Project Report

IDG2012 – Web accessibility, usability, and ethics: final assignment

Introduction

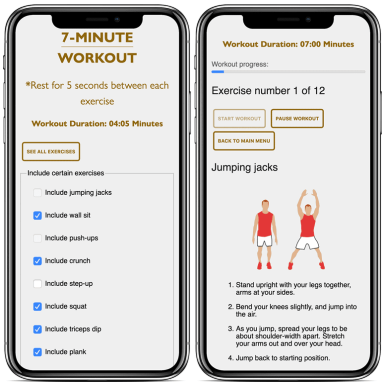
In this project, we made a single-page website that instructs people through a 7-minute workout. The page allows the user to choose predefined exercises and exclude certain body parts. We based the visual design of the page on a given poster containing the figures and text. We also established the accessibility and usability standards of predefined personas. These personas and their unique situations ensure a broad and diverse audience can use this web page. This way, we create a more including workout routine.

Usability

To make the webpage more usable, we have carried out multiple practices and tests.

Responsive Design

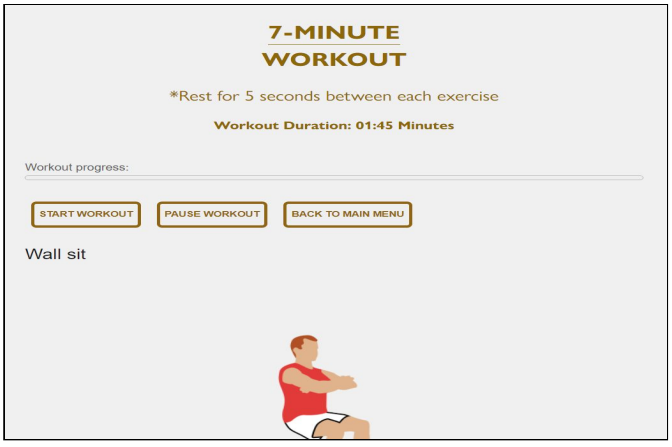
The page has a responsive design. A responsive design makes the webpage usable on a wide range of different devices. The page works well on larger screens found on desktops and laptops, medium-sized screens on tablets, and smaller screens on mobile. We even got the webpage working on the smallest screen we could find – an Apple Watch. This way, by testing our page on multiple devices, we can ensure all our personas will be able to access the page.



Visibility of system status

When accessing the web application, the user will realize which state the website is in because of the informative headers and text. As shown in the photo below, the system will ask users to do a specific task. This will let them know that the website is in a state which requires user interaction to proceed.

When a user has checked any of the checkboxes, the user will get visual feedback. The visual feedback will let the user know that they are doing something right and that user interaction is supposed to be present. When a user has started a workout by choosing the desired exercises and clicking on the "start workout" button, the system provides a new interface. The new interface shows the user three buttons with a clear label that describes their function. The progress bar is also shown but is not yet filled, telling the user that some action needs to occur. When a user clicks the "start workout" button and starts the workout, the button gets more transparent as visual feedback. The progress-bar starts to fill according to the workout length. The workout begins, and the user is now aware that the system is in a state where the user is supposed to watch and, in this case, do the exercises shown physically.



User control

The user has complete control over the parts on the website. The control ranges from choosing exercises, excluding certain body parts, deciding when to start the workout and when to pause it.

We used checkboxes to let the user decide which exercises to include. Since checkboxes come with a "undo" function, you can just as quickly uncheck a checkbox as you would

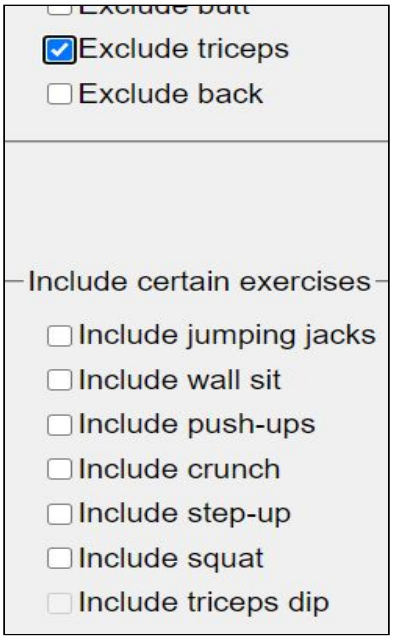
check it. This built-in functionality gives the user an effortless way to undo an unwanted action.

A user can go back to a "safe state" by clicking the "back to main menu"-button. The "back to main menu"-button will clear everything and let the user start from scratch, like an "emergency-exit."

We also added a page that comes as a bonus to the main page. It is a page explaining all the exercises, so the user can see which exercises they want to do before they include them in their workout. This page also contains the same "back to menu" button at the top so that the user can go back to the main page.

Error prevention

One of the first errors that one might think of, is when someone has selected a couple of exercises and then checks out one of the checkboxes that are supposed to exclude that body part. A user might think that an error will occur, but that is not the case. We have ensured that exercises that focus on a particular body part will get removed from the workout, both in the user interface and in the JavaScript code. We have included an illustration of what happens to the checkboxes when a user checks on "exclude triceps," the exercise "triceps dip" becomes unavailable.



Recognition rather than recall

Recognition is one of our primary strengths, all thanks to our layout. All buttons share the same design and are made recognizable. The buttons are also placed where users will easily spot them: close to where their function lies. For example, the pause button is placed near its functionality, just above where the action happens.

Our design also has a step-by-step construction. The user will not need to remember how to use the website. Step 1 is choosing the body parts the user wants to exclude. Step 2 is choosing the exercises the user wishes to include. Step 3 is starting the workout. Checkboxes also work as identifiers for letting the user know which exercises are selected. They do not have to remember which exercises they chose.

Flexibility for power users

As mentioned above, our website has a step-by-step construction. However, power users that are comfortable with our website will want to be more efficient. We have solved this issue by letting users scroll down to the bottom where the user can include exercises, allowing them to skip the "exclude body parts"-section. Suppose a power user wants a faster solution. In that case, they can press the "start workout"-button without including any exercises. Pressing the button without any exercises selected will automatically have every exercise on their workout, without needing to click on every exercise they wanted.

Consistency

Controls in our web application behave as users would expect it to and similar to other websites. Buttons like "start workout", "pause workout," and "continue workout" are all controls that work just as in most other related websites. The step-by-step construction is also something one can see on many other training/workout websites (freetrainers 2020) and (fitnessyard 2020).

Aesthetic and minimalist design

The webpage uses a basic setup without many bells and whistles. The Elements shown on the page all support the user's primary goals.

User testing and how it changed our design

For this project, we did not manage to user test the page with a realistic target group. We have tried to put ourselves – and some of our fellow students – in the shoes of the predefined personas. While the user-testing was not optimal, it has resulted in a few design changes, leading to increased usability.

We tested the usability by giving our test objects a specific task. A task we gave was as follows: "open the page, exclude arms, choose the lunge, running and plank workout, start the workout, then lastly pause the workout after completing the running workout."
The feedback we got was that our original layout did not make sense. Our original design displayed the chosen exercises at the top, a fieldset for including exercises in the middle, and a fieldset for excluding specific body parts at the bottom. In other words, our original design displayed the results at the top, resulting in the entire page to shift down after including an exercise.
We changed the design, making the included exercises show up at the bottom, near the "start workout" button. This way, the page will not shift down. It also gives a visual reminder of what exercises are included before starting.
We also switched the position of the two fieldsets based on the feedback. We made excluding certain body parts the first fieldset and including exercises the second. When excluding a body part, the exercises containing that body part will be disabled. Disabled checkboxes are easy to skip.

Accessibility

To make the webpage accessible, we have carried out multiple practices and tests.

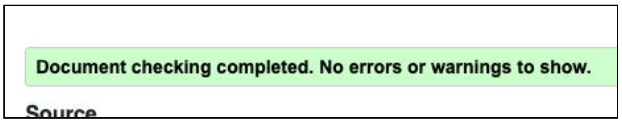
Used correct HTML tags whenever possible

There are multiple advantages to using the built-in HTML tags. These tags provide a semantic value that both sighted users and assistive technologies can benefit from.
For example, these built-in HTML tags already have keyboard accessibility out of the box. Screen readers also benefit from these built-in tags. As a user moves through the content, the screen reader reads each header and paragraph, explaining their level and what they contain.
On our webpage, we have made sure to do the following:

- On our webpage, the first heading is an `<h1>` tag, and the second heading is a `<h2>` etc.
- A <header> tag encloses the header.
- A <main> tag encloses the main content.
- Buttons use the `<button>` tag.
- Forms are enclosed with a `<fieldset>` tags, and the `<input>` elements inside the forms have labels.

- Images use the <img> tag with an "alt" attribute to ensure both sighted users and blind users understand what the image contains.
- Images that provide a caption are enclosed with a <figure> and <figcaption> tag.
- Ordered lists and their items use the <ol> and <li> tags.
- The progress bar shown during a workout uses the <progress> tag.

These tags also help us to create a meaningful structure of the page. Where the <h1> tag comes before a <h2> tag and the <header> before the <main> tag. The page's structure helps users with screen readers as the screen readers read from top to bottom. A good structure will therefore provide a logical order of items read out to the user.
To ensure our HTML and CSS use the correct tags, we validated both of them. The validator gave no errors.



Providing Meaningful text labels

Throughout the page, we have provided labels for elements that need it. Then, we made sure that these labels got meaningful content. We have attempted to make a label's content describe their functionality – even without any surrounding context. In addition to this, the labels in the <form> can be clicked to alter their matching <input> element. This feature provides us with a bigger hit target for the input as a bonus.

Text alternatives to images

Because a blind user cannot see the images presented on our page, we must provide an alternative text. Therefore, we have given every image an <alt> attribute that can be read aloud to the user or displayed on the page if the image fails to load. Our alternative text directly describes the associated image. For example, we gave the "jumping Jacks" illustration the alternative text "Person demonstrating jumping jacks."
Behind the scenes, there is an image of the next exercise that is not shown to the user but only used to make the page work. We gave this image an empty alt attribute. An empty alt attribute makes the screen reader recognize the picture, but it does not attempt to describe it

to the user. This technique also prevents empty alt attributes errors in the accessibility validation tools (Mozilla, 2020a).

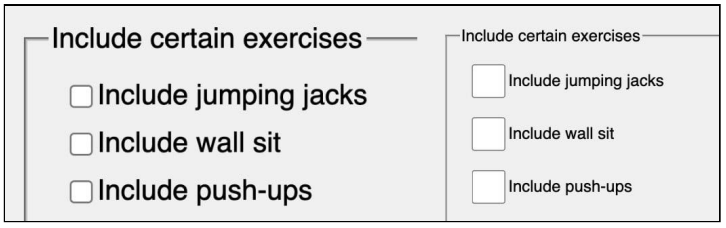
Color Contrast

All text present on the page has at least a contrast of 4.5:1, satisfying the WCAG 2.0 level AA requirement. Our most used color combination (foreground #8E6709 on background #F0F0F0) provides a 4.5:1 contrast ratio. We use this color combination on the headings, figure caption, and buttons. The contrast is also enough to satisfy the 3:1 contrast for graphical objects and user interface components. (World Wide Web Consortium, 2018a)

Large enough touch targets

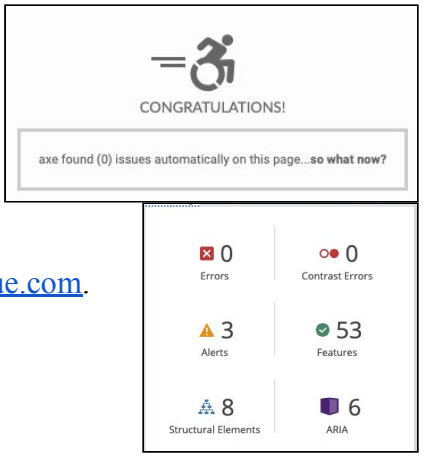
The design of the webpage adapts according to the primary input of the device. On a device with a mouse, the checkboxes are a little smaller than on a device with a touch screen. When the page is displayed on a device with a touch screen, touch targets get bigger. The bigger touch targets benefit multiple users. W3 lists some of the users: "Users with mobility impairments such as hand tremors, users who have difficulty with fine motor movements, users who access a device using one hand, users with large fingers, or who are operating the device with only a part of their finger or knuckle." (World Wide Web Consortium, 2018c)

Our touch targets (including buttons) receive a minimum size of 44x44 CSS pixels. This size satisfies WCAG success criterion 2.5.5.



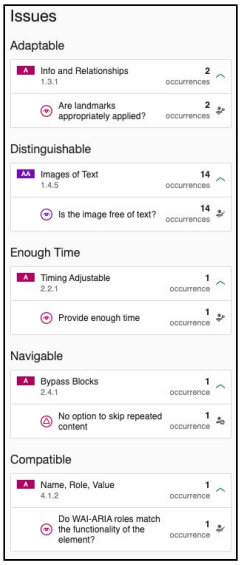
Accessibility validators

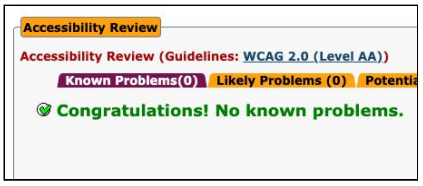
There are multiple websites and web extensions that allow us to validate the accessibility of our webpage. We have used a handful of these to ensure we do not miss any accessibility flaws. The tools we used include



- "axe - Web Accessibility Testing" by www.deque.com.
  - Result: no problems found.
- "WAVE Evaluation Tool" by WebAIM.
  - Result: 3 alerts.

        - "Orphaned form label."
        - "Possible heading." (x2)
- "Siteimprove Accessibility Checker" by Siteimprove.
    - Result: warnings that always will appear since a manual review is needed.
        - "Are landmarks appropriately applied?"
        - "Is the image free of text?"
        - "Provide enough time."
        - "No option to skip repeated content."
        - "Do WAI-ARIA roles match the functionality of the element?"



- "AChecker" by www.achecker.ca.
    - Result: no problems found.
    - 

What we do with alerts and warnings

We go through every warning and alert, making sure they will not affect the accessibility of the web page. Here are some of the decisions we made for the warnings and alerts above.

Orphaned form label

This alert shows up in the WAVE extension when it comes over the label for the progress tag. We have chosen to ignore this alert as W3Schools says: "Tip: Always add the `<label>` tag for best accessibility practices!" (W3Schools). Mozilla also demonstrates the progress element with an associated `<label>` tag. (Mozilla, 2020b)

Possible heading

This alert shows up in the WAVE extension. It gives a warning because it sees a `<p>` element that looks like a heading. We get this error because the `<p>` tag contains fewer than 50 characters and is 20 pixels or bigger *or* 16 pixels or bigger and bold.

We have chosen to ignore this alert because the relevant `<p>` tag is generated using JavaScript and does not contain content all the time. If we would change the <p> tag to a `<h1>`, `<h2>`, or `<h3>` tag we would get an error notifying us of an empty header instead.

Are landmarks appropriately applied?

This alert shows up automatically will always appear since a manual review is needed. We have used the `<header>` and `<main>` tag to appropriately divide the page's content. We can therefore ignore this warning.

Is the image free of text?

This alert shows up automatically will always appear since a manual review is needed. It refers to the images displaying the workouts. These images do not contain text, and we can therefore also ignore this alert.

Provide enough time

Again, this error shows up automatically. It warns us that if our page has some timed events or time limits, the user must be allowed to stop, adjust, or extend it. Our page's only timed event is when a workout changes to the next one after 30 seconds. We have therefore added a pause functionality that allows the user to stop the timed event. When the user presses the pause button, the page will not move on to the next exercise.

No option to skip repeated content

We receive this alert because there is no "skip to content" link on the page. We have chosen to ignore this alert because the alert is aimed at websites with multiple pages. "A means of bypassing blocks of content that are *repeated on multiple web pages* must be provided."

Do WAI-ARIA roles match the functionality of the element?

This warning will always appear for all WAI-ARIA roles since a manual review is needed. In our case, it is reporting the "role = status" attribute we have on the text indicating which exercise the user is currently on. We have added this attribute as this text provides information on a process's progress, following the WCAG guidelines. (World Wide Web Consortium, 2018b)

Testing with screen readers

We also tested our webpage with Windows 10 Narrator on desktop and Apple VoiceOver on macOS and iOS. To keep the testing fair and realistic, we carried out the same test as the usability test. ("open the page, exclude arms, choose the lunge, running and plank workout, start the workout, then lastly pause the workout after completing the running workout."). We enabled the screen reader in our system settings and performed the test once with our eyes

open. Then again, but with our eyes closed, we could perform these tasks without any problems after getting used to how a screen reader works.

Conclusion

We have created a highly accessible web application for people who have a disability but want to get healthier by working out. Users will meet a comfortable design throughout the web application that satisfies all our personas useability needs. Smooth and accessible functionality that will make the journey feel easy. The website contains the functionality and the design to get any user in shape.

Highlights

- Effectively removed the error possibilities. For example, the possibility of adding an exercise that is supposed to be excluded by the "exclude body parts" field.
- Implemented an actual voice that reads out the different exercises so that users will be updated by sound.
- A recognizable and minimalistic design that is consistent in terms of current design trends.
- Creating essential functionalities and making them accessible for users with a disability.
- Responsive design. Web Application looks good from multiple devices with different screen ratios, including laptops, tablets, and mobile.
- We were prioritizing accessibility and useability over aesthetic design. Functionalities are made accessible. For example: in one of our early versions, users could click on the exercise's image to remove it. Since it was not accessible by blind users who cannot use the mouse to click, we removed it and made checkboxes so that they could be removed.
- The main part of the website is a single page application. However, we added an extra page to read about the different exercises before doing them. We found it more reasonable to add a new page rather than implementing it on the front page.
- Our web application is made for those given personas. Users with the same disabilities will have little to no struggle at all using this web application.
- No significant errors from any of the validators.

Possible future enhancements

- Countdown or timer that shows in seconds how long into the exercise the user.
- A smoother progress bar that slowly fills based on how far into the exercise the user is.
- “Previous exercise” and “skip exercise” buttons to let the user skip the current exercise or go back to the previous one.
- Save workouts so the user can access them at a later time.
- Choose each exercise’s length and pauses
    - We had this function for the pauses but removed it since the task specifically said 5 seconds break.
        - We also removed a third fieldset containing extra pauses due to this.
- Let the user choose from a range of workouts that are predefined workouts that target a particular body part or is a good workout overall.
- Let the user add more exercises mid-session, in case the user forgot to add an exercise.
- A button that takes the user to the main menu without losing chosen exercises.

Sources

FitnessYard (2020) https://en.fitnessyard.com/fitness/workout-plan-builder (retrieved: 29.11.2020)

freetrainers (2020) https://www.freetrainers.com/exercise/config/ft2/workout_home/ (retrieved 29.11.2020)

Mozilla (2020a) *HTML: A good basis for accessibility.* Available at https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML (retrieved: 29.11.2020).

Mozilla (2020b) *<progress>: The Progress Indicator element.* Available at https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress (retrieved: 29.11.2020).

W3Schools (unknown date) *HTML <progress> Tag.* Available at https://www.w3schools.com/tags/tag_progress.asp (retrieved: 29.11.2020).

World Wide Web Consortium (2018a) *Understanding Success Criterion 1.4.3: Contrast (Minimum)* Available at https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html (retrieved: 29.11.2020).

World Wide Web Consortium (2018b) *Web Content Accessibility Guidelines (WCAG) 2.1* Available at https://www.w3.org/TR/WCAG21/#dfn-status-messages (retrieved: 30.11.2020)

World Wide Web Consortium (2018c) *Understanding Success Criterion 2.5.5: Target Size* Available at https://www.w3.org/WAI/WCAG21/Understanding/target-size.html (retrieved: 30.11.2020)

Tools

Achecker. Available at: https://achecker.ca/checker/index.php.

Apple VoiceOver (both macOS and iOS).

axe™ - The Standard in Accessibility Testing. Available at: https://www.deque.com/axe/.

Markup Validation Service. Available at: https://validator.w3.org/.

Microsoft Narrator.

Siteimprove Accessibility Checker. Available at: https://siteimprove.com/nb-no/core-platform/integrations/browser-extensions/.

The W3C CSS Validation Service. Available at:https://jigsaw.w3.org/css-validator/validator

WAVE Browser Extensions. Available at: https://wave.webaim.org/extension/.