Accessibility Research Dev Notes

Design

- Colour scheme: dark blue (#395085), light blue (#7b94bf), and a lighter blue for hover and selected states (#b3c6e6).

- Font-Family: Arial, Helvetica, sans-serif.

- All caps text.

- Rounded corners.

- Blue striped background.

- “Accessible Media Player” title at top of page.

- White background container below that contains the nav and all media content.

- 3 nav items: Movie, TV, and Music.

- Content from top down after being selected from nav: video / cover art, media name, timestamp, all media controls, transcript button, transcript (when opened).

Goal Summary

- Create a media player that is accessible to everyone. Incorporate accessibility controls for multimedia to allow those who are disabled or differently abled to consume the media in a way that is suitable for their needs.

Accessibility Implementations to Include

- Closed captioning on videos for the hearing impaired and users in noisy environments.

- Description tracks on videos to provide audio descriptions to the visually impaired.

- Transcripts to videos and lyrics to music for hearing impaired, users in noisy environments, and non-native English speakers.

- Custom media controls to allow users with limited mobility the ability to use the keyboard to fully control the media.

public / audio Folder

- Contains all audio files. In this case, the MP3 file for the song tab.

public / captions Folder

- Contains WEBVTT files that provide the videos with the information needed to show closed captioning (i.e. timestamps and captions).

public / css Folder

- Contains all of the CSS files that style the page.

- Compiled with SASS.

public / descriptions Folder

- Contains WEBVTT files that provide the videos with descriptions (i.e. timestamps and descriptions) of what is happening in the video so that screen readers can provide this information to visually impaired users.

public / images Folder

- Contains all images used in the project. In this case, the song cover art and the favicon.

public / videos Folder

- Contains all videos used in the project.

public / js / main.js

- showMedia(media) displays the selected media. It first hides any shown media, then hides all controls from media, followed by displaying the selected media and hiding the closed caption button for music.

- getMedia(event) determines which route should be shown. When a nav item is clicked, the URL is set to equal /media/{id of clicked nav item}. It then fetches the URL, creating a JSON result and sending the data to showMedia(media). Finally, it changes the classes of the nav items to highlight the selected item.

- playPause() will play the media and change the button text to 'Play' if the media is paused, and vice versa.

- stop() pauses the media, rewinds it back to the beginning, and sets the play/pause button text to 'Play'.

- rewind() rewinds the media by 5 seconds. It will not rewind past 0.

- fastForward() fast-forwards the media by 5 seconds. If the fast-forward exceeds the runtime of the media, it pauses the media, rewinds it back to the beginning, and sets the play/pause button text to 'Play'.

- volumeDown() turns the volume of the media down by 5%. Does not allow volume to go below 0.

- volumeUp() turns the volume of the media up by 5%. Does not allow volume to go above 1.

- captionToggle() turns the closed captioning on if turned off and vice versa. It also sets the opacity of the button to 50% if turned off, and 100% if turned on.

- transcriptToggle() displays the transcript or lyrics if hidden and vice versa. Changes the button to say ‘Show’ when transcript is hidden and ‘Hide’ when transcript is shown.

- updateTime() displays the current time that the playing media is at. It first sets and formats the values for minutes and seconds, then it combines them and updates the HTML text to read the current time.

routes / index.js

- Defines the different routes that a user can reach.

- ‘ / ’ is the default route, the home page. This route will display the title and the nav.

- ‘ /media/:id ‘ is the selected media route. This route will dynamically display the content for the selected media.

sass Folder

- Contains all of the files that compile to CSS to style the page.

utils / sql.js

- Uses the information provided in config.js to create a connection to the database.

views / error.hbs

- Dynamically displays error information if the error page is reached.

views / errorPage.hbs

- Creates the shell for the error page.

views / home.hbs

- Dynamically displays the content for the selected media.

- {{{media_code}}} fetches the video or audio code from the database and inserts it in the file.

- {{media_name}} fetches the name of the media and displays it on the page.

- ‘controls’ contains all of the media controls.

- ‘transcriptArea’ contains the transcript button (which shows or hides the transcript on click) and the transcript itself, which is code that is fetched from the database and inserted into the file.

views / layout.hbs

- Creates the shell for the page.

- Contains the title and the nav items which determine what content will be displayed when clicked.

- home.hbs is inserted into {{{body}}} below.

.gitignore

- Specifies all of the folders and file types that Git should ignore when the project is pushed.

app.js

- Sets up the port connection, routes, and sets Handlebars as the view engine.

config.js

- Defines the information needed to connect to the database.

gulpfile.js

- Sets up Gulp for compiling SASS.

package-lock.json and package.json

- Contains all of the information pertaining to Node and installed packages.

Research Sources

- The following sources were used as references or tutorials for functions used in main.js. Functions may be similar in some instances but have been changed to work with my code (namely, playPause(), stopMedia(), rewind(), fastForward(), and updateTime()).

- https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Multimedia

- https://github.com/mdn/learning-area/blob/master/accessibility/multimedia/audio-transcript-ui/main.js

- https://mdn.github.io/learning-area/accessibility/multimedia/audio-transcript-ui/

- https://developer.mozilla.org/en-US/docs/Web/Guide/Audio_and_video_delivery/Adding_captions_and_subtitles_to_HTML5_video