evaluate_L4

December 9, 2025

1 Set up environment (Google Colab)

[2]:

```
!pip install selenium
!pip install pandas
!pip install ollama
```

```
Collecting selenium
  Downloading selenium-4.39.0-py3-none-any.whl.metadata (7.5 kB)
Requirement already satisfied: urllib3<3.0,>=2.5.0 in
/usr/local/lib/python3.12/dist-packages (from
urllib3[socks]<3.0,>=2.5.0->selenium) (2.5.0)
Collecting trio<1.0,>=0.31.0 (from selenium)
  Downloading trio-0.32.0-py3-none-any.whl.metadata (8.5 kB)
Collecting trio-websocket<1.0,>=0.12.2 (from selenium)
  Downloading trio_websocket-0.12.2-py3-none-any.whl.metadata (5.1 kB)
Requirement already satisfied: certifi>=2025.10.5 in
/usr/local/lib/python3.12/dist-packages (from selenium) (2025.11.12)
Requirement already satisfied: typing_extensions<5.0,>=4.15.0 in
/usr/local/lib/python3.12/dist-packages (from selenium) (4.15.0)
Requirement already satisfied: websocket-client<2.0,>=1.8.0 in
/usr/local/lib/python3.12/dist-packages (from selenium) (1.9.0)
Requirement already satisfied: attrs>=23.2.0 in /usr/local/lib/python3.12/dist-
packages (from trio<1.0,>=0.31.0->selenium) (25.4.0)
Requirement already satisfied: sortedcontainers in
/usr/local/lib/python3.12/dist-packages (from trio<1.0,>=0.31.0->selenium)
(2.4.0)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages
(from trio<1.0,>=0.31.0->selenium) (3.11)
Collecting outcome (from trio<1.0,>=0.31.0->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.12/dist-
packages (from trio<1.0,>=0.31.0->selenium) (1.3.1)
Collecting wsproto>=0.14 (from trio-websocket<1.0,>=0.12.2->selenium)
  Downloading wsproto-1.3.2-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in
/usr/local/lib/python3.12/dist-packages (from
urllib3[socks]<3.0,>=2.5.0->selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.16.0 in /usr/local/lib/python3.12/dist-
```

```
packages (from wsproto>=0.14->trio-websocket<1.0,>=0.12.2->selenium) (0.16.0)
Downloading selenium-4.39.0-py3-none-any.whl (9.7 MB)
                         9.7/9.7 MB
32.6 MB/s eta 0:00:00
Downloading trio-0.32.0-py3-none-any.whl (512 kB)
                         512.0/512.0 kB
42.3 MB/s eta 0:00:00
Downloading trio_websocket-0.12.2-py3-none-any.whl (21 kB)
Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Downloading wsproto-1.3.2-py3-none-any.whl (24 kB)
Installing collected packages: wsproto, outcome, trio, trio-websocket, selenium
Successfully installed outcome-1.3.0.post0 selenium-4.39.0 trio-0.32.0 trio-
websocket-0.12.2 wsproto-1.3.2
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages
(2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-
packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-
packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-
packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Collecting ollama
  Downloading ollama-0.6.1-py3-none-any.whl.metadata (4.3 kB)
Requirement already satisfied: httpx>=0.27 in /usr/local/lib/python3.12/dist-
packages (from ollama) (0.28.1)
Requirement already satisfied: pydantic>=2.9 in /usr/local/lib/python3.12/dist-
packages (from ollama) (2.12.3)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages
(from httpx>=0.27->ollama) (4.12.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-
packages (from httpx>=0.27->ollama) (2025.11.12)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-
packages (from httpx>=0.27->ollama) (1.0.9)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages
(from httpx>=0.27->ollama) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-
packages (from httpcore==1.*->httpx>=0.27->ollama) (0.16.0)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.12/dist-packages (from pydantic>=2.9->ollama) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in
/usr/local/lib/python3.12/dist-packages (from pydantic>=2.9->ollama) (2.41.4)
Requirement already satisfied: typing-extensions>=4.14.1 in
/usr/local/lib/python3.12/dist-packages (from pydantic>=2.9->ollama) (4.15.0)
Requirement already satisfied: typing-inspection>=0.4.2 in
```

```
/usr/local/lib/python3.12/dist-packages (from pydantic>=2.9->ollama) (0.4.2)
Downloading ollama-0.6.1-py3-none-any.whl (14 kB)
Installing collected packages: ollama
Successfully installed ollama-0.6.1
```

[8]:

```
!sudo apt update
!sudo apt install -y pciutils
!curl -fsSL https://ollama.com/install.sh | sh
```

```
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease
[3,632 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
InRelease [1,581 B]
Hit:3 https://cli.github.com/packages stable InRelease
Get:4 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ Packages [83.6
kB]
Get:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
Packages [2,201 kB]
Hit:6 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:7 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:9 https://r2u.stat.illinois.edu/ubuntu jammy InRelease [6,555 B]
Get:10 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:11 https://r2u.stat.illinois.edu/ubuntu jammy/main all Packages [9,519 kB]
Get:12 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
[18.1 kB]
Hit:13 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
InRelease
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [3,904
kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages
[3,573 kB]
Hit:16 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Get:17 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy/main amd64
Packages [38.5 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages
[1,598 kB]
Get:19 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages
[6,287 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
[1,287 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64
Packages [6,081 kB]
Get:22 https://r2u.stat.illinois.edu/ubuntu jammy/main amd64 Packages [2,844 kB]
Fetched 37.8 MB in 4s (9,197 kB/s)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
```

```
72 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: Skipping acquire of configured file 'main/source/Sources' as
repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem
to provide it (sources.list entry misspelt?)
Reading package lists… Done
Building dependency tree… Done
Reading state information… Done
The following additional packages will be installed:
  libpci3 pci.ids
The following NEW packages will be installed:
  libpci3 pci.ids pciutils
0 upgraded, 3 newly installed, 0 to remove and 72 not upgraded.
Need to get 343 kB of archives.
After this operation, 1,581 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 pci.ids all
0.0~2022.01.22-1ubuntu0.1 [251 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libpci3 amd64 1:3.7.0-6
[28.9 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 pciutils amd64 1:3.7.0-6
[63.6 kB]
Fetched 343 kB in 2s (204 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78,
<> line 3.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package pci.ids.
(Reading database … 121713 files and directories currently installed.)
Preparing to unpack …/pci.ids_0.0~2022.01.22-1ubuntu0.1_all.deb …
Unpacking pci.ids (0.0~2022.01.22-1ubuntu0.1) …
Selecting previously unselected package libpci3:amd64.
Preparing to unpack …/libpci3_1%3a3.7.0-6_amd64.deb …
Unpacking libpci3:amd64 (1:3.7.0-6) …
Selecting previously unselected package pciutils.
Preparing to unpack …/pciutils_1%3a3.7.0-6_amd64.deb …
Unpacking pciutils (1:3.7.0-6) …
Setting up pci.ids (0.0~2022.01.22-1ubuntu0.1) …
Setting up libpci3:amd64 (1:3.7.0-6) …
Setting up pciutils (1:3.7.0-6) …
Processing triggers for man-db (2.10.2-1) …
Processing triggers for libc-bin (2.35-0ubuntu3.8) …
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic
link
```

```
/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero_v2.so.0 is not a
symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic
link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.1 is not a symbolic link

>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 bundle
######################################################################## 100.0%
>>> Creating ollama user…
>>> Adding ollama user to video group…
>>> Adding current user to ollama group…
>>> Creating ollama systemd service…
WARNING: systemd is not running
>>> NVIDIA GPU installed.
>>> The Ollama API is now available at 127.0.0.1:11434.
>>> Install complete. Run "ollama" from the command line.
```

[29]:

```python
import threading
import subprocess
import time

def run_ollama_serve():
  subprocess.Popen(["ollama", "serve"])
```

```
thread = threading.Thread(target=run_ollama_serve)
thread.start()
time.sleep(5)
```

2 Prepare data/files for testing

[30]:

```
import ollama
import json as js
import pandas as pd
import requests
```

[31]:

```
ollama.pull("qwen3-vl:8b")
```

[31]:

```
ProgressResponse(status='success', completed=None, total=None, digest=None)
```

[40]:

```
# Path to your image file
chatgpt_homepage_img_path = "images/chatgpt_loggedin.jpeg"
chatgpt_languages_img_path = "images/chatgpt_languages.jpeg"
claude_homepage_img_path = "images/claude_loggedin.jpeg"
claude_languages_img_path = "images/claude_languages.jpeg"
```

[41]:

```
# Open the image file in binary read mode
with open(chatgpt_homepage_img_path, "rb") as f:
    chatgpt_bytes = f.read()

with open(chatgpt_languages_img_path, "rb") as f:
    chatgpt_languages_bytes = f.read()

with open(claude_homepage_img_path, "rb") as f:
    claude_bytes = f.read()

with open(claude_languages_img_path, "rb") as f:
    claude_languages_bytes = f.read()
```

3 Test L4 Indicators using Qwen3-vl:2B - ChatGPT

3.1 Create an empty dataframe for storing test results

[63]:

```
chat_gpt_test_results_df = pd.DataFrame(columns=["Website", "L4_Indicator",␣
  ↪"Assigned_Score", "Max_Score", "Reasoning"])
```

3.2 L3 Subdimension: The AI is accessible and inclusive across abilities and language

3.2.1 L4: WCAG-aligned accessibility features available

This L4 category covers a wide variety of possible accessibility features as defined in the WCAG 2.1 Guidelines. Given my work as an individual and not a group for this project, the scope of these guidelines is too broad to be sufficiently covered by this work.

Instead, I have chosen to evalute this L4 category by making it a composite score of all the following L4 categories that are evalutated by this notebook. The reasoning for this being that every rating that is being assigned is based on one or more off the Success Criterias listed in the WCAG version 2.1 guidelines, so they fall under the umbrella of "WCAG-aligned accessibility features."

This notebook will act as a proof of concept that additional guidelines and features can be implemented into this evaluation pipeline, following the same structure and build up this composite WCAG-alignment score, so long as the ratings are based on WCAG-guidance as I have done for this project.

As a result, this section will be evaluated again at the end of this notebook once all other L4s have been scored. It is placed here for the time being to remain within the logical grouping as defined in the AI Ethics Index Tree (under the L3 Subdimension: The AI is accessible and inclusive accross abilities and langauges).

3.2.2 L4: Multilingual support for intended locales

[64]:

```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
 ↪help page for ChatGPT regarding what languages the platform is localized␣
 ↪into. Your task is to scan the HTML page for the list of supported␣
 ↪langauges, count the number, then score the page based on the number of␣
 ↪supported websites on the following scale:

0: The website supports 5 or fewer languages
1: The website supports 10 or fewer languages
2: The website supports 20 or fewer languages
3: The website supports 32 or fewer languages
4: The website supports 33 or more languages.

The response should be written in valid JSON format. The following example␣
 ↪below shows properly formatted output. The structure, Website, L4_Indicator␣
 ↪and Max_Score properties should not be changed. You are only writing the␣
 ↪value for Score and Reasoning in the response.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Multilingual support for intended locales",
        "Assigned_Score": ,
        "Max_Score": 4,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_languages_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

Print the results of the prompt for easy reading

[65]:

```
print(eval_result)
```

```
{
    "results": [
        {
            "Website": "ChatGPT",
            "L4_Indicator": "Multilingual support for intended locales",
            "Assigned_Score": 4,
            "Max_Score": 4,
            "Reasoning": "The page lists 39 supported languages, which meets the
criterion for 33 or more languages (score 4)."
        }
    ]
}
```

Append results to the final dataframe for comparison later

[66]:

```
out = js.loads(eval_result)
out
```

[66]:

```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Multilingual support for intended locales',
   'Assigned_Score': 4,
   'Max_Score': 4,
   'Reasoning': 'The page lists 39 supported languages, which meets the
criterion for 33 or more languages (score 4).'}]}
```

[67]:

```
new_df = pd.DataFrame(out['results'])
new_df
```

```
[67]:    Website                              L4_Indicator  Assigned_Score  \
      0  ChatGPT  Multilingual support for intended locales               4

         Max_Score                                          Reasoning
      0          4  The page lists 39 supported languages, which m…
```

```
[68]: chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df,new_df])
      chat_gpt_test_results_df
```

```
[68]:    Website                              L4_Indicator Assigned_Score  \
      0  ChatGPT  Multilingual support for intended locales              4

        Max_Score                                          Reasoning
      0         4  The page lists 39 supported languages, which m…
```

3.2.3 L4: Motor and voice accessibility options (voice input, large targets, reduced precision)

This L4 indicator is functionally identical to the L4 Indicator “Senior motor and voice accessibility options (voice inputs, large targets, reduced precisions.) so scores given to that L4 will be duplicated for this L4 category.

3.3 L3 Subdimension: The AI supports accessibility & cognitive-load mitigation for seniors

3.3.1 L4: Senior focused simplification and stepwise assistance available

```
[69]: # Prompt for this L4 Indicator
      prompt_in = """
      This image shows the homepage for ChatGPT's chat model interaction interface.␣
        ↪You are tasked with evaluating the accessibility of help menu functionality␣
        ↪on this home page based on the unmodified presentation of the homepage upon␣
        ↪accessing it.

      Based only on what you see in the image of the homepage, rate the accessibility␣
        ↪of help functionality on the following scale ranging from 0 to 2:

      0: There are no obvious help icons or links present on the homepage.
      1: There are help icons or links available on the page, but they are not␣
        ↪immediately obvious (for example, they are presented with only an icon and␣
        ↪no text, or are in a very small size, smaller than 44 by 44 CSS pixels).
      2: A help icon or link is available on the home page and is easily visible with␣
        ↪text and in a size larger than 44 by 44 CSS pixels.

      The response should be in valid JSON format. The following example below shows␣
        ↪properly formatted output. The structure, Website, L4_Indicator and␣
        ↪Max_Score properties should not be changed. You are only writing the value␣
        ↪for Score and Reasoning in the response.
```

```
{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Senior-focused simplification & stepwise guidance␣
 ↪availabile",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

Print the results of the prompt for easy reading

[70]:

```
print(eval_result)
```

```
{
    "results": [
        {
            "Website": "ChatGPT",
            "L4_Indicator": "Senior-focused simplification & stepwise guidance
availabile",
            "Assigned_Score": 0,
            "Max_Score": 2,
            "Reasoning": "No help icons or links are present on the homepage.
The top-right corner contains user and settings icons, but these do not
constitute a help functionality and are not accompanied by descriptive text or
sized appropriately for accessibility."
        }
    ]
}
```

[71]:

```
out = js.loads(eval_result)
out
```

```
[71]: {'results': [{'Website': 'ChatGPT',
         'L4_Indicator': 'Senior-focused simplification & stepwise guidance
      availabile',
         'Assigned_Score': 0,
         'Max_Score': 2,
         'Reasoning': 'No help icons or links are present on the homepage. The top-
      right corner contains user and settings icons, but these do not constitute a
      help functionality and are not accompanied by descriptive text or sized
      appropriately for accessibility.'}]}
```

```
[72]: new_df = pd.DataFrame(out['results'])
      new_df
```

```
[72]:    Website                                          L4_Indicator  Assigned_Score  \
      0  ChatGPT  Senior-focused simplification & stepwise guida…               0

         Max_Score                                          Reasoning
      0          2  No help icons or links are present on the home…
```

```
[73]: chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df,new_df])
      chat_gpt_test_results_df
```

```
[73]:    Website                                          L4_Indicator Assigned_Score  \
      0  ChatGPT           Multilingual support for intended locales               4
      0  ChatGPT  Senior-focused simplification & stepwise guida…              0

        Max_Score                                          Reasoning
      0         4  The page lists 39 supported languages, which m…
      0         2  No help icons or links are present on the home…
```

3.3.2 L4: Senior motor and voice accessibility options (voice input, large targets, reduced precision)

```
[74]: l4_results = []
```

```
[75]: # Prompt for this L4 Indicator
      prompt_in = """
      This image shows the homepage for ChatGPT's chat model interaction interface.␣
        ↪You are tasked with evaluating the accessibility of this page in terms of␣
        ↪motor accessibility and voice input options.

      Based only on what you see in the image of the homepage, rate the webpage on␣
        ↪the following scale ranging from 0 to 2:

      0: All input targets (buttons, links, images, etc.) are below the WCAG␣
        ↪recommended size of 44 by 44 CSS pixels.
```

```
1: Some input targets (buttons, links, images, etc.) are below the WCAG␣
 ↪recommended size of 44 by 44 CSS pixels.
2: All input targets (buttons, links, images, etc.) are at or above the WCAG␣
 ↪recommended size of 44 by 44 CSS pixels.

The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Senior motor and voice accessibility options (voice␣
 ↪input, large targets, reduced precision)",
        "Assigned_Score":  ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[76]:

```
out = js.loads(eval_result)
out
```

[76]:

```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Senior motor and voice accessibility options (voice input,
large targets, reduced precision)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': "The microphone icon for voice input and the '+' button next to
'Ask anything' are likely smaller than 44x44 CSS pixels, indicating some input
targets are below the WCAG recommended size for motor accessibility."}]}
```

[77]:

```
l4_results.append(out)
```

[78]:

```
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪motor accessibility and voice input options.

Based only on what you see in the image of the homepage, rate the webpage on␣
  ↪the following scale ranging from 0 to 2:

0: The website provides no visible voice input accessibility options.
1: The website provides a voice input mode, but does not indicate it clearly␣
  ↪(ex. uses an image but does not label it with text)
2: The website provides a voice input mode that is clearly identifiable by both␣
  ↪image and text.

The response should be in JSON format. The following example below shows␣
  ↪properly formatted output. The structure, Website, L4_Indicator and␣
  ↪Max_Score properties should not be changed. You should insert your values␣
  ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
  ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Senior motor and voice accessibility options (voice␣
  ↪input, large targets, reduced precision)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[79]:

```
out = js.loads(eval_result)
out
```

[79]:

```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Senior motor and voice accessibility options (voice input,
large targets, reduced precision)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': "The page includes a microphone icon (image) for voice input,
but there is no explicit text label indicating its function (e.g., 'Voice Input'
or 'Speak'). While the microphone icon is a common visual cue for voice input,
the lack of accompanying text means it does not clearly indicate the voice input
mode as per the criteria for score 2."}]}
```

[80]:

```
l4_results.append(out)
```

Aggregate scores and reasoning for the different subsections evaluated for this L4 indicator

[81]:

```
combined_results = {}
combined_results["Website"] = ""
combined_results["L4_Indicator"] = ""
combined_results["Assigned_Score"] = 0
combined_results["Max_Score"] = 0
combined_results["Reasoning"] = ""

for json_obj in l4_results:
    for result in json_obj["results"]:
        for field in result:
            if field == "Assigned_Score":
                combined_results[field] += result[field]
            elif field == "Reasoning":
                combined_results[field] += " "
                combined_results[field] += result[field]
            elif field == "Max_Score":
                combined_results[field] += result[field]
            else:
                combined_results[field] = result[field]

print(combined_results)
```

```
{'Website': 'ChatGPT', 'L4_Indicator': 'Senior motor and voice accessibility
options (voice input, large targets, reduced precision)', 'Assigned_Score': 2,
'Max_Score': 4, 'Reasoning': " The microphone icon for voice input and the '+'
button next to 'Ask anything' are likely smaller than 44x44 CSS pixels,
indicating some input targets are below the WCAG recommended size for motor
accessibility. The page includes a microphone icon (image) for voice input, but
there is no explicit text label indicating its function (e.g., 'Voice Input' or
'Speak'). While the microphone icon is a common visual cue for voice input, the
lack of accompanying text means it does not clearly indicate the voice input
mode as per the criteria for score 2."}
```

[82]:
```
new_df = pd.DataFrame(combined_results, index=[0])
new_df
```

[82]:
```
   Website                                      L4_Indicator  Assigned_Score  \
0  ChatGPT  Senior motor and voice accessibility options (…               2

   Max_Score                                          Reasoning
0          4   The microphone icon for voice input and the '…
```

[83]:
```
chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df,new_df])
chat_gpt_test_results_df
```

[83]:
```
   Website                                      L4_Indicator Assigned_Score  \
0  ChatGPT          Multilingual support for intended locales               4
0  ChatGPT  Senior-focused simplification & stepwise guida…              0
0  ChatGPT  Senior motor and voice accessibility options (…              2

  Max_Score                                          Reasoning
0         4  The page lists 39 supported languages, which m…
0         2  No help icons or links are present on the home…
0         4   The microphone icon for voice input and the '…
```

3.3.3 L4: Plain-language & readability thresholds met (senior-appropriate)

[84]:
```
l4_results = []
```

[85]:
```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of this page in terms of␣
 ↪text spacing features.

Based only on what you see in the image of the homepage, rate the webpage on␣
 ↪the following scale ranging from 0 to 2:

0: All of the visible text items have line spacing below 1.5 times the font␣
 ↪size.
1: Some of the visible text items have line spacing below 1.5 times the font␣
 ↪size.
2: All of the visible text items have line spacing at or above 1.5 times the␣
 ↪font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
 ↪given and this should be mentioned in the reasonsing.
```

```
The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[86]:

```
eval_result
```

[86]:

```
'{\n    "results": [\n        {\n            "Website": "ChatGPT",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n            "Assigned_Score": 2,\n            "Max_Score": 2,\n
"Reasoning": "All visible text items with multiple lines (e.g., sidebar menu
items, \'Greg Knapp\' and \'Free\' in the bottom left) exhibit line spacing at
or above 1.5 times the font size. Single-line text elements (e.g., \'What\'s on
your mind today?\', \'Ask anything\', \'Get Plus\') do not require line spacing
evaluation, as line spacing applies to multi-line text. No visible text items
violate the line spacing threshold of 1.5x font size, resulting in a maximum
score of 2."\n        }\n    ]\n}'
```

[87]:

```
out = js.loads(eval_result)
out
```

[87]:
```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "All visible text items with multiple lines (e.g., sidebar menu
items, 'Greg Knapp' and 'Free' in the bottom left) exhibit line spacing at or
above 1.5 times the font size. Single-line text elements (e.g., 'What's on your
mind today?', 'Ask anything', 'Get Plus') do not require line spacing
evaluation, as line spacing applies to multi-line text. No visible text items
violate the line spacing threshold of 1.5x font size, resulting in a maximum
score of 2."}]}
```

[88]:
```
l4_results.append(out)
```

[89]:
```
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of this page in terms of␣
 ↪text spacing features.

Based only on what you see in the image of the homepage, rate the webpage on␣
 ↪the following scale ranging from 0 to 2:

0: All of the visible text items have spacing between paragraphs below 2 times␣
 ↪the font size.
1: Some of the visible text items have spacing between paragraphs below 2 times␣
 ↪the font size.
2: All of the visible text items have spacing between paragraphs at or above 2␣
 ↪times the font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
 ↪given and this should be mentioned in the reasonsing.

The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[90]:
```
eval_result
```

[90]:
```
'{\n    "results": [\n        {\n            "Website": "ChatGPT",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n            "Assigned_Score": 0,\n            "Max_Score": 2,\n
"Reasoning": "All visible text items (sidebar menu items, main heading, and
input field placeholder) have spacing between paragraphs below 2 times the font
size. The spacing between menu items in the left sidebar, between the main
heading and input field, and other adjacent text elements consistently fall
below the 2x font size threshold."\n        }\n    ]\n}'
```

[91]:
```
out = js.loads(eval_result)
out
```

[91]:
```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 0,
   'Max_Score': 2,
   'Reasoning': 'All visible text items (sidebar menu items, main heading, and
input field placeholder) have spacing between paragraphs below 2 times the font
size. The spacing between menu items in the left sidebar, between the main
heading and input field, and other adjacent text elements consistently fall
below the 2x font size threshold.'}]}
```

[92]:
```
l4_results.append(out)
```

[93]:
```
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪text spacing features.
```

```
Based only on what you see in the image of the homepage, rate the webpage on␣
  ↪the following scale ranging from 0 to 2:

0: All of the visible text characters have spacing between characters below 0.
  ↪12 times the font size.
1: Some of the visible text characters have spacing between characters below 0.
  ↪12 times the font size.
2: All of the visible text characters have spacing between characters at or␣
  ↪above 0.12 times the font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
  ↪given and this should be mentioned in the reasoning.

The response should be in JSON format. The following example below shows␣
  ↪properly formatted output. The structure, Website, L4_Indicator and␣
  ↪Max_Score properties should not be changed. You should insert your values␣
  ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
  ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Plain-language & readability thresholds met␣
  ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[96]:

```
eval_result
```

```
[96]: '{\n    "results": [\n        {\n            "Website": "ChatGPT",\n
      "L4_Indicator": "Plain-language & readability thresholds met (senior-
      appropriate)",\n            "Assigned_Score": 2,\n            "Max_Score": 2,\n
      "Reasoning": "The visible text elements in the ChatGPT homepage (e.g., menu
      items like \'New chat\', heading text, and input prompts) display adequate
      character spacing (kerning) consistent with standard accessibility design
      practices. No visible text elements exhibit character spacing below 0.12 times
      the font size, as typical UI text styling for readability and accessibility
      ensures sufficient inter-character spacing."\n        }\n    ]\n}'
```

```
[94]: out = js.loads(eval_result)
      out
```

```
[94]: {'results': [{'Website': 'ChatGPT',
         'L4_Indicator': 'Plain-language & readability thresholds met (senior-
      appropriate)',
         'Assigned_Score': 2,
         'Max_Score': 2,
         'Reasoning': "The visible text elements in the ChatGPT homepage (e.g., menu
      items like 'New chat', heading text, and input prompts) display adequate
      character spacing (kerning) consistent with standard accessibility design
      practices. No visible text elements exhibit character spacing below 0.12 times
      the font size, as typical UI text styling for readability and accessibility
      ensures sufficient inter-character spacing."}]}
```

```
[95]: l4_results.append(out)
```

```
[97]: prompt_in = """
      This image shows the homepage for ChatGPT's chat model interaction interface.␣
        ↪You are tasked with evaluating the accessibility of this page in terms of␣
        ↪text spacing features.

      Based only on what you see in the image of the homepage, rate the webpage on␣
        ↪the following scale ranging from 0 to 2:

      0: All of the visible text words have spacing between words below 0.16 times␣
        ↪the font size.
      1: Some of the visible text words have spacing between words below 0.16 times␣
        ↪the font size.
      2: All of the visible text words have spacing between words below 0.16 times␣
        ↪the font size.

      If any criteria is not applicable to the webpage, the maximum score should be␣
        ↪given and this should be mentioned in the reasoning.
```

```
The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[100]:

```
eval_result
```

[100]:

```
'{\n    "results": [\n        {\n            "Website": "ChatGPT",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n            "Assigned_Score": 2,\n            "Max_Score": 2,\n
"Reasoning": "The visible text elements (e.g., \\"What\'s on your mind
today?\\", \\"Ask anything\\", sidebar items like \\"New chat\\", \\"Search
chats\\", and footer elements) show adequate word spacing. In standard UI
design, proper word spacing typically meets or exceeds the 0.16× font size
threshold for readability. The spacing between words appears sufficient,
indicating no words fall below the 0.16× font size criterion, thus qualifying
for a maximum score of 2."\n        }\n    ]\n}'
```

[98]:

```
out = js.loads(eval_result)
out
```

[98]:

```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': 'The visible text elements (e.g., "What\'s on your mind today?",
"Ask anything", sidebar items like "New chat", "Search chats", and footer
elements) show adequate word spacing. In standard UI design, proper word spacing
typically meets or exceeds the 0.16× font size threshold for readability. The
spacing between words appears sufficient, indicating no words fall below the
0.16× font size criterion, thus qualifying for a maximum score of 2.'}]}
```

[99]:

```
l4_results.append(out)
```

[101]:

```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ChatGPT's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪the reading level of the text.

Based only on what you see in the image of the homepage, rate the webpage on␣
  ↪the following scale ranging from 0 to 2:

0: All of the visible text is above a 7th grade reading level.
1: Some of the visible text is above a 7th grade reading level
2: All of the visible text is at or below a 7th grade reading level.

If any criteria is not applicable to the webpage, the maximum score should be␣
  ↪given and this should be mentioned in the reasoning.

The response should be in JSON format. The following example below shows␣
  ↪properly formatted output. The structure, Website, L4_Indicator and␣
  ↪Max_Score properties should not be changed. You should insert your values␣
  ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
  ↪the image.

{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Plain-language & readability thresholds met␣
  ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[102]:
```
eval_result
```

[102]:
```
'{\n    "results": [\n        {\n        "Website": "ChatGPT",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n        "Assigned_Score": 2,\n        "Max_Score": 2,\n
"Reasoning": "All visible text on the page (e.g., \'What\'s on your mind
today?\', \'Ask anything\', navigation items like \'New chat\', \'Search
chats\', \'Library\', \'Projects\') uses simple, common vocabulary and short
phrases that are easily understood at a basic reading level. None of the text
appears to require a 7th grade reading level or higher; the language is
conversational and accessible to younger readers."\n        }\n    ]\n}'
```

[103]:
```
out = js.loads(eval_result)
out
```

[103]:
```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "All visible text on the page (e.g., 'What's on your mind
today?', 'Ask anything', navigation items like 'New chat', 'Search chats',
'Library', 'Projects') uses simple, common vocabulary and short phrases that are
easily understood at a basic reading level. None of the text appears to require
a 7th grade reading level or higher; the language is conversational and
accessible to younger readers."}]}
```

[104]:
```
l4_results.append(out)
```

Aggregate scores and reasoning for the different subsections evaluated for this L4 indicator

[105]:
```
combined_results = {}
combined_results["Website"] = ""
combined_results["L4_Indicator"] = ""
combined_results["Assigned_Score"] = 0
```

```python
combined_results["Max_Score"] = 0
combined_results["Reasoning"] = ""

for json_obj in l4_results:
    for result in json_obj["results"]:
        for field in result:
            if field == "Assigned_Score":
                combined_results[field] += result[field]
            elif field == "Reasoning":
                combined_results[field] += " "
                combined_results[field] += result[field]
            elif field == "Max_Score":
                combined_results[field] += result[field]
            else:
                combined_results[field] = result[field]

print(combined_results)
```

```
{'Website': 'ChatGPT', 'L4_Indicator': 'Plain-language & readability thresholds
met (senior-appropriate)', 'Assigned_Score': 8, 'Max_Score': 10, 'Reasoning': '
All visible text items with multiple lines (e.g., sidebar menu items, \'Greg
Knapp\' and \'Free\' in the bottom left) exhibit line spacing at or above 1.5
times the font size. Single-line text elements (e.g., \'What\'s on your mind
today?\', \'Ask anything\', \'Get Plus\') do not require line spacing
evaluation, as line spacing applies to multi-line text. No visible text items
violate the line spacing threshold of 1.5x font size, resulting in a maximum
score of 2. All visible text items (sidebar menu items, main heading, and input
field placeholder) have spacing between paragraphs below 2 times the font size.
The spacing between menu items in the left sidebar, between the main heading and
input field, and other adjacent text elements consistently fall below the 2x
font size threshold. The visible text elements in the ChatGPT homepage (e.g.,
menu items like \'New chat\', heading text, and input prompts) display adequate
character spacing (kerning) consistent with standard accessibility design
practices. No visible text elements exhibit character spacing below 0.12 times
the font size, as typical UI text styling for readability and accessibility
ensures sufficient inter-character spacing. The visible text elements (e.g.,
"What\'s on your mind today?", "Ask anything", sidebar items like "New chat",
"Search chats", and footer elements) show adequate word spacing. In standard UI
design, proper word spacing typically meets or exceeds the 0.16× font size
threshold for readability. The spacing between words appears sufficient,
indicating no words fall below the 0.16× font size criterion, thus qualifying
for a maximum score of 2. All visible text on the page (e.g., \'What\'s on your
mind today?\', \'Ask anything\', navigation items like \'New chat\', \'Search
chats\', \'Library\', \'Projects\') uses simple, common vocabulary and short
phrases that are easily understood at a basic reading level. None of the text
appears to require a 7th grade reading level or higher; the language is
conversational and accessible to younger readers.'}
```

```
[106]: new_df = pd.DataFrame(combined_results, index=[0])
       new_df
```

```
[106]:    Website                                        L4_Indicator  Assigned_Score  \
       0  ChatGPT  Plain-language & readability thresholds met (s…               8

          Max_Score                                          Reasoning
       0         10   All visible text items with multiple lines (e…
```

```
[107]: chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df,new_df])
       chat_gpt_test_results_df
```

```
[107]:    Website                                        L4_Indicator Assigned_Score  \
       0  ChatGPT           Multilingual support for intended locales               4
       0  ChatGPT  Senior-focused simplification & stepwise guida…             0
       0  ChatGPT  Senior motor and voice accessibility options (…             2
       0  ChatGPT  Plain-language & readability thresholds met (s…             8

         Max_Score                                          Reasoning
       0         4   The page lists 39 supported languages, which m…
       0         2   No help icons or links are present on the home…
       0         4    The microphone icon for voice input and the '…
       0        10    All visible text items with multiple lines (e…
```

3.3.4 L4: Cognitive-load reduction patterns present (chunking, progressive disclosure)

```
[108]: # Prompt for this L4 Indicator
       prompt_in = """
       This image shows the homepage for ChatGPT's chat model interaction interface.␣
         ↪You are tasked with evaluating the accessibility of this page in terms of␣
         ↪cognitive load reduction through proper header and label text.

       Based only on what you see in the image of the homepage, rate the accessibility␣
         ↪of help functionality on the following scale ranging from 0 to 2:

       0: All visible text headers and labels do not clearly explain the purpose of␣
         ↪their associated interactive elements.
       1: Some visible text headers and labels do not clearly explain the purpose of␣
         ↪their associated interactive elements.
       2: All visible text headers and labels clearly explain the purpose of their␣
         ↪associated interactive elements.

       The response should be in valid JSON format. The following example below shows␣
         ↪properly formatted output. The structure, Website, L4_Indicator and␣
         ↪Max_Score properties should not be changed. You are only writing the value␣
         ↪for Score and Reasoning in the response.
```

```
{
    "results": [
        {
        "Website": "ChatGPT",
        "L4_Indicator": "Cognitive-load reduction patterns present (chunking,␣
 ↪progressive disclosure)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[chatgpt_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[109]:
```
eval_result
```

[109]:
```
'{\n    "results": [\n        {\n        "Website": "ChatGPT",\n
"L4_Indicator": "Cognitive-load reduction patterns present (chunking,
progressive disclosure)",\n        "Assigned_Score": 2,\n        "Max_Score":
2,\n        "Reasoning": "All visible text headers and labels (e.g., \'New
chat\', \'Search chats\', \'Ask anything\') clearly explain the purpose of their
associated interactive elements, reducing cognitive load by providing immediate
context for user actions."\n        }\n    ]\n}'
```

[110]:
```
out = js.loads(eval_result)
out
```

[110]:
```
{'results': [{'Website': 'ChatGPT',
   'L4_Indicator': 'Cognitive-load reduction patterns present (chunking,
progressive disclosure)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "All visible text headers and labels (e.g., 'New chat', 'Search
chats', 'Ask anything') clearly explain the purpose of their associated
interactive elements, reducing cognitive load by providing immediate context for
user actions."}]}
```

Append results to the final dataframe for comparison later

[112]:

```
new_df = pd.DataFrame(out['results'])
new_df
```

[112]:

```
   Website                                        L4_Indicator  Assigned_Score  \
0  ChatGPT  Cognitive-load reduction patterns present (chu…                2

   Max_Score                                           Reasoning
0          2  All visible text headers and labels (e.g., 'Ne…
```

[113]:

```
chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df,new_df])
chat_gpt_test_results_df
```

[113]:

```
   Website                                        L4_Indicator Assigned_Score  \
0  ChatGPT           Multilingual support for intended locales              4
0  ChatGPT  Senior-focused simplification & stepwise guida…               0
0  ChatGPT  Senior motor and voice accessibility options (…               2
0  ChatGPT  Plain-language & readability thresholds met (s…               8
0  ChatGPT  Cognitive-load reduction patterns present (chu…               2

  Max_Score                                           Reasoning
0         4  The page lists 39 supported languages, which m…
0         2  No help icons or links are present on the home…
0         4   The microphone icon for voice input and the '…
0        10   All visible text items with multiple lines (e…
0         2  All visible text headers and labels (e.g., 'Ne…
```

3.4 Finalize ChatGPT Test Results

[115]:

```
chat_gpt_test_results_df.reset_index(inplace=True)
chat_gpt_test_results_df
```

[115]:

```
   index  Website                                        L4_Indicator  \
0      0  ChatGPT           Multilingual support for intended locales
1      0  ChatGPT  Senior-focused simplification & stepwise guida…
2      0  ChatGPT  Senior motor and voice accessibility options (…
3      0  ChatGPT  Plain-language & readability thresholds met (s…
4      0  ChatGPT  Cognitive-load reduction patterns present (chu…

  Assigned_Score Max_Score                                           Reasoning
0              4         4  The page lists 39 supported languages, which m…
1              0         2  No help icons or links are present on the home…
2              2         4   The microphone icon for voice input and the '…
3              8        10   All visible text items with multiple lines (e…
4              2         2  All visible text headers and labels (e.g., 'Ne…
[116]: chat_gpt_test_results_df.drop(columns=['index'], inplace=True)
       chat_gpt_test_results_df
```

```
[116]:    Website                                     L4_Indicator Assigned_Score  \
       0  ChatGPT          Multilingual support for intended locales              4
       1  ChatGPT  Senior-focused simplification & stepwise guida…             0
       2  ChatGPT  Senior motor and voice accessibility options (…             2
       3  ChatGPT  Plain-language & readability thresholds met (s…             8
       4  ChatGPT  Cognitive-load reduction patterns present (chu…             2

         Max_Score                                          Reasoning
       0         4  The page lists 39 supported languages, which m…
       1         2  No help icons or links are present on the home…
       2         4   The microphone icon for voice input and the '…
       3        10   All visible text items with multiple lines (e…
       4         2  All visible text headers and labels (e.g., 'Ne…
```

3.4.1 Duplicate L4 Motor and voice accessibility options (voice input, large targets, reduced precision) L4

Since the L4 for Senior motor and voice accessibility options (voice input, large targets, reduced precision) was essentially identical to the L4 in the other L3 Subcategory, the score for the senior focused L4 is duplicated to reflect its presence in two separate L4s on the tree

```
[120]: # Replicate each row 3 times
       seniors_row = chat_gpt_test_results_df.iloc[[2]]
       seniors_row['L4_Indicator'] = "Motor and voice accessibility options (voice␣
        ↪input, large targets, reduced precision)"
       seniors_row
```

```
/tmp/ipython-input-623812384.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  seniors_row['L4_Indicator'] = "Motor and voice accessibility options (voice
input, large targets, reduced precision)"
```

```
[120]:    Website                                     L4_Indicator Assigned_Score  \
       2  ChatGPT  Motor and voice accessibility options (voice i…             2

         Max_Score                                          Reasoning
       2         4   The microphone icon for voice input and the '…
```

```
[125]: chat_gpt_test_results_df = pd.concat([chat_gpt_test_results_df, seniors_row]).
        ↪reset_index().drop(columns=['index'])
       chat_gpt_test_results_df
```

```
[125]:    Website                                       L4_Indicator Assigned_Score  \
       0  ChatGPT          Multilingual support for intended locales              4
       1  ChatGPT  Senior-focused simplification & stepwise guida…             0
       2  ChatGPT  Senior motor and voice accessibility options (…             2
       3  ChatGPT  Plain-language & readability thresholds met (s…             8
       4  ChatGPT  Cognitive-load reduction patterns present (chu…             2
       5  ChatGPT  Motor and voice accessibility options (voice i…             2

         Max_Score                                          Reasoning
       0         4  The page lists 39 supported languages, which m…
       1         2  No help icons or links are present on the home…
       2         4   The microphone icon for voice input and the '…
       3        10   All visible text items with multiple lines (e…
       4         2  All visible text headers and labels (e.g., 'Ne…
       5         4   The microphone icon for voice input and the '…
```

3.4.2 Return to L4 WCAG-aligned accessibility features available L4

Now that we have completed the other L4 indicators, we can aggregate them to get a score for the WCAG compliance metric

```
[126]: wcag_alignment_assigned_score = chat_gpt_test_results_df['Assigned_Score'].sum()
       wcag_alignment_max_score = chat_gpt_test_results_df['Max_Score'].sum()
```

```
[127]: wcag_alignment_values = {
           "Website": "ChatGPT",
           "L4_Indicator": "WCAG-aligned accessibility features available",
           "Assigned_Score": wcag_alignment_assigned_score,
           "Max_Score": wcag_alignment_max_score,
           "Reasoning": "This value is the summation of all the other L4 indicators␣
        ↪handled in this test, since they are meant to be aligned with WCAG guidance."
       }
```

```
[129]: new_df = pd.DataFrame(wcag_alignment_values, index=[6])
       new_df
```

```
[129]:    Website                                   L4_Indicator  Assigned_Score  \
       6  ChatGPT  WCAG-aligned accessibility features available              18

          Max_Score                                          Reasoning
       6         26  This value is the summation of all the other L…
```

```
[130]: final_chat_gpt_scores_df = pd.concat([chat_gpt_test_results_df, new_df])
```

```
[131]: final_chat_gpt_scores_df
```

```
[131]:    Website                                       L4_Indicator Assigned_Score  \
       0  ChatGPT          Multilingual support for intended locales              4
```

```
1  ChatGPT  Senior-focused simplification & stepwise guida…              0
2  ChatGPT  Senior motor and voice accessibility options (…              2
3  ChatGPT  Plain-language & readability thresholds met (s…              8
4  ChatGPT  Cognitive-load reduction patterns present (chu…              2
5  ChatGPT  Motor and voice accessibility options (voice i…              2
6  ChatGPT      WCAG-aligned accessibility features available             18

  Max_Score                                             Reasoning
0         4  The page lists 39 supported languages, which m…
1         2  No help icons or links are present on the home…
2         4   The microphone icon for voice input and the '…
3        10   All visible text items with multiple lines (e…
4         2  All visible text headers and labels (e.g., 'Ne…
5         4   The microphone icon for voice input and the '…
6        26  This value is the summation of all the other L…
```

4 Test L4 Indicators using Qwen3-vl:2B - ClaudeAI

4.1 Create dataframe for storing Claude test results

[162]:
```
# Create a dataframe to hold ClaudeAI test results
claudeai_test_results_df = pd.DataFrame(columns=["Website", "L4_Indicator",␣
 ↪"Assigned_Score", "Max_Score", "Reasoning"])
```

4.2 L3 Subdimension: The AI is accessible and inclusive across abilities and language

4.2.1 L4: WCAG-aligned accessibility features available

This L4 category covers a wide variety of possible accessibility features as defined in the WCAG 2.1 Guidelines. Given my work as an individual and not a group for this project, the scope of these guidelines is too broad to be sufficiently covered by this work.

Instead, I have chosen to evalute this L4 category by making it a composite score of all the following L4 categories that are evalutated by this notebook. The reasoning for this being that every rating that is being assigned is based on one or more off the Success Criterias listed in the WCAG version 2.1 guidelines, so they fall under the umbrella of "WCAG-aligned accessibility features."

This notebook will act as a proof of concept that additional guidelines and features can be implemented into this evaluation pipeline, following the same structure and build up this composite WCAG-alignment score, so long as the ratings are based on WCAG-guidance as I have done for this project.

As a result, this section will be evaluated again at the end of this notebook once all other L4s have been scored. It is placed here for the time being to remain within the logical grouping as defined in the AI Ethics Index Tree (under the L3 Subdimension: The AI is accessible and inclusive accross abilities and langauges).

4.2.2 L4: Multilingual support for intended locales

[163]:
```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ClaudeAIs's chat model interaction interface␣
 ↪help page for ClaudeAI regarding what languages the platform is localized␣
 ↪into. Your task is to scan the HTML page for the list of supported␣
 ↪langauges, count the number, then score the page based on the number of␣
 ↪supported websites on the following scale:

0: The website supports 5 or fewer languages
1: The website supports 10 or fewer languages
2: The website supports 20 or fewer languages
3: The website supports 32 or fewer languages
4: The website supports 33 or more languages.

The response should be written in valid JSON format. The following example␣
 ↪below shows properly formatted output. The structure, Website, L4_Indicator␣
 ↪and Max_Score properties should not be changed. You are only writing the␣
 ↪value for Score and Reasoning in the response.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Multilingual support for intended locales",
        "Assigned_Score": ,
        "Max_Score": 4,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_languages_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[164]:
```
eval_result
```

```
[164]: '{\n    "results": [\n        {\n            "Website": "ClaudeAI",\n
       "L4_Indicator": "Multilingual support for intended locales",\n
       "Assigned_Score": 2,\n            "Max_Score": 4,\n            "Reasoning": "The
       page lists 11 supported languages, which is within the 20 or fewer languages
       category."\n        }\n    ]\n}'
```

```
[165]: out = js.loads(eval_result)
       out
```

```
[165]: {'results': [{'Website': 'ClaudeAI',
          'L4_Indicator': 'Multilingual support for intended locales',
          'Assigned_Score': 2,
          'Max_Score': 4,
          'Reasoning': 'The page lists 11 supported languages, which is within the 20
       or fewer languages category.'}]}
```

```
[166]: new_df = pd.DataFrame(out['results'])
       new_df
```

```
[166]:     Website                               L4_Indicator  Assigned_Score  \
       0  ClaudeAI  Multilingual support for intended locales               2

          Max_Score                                          Reasoning
       0          4  The page lists 11 supported languages, which i…
```

```
[167]: claudeai_test_results_df = pd.concat([claudeai_test_results_df,new_df])
       claudeai_test_results_df
```

```
[167]:     Website                               L4_Indicator Assigned_Score  \
       0  ClaudeAI  Multilingual support for intended locales              2

         Max_Score                                          Reasoning
       0          4  The page lists 11 supported languages, which i…
```

4.2.3 L4: Motor and voice accessibility options (voice input, large targets, reduced precision)

This L4 indicator is functionally identical to the L4 Indicator “Senior motor and voice accessibility options (voice inputs, large targets, reduced precisions.) so scores given to that L4 will be duplicated for this L4 category.

4.3 L3 Subdimension: The AI supports accessibility & cognitive-load mitigation for seniors

4.3.1 L4: Senior focused simplification and stepwise assistance available

[168]:

```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ClaudeAIs's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of help menu functionality␣
 ↪on this home page based on the unmodified presentation of the homepage upon␣
 ↪accessing it.

Based only on what you see in the image of the homepage, rate the accessibility␣
 ↪of help functionality on the following scale ranging from 0 to 2:

0: There are no obvious help icons or links present on the homepage.
1: There are help icons or links available on the page, but they are not␣
 ↪immediately obvious (for example, they are presented with only an icon and␣
 ↪no text, or are in a very small size, smaller than 44 by 44 CSS pixels).
2: A help icon or link is available on the home page and is easily visible with␣
 ↪text and in a size larger than 44 by 44 CSS pixels.

The response should be in valid JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You are only writing the value␣
 ↪for Score and Reasoning in the response.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Senior-focused simplification & stepwise guidance␣
 ↪availabile",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)
```

```
# Save the model's response as json
eval_result = response['response']
```

[169]:
```
print(eval_result)
```

```
{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Senior-focused simplification & stepwise guidance
availabile",
        "Assigned_Score": 2,
        "Max_Score": 2,
        "Reasoning": "The 'Help me write' button is clearly visible with text
and serves as a help-related link. It meets the criteria of being easily visible
with text and sized larger than 44 by 44 CSS pixels, as standard interactive
elements in web interfaces typically adhere to touch target sizes (44x44 or
larger)."
        }
    ]
}
```

[170]:
```
out = js.loads(eval_result)
out
```

[170]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Senior-focused simplification & stepwise guidance
availabile',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "The 'Help me write' button is clearly visible with text and
serves as a help-related link. It meets the criteria of being easily visible
with text and sized larger than 44 by 44 CSS pixels, as standard interactive
elements in web interfaces typically adhere to touch target sizes (44x44 or
larger)."}]}
```

[171]:
```
new_df = pd.DataFrame(out['results'])
new_df
```

[171]:
```
    Website                                       L4_Indicator  \
0  ClaudeAI  Senior-focused simplification & stepwise guida…

   Assigned_Score  Max_Score  \
0               2          2

                                           Reasoning
0  The 'Help me write' button is clearly visible …
```

[172]:
```
claudeai_test_results_df = pd.concat([claudeai_test_results_df,new_df])
claudeai_test_results_df
```

[172]:
```
    Website                                      L4_Indicator Assigned_Score  \
0  ClaudeAI          Multilingual support for intended locales              2
0  ClaudeAI  Senior-focused simplification & stepwise guida…              2

  Max_Score                                         Reasoning
0         4  The page lists 11 supported languages, which i…
0         2  The 'Help me write' button is clearly visible …
```

4.3.2 L4: Senior motor and voice accessibility options (voice input, large targets, reduced precision)

[173]:
```
l4_results = []
```

[174]:
```
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of this page in terms of␣
 ↪motor accessibility and voice input options.

Based only on what you see in the image of the homepage, rate the webpage on␣
 ↪the following scale ranging from 0 to 2:

0: All input targets (buttons, links, images, etc.) are below the WCAG␣
 ↪recommended size of 44 by 44 CSS pixels.
1: Some input targets (buttons, links, images, etc.) are below the WCAG␣
 ↪recommended size of 44 by 44 CSS pixels.
2: All input targets (buttons, links, images, etc.) are at or above the WCAG␣
 ↪recommended size of 44 by 44 CSS pixels.

The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Senior motor and voice accessibility options (voice␣
 ↪input, large targets, reduced precision)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[175]:
```
out = js.loads(eval_result)
out
```

[175]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Senior motor and voice accessibility options (voice input,
large targets, reduced precision)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': "Several input targets such as 'Help me write', 'Learn about',
'Analyze Image', 'Summarize text', and '+ See More' buttons appear to be smaller
than the 44x44 CSS pixel recommendation for touch targets. While elements like
'Start new' and voice input icons may meet or exceed the size requirement, the
presence of multiple smaller buttons indicates 'some input targets are below the
WCAG recommended size'."}]}
```

[176]:
```
l4_results.append(out)
```

[177]:
```
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪motor accessibility and voice input options.

Based only on what you see in the image of the homepage, rate the webpage on␣
  ↪the following scale ranging from 0 to 2:

0: The website provides no visible voice input accessibility options.
1: The website provides a voice input mode, but does not indicate it clearly␣
  ↪(ex. uses an image but does not label it with text)
2: The website provides a voice input mode that is clearly identifiable by both␣
  ↪image and text.

```

```
The response should be in JSON format. The following example below shows␣
  ↪properly formatted output. The structure, Website, L4_Indicator and␣
  ↪Max_Score properties should not be changed. You should insert your values␣
  ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
  ↪the image.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Senior motor and voice accessibility options (voice␣
  ↪input, large targets, reduced precision)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[178]:

```
out = js.loads(eval_result)
out
```

[178]:

```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Senior motor and voice accessibility options (voice input,
large targets, reduced precision)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': 'The page includes a microphone icon (image) for voice input,
but no accompanying text label to explicitly indicate its function. While the
microphone icon is a standard visual cue for voice input, the absence of
descriptive text means it is not clearly identifiable by both image and
text.'}]}
```

[179]:

```
l4_results.append(out)
```

Aggregate scores and reasoning for the different subsections evaluated for this L4 indicator

[180]:

```python
combined_results = {}
combined_results["Website"] = ""
combined_results["L4_Indicator"] = ""
combined_results["Assigned_Score"] = 0
combined_results["Max_Score"] = 0
combined_results["Reasoning"] = ""

for json_obj in l4_results:
    for result in json_obj["results"]:
        for field in result:
            if field == "Assigned_Score":
                combined_results[field] += result[field]
            elif field == "Reasoning":
                combined_results[field] += " "
                combined_results[field] += result[field]
            elif field == "Max_Score":
                combined_results[field] += result[field]
            else:
                combined_results[field] = result[field]

print(combined_results)
```

```
{'Website': 'ClaudeAI', 'L4_Indicator': 'Senior motor and voice accessibility
options (voice input, large targets, reduced precision)', 'Assigned_Score': 2,
'Max_Score': 4, 'Reasoning': " Several input targets such as 'Help me write',
'Learn about', 'Analyze Image', 'Summarize text', and '+ See More' buttons
appear to be smaller than the 44x44 CSS pixel recommendation for touch targets.
While elements like 'Start new' and voice input icons may meet or exceed the
size requirement, the presence of multiple smaller buttons indicates 'some input
targets are below the WCAG recommended size'. The page includes a microphone
icon (image) for voice input, but no accompanying text label to explicitly
indicate its function. While the microphone icon is a standard visual cue for
voice input, the absence of descriptive text means it is not clearly
identifiable by both image and text."}
```

[181]:

```python
new_df = pd.DataFrame(combined_results, index=[0])
new_df
```

[181]:

```
    Website                                       L4_Indicator  \
0  ClaudeAI  Senior motor and voice accessibility options (…

   Assigned_Score  Max_Score  \
0               2          4

                                           Reasoning
0   Several input targets such as 'Help me write'…
```

```
[182]: claudeai_test_results_df = pd.concat([claudeai_test_results_df,new_df])
       claudeai_test_results_df
```

```
[182]:     Website                                       L4_Indicator Assigned_Score  \
       0  ClaudeAI          Multilingual support for intended locales              2
       0  ClaudeAI  Senior-focused simplification & stepwise guida…              2
       0  ClaudeAI  Senior motor and voice accessibility options (…              2

         Max_Score                                          Reasoning
       0         4  The page lists 11 supported languages, which i…
       0         2  The 'Help me write' button is clearly visible …
       0         4   Several input targets such as 'Help me write'…
```

4.3.3 L4: Plain-language & readability thresholds met (senior-appropriate)

```
[ ]: l4_results = []
```

```
[183]: # Prompt for this L4 Indicator
       prompt_in = """
       This image shows the homepage for ClaudeAI's chat model interaction interface.␣
         ↪You are tasked with evaluating the accessibility of this page in terms of␣
         ↪text spacing features.

       Based only on what you see in the image of the homepage, rate the webpage on␣
         ↪the following scale ranging from 0 to 2:

       0: All of the visible text items have line spacing below 1.5 times the font␣
         ↪size.
       1: Some of the visible text items have line spacing below 1.5 times the font␣
         ↪size.
       2: All of the visible text items have line spacing at or above 1.5 times the␣
         ↪font size.

       If any criteria is not applicable to the webpage, the maximum score should be␣
         ↪given and this should be mentioned in the reasonsing.

       The response should be in JSON format. The following example below shows␣
         ↪properly formatted output. The structure, Website, L4_Indicator and␣
         ↪Max_Score properties should not be changed. You should insert your values␣
         ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
         ↪the image.

       {
           "results": [
               {
               "Website": "ClaudeAI",
            "L4_Indicator": "Plain-language & readability thresholds met (senior-appropriate)",
            "Assigned_Score": ,
            "Max_Score": 2,
            "Reasoning":
            }
        ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[184]:
```
eval_result
```

[184]:
```
'{\n    "results": [\n        {\n            "Website": "ClaudeAI",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n            "Assigned_Score": 2,\n            "Max_Score": 2,\n
"Reasoning": "Line spacing is not applicable to any visible text items, as all
text elements (e.g., headings, buttons, input placeholders) are single-line. The
criteria requires evaluating line spacing (vertical space between multiple lines
of text), which does not exist here. Thus, the criteria is not applicable, and
the maximum score is assigned."\n        }\n    ]\n}'
```

[185]:
```
out = js.loads(eval_result)
out
```

[185]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': 'Line spacing is not applicable to any visible text items, as
all text elements (e.g., headings, buttons, input placeholders) are single-line.
The criteria requires evaluating line spacing (vertical space between multiple
lines of text), which does not exist here. Thus, the criteria is not applicable,
and the maximum score is assigned.'}]}
```

[186]:
```
l4_results.append(out)
```

[187]:

```
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of this page in terms of␣
 ↪text spacing features.

Based only on what you see in the image of the homepage, rate the webpage on␣
 ↪the following scale ranging from 0 to 2:

0: All of the visible text items have spacing between paragraphs below 2 times␣
 ↪the font size.
1: Some of the visible text items have spacing between paragraphs below 2 times␣
 ↪the font size.
2: All of the visible text items have spacing between paragraphs at or above 2␣
 ↪times the font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
 ↪given and this should be mentioned in the reasonsing.

The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
```

```
eval_result = response['response']
```

[188]:
```
eval_result
```

[188]:
```
'{\n    "results": [\n        {\n        "Website": "ClaudeAI",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n        "Assigned_Score": 1,\n        "Max_Score": 2,\n
"Reasoning": "Some visible text items (e.g., spacing between \'No Chat History\'
and \'Gregory Knapp\' in the sidebar) have minimal vertical spacing that appears
to be below 2 times the font size. This indicates that not all text spacing
meets the threshold of at or above 2 times the font size."\n        }\n    ]\n}'
```

[189]:
```
out = js.loads(eval_result)
out
```

[189]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': "Some visible text items (e.g., spacing between 'No Chat
History' and 'Gregory Knapp' in the sidebar) have minimal vertical spacing that
appears to be below 2 times the font size. This indicates that not all text
spacing meets the threshold of at or above 2 times the font size."}]}
```

[190]:
```
l4_results.append(out)
```

[191]:
```
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪text spacing features.

Based only on what you see in the image of the homepage, rate the webpage on␣
  ↪the following scale ranging from 0 to 2:

0: All of the visible text characters have spacing between characters below 0.
  ↪12 times the font size.
1: Some of the visible text characters have spacing between characters below 0.
  ↪12 times the font size.
2: All of the visible text characters have spacing between characters at or␣
  ↪above 0.12 times the font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
  ↪given and this should be mentioned in the reasonsing.

```

```
The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[192]:
```
out = js.loads(eval_result)
out
```

[192]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "All visible text elements (e.g., 'How can I help you?', 'Type a
message…', menu options) exhibit standard, non-restricted letter spacing. The
UI follows typical web design conventions where character spacing exceeds 0.12×
font size, ensuring readability without visible character crowding or
overlapping."}]}
```

[193]:
```
l4_results.append(out)
```

[194]:

```
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
 ↪You are tasked with evaluating the accessibility of this page in terms of␣
 ↪text spacing features.

Based only on what you see in the image of the homepage, rate the webpage on␣
 ↪the following scale ranging from 0 to 2:

0: All of the visible text words have spacing between words below 0.16 times␣
 ↪the font size.
1: Some of the visible text words have spacing between words below 0.16 times␣
 ↪the font size.
2: All of the visible text words have spacing between words below 0.16 times␣
 ↪the font size.

If any criteria is not applicable to the webpage, the maximum score should be␣
 ↪given and this should be mentioned in the reasonsing.

The response should be in JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You should insert your values␣
 ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
 ↪the image.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning":
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
```

```
eval_result = response['response']
```

```
[195]: out = js.loads(eval_result)
       out
```

```
[195]: {'results': [{'Website': 'ClaudeAI',
          'L4_Indicator': 'Plain-language & readability thresholds met (senior-
       appropriate)',
          'Assigned_Score': 0,
          'Max_Score': 2,
          'Reasoning': "Based on the visible text elements (e.g., 'How can I help
       you?', 'Type a message…', button labels), the spacing between words is below
       0.16 times the font size for all visible text. For example, with a typical font
       size of 14px for body text, 0.16 * 14 = 2.24px; standard word spacing of ~2px is
       below this threshold. Similarly, larger headings (e.g., 24px font size) have
       word spacing below 0.16x. Thus, all visible text words meet the condition for a
       score of 0."}]}
```

```
[196]: l4_results.append(out)
```

```
[197]: # Prompt for this L4 Indicator
       prompt_in = """
       This image shows the homepage for ClaudeAI's chat model interaction interface.␣
         ↪You are tasked with evaluating the accessibility of this page in terms of␣
         ↪the reading level of the text.

       Based only on what you see in the image of the homepage, rate the webpage on␣
         ↪the following scale ranging from 0 to 2:

       0: All of the visible text is above a 7th grade reading level.
       1: Some of the visible text is above a 7th grade reading level
       2: All of the visible text is at or below a 7th grade reading level.

       If any criteria is not applicable to the webpage, the maximum score should be␣
         ↪given and this should be mentioned in the reasonsing.

       The response should be in JSON format. The following example below shows␣
         ↪properly formatted output. The structure, Website, L4_Indicator and␣
         ↪Max_Score properties should not be changed. You should insert your values␣
         ↪for Assigned_Score and Reasoning based on the results of your analysis of␣
         ↪the image.

       {
           "results": [
               {
               "Website": "ClaudeAI",
            "L4_Indicator": "Plain-language & readability thresholds met␣
 ↪(senior-appropriate)",
            "Assigned_Score": ,
            "Max_Score": 2,
            "Reasoning":
            }
        ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[198]:

```
eval_result
```

[198]:

```
'{\n    "results": [\n        {\n        "Website": "ClaudeAI",\n
"L4_Indicator": "Plain-language & readability thresholds met (senior-
appropriate)",\n        "Assigned_Score": 1,\n        "Max_Score": 2,\n
"Reasoning": "The visible text includes \'Analyze Image\', which contains the
word \'Analyze\' (Flesch-Kincaid Grade Level ~7.5), placing it above a 7th grade
reading level. All other visible text elements (e.g., \'Start new\', \'How can I
help you?\', \'Create an image\', \'Help me write\', etc.) are at or below the
7th grade level. Thus, some visible text is above 7th grade while others are
not, resulting in a score of 1."\n        }\n    ]\n}'
```

[199]:

```
out = js.loads(eval_result)
out
```

[199]:

```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Plain-language & readability thresholds met (senior-
appropriate)',
   'Assigned_Score': 1,
   'Max_Score': 2,
   'Reasoning': "The visible text includes 'Analyze Image', which contains the
word 'Analyze' (Flesch-Kincaid Grade Level ~7.5), placing it above a 7th grade
reading level. All other visible text elements (e.g., 'Start new', 'How can I
help you?', 'Create an image', 'Help me write', etc.) are at or below the 7th
grade level. Thus, some visible text is above 7th grade while others are not,
resulting in a score of 1."}]}
```

[200]:

```
l4_results.append(out)
```

Aggregate scores and reasoning for the different subsections evaluated for this L4 indicator

[201]:

```
combined_results = {}
combined_results["Website"] = ""
combined_results["L4_Indicator"] = ""
combined_results["Assigned_Score"] = 0
combined_results["Max_Score"] = 0
combined_results["Reasoning"] = ""

for json_obj in l4_results:
    for result in json_obj["results"]:
        for field in result:
            if field == "Assigned_Score":
                combined_results[field] += result[field]
            elif field == "Reasoning":
                combined_results[field] += " "
                combined_results[field] += result[field]
            elif field == "Max_Score":
                combined_results[field] += result[field]
            else:
                combined_results[field] = result[field]

print(combined_results)
```

```
{'Website': 'ClaudeAI', 'L4_Indicator': 'Plain-language & readability thresholds
met (senior-appropriate)', 'Assigned_Score': 8, 'Max_Score': 14, 'Reasoning': "
Several input targets such as 'Help me write', 'Learn about', 'Analyze Image',
'Summarize text', and '+ See More' buttons appear to be smaller than the 44x44
CSS pixel recommendation for touch targets. While elements like 'Start new' and
voice input icons may meet or exceed the size requirement, the presence of
multiple smaller buttons indicates 'some input targets are below the WCAG
recommended size'. The page includes a microphone icon (image) for voice input,
but no accompanying text label to explicitly indicate its function. While the
microphone icon is a standard visual cue for voice input, the absence of
descriptive text means it is not clearly identifiable by both image and text.
Line spacing is not applicable to any visible text items, as all text elements
(e.g., headings, buttons, input placeholders) are single-line. The criteria
requires evaluating line spacing (vertical space between multiple lines of
text), which does not exist here. Thus, the criteria is not applicable, and the
maximum score is assigned. Some visible text items (e.g., spacing between 'No
Chat History' and 'Gregory Knapp' in the sidebar) have minimal vertical spacing
that appears to be below 2 times the font size. This indicates that not all text
spacing meets the threshold of at or above 2 times the font size. All visible
text elements (e.g., 'How can I help you?', 'Type a message…', menu options)
exhibit standard, non-restricted letter spacing. The UI follows typical web
design conventions where character spacing exceeds 0.12× font size, ensuring
readability without visible character crowding or overlapping. Based on the
visible text elements (e.g., 'How can I help you?', 'Type a message…', button
labels), the spacing between words is below 0.16 times the font size for all
visible text. For example, with a typical font size of 14px for body text, 0.16
* 14 = 2.24px; standard word spacing of ~2px is below this threshold. Similarly,
larger headings (e.g., 24px font size) have word spacing below 0.16x. Thus, all
visible text words meet the condition for a score of 0. The visible text
includes 'Analyze Image', which contains the word 'Analyze' (Flesch-Kincaid
Grade Level ~7.5), placing it above a 7th grade reading level. All other visible
text elements (e.g., 'Start new', 'How can I help you?', 'Create an image',
'Help me write', etc.) are at or below the 7th grade level. Thus, some visible
text is above 7th grade while others are not, resulting in a score of 1."}
```

[202]:

```python
new_df = pd.DataFrame(combined_results, index=[0])
new_df
```

[202]:

```
    Website                                      L4_Indicator  \
0  ClaudeAI  Plain-language & readability thresholds met (s…

   Assigned_Score  Max_Score  \
0               8         14

                                     Reasoning
0   Several input targets such as 'Help me write'…
```

[203]:

```python
claudeai_test_results_df = pd.concat([claudeai_test_results_df,new_df])
claudeai_test_results_df
```

[203]:

```
    Website                                      L4_Indicator Assigned_Score  \
0  ClaudeAI          Multilingual support for intended locales               2
0  ClaudeAI  Senior-focused simplification & stepwise guida…              2
0  ClaudeAI  Senior motor and voice accessibility options (…              2
0  ClaudeAI  Plain-language & readability thresholds met (s…              8

  Max_Score                                        Reasoning
0         4  The page lists 11 supported languages, which i…
0         2  The 'Help me write' button is clearly visible …
0         4   Several input targets such as 'Help me write'…
0        14   Several input targets such as 'Help me write'…
```

4.3.4 L4: Cognitive-load reduction patterns present (chunking, progressive disclosure)

[204]:

```python
# Prompt for this L4 Indicator
prompt_in = """
This image shows the homepage for ClaudeAI's chat model interaction interface.␣
  ↪You are tasked with evaluating the accessibility of this page in terms of␣
  ↪cognitive load reduction through proper header and label text.
```

```
Based only on what you see in the image of the homepage, rate the accessibility␣
 ↪of help functionality on the following scale ranging from 0 to 2:

0: All visible text headers and labels do not clearly explain the purpose of␣
 ↪their associated interactive elements.
1: Some visible text headers and labels do not clearly explain the purpose of␣
 ↪their associated interactive elements.
2: All visible text headers and labels clearly explain the purpose of their␣
 ↪associated interactive elements.

The response should be in valid JSON format. The following example below shows␣
 ↪properly formatted output. The structure, Website, L4_Indicator and␣
 ↪Max_Score properties should not be changed. You are only writing the value␣
 ↪for Score and Reasoning in the response.

{
    "results": [
        {
        "Website": "ClaudeAI",
        "L4_Indicator": "Cognitive-load reduction patterns present (chunking,␣
 ↪progressive disclosure)",
        "Assigned_Score": ,
        "Max_Score": 2,
        "Reasoning": ""
        }
    ]
}
"""

# Interact with the vision model
response = ollama.generate(
    model="qwen3-vl:8b",  # Use the name of the vision model you pulled
    prompt=prompt_in,
    images=[claude_bytes]
)

# Save the model's response as json
eval_result = response['response']
```

[205]:

```
print(eval_result)
```

```
{
    "results": [
        {
            "Website": "ClaudeAI",
            "L4_Indicator": "Cognitive-load reduction patterns present
(chunking, progressive disclosure)",
            "Assigned_Score": 2,
            "Max_Score": 2,
            "Reasoning": "All visible text headers and labels clearly explain
the purpose of their associated interactive elements. The main header 'How can I
help you?' provides a clear context. The input field label 'Type a message…'
explicitly describes the expected action. Interactive elements like 'Create an
image', 'Help me write', 'Analyze image', and 'Summarize text' each have labels
that directly state their function, eliminating ambiguity about their purpose."
        }
    ]
}
```

[206]:
```
out = js.loads(eval_result)
out
```

[206]:
```
{'results': [{'Website': 'ClaudeAI',
   'L4_Indicator': 'Cognitive-load reduction patterns present (chunking,
progressive disclosure)',
   'Assigned_Score': 2,
   'Max_Score': 2,
   'Reasoning': "All visible text headers and labels clearly explain the purpose
of their associated interactive elements. The main header 'How can I help you?'
provides a clear context. The input field label 'Type a message…' explicitly
describes the expected action. Interactive elements like 'Create an image',
'Help me write', 'Analyze image', and 'Summarize text' each have labels that
directly state their function, eliminating ambiguity about their purpose."}]}
```

[208]:
```
new_df = pd.DataFrame(out['results'])
new_df
```

[208]:
```
    Website                                       L4_Indicator  \
0  ClaudeAI  Cognitive-load reduction patterns present (chu…

   Assigned_Score  Max_Score  \
0               2          2

                                           Reasoning
0  All visible text headers and labels clearly ex…
```

[209]:
```
claudeai_test_results_df = pd.concat([claudeai_test_results_df,new_df])
```

[210]:
```
claudeai_test_results_df
```

[210]:
```
    Website                                       L4_Indicator Assigned_Score  \
0  ClaudeAI          Multilingual support for intended locales              2
0  ClaudeAI  Senior-focused simplification & stepwise guida…              2
0  ClaudeAI  Senior motor and voice accessibility options (…              2
0  ClaudeAI  Plain-language & readability thresholds met (s…              8
0  ClaudeAI  Cognitive-load reduction patterns present (chu…              2

  Max_Score                                          Reasoning
0         4  The page lists 11 supported languages, which i…
0         2  The 'Help me write' button is clearly visible …
0         4   Several input targets such as 'Help me write'…
0        14   Several input targets such as 'Help me write'…
0         2  All visible text headers and labels clearly ex…
```

4.4 Finalize ClaudeAI Test Results

[211]:

```
claudeai_test_results_df.reset_index(inplace=True)
claudeai_test_results_df
```

[211]:

```
   index   Website                                       L4_Indicator  \
0      0  ClaudeAI          Multilingual support for intended locales
1      0  ClaudeAI  Senior-focused simplification & stepwise guida…
2      0  ClaudeAI  Senior motor and voice accessibility options (…
3      0  ClaudeAI  Plain-language & readability thresholds met (s…
4      0  ClaudeAI  Cognitive-load reduction patterns present (chu…

  Assigned_Score Max_Score                                          Reasoning
0              2         4  The page lists 11 supported languages, which i…
1              2         2  The 'Help me write' button is clearly visible …
2              2         4   Several input targets such as 'Help me write'…
3              8        14   Several input targets such as 'Help me write'…
4              2         2  All visible text headers and labels clearly ex…
```

[212]:

```
claudeai_test_results_df.drop(columns=['index'], inplace=True)
claudeai_test_results_df
```

[212]:

```
    Website                                       L4_Indicator Assigned_Score  \
0  ClaudeAI          Multilingual support for intended locales              2
1  ClaudeAI  Senior-focused simplification & stepwise guida…              2
2  ClaudeAI  Senior motor and voice accessibility options (…              2
3  ClaudeAI  Plain-language & readability thresholds met (s…              8
4  ClaudeAI  Cognitive-load reduction patterns present (chu…              2

  Max_Score                                          Reasoning
0         4  The page lists 11 supported languages, which i…
1         2  The 'Help me write' button is clearly visible …
2         4   Several input targets such as 'Help me write'…
3        14   Several input targets such as 'Help me write'…
4         2  All visible text headers and labels clearly ex…
```

4.4.1 Duplicate L4 Motor and voice accessibility options (voice input, large targets, reduced precision) L4

Since the L4 for Senior motor and voice accessibility options (voice input, large targets, reduced precision) was essentially identical to the L4 in the other L3 Subcategory, the score for the senior focused L4 is duplicated to reflect its presence in two separate L4s on the tree

```
[213]: # Replicate each row 3 times
       seniors_row = claudeai_test_results_df.iloc[[2]]
       seniors_row['L4_Indicator'] = "Motor and voice accessibility options (voice␣
        ↪input, large targets, reduced precision)"
       seniors_row
```

```
/tmp/ipython-input-661153912.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  seniors_row['L4_Indicator'] = "Motor and voice accessibility options (voice
input, large targets, reduced precision)"
```

```
[213]:     Website                                        L4_Indicator Assigned_Score  \
       2  ClaudeAI  Motor and voice accessibility options (voice i…               2

          Max_Score                                          Reasoning
       2          4   Several input targets such as 'Help me write'…
```

```
[214]: claudeai_test_results_df = pd.concat([claudeai_test_results_df, seniors_row]).
        ↪reset_index().drop(columns=['index'])
       claudeai_test_results_df
```

```
[214]:     Website                                        L4_Indicator Assigned_Score  \
       0  ClaudeAI          Multilingual support for intended locales              2
       1  ClaudeAI  Senior-focused simplification & stepwise guida…               2
       2  ClaudeAI  Senior motor and voice accessibility options (…               2
       3  ClaudeAI  Plain-language & readability thresholds met (s…               8
       4  ClaudeAI  Cognitive-load reduction patterns present (chu…               2
       5  ClaudeAI  Motor and voice accessibility options (voice i…               2

          Max_Score                                          Reasoning
       0          4  The page lists 11 supported languages, which i…
       1          2  The 'Help me write' button is clearly visible …
       2          4   Several input targets such as 'Help me write'…
       3         14   Several input targets such as 'Help me write'…
       4          2  All visible text headers and labels clearly ex…
       5          4   Several input targets such as 'Help me write'…
```

[223]:

```
claudeai_test_results_df.iloc[3, 3] = 10 # For some reason the max score was␣
  ↪counted incorrectly, even though the assigned was correct
```

[224]:

```
claudeai_test_results_df
```

[224]:

```
    Website                                       L4_Indicator Assigned_Score  \
0  ClaudeAI          Multilingual support for intended locales              2
1  ClaudeAI  Senior-focused simplification & stepwise guida…              2
2  ClaudeAI  Senior motor and voice accessibility options (…              2
3  ClaudeAI  Plain-language & readability thresholds met (s…              8
4  ClaudeAI  Cognitive-load reduction patterns present (chu…              2
5  ClaudeAI  Motor and voice accessibility options (voice i…              2

  Max_Score                                          Reasoning
0         4  The page lists 11 supported languages, which i…
1         2  The 'Help me write' button is clearly visible …
2         4   Several input targets such as 'Help me write'…
3        10   Several input targets such as 'Help me write'…
4         2  All visible text headers and labels clearly ex…
5         4   Several input targets such as 'Help me write'…
```

4.4.2 Return to L4 WCAG-aligned accessibility features available L4

Now that we have completed the other L4 indicators, we can aggregate them to get a score for the WCAG compliance metric

[225]:

```
wcag_alignment_assigned_score = claudeai_test_results_df['Assigned_Score'].sum()
wcag_alignment_max_score = claudeai_test_results_df['Max_Score'].sum()
```

[226]:

```
wcag_alignment_values = {
    "Website": "ClaudeAI",
    "L4_Indicator": "WCAG-aligned accessibility features available",
    "Assigned_Score": wcag_alignment_assigned_score,
    "Max_Score": wcag_alignment_max_score,
    "Reasoning": "This value is the summation of all the other L4 indicators␣
  ↪handled in this test, since they are meant to be aligned with WCAG guidance."
}
```

[227]:

```
new_df = pd.DataFrame(wcag_alignment_values, index=[6])
new_df
```

[227]:

```
    Website                                   L4_Indicator  Assigned_Score  \
6  ClaudeAI  WCAG-aligned accessibility features available              18

   Max_Score                                          Reasoning
6         26  This value is the summation of all the other L…
```

[228]:

```
final_claudeai_gpt_scores_df = pd.concat([claudeai_test_results_df, new_df])
```

```
[229]: final_claudeai_gpt_scores_df
```

```
[229]:     Website                                       L4_Indicator Assigned_Score  \
       0  ClaudeAI          Multilingual support for intended locales              2
       1  ClaudeAI  Senior-focused simplification & stepwise guida…              2
       2  ClaudeAI  Senior motor and voice accessibility options (…              2
       3  ClaudeAI  Plain-language & readability thresholds met (s…              8
       4  ClaudeAI  Cognitive-load reduction patterns present (chu…              2
       5  ClaudeAI  Motor and voice accessibility options (voice i…              2
       6  ClaudeAI      WCAG-aligned accessibility features available             18

         Max_Score                                          Reasoning
       0         4  The page lists 11 supported languages, which i…
       1         2  The 'Help me write' button is clearly visible …
       2         4   Several input targets such as 'Help me write'…
       3        10   Several input targets such as 'Help me write'…
       4         2  All visible text headers and labels clearly ex…
       5         4   Several input targets such as 'Help me write'…
       6        26  This value is the summation of all the other L…
```

5 View Final Results of both tests

```
[230]: final_chat_gpt_scores_df
```

```
[230]:    Website                                       L4_Indicator Assigned_Score  \
       0  ChatGPT          Multilingual support for intended locales              4
       1  ChatGPT  Senior-focused simplification & stepwise guida…              0
       2  ChatGPT  Senior motor and voice accessibility options (…              2
       3  ChatGPT  Plain-language & readability thresholds met (s…              8
       4  ChatGPT  Cognitive-load reduction patterns present (chu…              2
       5  ChatGPT  Motor and voice accessibility options (voice i…              2
       6  ChatGPT      WCAG-aligned accessibility features available             18

         Max_Score                                          Reasoning
       0         4  The page lists 39 supported languages, which m…
       1         2  No help icons or links are present on the home…
       2         4   The microphone icon for voice input and the '…
       3        10   All visible text items with multiple lines (e…
       4         2  All visible text headers and labels (e.g., 'Ne…
       5         4   The microphone icon for voice input and the '…
       6        26  This value is the summation of all the other L…
```

```
[231]: final_claudeai_gpt_scores_df
```

```
[231]:     Website                                       L4_Indicator Assigned_Score  \
       0  ClaudeAI          Multilingual support for intended locales              2
       1  ClaudeAI  Senior-focused simplification & stepwise guida…              2
```

```
2  ClaudeAI  Senior motor and voice accessibility options (…             2
3  ClaudeAI  Plain-language & readability thresholds met (s…             8
4  ClaudeAI  Cognitive-load reduction patterns present (chu…             2
5  ClaudeAI  Motor and voice accessibility options (voice i…             2
6  ClaudeAI      WCAG-aligned accessibility features available            18

  Max_Score                                        Reasoning
0         4  The page lists 11 supported languages, which i…
1         2  The 'Help me write' button is clearly visible …
2         4   Several input targets such as 'Help me write'…
3        10   Several input targets such as 'Help me write'…
4         2  All visible text headers and labels clearly ex…
5         4   Several input targets such as 'Help me write'…
6        26  This value is the summation of all the other L…
```

Given that this is a somewhat simple metric (with plenty of room for expansion through the implementation of additional WCAG guidelines, either in here or in other L3 Subcategories throughout the AI Ethics Index, it is not surprising that the results were quite similar.

Both landing pages take inspiration for each other and focus on simplicity, since it is a more common modern design philosophy especially in tech companies, and to keep the focus on the chat models themselves.

Interestingly, although the overall L4 Metric, WCAG alignment resulted in an equal score, indicating that both models had the same aggregate score across all the other L4s, there was one difference between the two.

ChatGPT received a 4/4 for the language support, since it supports over 40 different langauges, at least as of 12/9/2025, while the language support for Claude is quite low in comparison. On the other hand, ClaudeScored better for help functionality and guidance given the presence of a help me write function while ChatGPT provides no visible guidance for a user beyond what is presented on the screen.

This result shows that at least from a reasoning perspective, Qwen is able to view the images, find discernable differences and grade accordingly based on the instructions given.