User Story 1. As a user I want to have a simple way to submit my request so that's it's easy to make a search

1. Search input feature
2. Acceptance Criteria:
    a. A form that allows user to submit an API URL
    b. Once the form is submitted by user (via button), server makes the GET call
    c. Error Handling: if the URL submitted is incorrect, return a clear error message to the user
    d. (STRETCH: more detailed search options)
    e. (STRETCH: ability to get ideas of similar/past/popular previous searches)
    f. (STRETCH: if user types in the URL with their key, we provide an error/warning message)
3. Size: medium

User Story 2. As a user I want see my results in a clear, simple format so that I can easily see the key value pairs

1. Display results feature
2. Acceptance Criteria:
    a. A view of the result (targeting javascript format) that the user can simply copy and paste into their code to get the relevant information
    b. Error Handling: if the there is an issue with a display, they get a clear error message and how to fix the problem (e.g. redirect to home screen to restart search)
    c. (STRETCH: button that saves the text automatically (so they don't have to use ctrl+c)
4. Size: medium

User Story 3. As a user I'd like intuitive and easy to read layout of the site that is mobile friendly so my user can navigate with ease

1. Layout
2. - mobile first content design
    - smaccs css structure
    - JSON text styling
    - Recommended API's styling
3. - breakpoint tests
    - color contrast testing (2 check marks)
    - solicit outside feedback

User Story 4. As a user, I'd like recommendations related to the API's I'm accessing so I can discover new resources.

1. Recommendations
2. - Using the public API project API query and display info about APIs
    - If no results return from the API we will display API's other users have used
3. - successfully communicate with public APIs API

- error handling for no results found

User Story 5. As a user I want to have a button that allows me to save my search results, so that it can be added which can be used by future users for recommendation or reference.
1. Add to API search public history database
2. Acceptance Criteria:
    d. A button that saves the search results into a public database
    e. Once the form is submitted by user (via button), server makes a POST call to update the SQL database
    f. User receives confirmation that the save worked and their search data has been successfully added to the public database
    g. The public database does not include any PII of the person submitting the data to the db
    h. Error Handling: if the save fails, we notify the user that it failed and give steps on how to save (if we know the error, if not provide steps to start the search from the start)
    i. (STRETCH: ideally, we would just take care of any errors for them if we successfully received the API data)
4. Size: medium

User Story 6. As a user I'd love to learn something about the geniuses who created this project and hire them at 20% higher than standard junior dev salary.
1. About Me
2. Separate About Me .ejs linked in the nav bar that renders images of all of our beautiful faces with a small blurb about us
3. Navigate to the page and ensure all content is showing up properly