

NNELS



Accessibility Report for TumbleBooks

Conducted and published by the [National Network for Equitable Library Service \(NNELS\)](#), Vancouver BC, June 2021.

This report was written with support from the Government of British Columbia, with support from Northwest Territories Public Library Services, Department of Education, Culture and Employment, Government of the Northwest Territories, and the Government of Canada's Social Development Partnerships Program - Disability Component.

The opinions and interpretations in this publication are those of the author and do not necessarily reflect those of the Government of Canada.

This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)

About NNELS

The [National Network for Equitable Library Service \(NNELS\)](#) is a digital public library of ebooks for Canadians with print disabilities, and an advocate for an accessible and equitable reading ecosystem for Canadians with print disabilities¹. NNELS supports principles of openness, inclusion, and choice. NNELS is hosted by the [BC Libraries Cooperative](#), a community service not-for-profit cooperative and a national leader in information and technology services.

Our team of Accessibility Testers has expert knowledge in the areas of accessibility testing, analysis, software development, and leadership. The team works to educate and advise publishers, technology vendors, and public libraries on best practices for accessibility. Our testers have lived experience with a range of print disabilities, including blindness, low vision, and learning disabilities.

¹ Print disabilities are defined by Canada's Copyright Act and include visual, mobility, or comprehension impairments such as dyslexia.

Accessibility Summary

This report covers accessibility findings of TumbleBooks, BiblioEnfants, TeenBookCloud, and TumbleMath. The user experience of these reading platforms for users of assistive technologies is inconsistent: some functions are fully accessible, and others are difficult or impossible for screen reader users to access.

The website is more consistent and usable than the mobile apps, which pose serious accessibility barriers. Many of the challenges when using the iOS or Android apps relate to unlabeled or mislabeled buttons, which makes their function unclear for screen-reader users. While the text of ebooks is readable with a screen-reader in iOS and Android apps, there are significant problems for navigating ebooks with assistive technologies.

Across all platforms, visual adjustment preferences are available, but there is no dyslexic font.

TumbleMath: the most significant accessibility barrier for blind learners is the lack of image descriptions for visual content.

TeenBookCloud website: If the top menu is collapsed, the button to open it does not respond to a screen-reader and is not reachable with a keyboard.

Introduction

TumbleBooks Library provides several catalogues to public libraries and educators. These include a wide range of text, audio, and graphical novels for children and teens. Using screen-reading and magnification software, our team assessed the usability of the following products across supported platforms.

- TumbleBooks (apps and website)
- TeenBookCloud
- BiblioEnfants
- TumbleMath

A complete list of all the software and operating systems used in this assessment can be found in the *Systems and Assistive Technology* section of this report. The objective of this assessment is to determine the usability experience of readers with print disabilities and to what extent they can access audiobooks and ebooks through their local public library effectively and efficiently. It is important that readers with print disabilities can choose reading systems that offer the accessibility features they require.

The results show consistent accessibility and usability barriers across all products and operating systems. Some of the accessibility barriers were found across multiple products, and they will still be documented in each of the relevant sections below. Where necessary, recommendations will be added to help developers work toward a more accessible product.

Introduction to Assistive Technology

All mainstream operating systems include built-in screen readers (Narrator on Windows, VoiceOver on Apple devices, and TalkBack on Android) that read the contents of the screen out loud, allowing users with visual disabilities to browse apps and websites, send and receive texts and emails, and accomplish many other tasks with ease. Keyboard commands and custom touch gestures provide a flexible way for a user to find and interact with the controls on-screen. Windows also has alternative screen-reading software available, most notably a commercial option called Job Access with Speech (JAWS) and a free and open-source option called Non-Visual Desktop Access (NVDA). The text spoken by a screen-reader can be sent to a refreshable braille device. Mainstream operating systems are also equipped with user interface magnification, large text options, and high contrast viewing mode to assist people with low vision.

To ensure usability and accessibility of an application by people with print disabilities, all functions and controls must be accessible using assistive technologies (including screen readers, screen magnification software, and switches and voice controls for people with mobility impairments). The DAISY Consortium explains that the basic assumption of accessibility evaluations is that reading systems “should support reading with eyes, ears, and fingers.” (DAISY Consortium, 2017). It should be possible for users to read the content of the document by:

- Reading the text with screen readers or self-voicing text to speech (TTS) applications.
- Adjusting the display including font size, alignment, and color contrast, or combination of some or all these options.
- Reading the text with a refreshable braille display.
- Reading with assistive technologies designed for persons with dyslexia or other disabilities.
- Reading with the app’s built-in read aloud functions.

The principles for accessible reading systems can be found in the Web Content Accessibility Guidelines (WCAG 2.0) which is also used to make recommendations for the mobile applications and website. The principles of the Web Content Accessibility Guidelines state that all buttons and controls must be accessible and useable by people with disabilities:

- Content must be perceivable.
- Interface elements in the content must be operable.
- Content and controls must be understandable.
- Content must be robust enough to work with current and future technologies.

Accessibility Performance and Recommendations

In the following sections, we dive deeper into specific accessibility issues found in each TumbleBook Library product. The testing results and related recommendations are discussed as they pertain to:

- Signing In

- Discovering and Managing Books
- Searching for Books
- Audio and Video
- Catalogue Items
- Visual Adjustment
- TumbleMath Quizzes

The accessibility issues are highlighted and discussed in each section (as applicable). The *developer recommendations* sections contain the suggestions for improving the interface of each platform.

TumbleBooks

iOS and Android Apps

The Android and iOS TumbleBooks apps behave very similarly and seem to be based on the same web view, so they will both be discussed together in the following sections. If there are differences between the performance on iOS and that of Android, those differences will be noted where necessary.

- Tested iOS Version: 14.3, 14.4
- Tested iOS App Version: 2.37
- Tested Android Version: 10
- Tested Android App Version: 1.38

The most serious accessibility barriers, found when using the apps with a screen-reader, relate to button labels. Screen-readers rely on text labels to identify buttons and other controls to the user. When these labels are not added, the user may need to rely on trial and error to access sections of the app, and they may not be able to locate the function at all. For instance, many screen-reader users were not able to find the logout button because the button label identifies it as “app test”.

Signing In

The process of logging in is largely accessible, but testers found several labelling issues throughout this screen:

- The selectors that allow one to choose between school or library login have no control type. This makes it difficult to determine that they are related to each other. Changing these controls to radio buttons would solve this problem.
- The about button has the word “button” in its label, resulting in most screen-readers identifying it as “about button button.”
- Most testers who rely on a screen-reader did not find the option to log out of the app. This is because the button has a text label of “app test”, which VoiceOver and TalkBack announced in place of “log out”.
- The help button has no label at all; testers only found it through trial and error.

Discovering and Managing Books

The multiple submit buttons available on the search screen (for each search option: author, title, etc.) could become confusing for screen-reader users who may not instantly know which text field is next to each button. Changing the label of each to include the criteria name, such as "submit author search," would help eliminate this confusion.

As the user types a search query, the app will show autocomplete results at the top of the screen. These appear below the search field visually, but because of where they are placed in the DOM (document object model), VoiceOver and TalkBack present them out of order when using swipe gestures. A user would need to explore the screen by touch or swipe past every other to find the suggestions, and most users may not know there *are* suggestions.

While most links on the main screen are labeled, each item in the list of sections (such as read-alongs and ebooks) has a plus sign after it. When VoiceOver and TalkBack users navigate the app with swipe gestures, they must move past each of these + controls, making the navigation process unnecessarily cumbersome. Similarly, each item in the footer menu has a vertical line which is read as a separate control.

Items can be browsed in either cover or detail views. Cover view provides the book title in the alt-text of each cover. In the detailed view, additional information such as book's author is displayed next to each item but navigating with a screen-reader is much more cumbersome because of the extra controls for each book. Since all screen-readers have a method for moving to the next or previous heading, having each book title as a heading level 2 would make this process much easier while still retaining the same information.

The button to add a book to Favourites is improperly labeled "submit". On iOS, when a book is added to favorites, the resulting alert reads out a URL (<https://mobile.tumblr.com>). This is quite verbose and confusing, especially for young readers.

Audio and Video

There are multiple interfaces for listening to audio and video content, and the accessibility of each is quite different. Testers were not able to determine the factors that cause a piece of content to be opened in a particular view, so example titles will be given.

The view used for language learning is not usable with a screen-reader. Activating any of the links or play controls has no effect, and on iOS, VoiceOver's focus jumps to the top of the page when a link is selected. The previous and next buttons are mislabeled as » and « symbols on iOS, and have no label at all on Android.

There are two views for playing video content. The first, used for titles such as *Ada Lovelace*, is usable, but with some misleading or incorrect labels. For toggle buttons, each state is read twice (for instance, "volume on volume off volume on".) The *Play* button is labeled "previous", and the *Back* button is read as "back arrow". The *Help* button is called "Help on help off help off", and nothing happens when it is pressed. Testers on iOS also reported that VoiceOver became more and more unresponsive as video playback continued, until it was necessary to close and re-open the app.

The second video view, used for titles such as *The Secret Co-creator of Batman*, is much more accessible, with clear labels on buttons and time readouts. Users need to double-tap on the video to show controls, so a hint to screen-reader users that instructs them to do so would be helpful. Testers on iOS also reported that when the *Done* button is pressed, the play controls on the resulting page do nothing when pressed. It is necessary to press the "back arrow" button and then select another video instead. On Android, the controls could not be located by swiping, and testers needed to find them by exploring the screen and activate them before they disappeared.

Ebooks

- While the text of ebooks is readable with a screen-reader, testers were met with many challenges when trying to navigate a book.
- In read-along titles, the controls for playback need to be located by touch, as they cannot be reached with swipe gestures. The *Play* button is also labeled "next".
- A screen-reader user should be able to scroll in any of the four directions with a three-finger swipe on iOS, or a two-finger swipe on Android. The text of a book cannot be scrolled in this way using VoiceOver, though it works correctly on Android. This is because VoiceOver checks whether the text is scrollable rather than simply passing through the swipe gesture to the app. It is possible to swipe right or read continuously by paragraph on both operating systems. There is a redundant piece of text that says "1 of 6", "2 of 6", etc. depending on the chosen page.
- The controls that appear when tapping the book text cannot be activated with either VoiceOver or TalkBack. These include buttons related to bookmark and chapter selection, as well as visual settings. Nothing happens when one of these controls is double-tapped; they simply disappear and leave the user where they started. This is a significant barrier to being able to switch chapters and change reading preferences. Additionally, low-vision testers reported that these controls are quite small and could easily be changed to occupy more of the screen for better visibility.

Visual Adjustment

Adjustments can be made to the size and appearance of text to create a good reading environment for low-vision readers. However, these settings do not save between books, so it is necessary to re-configure them every time a new title is opened. When the visual settings screen is opened, the text of the current book is dimmed and mostly obscured, making it difficult to see changes as they are made. Finally, while text spacing can be changed, a dyslexic font is not available, and adjusting the size of text does not automatically adjust character spacing, leaving the newly-enlarged characters and words with less and less space between them until the appropriate settings are manually configured to account for the new text size.

Development Recommendations

- Remove redundant "+" controls between menu items on the home screen.
- Label the buttons and text fields in the search screen to accurately reflect the criteria they search for
- Add headings to book titles in the details view
- Ensure all actionable buttons and other controls have a clear label

- Make text spacing settings scale with text size
- Add a dyslexic font
- Ensure it is possible to use controls such as chapter menu and text options with VoiceOver and TalkBack

TumbleBooks Website

The website was tested across all major operating systems using a variety of assistive technology and browser combinations. Testers found that most sections of the website were usable, but many controls were missing a clear label, making basic functions unnecessarily hard to find.

The sections below discuss specifics of the tested website features. If a web browser is not mentioned by name, it should be assumed that every browser exhibits the same behaviour.

General

On the home page, headings and lists are used well to give some semantic structure to the page. A nice touch would be to use <nav> elements for the two menus at the beginning of the page. Using <main> and <footer> HTML elements for the main content and footer area would also help screen reader users better understand the page structure.

In many cases, the <title> of each page is quite similar and does not accurately describe the section of the website being displayed. Having unique titles for every page, such as the name of the current book or section of the website, would make it much easier for users to determine which page they are on by checking the title bar.

Signing In

Many testers did not need to sign in, as their libraries had convenient links to launch the platform directly. Of those who did, all found it to be an accessible process.

Discovering and Managing Books

Much like the mobile app, the website contains categories such as read-alongs and story books, each with their own view. Screen-reader users noticed that each of these category names were presented on two lines, which causes some screen-reading software (such as NVDA) to announce the lines separately, as though there is one link called “Read” and another called “alongs”.

After selecting one of these categories, a page loads with recommended titles from that category. When books are shown in the default cover view, each book cover has an alt-text of “book cover”, which makes it impossible to determine the title of the book. The only way for a screen-reader user to find the title is to open the details page for each book. These “book cover” images are also seen on the book details page under the *You May Also Like* heading. When using the *Sort By Title* page, the cover images have alt-text that contains the book title.

The pagination links (to move through pages of results) are accessible to screen-readers, but the » and « symbols are often not announced unless a higher punctuation verbosity is

set. To avoid confusion, especially for young readers, text such as “previous page” and “next page” should be used. Screen-reader users also could not determine which page of the search results were displayed. This could be accomplished by adding a piece of text indicating which results are currently shown (results 1-10 of 35), or by removing the link markup from the currently active page number.

The combobox to sort results by author, newest item, etc. is accessible. However, in Windows browsers using the arrow keys to change the filter will result in the page immediately reloading, which causes focus to be moved away from the combobox. Users of these browsers can press alt+down to interact with the combobox, but it’s considered good practice to only reload the page after a confirm button is pressed.

The *Book Details* page has no heading hierarchy; every heading is at level 2. The book title should be a level 1 heading so a user can quickly find the relevant section of the page. The other headings (*Book Details, You May Also Like*) can remain at level 2, since they fall below the book title on the page.

The playlist section for audiobooks is quite difficult to use with a screen-reader. Each item in the playlist is an image with an alt-text of “playlist”, which prevents the user from identifying items. The *Remove* button next to each item does not have a label (which is also true on the *Favourites* screen as well).

When playing a playlist, having media controls hidden by default is problematic in some browsers. Firefox blocks content from playing automatically, and it is difficult to reach the video element in order to reveal controls. From a screen-reader perspective, it would be better to have controls shown or a visible button to accomplish this.

Searching for Books

The search page contains separate text fields for title, author, and subject, as well as separate *Go* buttons for each of the extensive search options. These text fields have no label associated with them, so when using tab or other shortcut keys to move to the next or previous text field, it is unclear which of the three is in focus. In addition to these labels, being able to identify each *Go* button by the type of search it performs would be helpful. This could be accomplished with the title attribute, which is shown when the user hovers with a mouse and spoken shortly after a screen-reader user navigates to the button.

When typing a search query, screen-readers are informed that one autocomplete result is available, even if there are none. Pressing the down arrow to move to that result will clear the text field. These suggestions do read correctly when actually present and can be navigated with up and down arrows. Pressing enter while focused on a search field does not perform the search; the user needs to tab or otherwise navigate to the *Go* button and activate it.

Each search result is presented as a level 2 heading. Unfortunately, the book details heading is also at level 2. For consistency and to allow easier navigation of search results, this should be changed to level 3, as the details are part of the search results at level 2. Pagination links are announced as « and » symbols, adding text labels to those links (such as “next page” and “previous page”) would be clearer.

Story books

When playing storybooks in the audio player, the *Play/Pause* button is mislabeled “previous” regardless of whether the audio is playing or paused. Additionally, when the *Play/Pause* button is pressed, the focus jumps away from the button, which is not expected behaviour. The label of the *Help* button switches between “help on” and “help off”, but no additional content appears for screen-readers when it is pressed.

The remaining and elapsed times show numbers without any accompanying text. For screen-reader users, this could be confusing without further context. The best way to differentiate the two is to repeatedly read the numbers while the book is playing.

Read-alongs

Most of the controls have well labeled alternative text and much of the interface can be navigated with the keyboard. Having an entire chapter visible at once allows a screen-reader to use continuous reading rather than needing to read each page individually.

Screen-readers often navigate text without visually scrolling the screen. In this book viewer, such behavior means that the page indicator does not change as the screen-reader navigates, and the page navigation commands will not work consistently. In simple terms, the book reader will “think” the user is still on the first page of the chapter, even though the screen-reader is navigating a different part of the text. This is a difficult problem to solve, so to avoid confusion, this indicator could be hidden from screen-readers using an aria-hidden attribute.

When using the arrow keys to navigate the list of chapters in the chapter menu, the book text automatically updates to reflect the new chapter without the user needing to press enter. WCAG discourages this practice and recommends using a “go” or “submit” button to confirm the list box selection.

In the notes area, the close option is not available with keyboard navigation. The help text is read with a screen reader, though the user will have to explore the page to find that it is available. A similar situation occurs with the chapter menu, text and colour options, bookmarks, and notes. For many of the options, selecting the choice again closes the item if it is open. For example, this happens with *Chapter Menu*, *Text Options*, and *Colour Options*. It does not happen with the *My Notes* area. These buttons do not properly indicate their state—in fact, they are not buttons at all—so neither of these situations are particularly intuitive.

It is also possible to expand more than one of these options simultaneously, leading to page clutter and potential confusion. For example, a screen reader user can choose *Text Options* and *Colour Options*, and then both sets of options are available.

One alternative is to use a tab panel for these options so that a screen reader could better indicate which is selected and design the page so that only one could be active at a time. This would also encourage the screen reader user to explore the area below the tabs for new content.

Ebooks

The ebook view is similar to the read-along view but contains some additional accessibility barriers. Perhaps the most serious is the inability to use the chapter list and other expandable options. This appears to happen because those options only remain visible for a few seconds before disappearing. This is often not enough time for a keyboard user to find and choose the desired option. These controls are also not reachable with the tab key.

It is possible to pass the left or right arrow through to the webpage, which will turn to the previous or next page. However, as discussed with the read-along view, the screen-reader shows the entire chapter at once, so the user would have to press the right arrow several times to move to the next chapter of a book and there is no indication of when the chapter changes. A simple solution to this problem would be to include a “next chapter” control on the page, as well as adding a hotkey such as ctrl+left and ctrl+right to move between chapters. Adding the chapter number to the title bar would also give screen-reader users an easy way to determine which chapter they are on.

It is only possible to set one bookmark, and when returning to that bookmark, focus is shifted to the page that contains it. This does not set screen-reader focus to the desired page since the screen-reader sees the entire chapter at once, not just the text that is visually shown. This is one instance where setting focus to the specific page would be desirable.

Visual Adjustment

Overall, low-vision readers found the website easy to use. The ability to adjust text and background colour independently was appreciated. Character, word, and line spacing can be changed. Some users reported inconsistent behaviour when enlarging text, including an inability to scroll through pages and in one case, the text disappearing entirely.

The website lacks a dyslexic font and does not have a way to disable hyphenation, which could be particularly problematic to young readers with dyslexia. While font can be changed independently in most browsers, it would be helpful to have such an option included with other visual settings so that usage is as simple as possible.

Developer Recommendations

- Ensure all actionable controls have an accurate text label
- Change the alt-text of book covers to include the title of the book
- Ensure the ebook reader controls are not hidden until a close button is pressed
- Change pagination links to have a text label such as “go to next page”
- Add hyphenation and a dyslexic font to visual settings
- Make use of HTML landmarks such as main, nav, header, and footer to add structure to the page
- Do not allow more than one control to be opened at a time in ebooks or read-alongs
- Change the chapter selection control in read-alongs to a combobox and do not change the chapter until focus is moved away from the control, or until a confirm button is pressed
- Add buttons and hotkeys to move to next and previous chapter

- Label the buttons and text fields in the search screen to accurately reflect the criteria they search for
- In search results, change the book details information heading to level 3 so that book titles are the only level 2 heading
- Ensure help text is shown in book players when the help button is pressed

BiblioEnfants

The BiblioEnfants website is the French version of TumbleBooks. It is loaded automatically when choosing French from the language dropdown menu, and both platforms share the same code. All information in the TumbleBooks section above has been verified as accurate on the BiblioEnfants website as well. There is no dedicated mobile application.

Screen-readers respect the language of the site and will automatically begin speaking in French if an appropriate voice is available. This is good. However, most of the accessibility labels are not translated into French. These include:

- The “Book Cover” labels for each book in cover view, which are also incorrect (the label is “Book Cover” instead of the book title)
- The Read Online and Add to Favourites links on a book’s details page
- The Chapter Menu, Bookmark, Text Options, Colour Options, My Notes, and Help buttons at the bottom of the ebook reader
- All options contained within these screens (such as Choose the Font, Text Size, and Set Text Spacing To)
- The Advanced Search link on the homepage
- The entire advanced search page, including all buttons, text fields, and options (except for genres), even though the language selection on that page is set to French.

Improving the localization of these labels will make the website more accessible, though it will still contain significant barriers as noted in the TumbleBooks section.

TeenBookCloud

This platform does not have mobile applications, so only the website was tested. TeenBookCloud appears to share some code with TumbleBooks, so some of the issues discussed in these sections will be similar.

Screen-reader users found that many elements of the main pages were accessible, but accessibility issues still created confusion and inefficient navigation in some cases. Visually, the site was described as overly colourful for some low-vision readers, but quite intuitive and easy to use.

General

If the top menu is visible, the options are shown as links that can be easily found and activated with a screen-reader. However, when the menu is collapsed (which may happen on a smaller device), the button to open it does not respond to a screen-reader and is not reachable with a keyboard. When designing custom controls such as this one, it is important to ensure they respond to keyboard input as well as mouse clicks, as not

everyone uses a mouse. If this control were a link or a button, it could then be reached by the tab key and activated with enter or the space bar.

Discovering Books

The main page contains several categories, each one showing a selection of relevant books. These include fiction, nonfiction, enhanced ebooks, graphic novels, and more. Since these are separated by headings, it is easy for a screen-reader user to use their heading navigation commands to explore the available categories.

At the end of each category, screen-readers encounter a poorly labeled link with a name such as “BooksList categoryID=207.” These links will load a complete list of books from their respective category. After these links are two more seemingly redundant links called “Home”, which do in fact lead back to the same homepage.

Once a category’s page is loaded, the user can choose a detailed view or cover view. The cover view is simple and accessible, and each book cover is labeled with the book’s title. In the detailed view, the interface is usable with minor accessibility issues:

- The Add to Favourites button is labeled “submit”, so users may not be able to find it. This is also true on a book’s information page.
- The order of controls on this page makes it difficult to know which book the *Read Online* and *Add To Favourites* buttons correspond to. This would be much clearer if the heading containing the title of the book was moved so that it comes before these buttons.

Comboboxes are used to change between cover and detail views, or to apply a different sorting filter. When these are navigated to with the arrow keys, the page immediately reloads, throwing the user away from the combobox. A workaround is to press alt and down-arrow to “open” the combobox, make a selection, and then press enter to confirm it. However, it’s good practice to design comboboxes that behave this way by default, only reloading the page when a *Go* button is pressed. As noted in other sections of this report, most comboboxes across TumbleBooks products behave this way.

Pagination links use » and « symbols to indicate the next/previous page. Not all screen readers will read these by default. If they are not read, users will encounter silence when they arrow to those links. Therefore, each link should be labeled with text such as “next page” and “previous page” to be as clear as possible. One way to accomplish this without changing the visual layout is by giving the links an [aria-label](#).

Searching for Books

The *Advanced Search* page contains individual text fields for each search type, such as author and title. These text fields lack a label, so when navigating with the tab key, the name of the text field is not read. The user needs to explore the page and find the label next to its associated text field. Using an attribute such as [aria-label](#) would allow quicker, less frustrating searches to be performed.

Audio and Video

The player used for audio books and enhanced ebooks is quite accessible and easy to use. As with TumbleBooks, the elapsed and remaining time readouts do not have a label to identify them, so a screen-reader will read the numbers but a user may not know what the numbers mean. Bookmarks can be set and selected, though only one can be created at a time.

The video player is very accessible, apart from a single unlabeled button whose functionality is not known and which appears to do nothing when pressed. All other controls are labeled and reachable with the keyboard. The videos do not have audio description, though testers found it easy to follow the audio of most video content.

Ebooks

There are two possible views for ebooks, one that shows a single page at a time and one that shows an entire chapter at a time. It is possible to read text content in both views, but different sets of accessibility issues exist for each. The chapter view is also used for enhanced ebooks, and all accessibility issues described below still apply.

The single-page reader allows you to “tap the middle of the screen” to bring up a menu of options. Screen-reader users were able to do this by finding a blank line at the bottom of the page and pressing enter. However, the menu of options disappears after several seconds if an option is not chosen. For a screen-reader user, this is not even long enough to navigate back to the top of the page and read through the menu fully. Such a user would need to first figure out how to open the menu, then find a way to navigate to their desired option within the few seconds the menu remains visible. This would also be very problematic for users with motor disabilities, and challenging for users with cognitive disabilities.

Most basic functionality of the second ebook view (which displays an entire chapter as a time) is accessible and easy to use. The text is adjustable and readable with a screen-reader, and chapters can be easily selected.

Much like TumbleBooks, it is not possible to move directly to the next chapter, only to the next page. Since a screen-reader sees all text of a chapter at once, there is no indication that the page has been turned, and a screen-reader user will find themselves pressing the arrow key with seemingly no effect, until the next chapter loads. At minimum, a button or hotkey to move to the next or previous chapter would be helpful. Adding the current chapter title to the title bar would also be helpful for screen-reader users, who often check the window title for a hint about where they are.

The chapter menu brings up a list box containing all chapters of the book. This exhibits the same behaviour as comboboxes: Pressing down-arrow to move to a new chapter will immediately reload the page and throw focus away from the chapter list. One can still press alt+down-arrow to open the chapter list, but once a chapter is selected with the enter key, there is no indication that it has loaded and the list needs to be manually closed.

It is possible to expand more than one of the reader options simultaneously, leading to page clutter and potential confusion. For example, a screen reader user can choose *Text*

Options and *Colour Options*, and then both sets of options appear at the same time. One choice may be to use a tab panel for these options so that a screen reader could better indicate which is selected and design the page so that only one could be active at a time. This would also encourage the screen reader user to explore the area below the tabs for new content.

Enhanced Ebooks: Graphic Novels

The content of graphic novels is not accessible to any screen-reader. Even text appears to be displayed as part of the images, so screen-readers are unable to read that text. The links for chapter and page menu appear to be clickable with a screen-reader, but the chapter and page lists can never be found and do not appear in tab order. Lastly, it is not possible for screen-reader users to determine where they are in the graphic novel, as no part of the page contains this information, and it is not included in the page title.

Visual Adjustment

Overall, low-vision readers found the website to be easy to use. The ability to adjust text and background colour independently was appreciated. Character, word, and line spacing can be changed. The website lacks a dyslexic font and does not have a way to disable hyphenation, which could be problematic to readers with dyslexia. While font can be changed independently in most browsers, it would be helpful to have such an option included with other visual settings so that usage is as simple as possible.

Developer Recommendations

- Ensure all actionable controls have an accurate text label
- Ensure the single-page ebook reader's controls remain visible until a close button is pressed
- Change pagination links to have a text label such as "next page"
- Add hyphenation and a dyslexic font to visual settings
- Make use of HTML landmarks such as main, nav, header, and footer to add structure to the page
- Do not reload the page when selecting an option in a combobox, instead wait until focus is moved away from the control, or until a confirm button is pressed
- Do not allow more than one modal to be opened at a time in the full-chapter ebook viewer
- Add buttons and hotkeys to move to next and previous chapter
- Label the buttons and text fields in the search screen to accurately reflect the criteria they search for
- In search results, change the book details information heading to level 3 so that book titles are the only level 2 heading in the page

TumbleMath

TumbleMath is an online tool designed to teach math using stories, and to then quiz math learners on these concepts. NNELS was provided with test accounts through the Northwest Territories Public Library.

The most significant accessibility barrier for blind learners is the lack of image descriptions for visual content. Many of the questions on quizzes contain images, and none have alternative text to describe the content of the image. Some stories also contain image-based information which is not always described by the audio narration. Lack of text-based content also means that a low-vision user would not be able to adjust the contrast and appearance as easily. The website also suffers from lack of labels on many of the important links, inconsistent heading hierarchy, and unnecessary clutter that interferes with a screen-reader user's exploration of the page.

General

Navigation of the website's homepage is largely accessible, but better use of landmarks and heading hierarchy would make the pages easier to use. For instance, all headings on the homepage are at level 2, with no heading at the start of the main content. Some pages of the site have such a level 1 heading, including the [Common Core Portal](#), but this is rare and not consistent. It would be helpful to have a level 1 heading at the start of the main content on every page. Landmarks such as <main> and <nav> would also provide more semantic structure to the pages, which would give users alternate ways to navigate through large sections of the site.

Much like other TumbleBooks products, the title of every page is the same (in this case "TumbleMath.") Since screen-readers all offer a command to check the title of the current window, adding more useful titles to various sections of the site would help users determine where they are and keep track of multiple open tabs.

Below the list of book categories, a screen-reader user will encounter seven images with the alt-text of "TumbleMath". These are reported as clickable, but appear to do nothing when clicked, so serve only to clutter the page. If they are not useful, they can be hidden by setting their alt-text to an empty set of quotes. If they are useful, the alt-text should be improved so their meaning becomes clear, and they should be given a role of link or button. Similarly, two seemingly non-functional links can be found, these are read as "Home" after the list of titles in each category.

Searching for Titles

The *Advanced Search* page contains individual text fields for each search type, such as title, author, and publisher. These text fields lack a label, so when navigating with the tab key, the name of the text field is not read. The user needs to explore the page and find the label next to its associated text field. Using an attribute such as [aria-label](#) would allow quicker, less frustrating searches to be performed.

When no autocomplete results are available for a search, the screen-reader still reports that one result is shown. Pressing the up or down-arrow will alternately clear and restore the currently entered text.

On the results page, each title has two headings associated with it, one for the book title and another for "Book Details." Changing the book details heading to level 3 would allow users to navigate through results easily by moving through the title headings at level 2. Additionally, the *Add to Favourites* button on this page is unlabeled.

Exploring Titles

Once a category's subpage is loaded using one of the links in the top menu, the user can choose detailed view or cover view. Cover view is simple and accessible, and each book cover is labeled with the book's title. In detailed view, the interface is usable with minor accessibility issues:

- The *Read Online* link is badly labeled, a screen-reader will read a meaningless string of letters and numbers such as "H5Player ProductID=437&book=/H..." The button to add a book to favourites has no label at all.
- The order of controls on this page makes it difficult to know which book the *Read Online* and *Add to Favourites* buttons correspond to. This would be much clearer if the headings containing the titles of the book were moved so that they come before these buttons.

Comboboxes are used to change views or to switch to a different sort filter. When these are navigated to using the arrow keys, the page immediately reloads, moving the user away from the combobox. A workaround is to press alt and down-arrow to "open" the combobox, make a selection, and then press enter to confirm it. However, it's good practice to design comboboxes that behave this way by default, only reloading the page when a *Go* button is pressed.

Pagination links use » and « symbols to indicate the next/previous page. Not all screen readers will read these symbols by default. If they are not read, users will encounter silence when they arrow to those links. Therefore, each link should be labeled with text such as "next page" and "previous page" to be as clear as possible. One way to accomplish this without changing the visual layout is by giving the links an [aria-label](#).

Once a book is chosen and its details page is open, all action links on the book details page are badly labeled. The *Read Online* link has the same meaningless label (of letters and numbers) as described above. The *Add to Favourites* button is labeled "Submit", the *Quiz* link has a label such as "2.OA.A.1", and *Lesson* links have names such as "BookLessonPlans ProductID=395". This final label unintentionally gives a hint about the link's purpose, but it is still incorrect.

Reading Books

None of the tested books contain text which is readable by a screen-reader. If text content is available, it is vital that all users are able to access it and adjust it according to their reading needs.

In the player used for books with audio narration, the *Play* button is mislabeled "Previous", which can cause confusion since a screen-reader will find two *Previous* buttons. Whenever this button is pressed, focus is moved away from it, so it is not possible to play or pause the book quickly. The *Previous* and *Next* buttons do not have this problem and can therefore be pressed repeatedly to scroll through pages of the book. The *Help* button appears to do nothing when pressed. If new content is loaded, a screen-reader cannot see it.

Since blind users are limited to the audio narration, they may miss out on key information that will help them understand the material. The narrator will sometimes describe the visual content adequately, but this is not done as an accessibility aid, so they cannot be relied upon to do so, nor is there a way to know how accessible a book will be until it is read. Full access to the text and math content of these books would provide a consistent learning environment.

Video

Video controls differ by browser, so the accessibility is not controlled by the website (but the controls are largely accessible.) Much like the content of books, the accessibility of the content differs from one video to the next, but it is not possible to follow most of them by relying only on the audio since you are expected to be able to see the math and text being displayed.

Quizzes

The interface for choosing and completing quizzes is quite different from anything found in other TumbleBooks products, so it will be covered in some depth below. In general, text-only questions are accessible, but questions which contain graphics are not. This could be fixed by adding alt-text to the graphics.

Choosing a Quiz

The link for *Quiz Portal* in the navigation links does not appear to do anything when pressed. This may be because it visually causes the links for *Common Core* and *TumbleMath Quiz Portal* to appear, but with a screen reader, they are always available in the navigation as a nested list after the *Quiz Portal* item. This nested list is helpful since it conveys the page hierarchy.

On the page for *Common Core*, the table headers use standard table data (TD) tags rather than using table header (TH) tags. It is possible to understand the content by reading down the table, but the table appears to be used more for layout than for an actual table since it only uses one column. Using heading tags followed by a list with inline items (horizontal and not vertical to match the current layout) would provide a similar page structure but eliminate the table. This would be easier to navigate (using list and heading navigation commands) and would make more sense to a screen-reader user—especially someone younger who may not be used to reading needlessly complex tables.

On the page of the *TumbleMath Quiz Portal*, the table has headers, but there are several headers that apply only to a part of the table. Some screen-readers (such as NVDA) read all of the headers that apply to a column, which results in several unrelated headers being read. For example, in K.1 which is in the kindergarten section, the headers are read as "grade 1 grade 2 grade 3 grade 4 grade 5". This table would need to be marked up in more detail so that headers apply only to their appropriate cells, or divided into multiple tables, one for each grade.

Within each grade, there are different math concepts, and within each math concept there are several quiz sections. The description for each quiz section is in a separate row below its quiz number. One suggestion would be to combine the quiz section number and

description into the same cell, so that they are more closely related when moving by cell with a screen reader. This would eliminate the need to repeatedly move between the description row and the quiz list row.

After choosing the *Quizzes* link, to move from a summary to the individual quizzes, another page loads with all of the quizzes for that math concept. This page begins with a main heading at level 2. A level 1 would be preferred. The button to add this page to favorites is read as “submit” in some browsers, and completely unlabeled in others. The link to start the quiz is not labeled.

If a math concept from *Common Core* is selected, additional buttons are available in the summary page to view a lesson plan or related books. Both of these links have no label. The button to open a lesson plan PDF is also not labeled. Lesson plan PDFs were not tested extensively to determine whether a majority of them are graphical, but the text of the PDF documents is readable with a screen-reader.

Taking a Quiz

Image-based questions are not accessible with screen-readers and cannot be read or answered by someone who relies on one. Screen-readers do read text-based questions and they can be answered successfully.

Form fields such as edit boxes and radio buttons are consistently not labeled throughout the quiz-taking interface. This includes the text field for entering a name, as well as the answer choices. The result is that screen-reader users who navigate with the tab key will be told they have reached a text field or a radio button but will not be told about its purpose and will need to further explore the page to find the label. Some screen-readers will guess at the name of each control based on nearby text labels; NVDA does not.

The *Print Questions*, *About*, and *Back* links are not reachable with the tab key. The *About* link reveals help text that a screen-reader user can only find by moving to the bottom of the page. This would be better in a dialog that gains focus when the link is activated and keeps the user’s focus inside it until dismissed.

The *Back* link only functions when a quiz is not in progress, but screen-readers do not see this link as disabled. As a result, the link will appear to do nothing when taking a quiz. This should be hidden or otherwise marked up in a way that conveys this function non-visually.

The link to move to the next question is not properly labeled. Various browsers will read different labels, but they are never informative. In Chrome, it is sometimes read as an unlabeled graphic and sometimes as “btn_next.” In Firefox it is read using part of the URL: “quiz ProductID=2MDC8&IsCC=1&Bo...” It is also possible to move to the next question without choosing an answer. If this is not intentional, a warning would be helpful so the user knows they should guess the answer. Finally, a screen-reader user is given no indication that new content has loaded after activating this link and may not know the page has changed at all. This could be fixed by refocusing the user on the next question or adding an ARIA alert to let the user know the question is loaded.

Once a quiz is complete, it is possible to review the results and determine which questions were incorrect. The heading hierarchy is incorrect on this page: All headings are level 3, including the quiz title, the percentage score, and each of the incorrect questions. The links to *Print Results*, *Retake*, and *go Back* are visible to a screen-reader but not reachable with the tab key. The *Back* link does nothing when activated on the results page.

Visual Adjustment

Very few appearance options were found on the site. Since much of the important content is contained in images and videos, the format will be quite inflexible and difficult to adjust visually. In many cases magnifying an image distorts it, and even when it does not the font and contrast are not adjustable, which could also affect users with learning disabilities such as dyslexia.

Developer Recommendations

- Label all graphical buttons so screen-reader users can navigate the site effectively
- Add alt-text to image-based quiz questions where possible, or replace images entirely with text
- Make sure heading hierarchy is consistent on all pages, such as changing book details headings to level 3, and ensuring the first heading on a page is at level 1
- Add global settings to adjust text size and appearance
- Add labels to text fields, radio buttons, and other form controls, perhaps by using the [for attribute](#) on existing labels
- Eliminate homepage clutter, such as the multiple “TumbleMath” images
- Make use of HTML landmarks such as main, nav, header, and footer to add structure to the page
- Do not reload the page when selecting an option in a combobox, instead wait until focus is moved away from the control, or until a confirm button is pressed

Conclusion

This report provides an overview of accessibility barriers in TumbleBooks, BiblioEnfants, TumbleMath, and TeenBookCloud. The current user experience of the reading platforms could best be described as inconsistent, with some functions being fully accessible and others difficult or impossible to use. The TumbleBooks mobile apps, which have not been updated in several years, have a number of serious accessibility bugs. The website appears to be more consistent and usable. In the case of TumbleMath, much of the content is highly visual without text alternatives, and it will be frustrating and difficult for someone who is partially or completely blind to locate materials that are easy to follow.

In general, these websites would all benefit from more specific use of HTML to semantically link and identify controls. Many of the most serious barriers across all websites and apps are due to a lack of text labels, one of the easiest problems to solve on the development side. When a product is targeted toward children, it is especially important to ensure navigation and labels are consistent. By implementing the feedback and developer recommendations in this report, the TumbleBooks platform could provide an inclusive and engaging experience for print-disabled readers.

Systems and Assistive Technology

Operating Systems

- Windows 10 (20H2)
- iOS 14.2+
- Android 10

Mobile Applications

- TumbleBooks 2.37 (iOS)
- TumbleBooks 1.38 (Android)

Browsers

- Chrome 86 through 88
- Firefox 84
- Safari (iOS)

Screen-readers

- NVDA 2020.3 (Windows)
- JAWS 2021 (Windows)
- VoiceOver Touch (iOS)
- TalkBack (Android)

Magnification

- Magnifier (Windows)
- ZoomText 2020 (Windows)
- Zoom (iOS)

Contributors

The following people contributed to this report, including testers and editors.

- Karoline Bourdeau
- Leah Brochu
- Mélissa Castilloux
- Simon Jaeger
- Daniella Levy-Pinto
- David Kopman
- Riane Lapaire
- Ka Li
- Richard Marion
- Laetitia Mfamobani
- Deanna Ng
- Megan Selmer
- John Ylioja