**Project Progress: Website Archaeology Tool**

**Introduction**

The Website Archaeology Tool aims to enhance web development education for novices by visualizing static websites’ HTML, CSS, and JavaScript as a 3D “excavation” metaphor, promoting STEM engagement. By parsing MIT-licensed GitHub repositories (HTML5 Boilerplate, Minimal Mistakes, Clean Blog) and presenting their structure in a 3D view with a 2D timeline, the tool makes abstract concepts like tag hierarchies tangible. This document reviews web development education, visualization tools, similar tools, datasets, methodology, expected results, and progress, addressing research gaps and questions.

**Progress Update**

As of July 2, 2025, I’ve achieved 500 XP in Codecademy’s Python/Computer Science paths, mastering loops, functions, and dictionaries. This enabled development of parse\_html.py and parse\_css.py to extract HTML tags (e.g., {'div': 2}) and CSS rules from datasets like HTML5 Boilerplate. Fixed a FileNotFoundError in parse\_css.py by updating the path to dist/css/style.css. All code is pushed to 23098729 files on onedrive

**Web Development Education**

Web development skills (HTML, CSS, JavaScript) are critical for digital literacy and STEM education. W3Schools offers beginner-friendly tutorials but struggles to convey complex relationships like CSS inheritance (W3Schools HTML Tutorial, 2025). CodePen provides live coding for real-time previews, yet its advanced focus limits novice accessibility (CodePen About, 2025). Neither tool visualizes code structures in 3D, a gap my tool addresses.

**Visualization Tools**

Visualization tools enhance coding education. BeautifulSoup parses HTML efficiently, ideal for analyzing website structures, but lacks visual output for beginners (BeautifulSoup Documentation, 2025). Three.js supports 3D web visualizations, used in simulations, but requires coding expertise (Three.js Documentation, 2025). Scratch engages young learners with block-based programming, yet misses web-specific concepts (Resnick et al., 2009). My tool combines BeautifulSoup’s parsing with Three.js’s 3D visualization for educational impact.

**Similar Tools**

* **Wayback Machine (Internet Archive)**:
  + Description: Archives website snapshots over time, showing historical versions (e.g., github.com in 2015).
  + Features: Saves HTML/CSS/images, displays a timeline of snapshots, renders past versions.
  + Use Case: Researchers study website evolution.
  + Access: Free, web-based.
  + Relevance: Tracks website history like my tool but focuses on 2D snapshots, not 3D visualization or code parsing.
* **Page Inspect (Ahrefs)**:
  + Description: Analyzes HTML/text changes for SEO impact.
  + Features: Compares HTML changes, shows side-by-side diffs.
  + Use Case: SEO professionals monitor content updates.
  + Access: Paid (Ahrefs subscription).
  + Relevance: Similar to my parsing/timeline but SEO-focused, lacking 3D visualization or educational use.
* **PowerMapper**:
  + Description: Generates visual sitemaps for website structure analysis, used by MIT/NASA.
  + Features: Scans pages, creates 2D sitemaps (tree, circle).
  + Use Case: Developers analyze site structure.
  + Access: Paid, free trial.
  + Relevance: Parses structure like my tool but uses 2D sitemaps, not 3D or commit-based history.

**What Makes My Tool Different**

* **3D “Excavation” View**: Visualizes HTML tags (e.g., <div> in blue) as 3D layers using Three.js, unlike Wayback Machine’s 2D renders or PowerMapper’s sitemaps. Example: Parsing HTML5 Boilerplate’s index.html to show nested tags as 3D blocks.
* **Commit-Based Tracking**: Analyzes Git commits (e.g., HTML5 Boilerplate v8.0.0–v8.0.2), unlike Wayback Machine’s irregular snapshots or Page Inspect’s SEO focus. Example: Shows <meta> tag addition in Minimal Mistakes.
* **Educational Focus**: Targets STEM learners, unlike archiving (Wayback Machine) or SEO (Page Inspect) tools. Example: Students learn responsive design via Clean Blog’s CSS changes.
* **Open-Source Datasets**: Uses MIT-licensed repositories, ensuring accessibility, unlike paid tools (Page Inspect, PowerMapper).

**Comparison Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **My Tool** | **Wayback Machine** | **Page Inspect** | **PowerMapper** |
| Parsing | HTML/CSS/JS (BeautifulSoup, cssutils) | Full-page snapshots | HTML/text for SEO | Page structure |
| Visualization | 3D view (Three.js), 2D timeline | 2D timeline, page renders | Side-by-side diffs | 2D sitemaps |
| Data Source | MIT-licensed GitHub repositories | Internet-wide websites | Live websites | Live/archived sites |
| Purpose | STEM education | Historical archiving | SEO analysis | Site mapping |
| Version Tracking | Commit-based (e.g., v8.0.0–v8.0.2) | Snapshot dates | SEO change logs | None |
| Access | Open-source, local/web-based | Free, web-based | Paid | Paid, free trial |

**Gaps in Existing Tools**

* **Lack of 3D Visualization**: Existing tools use 2D views (timelines, sitemaps). My 3D view makes website structure intuitive.
* **Limited Educational Focus**: Wayback Machine, Page Inspect, and PowerMapper target archivists or professionals, not students. My tool teaches web development.
* **Generic Version Tracking**: Wayback Machine’s snapshots are irregular; Page Inspect focuses on SEO. My tool uses precise Git commits.
* **Accessibility**: Paid tools (Page Inspect, PowerMapper) limit student access. My tool uses free, open-source datasets.

**How My Tool Helps**

* **Educational Value**: Visualizes code as 3D “artifacts,” teaching students (e.g., Clean Blog’s CSS for media queries).
* **Research Support**: Enables studying website evolution (e.g., HTML5 Boilerplate’s accessibility tags).
* **Accessibility**: Open-source datasets/tools ensure reproducibility.
* **Engagement**: Interactive 3D view/timeline engages STEM learners.

**Research Questions**

1. How does 3D visualization enhance novices’ understanding of web development structures?
2. Can an archaeological metaphor increase STEM engagement among learners?
3. Is static website parsing sufficient for creating effective 3D educational visualizations?

**Datasets**

Three MIT-licensed GitHub repositories are used:

* **HTML5 Boilerplate** (https://github.com/h5bp/html5-boilerplate, ~3,000 commits): Template with index.html, dist/css/style.css, js/main.js. Commits v8.0.0–v8.0.2 show HTML/CSS changes. Clone: git clone https://github.com/h5bp/html5-boilerplate.git.
* **Minimal Mistakes** (https://github.com/mmistakes/minimal-mistakes, ~2,500 commits): Jekyll theme with static \_site/index.html, assets/css/main.css. Commits v4.24.0–v4.26.0 provide history. Requires Jekyll build (jekyll build).
* **Clean Blog** (https://github.com/StartBootstrap/startbootstrap-clean-blog, ~100 commits): Bootstrap theme with index.html, css/clean-blog.min.css. Commits v5.0.0–v5.0.2 show updates.

**Suitability**: Static websites with commit history for parsing/visualization.

**Data Use**: BeautifulSoup/cssutils parse files; Three.js creates 3D views; JavaScript/CSS timeline shows changes.

**Validation**: Compare parsed tag/CSS counts to manual counts (e.g., <div> tags in index.html via text editor) across commits.

**Methodology**

Using design-based research, the tool parses three static websites with BeautifulSoup (HTML tags), cssutils (CSS rules), and JavaScript parsing, visualized via Three.js (3D “excavation” view, e.g., <div> as blue blocks) and a 2D timeline (JavaScript/CSS), served by Flask. Validation compares parsed outputs to manual analysis (e.g., tag counts). No human subjects are involved, using public datasets. This addresses RQ1 (visualization impact), RQ2 (metaphor engagement), and RQ3 (parsing sufficiency).

**Expected Results**

* 3D visualization (Three.js) of website structures (e.g., <div> as blue blocks with tooltips).
* 2D timeline (JavaScript/CSS) of commit changes (e.g., <meta> tag added).
* Parsed data (e.g., {'html': 1, 'div': 2}, CSS rule counts) validated against manual analysis.
* Impact: Engages STEM students, supports research on website evolution.

**References**

* BeautifulSoup (2025). BeautifulSoup Documentation. https://www.crummy.com/software/BeautifulSoup/ (Accessed: 24 June 2025).
* CodePen (2025). About CodePen. https://codepen.io/about/ (Accessed: 24 June 2025).
* HTML5 Boilerplate (2025). GitHub Repository. https://github.com/h5bp/html5-boilerplate (Accessed: 2 July 2025).
* Minimal Mistakes (2025). GitHub Repository. https://github.com/mmistakes/minimal-mistakes (Accessed: 2 July 2025).
* Clean Blog (2025). GitHub Repository. https://github.com/StartBootstrap/startbootstrap-clean-blog (Accessed: 2 July 2025).
* Resnick, M., et al. (2009). ‘Scratch: Programming for All’, Communications of the ACM, 52(11), pp. 60–67. doi:10.1145/1592761.1592779.
* Three.js (2025). Three.js Documentation. https://threejs.org/docs/ (Accessed: 24 June 2025).
* W3Schools (2025). HTML Tutorial. https://www.w3schools.com/html/ (Accessed: 24 June 2025).