DESIGNING INCLUSIVE WEB INTERFACES: A STANDARDS-COMPLIANT ACCESSIBILITY MENU WIDGET FOR ASSISTIVE USER EXPERIENCE IN FRONT-END APPLICATIONS

Plamenna Petrova
St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria
plamennavp@abv.bg

Juliana Dochkova-Todorova
St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria
doskova@ts.uni-vt.bg

ПРОЕКТИРАНЕ НА ПРИОБЩАВАЩИ УЕБ ИНТЕРФЕЙСИ: КОМПОНЕНТ МЕНЮ ЗА ДОСТЪПНОСТ, СЪВМЕСТИМ СЪС СТАНДАРТИТЕ ЗА ПОДПОМАГАЩО ПОТРЕБИТЕЛСКО ИЗЖИВЯВАНЕ ВЪВ ФРОНТЕНД ПРИЛОЖЕНИЯ

Abstract

This paper presents the design, stages of development, implementation, and evaluation of a standards-compliant accessibility menu widget, built as a reusable frontend component with the programming language TypeScript. The study addresses the growing requirements for inclusive web interfaces with the integration of customizable accessibility profiles and features in the widget as well as multilingual options, tailored to the needs of users, diagnosed with visual, cognitive, neurological, and reading impairments. The paper grounds its methodology in recommended WCAG practices for digital accessibility. It includes a comparative analysis of existing software solutions for accessible menus and text-to-speech web technologies. Particular emphasis is placed on the design rationale behind profile-specific configurations and the portability of the widget. The findings of this work aim at contributing to the broader field of human-centered interface engineering by offering a technically robust component for improving the user experience in terms of accessibility.

Keywords: *Digital Accessibility; Inclusive Systems Design; Accessibility Menu Widget; Frontend Development; WCAG; Multilingualism; User Experience (UX); Usability.*

1. INTRODUCTION

In today’s world, where the digitalization is increasing at a fast pace, the access to information and services has become a fundamental right with the widespread adoption of the internet leading to the radical transformation of the means of communication, work, and learning and thus making the digital interfaces a critical point of access for everyone. Hence, the digital accessibility is an entirely legal and ethical obligation, which has to guarantee that all individuals, including those who fall under the group of users with special needs, can perceive, understand, navigate, and interact with web interfaces [1] and therefore should not be regarded as merely an optional feature or addition for the web design. In this connection, the adherence to established and widely used standards such as the Web Content Accessibility Guidelines (WCAG) must be set as a main priority for the prototyping, creation, and testing of digital applications or ecosystems which strive for inclusivity and for providing equivalent and non-discriminatory user experiences.

Despite the existence of common accessibility standards, the design of many front-end applications still presents significant barriers for the interaction of users who have visual, cognitive, neurological, and reading impairments, because they frequently face challenges with the readability of the site content, the navigation between the inner sections of a web page or between separate web pages, and the triggering of elements with transitions or animations [2], [3]. While some of the assistive technologies like screen readers offer a baseline level of support for such cases, they are not always sufficient for the customization of the user experience of a greater spectrum [4], [5]. Furthermore, many of the existing accessibility solutions are either not constructed as portable, reusable components or generally lack a standards-compliant foundation, which creates a persistent accessibility gap, where the onus is often on the users to make adaptations to the interface, rather than the interface adapting and adjusting itself to match to their unique needs [6], [7], [8].

The paper addresses this challenge by detailing the design, development, and evaluation of an accessibility menu widget, which is built as a reusable front-end component using the programming language TypeScript for applications, leveraging the frameworks React and Next.js, and by following WCAG accessibility guidelines. The widget aims at being a portable solution for improving the user experience in front-end interfaces and its core functionality is centered on a customizable accessibility menu, consisting of settings for specific disability profiles as well as multilingual options. The widget's design rationale and implementation are grounded in WCAG practices, where the accessibility profiles are tailored for the assistance of users with various forms of disabilities and their features range from adjustable font sizes, color contrast and sections focus or highlight modes, text-to-speech synthesis, and dyslexia-friendly typography.

The rest of the paper is structured as follows: Section 2 delves into the underpinnings of inclusive and accessible systems design with an exploration of a literature review of principes for digital accessibility, usability, the WCAG standards, and an overview of documented solutions in this field. Section 3 describes the research methodology and the stages of the development process of the accessibility menu widget. Section 4 contains a comparative analysis of public software solutions for accessibility menus and text-to-speech web technologies. Section 5 exemplifies the design decisions and the implemented functionalities of the widget. Section 6 discusses the findings and prospects of the study, and Section 7 concludes the paper with a summary of the yielded research insights and suggestions for future work.

2. Related Work

Accessibility has become a central topic of discussion, since it affects the quality of life in society, which means that a product or service is designed or modified so that it can be used by everyone regardless of their ability or disability without being imposed to barriers or impediments. In the pursuit of a society with inclusivity and equality traits, it is essential that all individuals are afforded the same opportunities to be active participants in it and by ensuring that every citizen can access products and services in a physical and digital way the social inclusion and equal rights get promoted. The most common restrictions that result in exclusion apply to visual dysfunctions, hearing impairments, mobility limitations, intellectual and transitional disabilities, and those that appear and unfortunately progress with ageing [9].

Digital accessibility plays an integral role in modern times, primarily due to the significant portion of the population living with one or multiple disabilities [10]. In light of this, official statistical data from the World Health Organization (WHO) show that 16% of the global population, which amounts to ~1.3 billion people, have a diagnosis of a disability that profoundly impacts them [11]. To prevent people with disabilities being excluded from participating fully in the digital society, the need for accessible digital content and services arises, as highlighted by the statistics [12].

The exact definition of digital accessibility is given by the Web Accessibility Initiative (WAI), which implies that individuals with special needs should be able to achieve the actions of access, navigation, perception, and interaction with content [1]. It also refers to the good practices of making the design of digital systems and services, among which are websites, software, and other web-based products accessible for people with visual, auditory, motor, or cognitive impairments without encountering obstacles, discrimination, or diminishing their degree of independence [10].

For these reasons, digital accessibility is not only a social responsibility, but also crucial for legal compliance and international organizations policies, which are shaped by the ongoing processes of the globalization [13], [14]. Legislation is indeed the instrument that specifies the regulations, sets the norm, and enables their practical implementation. The international accessibility community unites around documents, which are recognized worldwide, such as the Convention on the Rights of Persons with Disabilities, the Marrakesh Treaty, and two directives of the European Union (EU) - The Web Accessibility Directive (2016/2102) and The European Accessibility Act (2019/882) [14].

Usability and accessibility are the acknowledged terms in the context of enhancing the user experience on the World Wide Web. While there is no universally accepted view or definition for usability, as per the standard ISO 9241-11 a product is considered usable if the end-users can accomplish tasks effectively, efficiently, and satisfyingly [15], [16]. The usability of websites is strongly influenced by the extent of their accessibility, which is explained by IS0 9241-171 as: "the usability of a product, service, environment or facility by people with the widest range of capabilities" [15], [17]. It goes beyond simply usability for people with certified disabilities by seeking to attain the highest threshold of effectiveness, efficiency, and satisfaction for people with the full scope of capabilities in different scenarios [17].

To bolster the accessibility of web content and services, the World Wide Web Consortium (W3C) introduced the Web Content Accessibility Guidelines (WCAG) [18], which are continuously being developed and WCAG 2.2 is their latest version [19]. The guidelines are organized into four main principles often referred to by the acronym POUR: Perceivable - the perception of web content by users should happen with senses for seeing, hearing or touching, Operable – web content should be functional with input devices like a keyboard and mouse or other alternatives, Understandable – a multitude of users can understand the content which also must be presented in a consistent, predictable, and readable form, and Robust – web content should be compatible with various user agents, browsers, devices, and assistive technologies [20]. WCAG has three levels of conformance: A, AA, and AAA, where the former two are targeted for the majority of digital platforms, while Level AAA represents the highest possible standard of accessibility [9], [20].

The project Digital Accessibility for People with Special Needs: Methodology, Conceptual Models and Innovative EcoSystems [21], conducts fundamental research in the area of information and communication technologies (ICT) by concentrating on the availability and improvement of digital accessibility of diverse objects and knowledge for people, diagnosed with visual and cognitive impairments. The research's tasks emphasize on the: 1) development of new methods and models for evaluating the digital accessibility of objects like: documents, websites, software applications, and serious games, 2) selection, systematization, and analysis of existing technologies for digitally accessible solutions, 3) construction of optimal and applicable conceptual models for digital accessibility, 4) sharing of web content in socially justified and accessible ecosystems, 5) publication of a collection of digital resources for educational purposes and dissemination of the results for specialized robotic training for individuals with disabilities, and 6) introduction of accessible repositories and software environments with open access for the knowledge acquisition of disabled people. The researchers working on this project have published several consecutive and interconnected papers, covering the listed tasks [22], [23], [24], [25], [26], [27], [28].

3. Methodology

This study employs an engineering-focused methodology based on the ADDIE model (Analysis, Design, Development, Implementation, Evaluation), which is an extensively utilized instructional design framework in the educational sphere [29], first conceptualized and implemented at the Florida State University [30]. Instructional Design technology is the use of technological tools to design, develop, and deliver learning materials and experiences [31]. The application of this approach to the current research project is linked to the functioning of the accessibility menu widget, which in essence, represented a probe assistive tool in its early versions, when teaching bachelor's degree students content management systems-oriented university disciplines, rooted in STEM/STEAM principles in a blended learning environment.

In [32] the assertion is made that the design process of the ADDIE model, whose graphical illustration is shown on Fig. 1, is both sequential and iterative. On the one hand, the sequential aspect of the assertion is clarified by the fact that the starting stage of the model is analysis, which is usually carried out through needs analysis, succeeded by obtaining data to serve as input for designing a particular program or functionality. The next stages encompass the development and implementation of the program or functionality, which are then concluded with their assessment in the final stage. On the other hand, the iterative aspect of the assertion is articulated with the explanation that the developers can go back and forth between the stages because of the mutual reliance of the five components.

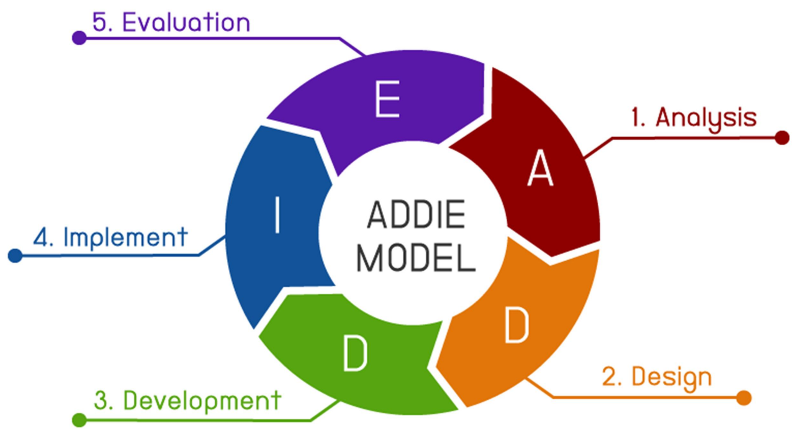

Fig. 1. The ADDIE Model

The five phases of the ADDIE framework relate to the repeated evolutionary lifecycle of the accessibility menu widget and its technical evaluation, as detailed in the following points:

1. Analysis – This initial phase involved the determination of the needs of the study as well as the functional requirements of the accessibility widget, where the foundational research was conducted through a review of topical academic and technical literature. The WCAG guidelines were consulted to discern the recommendations for accessible web content. A comparative analysis of existing and publicly available accessibility menus and text-to-speech web technologies was also undertaken, which was pivotal for the identification of the limitations surrounding the current solutions and for

outlining the specific problems that the constructed widget intends to solve by paying close attention to the clear understanding of the conditions of the disabled users.

2. Design - based on the analysis phase the main goals and architectural dependencies of the solution were determined. The design objectives revolved around creating a modular and reusable frontend component for personalized accessibility settings like:
    - color blindness headings and links marking options
    - text-to-speech conversion support with voice and rates configuration
    - a dictionary with a search by keyword term and display of explanatory results
    - reading masks, sitemaps, visual themes modes for light, dark, smart, high, color inversion and saturation contrast, animations stopping and images hiding
    - font size, line height, and letter spacing adjustments, buttons enlargement
    - dyslexia and readable font, text magnifier as well as reading guides
    - change of background, text, titles, and links color from a color palette
    - text alignment – in left, center, right, and justified positions
    - titles, links, focus and hover highlight
    - big cursors in white and black hues
    - content filtering for headings, tag landmarks and links with redirection to their source location

The key decisions for the design of the widget were the integration of distinct accessibility profiles for blind, elderly, motor impaired, visually impaired, color blind, dyslexia, cognitive and learning, seizure and epileptic and ADHD (Attention-Deficit/Hyperactivity Disorder) as well as multiple languages selection. In this phase the conceptual blueprint for the interface of the accessibility menu was also produced, which prioritized the tailoring of the user experience over one-size-fits-all features.

3. Development - the development phase centered on the implementation of the conceptualized blueprint from the design stage, which led to the building of the widget as a frontend component with reusability characteristics using the TypeScript programming language in a Next.js progressive web application. A substantial period was dedicated to the speech synthesis configurational logic. The choice of the codebase and environment was driven by type-safety, versatility, and ease of maintenance benefits. Moreover, during the construction of the widget the WCAG 2.2 Level AA guidelines were strictly adhered to from its inception.
4. Implementation - the implementation phase dealt with making the final developed widget available for use. By pertaining to the practices of instructional design a field trial with student participants with visual impairments was performed apart from the realization of the widget in a technical context. The focus of this stage was put on achieving portability for the solution and its embedding into React web environments like the environment of the progressive web application, as the framework Next.js directly originates from the React library.
5. Evaluation – in this ending phase the functionalities of the engineered solution were evaluated against the comparative analysis of the existing accessibility menu widgets. Additionally, the custom accessibility profiles and the multilingual options were assessed to test whether they address the identified requirements in the initial phase. The findings from this evaluation were then discussed regarding the place and applicability of the widget in the landscape of similar current technologies.

4. Comparative Analysis of Existing Software Solutions for Accessibility Menus and Text-to-speech Conversion

Accessibility Menu Widgets

The digital accessibility software industry is defined by an array of solutions aimed at assisting websites in complying with international standards and legal requirements for accessible tools and provisions, most notably the WCAG recommendations. These solutions typically manifest as a customizable widget placed on a website, offering conditional adjustments on demand for visitors, which especially individuals with visual, auditory, cognitive, and motor disabilities can use to their advantage. A prominent offering is a screen reader option with speech synthesis, allowing users who experience visual or learning difficulties to have on-screen text read aloud by a synthetic narrator voice so that the content consumption could be facilitated for them. The analysis examines sets of distinct features belonging to three of the leading solutions for portable accessibility menu widgets: Skynet Technologies, Equal Web, and UserWay. While each platform has a varying degree of automation, their chief strength lies in the accessibility functionalities they provide. Table 1 below presents a direct comparison of the mentioned platforms on the criteria key features, distinctions, drawbacks, and compliance.

Table 1. Comparison of leading solutions for accessibility menu widgets

Widget Criteria	Skynet Technologies	Equal Web	UserWay
Key features	separate accessibility profiles, contrast themes, screen reader, voice navigation, font scaling and text alignment, background and text color changes, big cursors, hover and focus highlight, content filtering, language selection, dictionary, stop animations action, oversize widget toggle button	separate accessibility profiles with subgroups, screen reader adjustment for blind users, text reader, keyboard navigation and smart navigation by numeric keys, voice commands, color themes, custom color settings, font sizing, big cursors, readable font and mode, links and headings highlight, dictionary, buttons enlargement	separate accessibility profiles, contrast themes, screen reader, font size and spacing controls, tooltips, saturation, dyslexia friendly font, page structure, cursor and reading masks, images hiding, links highlighting, voice navigations, animations pausing, oversize widget, language selection

Widget Criteria	Skynet Technologies	Equal Web	UserWay
Distinctions	profiles for seizures, epilepsy, ADHD, elderly, learning, blind, visually impaired, dyslexia, motor impaired, screen reader with keyboard navigation, color blindness options for each type, primary colors dropdowns, widget placement in eight directions, fast terms finding in the dictionary, easy reset to initial state, images hiding, readable and dyslexia font, accessibility statement, content filtering in a dialog for headings, landmarks and links, sounds muting, languages search and filtering in a dialog, clicking on buttons with sounds	profiles for epilepsy, ADHD, elderly, learning, blind, visually impaired, dyslexia, motor impaired, accessibility features for each profile subgroup, color bar for change of the site colors, blinks and flashes blocking, images descriptions, buttons sizes, screen zoom, captions to audio and video content, AI-powered dictionary, accessibility mode, boundaries for spacing and sizing controls, time period for hiding the accessibility menu, AI widget assistant with embedded chat, configurable accessibility options	profiles for motor impaired, blind, color blind, dyslexia, low vision, cognitive and learning, seizures and epileptic, ADHD, screen reader with speed rate cycle, contrast, spacing, text size and alignment, saturation, dyslexia font, cursor size with several adjustments which can be changed on click, page structure in a dialog for headings, landmarks and links, widget horizontal moving with the cursor, filtering for language selection, the oversize mode increases the content size for the widget configurations, streamlined reset of settings
Drawbacks	hard to use voice navigation and talk to type settings, virtual keyboard not operable for users without disabilities, hiding the widget results in entirely disabling it, read more cannot be closed with back navigation, the text magnifier opens large text sections, no lower and upper boundaries for scaling, using a lot of the settings reverts the interface	accessibility profiles with accordion style and on-off toggle buttons, expanded sections, small dropdown menu for language selection, no specific color blindness settings except for the contrast background themes, dictionary with descriptions of words by mouse and no keyboard input, hard widget button moving with drag and drop, no reset action, small links for turning off the functionalities and the accessibility statement	language selection in the same region as the widget, two scrollers for the widget itself and the list of languages, color blind profile linked only to the smart contrast theme, oversized widget toggle button at the center part of the region, only two or four rates control cycle buttons for the most of the accessibility options, widget moving radio buttons in left or right directions, widget dragging only by clicking on the top ribbon, accessibility statement as a small bottom located link
Compliance	WCAG 2.0, 2.1, 2.2, browsers support by default, screen readers compatibility by default, not included in the main documentation	WCAG 2.2 and support for all browsers and screen readers (JAWS, NVDA, VoiceOver)	WCAG 2.1 & 2.2 conformance, browsers support by default, screen readers compatibility by default, not included in the main documentation

The comparative analysis of the three solutions for accessibility menu widgets reveals differences in their strengths and design approaches. The component of Skynet Technologies excels in high level of personalization and profile configurations, giving precedence to the freedom of user control. Its main advantage stems from the granularity of its settings and the ability to apply them quickly through tailored profiles. In contrast, Equal Web distinguishes itself with its integration of AI-powered functionalities with intelligent assistance as well as more ordered and adjustable, though at times cumbersome, user interface. UserWay, on the other hand, advocates for features simplicity and a streamlined user experience. While it may have fewer granular controls compared to its counterparts, its straightforward design also makes it highly accessible and user-friendly. All of the solutions support the most recent version of WCAG, where only in Equal Web's documentation the compatibility with screen readers technologies is explicitly stated, even if it is understood by default.

Text-to-speech Conversion Web Technologies

The implementation of text-to-speech conversion into web applications is important for enhancing the accessibility and the user experience and hence developers have to decide whether to utilize a native browser-based solution or to opt for a more robust, cloud-based service. The Web Speech API's SpeechSynthesis documented by Mozilla Developer Network (MDN) is a standardized browser interface which operates without external dependencies, whereas Google Cloud Text-to-Speech (TTS) and Azure AI Speech are commonly recognized scalable, cloud-hosted platforms. Table 2 compares these three technologies by outlining their respective strengths and weaknesses, considering technical and practical criteria.

Table 2. Comparative overview of popular text-to-speech conversion web technologies

Technology Feature	Web Speech API	Google Cloud TTS	Azure AI Speech
API Type	browser native	cloud-based (Rest API, SDKs)	cloud-based (Rest API, SDKs)
Voice Quality	variable, dependent on operating system	high-fidelity	high-fidelity
Voice Selection	dependent on operating system	extensive (hundreds of voices)	very extensive (hundreds of voices)
Customization	basic (pitch, rate)	advanced for pitch, rate, emphasis	highly advanced (speaking styles, custom voices)
Cost	free	pay-as-you-go per character	pay-as-you-go per character
Connectivity	Offline	Online only	Online only

The comparative overview of the text-to-speech conversion web technologies in the above table unveils clear distinctions in their design philosophies and the intended use cases. The Web Speech API is a pragmatic, cost-free alternative for basic, client-side applications where the ease of work and the offline access are paramount. Its main benefit is the lack of requirements for external fees or internet connectivity, but this comes at the expense of voice quality and customization, limited by the operating system of the user. Conversely, the cloud-based services Google Cloud TTS and Azure AI Speech create opportunities for solutions with more power and flexibility, which can be strengthened by superior voice quality, extensive voice libraries, and advanced adjustments, although they always need a stable internet connection and usage fee must be paid for them periodically.

5. Design and Functionalities of the Developed Accessibility Menu Widget

The developed accessibility menu widget is designed for optimal visibility and discoverability with its drawer opening button placed at the bottom right corner of the web pages, which when clicked slides the sheet from right to left accompanied by a light dimming animation. The accessibility menu button is illustrated with an accessibility icon for better guidance. The region of the menu encompasses easily navigable and organized subsections for accordions and a grid structure, the latter often consisting of elements in two columns which play the role of buttons:

- A dedicated top region with a closing button for the drawer with instructions for a keyboard combination helping manual opening/closing of the menu which is positioned on the left as well as a selected language abbreviation button that triggers a dialog for choosing the working language of the interface on the right. The left and right actions are justified to the opposite ends of the region.
- The dialog for the language selection, displayed in Fig.2, is oriented on the center part of the screen and is comprised of a search input for filtering language results on keypress which are structured in a scrollable grid with two columns. Each column is a button with an icon for the language abbreviation and the full name of it in the target language and in English. The choice of a language immediately translates the interface and closes the dialog with no intrusive animations.

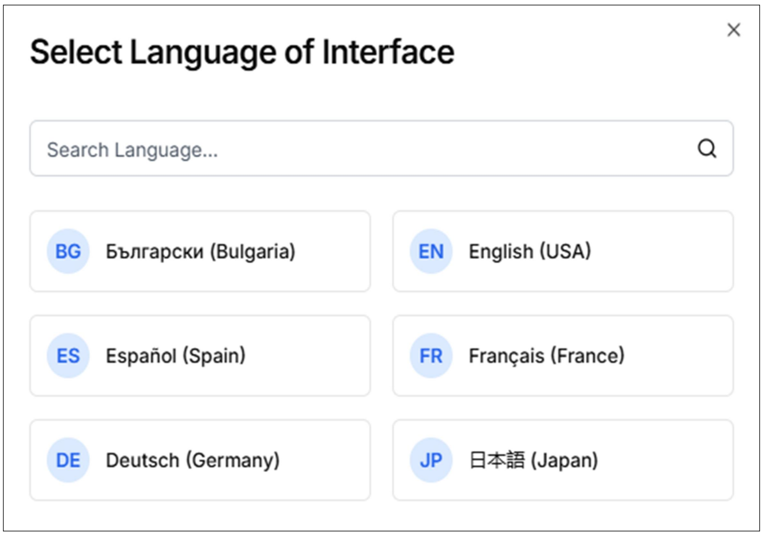

Fig. 2. The interface language selection dialog

- An accessibility icon also indicates the accordion item for the accessibility profiles, shown in Fig. 3, which are located underneath the top section. When the accordion is in open state it uncovers the profiles in the order of: Blind, Elderly, Motor Impaired, Visually Impaired, Color Blind, Dyslexia, Cognitive and Learning, Seizure and Epileptic, and ADHD:
  - The blind profile activates the screen reader which uses a text-to-speech engine to elucidate that the feature can be used either as a text reader for reading text and descriptions on mouse hover or for a keyboard navigation with strict command-key mappings or hotkeys. These correspond to explicit actions like to navigate focusable elements, to read text lines or to start, pause and resume reading.
  - The elderly profile applies a big reading guide line.
  - The motor impaired profile turns on the settings for stopping the animations and magnifies the text.
  - The visually impaired profile initiates the options for high saturation theme, readable font, a big white cursor, text magnifying tooltips and suspends the animated visual elements.

  - The dyslexia profile replaces the current font with the dyslexia friendly and open-source font OpenDyslexic.
  - The cognitive and learning profile puts on a bright reading mask, dims the background and stops the animations.
  - The ADHD profile is associated with changing the color theme to low saturation.
  - The seizure and epileptic profile halts the animations and makes the interface lowly saturated.
- Beneath the accessibility profiles accordion item is the location of the accessibility settings which are either clickable buttons or contain combo boxes, color pickers or other interactable elements like chevron icons. In more detail the two-column accessibility settings grid, viewable in Fig. 4, is reserved for the functionalities described per rows:
  - At the starting row the color blindness option has a searchable combo box for all subtypes of the condition: protanomaly (reduced red sensitivity), deuteranomaly (reduced green sensitivity), tritanomaly (reduced blue sensitivity), protanopia (no red cones), deuteranopia (no green cones), tritanopia (no blue cones), achromatomaly (incomplete color blindness) and achromatopsia (complete color blindness). Next to it is the screen reader linked to the blind profile.
  - The second row is allocated for the dictionary and the reading mask. The dictionary button opens a dialog for a search box that fetches results from a public API. The results get refreshed on key down and list matching parts of the speech, i.e. nouns, verbs, adjectives and adverbs, sorted by the relevance of their meaning.
  - The third and fourth rows include the color theme changing buttons for inverting colors, light contrast, dark contrast, and high contrast.
  - In the fifth row the options for smart contrast in support of visually impaired individuals and the font size adjustment controls can be found.
  - The sixth row has similar adjustment controls for line height and letter spacing, while the seventh row shares the content scaling option and the dyslexia font button.
  - The eighth row has the readable font and stop animations buttons.
  - The ninth row is for the reading guide, and the background color picker and the tenth - for the text and title color pickers.
  - The place of the text alignment options is on the eleventh and twelfth rows – the text can be aligned to the right, center, left or justified.
  - The thirteenth row has the buttons for the big white and black cursors, while on the fourteenth row the hide images and text magnifier settings are located.
  - The highlight links and titles buttons are positioned on the fifteenth row and the highlight hover and focus ones - on the sixteenth.
  - The monochrome, low saturation, medium saturation and high saturation color themes buttons are all situated in the seventeenth and eighteenth rows, respectively.
  - The nineteenth row is for the options for text muting and adding alternative images descriptions.
  - The last twentieth row is occupied by the settings for buttons enlargement and content filtering. The content filtering button opens a dialog with tabs for all headings in a web page, landmarks for HTML

tags such as header, nav, form, aside, section, etc., and links pointing to concrete locations.

Special attention is merited to the text-to-speech conversion logic which allows the configuration of voices and rates from the Web Speech API's SpeechSynthesis. The preset text reader on mouse hover can be switched to the function for key mappings navigation. Those options are available by toggling a separate dialog under the accessibility settings grid. The dual text-to-speech setup is dictated by the different possibilities for user preferences, since blind people generally work with keyboards to navigate through websites, whereas individuals with low to medium visual impairments favor the auditory experience of speech synthesis on cursor dwell.

The sheet of the accessibility menu can be moved to top, right, bottom, and left directions, where the placement options can be accessed from an expandable accordion item after the text-to-speech settings dialog button. The end of the widget region is marked by an oversize widget toggle control, and large buttons for: an accessibility statement with reference to the official page of the WCAG standards, resetting the already applied accessibility settings, and hiding the widget for a period selected from a time picker.

The activation of the accessibility profiles happens singularly, whereas multiple accessibility settings can be triggered simultaneously. The users are advised against enabling a lot of options at once, as the interface can get overwhelmed by the diverse functions. The font size, line height, letter spacing, and content scaling options have a default rate of 100%, lowest boundary of 20% and highest boundary of 200%, thus balancing the adjustments limits. The enabled accessibility options get saved on every content rendering, page refresh and website reinitialization. The reset action is useful when too many activated settings deteriorate the user experience or when fresh configurations are necessary.

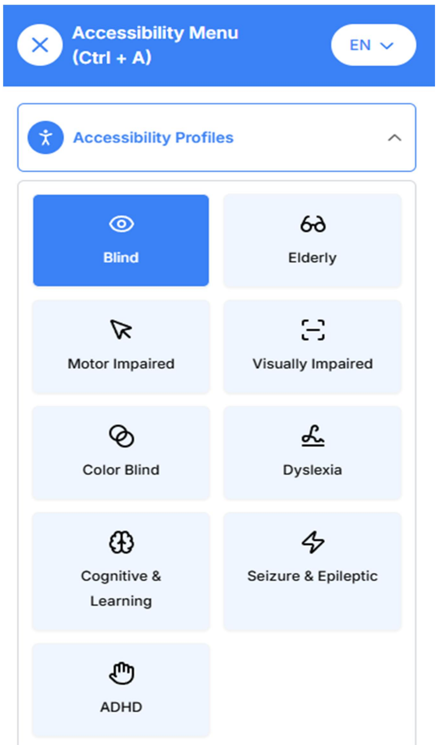

Fig. 3. The accessibility profiles and settings grids – 1

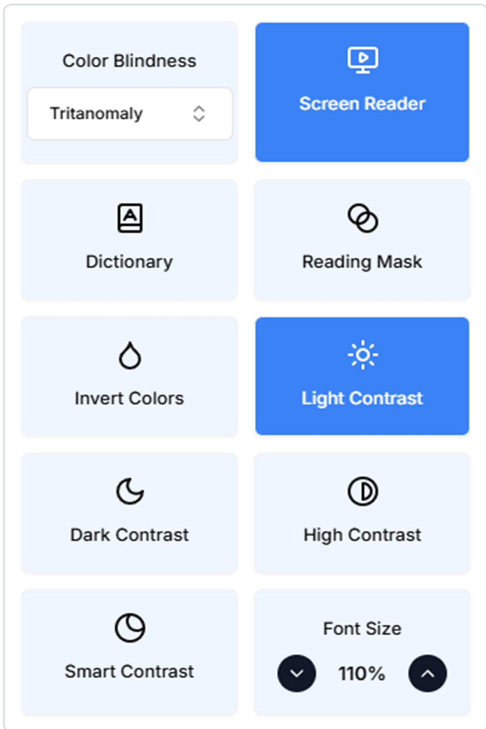

Fig. 4. The accessibility profiles and settings grids – 2

6. RESULTS AND DISTUSION

The development of the accessibility menu widget can be perceived as a direct application of the principles of inclusive systems design and human-centered interface engineering. In these fields a paradigm shift from designing for the impairment to designing for the person is advocated, and ultimately, for everyone, which takes the spirit of accessible design a step further in user interfaces, thereby transcending the straightforward compliance with accessibility standards [33]. Inclusive design is a fundamental strategy to create better products for all users rather than invaluable add-ons, promoting social solidarity, equality, and justice [34]. The idea of one product meeting different needs [35] is embodied in the configurable profiles and settings of the widget, which serve as a concrete example of this design philosophy, where the identified considerations and the potential limitations of users with various impairments were the driving force behind the design, functionalities, implementation and evaluation of the component. The final results conclude the evaluation stage within the iterative ADDIE model, validating the design choices made to overcome the shortcomings of existing solutions. The constructed widget aspires to meet as many as WCAG 2.2 Level AA recommendations as possible with all implemented accessibility features.

The comparative analysis for leading accessibility menu widgets solutions provided critical benchmarks for the design of the current widget. While each of the analyzed solutions had noteworthy strengths, key drawbacks were also observed. In this context the engineered component outweighs and combines the best features of its counterparts by making efforts to resolve the pinpointed weaknesses with uncomplicated reset behavior to revert registered configurations, no lagging transitions, well-defined adjustments boundaries, granular and easy to enable controls, simpler settings structure, and a good variety of color blindness, colors and periods helper interface elements. The one-click buttons for the accessibility profiles and options are akin to the approach of Skynet Technologies, drawing inspiration from the simplistic design style of its widget. The language selection dialog is both searchable and immediately sets changes, avoiding clunky dropdowns or separate regions found in other solutions. The quad-directional drawer placement makes the interface more controllable without encumbering users with a small area for drag-and-drop.

A main strategic decision was the choice of a text-to-speech conversion web technology. While the voice quality in cloud-based services is vastly superior and their customizations are more advanced, they require continuous internet connectivity and incur usage costs. The Web Speech API's Speech Synthesis was picked for the design and implementation purpose of the portable and reusable frontend component, namely for the created widget to rely on minimal external dependencies. The free distribution of the Web Speech API is traded-off in voice quality, but this downside is mitigated by the dual-functionality for both on-hover text reading and keyboard navigation with hotkeys, which caters to differentiated visual impairments and user preferences.

The field trial with student participants affirmed the usability of the widget as a teaching and assistive tool and at the same time the architectural solution, whose programmatic logic is built with TypeScript, contributes to the long-term viability of the component with type-safety and reduced maintenance overhead. The solution was subsequently refactored into a library, designed and composed to be easily integrated into React and Next.js projects. The library was primarily assessed for web, desktop and mobile operating systems featuring a Next.js progressive web application which is generally used as an educational platform for a blended learning environment by the students as well as in smaller bespoke React applications. Given that the library is completed, the portability of the widget can be accomplished and with this the embedding and testing of the component by a community of web developers is sought.

Despite the successful validation of the realized widget, it is important to acknowledge the limitations of the present study. The field trial with student participants, among whom are individuals with visual impairments attests the usability of the solution in an informal circle of testers with gathering of oral feedback. However, it does not involve a formal evaluation with a diverse group of users, predominantly consisting of disabled people having alike or similar conditions to all of the accessibility profiles from the menu. Additionally, even though the component is designed to be standards-compliant, it has not yet undergone an expert led accessibility audit. These constraints, along with the inherent trade-offs for voices and rates in using the Web Speech API, necessitate further and public testing to fully assess whether the applicability of the widget is long-lasting or new design plans must be arranged and constructible measures must be taken to optimize its functioning and portability.

7. Conclusion

The conducted research focused on the repeated stages for analysis, design, development, implementation, and evaluation of the target accessibility menu widget, strictly following the ADDIE model and incorporating the principles of the broader fields of inclusive systems design and human-centered engineering. The widget encourages a "design for all" philosophy through its configurable accessibility profiles and granular settings. Its architecture is crafted to make the component visible, discoverable and easy to use, with its features for expandable accordions, clickable buttons, filterable language selection and customizable inner interface elements constituting a structure for fast navigation and actions organization. The great diversity of profile-specific or additional adjustable functionalities accommodates a wide range of impairments from visual to motor to cognitive and learning and seizure and epileptic, demonstrating the potential of the component as a versatile assistive tool.

The design choices of the work were informed by a comparative analysis of leading accessibility solutions. The developed widget strategically integrates the good offerings of its counterparts such as the one-click profile enabling, color blindness types, multiple simultaneously activated options, unobtrusive reset behavior, while sorting out common weaknesses like lagging transitions and overly grouped and complicated settings. The deliberate decision to utilize the Web Speech API sets aspiration for the component remaining portable and reusable with a minimum of extrinsic dependencies. The preliminary evaluation with the field trial involving student participants validated the widget's usability as a teaching instrument and its TypeScript architecture in the composed library is highly suitable for React and Next.js projects.

Looking forward, future work on the study will prioritize the alleviation of the current limitations of the widget as well as an expansion of its capabilities. A primary objective is a formal evaluation with a diverse group of users, more specifically individuals with disabilities, and the undergoing of an audit by accessibility experts to officially test the compliance with the WCAG 2.2 Level AA guidelines. In parallel, new features will be developed such as talk and type for hands-free text input, a virtual keyboard, and advanced voice navigation. Furthermore, it is envisioned to leverage artificial intelligence for dynamic content summarization, predictive text assistance, a chat for dictionary terms and overall orientation help in the websites. To turn the component into an open-source solution, it will be published as a public npm library, which is expected to optimize its functionality and solidify its applicability as a tool that improves the assistive user experience.

Acknowledgements

This work was partially supported by St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria under Project No. FSD-31-328-18/23.04.2025, 2025, "Methodology for Modeling and Developing Information Systems with Artificial Intelligence in the Education Sector".

References

1. WAI, W. W. A. I. (2022). Introduction to Web Accessibility. Web Accessibility Initiative (WAI). Available at: https://www.w3.org/WAI/fundamentals/accessibility-intro/ (last view: 25-08-2025)
2. Devarapalli, C. A. (2022). Accessibility in User Interfaces: Confronting Common Challenges. Journal of Artificial Intelligence and Computer Communication, 1(1), 242. DOI: https://doi.org/10.47363/jaicc/2022(1)242
3. Literary Vieritz, H., Schilberg, D., & Jeschke, S. (2013). User Interface Modelling for Accessible Web Applications with the Unified Modelling Language. In Automation, Communication and Cybernetics in Science and Engineering 2011/2012 (pp. 759–768). Springer. DOI: https://doi.org/10.1007/978-3-642-33389-7_70
4. Antonelli, H. L., & Fortes, R. P. M. (2015). A Support for Developers Implement the Accessibility Guidelines Regarding to Web Menus. Proceedings of the Brazilian Symposium on Multimedia and the Web, 21–24. DOI: https://doi.org/10.1145/2820426.2820445
5. Elavsky, F., Nadolskis, L., & Moritz, D. (2023). Data Navigator: An accessibility-centered data navigation toolkit. IEEE Transactions on Visualization and Computer Graphics, 30(1), 1–11. DOI: https://doi.org/10.1109/TVCG.2023.3327393
6. Martín, A. E., Rossi, G., Cechich, A., & Gordillo, S. (2010). Engineering Accessible Web Applications. An Aspect-Oriented Approach. World Wide Web, 13(4), 419–440. DOI: https://doi.org/10.1007/S11280-010-0091-3
7. Diniz, S., & Gama, K. (2024). Design Systems and Component Packages as an Interface for Accessibility. Proceedings of the Brazilian Symposium on Software Engineering, 3566. DOI: https://doi.org/10.5753/sbes.2024.3566
8. Han, M.-R., & Park, P. (2009). A study of interface design for widgets in web services through usability evaluation. Proceedings of the International Conference on Information Systems, 1656109. DOI: https://doi.org/10.1145/1655925.1656109
9. Krok, E. (2024). Digital Accessibility as an Essential Element of an Inclusive Organization. European Research Studies Journal, 27, 857–876. DOI: https://doi.org/10.35808/ersj/3752
10. Chemnad, K., & Othman, A. (2024). Digital accessibility in the era of artificial intelligence—Bibliometric analysis and systematic review. Frontiers in Artificial Intelligence, 7. DOI: https://doi.org/10.3389/frai.2024.1349668
11. World Health Organization. (2022). Global Report on Health Equity for Persons With Disabilities.
12. Dobransky, K., & Hargittai, E. (2016). Unrealized potential: Exploring the digital disability divide. Poetics, 58, 18–28. DOI: https://doi.org/10.1016/j.poetic.2016.08.003
13. Csontos, B., & Heckl, I. (2025). Five years of changes in the accessibility, usability, and security of Hungarian government websites. Universal Access in the Information Society, 1-25. DOI: https://doi.org/10.1007/s10209-025-01223-5
14. Sabev, N. (2025). Fundamental Legislation on Digital Accessibility Issues. In M. E. Auer & T. Rüütmann (Eds.), Futureproofing Engineering Education for Global Responsibility. ICL 2024. Lecture Notes in Networks and Systems, vol 1281 (pp. 317–328). Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-83520-9_27
15. ISO/TC 159/SC 4 Ergonomics of human-system interaction (Subcommittee). (1998). Ergonomic requirements for office work with visual display terminals (VDTs): Guidance on usability (ISO 9241-11). International Organization for Standardization.

16. Boyalakuntla, K., Venigalla, A. S. M., & Chimalakonda, S. (2021). WAccess - A Web Accessibility Tool based on the latest WCAG 2.2 guidelines. DOI: https://doi.org/10.48550/arXiv.2107.06799
17. Todorov, T., & Dochkova-Todorova, J. (2023). Digital Accessibility of Research and Educational Resources. Cultural and Historical Heritage: Preservation, Presentation, Digitalization, 9, 115–122. DOI: https://doi.org/10.55630/KINJ.2023.090208
18. World Wide Web Consortium (W3C). World Wide Web Consortium (W3C). Available at: https://www.w3.org/ (last view: 25-08-2025)
19. World Wide Web Consortium (W3C). Web Content Accessibility Guidelines (WCAG) 2.2. Available at: https://www.w3.org/TR/WCAG22/ (last view: 25-08-2025)
20. Dochkova-Todorova, J., & Varbanova, S. (2023). Web Design and Accessibility. Innovative STEM Education, 5, 201–207. DOI: https://doi.org/10.55630/STEM.2023.0523
21. Digital Accessibility for People with Special Needs: Methodology, Conceptual Models and Innovative EcoSystems. Available at: https://www.math.bas.bg/vt/ab/index.php?lang=en (last view: 25-08-2025)
22. Dimitrova, M., Bogdanova, G., Noev, N., Sabev, N., Angelov, G., Paunski, Y., Todorova, M., & Krastev, A. (2023). Digital Accessibility for People with Special Needs: Conceptual Models and Innovative Ecosystems. TechRxiv. DOI: https://doi.org/10.36227/techrxiv.23968047.v1
23. Bogdanova, G., Todorov, T., & Noev, N. (2023). Model of Taxonomy for Accessibility Ontology. Digital Presentation and Preservation of Cultural and Scientific Heritage, 13, 69–74. DOI: https://doi.org/10.55630/dipp.2023.13.6
24. Sabev, N., Bogdanova, G., Noev, N., Todorova, M., & Kapralov, S. (2024). Methodology and conceptual model for digital accessibility assessment of websites and documents for people with impaired and lost vision. In M. E. Auer, U. R. Cukierman, E. Vendrell Vidal, & E. Tovar Caro (Eds.), Towards a hybrid, flexible and socially engaged higher education. ICL 2023. Lecture Notes in Networks and Systems, vol 911 (pp. 463–473). Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-53382-2_45
25. Bogdanova, G., Todorov, T., Noev, N., Sabev, N., Chehlarova, N., Todorova, M., & Krastev, A. (2024). An ecosystem for the provision of digital accessibility for people with special needs. Information, 15(6), 315. DOI: https://doi.org/10.3390/info15060315
26. Bogdanova, G., Todorov, T., Dochkova-Todorova, J., Noev, N., & Sabev, N. (2024). Design and development model of a web accessibility ecosystem. Information, 15(10), 613. DOI: https://doi.org/10.3390/info15100613
27. Bogdanova, G., Todorov, T., Noev, N., Tomov, Z., & Chehlarova, N. (2024). Model of Computer Game at Education of Visually Impaired People. In 2024 International Conference on Information Technology Based Higher Education and Training (ITHET) (pp. 1–7). IEEE. DOI: https://doi.org/10.1109/ITHET61869.2024.10837626
28. Todorov, T., Bogdanova, G., Dochkova-Todorova, J., & Sabev, N. (2024). Accessibility of E-Learning and STEM Education. TEM Journal, 13(4), 3505–3513. DOI: https://doi.org/10.18421/TEM134-83
29. Li, X., He, Y., Song, H., & Gao, G. (2023). Applying the ADDIE Model to Optimize Blended Foreign Language Teaching Practice: A Case Study of the "Sino-Japanese Intercultural Communication" Course. Academic Journal of Management and Social Sciences, 4(2), 63-67. DOI: https://doi.org/10.54097/ajmss.v4i2.12128
30. Duyster Borreda, J. (2022). International models and influences on Catalan nationalism: regionalism, gender and the ethnic-civic dichotomy, 1880–1920. National Identities, 24(5), 539–557. DOI: https://doi.org/10.1080/14608944.2021.1925348
31. Zulkepli, N., Albakri, I., Megat Khalid, P., & Farish, M. F. (2024). Using the ADDIE Model to Develop an Online Professional Development Program for Non-Specialist ESL Teachers. Arab World English Journal, 10(19), 302-313. DOI: https://doi.org/10.24093/awej/call10.19
32. Molenda, M. (2015). In search of the elusive ADDIE model. Performance Improvement, 54(2), 40–42.
33. Interaction Design Foundation. Inclusive Design. Available at: https://www.interaction-design.org/literature/topics/inclusive-design (last view: 25-08-2025)

34. Lessa, G. (2025). Inclusive design for cognitive disabilities. Revista ft, 29(142), 8–9. DOI: https://doi.org/10.69849/revistaft/ra10202501081508
35. Basham, J. D., Blackorby, J., & Marino, M. T. (2020). Opportunity in crisis: The role of universal design for learning in educational redesign. Learning Disabilities: A Contemporary Journal, 18(1), 71-91.

Received: 25-08-2025 Accepted: 18-12-2025 Published: 29-12-2025

Cite as:
Petrova, P., Dochkova-Todorova, J. (2025). "Designing Inclusive Web Interfaces: a Standards-compliant Accessibility Menu Widget for Assistive User Experience in Front-end Applications", Science Series "Innovative STEM Education", volume 07, ISSN: 2683-1333, pp. 185-200, 2025. DOI: https://doi.org/10.55630/STEM.2025.0716