

The Most Violated WCAG 1.0 Guidelines by the Developers of e-Government Websites

M. Basel Al Mourad

Abstract—Most governments today are moving toward information society where e-Government websites are universally accessible by all citizens. Accessible websites means that people with disabilities can use the Web and the website developers must follow the WWW Consortium (W3C)'s Web Content Accessibility Guidelines (WCAG). This paper seeks to examine the most violated accessibility guidelines by the developer of the web portals. For this purpose, we examined 21 e-Government websites. We identified and discussed the most violated WCAG. Our findings may raise awareness to the web developer about these violations and encourage them on how to build better and accessible websites.

Index Terms—Web accessibility, WCAG 1.0 checkpoint violation, disability, e-government.

I. INTRODUCTION

Currently most Web sites have accessibility barriers that make it difficult or impossible for many people with disabilities to use the Web [1]. As more accessible Web sites and software become available, people with disabilities are able to use and contribute to the Web more effectively.

Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities (i.e. including visual, auditory, physical, speech, cognitive, and neurological disabilities) can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging. Furthermore, Web accessibility also benefits people without disabilities. For example, a key principle of Web accessibility is designing Web sites and software that are flexible to meet different user needs, preferences, and situations [2].

The World Wide Web Consortium's (W3C) commitment to lead the Web to its full potential includes promoting a high degree of usability for people with disabilities [3]. The (W3C)'s Web Accessibility Initiative (WAI) develops strategies, guidelines, and resources to help make the Web accessible to people with disabilities. The WAI consists of several working groups and special interest groups, each with its own focus. The Web Content Accessibility Guidelines Working Group (WCAG) produces guidelines, techniques and other supporting documents relating to the accessibility of Web content. Web content refers to any information you may find on a Web site: text, images, forms, sound, video, etc.,

regardless whether these were produced on the server side or on the client side (with a client-side scripting language such as JavaScript). Thus, the guidelines also apply to rich internet applications.

The working group consists of representatives from industry, accessibility consultancies, universities, organizations that represent end users, and other accessibility experts.

The working group published the Web Content Accessibility Guidelines 1.0 (WCAG 1.0) [3] as a W3C Recommendation in 1999, followed by techniques and documents in 2000. In 2001, the working group started to work on WCAG 2.0, which became a W3C Recommendation on 11 December 2008

Although WCAG 2.0 had emerged as an ISO/IEC 40500:2012 International accessibility standard and it has been recommended for adoption by the W3C, WAI's most e-portal a developer have been using the WCAG 1.0 guidelines as reference to demonstrate their compliance to web accessibility principles. This was reflected in their local accessibility policies, website design, maintenance, and accessibility testing activities. Section II discusses the research motivation. Section III presents the web content accessibility guidelines. Section IV presents the research methodology. Section V presents our the most violated accessibility guidelines and discusses our recommendation. Finally Section VI presents the conclusion.

II. RESEARCH MOTIVATION

Most governments today are promoting the transition of their countries towards an information society where e-Government websites are becoming the primary gateways to citizens and businesses for government information and e-service delivery [4]. E-Government can be broadly defined as the unification of information and communication technologies, and administrative practices to provide government e-services to citizens, businesses and other e-Governments [5]. The benefits of online government e-services include better efficiency, user convenience and more citizen political involvement [6]. To enable all citizens to benefit from the full potential of e-Government services, it is important to secure universal accessibility. This accessibility enables people with disabilities to take full advantage of the information and services offered by e-Governments; the same way a person with no disability would.

To enable web accessibility, most e-governments have been using the WCAG 1.0 guidelines as reference to probe their compliance to web accessibility principles. This was

Manuscript received January 10, 2015; revised February 16, 2016. This work was supported by Zayed University research grant R14040.

M. B. Almourad is with Zayed University, P. O. Box 19282, Dubai, UAE (e-mail: basel.almourad@zu.ac.ae).

reflected in their local accessibility policies, website design, maintenance, and accessibility testing activities. Numerous research for assessing the usability of e-Government and commercial web sites revealed that most of the websites were unusable for people with disabilities. For example, [1] assessed the accessibility of e-Government websites for 12 developing and developed countries. She identified serious accessibility issues for the tested e-Government sites, even for websites belonging to governments who stated adherence to W3C accessibility standards and UN legislations.

A survey in 2008 by the UK Public Accounts Committee found that in the past six years the quality of government web sites has only improved slightly and one in six has actually worsened, and one-third of sites failed to meet the Cabinet Office's accessibility standards [7].

An accessibility case study related to the 21 Dubai e-government websites revealed that only two web sites have fully complied with the WAI-A conformance level and thus met the minimum WCAG accessibility requirement for people with disability. In addition, none of the 21 e-Government websites was found to be fully compliant with the WAI-AA or WAI-AAA web accessibility conformance level [8], [9].

This research work is motivated by the need of raising awareness among developers and webmasters about the importance of universal accessibility. We have tried to highlight the most violated WCAG 1.0 checkpoints. This would help developer to pay more attention to those checkpoints and try to avoid the same mistakes that many developers are falling in.

III. WEB CONTENT ACCESSIBILITY GUIDELINES

TABLE I: WAI CONFORMANCE CLAIMS

Conformance Level	Website Accessibility Checkpoint
WAI-A (basic accessibility)	All priority 1 checkpoints are met. This is the minimum (basic) W3C requirement. Otherwise one or more groups of people will find it impossible to access information from the website. This is the minimum requirement and must be met.
WAI-AA (intermediate accessibility)	All priority 1 and 2 checkpoints are satisfied; otherwise one or more groups of people will find it difficult to access information from the website. This conformance level status should be met, as it will remove significant barriers to accessing Web documents.
WAI-AAA (high accessibility)	All priority 1, 2 and 3 checkpoints are satisfied; otherwise one or more groups of people will find it somehow difficult to access information from the website. Web developers to improve access to Website documents may address this conformance level status.

WCAG 1.0 consists of 14 guidelines [10], which are general principles of accessible design. Each guideline covers a basic theme of web accessibility and is associated with one or more checkpoints describing how to apply that guideline to particular features of webpages. The guidelines have a total of 65 checkpoints. Each checkpoint is assigned to a priority level that is based on the checkpoint's impact on accessibility. The full listing of the guideline checkpoints under each priority is

provided in Appendix 1. Accordingly, the W3C Web Accessibility Initiative (WAI) defines three possible accessibility conformance levels (sometimes called priority levels), as illustrated in Table I.

IV. RESEARCH METHODOLOGY

We have adopted a quantitative research method to measure the most violated WCAG 1.0 checkpoints and provide guidance for the web developer and webmasters on how to avoid these violations. We selected 21 Dubai e-government websites as a case study for our measurement testing. Several automatic accessibility tools are available to assess web accessibility and which and how many WCAG 1.0 checkpoints are violated. We selected the 1.10 standalone version of the Web Accessibility Assessment Tool (WaaT) (<http://sourceforge.net/projects/waat>) to measure the WCAG 1.0 violated checkpoints. The tool can test all the conformance levels (A, AA, and AAA).

WaaT is one of the free tools created within the premises of the ACCESSIBLE project, which is a targeted research project (STREP) within the European Union's ICT program of FP7 (www.accessible-eu.org/index.php/project.html). WaaT provides a personalized accessibility assessment methodology, allowing users to select between different set of accessibility guidelines, impairments, and personas. The tool can test conformance against all the conformance levels (A, AA, and AAA). WaaT enables the tester to choose one page (such as the home or index page) or the entire web site pages for automated accessibility testing. For each accessibility conformance level, WaaT provides a detailed report of detected issues, as well as additional tagged warnings that require manual inspection and human judgment on the part of the tester. The tester can extract the number of issues that corresponds to violated checkpoints. WaaT also provides details about the lines where the errors and warnings appeared in the HTML source code. These assessment results are also available as a single PDF report or as an Evaluation and Report Language (EARL) machine-readable (RDF) report. We run WaaT to test accessibility conformance on the 21 Dubai e-Government Websites. For each website we extracted the number of issues for each violated checkpoint. We gathered all the extracted numbers in an Excel sheet for analysis.

V. RESULT AND DISCUSSION

Table II summarizes the number of violation of every checkpoint for each tested Dubai e-government website. The website names are hidden for confidentiality and have been renamed O1 to O21. We have only included the checkpoints that are violated. The checkpoints that are not violated are omitted from this discussion. As shown in Table II, none of the 21 tested websites is fully free from accessibility violation. However, websites (O7, O12, O1 and O20) have generated few violations. As shown in the chart in Fig. 1, checkpoints 11.2, 3.4, 5.5, 1.1, 6.3, 12.4, 4.3 and 3.5 are the most violated checkpoints. In the following, we will be discussing the checkpoints that are violated and ignore the checkpoints that

are violated fewer times and less frequently.

Checkpoint 11.2 is violated 1871 times and by nearly all the tested websites. The checkpoint 11.2 is recommendation for “Avoiding deprecated features of W3C technologies”. It

seems that tags and attributes that are not on the list of approved tags and attributes are still used. For example, the and the <color> attributes have been deprecated but still appear to represent the font and color with HTML tags.

TABLE II: CHECKPOINT VIOLATIONS BY EVERY ORGANISATION

Org.	Priority 1				Priority 2								Priority 3				
	1.1	6.1	6.2	6.3	3.2	3.4	3.5	3.6	7.3	11.2	12.2	12.3	12.4	1.5	4.3	5.5	10.4
O1	32	3	5			93	1			155			2	3	3	40	1
O2	57			7	1	46	1		1	233		1	4		3	59	
O3						38	1			63		1	3			13	2
O4	26			64		24	1		1	137			2		1	5	
O5	37			10		29	1	3		101			4	4	1	43	5
O6	11					89	1			70	1	1			1	111	
O7	2							1		1					1		
O8	31			5		122			1	123			3		1	70	2
O9	12			1		41	1			20		1	3		1	29	2
O10	18			10		216	1			158			2			147	
O11	31			10	1	206	1			321			8		1	66	1
O12			1	1		1	2			1			1		1		1
O13			1	1		1	1			1			1		1		1
O14	31			3		16	1			56			3		1	139	6
O15			2			2				13					1	1	1
O16	4					47	1			109			6	3	1	18	1
O17	88		16	16		180			1	185			2		1	50	1
O18	39					98	1			20			1		1		1
O19	20		1	6		53	1		1	91			2				
O20	1									1							
O21	10			3		11	1	1		12			2			5	2
Sum	450	3	26	137	2	1313	18	4	5	1871	1	4	49	10	20	796	27

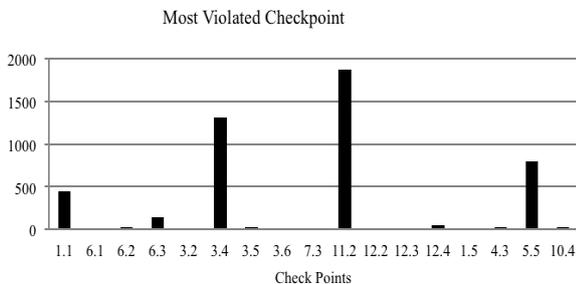


Fig. 1. The most violated check points.

Web programmers need to be aware that it is important to avoid deprecated elements so that the Web page is compliant with current browser technology. Web programmers are encouraged to use style sheet as using deprecated elements will make a website incompatible with the latest version of HTML and may, despite good intentions, decrease its accessibility.

Checkpoint 3.4 is violated 1313 times. It is a recommendation for “Using relative rather than absolute units in markup language attribute values and style sheet property values”. Relative length units specify a length relative to another length property and this means that the unit is attached to another property such as font. Relative units can also be changed and do not stay fixed as an absolute unit would. For example, if a font size was set using absolute units it would remain that way regardless if the user wanted to change it. Users that have sight disabilities would find this

checkpoint useful as they sometimes need the font on web pages to be made larger. A website that uses style sheets with relative units will allow users to change the font size of a page from their browser to make it more accessible/bigger. However, if absolute units were used this would not be possible as this unit is fixed.

Checkpoint 5.5 is violated 796 times. It is a recommendation for “Providing summaries for tables” using SUMMARY attribute in a table tag. Users with sight impediments and who use screen readers may need information about how the information within a table relates to each other. A summary would give the non-visual user a clear indication of the relationships within a table and how the content in the table relates to the other content on the page.

Checkpoint 1.1 is violated 450 times. It is a recommendation for “Providing a text equivalent for every non-text element”. The non-text element includes images, audio tracks, videos, Client-Server maps, and programmable objects such as Java applets. Disabled users that need to use assistive technologies such as screen readers and Braille terminals must have alternative text to tell the user what the content is. If this is not the case then the user with the slight and possible hearing impediment cannot access the content. Certain Disability Discrimination Acts would describe this as discriminating against these users due to the content not being accessible. Disadvantaged users who have limited/slow Internet speed might disable the above elements on their

computers or browsers to allow the Website to load quicker. Such users need alternative text to inform them of the content.

Checkpoint 6.3 is violated 137 times. It is a recommendation for “Ensuring that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page”. Many developers think that this checkpoint will prevent them from using scripts. This checkpoint encourages developers to find an alternative method to get the job done without scripts. For example, if scripting is used for things such as interactive navigation, developers should make sure that users could achieve the navigation without scripting.

Checkpoint 12.4 is violated 49 times. Although the number of violation is less than the other checkpoints, it is violated by almost all the websites. The recommendation for this checkpoint is “Associate labels explicitly with their controls”. This applies to form controls where each control on a form such as a check, text box, etc., should include a label informing the user why they need to check or add text in the box provided. Users with visual impairments that need aid such as screen readers to access a page will find this checkpoint very helpful. Screen reader users cannot read the content or see labels next to form controls and so it is important that these labels are implemented so a screen reader can understand what the text says and which part of the form it belongs to.

Checkpoint 4.3 is violated 20 times. The recommendation for this checkpoint is “Identify the primary natural language of a document”. The Web developer should set the "lang" attribute on the HTML element. In XML or uses "xml:lang". Server operators should configure servers to take advantage of HTTP content negotiation mechanisms so that clients can automatically retrieve documents of the preferred language. It's very important for screen readers to know the base language. Screen readers can adapt their pronunciation to the language that is specified. When the language is not specified the program will have to guess, or will request the user to specify the language. A further benefit is that Search-engine spiders recognize the language in which the content on a page is written. Some search engines let visitors filter the search results by their preferred language. Search engines can guess the language on a page when this is not indicated in de mark-up (domain name, words in the content), but this may lead to Dutch pages being recognized as German. Also, spell checks and automated translations produce better results.

Checkpoint 3.5 is violated 18 times. The recommendation for this checkpoint is “Use header elements to convey document structure and use them according to specification”. The checkpoint recommends using HTML headings such as <H1> <H2> <H3> in a structure to allow an Internet user to see the structure of the page. Skipping heading-elements or using heading-elements for font-formatting lead to confusion with the content-structure. Distinguishing the structure of a document offers a number of advantages, including improved accessibility, manageability, and portability.

Over all the number of errors in priority 1, 2 and 3 are 616, 3267 and 853. Clearly, the conformance to the WAI-A (basic accessibility) was the highest followed by WAI-AAA (high accessibility), and then by WAI-AA (intermediate

accessibility).

VI. CONCLUSION

This paper attempted to investigate the extent to which e-Government websites conformed to WCAG 1.0 accessibility guidelines. Many of the guideline checkpoints are violated. We have identified the most violated guideline checkpoints. Our aim is to raise the awareness to the web developer so they can avoid these violations in the future and ultimately making e-Government web sites accessible by all spectrum of the society.

APPENDICES

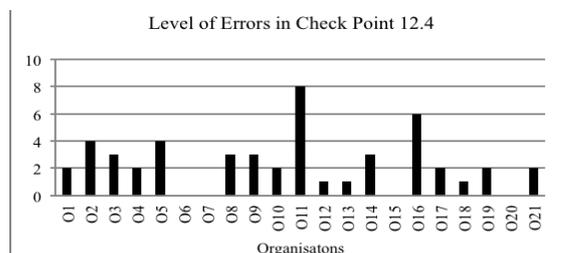
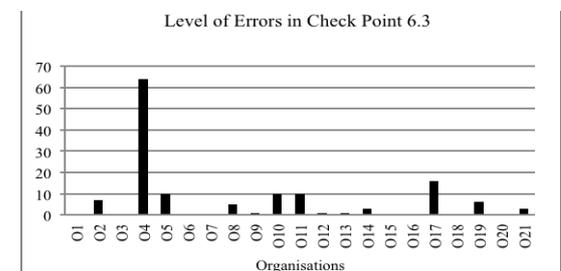
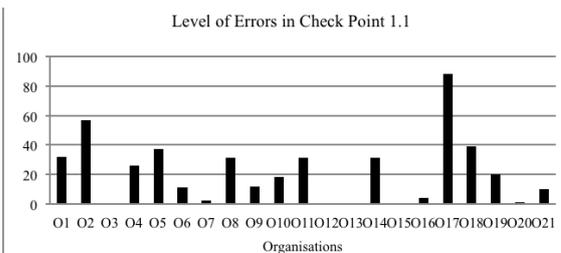
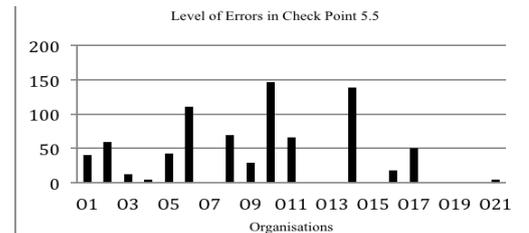
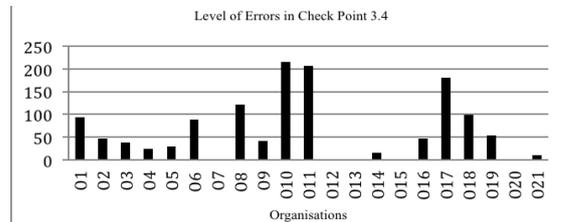
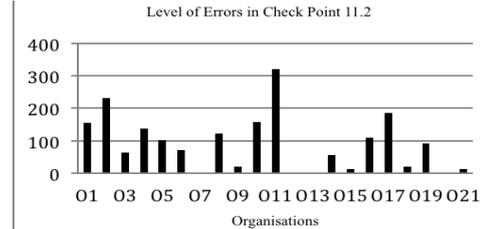
APPENDIX I: LISTING OF THE GUIDELINE CHECKPOINTS UNDER EACH PRIORITY

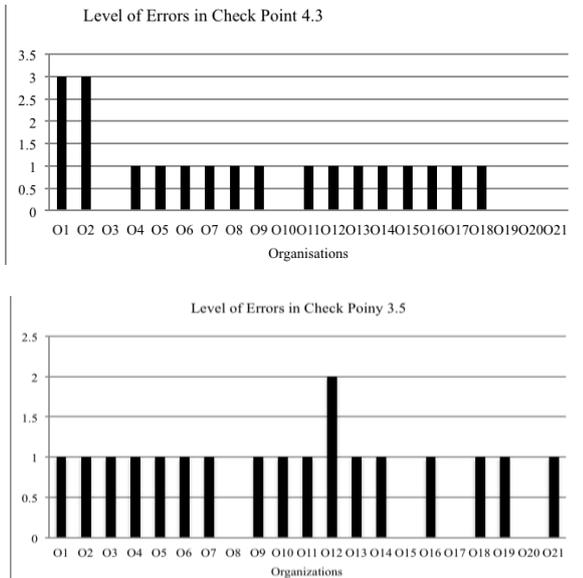
Priority 1 Check points	
1.1	Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ascii art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video
1.2	Provide redundant text links for each active region of a server-side image map
1.3	Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation
1.4	For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation
2.1	Ensure that all information conveyed with color is also available without color, for example from context or markup
4.1	Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions)
5.1	For data tables, identify row and column headers
5.2	For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells
6.1	Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document
6.2	Ensure that equivalents for dynamic content are updated when the dynamic content changes
6.3	Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page
7.1	Until user agents allow users to control flickering, avoid causing the screen to flicker
8.1 a	Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies
9.1	Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape
11.4	If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page
12.1	Title each frame to facilitate frame identification and navigation
14.1	Use the clearest and simplest language appropriate for a site's content
Priority 2 Check points	
2.2 a	Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen. [for

	images] .
3.1	When an appropriate markup language exists, use markup rather than images to convey information.
3.2	Create documents that validate to published formal grammars.
3.3	Use style sheets to control layout and presentation.
3.4	Use relative rather than absolute units in markup language attribute values and style sheet property values.
3.5	Use header elements to convey document structure and use them according to specification.
3.6	Mark up lists and list items properly.
3.7	Mark up quotations. Do not use quotation markup for formatting effects such as indentation.
5.3	Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).
5.4	If a table is used for layout, do not use any structural markup for the purpose of visual formatting.
6.4	For scripts and applets, ensure that event handlers are input device-independent.
6.5	Ensure that dynamic content is accessible or provide an alternative presentation or page.
7.2	Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).
7.3	Until user agents allow users to freeze moving content, avoid movement in pages.
7.4	Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.
7.5	Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.
8.1b	Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies
9.2	Ensure that any element that has its own interface can be operated in a device-independent manner.
9.3	For scripts, specify logical event handlers rather than device-dependent event handlers.
10.1	Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.
10.2	Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.
11.1	Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.
11.2	Avoid deprecated features of W3C technologies.
12.2	Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.
12.3	Divide large blocks of information into more manageable groups where natural and appropriate.
12.4	Associate labels explicitly with their controls.
13.1	Clearly identify the target of each link.
13.2	Provide metadata to add semantic information to pages and sites.
13.3	Provide information about the general layout of a site (e.g., a site map or table of contents).
13.4	Use navigation mechanisms in a consistent manner.
Priority 3 Check points	
4.2	Specify the expansion of each abbreviation or acronym in a document where it first occurs.
4.3	Identify the primary natural language of a document.
9.4	Create a logical tab order through links, form controls, and objects.
9.5	Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.
10.3	Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.
10.4	Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.
10.5	Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.
11.3	Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.)

13.5	Provide navigation bars to highlight and give access to the navigation mechanism.
13.6	Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.
13.7	If search functions are provided, enable different types of searches for different skill levels and preferences.
13.8	Place distinguishing information at the beginning of headings, paragraphs, lists, etc.
13.9	Provide information about document collections (i.e., documents comprising multiple pages.).
13.10	Provide a means to skip over multi-line ASCII art.
14.2	Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.
14.3	Create a style of presentation that is consistent across pages.

APPENDIX II: LEVEL OF ERRORS IN CHECK POINTS





[4] M. Goodwin, D. Susar, A. Nietzio, M. Snaprud, and C. S. Jensen, "Global web accessibility analysis of national government portals and ministry web sites," *Journal of information Technology and Politics*, vol. 8, no.1, pp. 41-67, 2011.

[5] E. Deakins and S. M. Dillon, "E-government in New Zealand: The local authority perspective," *The International Journal of Public Sector Management*, vol. 15, no. 5, pp. 375-398, 2002.

[6] R. Freeman and P. Loo, "Web 2.0 and e-government at the Municipal Level," in *Proc. 5th International Conference on e-Government*, Boston, MA, 2009

[7] E. Stewart. (April 2008). Watchdog calls for shake-up of government websites. [Online]. Available: <http://www.guardian.co.uk/politics/2008/apr/29/whitehall.egovment>.

[8] M. B. Al Mourad and F. Kamoun, "Accessibility evaluation of Dubai e-Government websites: Findings and implications," *Journal of E-Government Studies and Best Practices*, 2013.

[9] F. Kamoun and M. B. Almourad, "Accessibility as an integral factor in e-government web site evaluation: The case of Dubai e-government," *Journal of Information Technology & People*, vol. 27, no. 2, 2014.

[10] WCAG 1.0. (September 14, 2011). Web Content Accessibility Guidelines (WCAG). [Online]. Available: <http://www.w3.org/TR/WCAG10/>

REFERENCES

[1] J. Kuzma, "Global E-Government web accessibility: A case study," in *Proc. the British Academy of Management 2010 Conference*, University of Sheffield, UK, September 2010.

[2] J. J. Garrett and J. James, "Elements of user experience, the user-centered design for the web and beyond," *Pearson Education*, 2010.

[3] WCAG. (December 10, 2011). Web content accessibility guidelines (WCAG). [Online]. Available: <http://www.w3.org/WAI/intro/wcag.php>



M. B. Almourad has received his BSc in informatics engineering from Aleppo University, Syria, and his PhD degree in computer science from Cardiff University of Wales, United Kingdom. He has served in various international universities. He is currently an associate professor in Zayed University, Dubai, UAE. Dr. Al-Mourad is active in several research areas where he published numerous articles including: IT service management, heterogeneous data management, semantic web, web accessibility and cloud computing.