|  |  |
| --- | --- |
| Reviewer |  |
| Developer (Person who wrote the code) |  |
| Date code received from the developer |  |
| Date review sent to the developer |  |

Instructions

1. The reviewer will complete the “Beta” column of this form. The developer will revise their code based on the review.
2. After revising their code, the developer will fill out the “Prod.” (Production) column of this form to indicate what they have changed.
3. The developer will fill out the “Professional development tools and techniques” table on the last page and insert screen-shots.

|  |  |  |
| --- | --- | --- |
| ***Programming style for all programs*** | **Beta** | **Prod** |
| Is proper indentation used? |  |  |
| Are comments used appropriately? |  |  |
| Do variable names use camel case? (camelCase for example)  Are variables declared with let or const? |  |  |
| Do function/method names use camel case?  Do functions/methods pass parameters and return values appropriately? |  |  |
| Do class names use title case?  Do classes effectively encapsulate the functionality of the “things” in the solution space? |  |  |
| Does the code take advantage of opportunities to use ES6 syntactical elements such as let, const, arrow functions classes? |  |  |
| Content, style and functionality are provided in separate files (.html, .css, .js)?  Inline styles are used only sparingly?  Event handlers are created in JS code not in html? |  |  |

General comments and notes:

|  |  |  |
| --- | --- | --- |
| ***AJAX enabled Bookmark application*** | **Beta** | **Prod** |
| Completed AJAX features of the application?   * Application satisfies all requirements and functions correctly? |  |  |
| Edited the constructor?   * Added instance variables for the web API URL as well as the api key that the application uses to make the Ajax call? |  |  |
| Edited the createBookmark method?   * Used the fetch method to make the Ajax call? * Used the then method of the Promise that’s returned to parse the json when the response comes back. * Used the then method of the Promise that’s returned to retrieve the url as well as the title from the object that was created (from the api call) and adds those elements to the bookmark? * Used the catch method of the Promise that’s returned to log any error to the console? |  |  |

|  |  |  |
| --- | --- | --- |
| ***Weather application*** | **Beta** | **Prod** |
| Completed Weather application?   * Application satisfies all requirements and functions correctly? |  |  |
| Created the constructor?   * Initialized an instance variable named state that contains all “data” for the app? * Initialized instance variables for each of the UI elements used in the app? * Called the method bind to associate the class with the keyword this in methods called in response to events? * Called the method onFormSubmit? |  |  |
| Created the onFormSubmit method?   * Called preventDefault to prevent the form from being submitted? * Retrieved the zipcode from the UI element? * Called fetch to get location based on zipcode from openweathermap? * Called then to parse the response? * Called then and to process the json object? * Stored the city, latitude and longitude in the state instance variable? * Called catch to log any errors to the console or to alert the user? * Called fetch to get weather info for the zipcode from openweathermap   + Formatted the parameter to include the url as well as the apikey? * Called then to parse the data returned from the ajax call when it completes successfully? * Called then to retrieve the data from the json object? * Called parseForecast and stored the result in the state instance variable? * Called renderWeatherList? * Called catch to log any errors to the console? |  |  |
| Created the renderWeatherListItem method?   * Used 2 parameters, weatherDay and index? * Returns a template literal that contains the html for the weather information for one day? |  |  |
| Created the renderWeatherList method?   * Used one parameter, weatherDays? * Created the html for all of the weatherListItems? Used an array function such as map or reduce to process each weatherDay? * Assigned the resulting html to the innerHTML property of the weatherList element on the page? * Added click event handler that calls renderCurrentDay with the index as a parameter to each weatherListItem? * Called the method clearCurrentDay? |  |  |
| Created the renderCurrentDay method?   * Used one parameter, index? * Used a template literal to format the html for more detailed weather information for one day? * Assigned the resulting html to the innerHTML property of the currentDay element on the page |  |  |
| Created the clearCurrentDay? |  |  |
| Instantiated an object after the page has loaded? |  |  |
| EXTRA CREDIT:   * Did significant styling of the UI for the application? * Functionality of the application demonstrates mastery of technical complexity? |  |  |

|  |  |  |
| --- | --- | --- |
| ***Professional development tools and techniques*** | **Beta** | **Prod** |
| NPM was used to manage modules?   * package.json file was created? * webpack.config.js was created? * Modules including webpack, webpack-dev-server, babel, css-loader, style-loader, less-loader and image-loader are installed? * Scripts were added to facilitate development tasks for bundling and hot module replacement while developing using the webpack-dev-server? |  |  |
| **Production** versions are uploaded to citstudent? |  |  |
| URL for Bookmarker Version 2 on citstudent: | | |
| URL for Weather on citstudent? | | |
| URL for Bookmarker Version 2 GitHub repo: | | |
| URL for Weather GitHub repo: | | |

One thing that you learned from completing the lab:

Insert your screenshots below.