Week 6 Section QuickCheck

Given the following HTML (only the body is shown), write a JavaScript program that adds a click event handler to the 'Go Fetch' button such that when clicked:

- Gets the current input from the text input box
- Uses AJAX to attempt to fetch a search query from the term given in the input text box. The URL you will fetch from is https://some-fake-api/words. This API accepts a query parameter for searching for a term called word. So one possible request is https://some-fake-api/words?word=hello. It returns a definition in JSON with one key ("definition"). For example, passing word=hello returns the following response: `{ definition : "a greeting" }`
- Uses DOM manipulation to put the definition value into the #results paragraph

To keep it simple, assume that the term that is typed in will be a simple term (no spaces). If an error occurred, place an error message of your choice inside p#error. Use the AJAX fetch syntax introduced in lecture. We have provided $ and checkStatus for you as helper functions.

```html
<!-- Begin HTML -->
<body>
  <h1>Fetch word definitions!</h1>
  <input id="search-term" type="text">
  <button id="fetch">Go Fetch!</button>
  <p id="results"></p>
  <p id="error"></p>
</body>
<!-- End HTML -->
```

Write your JS solution on the back of this page.

```javascript
/* Begin JS */
(function() {
  const URL = "https://some-fake-api/words";

  window.addEventListener("load", initialize);

  // finish the JS code here
  function initialize() {



  }

  // add other functions here:



/* ============================ Helper Functions ============================ */
  function checkStatus(response) {
    if (response.status >= 200 && response.status < 300 || response.status == 0) {
      return response.text();
    } else {
      return Promise.reject(new Error(response.status + ": " + response.statusText));
    }
  }

  function $(id) {
    return document.getElementById(id);
  }
})();
/* end JS */
```