International Journal of Scientific Research and Modern Education (IJSRME)
Impact Factor: 7.137, ISSN (Online): 2455 - 5630
(www.rdmodernresearch.com) Volume 7, Issue 1, 2022

WEB TECHNOLOGIES AND AUTOMATICALLY DETECTED ACCESSIBILITY ERRORS ACROSS HOME PAGES IN THE WEBAIM 2021 SURVEY

Divya Vishwanath* & Paul Johnson Asamoah**
* Janardhan Rai Nagar Rajasthan Vidyapeeth, Udaipur, Rajasthan, India
** Power Life University College, Accra, Ghana, West Africa

Cite This Article: Divya Vishwanath & Paul Johnson Asamoah, "Web Technologies and Automatically Detected Accessibility Errors Across Home Pages in the WebAIM 2021 Survey", International Journal of Scientific Research and Modern Education, Volume 7, Issue 1, Page Number 145-154, 2022.
Copy Right: © IJSRME, 2022 (All Rights Reserved). This is an Open Access Article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract:

This analysis examines variation in automatically detected accessibility errors across web technologies to inform international digital accessibility assessment. It consolidates published summaries for 25 technology groups from the February 2021 WebAIM Million audit, comprising eight JavaScript frameworks, 12 JavaScript libraries and five web frameworks. Descriptive comparisons examine reported mean errors per home page within each category. Framework means ranged from 50.4 to 67.3 errors per page, library means from 57.4 to 79.0, and web framework means from 40.7 to 66.6. The widest difference occurred among web frameworks, reaching 25.9 errors per page. These patterns demonstrate variation among evaluated technology groups without identifying the contribution of individual technologies, because page memberships can overlap. The analysis provides a consolidated descriptive comparison and clarifies why technology labels cannot substitute for evaluating rendered interfaces. Its contribution is to distinguish reported group differences from technology specific causal effects. For organisations serving international audiences, the findings support targeted accessibility audits that combine automated detection with examination of page content and interface behavior.

Key Words: Accessibility Assessment; Automated Evaluation; Descriptive Analysis; Digital Inclusion; Web Technologies.

1. Introduction:

Web accessibility determines whether people with disabilities can perceive, understand and interact with digital information and services. Barriers in page content and interface controls can restrict access to education, employment, commerce and public services. In its February 2021 evaluation of one million highly ranked website home pages, WebAIM detected an average of 51.4 accessibility errors per page and found at least one detectable Web Content Accessibility Guidelines failure on 97.4% of pages. These figures describe automatically detectable failures in the evaluated sample rather than all barriers experienced by users. They nevertheless demonstrate the scale of the accessibility problem among widely visited websites.

Web development technologies are relevant to this problem because frameworks and libraries contribute to the structures and interactive elements that browsers present to users. Their influence depends on how developers combine those resources with content, styles and accessibility features. A technology may provide components that support accessible design, but its presence on a page does not ensure that the resulting interface is accessible. Conversely, an error detected on a page that uses a technology cannot automatically be attributed to that technology. The outcome reflects the rendered page and the interaction of its components.

The distinction between source code and rendered content is particularly important for interactive interfaces. Fernandes et al. (2012) examined accessibility evaluation for rich internet applications and showed the need to assess content introduced through browser processing and user events. Their work highlights a measurement problem: an evaluation that observes only one representation or state of an interface may miss barriers present in another. The present analysis uses WebAIM results for home pages after scripts and styles were applied. It does not assess every possible interactive state or change in accessibility across software versions.

Evidence also indicates that detected accessibility differs across groups of pages associated with development technologies. Duarte et al. (2016) evaluated 1,669 pages from different domains and examined technologies including programming languages, web frameworks, JavaScript frameworks and content management systems. Their analysis reported differences in automatically assessed accessibility among technology groups. This finding supports technology specific investigation, while leaving open how much of an observed difference reflects the technology, its implementation or other characteristics of the pages where it is used.

Measurement further conditions the interpretation of such comparisons. Vigo and Brajnik (2011) showed that automated accessibility metrics must be evaluated against properties such as validity, reliability and sensitivity. Frazão and Duarte (2020) found differences in the coverage of accessibility criteria across automated evaluation tools. A detected error count is therefore informative about the barriers identified under a particular

method, but it is not a complete measure of user experience or accessibility conformance. Comparing technology groups requires a common interpretation of that outcome and attention to the possibility that the same page appears in more than one group.

The closest evidence for this analysis is the WebAIM Million report for February 2021, which already presents mean detected errors for individual technologies. The report establishes that the listed technology groups have different averages and cautions against attributing those differences directly to the technologies. The present analysis does not introduce a new source of page observations or claim that these patterns were previously undocumented. Its contribution is a focused synthesis of the published results across three development technology categories, using a consistent outcome and identifying the limits of comparisons among overlapping groups.

This study aims to examine how the reported mean number of automatically detected accessibility errors varies across home pages associated with JavaScript frameworks, JavaScript libraries and web frameworks. The overarching research question is: What mean numbers of detected accessibility errors are reported for home pages associated with individual technologies in these three categories? The specific objectives are to describe the published framework means, examine the corresponding library means and compare the reported means for web frameworks.

The analysis uses 25 technology group observations drawn from the February 2021 WebAIM report: eight JavaScript frameworks, 12 JavaScript libraries and five web frameworks. Each observation contains the published number of associated home pages and the mean number of WAVE detected errors per page. The cross-sectional design permits descriptive comparisons within the selected categories. It does not identify causal effects, test changes across interface versions or measure the separate contribution of each technology to a page. The findings can inform accessibility auditing by identifying technology groups whose associated pages had different reported error averages and may warrant closer examination. Their transferability lies in the method of comparing reported groups under a common evaluation, while interpretation remains specific to the selected home pages, technologies and detection tool. Further page level and user-based evidence would be needed to explain the observed differences and assess barriers beyond those detected automatically.

2. Theory, Conceptual Framework and Research Questions:

2.1 Theoretical Foundation:

This analysis applies an accessibility engineering perspective to the relationship between web technologies and automatically detected accessibility errors. The perspective treats accessibility as an outcome of interactions among page structure, content, developer choices, browser rendering and assistive technologies. The World Wide Web Consortium identifies perceivability, operability, understandability and robustness as principles for accessible web content. Its accessibility specification also explains how semantic information helps assistive technologies communicate interface structure and behavior to users, as stated by the World Wide Web Consortium in 2017 and 2018.

The theoretical expectation is that development technologies can shape the opportunities and constraints involved in producing accessible pages. Frameworks and libraries provide reusable structures and components, but developers determine how those resources are applied. Accessible features may therefore reduce barriers when implemented correctly, while inaccessible content or implementation choices may introduce errors. This mechanism does not imply that a technology causes every error detected on a page. In an evaluation of 1,669 pages, Duarte and colleagues in 2016 reported differences in detected accessibility across technologies, providing empirical support for examining technology groups separately.

The dependent variable is the mean number of accessibility errors detected per home page within each reported technology group. WebAIM evaluated rendered pages after scripts and styles were applied. The resulting measure reflects errors detectable by its evaluation tool, rather than every barrier experienced by users. Campoverde Molina and colleagues in 2020 also documented the use of automated methods in accessibility evaluation. The distinction matters because an automated error count does not establish complete accessibility or conformance.

2.2 JavaScript Frameworks and Detected Accessibility Errors:

JavaScript frameworks provide structures for building interactive web interfaces. These structures can influence how page elements and user interface behavior are produced. Their accessibility implications depend on the features provided by the framework and how developers use them.

The available data identify eight JavaScript frameworks and report the mean detected errors for pages associated with each one. Differences among these means can describe variation across framework groups. They cannot establish whether a framework caused the errors or whether differences arise from other characteristics of the pages. Duarte and colleagues in 2016 support examining technology differences while recognizing that accessibility is assessed on the pages delivered to users.

Research question 1: What mean numbers of automatically detected accessibility errors are reported for home pages associated with each JavaScript framework?

2.3 JavaScript Libraries and Detected Accessibility Errors:

JavaScript libraries provide reusable functions or interface features that can be incorporated into pages alongside other technologies. Their contribution to a rendered interface depends on how those functions are combined with page content and other components. Consequently, pages associated with different libraries may have different detected error averages.

The dataset reports 12 JavaScript library groups. The analysis can compare their published means and identify variation within this category. Since a page can use more than one technology, the reported groups may overlap. An observed difference between library groups therefore remains descriptive and cannot be attributed to a library alone. The WebAIM report in 2021 provides the group counts and error averages used for this comparison.

Research question 2: What mean numbers of automatically detected accessibility errors are reported for home pages associated with each JavaScript library?

2.4 Web Frameworks and Detected Accessibility Errors:

Web frameworks support the development and presentation of web pages through structures and components used by developers. Their relationship with accessibility depends on both the resources they provide and their implementation in individual pages. A framework that supports accessible construction may still be present on pages with detectable barriers.

The dataset contains five web framework groups and their reported mean error counts. These observations allow the analysis to describe differences among the groups. They do not identify the source of an error or isolate the contribution of a framework from other technologies on a page. The WebAIM report in 2021 explicitly cautions against attributing observed error differences to a technology solely because it is present.

Research question 3: What mean numbers of automatically detected accessibility errors are reported for home pages associated with each web framework?

3. Data:

This section establishes the provenance and construction of the analytical sample, defines the observed measures, and explains their use in the research questions. The analysis uses published technology group summaries from the WebAIM accessibility evaluation.

3.1 Data Sources, Population, and Sample Construction:

The dataset was compiled from the technology tables in the WebAIM Million report for February 2021. WebAIM evaluated the home pages of one million websites using the WAVE accessibility engine and additional tools to identify web technologies. The report assessed rendered pages after scripts and styles had been applied. Its website selection drew primarily on the Majestic Million list, supplemented by other ranking sources. Pages that returned errors or contained fewer than ten HTML elements were excluded by WebAIM.

The target population for this analysis comprises the technology groups reported in the three selected categories: JavaScript frameworks, JavaScript libraries and web frameworks. The accessible sampling frame consists of the corresponding published technology tables. WebAIM reported technologies detected on more than 5,000 home pages. The analytical sample retains all 25 technology groups listed in those three tables.

The unit of analysis is a technology group, represented by one reported mean error count and its associated number of home pages. The data form a cross section for February 2021. The report covers highly ranked websites without restricting the analysis to one country or sector. It does not provide a geographic breakdown for the technology groups. The source report is publicly accessible; the workbook does not record the extraction date or a specific data licence.

Table 1: Sample construction

Stage	Technology Groups	Group Observations	Rule
Accessible frame	25	25	Technologies reported in the three selected tables
Exclusions after extraction	0	0	All reported groups in the selected categories retained
Final analytical sample	25	25	Eight JavaScript frameworks, 12 JavaScript libraries and five web frameworks

The one million home pages in the source audit are not 25 independent samples. A home page can appear in more than one technology group. Therefore, group page counts cannot be summed to obtain a count of unique pages.

3.2 Data Integration, Cleaning, and Quality Assurance:

The workbook consolidates the three technology tables from the same WebAIM report into one dataset. Each record retains the audit year, technology category, technology name, number of home pages, mean detected errors and source URL. No external dataset was merged, and no temporal matching was required.

Inspection of the workbook identified no missing values across the six recorded fields and no exact duplicate rows. All 25 observations have a technology category, technology name, page count and mean error count. No imputation, interpolation, outlier removal or replacement of missing values with zero was applied.

The reported means were retained at the precision published by WebAIM. The workbook does not contain page level error counts or information on overlap between technology groups. Consequently, verification is limited to the available group records and does not extend to recalculating the published means from individual pages.

3.3 Variable Measurement, Construction, and Validation:

The explanatory measure identifies the technology category and the individual technology reported within that category. The outcome is the published mean number of WAVE detected accessibility errors per home page associated with each technology. The number of home pages provides the size of each reported technology group; it is not used as an independent count of unique pages across the sample.

WebAIM defines the reported errors as automatically detected barriers with a high likelihood of representing accessibility conformance failures. Automated evaluation does not capture every accessibility barrier. The World Wide Web Consortium published testable accessibility criteria in 2018, while Campoverde Molina and colleagues in 2020 documented the use of automated evaluation in empirical accessibility research. These sources support interpreting the measure as detected errors rather than complete accessibility.

Table 2: Variable dictionary

Variable	Analytical Role	Definition and Construction	Source	Unit and Frequency	Direction and Limitation
Technology category	Explanatory grouping	Published classification as JavaScript framework, JavaScript library or web framework	WebAIM technology tables	Category in February 2021	Categorical; groups may overlap
Technology name	Explanatory identifier	Name of the technology reported within its category	WebAIM technology tables	Technology group in February 2021	Identifies the group; does not measure technology use intensity
Mean detected errors	Outcome	Published average number of WAVE detected errors per home page associated with a technology	WebAIM technology tables	Errors per page in February 2021	Higher values indicate more detected errors; the measure does not capture every barrier
Home page count	Group size	Published number of evaluated home pages associated with a technology	WebAIM technology tables	Home pages in February 2021	Counts may overlap across technology groups

Table 3 reports descriptive statistics across the 25 technology group records. Each group receives equal weight in these summaries. They do not represent the average across unique home pages.

Table 3: Descriptive statistics

Measure	Valid Observations	Mean	SD	Median	Minimum	Maximum
Mean detected errors per page	25	62.24	8.09	61.60	40.70	79.00
Home pages per technology group	25	68,558.92	131,325.07	20,473	5,144	624,415

The descriptive statistics summarize variation among the published technology group measures. They do not measure variation among individual home pages.

3.4 Analytical Framework, Identification, and Reproducibility:

The analysis is descriptive. It examines the reported mean number of detected accessibility errors for each technology within the three categories. The three research questions are addressed by presenting and comparing the published group means within their respective categories.

The analysis does not estimate causal effects or test statistical differences between independent groups. Technology groups may overlap, and the workbook does not contain the page level observations or group variances needed to account for that overlap. Duarte and colleagues in 2016 examined development technologies alongside automated accessibility measures, illustrating the relevance of technology specific comparisons without establishing causation for the present dataset.

Interpretation remains conditional on WebAIM's page selection, technology detection and WAVE evaluation. The World Wide Web Consortium in 2017 explained that accessible interfaces depend on semantic information about structure and behavior. The available error counts summarize detected problems but do not identify the implementation decisions that produced them. Alsaeedi in 2020 also demonstrated that accessibility evaluation tools can differ in their detection performance.

The workbook preserves the reported group names, counts, means and source link. These fields allow the descriptive comparisons to be reproduced directly from the published tables. The extraction date and page level source records are not documented in the workbook.

4. Methods:

4.1 Research Design and Analytical Scope:

The analysis adopts a descriptive cross sectional design to examine differences in published accessibility error averages across web technology groups. The unit of analysis is a technology group reported in the February 2021 WebAIM Million evaluation. Each observation identifies a technology, its category, the number of associated home pages and the mean number of automatically detected errors per page. The analytical objective is to describe these reported averages within JavaScript frameworks, JavaScript libraries and web frameworks. The design addresses variation among observed groups without estimating the consequences of adopting or replacing a technology.

The comparison uses the published means as its primary evidence. Differences between technologies are interpreted as differences between the evaluated page groups associated with them. A home page may contain several technologies, and the workbook does not identify individual pages or their overlapping memberships. Consequently, the observed differences cannot separate the contribution of a technology from page content, implementation choices or other technologies. Duarte and colleagues in 2016 examined development technologies alongside automated accessibility measures, providing a relevant basis for technology specific comparisons while using a different empirical design.

The analysis concerns accessibility errors detected in rendered home pages rather than changes across software versions or all possible interface states. WebAIM applied the WAVE engine after scripts and styles had been processed. Fernandes and colleagues in 2012 demonstrated why dynamically introduced content matters for accessibility evaluation, including content produced through user events. The present analysis retains the narrower scope of the available source results and does not extend their interpretation to interactive states that were not separately recorded.

4.2 Sample and Measurement Alignment:

The sample construction in Table 1 establishes the analytical frame, while the dictionary in Table 2 defines the recorded measures. The analysis retains all 25 technologies listed in the three selected source tables: eight JavaScript frameworks, 12 JavaScript libraries and five web frameworks. Each technology contributes one group observation. The same observations are used throughout the descriptive analysis, with no additional exclusions caused by missing analytical fields.

The outcome is the published mean number of WAVE detected accessibility errors per home page associated with each technology. It is retained as a continuous numerical measure in errors per page. Technology names identify the individual groups, while the three categories organise the comparisons. These categories are treated as observed classifications rather than latent constructs or components of an index. No composite score, interaction, moderator or control variable is constructed.

The number of associated home pages describes the size of each reported group. It does not provide an independent count of unique pages across the dataset because group memberships may overlap. The analysis therefore does not sum the page counts to estimate total coverage or use them to calculate an overall mean across unique home pages. The group counts and error means remain separate measures with different analytical roles.

The reported outcome is specific to automated detection. Vigo and Brajnik in 2011 explained that accessibility metrics require interpretation in relation to their validity, reliability, sensitivity and intended application. Vigo and colleagues in 2013 further demonstrated limitations in the coverage and completeness of automated evaluation tools. These studies justify treating the source means as detected error averages rather than complete measures of accessibility, conformance or user experience.

4.3 Data Processing and Descriptive Procedures:

The three technology tables were consolidated into one workbook while preserving the published technology names, category assignments, page counts, error means and source URL. Inspection covered completeness of the recorded fields and exact duplicate rows. All 25 records contained the required analytical information, and no exact duplicates were identified. The analysis retained every record without imputation, interpolation, rescaling or outlier removal. The reported means were preserved at the precision available in the source.

Descriptive calculations were performed using Python 3.12.14 and its standard statistics functions. For the error means and associated page counts, the analysis calculated the number of valid observations, arithmetic mean, sample standard deviation, median, minimum and maximum. The standard deviation used the conventional denominator based on the number of observations less one. These calculations produced the descriptive summaries in Table 3 and describe variation among the 25 technology group records.

Each technology group received equal weight when calculating the overall descriptive summaries. This choice matches the unit of analysis and ensures that the summaries describe the set of reported technologies. It

also means that a technology associated with fewer pages contributes equally to a technology associated with more pages. The average of the group means must therefore be interpreted as an unweighted summary across technologies, not as the mean error count across the underlying home pages.

The three research questions were answered through separate comparisons within their respective technology categories. For each category, the analysis identified the lowest and highest reported mean and calculated their arithmetic difference in errors per page. Selected pairwise differences used the same subtraction procedure. Table 4 preserves the individual group means and page counts so that these comparisons can be independently checked.

The analysis retained absolute error averages rather than constructing error density or a normalised accessibility index. The source report emphasises detected errors per page and cautions that error density can change with the number of page elements without representing an equivalent change in accessibility barriers. The workbook also lacks the group specific element counts required to construct a comparable density measure.

4.4 Measurement Assessment, Inference and Reproducibility:

Measurement assessment focused on the consistency of the recorded fields and their interpretation. The workbook contains published averages rather than individual error counts, so the analysis does not independently recalculate each source mean. It preserves the source definitions and separates the mean error measure from the number of pages contributing to it. No reliability coefficient or factor analysis is applied because the measures are observed numerical summaries and identifiers rather than questionnaire based latent constructs.

The evaluation method remains an important boundary on interpretation. Alsaeedi in 2020 reported differences in the performance of accessibility evaluation tools. Frazão and Duarte in 2020 likewise identified differences in criterion coverage across evaluation engines. These findings support using a common detection method for the present comparisons, but they do not establish that the resulting rankings would remain unchanged under another tool or a broader evaluation involving users and experts. No such alternative evaluation was implemented here.

The workbook preserves the information needed to reproduce the implemented descriptive calculations: category labels, technology names, group page counts, error means and the source URL. Replication consists of retaining the same 25 records, applying equal weights to the group observations, calculating the stated summaries and comparing the reported means within each category. The original extraction date is not documented in the workbook. The methodological contribution is a transparent consolidation and comparison of published accessibility summaries, rather than a new estimator, identification strategy or validated measurement instrument.

5. Findings:

The analysis examines automatically detected accessibility errors across JavaScript frameworks, JavaScript libraries and web frameworks. It answers the three research questions through comparisons of published error averages for 25 technology groups. The findings describe the observed patterns and their measurement boundaries.

5.1 Sample Characteristics and Measurement:

The analytical sample comprises 25 technology groups from the WebAIM Million evaluation conducted in February 2021. It includes eight JavaScript frameworks, 12 JavaScript libraries and five web frameworks. Each observation represents one technology group, its number of associated home pages and its published mean number of detected errors per page. All observations contain the information needed for these comparisons.

Across the 25-technology group means, the unweighted average was 62.24 errors per page. The median was 61.60, the standard deviation was 8.09, and the observed means ranged from 40.70 to 79.00. These statistics describe the distribution of technology group means. Each technology has the same weight in this summary, regardless of the number of pages associated with it.

The WebAIM audit evaluated the rendered content of one million home pages using the WAVE accessibility engine. The overall audit average was 51.4 detected errors per page. This figure refers to the complete set of evaluated home pages, whereas the average of 62.24 refers to the 25 selected technology group means. The two figures have different units of aggregation and should not be treated as equivalent population estimates.

The outcome captures errors detectable by WAVE after scripts and styles have been applied. It does not measure every accessibility barrier or establish full conformance with accessibility guidelines. Campoverde Molina and colleagues in 2020 documented the use of automated and human evaluation in accessibility research, while Vigo and colleagues in 2013 examined the limitations of relying solely on automated tools. These findings support interpreting the reported counts as evidence of detected barriers within the scope of the evaluation method.

5.2 Accessibility Errors Across Technology Groups:

Table 4 presents the reported page count and mean number of detected errors for every technology group. The table answers the research questions at the level supported by the dataset: the average error count reported for pages associated with each technology.

Table 4: Automatically detected accessibility errors by technology group

Technology Category	Technology	Home Pages	Mean Errors Per Page
JavaScript framework	React	26,773	50.4
JavaScript framework	MooTools	8,270	54.8
JavaScript framework	GSAP	6,612	57.9
JavaScript framework	RequireJS	13,621	60.7
JavaScript framework	Prototype	5,313	62.6
JavaScript framework	Backbone.js	6,003	63.9
JavaScript framework	Angular	5,775	64.9
JavaScript framework	Vue.js	18,358	67.3
JavaScript library	Lodash	20,964	57.4
JavaScript library	Modernizr	76,072	57.8
JavaScript library	jQuery Migrate	203,527	58.5
JavaScript library	Lightbox	41,883	60.3
JavaScript library	jQuery	624,415	61.6
JavaScript library	Slick	76,858	63.8
JavaScript library	prettyPhoto	19,425	66.8
JavaScript library	jQueryUI	72,798	69.7
JavaScript library	Moment.js	18,359	71.3
JavaScript library	FancyBox	20,473	73.4
JavaScript library	Underscore.js	36,247	74.0
JavaScript library	Select2	9,567	79.0
Web framework	Ruby on Rails	5,144	40.7
Web framework	MicrosoftASP.net	58,361	56.2
Web framework	ZURB Foundation	16,654	57.4
Web framework	Bootstrap	264,238	59.1
Web framework	Animate.css	58,263	66.6

The table shows variation both within and across the three technology categories. The lowest group mean was recorded for Ruby on Rails, while the highest was recorded for Select2. These values describe the pages in the reported groups. They do not establish that either technology produced the observed errors, because a page may contain several technologies and other characteristics that affect its rendered content.

5.3 JavaScript Frameworks:

Research question 1 asks what mean numbers of detected accessibility errors were reported for home pages associated with each JavaScript framework. React had the lowest reported mean at 50.4 errors per page, followed by MooTools at 54.8 and GSAP at 57.9. Vue.js had the highest mean at 67.3. The difference between React and Vue.js was 16.9 errors per page.

The remaining framework groups occupied a narrower interval. RequireJS had a mean of 60.7, Prototype 62.6, Backbone.js 63.9 and Angular 64.9. The difference between RequireJS and Angular was 4.2 errors per page. These values show that the framework groups were not identical in their reported averages, although the available data do not show whether any observed difference would persist after accounting for page characteristics.

The group sizes also varied. React was associated with 26,773 home pages, while Angular was associated with 5,775. The difference in group size does not explain the difference in error means, but it defines the number of pages contributing to each published average. The workbook does not contain the distribution of errors within either group, so it cannot determine how widely individual page counts varied around those means. Duarte and colleagues in 2016 reported differences in automatically assessed accessibility across web development technology groups. The present framework results are consistent with the relevance of examining such groups separately. They do not establish a causal effect of framework choice, because the available

observations do not separate framework use from the content and implementation of the pages on which it was detected.

5.4 JavaScript Libraries:

Research question 2 asks what mean numbers of detected accessibility errors were reported for home pages associated with each JavaScript library. Lodash had the lowest reported mean at 57.4 errors per page, followed by Modernizr at 57.8 and jQuery Migrate at 58.5. Select2 had the highest mean at 79.0, producing a difference of 21.6 errors per page between the lowest and highest library groups.

The upper part of the distribution included Moment.js at 71.3, FancyBox at 73.4 and Underscore.js at 74.0. Their means exceeded those of Lodash, Modernizr and jQuery Migrate. The ordering provides a descriptive comparison of the reported page groups, but it cannot show whether the same kinds of errors accounted for the differences. The dataset contains total mean errors rather than error counts by accessibility criterion.

jQuery was associated with 624,415 home pages, the largest page count among the 25 technology groups, and had a reported mean of 61.6 errors per page. Select2 was associated with 9,567 pages and had a higher mean. These observations show that group prevalence and the reported error average are distinct properties. The data do not establish a relationship between the number of pages using a library and the accessibility of individual pages.

Interpretation also depends on the evaluation tool. Alsaeedi in 2020 found differences in the performance of web accessibility evaluators, and Kumar and colleagues in 2021 reported that no single evaluated tool was ideal across all assessed aspects. The library means should therefore be understood as WAVE results under the WebAIM evaluation, rather than comprehensive counts of all possible accessibility barriers.

5.5 Web Frameworks:

Research question 3 asks what mean numbers of detected accessibility errors were reported for home pages associated with each web framework. Ruby on Rails had the lowest reported mean at 40.7 errors per page. MicrosoftASP.net followed at 56.2, ZURB Foundation at 57.4, Bootstrap at 59.1 and Animate.css at 66.6.

The difference between Ruby on Rails and Animate.css was 25.9 errors per page. This was the widest range within the three technology categories. The range was 16.9 for JavaScript frameworks and 21.6 for JavaScript libraries. These comparisons describe the spread of the reported group means. They do not establish that one category has greater population variability, because the categories contain different numbers of technologies and the groups may overlap.

Bootstrap had the largest web framework group, with 264,238 associated home pages. Its mean of 59.1 was 18.4 errors per page higher than the Ruby on Rails group mean and 7.5 lower than the Animate.css group mean. The comparison identifies differences among published group averages. It cannot establish whether the frameworks, page content, design decisions or combinations of technologies contributed to those differences.

The WebAIM report cautions that the presence of a technology alongside a higher error count does not establish that the technology caused the errors. This qualification applies to the web framework results and to the other technology categories.

5.6 Answers to the Research Questions:

The first research question is answered by the eight JavaScript framework means, which range from 50.4 to 67.3 errors per page. The second is answered by the 12 JavaScript library means, which range from 57.4 to 79.0. The third is answered by the five-web framework means, which range from 40.7 to 66.6.

The evidence shows that pages grouped by detected technology had different average error counts in the WebAIM evaluation. The comparisons identify technology groups for which further accessibility examination may be useful. They do not validate a causal model or establish that changing a framework or library would produce a particular reduction in errors.

The available workbook does not contain individual page observations, within group variances or the extent of overlap between technology groups. Confidence intervals and independent group significance tests cannot therefore be calculated from these summaries. The findings remain descriptive and apply to the published technology groups in the February 2021 audit.

6. Discussion:

Accessibility barriers remain visible across the web technologies examined, even where the tools used to build pages differ. Table 4 shows variation in the mean number of detected errors within each of the three technology categories. The value of this analysis lies in bringing those published technology group measures into one comparison: it makes the range within each category visible without treating a technology name as a complete measure of accessibility. The result refines the interpretation of technology-based comparisons. A lower group average identifies a pattern in the pages evaluated; it does not establish that the technology itself makes a page accessible. Vigo and Brajnik in 2011 showed that accessibility metrics differ in their properties and intended uses, reinforcing the need to interpret a detected error count in relation to how it was produced.

The observed differences are consistent with an accessibility engineering perspective in which the rendered page is shaped by both development resources and their implementation. In Table 4, the JavaScript

framework groups span 16.9 errors per page between their lowest and highest reported means. This spread makes a single description of framework associated accessibility inadequate for the technologies examined. It does not, however, reveal the pathway through which any difference arose. Fernandes and colleagues in 2012 examined accessibility evaluation of rich internet applications and emphasized the relevance of browser processing to the content delivered to users. The present comparison extends that concern to the interpretation of published technology groups: the relevant outcome is observed on rendered pages that may combine several technologies and developer choices.

The library results add a distinct insight about the relationship between technology presence and measurement. The most prevalent library group did not have the highest reported error mean, while the highest mean belonged to a substantially smaller group. This pattern in Table 4 shows that the size of a technology group and its mean detected error count convey different information. Neither measure alone identifies which technology groups account for the most unique inaccessible pages, because pages can appear in multiple groups and the dataset does not identify the overlap. Frazão and Duarte in 2020 found that automated evaluation tools differ in their coverage of accessibility criteria. Accordingly, the observed library ranking should be interpreted as a comparison under one evaluation method, not as a complete ranking of accessibility across libraries.

The web framework results expose a constraint on interpreting large scale accessibility summaries. Table 4 reports a wider range among the five-web framework means than among the technology groups in either JavaScript category. That descriptive difference is useful for identifying where reported averages vary, but it does not establish a dominant pathway or explain why the variation occurs. The groups contain different technologies and numbers of pages, and their members may overlap. Lopes and colleagues in 2010 documented substantial variation in accessibility across a large collection of web pages. The present findings place that concern at the technology group level and show why page level evidence is needed to distinguish technology related patterns from differences in the pages where those technologies appear.

The international relevance of the findings rests on their application to web technologies used across borders. The WebAIM sample was not restricted to a single country, but the available technology summaries do not identify geographic locations. The analysis therefore cannot establish differences between advanced and emerging economies. For developers and accessibility auditors, Table 4 supports using automated group comparisons to identify pages for closer examination, followed by checks of the specific rendered content and user interactions. The World Wide Web Consortium research report on accessibility metrics in 2012 emphasized the need to assess the validity and reliability of accessibility measures. A useful next step is to combine page level technology records with automated and human evaluations so that future work can examine whether the observed group differences remain after accounting for overlapping technologies and page characteristics.

7. Conclusion and Implications:

Accessible digital services depend on the quality of the interfaces that people encounter across the web. This analysis shows that automatically detected accessibility errors differ among groups of home pages associated with different development technologies. Its contribution is a focused synthesis of published audit results across three technology categories, making variation within each category visible under a common evaluation method. The findings refine an accessibility engineering perspective by showing why a technology label alone is insufficient to characterize the accessibility of a rendered page. For developers and organisations, the results support using automated comparisons to identify groups of pages for closer review, followed by checks of content, interface behavior and assistive technology compatibility. For policymakers and institutions responsible for digital inclusion, the evidence supports accessibility assessment that combines automated detection with expert and user evaluation. Such practices can direct attention to barriers that restrict access to information and services, while avoiding decisions based solely on a technology's name or its average error count. Because the audited websites were not restricted to one country, this approach is relevant to organisations serving international audiences.

The available data provide group averages from one audit and do not identify the causes of detected errors or the extent of overlap among page groups. Future research can use page level observations to examine technology combinations, page characteristics and specific error types. Repeated evaluations could assess changes over time, while geographic information and user testing could strengthen comparisons across settings. These extensions would help determine whether the observed patterns persist under broader measures of accessibility.

References:

1. Alsaeedi, A. 2020. Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: Proposed frameworks. Information, 11, issue 1, article 40. https://doi.org/10.3390/info11010040
2. Campoverde-Molina, M., Luján-Mora, S., & Valverde, L. 2020. Empirical studies on web accessibility of educational websites: A systematic literature review. IEEE Access, 8, 91676 to 91700. https://doi.org/10.1109/ACCESS.2020.2994288

3. Duarte, C., Matos, I., Vicente, J., Salvado, A., Duarte, C. M., & Carriço, L. 2016. Development technologies impact in web accessibility. Proceedings of the 13th International Web for All Conference, article 6, 1 to 4. Association for Computing Machinery. https://doi.org/10.1145/2899475.2899498
4. Fernandes, N., Costa, D., Neves, S., Duarte, C., & Carriço, L. 2012. Evaluating the accessibility of rich internet applications. Proceedings of the International Cross Disciplinary Conference on Web Accessibility, article 13. Association for Computing Machinery. https://doi.org/10.1145/2207016.2207019
5. Frazão, T., & Duarte, C. 2020. Comparing accessibility evaluation plug ins. Proceedings of the 17th International Web for All Conference, article 20, 1 to 11. Association for Computing Machinery. https://doi.org/10.1145/3371300.3383346
6. Kumar, S., Shree, J. D. V., & Biswas, P. 2021. Comparing ten WCAG tools for accessibility evaluation of websites. Technology and Disability, 33, issue 3. https://doi.org/10.3233/TAD-210329
7. Lopes, R., Gomes, D., & Carriço, L. 2010. Web not for all: A large scale study of web accessibility. Proceedings of the International Cross Disciplinary Conference on Web Accessibility, article 10. Association for Computing Machinery. https://doi.org/10.1145/1805986.1806001
8. Vigo, M., & Brajnik, G. 2011. Automatic web accessibility metrics: Where we are and where we can go. Interacting with Computers, 23, issue 2, 137 to 155. https://doi.org/10.1016/j.intcom.2011.01.001
9. Vigo, M., Brown, J. A., & Conway, V. L. 2013. Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. Proceedings of the 10th International Cross Disciplinary Conference on Web Accessibility, article 1. Association for Computing Machinery. https://doi.org/10.1145/2461121.2461124
10. WebAIM. 2021. The WebAIM Million: An annual accessibility analysis of the top 1,000,000 home pages. https://webaim.org/projects/million/2021/
11. World Wide Web Consortium. 2012. Research report on web accessibility metrics. https://www.w3.org/TR/accessibility-metrics-report/
12. World Wide Web Consortium. 2017. Accessible rich internet applications 1.1. https://www.w3.org/TR/2017/REC-wai-aria-1.1-20171214/
13. World Wide Web Consortium. 2018. Web content accessibility guidelines 2.1. https://www.w3.org/TR/2018/REC-WCAG21-20180605/