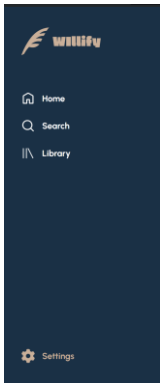
Willify Website Documentation

1. Navbar
   The following navigation bar is made with position fixed so that it doesn't get scrolled when the website is scrolled. Yet, as position fixed makes the layer (z-Index) different from the rest of the website content, so there is a need of adding the property of margin left as big as the navbar's width as it is a left navbar. The navbar will shrink when it is in mobile mode where there will only be icons with no description when it is triggered by the media query at a certain pixel size screen.

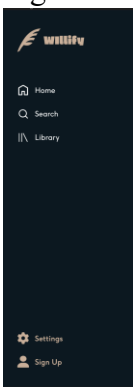
   Website Desktop

   Website Mobile

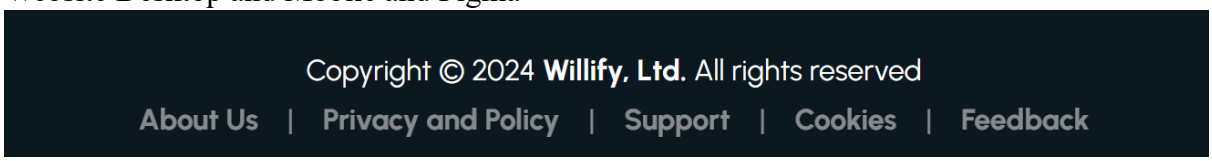
   

   Figma

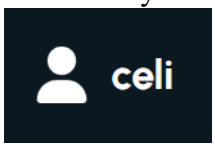
   

2. Footer
   The footer can be found in the end of all pages when scrolled within the <main> semantic tag. It consists of copyright text and navigation to some pages. The only active one is the navigation to the About Us page from the footer. The content in the footer is already capable of wrapping itself to solve responsivity problem, so there is no need to change the footer design through media query anymore.

   Website Desktop and Mobile and Figma

3. Account
   This icon and name is shown when the user has logged in. The name will follow the user's inputted name, where all the pages can access it through the params passed to each page. As Willify is the best streaming platform in the world, we create new things out of the stereotype, to log out from your account just simply triple click the icon or name. There's a hover interaction to help the user know that it is clickable. Click once to go to the settings, but since the project doesn't require me the developer to create the settings page, so there is no ability to click once and be redirected to settings page.

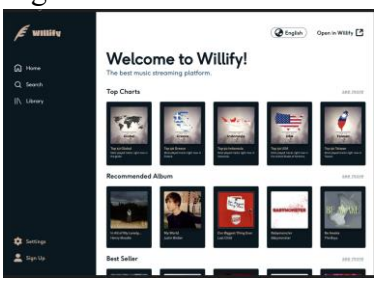
   

4. Home Page

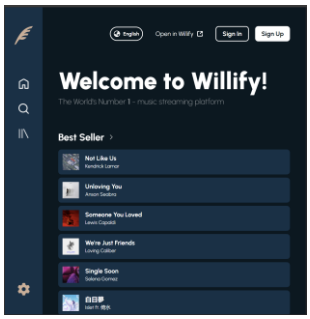
   Home Page is the first page that the user will be redirected to. Here there are navbar, welcoming words with a sentence describing Willify, best-seller section, recommended album section, top charts section, as well as footer. All the data of sections are rendered through javascript with hard code data are in the const.js file. This is done to make the code less repetitive. Here, users can go to the register page by clicking the buttons on the top right of the page. Both the sign in and sign-up button redirect the user to the sign-up page as there is only sign-up page. From the home page, clicking at one of the best-seller section's song tracks will redirect user to the song detail page, where the data will be dynamically presented with the id of the best-seller song track clicked is passed through the params to the song detail page; same thing for recommended section and top charts section that will be redirected to song page with dynamic presentation.

   Website Desktop

   Figma



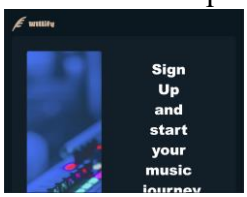
   Website Mobile



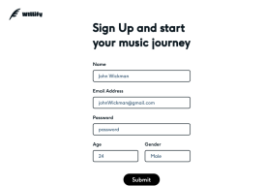
5. Register Page

   The register page consists of 5 inputs name with validation of minimum 3 characters long, email with a standard form of 'xxx@gmail.com', password with two validations minimum of 8 characters long and contains 1 of each uppercase, lowercase, numeric, and symbol, age with validation of age older than 10 years old, and gender displaying a dropdown with no validations but required to be filled. On the other hand, for the figma there is no validation and dropdown. In total, there are 5 inputs and 5 validations and only when all data are appropriate will the user be redirected to the home page, with the name of the user sent as params to the homepage. In the homepage, song page, and song detail page, sign-in sign-up buttons will be changed to account icon with the appropriate name being inputted. As for the responsivity, as there is not much content, only the submit button and age gender, and the image that are changed in the media query.

Website Desktop



Figma

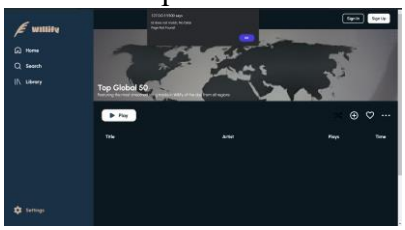


Website Mobile

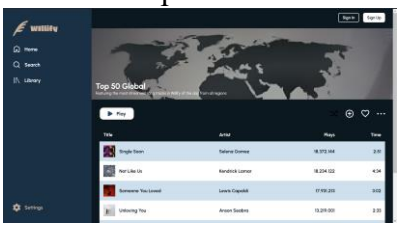
6. Song Page

This page shows the songs tracks that are available from the chart or album clicked. Similar to the homepage, in the web this page can be accessed by guest user leaving the sign-in sign-up button still there and will change to the account icon with the name if the user has registered with the help of username param, as for figma the user needs to sign-up to access the page. In the following design, it is assumed that the user is called celiiiiiii. As the data is dynamically presented with checking the id passed when clicked with the id from const.js. If the id doesn't match the data available in const.js (caused by not creating the data for that page), then there will be an alert that show up representing data is not available and you will be redirected back to home page. On the other hand, when the data is available, a list of independently scrollable song tracks will show up, with the title of the page (i.e., Top 50 Global, etc.) changed dynamically as well. All the list of song tracks can redirect the user to the song detail page where only there they will be able to play and see the lyrics of the song track they chose. This page is also responsive by shrinking the navbar and removing the other three columns in the song track list, which is made with display grid, leaving only the title column left to save more space.

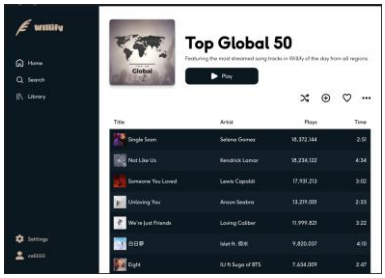
Web Desktop No Data



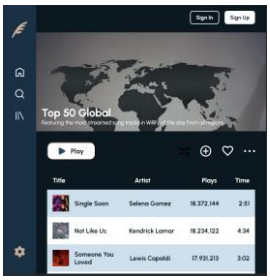
Web Desktop with Data



Figma



Website Mobile



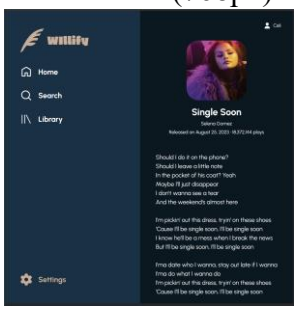
7. Song Detail Page

   In this page, the design in made with html and css at first, yet there is no data at all. Only dummy data are there at first, as all the essential data from title, artist, release date, number of plays, lyrics, to the width of progress bar for the song time, which is calculated in javascript, and the width of progress bar for the sound which is randomized. Similar to the song page, there will be username and id param that is sent to this page and checked with the data in const.js to dynamically present the appropriate data. Yet, this page is the only page among the five pages present in the web, where there is a need to register first to access the page and the name of the user is shown in the web shown in the top right corner. If the user hasn't signed up, an alert will show up and redirect them to the register page. As for figma, it remains the same as the song page. This page is also responsive with lots of changes from song general data and buttons in the left side put to position fixed so that it will not get scrolled when lyrics are scrolled, then creating a lengthy design with all in a block type arrangement with the help of display flex setting its flex direction to column. This change is achieved with the help of media query properties set in several different pixel screens.

Web Desktop

Figma

Web Mobile (768px)



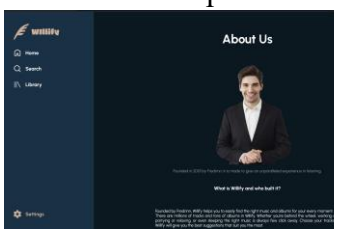
Web Mobile (320px)

8. About Us Page

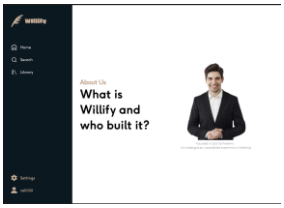
   The About Us page consist of the founder personal data and what Willify is arranged in and orderly matter where everything is within a block display with the help of display flex

setting its flex direction to column. Here, similar to homepage both in figma and website the user can access it without the need of registering. Another point, as most elements can be set to flex-wrap or text-wrap, there's not much code in media query to make it responsive. It's just setting the navbar for mobile that is done in the media query to keep the website responsive.

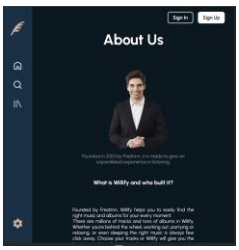
Website Desktop



Figma



Website Mobile



Notes:

- References:
  https://www.youtube.com/watch?v=In0nB0ABaUk
  https://www.youtube.com/watch?v=XF1_MlZ5l6M
- There are no copy pastes from those video, those are just references that are used for my knowledge to implement the knowledge and get to know the new javascript syntax
- Assets are grabbed for several different sources, all of them are saved in the assets folder and both figma and the web uses the same asset folder
- All 5 pages can be scrolled both in figma and website
- Footer can be found in all pages when scrolled to the end
- About Us page can be navigated through the About Us button located in the Footer of each page
- The logo icon can be used to navigate back to the homepage
- All account names are assumed not dynamically presented
- The website needs to be opened through live server, not through double clicking the html file in file explorer as some pages have data that are dynamically rendered by javascript when first loaded