AccessCheck

Purva Santosh Dusane
HCI 5840 – Fall 2025

General description of the project (2 pts)

(Revised)

AccessCheck is a beginner-friendly accessibility analyzer that checks public websites against basic WCAG 2.2 Level A (Tier 1) guidelines and provides results in a clear, non-technical report. The app is aimed at students, beginner web designers, educators, and small-business owners who want to understand and improve accessibility without needing deep technical skills.

The current version (Version 1B) operates as a command-line interface (CLI) where the user enters a website URL. The program analyzes the HTML and outputs a simple, human-readable report with *Pass / Warning / Fail* indicators. The report also saves automatically as a `.txt` file inside the `docs/` folder.

New in Version 1B (Revised):

- Modular design with **separate functions for each accessibility check.**
- Expanded checks beyond the original 3:
  - Page Title (`<title>`)
  - HTML Language Attribute (`<html lang="">`)
  - Image Alt Attributes
  - Heading Hierarchy and Order
  - Link Text Clarity (“click here” / “read more”)
  - Form Label Presence
- Added safe web requests using `User-Agent` headers.
- Added automatic report saving to a `docs/` folder.

Future Version (planned Version 2):
The next version will include a **Flask web dashboard** with a text box to enter URLs, visual cards showing results, and downloadable reports in PDF or CSV format.

Task Vignettes (User activity “flow”) (4 pts)

1. Quick Site Check

Maya, a design student, wants to review her portfolio website.
She opens the terminal and types:

```
python main_v1b.py
```

When prompted, she enters:

```
https://mayaportfolio.com
```

Within seconds, AccessCheck prints results such as:

```
[PASS] Page title found: Maya Portfolio
[PASS] HTML language set to “en”
[FAIL] 2 images are missing alt attributes
[WARN] Heading order skips from h1 → h3
[PASS] Links use meaningful text
[PASS] All form inputs have labels
```

A report file (`accessibility_report_v1b.txt`) is automatically saved inside the `docs/` folder.

Technical notes:

- Input: URL entered in the terminal.
- Processing: Program fetches HTML, parses with BeautifulSoup, runs six checks.
- Output: Readable summary printed + saved to a text file.

2. Re-Testing After Fixes

After updating her website with better alt texts, Maya re-runs the same command.
The new report shows:

```
[PASS] All images have alt text
[PASS] All other checks passed
```

Technical notes:

- The user can re-run the program as often as needed.
- Each report is saved separately with timestamps to track progress.

3. Web Dashboard

(Updated description)

In the next version, Maya can access a Flask-based dashboard instead of using the terminal.
She pastes her website URL into a box and clicks Analyze.

Results appear as **colored cards**:

- 🟢 Pass
- 🟡 Needs Attention
- 🔴 Fail

Each card includes a short explanation and, if available, the HTML line number of the issue.
Users can download results as PDF or CSV for record-keeping.

Technical "flow" (3 pts)

Overall Data Flow

User Input: Website URL typed in the command line.
Page Retrieval: AccessCheck uses `requests` to fetch the HTML code.
Parsing: The HTML is processed using `BeautifulSoup`.

Checks Performed:

- Verify `<title>` tag presence.
- Check if `<html>` has a language attribute.
- Scan for images missing `alt` attributes.
- Review heading levels for structure.
- Identify links with non-descriptive text.
- Ensure form inputs have corresponding labels.

Result Collection: Each check returns a status (PASS/WARN/FAIL).
Output: Results printed to the terminal and saved as a text report in the `docs/` folder.

Main Modules and Blocks

- Input Block: Collects URL from user.
- Fetch Block: Downloads page content.
- Parser Block: Uses BeautifulSoup to read HTML tags.
- Check Blocks: Runs the six accessibility tests.
- Report Block: Combines results and saves to a file.

Interaction Points

For now:

- The user interacts only through the terminal.

Later:

- The user will paste a URL into a web form and see results in a visual dashboard.