

# Jumpstart

In this project, you'll create a responsive landing page for a crowd-sourcing website. First, you will use HTML and CSS with Bootstrap to structure and style your web page. Then, you will use CSS to animate images on the page, the **Slick** jQuery plugin to add a functional carousel of images, and jQuery to hide the navigation bar as the user scrolls.

You should complete this project locally on your own computer. Be sure to consult the **design spec** for information about fonts, colors, and dimensions. You can preview a working version of the website **here**. Use your documentation research skills anytime you feel unsure of how to achieve an effect!

Mark the tasks as complete by checking them off

## Initial Setup

### 1.

Create a project folder named Jumpstart. Inside of that project folder, use best practices to organize your files. Be sure to include a standard **reset.css** file. If you want to see our recommended file structure, check the hint.

You can download the assets you need **here** 

Hint

jumpstart

```
l_ index.html
l_ favicon.png
l_ resources
    l_ css
        l_ reset.css
        l_ style.css
    l_ images
    l_ js
        l_ main.js
```

## 2.

Open **index.html** and include standard boilerplate code. Inside of the `<head>` element, right below the `<title>` element, use a `<link>` tag to add the favicon provided.

## 3.

Also inside the `<head>` element in **index.html**, include these Google Fonts:

- Open Sans with font weights 400, 600, and 700
- Monoton

## 4.

Your project will have four stylesheets. Include them in this order:

- **resources/css/reset.css** (a standard **reset.css** file)
- **bootstrap.min.css** (link to the CDN)
- **slick.css** (link to the CDN)

- **resources/css/style.css** (your page's custom style sheet)

## 5.

Your project will have three JavaScript files. Include them in this order:

- **jquery-3.1.1.min.js** (link to the CDN)
- **slick.min.js** (link to the CDN)
- **resources/js/main.js** (your page's custom JavaScript)

## Create the Layout of Your Page

## 6.

Add a `<nav>` section to your web page that is fixed to the top.

- Inside of the `<nav>` element, add a Bootstrap container.
- Inside of the Bootstrap container, add an unordered list that contains links.
- Use Bootstrap classes such as `navbar` and `navbar-fixed-top` to achieve the effects described above. For more information check the [Bootstrap documentation](#).

### Hint

The `<nav>` should have the following classes: `navbar`, `navbar-default`, and `navbar-fixed-top`.

The `<ul>` should have the following classes: `<nav>` and `<navbar-nav>`.

## 7.

To continue building the navigation section:

- The first item in the list should have the class `active`.
- You will need to override the built-in Bootstrap styles for the class `active` in `style.css`.
- Include the logo image provided. This image should hide when the browser shrinks.

### Hint

You can use these two selectors to style the active link:

```
.navbar-default .navbar-nav > .active > a
```

and

```
.navbar-default .navbar-nav > .active > a:hover
```

The logo image should have the following attributes:

```

```

## 8.

Before adding the carousel functionality, add a Bootstrap "jumbotron" section. Use a `<div>` tag with two classes: `jumbotron` and `jumbotron-bike`.

- In **style.css**, add **../images/jumbotron-bike.jpg** as the background image for this section.
- In this section in **index.html**, add the header text **Get Going**. Below that, add the following paragraph: **Jumpstart your startup with some crowdsourced seed money, or support one of our 300,000,000+ creators and get rewards.**
- You will need the **text-shadow** property to achieve the outlines of the fonts. Check the [Mozilla Developer Network \(MDN\)](#) for documentation on this property.

## 10.

Add the "Featured Project" section.

- The image of the featured project should take up half of the width of the screen on large devices and the entire width on smaller devices. Make a **<div>** with the correct classes to achieve this.
- Inside of that **<div>**, nest another **<div>** with the class **image-container**; we'll be referring to this class name to achieve a special effect later.

- Each image should be inside of its own `<div>` with the class `image-container`. Include that property in **style.css**.
- The section that describes the documentary and the section that lists the types of projects should each take up a quarter of the screen on large devices and half of the screen on smaller devices.

#### Hint

```
.image-container img {  
  opacity: 0.7;  
}
```

### 11.

Below the "Featured Project" section, add the "Projects" section.

- Each project should take up a third of the screen on large devices and the whole screen on smaller devices.
- Each image should be wrapped inside of a `<div>` tag with the class `image-container` so that later we can apply the same special effect.
- Notice that the images in this section are somewhat transparent. Style them that way in **style.css**.

### 12.

Below the "Projects" section, add the "Having trouble?" section.

- Use a single `<div>` to contain the `<strong>` text, the paragraph, and the button.
- The button can be implemented exactly the same as the button in the "jumbotron" section.
- Refer to the design spec

### 13.

Now that your page is fully laid out and styled, add media queries to make your page responsive to different screen sizes. Refer to the design spec for appropriate media queries and change the necessary styles.

## Scale and Increase Opacity of Images

### 14.

Notice that when you hover over the images in the "Featured Project" and "Projects" sections in the example site that the images transition; they zoom slightly and become less transparent. This is why we use the same class for both, `image-container`.

### 15.

Images within the `.image-container` class have a value for opacity declared in `style.css`. In that same selector, use a CSS transition on the transform property over 0.5 seconds and to change the opacity over 0.5 seconds. You may find the documentation for transformations [here](#).

Hint

```
.image-container img {  
  transition: property time, property time;  
}
```

You can declare both transitions in the same line, separated by a comma.

## 16.

Use the CSS `scale` transformation to scale the images to 1.3 times as large when the user hovers.

Hint

```
.image-container img:hover {  
  opacity: 1;  
  transform: transformation(value);  
}
```

You will need to hide the overflow within elements with the class `image-container` to achieve the effects described above.

You can use the code below to achieve this:

```
.image-container {  
  overflow: hidden;  
}
```

**Create the Carousel**

## 17.



To create the carousel, add a `<div>` tag with the class `carousel` that encloses the entire tag with the class `jumbotron`.

## 18.

Duplicate the entire `jumbotron <div>` three times.

The `carousel <div>` should now have four `jumbotron <div>`s stacked on top of each other.

You can copy the text needed for this section [here](#).

## 19.

Change the `jumbotron-bike` class of each of the three new `<div>` tags to reflect the names of the other three background images provided.

## 20.

In **style.css**, style the background images of each of the three new `<div>` tags you have created to be the three images that you have not yet used: **jumbotron-flowers.jpg**, **jumbotron-beer.jpg**, and **jumbotron-duffel.jpg**.

Hint

```
.jumbotron-flowers {  
  background-image: url("path/to/jumbotron-flowers.jpg");  
}
```

## 21.

Update the header and paragraph text of each of these jumbotron sections to reflect the text in the mock and the example website (which you can visit again [here](#)).

## 22.

Slick is a jQuery plugin that you will use to add the carousel functionality to the section you edited in the last step. Read the documentation for Slick [here](#).

## 23.

In **main.js**, when the document is ready, apply this configuration to the **carousel**:

- One slide should scroll at a time.
- The carousel should autoplay when the page is loaded.
- The speed should be 4000 milliseconds.
- Do not include arrow buttons.
- Do not allow the user to drag the images.
- Do not pause on "focus" or "hover".

## Hide the Navigation Bar

## 24.

In **main.js**, use jQuery to hide the navigation bar as the user scrolls. Follow the steps below to achieve this effect!

## 25.

Create a variable to store how far the user has scrolled and set it equal to `0`. Use proper JavaScript naming conventions and give this variable a name that accurately describes the value stored in it. We'll refer to this variable as `x` for our explanation.

## 26.

Use the `.scroll()` event handler to select the window. Inside of the event handler, include the control flow described below.

## 27.

Create a variable that stores the value of the `.scrollTop()` property of `this`. We'll refer to this variable as `y` for our explanation, but you should give it a more appropriate name.

Hint

```
var y = $(this).scrollTop();
```

## 28.

If `y` is greater than `x` by `50` or more, create a variable that stores the height of the navbar, `z` (remember to give the variable a descriptive name!).

Hint

```
if (y - x > 50) {  
  var z = $('.navbar').css('height');  
}
```

## 29.

Use the jQuery `.animate()` function to to change the top offset of the navbar to be the negative height of the navbar over 150ms (using the variable you just created).

Hint

```
$('.navbar').animate({top: '-' + z}, 150);
```

### 30.

Then, set `x` equal to `y` so that you always have a record of how far the user scrolled last time they finished scrolling. You need this record to compare in your control flow.

### 31.

Otherwise, if `x` is greater than `y` by 50 or more, use the `.animate()` function to change the `top` offset of the `navbar` to `0px` over 150 milliseconds.

### 32.

Set `x` to the value of `y` so that you always have a record of how far the user scrolled last time they finished scrolling. You need this record to compare in your control flow.

Congratulations! You have created a beautiful, professional, animated web page that draws users' eyes to important information and creates an engaging user experience.