

# Expanding Navigation

---



**Paul Cheney**

SPARTAN DESIGN UNIVERSITY

[spartandesignuniversity.com](http://spartandesignuniversity.com)



# Agenda



**Completed Project**

**SCSS for Navigation**

**Add JavaScript**

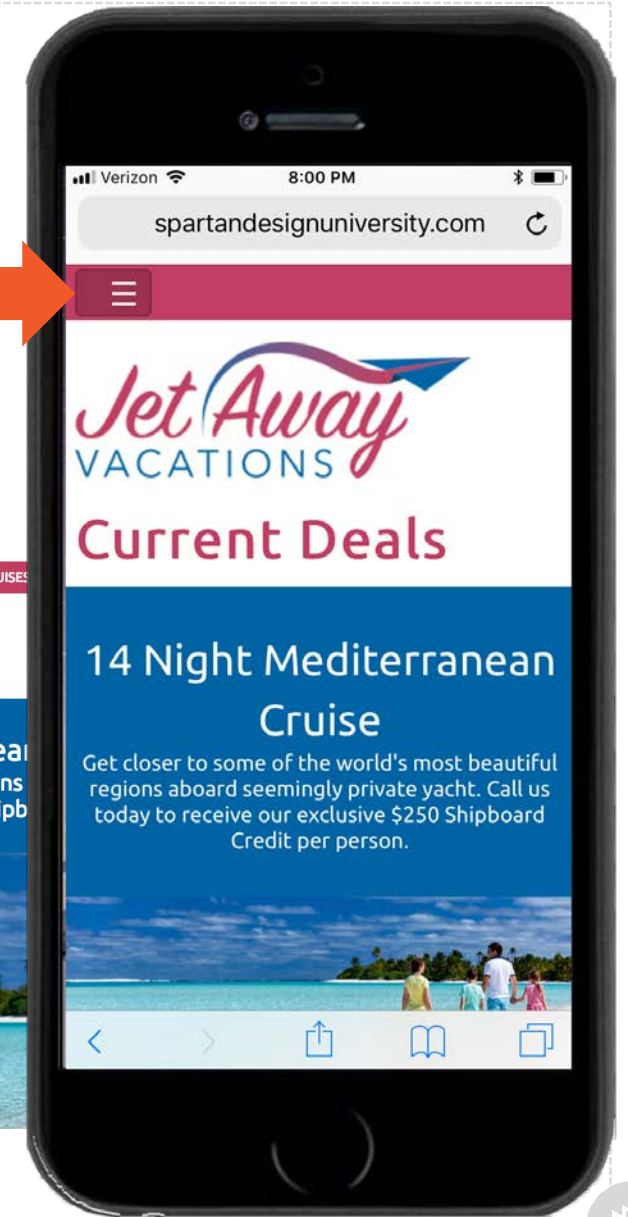
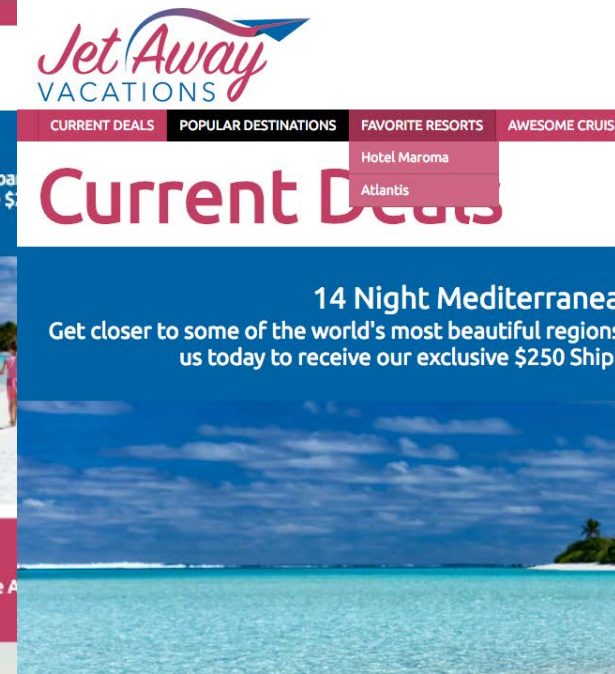
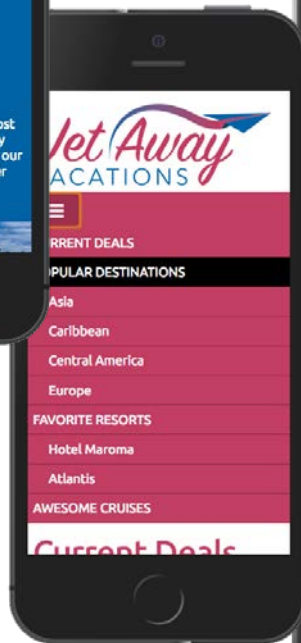
**CSS Grids**

**Testing**

**Application**



# Completed Project



# Hamburger Icon Options

MENU 

MENU 

**HTML ENTITY**

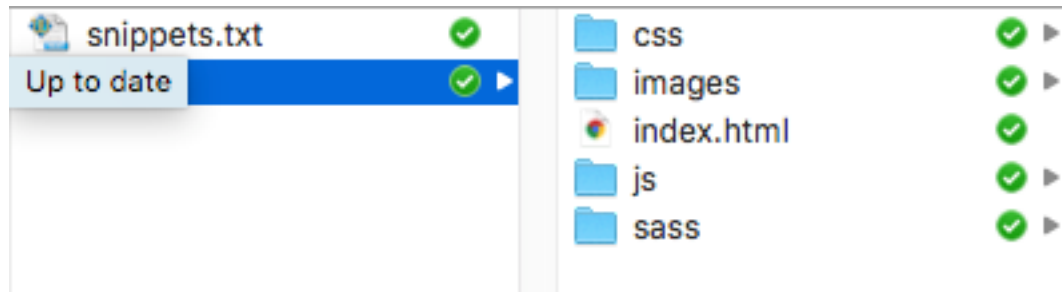


# The Hamburger Entity

[illegible]

Char	Dec	Hex	Entity	Name
☀	9728	2600		BLACK SUN WITH RAYS
☁	9729	2601		CLOUD
☂	9730	2602		UMBRELLA
☎	9731	2603		SNOWMAN
☼	9732	2604		COMET
★	9733	2605		BLACK STAR
☆	9734	2606		WHITE STAR
⚡	9735	2607		LIGHTNING
⚡	9736	2608		THUNDERSTORM
☼	9737	2609		SUN
⤴	9738	260A		ASCENDING NODE
⤵	9739	260B		DESCENDING NODE
♊	9740	260C		CONJUNCTION
♋	9741	260D		OPPOSITION
☎	9742	260E		BLACK TELEPHONE
☎	9743	260F		WHITE TELEPHONE
☐	9744	2610		BALLOT BOX
☑	9745	2611		BALLOT BOX WITH CHECK
☒	9746	2612		BALLOT BOX WITH X
X	9747	2613		SALTIRE
☂	9748	2614		UMBRELLA WITH RAIN DROPS
☕	9749	2615		HOT BEVERAGE

# Start File



# Delete Me



Please open the index file in your text editor and let's add an entity to the button using `&#9776;`. With the live preview running, we can instantly see the entity icon appear.

Let's open the small scss file and add nav with open and close curly's

Please set the background color to the variable `$menuColor`. We will be using a red color for this module. Now let's style the button.

Within the nav selector, add a button selector and let's change the size of the hamburger icon. Because we are using an entity, we can change it using font size. Let's use 1.5 rem and now we have a large ugly button.

Also because we are using a font, we can change the color using css instead of building a different Photoshop graphic. Please use `color: white`.

The rest of these changes are identical to the previous module.

We change the background color to 20% black using `rgba`

Use a border of 30% black

increase the width of the button

add margin to give some space around the button and finally

specify the radius... just in case the browser does not round it by default.





Styling the navigation items is identical to the last module so I have provided the code in the snippets file for this module. Please copy the first three selectors and past them just above the button selector.

If you look in the index file you will see that the second menu item has the class = active. In our browser the main menu item and the children have changed to black. In order to get ONLY the main menu item and NOT the children to change to black we will use a Greater Than symbol between the class name and the anchor... like this.

Now the medium SCSS is identical to the previous units so rather than repeating myself, please open the snippets file and copy the code for medium.  
Paste it starting on line 2

Then copy the three lines of code for the large screens from the snippets file.  
Paste it on the large file also starting on line 2.

Now lets preview our work starting with the small screen and expanding it out to medium and then to large. Notice that the page title stays below the navigation instead of being covered up as in the last module.  
Now its time for the JavaScript.



# Junk Me



# JavaScript Differences



`class="closed"`

`class="closed"`



`class=""`

`class="open"`



# ID's

```
27 <!-- NAV STARTS HERE -->
28 ▼ <nav class="clearfix">
29     <button id="hamburgerBtn">&#9776;</button>
30 ▼ <ul id="primaryNav">
31     <li><a href="index.html"><span>CURRENT </span>DEALS</a>
        </li>
32 ▼     <li class="active"><a href="#"><span>POPULAR
```

```
68 <!-- PHONE MENU HIDER -->
69 <script src="js/menuToggle.js"></script>
70 </body>
71 </html>
```



Lets start out building a function called toggleMenu  
Inside it we will send a message to the `console.log` of "It Worked"

Now we need to create a variable and assign it to the button with an id of hamburger button.

Then we type `x . onclick` which waits for a button click and triggers the function `toggleMenu`.

When we test our work, we can see our message in the console. So far everything is working out as it did in the previous module.  
Please remove the console log statement.



## Switch



if the **className** is "closed" then  
change the **className** to "open"  
otherwise change the **className** to "closed"

```
if (document.getElementById("primaryNav").className==="closed" ) {  
    document.getElementById("primaryNav").className="open";  
} else {  
    document.getElementById("primaryNav").className="closed";  
}
```

```
var y=document.getElementById("primaryNav");  
if (y.className==="closed") {  
    y.className="open";  
} else {  
    y.className="closed";  
}
```

Now that you have seen the code, you can either type it in or you can get a copy from the snippets file and paste over the entire script like this.

If we take a look at the unordered list we can see that the class name changes each time you click the button. Now all we have to do is tweak the CSS.

From the top of Current Deals to the bottom of Awesome Cruises is the unordered list. When the class for the ul is closed then we will set the max height to 0. When the class for the ul is open we will change the max height to 31rem which is 496 px which is enough for the 10 menu items we have. If you had more you may have to increase this value.

Now when we click the hamburger, the red color opens and closes but you will notice that the menu words are still there... even when it is closed.

To correct this problem we need to add to the nav ul... overflow of hidden.

Now watch this, we still have the problem of the menu disappearing on medium screens when the menu is closed on small screens.

Lets open the medium file and add ul.closed and set the max height to 31rem so it shows up.



You probably already noticed that when the page first loads, the menu was open. We want it to be closed so lets add a class= closed to the ul.

This works great for all three screens, but wait... were not done yet!

Lets return to the small file. To the unordered list lets add a transition so that it animates open and animates closed. We will have to do a -webkit prefix and -moz prefix as well as the default transition for the max height property. This is what we now have.

As we move between the small and medium screens you may notice that the menu animates. If you think this is cool, then your done. However, if this bothers you then all you need to do is move the three transition statements from the ul to the ul.open selector.

The result is no animating between views, however, you lose the closing animation. This may not be a big deal since most of the time the user selects a menu item and you don't see a closing animation anyway.

Well, now you know how to do it both ways.





# Junk Me



Before we test our work, let's take a look at using CSS grids to move the menu around. It's really easy.

First we will duplicate the start folder and rename it startGrid

Load the start grid folder into your text editor and refresh your live preview.

We want to have the nav bar at the top for small screens and below the header for medium and large screens.

Let's start by displaying the body of the html as a grid. The body contains the header, nav, main, and footer semantic elements.

Next we will set the rows to auto even though this is the default.

Finally we will create grid template areas called myNavigation, myHeader, myContent, and myFooter.

Then we will assign the header to the grid area myHeader which is the second row down

Next we assign the nav tag to the grid area myNavigation which is the top row.

Then we assign the main tag to display in the myContent and finally assign the footer tag to display in the myFooter grid area.

Now when we look at the page the navigation has moved to the top just like it should.



When we view the medium screen, the menu is still at the top.  
We want it below the header.

Copy the grid template areas from the small file.

Open the medium file and add a selector for the body tag.

Paste into the body tag.

Switch position with myNavigation and myHeader

Save and refresh and lets check out our finished page starting with small, then medium and finally large.

You have got to just love css grids!

Now its time to upload both of these to the sandbox and test our work.



# Delete Me

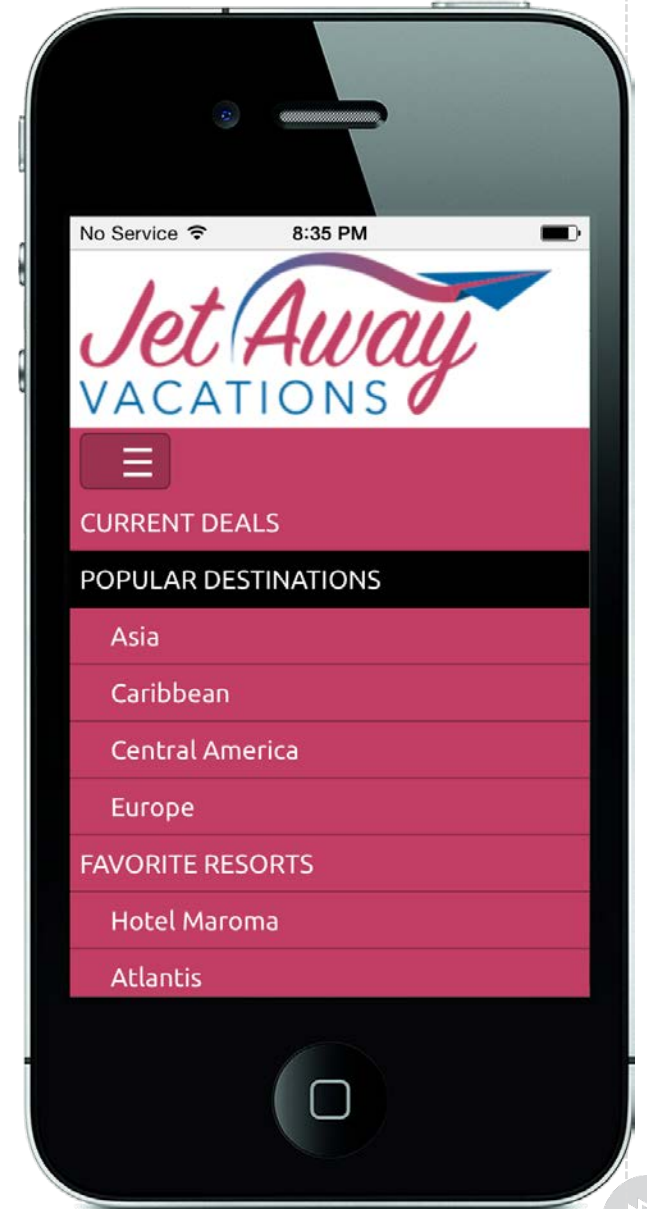
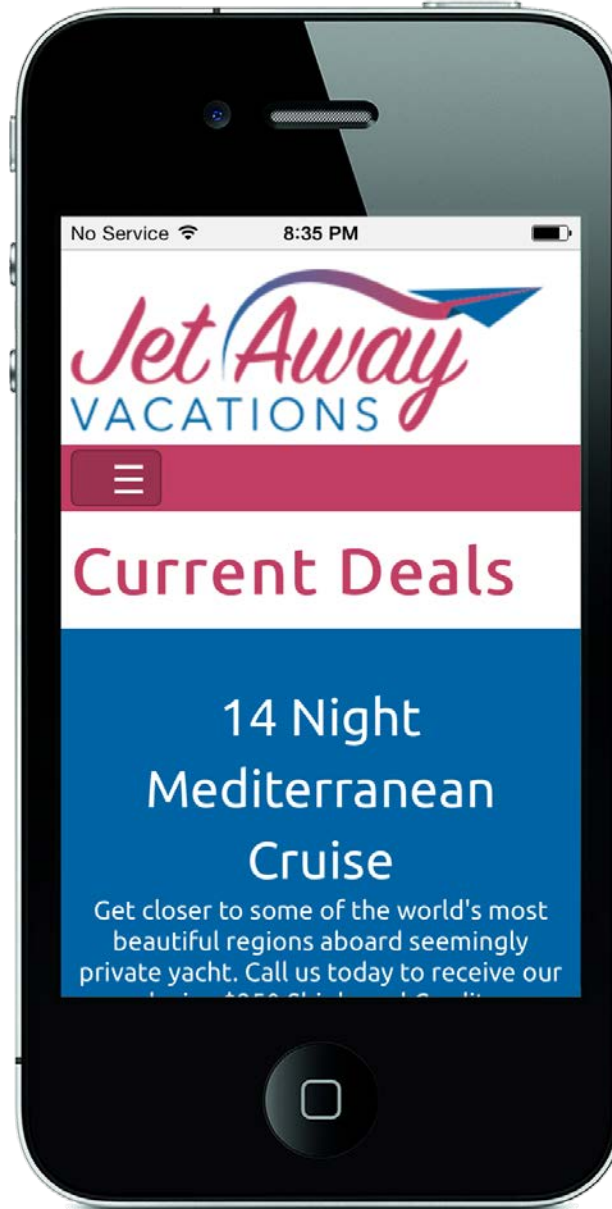


# Testing



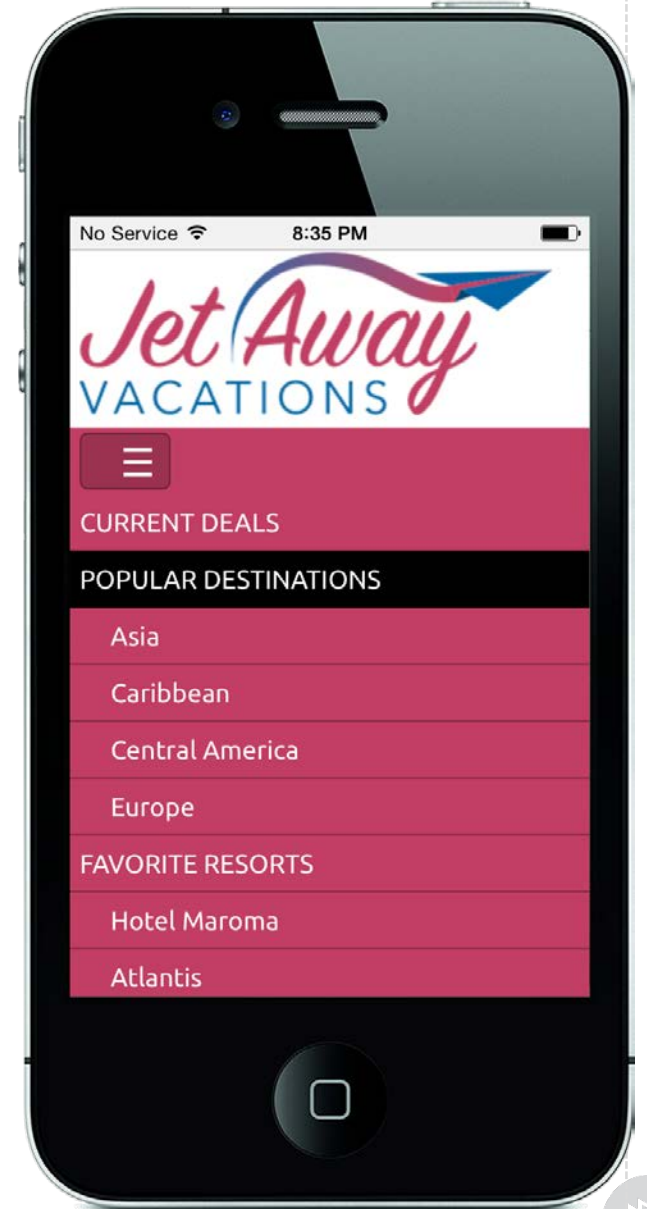
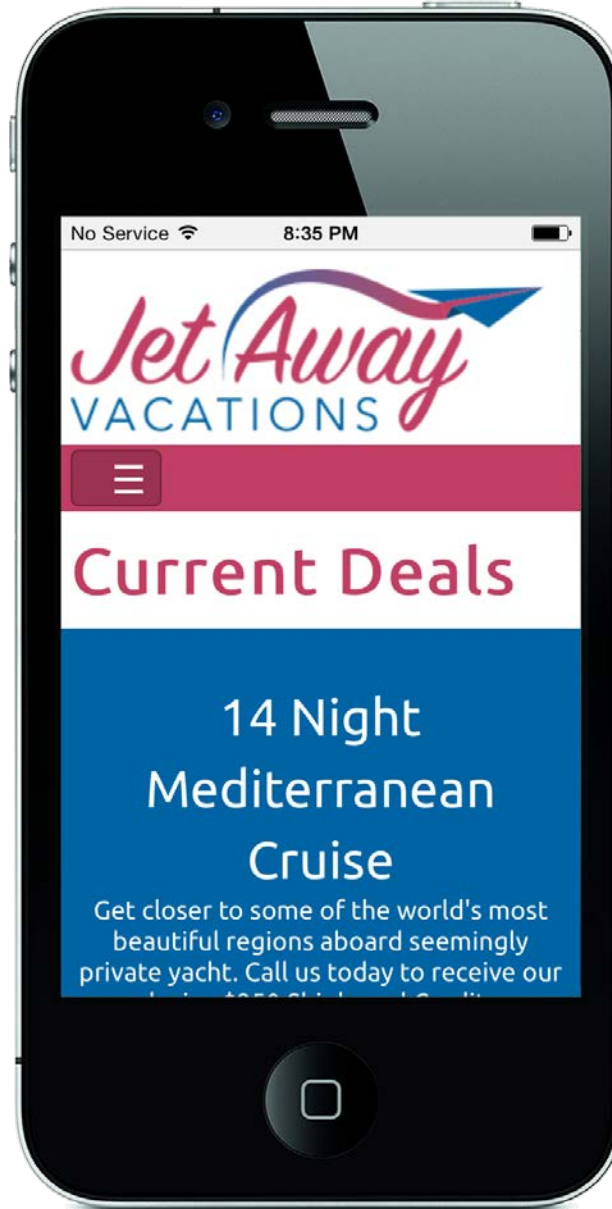
# iPhone 4

start



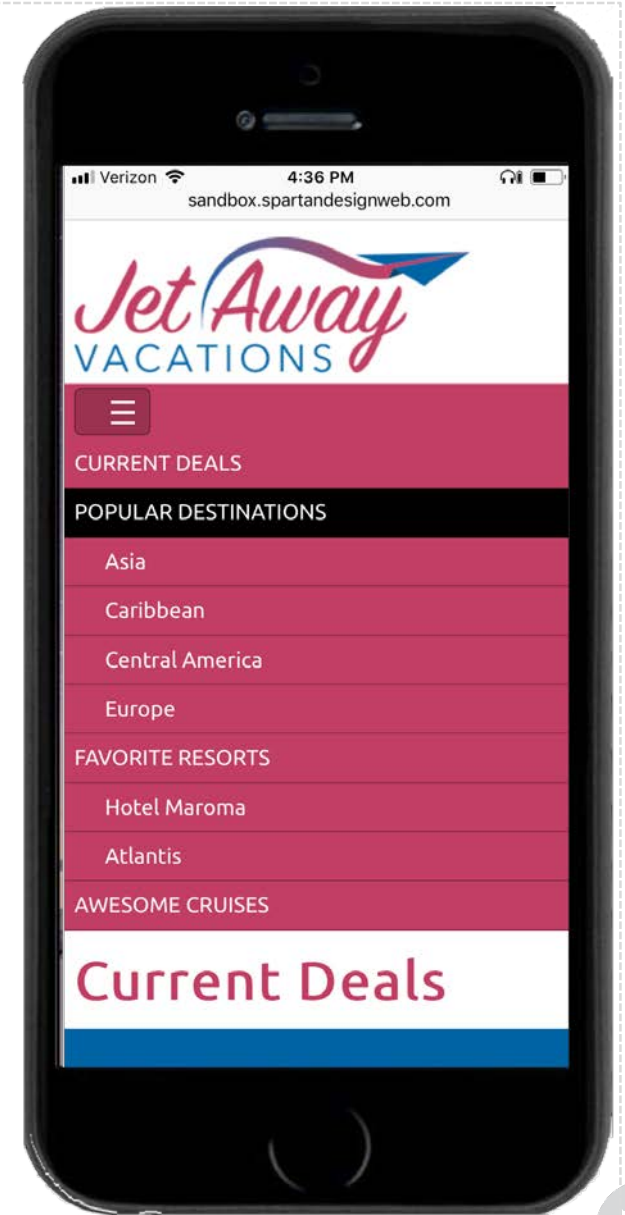
iPhone 4

startGrid



# iPhone 6

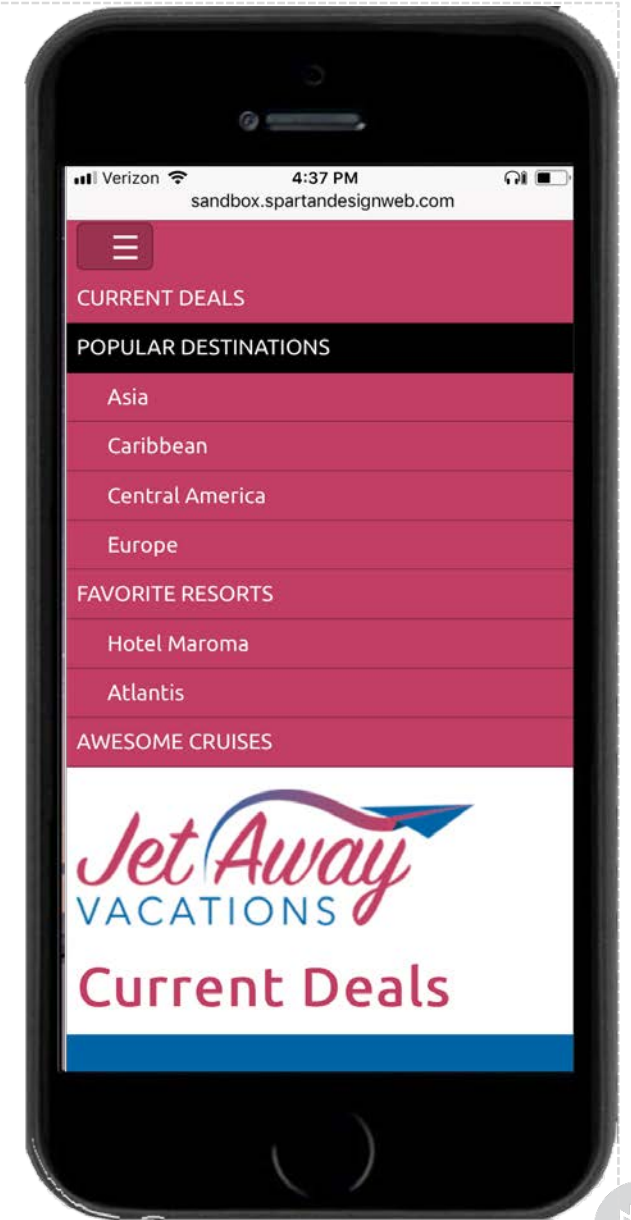
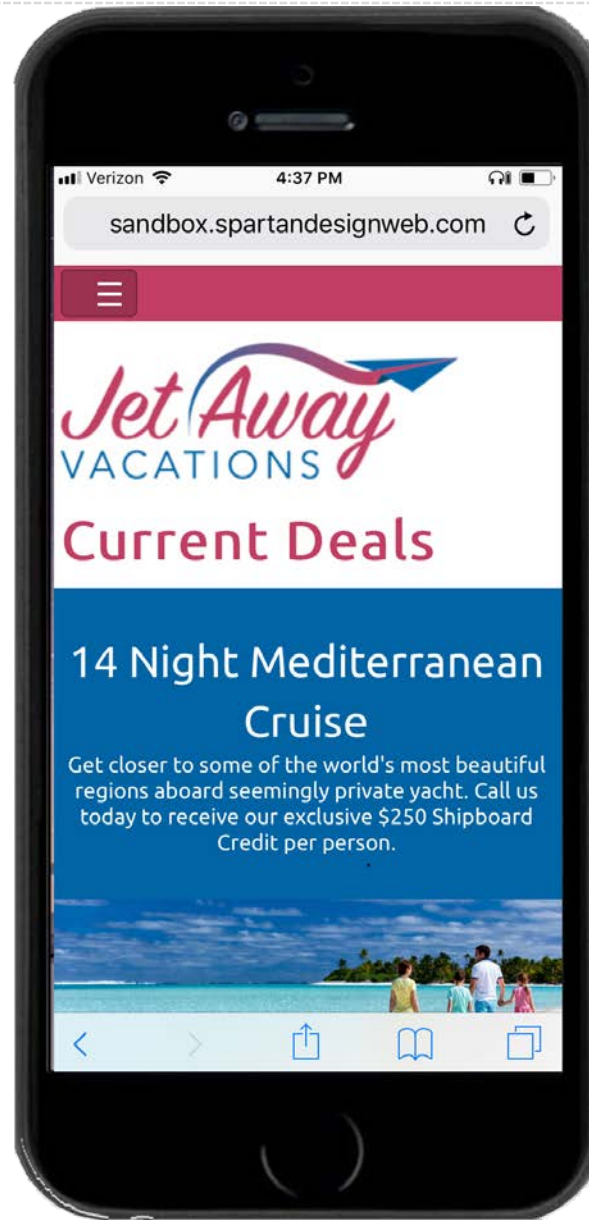
start





iPhone 6

startGrid



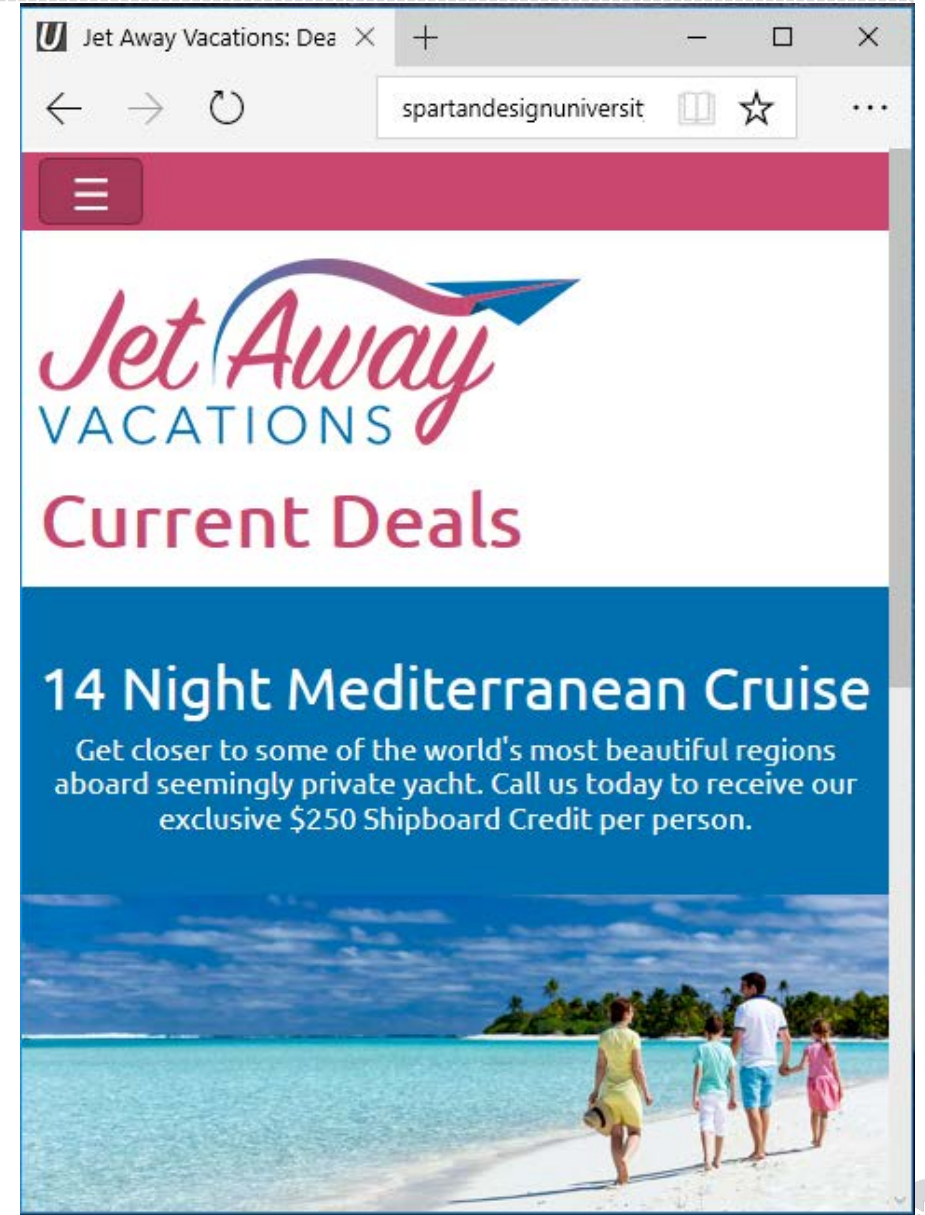
Internet Explorer  
9, 10 , 11, edge

start



# Internet Explorer 9, 10 , 11, edge15

startGrid





Application

Nine Items

Avoid Scrolling

Primary and Secondary

Horizontal Navigation

**Nine Items**

## Avoid Scrolling

## Primary and Secondary

## Horizontal Navigation



# Summary



Completed Project

SCSS for Navigation

Add JavaScript

CSS Grids

Testing

Application





Introduction

Coding the Navigation

Cover Navigation

Expanding Navigation

**Footer Navigation**

Expand on Touch I

Expand on Touch II

Icon Navigation

