## Project 1 - Build a Movie Lookup Site

**Introduction:**

Using your knowledge of forms and APIs, you’ll build a website to look up information about movies and display the information for your users.

**Tasks:**

1. Create a new local project that uses Vite, Prettier, and ESLint
2. Register for a free [OMDb API](https://www.omdbapi.com/) API key
3. Create a form that lets the user search the API using the following options:
   * API Key (required, string)  
     **Don’t include the API key in your code!** API keys are considered confidential secrets, like a social security number, that uniquely identifies you to the API you’re using. Including them in your code or in your codebase means that anyone who looks at your code can impersonate you, like identity fraud if someone were to steal your SSN. This is true of both backend and frontend development.  
     For this project, by making the API key an input field instead, you are able to keep it secure by only including it when you go to use it..
   * Title or IMDB ID (one or the other required, string)
   * Year (optional, number, greater than 1928, less than or equal to the current year, for whatever the current year is)
   * Short or long plot (required, one or the other)
4. Validate that all fields are correct, displaying error messages for any form input that’s incorrect
5. If all fields are correct, use the inputs to query the API, and disable all form fields and buttons  
   **Make sure you use HTTPS for your API call!** [HTTPS](https://en.wikipedia.org/wiki/HTTPS), as opposed to HTTP, is a secure connection and will encrypt both the URL and the contents, preventing others from seeing what’s sent and received.
6. If there’s a result, display the first movie or show’s title, year, rating, release date, and poster (if it has them) underneath the search form  
   **Don’t use innerHTML to add items to the page!** innerHTML can expose your website to a type of security vulnerability known as [cross-site scripting (XSS)](https://en.wikipedia.org/wiki/Cross-site_scripting) by allowing untrusted HTML to be injected onto your trusted page. Instead, use [document.createElement()](https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement) and [textContent](https://developer.mozilla.org/en-US/docs/Web/API/Node/textContent) to create HTML elements and set the text of elements where needed, and use [append](https://developer.mozilla.org/en-US/docs/Web/API/Element/append) and [remove](https://developer.mozilla.org/en-US/docs/Web/API/Element/remove) to add and remove elements as needed.
7. If there’s no result, display an error message that what they searched for could not be found underneath the search form
8. Whether there are results or not, re-enable all form fields and buttons
9. When a user searches again, remove any error messages or existing search results that may have been displayed from their previous search
10. The HTML, CSS, and JavaScript for the web page should be separate files. Use JavaScript module syntax (ES Modules) for your JavaScript
11. Submit your code by pushing it to GitHub, ensuring that there are no Prettier or ESLint errors

**Duration: 1 Week**