# Project Rubric

Your project will be evaluated by a Udacity code reviewer according to the **Restaurant Reviews project rubric**. Please review for detailed project requirements. The rubric should be a resource you refer to periodically to make sure your project meets specifications.

# Get the Starter Code

To begin, go ahead and clone the starter code from **this repository**. The code provided will serve as your baseline to begin development.

# Development Strategy

1. Get a **MapBox API key**

   - If you do not already have a MapBox account, go ahead and create one
   - MapBox API is free to use (no need to provide any payment information). If you would like to use Google Maps, there is a `google-maps` branch of the starter code. You'll need your own **Google Maps API key**

2. **Get the map on the screen**

   - Head over to your account and create a token. Replace the text `<your MAPBOX API KEY HERE>` inside of `main.js` with your key
   - Using Google Maps instead? Replace the text `YOUR_GOOGLE_MAPS_API_KEY` in `index.html` and `restaurant.html` with your own key
   - Now, what about rendering the map for individual restaurants? Check out `restaurant_info.js`. Where would be a good spot to place your API key?

3. **Convert the provided site to use a responsive design**. All responsiveness should be done with CSS (Bootstrap and other CSS frameworks should not be used)

   - Use appropriate document type declaration and viewport tags. Where would you add these tags in the `index.html` and `restaurant.html` files?
   - How does everything look after adding the appropriate tags? Are there any existing style values in `styles.css` that we can immediately fix to make things more dynamic? Hint: how does using a `%` value differ from using a strict `px` value?

- Create a responsive grid-based layout using CSS. Items such as the `restaurants-list` would be a great place to start. What attributes can you add to make sure these items fit a mobile screen (i.e., a smaller viewport)?
- Use media queries that provide fluid breakpoints across different screen sizes
- Use responsive images that adjust for the dimensions and resolution of any mobile device. For a more in-depth look at responsive images, check out our **Responsive Images** course

4. **Implement accessibility features in the site HTML**
   - How can you use attributes like `tabindex` to indicate if (or *where*) an element sits in keyboard navigation order (e.g., by pressing the `Tab` key on the keyboard)? What will be your first element? Your next?
   - Note that most of this HTML is generated by JavaScript functions (e.g., restaurant item HTML). How can we use JavaScript directly to create a tab index for each element desired?
   - What about semantics? Which attributes can we add (and to which page elements) to reflect the element's role?
   - Be sure to make both your main page *and* restaurant detail page accessible for users

5. **Add a ServiceWorker script** to cache requests to all of the site's assets (so that any page that has been visited by a user will be accessible when the user is offline)
   - Only caching needs to be implemented, no other ServiceWorker features
   - Which file(s) would you need to register a service worker? Be sure to have a file ready to handle the service worker
   - After the service worker is properly registered, which event(s) are you listening for? What is/are the response(s)?

6. **You may use external JavaScript libraries**, but UI frameworks may not be used. This includes (but is not limited to) jQuery, React, Angular, Vue, etc.

## Before Submitting

Make sure your code adheres to our HTML, CSS, JavaScript, and Git style guidelines.

- **HTML Style Guide**
- **CSS Style Guide**
- **JavaScript Style Guide**
- **Git Style Guide**

## Version Control

We recommend using Git from the very beginning. Make sure to commit often and to use well-formatted commit messages that conform to our **guidelines**.

## Submission Instructions

1. Push your project to GitHub (be sure to push the `master` branch)
2. On the project submission page choose the option 'Submit with GitHub'
3. Select the repository for this project
   - You may need to connect your GitHub account first

## Still Not Sure How to Begin?

You are welcome to check out these **student-curated resources** in Knowledge!