Madlibs Project

1. Create the folder you want to store the app in
2. Create a git repository: ***git init***
3. Create a git ignore file and place node modules in it
4. Install the devcamp-js-builder: ***npm devcamp-js-builder -g***
5. Run js-generate: ***js-generate***  
    add in app type and name—this creates the necessary files

node\_modules  
 proxy  
 src  
 components  
 app.js  
 style  
 main.css  
 static  
 index.html  
 webpack  
 config files  
 .babelrc  
 env.js

1. Install dependencies: ***npm install***
2. Make initial commit:  ***git status  
    git add .  
    git commit -m “initial commit”***
3. Create a git hub repository and push initial commit to it  
   ***git remote add origin*** [***https://github.com/cjaymarshall/exampletodelete.git***](https://github.com/cjaymarshall/exampletodelete.git) ***git push -u origin master***

Structure

index.html—

The index.html file is stored in a static folder—it contains in its body the div that the  
 app is dumped into. With this framework it is called labeled ***class = “app-wrapper”***  
 (in other tutorials it is called “root”)

app.js—

The app.js file is stored in the src folder—this contains the main components of the  
 react app. However, because this is in the context of bootstrap it is ultimately exported to the bootstrap.js file  
  
 In other frameworks this would be the index.js file

The app.js file

Need to import React and its Component class  
 (comma separate things you import)  
 (any classes you import go in curly braces)

***import React, {Component} from “react”;***  
  
 Create the App class as an extension of react component class this gets exported to bootstrap.js   
 also needs an export default statement  
 ***export default class App extends (Component) {  
  
 }***

The return statement usually contains all of the jsx code within a container div statement  
  
 ***render() {  
 return(  
 <div className = “app”>  
 <div className = “firstDivClassName”></div>  
 </div>  
 );  
 }***

**The Madlibs Project**

In this framework the app is built with jsx primarily in the app.js file—this does not require a ReactDOM statement because the component built here will get exported to the bootstrap.js file, which will take care of sending it to the DOM. However, there is a standard render() method which will return everything in our app.

Header

The first part of this project to build was the header. This was built as a series of nested react  
 div elements  
  
 The following code was the final code for the header in the app.js file—

***import React, { Component } from 'react';***

***export default class App extends Component {***

***render() {***

***return (***

***<div>***

***<div className="backgroundSkew">***

***<div className="madlib-heading">***

***<h1>Botegga Madlibs</h1>***

***<div className="madlib-subheading">***

***Fill out the fields below and click the generate button<br/>to see the Mad Lib story***

***</div>***

***</div>***

***</div>***

***</div>***

***);***

***}***

***}***

The following code was the styles for the header in main.scss

***$bottegaGreen: #00CB79;***

***$clearBlue: #45B0E5;***

***$mainGray: #CFCFCF;***

***$darkGray: #444444;***

***$font-primary: 'Montserrat', sans-serif;***

***.backgroundSkew {***

***width: 100vw;***

***height: 700px;***

***background-color:$darkGray;***

***transform: skewY(-10deg);***

***margin-top: -17rem;***

***padding-top: 17rem;***

***border-bottom: 1rem solid $bottegaGreen;***

***}***

***.backgroundSkew > \* {***

***transform: skewY(10deg)***

***}***

***.madlib-heading {***

***text-align: center;***

***margin: 20px;***

***padding-top: 46px;***

***color: white;***

***font-weight: 700;***

***font-family: $font-primary;***

***}***

***.madlib-subheading {***

***margin-bottom: 46px;***

***color: rgba(255,255,255,0.75);***

***font-size: 15px;***

***}***

Form Component

Strategy is to create components in separate files and export them to the app.js file

For the form component, create a file madlib\_form.js in components

Import react: ***import React, {Component} from ‘react’***;

Create a class called MadlibForm with render() and return() statements

***class MadlibForm extends Component {***

***render() {***

***return(***

***<h1>Bottega Madlibs Form Component</h1>***

***)***

***}***

***}***  
export: ***export default MadlibForm***

In the app.js file need to import madlib\_form.js: ***import MadlibForm from “./madlib\_form”;***  
 (requires ./ and does not want extension name)

In the app.js return() call MadlibForm: ***<MadlibForm />***