

Accessibility API usage

Pournami Krishnan and Maaike Dokter - 9 January 2019

1. UI Toolkit used

The UI toolkit we have used for our website is Bootstrap, which is currently on version 4.1.3. Since we make use of samples that come from previous versions of Bootstrap, our code makes use of version 3.3.4. Looking at the major changes between version 3 and version 4 (since there were many), none of them included accessibility features and therefore we decided to move forward with the older version of Bootstrap, since we could make use of the examples we had found.

Bootstrap was created by Mark Otto and Jacob Thornton, who worked at Twitter at the time. The framework was created to fix the problem of inconsistencies and maintenance problems across the different internal platforms: Bootstrap allowed for consistency and usage by many. Originally, the name of Bootstrap was Twitter Blueprint. Bootstrap became open sources on August 19 in 2011. Within version 3, responsive design by default was implemented, meaning that the layout would adjust to the characteristics of the device, which is something important for accessibility as well.

The link to the main webpage is <https://getbootstrap.com>.

The main reason why we have used Bootstrap is that it is one of the most used HTML, CSS and javascript frameworks out there. This means that many people already use it and there is a lot of documentation available. Since we were still somewhat unfamiliar with building websites from scratch, we could make use of the documentation on Bootstrap (as well as answers on blogs or forums) in order to fix problems and incorporate examples well. Websites, such as w3schools.com, explain code from scratch (including Bootstrap, for which they have a separate section on their webpage) and also mention accessibility, helping us to implement any UI components using Bootstrap whilst keeping the webpage accessible.

Bootstrap itself has a small page on accessibility on their webpage. This is less extensive than documentation we have seen for other UI Toolkits. However, as mentioned before, since many people make use of this toolkit, it made implementing the toolkit itself easier. Therefore, we decided that we would rather take a closer look at accessibility ourselves. The documentation they have on accessibility can be found at <https://getbootstrap.com/docs/4.1/getting-started/accessibility/>.

When looking at more specific webpages, such as the webpage on Collapse (<https://getbootstrap.com/docs/4.1/components/collapse/>), there is a separate section at the end on accessibility and how to include ARIA elements within the code to make specific elements accessible. Therefore, not all the accessibility information can be found in one overview, but specific examples do explain some accessibility features that should be implemented in order to create accessible websites.

2. The website

The website created for the project can be found on <http://128.199.226.170/Frozen/>. This webpage is a fan page for the Disney animation movie Frozen. Frozen is about two princess sisters, one of whom has powers to freeze things. However, she does not control this power well. When things get out of hand, she flees her city and her sister goes on a search for her. Even though the story is based on a fairytale, the movie has gained a lot of followers, including people in a slightly higher age range.

This website is focused on the many fans the movie has gained over the years. In the next section, each page will be briefly described and the use of Bootstrap within the pages will be mentioned.

2.1. OVERALL WEBSITE

Throughout the website, some features have been repeated on every webpage (such as the footer, the navigation bar and the skip to content link). One of these features makes use of Bootstrap, namely the navigation bar. Using the navigation bar, the users can navigate throughout the webpage and visit the different websites. Their current location within the website is highlighted in the navigation bar.

2.2. PAGE 1: HOME

The home page has a simple structure and only includes an image carousel besides the previously mentioned navigation bar. This page opens first when users visit the webpage. Here, the user sees a clear overview of what is available on the webpage by scanning the navigation bar. The image carousel provides some nice images of scenes from Frozen. By keeping the page structure very simple, it does not overwhelm the users when they visit.

The image carousel used on this home page makes use of Bootstrap. Moreover, the buttons used for going to the previous or next image as well as the pause and play buttons, make use of Bootstrap as well.

2.3. PAGE 2: STORY AND CHARACTERS

The second page of the website tells the plot of the beloved Disney movie. Moreover, a table is given providing information on the different characters in the movie, their characteristics and the actors that play them in the original version of the film. Besides the features that are provided throughout the website, this page only makes use of clean HTML code (including in the links in the text as well as the table) as well as normal CSS formatting. Therefore, no Bootstrap has been used within this page.

2.4. PAGE 3: GALLERY

The gallery page shows different thumbnails of images the user can press. When the user selects and presses an image, a pop-up will appear with the image enlarged. The user will then have three options of closing the pop-up, going to the previous image or going to the next image. Moreover, the user can click outside of the pop-up to make it disappear and they can scroll on the normal webpage when the mouse is not hovered on the pop-up. This page uses a lot of Bootstrap code in combination with javascript in order to make all the user interactions work.

2.5. PAGE 4: MUSIC

The music page provides three different buttons. When the user clicks on one of these buttons, the area expands and the lyrics to the selected song are shown. Clicking again will collapse the button again. This has been done for three different songs. In the end of the page, a list of the featured songs within the movie are given. The expandable menu, also known as an accordion, was created by using Bootstrap.

2.6. PAGE 5: WRITE TO OLAF

The last page of the website is a form where people can write a message to Olaf, the lovely snowman in the Disney movie. Here, users must include a nickname, a valid mail address as well as a message in order to be able to submit the form. When they have made an error, the error will be indicated and an error suggestion will be shown wherever possible. After successfully submitting the form, a status message will appear which will update the user that their message has been send.

3. Accessibility result

In total, four out of five pages make use of some Bootstrap elements. We will go through those four pages (we will skip the Story and Characters page, since it does not make use of any UI Toolkit). We will explain when and how Bootstrap has been implemented within the specific webpage.

3.1. OVERALL WEBSITE

The Bootstrap function Collapse was introduced to ensure that the website can be adapted to different window sizes. Since this does not require any UI elements, nothing had to be changed in order to make this accessible: adding this feature would make the website accessible in general since the website would adapt automatically to the advice it would be displayed at.

The navigation bar makes use of Bootstrap. The navigation bar layout has been designed through different code. The different elements look like buttons, but they have been programmed as links. After investigation on the internet, many websites create menu bars using links (e.g. <https://www.w3.org/WAI/tutorials/menus/structure/>). Therefore, we decided to not change these items to buttons since the user would be redirected to a new webpage, which would also be the case when pressing a link.

In order to ensure that a screen reader would not read the name of the page twice (or read the icon next to the name of the page), we did not give the icons alternative text: in this navigation bar, the icons are purely used as decoration.

When selecting the different pages in the navigation bar using the keyboard function, the selected page will underline in order to show the focus of the user. Moreover, the current page of the user is highlighted using a different colour.

3.2. HOME PAGE

The carousel in the homepage was created through using Bootstrap. The carousel makes use of parent and children items, which would require aria-attributes in order to be accessible. However, the way the carousel is used here, the items are not connected to one another and the parent-child attributes are not necessary for assistive products to interact with the UI element.

The one thing that we had to implement were the UI elements in the carousel, namely the previous and next arrows, as well as the play and pause buttons.

Both the previous and next arrows were created using the link element. Therefore, the *role* "button" was added to ensure assistive technologies would understand the user could click on these buttons. *Aria-labels* were added to describe what would happen when the buttons would be pressed (e.g. "Show previous picture").

The pause and play buttons were programmed as buttons and therefore did not need to add or change the role of the element. However, an *aria-label* was added here also to ensure the user would understand what would happen when the button was pressed (e.g. "Play slideshow").

3.3. GALLERY PAGE

The gallery page ensures that the image the user clicks on will pop-up as a large image on the screen. This feature uses Bootstrap, more specifically the *modal* code. When editing the *modal* code, some accessibility features needed to be implemented. First, we needed to ensure that the *role* was set to dialog in the code. Since the *modal* was only used to show images and not display text, no other things were adjusted in the overall modal code.

However, the pop-up images also included three user interface options, namely a close option, a previous option and a next option. Each of these items received the *role* button and an *aria-label* to describe their purpose.

Moreover, we needed to ensure that all the images would receive their own image title instead of a generic title created through the *modal* code that would be equal for every image pressed. Here, we adjusted this in the javascript code that was also used to make this page work. We ensured that the *alt text* of the image that was selected was also shown as an *alt text* when the *modal* code was working.

Moreover, code was added to ensure that the right focus remained on the elements: at first, the user must only focus on the different thumbnails. When the user presses enter on a thumbnail, the *modal* code will enlarge the picture and the focus shifts to the user to only be able to select the three options of close, previous and next. These options were added using javascript code.

3.4. MUSIC PAGE

To create the accordion on the music page, the accordion example of Bootstrap was implemented. This example code can be found on the webpage https://www.w3schools.com/bootstrap/bootstrap_collapse.asp.

Here, the three different items the user could click to expand were identified as a link. However, if an assistive technology would announce the item as a link, the user would expect to be redirected to another webpage. Therefore, the *role* button was added to each item in order to ensure that the user would understand they could click on the button and something would occur.

Moreover, the status could change from *expanded: false* to *expanded: true* when a user would click on the element. Since changing the element from link to button would be beneficial to all users, it was decided to not add these as ARIA elements but change it in the standard code so it would be changed for all users. By ensuring the *expanded* elements were added to the code, the *aria-expanded* elements were automatically triggered.

3.5. WRITE TO OLAF PAGE

The Write to Olaf page includes a form where users can enter their information. Javascript is used in order to check if information has been entered (correctly) and that a status message will appear if the user needs to make changes before being able to submit the form.

However, Bootstrap was used here in order to create the layout of the text boxes. Here, within each text-box, an *aria-label* was added in order to write a more specific instruction. Where a user with a good eyesight can read the *name* of the text box as well as the *placeholder* which gives an example of what to enter, the user who relies on audio for understanding the webpage will receive a more clear instruction on what is expected of them (e.g. The *aria-label* "Enter your mail address" versus the *name* "mail" and the *placeholder* "name@example.com").