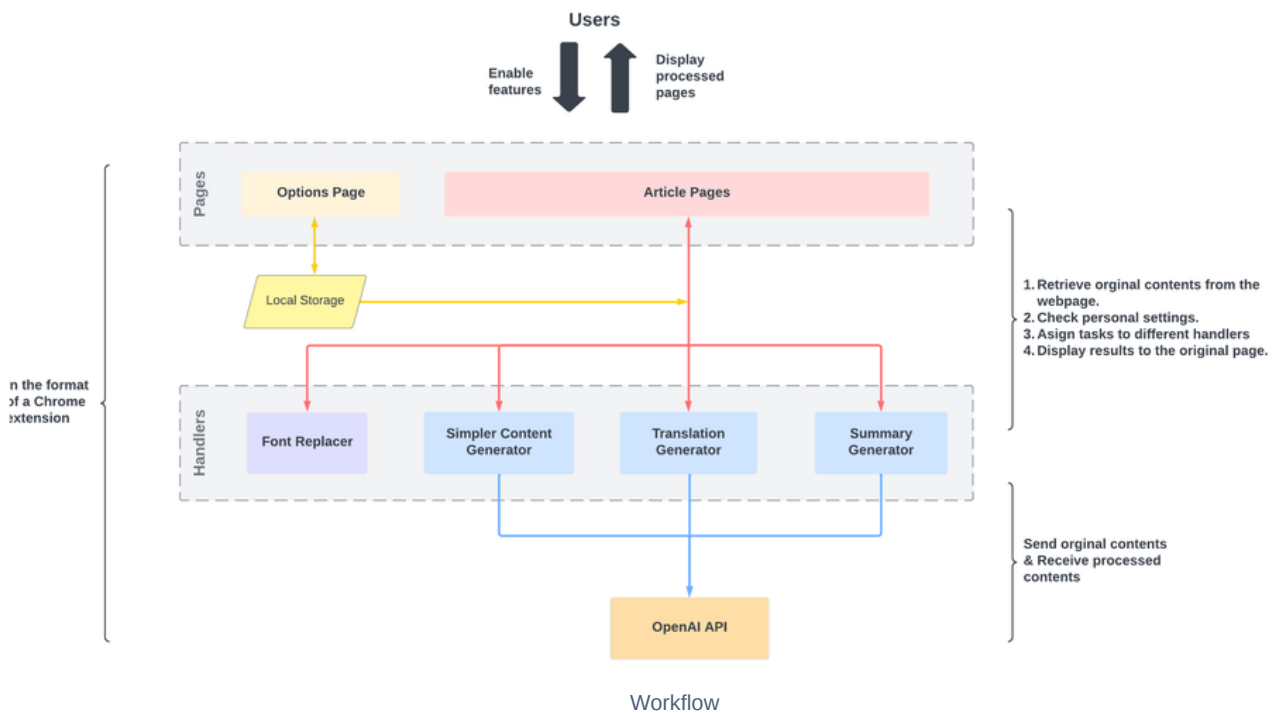
These four features can be freely combined and used in various ways, offering limitless possibilities waiting to be explored. Different readers will be able to find their preferred combinations among these features. No matter the combination, these features can greatly enhance their reading experience and make them fall in love with reading news on The Washington Post website.

Our goal is to ensure that everyone has equal access to information about the world through news and can read it in their preferred way. Read news, read society, and read the world.

## Technical Stacks

- JavaScript
- OpenAI API
- Chrome Extension APIs

## Implementation Plan



## User Interface

The user interacts with two primary types of pages: the personal settings page and the article pages.

### 1. Toolbar Popup & Options Page:

The toolbar popup and options page allow users to enable or disable the four available features and customize related settings. These choices are saved to the browser's local storage, ensuring persistence across sessions.

### 2. Article Pages:

When a user visits an article page, the system will first check the enabled features in the local storage. Based on this information, the corresponding handler will be triggered to implement the chosen functionalities.

## Extension Handlers

There are four essential handlers, each responsible for a specific task:

1. **Font Replacement:** If the user selects the font replacement feature, the extension invokes the font replacement handler immediately to handle the request.
2. **Simpler Versions Generation:**
  - a. For users opting for simpler versions of the article, the system reads the main content paragraph by paragraph and sends each paragraph to the OpenAI API for processing.
  - b. The processed data is then used to substitute the original text with the simplified version.
3. **Translation Generation:**
  - a. For users desiring translated articles, a similar process occurs, where the system reads each paragraph's content and sends it to the OpenAI API for translation.
  - b. The translated text is then displayed below the original text paragraph by paragraph.
4. **Summary Generation:**
  - a. If the user chooses the summary generation feature, the extension extracts the entire article content and sends it to the OpenAI API to generate a concise summary.
  - b. Expected Display Outcome :
    - i. At the beginning of the article, two collapsible sections will be displayed, offering short-read (200-word summary) and medium-read (500-word summary) options. Upon page refresh, only the collapsed form title will be visible. When the user wishes to read, they can click on the respective section to expand and reveal the specific summary content.
    - ii. Always load the summary content when loading the whole page, no matter user clicks or not.

## Tasks

Type	Task	Status	Details	Assignee
Preparation	Design the whole workflow	IN PROGRESS		
	POC	COMPLETED	<input checked="" type="checkbox"/> POC of content substitution and insertion <input checked="" type="checkbox"/> POC of chrome extension dev	
	Basic decisions		<input type="checkbox"/> Identifying what content should be shown on the user interface <input type="checkbox"/> Specifying the particular settings to be provided in the options page, for example, target language selection and text display styles for the translator. <input type="checkbox"/> How to solve feature conflicts? Functionality exclusivity? (e.g. translation & simplification)	
Chrome Extension Dev	init	COMPLETED	<input checked="" type="checkbox"/> Initialize the project with basic structure of an extension	
	User Interface		<input type="checkbox"/> Build a toolbar popup with four switches for feature selection <input type="checkbox"/> Side panel, tool tip, context menu. <input type="checkbox"/> Store options in the local storage  Doc: <a href="#">Chrome Extensions: Design the user interface - Chrome Developers</a> , <a href="#">chrome.storage - Chrome Developers</a>  Example: <a href="#">Chrome Extension development basics - Chrome Developers</a>	

	Options page		<input type="checkbox"/> Build a options page with detailed settings. <input type="checkbox"/> Store options in the local storage <a href="#">Chrome Extensions: Give users options - Chrome Developers</a> <a href="#">chrome.storage - Chrome Developers</a>	
Handlers	Font Replacer		<input type="checkbox"/> Implement font changing functionality	
	Simpler Version Generator		<input type="checkbox"/> extract paragraph contents <input type="checkbox"/> substitue paragraphs via script	
	Translation Generator		<input type="checkbox"/> extract paragraph contents <input type="checkbox"/> insert translated content into each paragraph via script <input type="checkbox"/> Translate each paragraph asynchronously from top to bottom, and display a loading icon for paragraphs that have not been translated yet.	
	Summary Generator		<input type="checkbox"/> extract the whole article content <input type="checkbox"/> insert two collapsible summary sections at the beginning of the article via script	
OpenAI	OpenAI API		<input type="checkbox"/> interact with 4 handlers <input type="checkbox"/> design prompts for API usage Example: <a href="#">openai-translator/openai-translator</a>	
Translation	Translation APIs		If time allows, integrating other better translation APIs to offer users more choices instead of using OpenAI for translation. For instance, Google Translate and DeepL.	

Useful Information to start with:

- [Chrome Extensions 101 - Chrome Developers](#)
- [Chrome Extensions architecture overview - Chrome Developers](#)
- [chrome.scripting - Chrome Developers](#)