|  |
| --- |
| **Submit the assignment on time.**  You will receive all points or 0 points for this item. Points are based on when the assignment was submitted.  Maximum score 6 |
| **Properly create a GitHub repository for this assignment**  Provide the link for the repository.  Maximum score 1 |
| **Properly use a .gitignore file.**  Ignore the node\_modules folder. The node\_modules folder should not be in the repository.  Maximum score 1 |
| **Properly push main to GitHub.**  The project should be in main.  Maximum score 1 |
| **Properly deploy this assignment to GitHub.**  The assignment should rendered in the github.io domain. Provide the link to the live site.  Maximum score 1 |
| **The page renders without errors.**  There should not be errors in the browser output and developer tools.  Follow THESE INSTRUCTIONS to fix the favicon.ico error: [CLICK HERE](https://codexacademy.moodlecloud.com/mod/assign/$@PAGEVIEWBYID*1685@$)  Maximum score 1 |
| **Properly create a VS Code workspace for this assignment.**  The workspace file should be in the GitHub repository.  Maximum score 1 |
| **Properly indent the code.**  Ok to use a VS Code extension like Prettier to automatically indent.  Maximum score 1 |
| **Properly use naming conventions.**  Camel case variables, functions, and classes. Capitalize class names. Handlers start with handle.  Boolean variables start with a question word. Functions start with a verb word.  Maximum score 1 |
| **Properly use default parameter values to indicate what data type parameters should be.**  Example: event = new Event()  Maximum score 1 |
| **Each module should only have 1 function or class that is exported.**  Maximum score 1 |
| **Module filenames match their module names.**  Example: The function handleSubmit is in a file called handleSubmit.js  Maximum score 1 |
| **Properly use React folder structure.**  src folder with Views folder, controllers, and utils. dist folder adjacent to src folder.  Maximum score 1 |
| **Properly place files in the folder structure.**  The Views folder should contain React components. The utils folder should contain reusable code. The modules folder should contain project specific code. The controllers folder should have code that controls the flow of how modules are used.  Maximum score 1 |
| **Properly create a React function component.**  A function that returns an HTML element (with nested elements).  Maximum score 1 |
| **Properly use a React function component.**  Maximum score 1 |
| **Properly configure Babel.**  Use babel.config.json. Convert JSX to JavaScript.  Maximum score 1 |
| **Properly configure Webpack.**  Use webpack.config.json. Watch for changes in code. Generate a bundled JavaScript file. Automatically use Babel.  Maximum score 1 |
| **Properly use a JavaScript file generated by Webpack.**  The JavaScript file should be connected to index.html  Maximum score 1 |
| **Properly embed a video of a product or service.**  Convert HTML attributes to React attributes.  Maximum score 1 |
| **Properly describe the product or service.**  Use real words. Make it appealing to customers.  Maximum score 1 |
| **Properly list features of the product or service.**  Use real words and descriptions.  Maximum score 1 |
| **Properly use Bootstrap to create a professional looking website.**  Ok to also use SASS and CSS to style.  Maximum score 1 |
| **Properly use a form that collects email addresses.**  Ok to collect names, phone numbers, and other relevant information.  Maximum score 1 |
| **Properly provide feedback that includes information collected by the form.**  Example: Thank you, John. Your information has been submitted.  Maximum score 1 |
| **Properly issue a discount or incentive to people who submit an email.**  Example: Your discount code is 2025DISCOUNT.  Maximum score 1 |
| **Properly process the form.**  Use a handler.  Maximum score 1 |