

Music Player Project



Paul Cheney

SPARTAN DESIGN UNIVERSITY

spartandesignuniversity.com



Agenda



Design/Build Template

Design/Build Album Page

Design/Build Artist Page

Design/Build Player Page



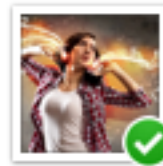
Demos Folder



a2h_Music.psd



Player_albums.psd



Player_artists.psd



snippets.txt



start



a2h_Music-assets



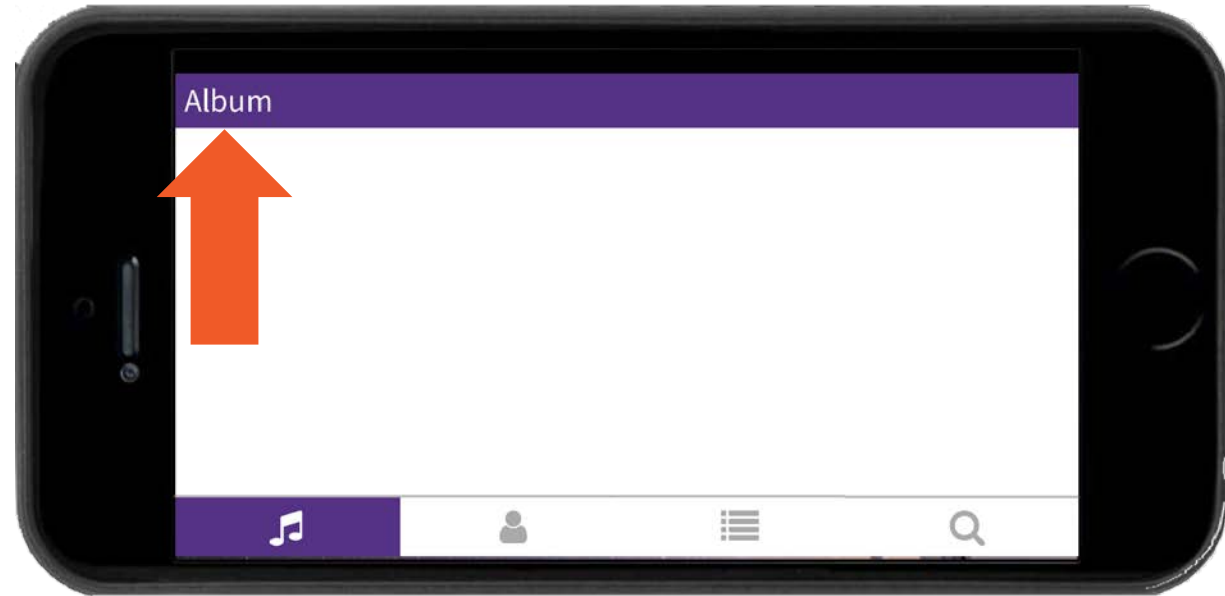
Player_albums-
assets



Player_artists-
assets



Design Template



Font Awesome

Flexbox

Fixed Position



Here are the changes I need you to make in the HTML of the start file

On line 9 change the title to "Album"

Right after that, add a link to the Google font family Source Sans Pro with the font weights 400 and 700

Next a link to Font Awesome's JavaScript file on their server. These links can also be found in the snippets in the demos of this unit.

Between the opening and closing header tags, add an h1 tag around Album

Next let's cut the page navigation comment and nav tags and paste them below the closing main tag

We will not be needing the footer tags so we can delete that section along with its comment

We will build our navigation using 4 divisions

Each division contains an anchor to link to each of the pages in our music player.

The first link is to index, the second to artist, the third to song and the last one to search.

Between the opening and closing anchors, we will add a Font Awesome link to

Font Awesome music

Font Awesome user

Font Awesome list and

Font Awesome search

Now we add a class of active to the first link

Finally we need to add a division with a class of wrapper around header, main and nav.

This is what it currently looks like on our phone simulator.



Now it's time for the SASS

Create a variable for primary color and use 553285

Create a variable for secondary color and use 36175E

Create a variable for ltgrey and use aaa

Create a variable for dkgrey and use 666

Create a variable for text and use the variable secondary

Create a variable for defaultFont and use Sans Source Pro and sans-serif as the backup

Create a variable for normal and use 400 weight

Create a variable for bold and use 700 weight

Finally Create a variable for gutter and use 2%

Now we can open the small sass file

In the body set the font-size to 16,
the font-family to the variable defaultFont
the font weight to the variable normal
the color to the variable text
and the background-color to white.

Setup a section for the id wrapper and
set the position to relative
the width to 100 view height and the
height to 100 view height.



Next lets take care of the header and set the background color to the variable primary the text color to white and add some padding of .5rem top and bottom and the variable gutter for the sides. Finally we will increase the font size to 1.5 rem.

This is what it currently looks like on our phone simulator. Lets work on the navigation links

The background color should be white since we want the links to stay at the bottom of the screen at all times, we will use position fixed and bottom zero Lets set the width to full width and separate it from the content with a border that is 1 px thick, solid and using the dark grey variable Finally we will display the nav as a flex item.

Now lets next the 4 divisions inside the nav and assign them a flex-basis of 25% We will then move the icons to the center



Now lets talk to the division anchors and ... display them as blocks
increase the font size to 1.7rem ... add padding of .5rem top and bottom and none at the sides
change the color of the icons to the lite grey variable
Here is what we have so far.

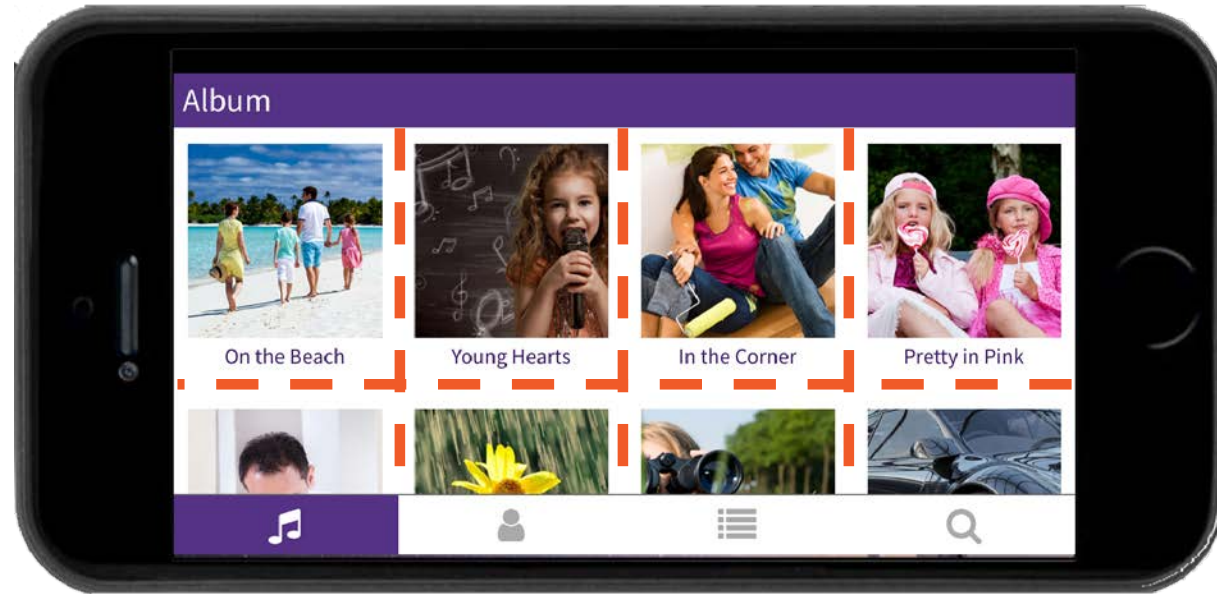
Lets activate the album anchor by talking to the div.active anchor and giving it a background color of the variable primary and setting the color of the icon to white.
This is what it looks like now.



DELETE ME



Album Template



















On the Beach

```
<figure>
  <a href="player.html">
    
  </a>
  <figcaption>On the Beach</figcaption>
</figure>
```



Creating Images

 a_beach200.jpg
 a_beach400.jpg
 a_girl200.jpg
 a_girl400.jpg
 a_paint200.jpg
 a_paint400.jpg
 a_pink200.jpg
 a_pink400.jpg
 a_rain200.jpg
 a_rain400.jpg
 a_sad200.jpg
 a_sad400.jpg
 a_search200.jpg
 a_search400.jpg
 a_wheel200.jpg
 a_wheel400.jpg



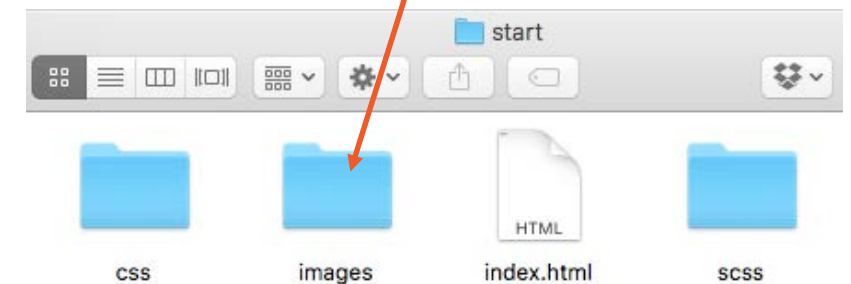
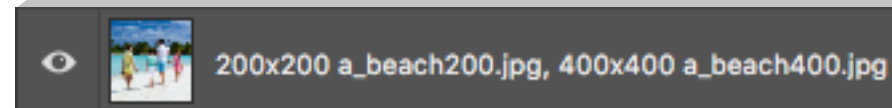
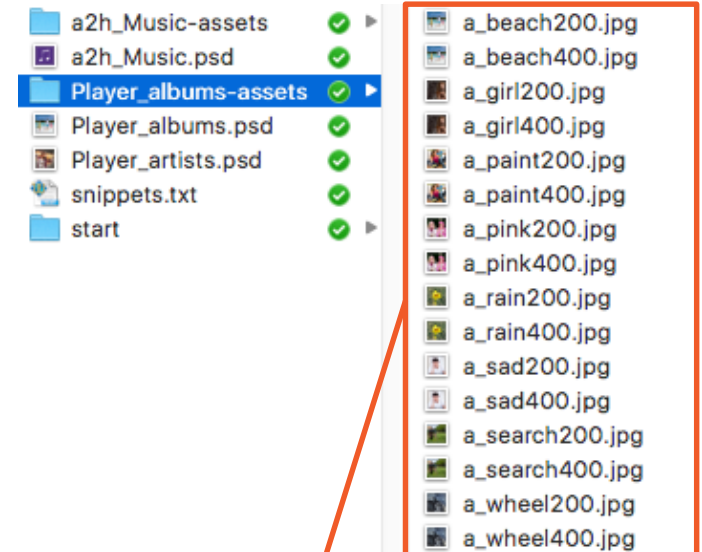
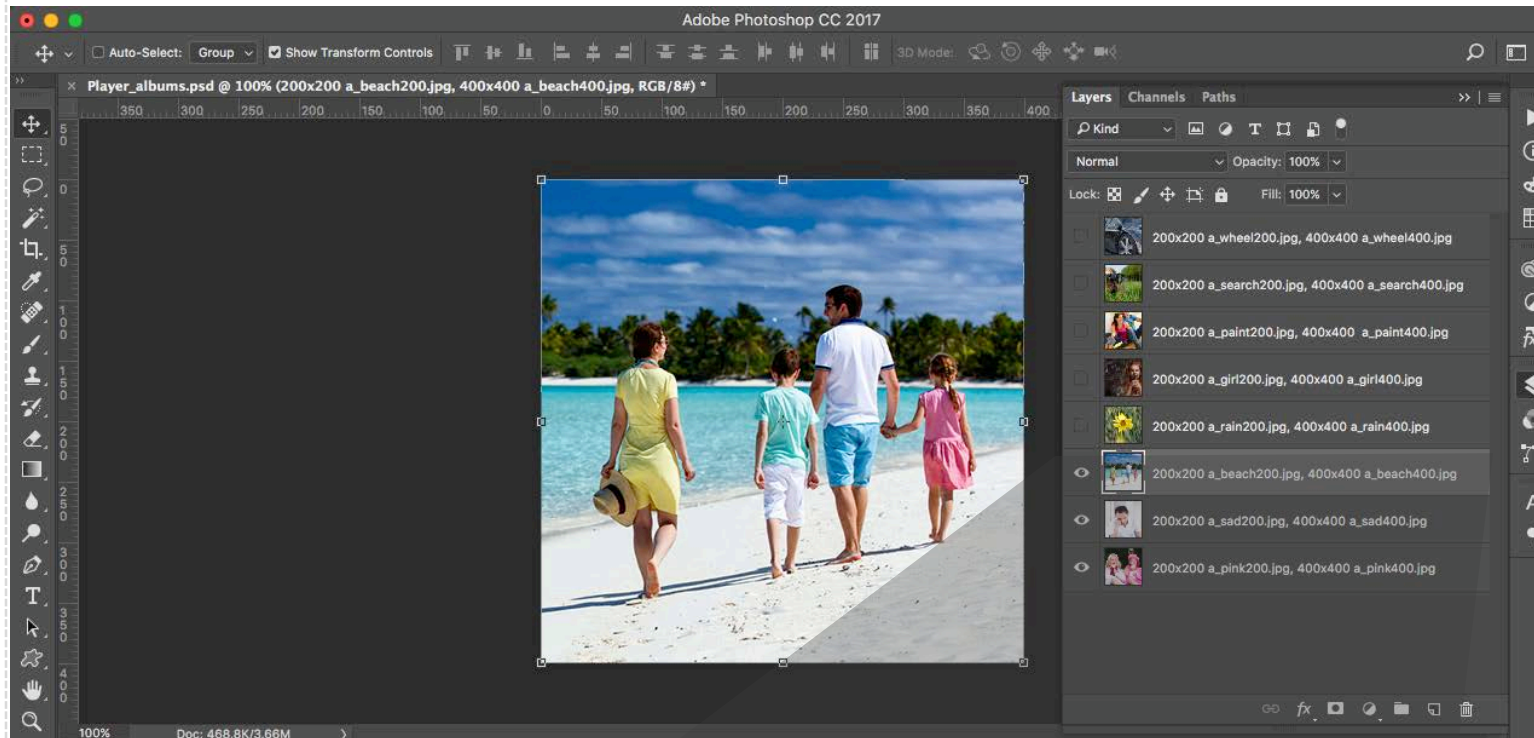
200px x 200px

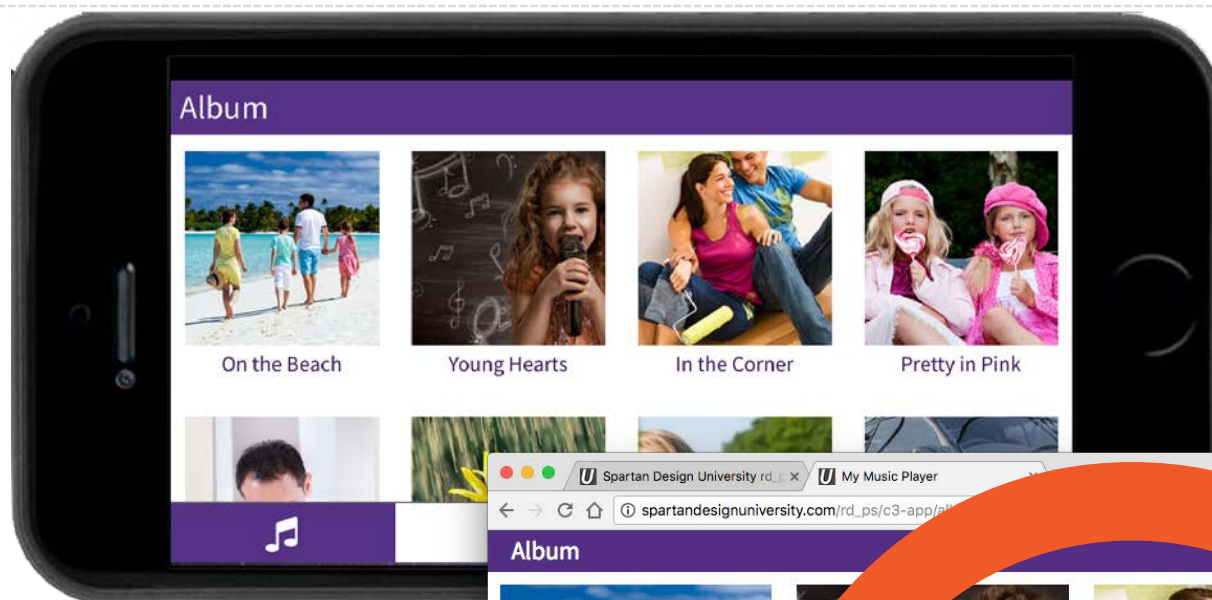
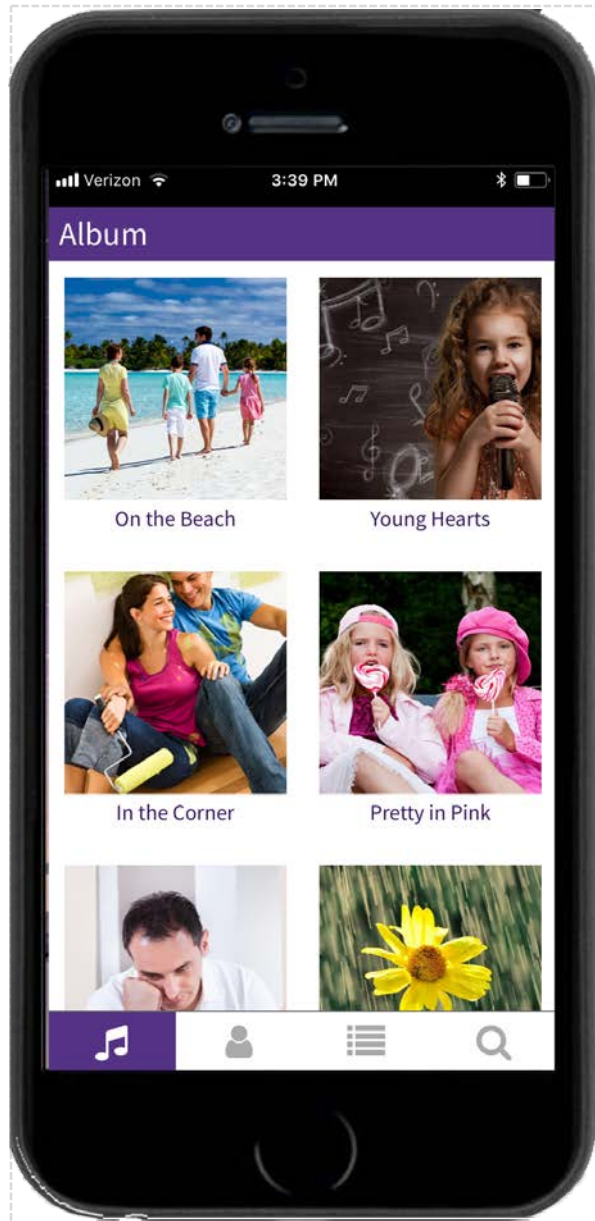


400px x 400px



Photoshop File

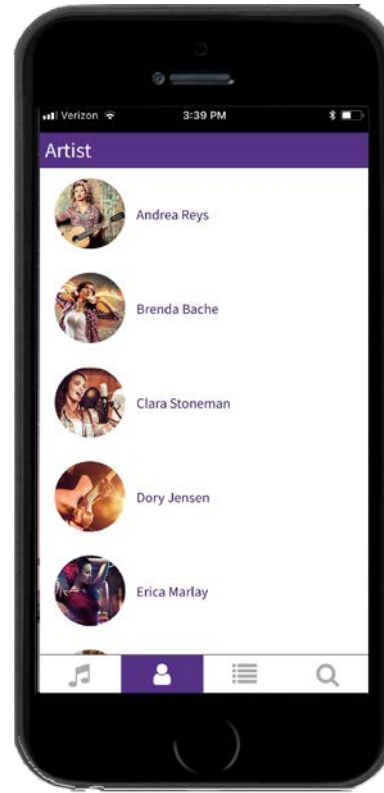




Lets open the `styles.scss` file and remove the reference to large
Now lets change the min width for medium to 29rem or 464 pixels which is a safe
breakpoint for phones between the landscape and portrait orientation.



Separating CSS



```
<!-- page content here -->  
<main class="album">
```

```
<!-- page content here -->  
<main class="artist">
```

```
<!-- page content here -->  
<main class="player">
```



Lets start by adding a class = album to the main tag.

Next we will add an opening and closing figure tag to hold the image and caption.

Since we want these to be clickable to play the song, we will next add an anchor with an href pointing to the player page which we will build later.

Below the closing anchor, add a figcaption that says "on the beach".

Now were ready for the responsive image.

First we add an image tag with the default source set to images beach200.jpg This is for dumb browsers

Now we will use a srcset based on device pixel density of dots of light as we covered in an earlier course.

Inside the quotes type a path to the beach200 image and tag it with a 1x for computer monitors and other 1x displays. Then type a comma and the path to the beach400 image and tag it for retina displays that are 2x or higher. This image will show on any modern phone.



Now lets open the small default file and create a section for main dot album

First we set it up as a grid

and then add two even fractions for the columns

We will let the number of rows be automatic

Now we need to create a nested figure and add some padding.

We will use .75rem for the top and bottom and 2vw for the sides.

We will tell the image in the figure to display at 100% of the figure

We can then style the caption by centering the text and adding some padding around the edges.

Now open the medium file and create a main.album rule. change the grid template columns to 4 equal fractions. That should do it.

Now lets test our page with the chrome tools up and the screen size set to 320 x 500.

Notice in the network tab that the beach200 image is loading correctly.

Now lets change our emulation to a phone and reload the page.

Now we can see the larger beach 400 image is loading correctly on the simulated retina display.



With one of these images in place we can simply copy and paste the other 7 figures and change the image reference or...

you can open the snippets file and copy the code for the remaining 7 images and paste it below the beach image.

Here is what we now have on a phone in portrait mode.

Here is a phone in landscape mode.

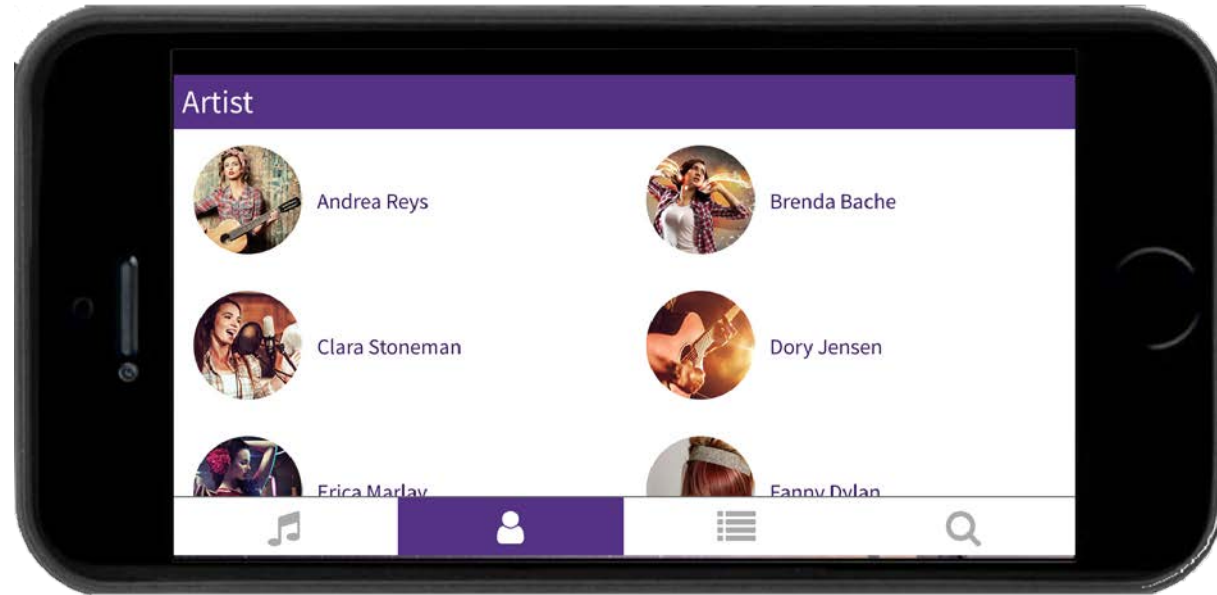
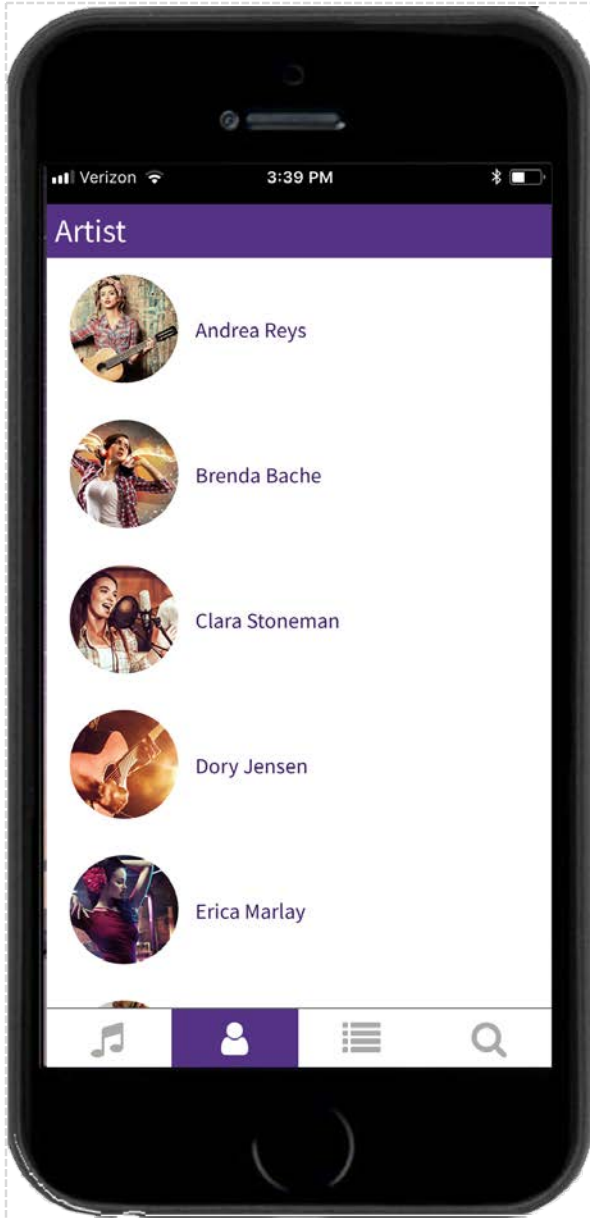
We can also look at the tablet in portrait mode... and landscape mode. Now lets do the artist page.



DELETE ME ARTIST



Artist Template



Grids

Clip Path

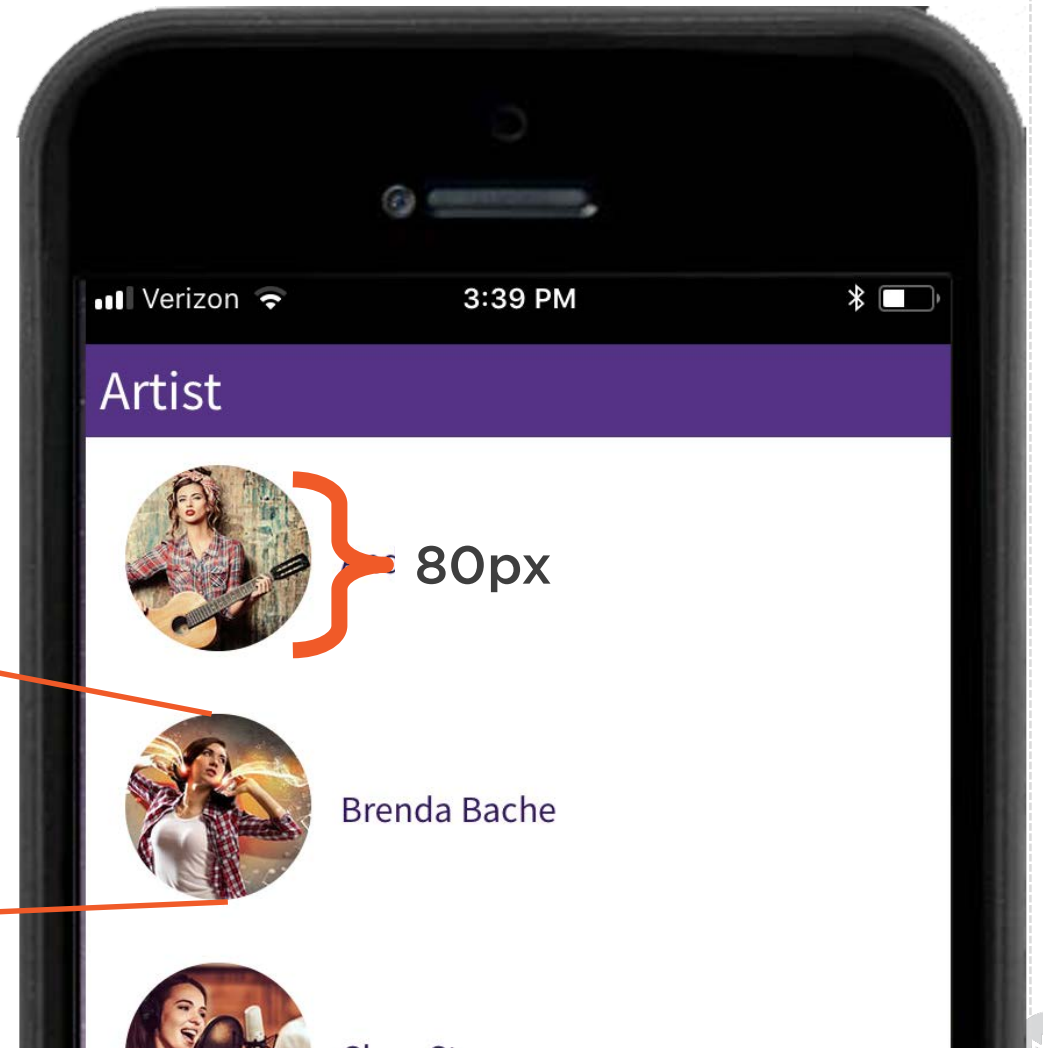
`nth-child()`



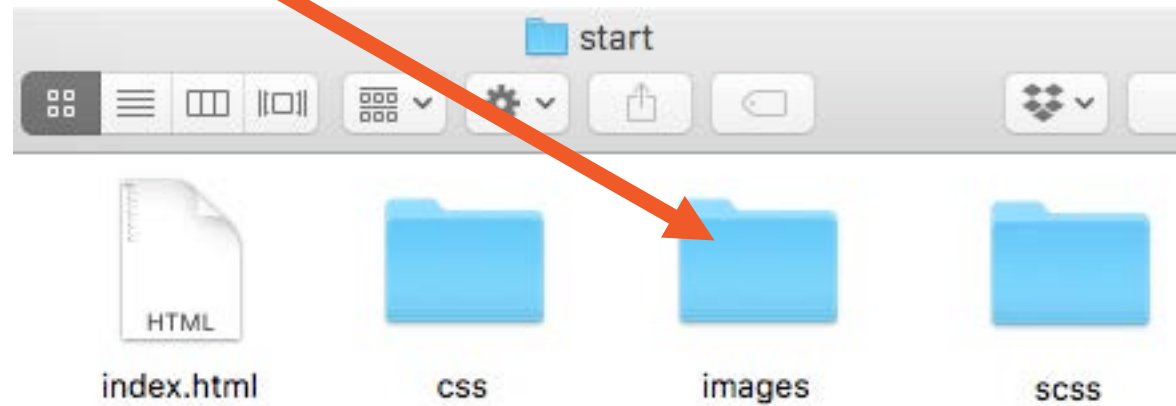
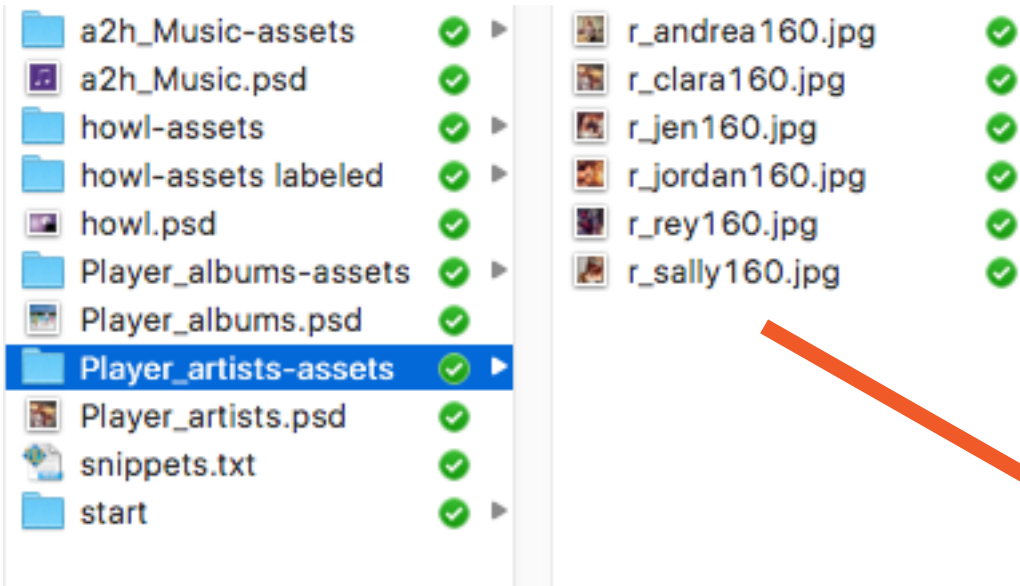
Images



160px x 160px



Images



First we need to duplicate the index page and rename it artist.html

Next we need to delete everything inside the main tag

Now we need to change the class from album to artist.

Lets create an opening and closing section tag for each artist.

Inside the section create a division with an anchor to the player.html file that we still have not built.

Inside the link add an image with a source to the andrea 160 image

Below that add a second open and close division and put Andrea Reys as the name.

This is what it looks like so far in our phone simulator. Since it is working correctly lets do the remaining 5 artists

The code you need is in the snippets file in the demos folder.

Now it looks like this in our phone.



Now lets jump into the CSS.

Remember that our CSS preprocessor will strip out all comments and empty lines so we can put in all the extra stuff we want too organize our code.

Now lets create the main dot artist section with an ending comment

We will display all the children of main in a grid

In the portrait mode, we will have one column and we will setup the rows as automatic.

Now lets talk to the children of main which will be sections

Once again we will use the nesting feature of sass and add a section with open and closing curly. Please put a comment by the closing curly brace.

Lets display each section as a grid inside the main artist grid.

The first grid column will be 80px wide for the image and the second will be auto to fill the rest of the space.

Lets add some margin around each section so the images do not touch each other or the edge of the screen.

You can now see the additional space separating the images.



Now we're going to do double nested for the image inside the section which is inside the main.

Add a width of 100%. This will make the images fit the 80px wide column

Next let's make the square images round by using clip path and a circle of 40 px at the center

We also need to do a webkit version of clip path for Safari.

This is what it looks like on our phone. Notice that the name automatically moves into the second cell in our grid.

Now let's do a double nest for the artist name using nth child and 2

We will use align self to center the name top to bottom and add padding at the left so the name does not touch the photo.

Here is our work on phone in portrait mode



Now lets change to landscape mode and we need to do a little more work.
In the medium file we need to talk to the main dot artist and simply change the number of columns to 2 equal fractions and save our work.

Here is the result in our phone.

Can YOU see the two additional changes we still need to make? Yes, the page title and the active menu item.

Back in the artist page, we will change the header to Artist
in the navigation we will move the class = active to the second division

Here is the final result for the artist page on a phone in portrait orientation

Here is the final result in phone landscape

Here is the results in tablet for portrait as well as landscape modes.

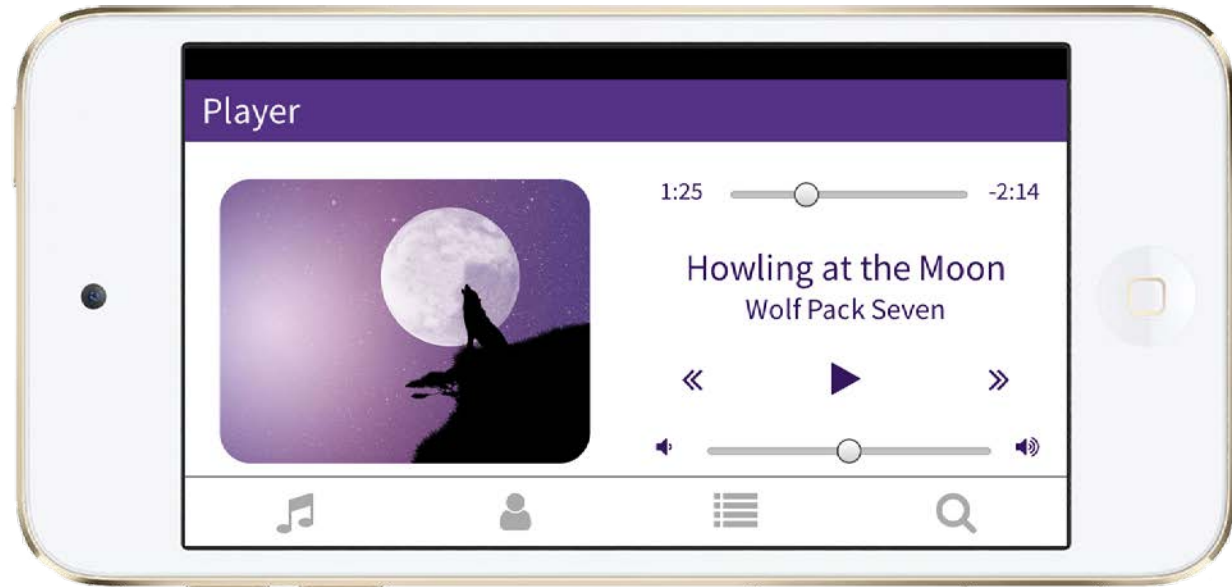
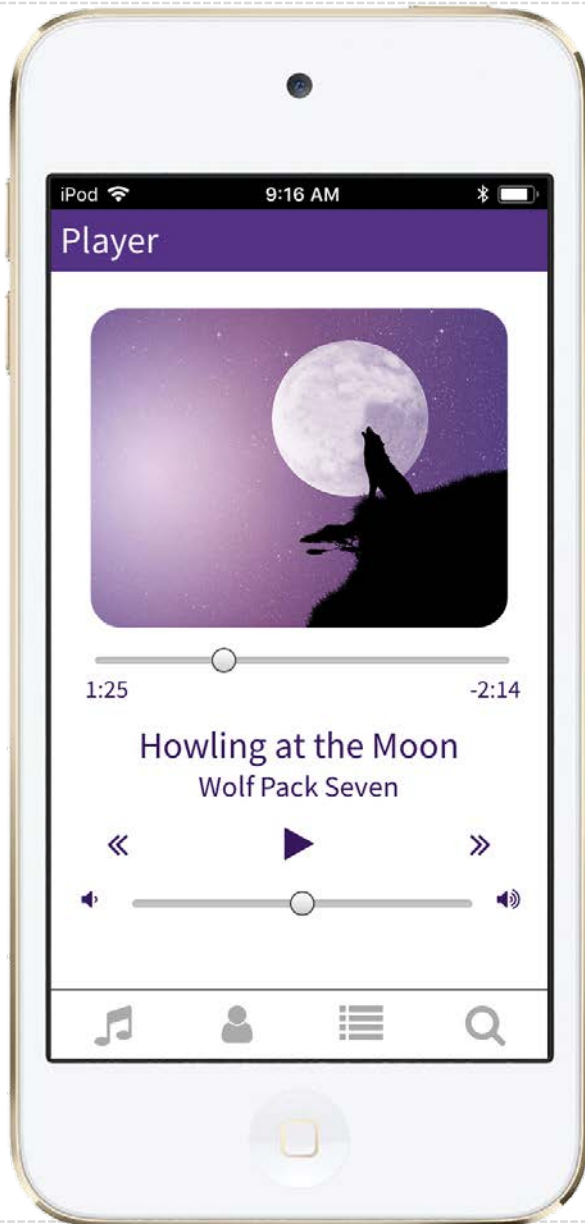


DELETE ME

Player



Player Page



Grids

Areas

Responsive Images

Range

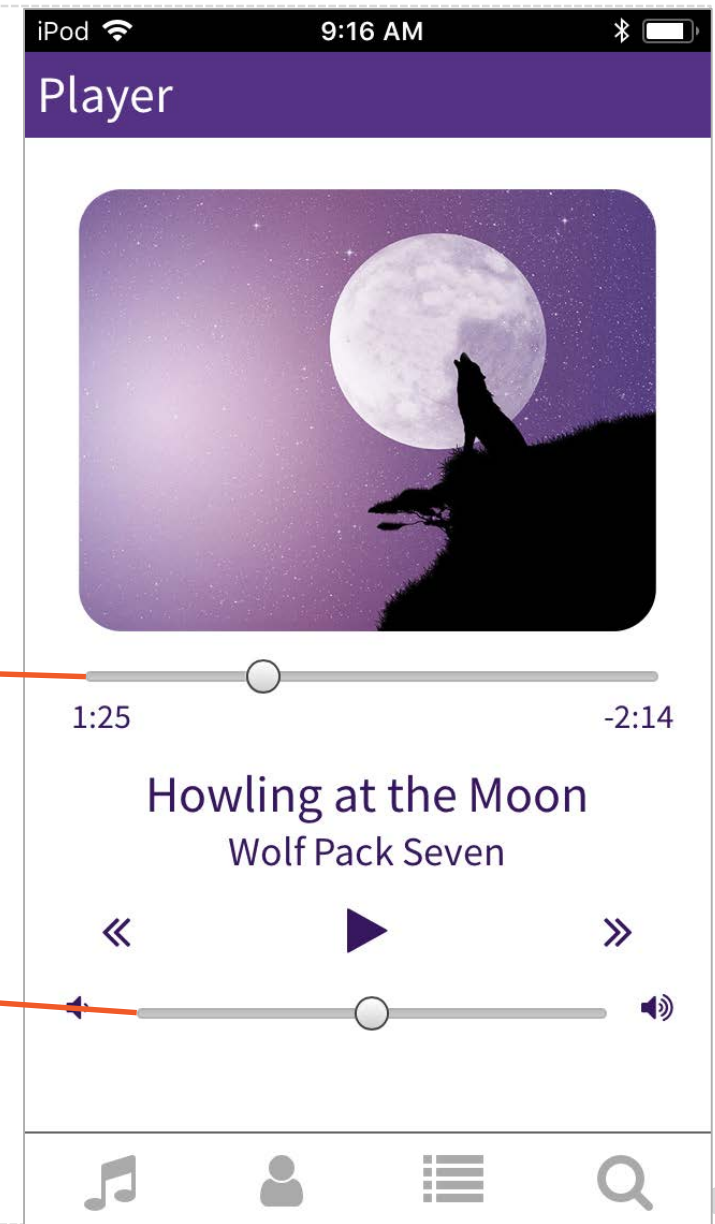
Radius

Font Awesome

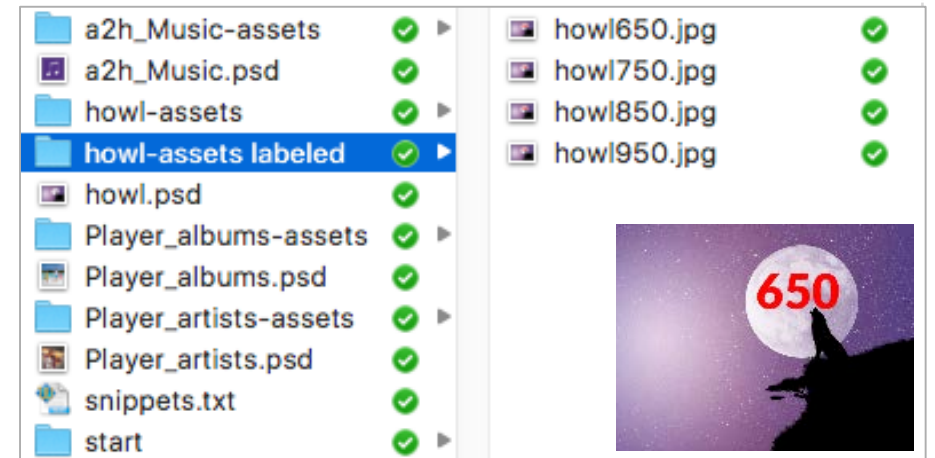
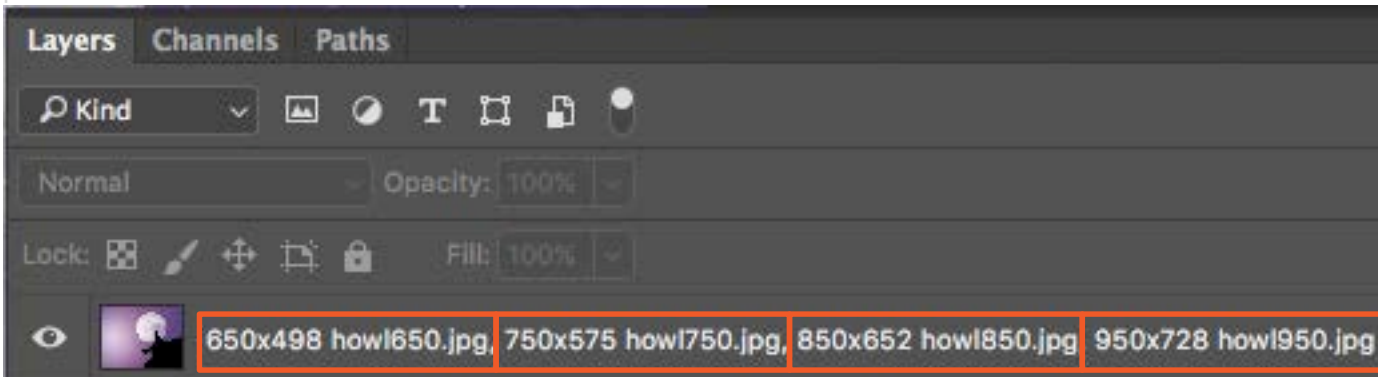
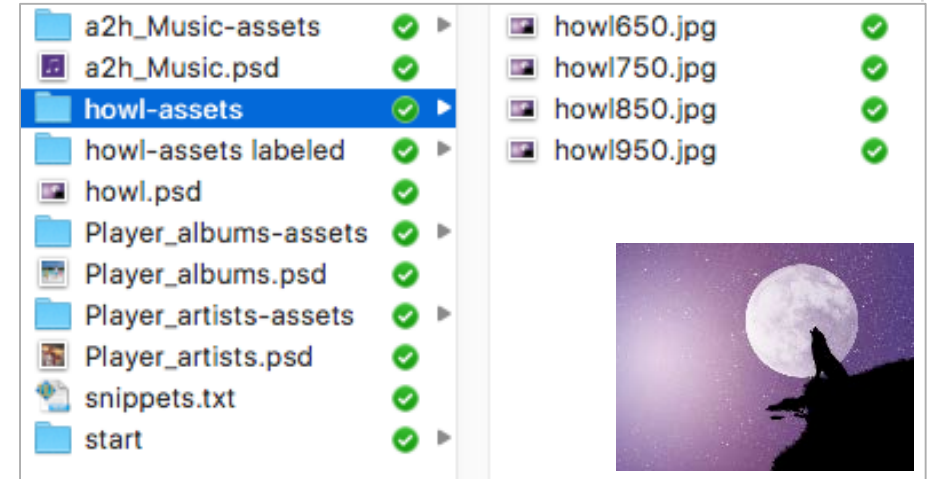
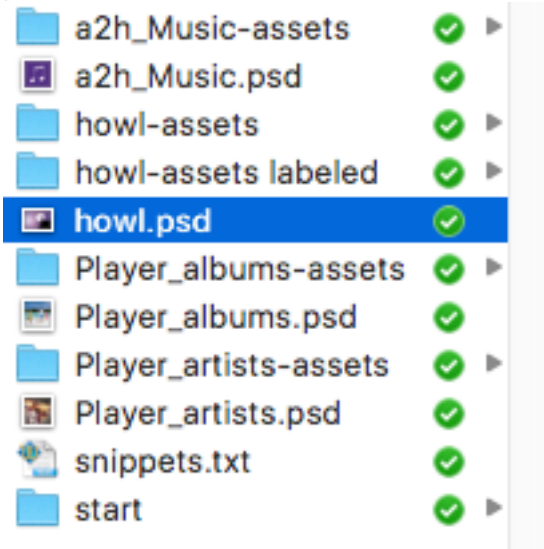




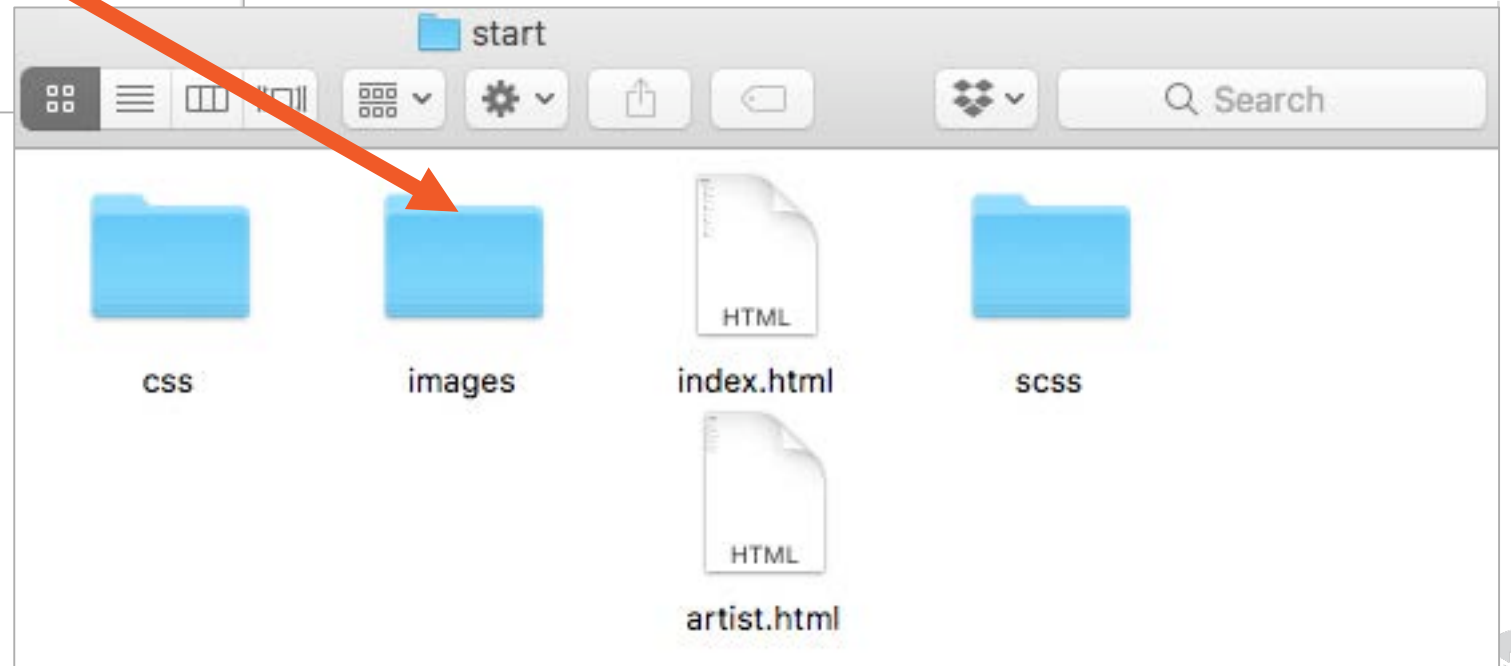
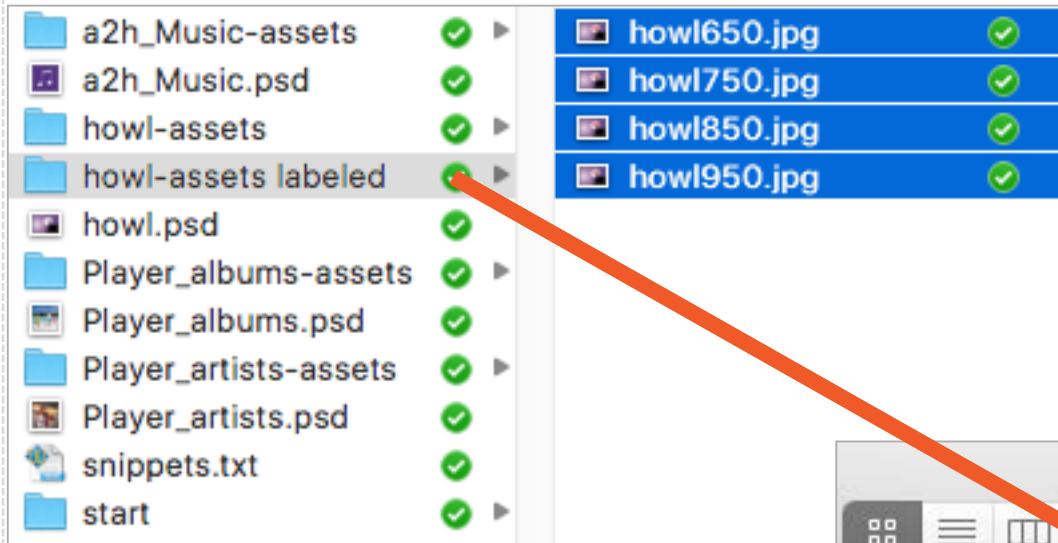
Chrome Emulator



Images



Move the Images

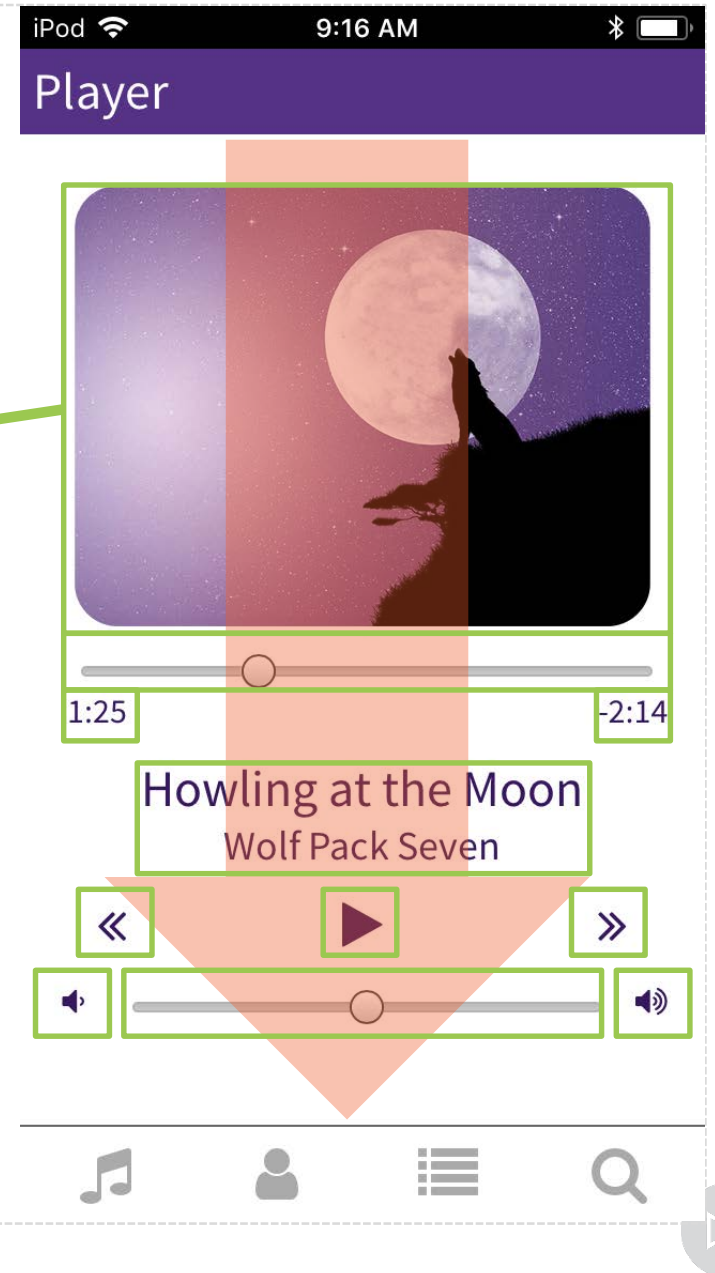


Next duplicate the artist.html file and rename it player.html
Open your text editor and change the header h1 to "player"
Select and Delete all the sections inside the main tag
Remove Class=Active from the second link



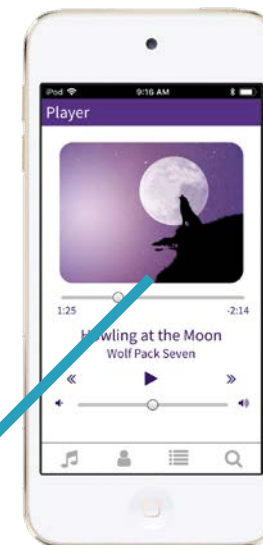
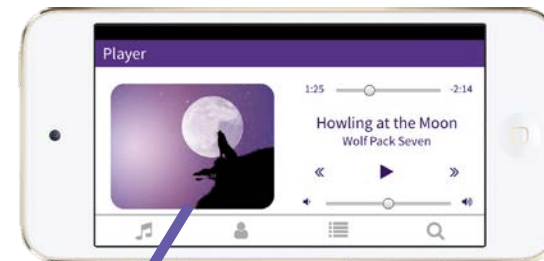
`<div class="startTime">`

`</div>`



Sizes Review

```
1 @import "variables";
2
3 @import "reset";
4
5 @import "small-default";
6
7 ▼ @media only screen and (min-width: 29rem) {
8     @import "medium";
9 } // end media query
```



sizes="(min-width: 29em) 50vw, 100vw "



Source Set Review

Screen Width * Pixel Density * Display Width = Final Size



568px

2x

50%

568 wide



375px

2x

100%

750 wide

srcset="

images/howl650.jpg 650w,

images/howl750.jpg 750w,

images/howl850.jpg 850w,

images/howl950.jpg 950w

"



Our first division will have a class of photo

The image source will point to the howl 650 image in case you have a dumb browser

The sizes and source set we just covered in detail and it should look like this.

Actually I have a copy of all this code in the snippets file for this demo so lets save our selves a lot of time and headache and just copy and paste all eleven divisions into the main tag and we will talk about them.

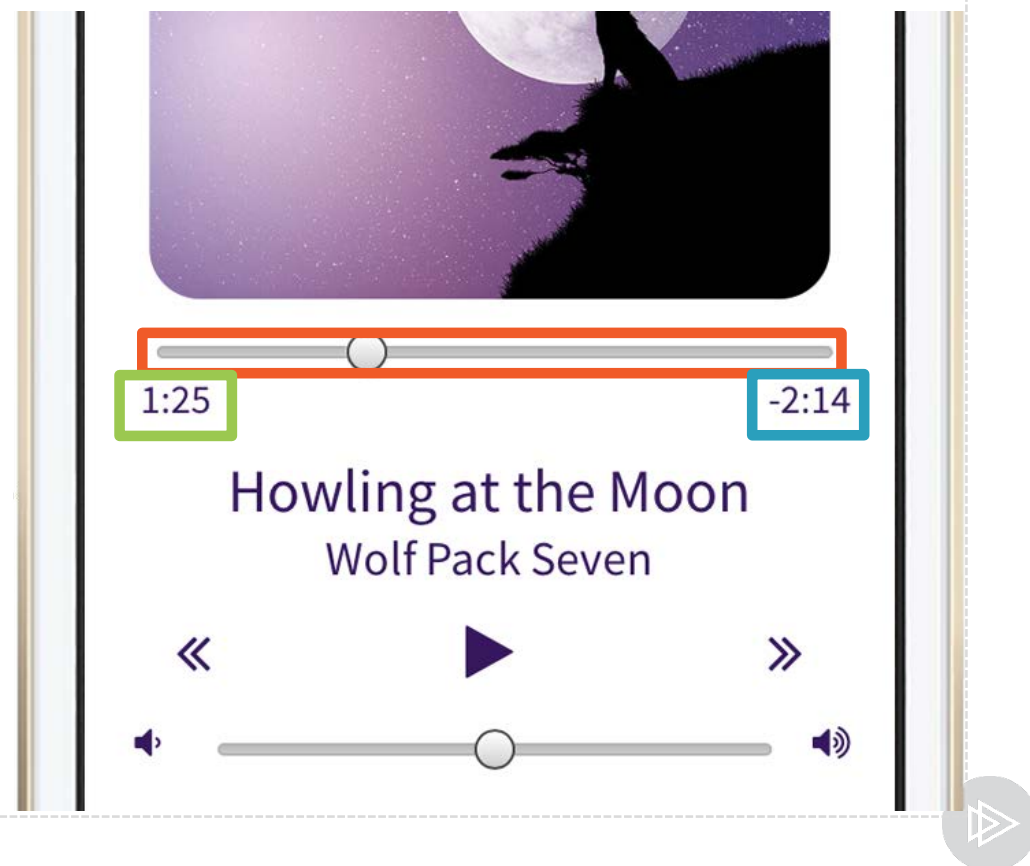


Time

```
<div class="startTime">1:25</div>
```

```
<div class="time"><input type="range" name="volume" min="0" max="100" value="30"></div>
```

```
<div class="endTime">-2:14</div>
```



Song Name and Band

```
<div class="title">  
  <span>Howling at the Moon</span>  
  <span>Wolf Pack Seven</span>  
</div>
```

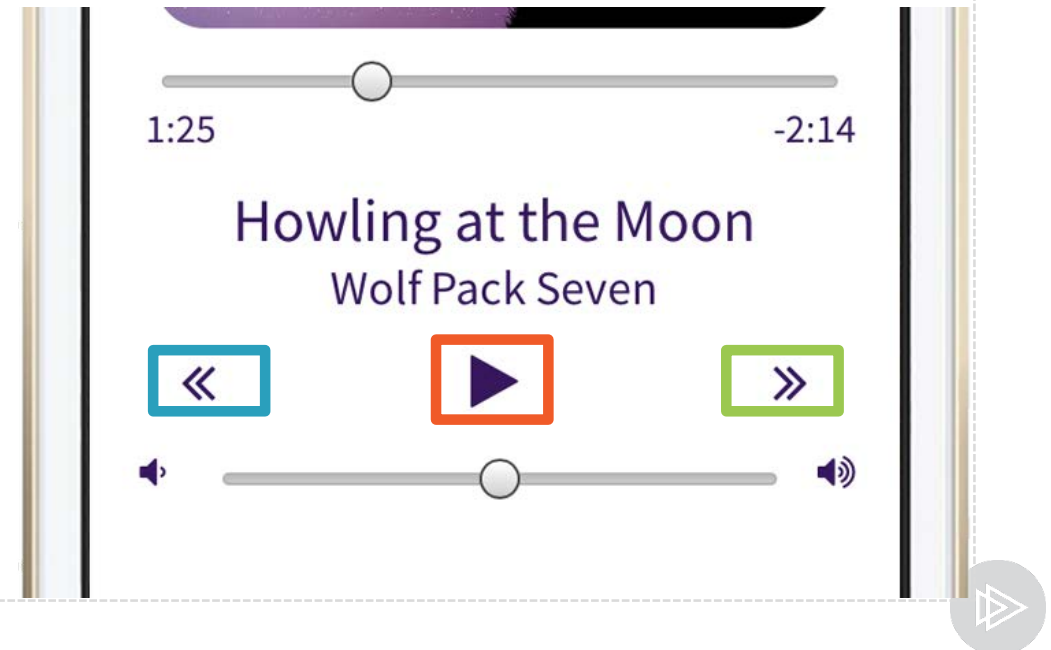


Controls

```
<div class="previous"><span class="fa fa-angle-double-left"></span></div>
```

```
<div class="play"><span class="fa fa-play"></span></div>
```

```
<div class="next"><span class="fa fa-angle-double-right"></span></div>
```

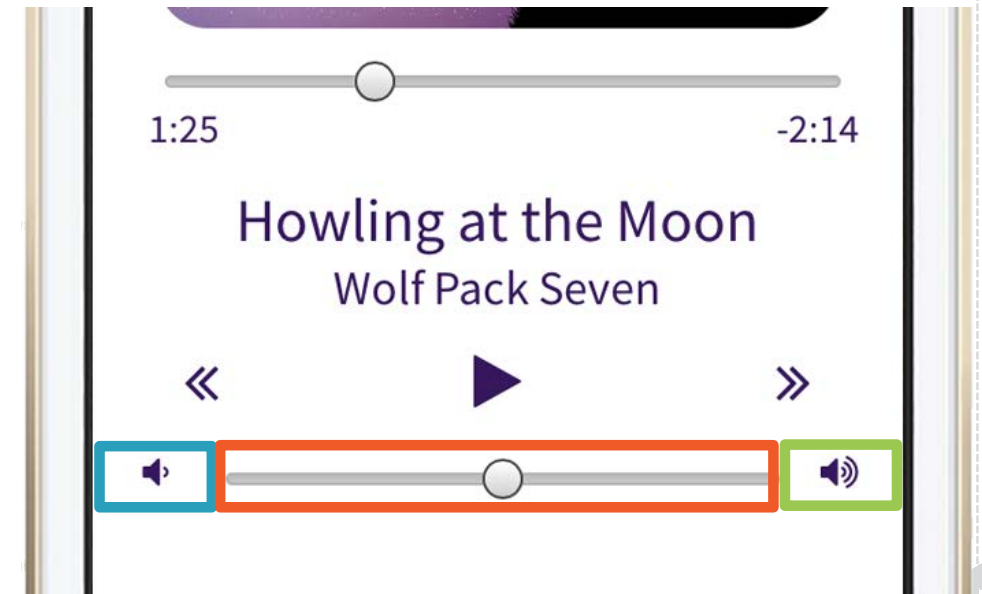


Volume

```
<div class="voldn"><span class="fa fa-volume-down"></span></div>
```

```
<div class="volume"><span><input type="range" name="volume" min="0" max="100"></span></div>
```

```
<div class="volup"><span class="fa fa-volume-up"></span></div>
```



Grid for Portrait Orientation

```
grid-template-columns: 50px auto 50px;
```

```
grid-auto-rows: auto;
```

```
grid-template-areas:
```

```
"photo photo photo"
```

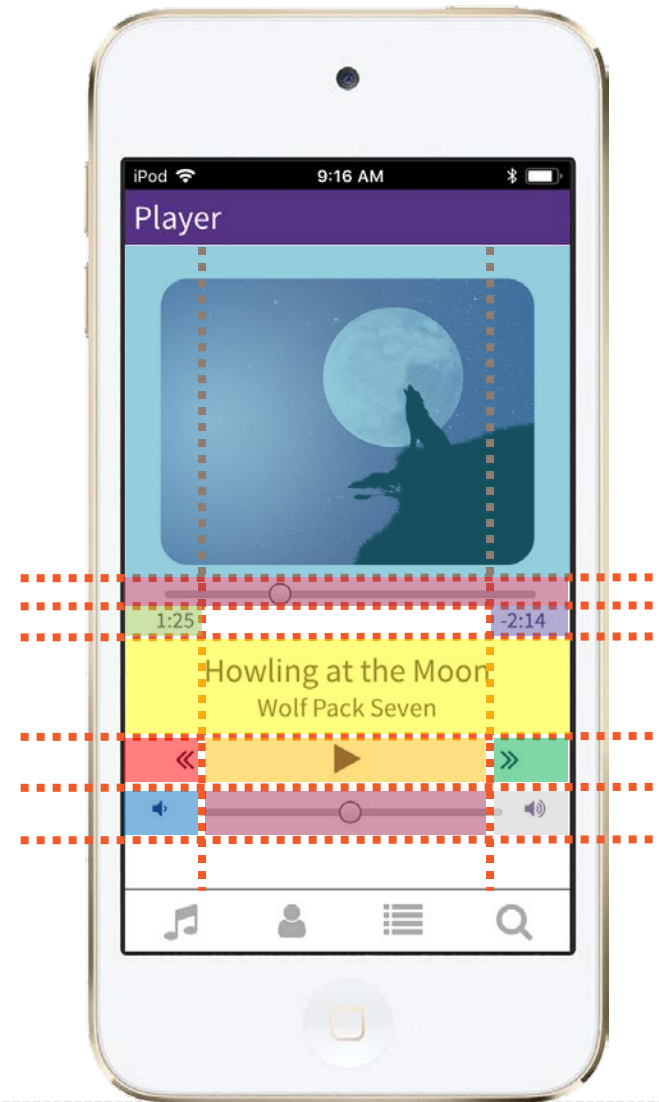
```
"time time time"
```

```
"startTime empty endTime"
```

```
"title title title"
```

```
"prev play next"
```

```
"dn vol up";
```



Lets jump back into brackets in the small file and create a new section for main with a class of player

Set it up as a grid
with the grid columns at 50 auto and 50
set the rows to automatic

Now create your template areas like this

Finally we need to add some margin to the top of the player so the image sits down a little.

When we view our page in a browser, we can see the margin at the top and that we still have a lot of work to do.

The next project is to assign all 11 divisions to their proper named place in the grid. Once again I have provided this bit of code for you in the snippets file of this unit.

Please copy this block of css and paste it in the small file below the grid template areas.

Lets look at our page and we can see that all the items are now in their proper place but we need some additional styling work.



Below the div.photo lets talk to the image.

Set the width to 84% and add margin of 0 at the top. 0 at the right, .5rem at the bottom and 8% at the left.

As you can see this centers the image.

Now lets round the corners using border-radius and 20px.

That looks great. Now lets work on the time indicator

Lets add justify self -- center which is the same a align center in a work processor and make the width 84%

If we use the chrome inspect tools we can see that the division for time is centered and 84% wide.

Now we need to make the range slider 100% wide and as you can see it's looking really good. Notice that the start time is touching the left edge.

Lets go back to the scss and tell the start time to text align right instead.

And there we have a nice balance. The title and band need some work now.

We will have both spans change to display as blocks

Then we will talk to the first span using nth child and set the font size and padding.

Now we talk to the second span and set the font size and padding.

This is the result and it looks much better.



As we move on we can see the player controls are too small and they need to be aligned differently.

In the css lets tell the previous button to justify self end which is like right align. and we will increase the font size.

For the play button, we will copy and paste and change the justification to center.

For the next button we will paste again and change the justification to start which is the default and leave the font size.

When we look at it in the browser it's much better but we need some space below so lets add padding-bottom to the play button of 1 rem.

This looks better so lets work on the volume section.

Lets move both of the speaker icons to the center using justify self.

That looks better but we need to work on the range input.

In the scss lets simply tell the input to be 100%

And it look marvelous in the browser.



Landscape Orientation

`grid-template-columns: 50% 50px auto 50px;`

`grid-template-rows: auto;`

`grid-template-areas:`

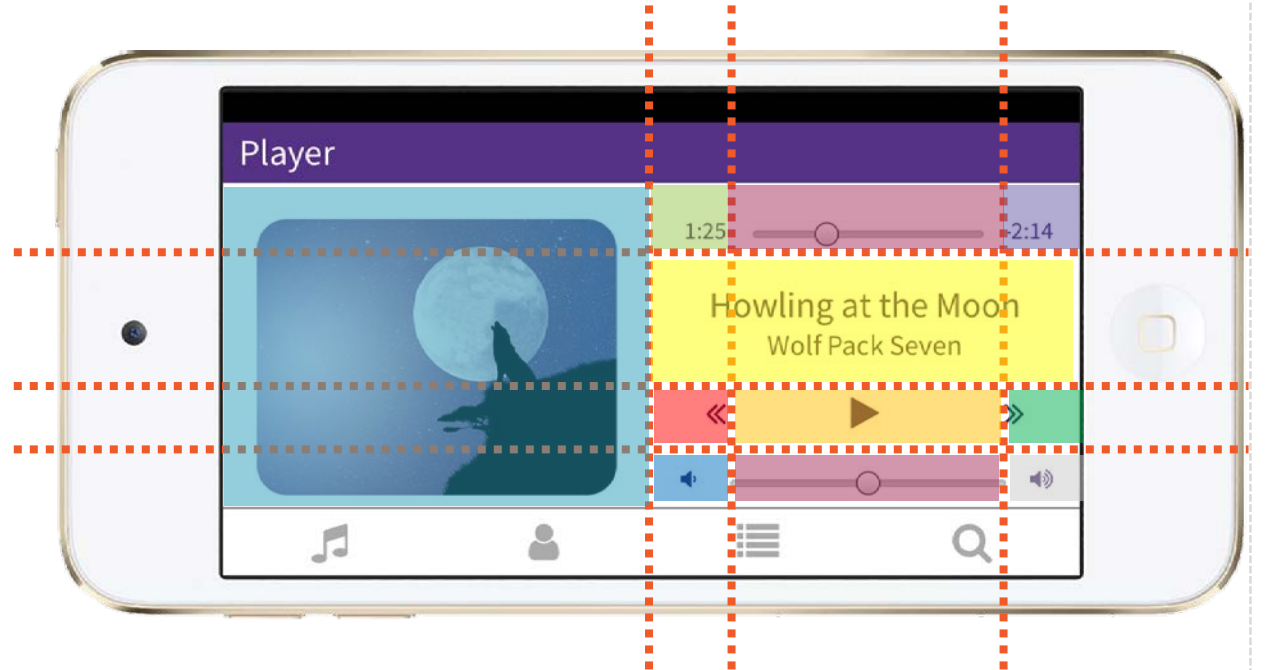
`"photo startTime time endTime"`

`"photo title title title"`

`"photo prev play next"`

`"photo dn vol up"`

`;`



Now lets go back to the text editor and type in all the code we just reviewed. This is what it looks like when we are done.

Isn't this great. Everything just fits into the proper grid cell based on the named areas and were done!

Lets upload this to our sand box test folder and try it out with real devices.



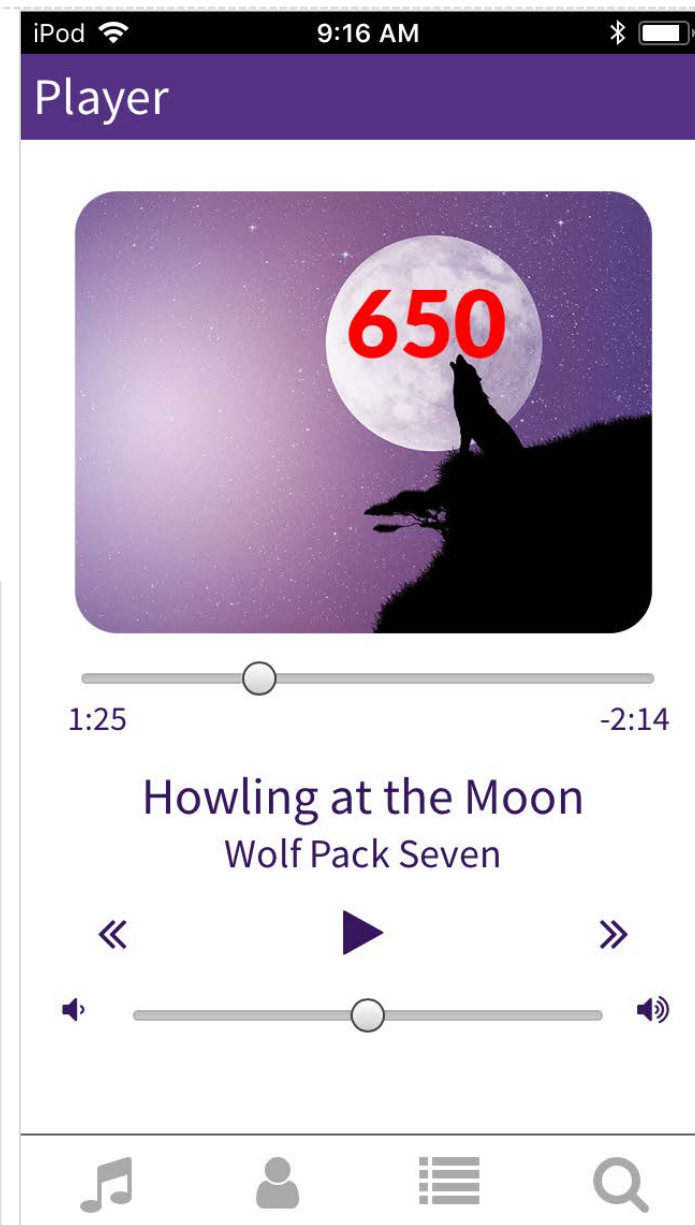
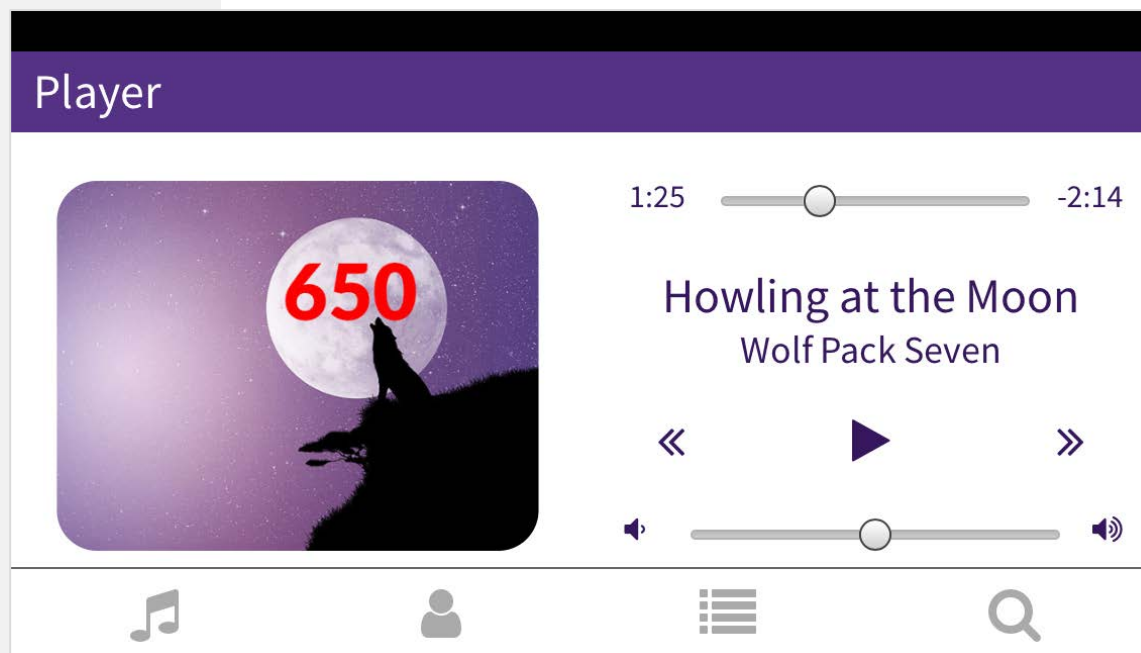
iPod Touch

Resolution:

320px x 568px

Pixel Ratio:

2x



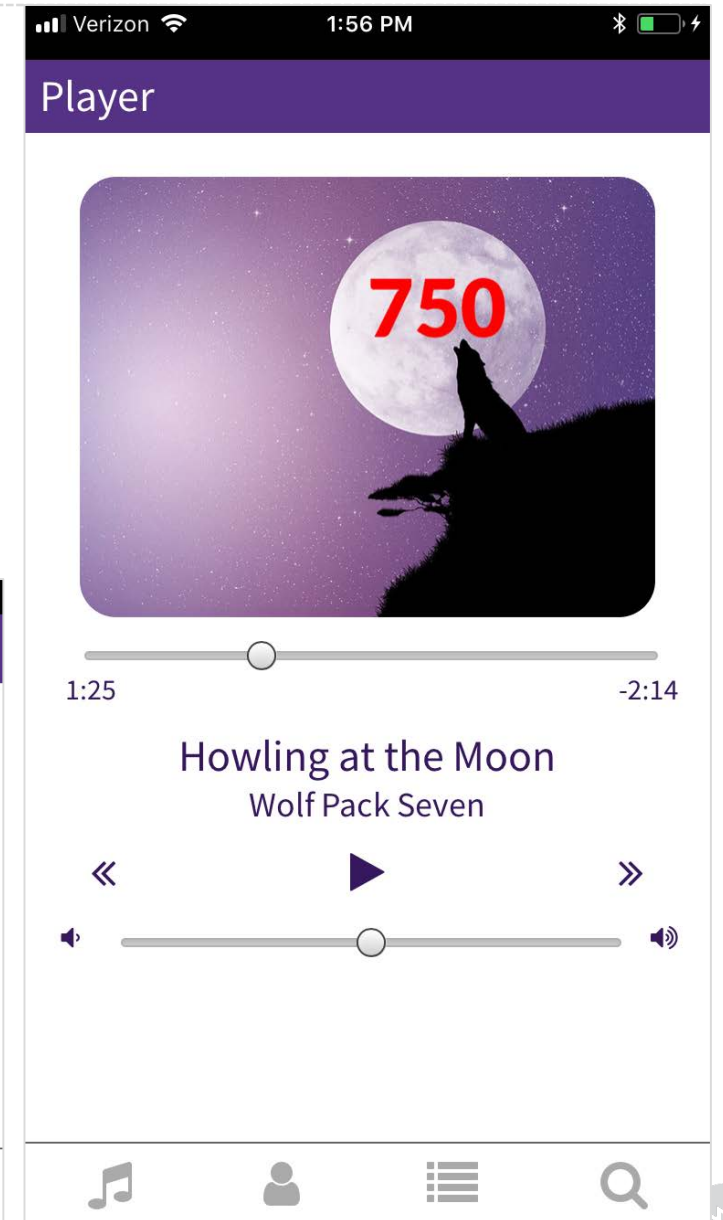
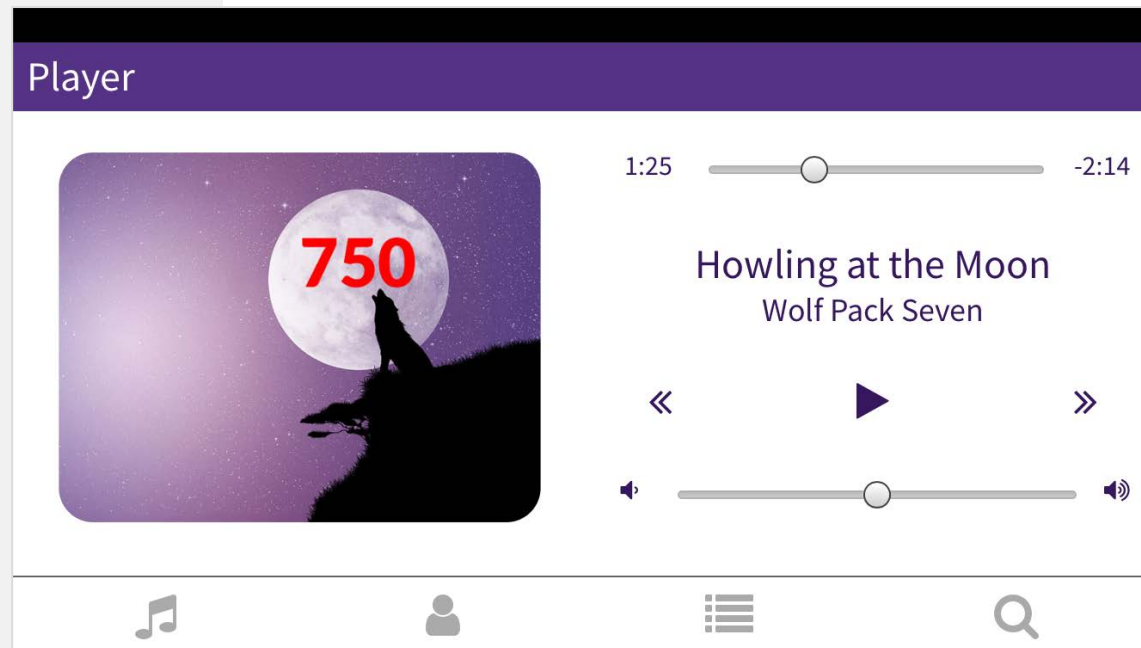
iPhone 6

Resolution:

375px x 667px

Pixel Ratio:

2x



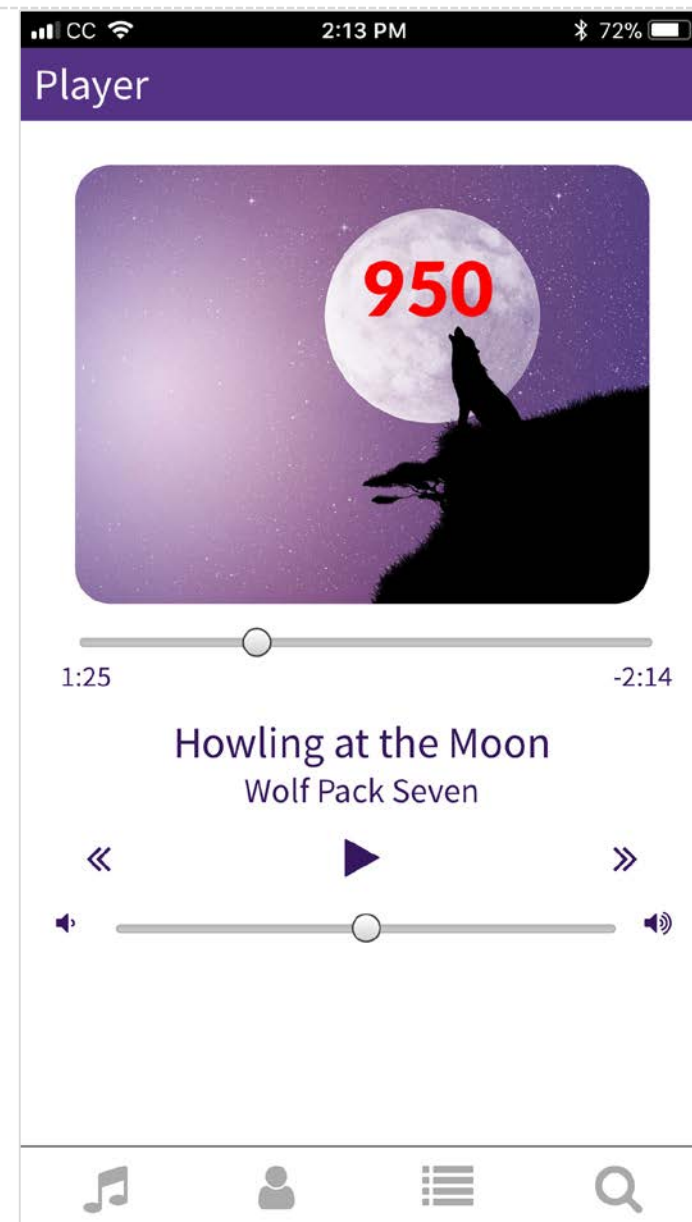
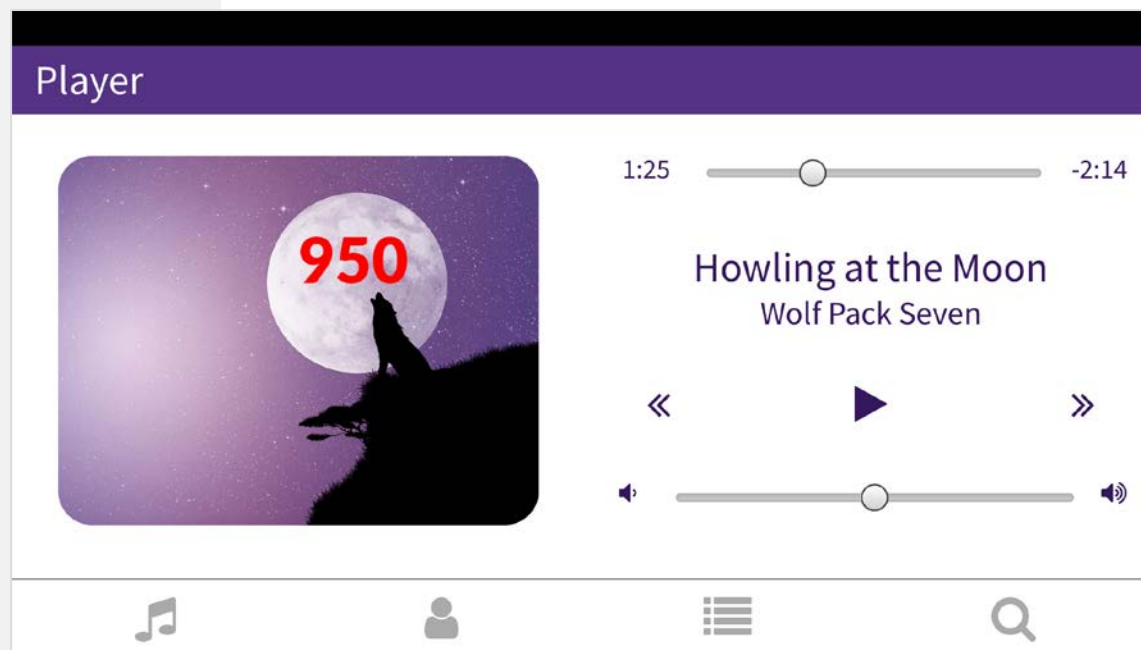
iPhone 6 Plus

Resolution:

414px x 736px

Pixel Ratio:

3x



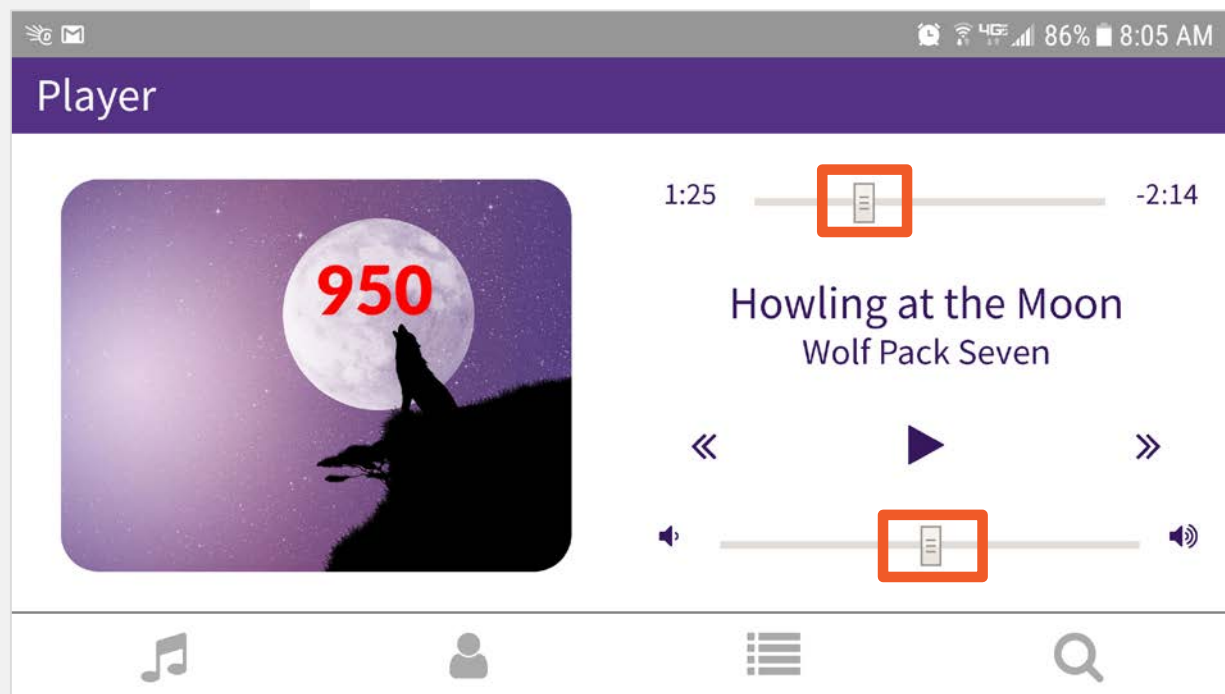
Galaxy 6

Resolution:

360px x 640px

Pixel Ratio:

4x



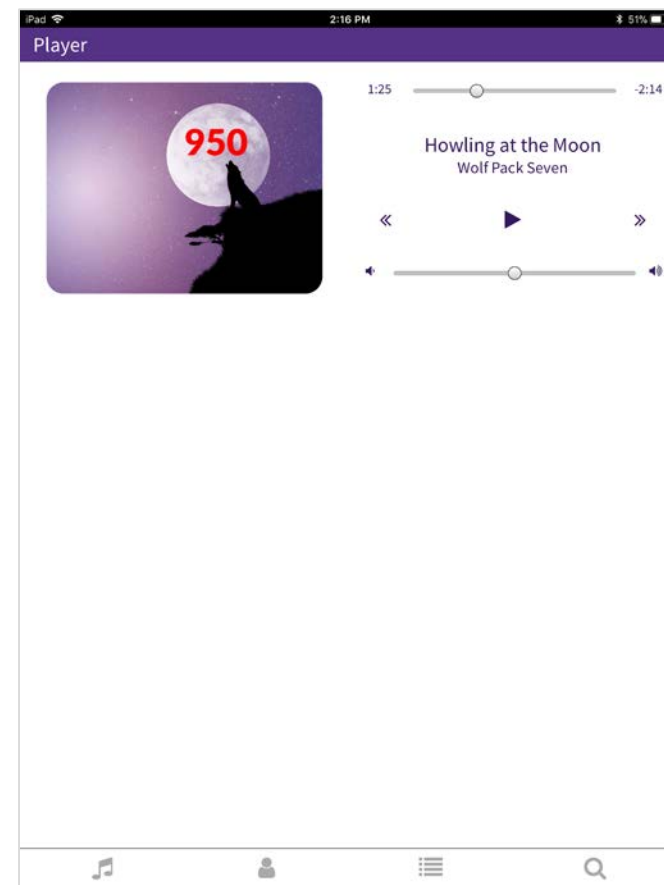
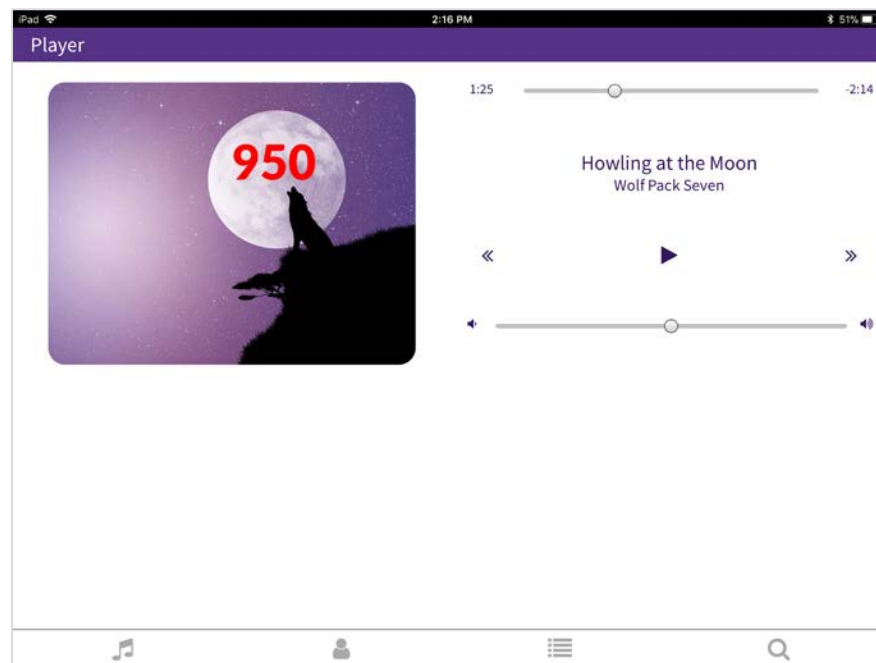
iPad 2

Resolution:

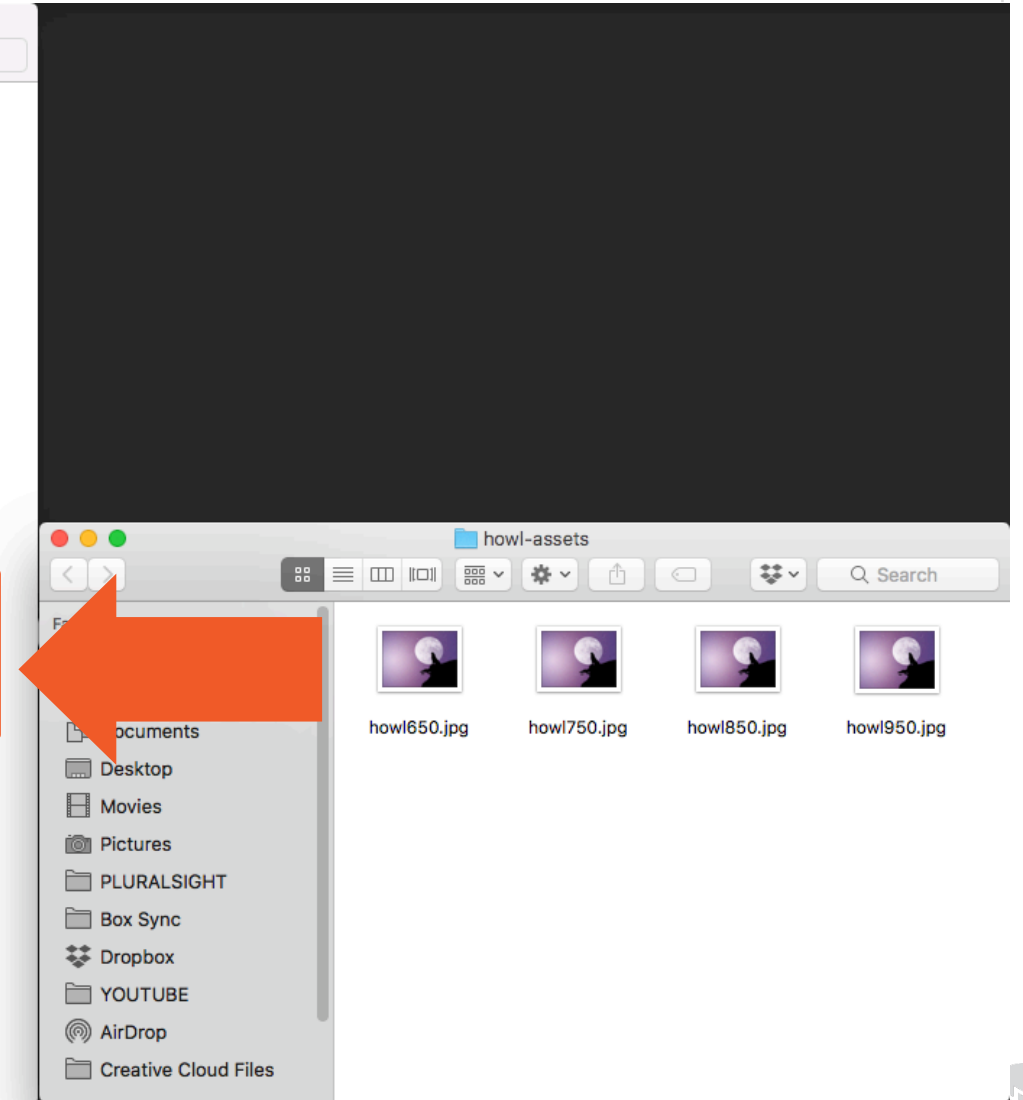
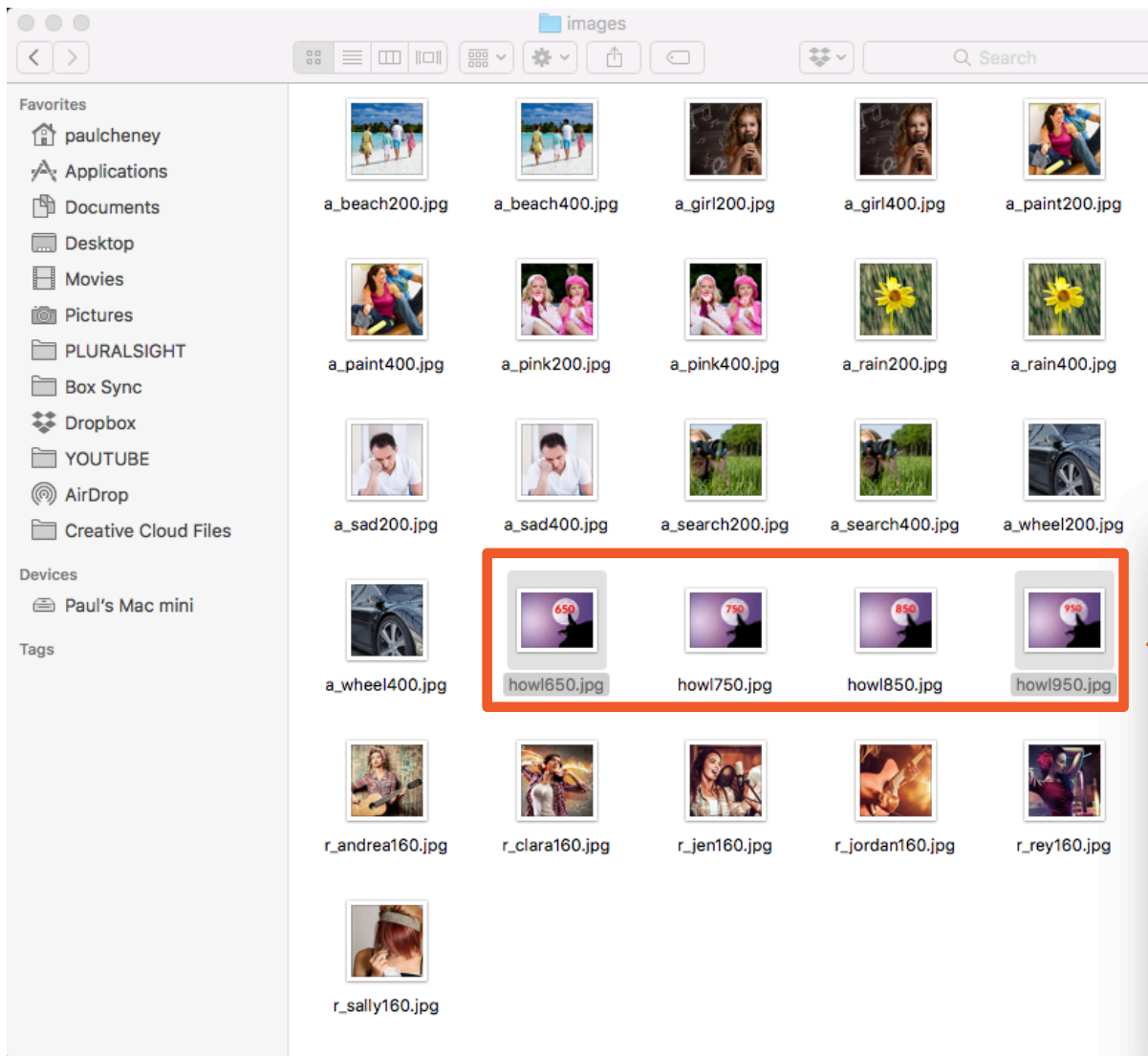
768px x 1024px

Pixel Ratio:

2x



Wrap Up



Summary



Design/Build Template

Design/Build Album Page

Design/Build Artist Page

Design/Build Player Page





Introduction

Twelve Column Grid

CSS Shapes

Flexbox

CSS Grids

Music Player

