UDC 004.9: [004.5 + 004.738.5 + 376]
DOI: 10.31891/CSIT-2021-5-12

TETIANA ZHYROVA, NATALIIA KOTENKO, VOLODYMYR TOKAR,
KARYNA KHOROLSKA, BOHDAN BEBESHKO
Kyiv National University of Trade and Economics

TESTING THE ACCESSIBILITY OF WEB-APPLICATIONS

The constant growth of digital information and online services highlights the need to protect the rights of people with disabilities in terms of web accessibility. Despite EU regulations aimed at harmonizing this process, some problems still remain unsolved. This article sheds light on legal documents protecting the rights of people with disabilities regarding web accessibility. The conducted profound analysis of abundant research and applied papers showed the lack of systematization of accessibility testing standards. The authors disclosed modern approaches to testing the accessibility of web applications depending on the special needs of users and elaborated the checklist for testing the accessibility of web-applications.

Keywords: accessibility, inclusive design, testing, web-applications.

ТЕТЯНА ЖИРОВА, НАТАЛІЯ КОТЕНКО,
ВОЛОДИМИР ТОКАР, КАРИНА ХОРОЛЬСЬКА, БОГДАН БЕБЕШКО
Київський національний торговельно-економічний університет

ТЕСТУВАННЯ ДОСТУПНОСТІ ВЕБ-ДОДАТКІВ

Постійне зростання цифрової інформації та онлайн-послуг підкреслює необхідність захисту прав людей з інвалідністю з точки зору веб-доступності. Незважаючи на правила ЄС, спрямовані на гармонізацію цього процесу, деякі проблеми все ще залишаються невирішеними. У цій статті було розглянуто нормативно-правові акти, які спрямовані на захист прав людей з обмеженими можливостями щодо веб-доступності, проаналізовано науково-практичну літературу, практичні публікації та інтернет-видання із зазначеної проблематики. З'ясовано, що питання доступності програмного забезпечення потребує ґрунтовного аналізу та систематизації, а також є необхідність у вдосконаленні стандартів тестування доступності веб-додатків. Здійснений аналіз низки досліджень та прикладних статей показав відсутність систематизації стандартів тестування доступності.

Як правило, доступність асоціюється саме з вадами зору, проте перелік вад, які потребують уваги щодо розробки універсального дизайну, набагато більше. Таким чином, для того щоб створювати додатки, які мають універсальний дизайн, доцільно здійснити класифікацію потенційних користувачів та визначитись з потребами кожної групи. Сформовано перелік вад, які потребують уваги щодо розробки універсального дизайну, проте зазначені види порушень не є повним переліком усіх можливих, але є найбільш поширеними за статистичними даними ВООЗ. Кожна з зазначених вад потребує відповідного дизайну та архітектури додатку щоб зробити його універсальним. В межах даної статті сформульовано вимоги до веб-додатку в залежності від когнітивних порушень.

Проаналізовано стандарти з питань доступності, систематизовано підходи до тестування доступності веб-додатків у залежності від особливих потреб користувачів. Тестування доступності допомагає зробити веб-додатки доступними для більшої кількості користувачів. Для того, щоб у повній мірі здійснити тестування доступності необхідно: визначити цільову аудиторію, визначити яким чином вони використовують додаток, встановити потенційні проблеми, які можуть виникнути у процесі користування веб-додатками. Автори розкрили сучасні підходи до перевірки доступності веб-додатків залежно від особливих потреб користувачів та розробили контрольний список для перевірки доступності веб-додатків, що враховує когнітивні порушення та опирається на стандарт WCAG 2.1.

Ключові слова: доступність, інклюзивний дизайн, тестування, веб-додатки.

Introduction

Issues of protection of the rights of people with disabilities both in EU member-states and Ukraine are outlined in many legislative documents [1, 2, 3, 4]. Although the EU and its partner countries are constantly working on improvement of conditions for people with disabilities, a number of issues remain unsolved, including information accessibility.

Digital accessibility becomes more popular as the amount of information and online services increases every day. The rapid development of information technology makes it possible to solve most problems of information accessibility for people with disabilities. However, this requires additional legislature to regulate the accessibility of hardware and software. To address these issues on October 26, 2016, the EU Parliament and the Council of the European Union passed the EU Web Accessibility Directive and on June 7th, 2019 the European Union formally adopted the European Accessibility Act. The Directive aims at creating a more standardized and harmonized framework of user-friendly in terms of accessibility websites and mobile applications of public sector bodies [5]. The Directive complements the European Accessibility Act which covers a wide range of products and services also in the private sector. Moreover, the EU legislation supports people with disabilities in other areas including electronic communications, audio-visual media services, eBooks, e-commerce and ICT equipment [6]. In accordance with this Directive, the following deadlines were set: September 23, 2019, all new public sector websites and apps should conform to the Directive; September 23, 2020, all new and existing public websites must conform to the Directive; June 23, 2021, all new and existing mobile apps must conform to the directive [5].

Although there are generally no requirements for the accessibility of private sector websites and mobile applications in EU member-states, business entities follow trends in protection of the rights of people with disabilities to avoid losing a significant segment of users. Therefore, software developers of websites and mobile applications must solve two tasks: software developing and testing.

Related Works

The desire to improve the quality of life for people with disabilities has been around for decades. The topic of accessibility is not new, but it is still relevant, as evidenced by the goal of the Disability Rights Strategy for 2021-2030, which aims at addressing various problems faced by people with disabilities.

The IT industry must also be accessible. The increase in digital information, the continued development of IT, and the transition to online life (intensified by the pandemic) require software to be adaptive for people with disabilities. The topic of accessibility of IT to people with disabilities was considered by the following scientists: Hosking D. [7], Krebs I. [8], Neveen I. [9], Won-Kyung Lee [10], Alves D. [11], Goncalves de Branco R. [12], Fathauer L. [13], etc. An analysis of pieces of research has shown that the software requirements for accessibility are not fully understood. Moreover, software accessibility testing is almost ignored as it was stated by K. Wille [14], N. Patil [15], E. Park [16], N. P. Kasaghatta Ramachandra [17], etc. It is worth mentioning that most researchers consider the accessibility of software for the visually impaired people and ignore other disabilities that also require a specially organized user interface.

The analysis of scientific literature, applied articles and online publications has shown that the issue of software accessibility requires systematization, and there is a need to improve the standards for accessibility testing. Thus, given the scope of the issue, the purpose of this article is a thorough analysis of accessibility standards, research and systematization of existing approaches to manual testing of web-application accessibility depending on the special needs of users, as well as elaborating a checklist for web-application accessibility testing.

Web-application Inclusiveness Testing

According to the Directive (EU) 2016/2102 of the European Parliament and of the European Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies [18], it is the web-applications of public sector bodies that require accessibility. According to the World Health Organization, more than one billion people, representing 15% of the world population, live with some form of disability, and as the population ages, this number is steadily increasing [19]. That is why to ensure equal rights and opportunities for people with disabilities, all software must be accessible, including web-applications.

Before considering the features of developing and testing web applications that would provide accessibility for people with disabilities, it is advisable to find out their needs, as well as to define such terms as “disability”, “person with disabilities”, and “accessibility”.

A disability is any condition of the body or mind (impairment) that makes it more difficult for the person with the condition to do certain activities (activity limitation) and interact with the world around them (participation restrictions) [20].

A person with a disability is a person with a persistent dysfunction of the body, which in interaction with the external environment can lead to a restriction of his or her life, as a result of which the state is obliged to create conditions for exercising his or her rights on equal basis with other citizens [20].

An analysis of Ukrainian and foreign information sources indicates that the terms “handicapped” and “a person with disabilities” are mostly synonymous, although in our opinion the latter category also includes people who are slightly limited in their physical or psychological capabilities, but do not receive social protection from the state and do not have the official status of a disabled person.

Quite often the terms “persons with special needs” and “persons with disabilities” are used as synonyms, but in the context of educational needs. Thus, according to the legislation of Ukraine, a person with special educational needs is a person with a disability who needs additional support to ensure higher education [21], and a person with disabilities is a person with physical and / or mental disabilities, which prevent the assimilation of educational programs without creating special conditions for education [22]. In our opinion, when it comes to the accessibility of web-applications and mobile applications, these terms are synonymous. For the objectives of this article, persons with special needs will be understood as those who have certain physical or mental disabilities and for whom the inclusive design of digital resources will ensure the full use.

Inclusive design is the process of making relevant software, which includes both websites and mobile applications, accessible to different audiences. Accessibility is the goal of this process, when different people are equally comfortable using the product. We also need to shed light on the concept of “universal design”. This is a broader concept, it originated in the field of architecture and indicates how comfortable the environment is for people with various physical and mental disorders [23].

As a rule, accessibility is associated with visual impairments, but the list of defects that need attention in the development of universal design is broader. Thus, in order to create applications that have a universal design, it is advisable to classify potential users and determine the needs of each group. The roaster of types of disorders that must be considered when developing inclusive applications embrace:

1. Sensory disturbances:

– Eyesight:

- moderate visual impairment: farsightedness, myopia, astigmatism;
- severe visual impairment: cataracts, glaucoma;
- blindness: complete, partial and color blindness. Color blindness: achromatopsia, monochromasia, dichromasia (protanopia, deuteranopia, tritanopia), trichromasia.

Visual problems require the ability to customize the interface, increasing or decreasing the text and adding contrast, and blindness requires sound with screen readers, color blindness requires adjusting the appropriate color design.

– Hearing: deafness.

The person relies entirely on sight, so the design should include easily readable information, audio information should be duplicated in the text format, video information should be accompanied by captions and translation into sign language.

2. Physical defects (motor disorders):

– absence or paralysis of one or two upper extremities, or parts thereof;
– impaired coordination of movements;
– muscular dystrophy;
– multiple sclerosis;
– Parkinson's disease;
– cerebral palsy.

Musculoskeletal disorders dictate the requirements for the size of elements on the screen and ease of use, as well as the interaction of applications with special input-output devices and other applications: screen reader, Braille display, speech recognition and voice control, sip-and -puff (SNP), inhale-exhale, switches, etc.

3. Cognitive disorders (disorders of memory, gnosis, praxis, speech, intelligence and attention):

– epilepsy;
– dementia;
– autism spectrum disorders;
– dyslexia;
– distracted attention;
– cognitive impairment caused by age changes [24];

In general, cognitive impairments require quality and simple content, available in a user-friendly interface that does not require unnecessary action.

These types of disorders are not a complete list of all possible ones, but they are the most common ones according to WHO statistics. Each of these flaws requires an appropriate application design and architecture to make it versatile. In this article, we will explore testing the accessibility of web applications that address some cognitive impairments. Each type of defect requires consideration of certain features (Table 1).

Table 1

Requirements to Web-applications depending on cognitive impairment

Cognitive impairment	Peculiarities	Requirements to Web-applications
Epilepsy	Flashing lights, images, and repetitive patterns may cause seizures in persons with photosensitive seizure disorder.	No flicker in the application. The design should be in calm tones, without repetitive patterns.
Dementia	Dementia is a syndrome in which there is deterioration in memory, thinking, behavior and the ability to perform everyday activities.	The application should have a surprisingly easy and clear interface. Content should be written in easy, understandable language, short sentences without the use of specific vocabulary and abbreviations. Any features that a potential user can use should have a short commentary with video accompaniment. If the web application involves filling out a form, it must be with a label, which clearly indicates what exactly the user needs to enter. The text of the placeholder should not be the main one, but only supplement the information from the Label. It is desirable that the user should be able to consult with the operator in case of questions.
Dyslexia	Dyslexia is a learning disorder that involves difficulty with reading due to problems of identifying speech sounds and learning how they relate to letters and words (decoding). Also called reading disability, dyslexia affects areas of the brain that process language.	All content of the application must be voiced. If the user fills out the form, it is desirable to voice the completed information as a check.
Attention Deficit Hyperactivity Disorder	The syndrome can have various manifestations: inattention, distraction, inability to concentrate; constant mobility, impulsiveness, including wasting money.	The content of the application should be quite short and informative, have a clear structure. Particular attention should be paid to self-checking the form. It is recommended that the user immediately receive information about the error in the filled field and a prompt for correction.

In order to provide people with disabilities with a high level of content accessibility, the Web Accessibility Initiative (WAI) within the World Wide Web Consortium (W3C) has developed the Web Content Accessibility Guidelines (WCAG) standard. The latest version of WCAG 2.1, was published on June 5, 2018 and contains all the requirements of previous versions. Thus, if the web application complies with WCAG 2.1, it will automatically also comply with all other versions of WCAG.

Each new version simply adds new recommendations to existing ones. The previous version of WCAG 2.0 was adopted as the ISO ISO / IEK 40500 standard, and WCAG 2.1 - as the standard of the European Union EN 301 549. Recommendations for ensuring the accessibility of web-content are the most universal, with a large number of countries adapting these requirements to their laws. A preliminary version of WCAG 3.0 has now been developed, which takes into account the accessibility not only of websites, but of all web-applications in general.

The standard contains a number of recommendations on what needs to be done to make the web-content accessible to people with disabilities. Recommendations are grouped according to four principles: perception, manageability, comprehensibility and reliability. These recommendations are called success criteria, and according to them, the application can receive one of three levels: A, AA or AAA. Accessibility testing at the appropriate level is performed by automated systems and testers who check whether the content of the site meets the relevant criteria and assess the usability of the platform.

The Accessibility Conformance Testing (ACT) Rules Format 1.0 defines a format for writing accessibility test rules. These test rules can be used for developing automated testing tools and manual testing methodologies. It provides a common format that enables any party involved in accessibility testing to document and share their testing procedures in a robust and understandable manner. This enables transparency and harmonization of testing methods, including methods implemented by accessibility test tools 25.

Although Accessibility Conformance Testing (ACT) Rules Format 1.0 and WCAG 2.1 have been developed, there is a need to supplement, specify and summarize these rules in the form of a checklist. The following is a checklist for testing the availability of web applications. Note that this check list takes into account only cognitive impairments and is based on the WCAG 2.1 standard.

1. Layout testing:

– *Comparison with the layout*. Since the layout testing is based on a comparison with the layout, it is advisable to test the accessibility documentation first, and then test the layout itself. When it comes to layout testing, keep in mind that constant page navigation is very important for users with cognitive impairments, so it' a good practice to maintain website consistency and not change pages regularly. Adapting to a new layout is time consuming and can be difficult. Layout testing will save a significant portion of money for the customer and provide a higher quality software product.

– *Location and size of elements*. User interface components and navigation must be operable. Items should be arranged in the usual way without hiding menus and tips. Navigational mechanisms that are repeated on multiple Web pages within a set of Web pages occur in the same relative order each time they are repeated, unless a change is initiated by the user.

– *Matching fonts (name, size, color).* Line height (line spacing) to at least 1.5 times the font size; Spacing following paragraphs to at least 2 times the font size; Letter spacing (tracking) to at least 0.12 times the font size; Word spacing to at least 0.16 times the font size. The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following. Large Text: large-scale text and images of large-scale text have a contrast ratio of at least 3:1;Incidental: text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement. Logotypes: text that is part of a logo or brand name has no contrast requirement.

– *Content (check for spelling and grammar errors).* Content should consist of short and clear sentences, not contain slang and specific vocabulary, unless required by the subject of the site. The human language of each passage or phrase in the content can be programmatically determined except for proper names, technical terms, words of indeterminate language, and words or phrases that have become part of the vernacular of the immediately surrounding text.

– *Cursor appearance*. Changing the shape of the cursor depending on the element on which it is hoisted gives the user a hint about the function of the element.

– *Headings, title of the page*. Selected content elements facilitate the perception of the text.

– *Browser extensions* that may affect the appearance of the program (for example, AdBlock).

– *Check the content* when the images are disabled (WebDeveloper mode), flash, JavaScript.

– *No blinking*, which is also included in the video content. Web pages do not contain anything that flashes more than three times in any one second period. Motion animation triggered by interaction can be disabled, unless the animation is essential to the functionality or the information being conveyed.

– *Application volume*. If the application provides audio (advertising, audio assistant, video assistant, etc.), you must allow the user to adjust the volume or turn it off altogether.

2. Localization testing.

– Checking the test sample for correct translation.

– Adaptability of the application after translation into different languages.
– Abbreviations – There are rules that either translate, transliterate, or leave them as they are, but in any case they should be understandable to people with cognitive impairments.
– Check fonts after translation into other languages.
– Check the search in all locations. It is necessary to take into account the possibility of a mistake in the word.

3. Functional testing. To conduct functional testing for accessibility for people with cognitive impairments, it is advisable to consider all the elements of the application and determine the purpose and features of each (Table 2).

Table 2

Features of testing the functionality of application elements for accessibility

Element	Function	Features
Button	After pressing some action occurs	The location and signature of the button should not raise questions about its purpose. The button can be pressed from the keyboard and voice command. The Enter button should work as Submit.
Input field	To convey a certain action and interaction	The user should receive clear instructions on the information expected in the input field. If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content. Check for trimming spaces in input fields; whether all editing methods work (Insert, Delete, Backspace, Ctrl + C / V / X / Z, etc.
Search	To find relevant information by the user	Voice search engine control, automatic error correction.
Login-form	To obtain the appropriate access rights by the user	The user must have enough time to fill out the form. When an authenticated session expires, the user can continue the activity without loss of data after re-authenticating. For Web pages that require the user to submit information, at least one of the following is true: reversible – submissions are reversible; checked - data entered by the user is checked for input errors and the user is provided an opportunity to correct them; confirmed - a mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.
Calendar, date and time	To select a date	Ability to enter the date from the keyboard using the input mask. Alternatively, the user should be prompted to select a date from the drop-down list.
Error messages	To inform the user about the incorrect operation of the application, or improper handling of the application by the user	Error messages should be clear and local, and should include a prompt to correct the error.
Hint	Provide help to the user	In addition to the standard help, the user can choose a video help that will show the way to solve the problem.
Pop up	Direct the user to the desired script.	Research has shown that 73% of Internet users do not approve pop-ups: considering people with cognitive impairments, pop-ups should not be used, or the user should be able to adjust their absence.

4. Compatibility testing. In addition to the typical compatibility testing, it is necessary to test the application with assistive technologies. When it comes to people with cognitive impairments, the technology that will be quite relevant is the screen reader. It is the screen reader that converts text from the screen into an audio stream of synthesized speech or a set of sounds of different frequencies. In addition, the screen reader allows you to quickly display a list of items (titles, links, buttons, sections, etc.) and switch between them, which greatly facilitates the work of the web application not only visually impaired but also people with dyslexia and dementia.

Conclusion

Web-accessibility is a topical and broad topic that requires extensive and in-depth research. It should be noted that web-accessibility is accessibility not only for people with disabilities, because creating accessible web-applications makes interacting with them more convenient for all users.

Accessibility testing helps make web-applications accessible to a broader range of users. In order to fully perform accessibility testing, it is necessary to identify the target audience, determine how they use the application, identify potential problems that may arise. This article investigated the peculiarities of the perception of information by people with cognitive impairments. The authors elaborated the appropriate checklist for accessibility testing of web-applications.

References

1. Law of Ukraine “On Fundamentals of Social Protection of Persons with Disabilities in Ukraine” # 1664-IX from 15.07.2021. URL: https://zakon.rada.gov.ua/laws/show/875-12#Text (Accessed: 01.10.2021).
2. Convention of the Rights of Persons with Disabilities # 9833/0/14-16/19 from 06.07.2016. URL: https://zakon.rada.gov.ua/laws/show/995_g71#n40 (Accessed: 01.10.2021).
3. United Nations Convention on the Rights of Persons with Disabilities. URL: https://ec.europa.eu/social/main.jsp?catId=1138&langId=en (Accessed: 01.10.2021).
4. Union of Equality: Strategy for the Rights of Persons with Disabilities 2021-2030. URL: https://ec.europa.eu/social/main.jsp?catId=738&langId=en&pubId=8376&furtherPubs=yes (Accessed: 01.10.2021).

5. EU Web Accessibility Compliance and Legislation. URL: https://www.deque.com/blog/eu-web-accessibility-compliance-and-legislation/ (Accessed: 01.10.2021).
6. Web Accessibility. URL: https://digital-strategy.ec.europa.eu/en/policies/web-accessibility (Accessed: 01.10.2021).
7. D.L. Hosking Promoting accessibility for disabled people using EU standardisation policy. URL: https://www.researchgate.net/publication/317216744_Promoting_accessibility_for_disabled_people_using_EU_standardisation_policy (Accessed: 01.10.2021).
8. Irene Krebs, Habil. Arnim Nethe Accessibility in the Web for Disabled Peopl. URL: https://www.researchgate.net/publication/302469320_Accessibility_in_the_Web_for_Disabled_People (Accessed: 01.10.2021).
9. Neveen I. Ghali, Omar S. Solima, Nashwa El-Bendary, Tamer M. Nassef Virtual Reality Technology for Blind and Visual Impaired People: Reviews and Recent Advances Advances in Robotics and Virtual Reality (pp.363-385). URL: https://www.researchgate.net/publication/216729433_Virtual_Reality_Technology_for_Blind_and_Visual_Impaired_People_Reviews_and_Recent_Advances
10. Won-Kyung Lee, Eun-Gyoung Seo Website Design for Improving Web Accessibility of Disabled People. URL: https://www.researchgate.net/publication/276050041_Website_Design_for_Improving_Web_Accessibility_of_Disabled_People (Accessed: 01.10.2021).
11. D. D. Alves, M. I. Cagnin and D. M. Barroso Paiva, “Accessibility in development of free software projects”, *2014 XL Latin American Computing Conference (CLEI)*, 2014, pp. 1-12, doi: 10.1109/CLEI.2014.6965154.
12. R. Goncalves de Branco, M. I. Cagnin and D. M. Barroso Paiva, “AccTrace: Accessibility in Phases of Requirements Engineering, Design, and Coding Software”, *2014 14th International Conference on Computational Science and Its Applications*, 2014, pp. 225-228, doi: 10.1109/ICCSA.2014.51.
13. L. Fathauer and D. M. Rao, “Accessibility in an educational software system: Experiences and Design Tips”, 2019 IEEE Frontiers in Education Conference (FIE), 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028402.
14. K. Wille, R. R. Dumke and C. Wille, “Measuring the Accessability Based on Web Content Accessibility Guidelines”, *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, 2016, pp. 164-169, doi: 10.1109/IWSM-Mensura.2016.032.
15. N. Patil, D. Bhole and P. Shete, “Enhanced UI Automator Viewer with improved Android accessibility evaluation features”, *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 2016, pp. 977-983, doi: 10.1109/ICACDOT.2016.7877733.
16. E. Park *et al*., “Development of Automatic Evaluation Tool for Mobile Accessibility for Android Application”, *2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, 2019, pp. 1-6, doi: 10.1109/SysCoBIoTS48768.2019.9028034.
17. N. P. Kasaghatta Ramachandra and C. Csallner, “Poster: Testing Web-Based Applications with the Voice Controlled Accessibility and Testing Tool (VCAT)”, *2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion)*, 2018, pp. 208-209.
18. Directive (EU) 2016/2102 of the European Parliament and of the European Council of 26 October 2016 on the accessibility of the websites and mobile applications of public sector bodies. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016L2102&from=EN (Accessed: 01.10.2021).
19. Disability and health. URL: https://www.who.int/news-room/fact-sheets/detail/disability-and-health (Accessed:01.10.2021).
20. Disability and Health Overview. URL: https://www.cdc.gov/ncbddd/disabilityandhealth/disability.html (Accessed: 07.11.2021).
21. Law of Ukraine “On Higher Education” from 01.07.2014 # 1556-VI URL: https://ips.ligazakon.net/document/view/t141556?an=11&ed=2021_04_23 (Accessed: 01.10.2021).
22. Explanatory Note to the Draft Law of Ukraine “On Education of Persons with Disabilities (On Special Education)”. URL: https://ips.ligazakon.net/document/GH2MW00A?an=4 (Accessed: 01.10.2021).
23. Inclusive Design: Arguments, Practice, Testing and Implementing. URL: https://telegraf.design/inklyuzyvnyj-dyzajn-argumenty-praktyky-testuvannya-ta-vtilennya/ (Accessed: 01.10.2021).
24. International Classification of Diseases. URL: https://mkb-10.com/ (Accessed: 01.10.2021).
25. Web Content Accessibility Guidelines. URL: https://www.w3.org/ (Accessed: 01.10.2021).

Tetiana Zhyrova **Тетяна Жирова**	Ph.D, Associate Professor of Software Engineering and Cyber Security Department, Kyiv National University of Trade and Economics, Kyiv, Ukraine, e-mail: zhyrova@knute.edu.ua, https://orcid.org/0000-0001-8321-6939, Scopus Author ID: 57201748093, ResearcherID: AAX-3589-2021, https://scholar.google.com.ua/citations?user=xVt988IAAAAJ&hl=uk	кандидат педагогічних наук, доцент кафедри інженерії програмного забезпечення та кібербезпеки, Київський національний торговельно-економічний університет, Київ, Україна.
Nataliia Kotenko **Наталія Котенко**	Ph.D, Associate Professor of Software Engineering and Cyber Security Department, Kyiv National University of Trade and Economics, Kyiv, Ukraine, e-mail: kotenkono@knute.edu.ua https://orcid.org/0000-0002-2675-6514, Scopus Author ID: 57223192989, ResearcherID: A-9361-2019, https://scholar.google.com.ua/citations?user=gv7IvNwAAAAJ&hl=uk	кандидат педагогічних наук, доцент кафедри інженерії програмного забезпечення та кібербезпеки, Київський національний торговельно-економічний університет, Київ, Україна.
Volodymyr Tokar **Володимир Токар**	D.Sc., Professor of Software Engineering and Cyber Security Department, Kyiv National University of Trade and Economics, Kyiv, Ukraine, e-mail: v.tokar@knute.edu.ua https://orcid.org/0000-0002-1879-5855, Scopus Author ID: 56505221900, ResearcherID: K-2640-2018, https://scholar.google.com.ua/citations?user=ifVAcAsAAAAJ&hl=uk	доктор економічних наук, професор, професор кафедри інженерії програмного забезпечення та кібербезпеки, Київський національний торговельно-економічний університет, Київ, Україна.
Karyna Khorolska **Каріна Хорольська**	Assistant of Software Engineering and Cyber Security Department, Kyiv National University of Trade and Economics, Kyiv, Ukraine, e-mail: k.khorolska@knute.edu.ua	асистент кафедри інженерії програмного забезпечення та кібербезпеки, Київський

	https://orcid.org/0000-0003-3270-4494 Scopus Author ID: 57223197070, ResearcherID: https://publons.com/researcher/4706604/karyna-khorolska/, https://scholar.google.com/citations?user=48XjmrMAAAAJ&hl=ru&oi=ao	національний торговельно-економічний університет, Київ, Україна.
Bohdan Bebeshko **Богдан Бебешко**	Senior Lecturer of Software Engineering and Cyber Security Department, Kyiv National University of Trade and Economics, Kyiv, Ukraine, e-mail: b.bebeshko@knute.edu.ua https://orcid.org/0000-0001-6599-0808 , Scopus Author ID: 57223193357, ResearcherID: https://publons.com/researcher/4706616/bohdan-bebeshko/, https://scholar.google.com/citations?view_op=list_works&hl=uk&user=mnfTUc8AAAAJ	старший викладач кафедри інженерії програмного забезпечення та кібербезпеки, Київський національний торговельно-економічний університет, Київ, Україна.