# HTTP5222 Assignment 2 (React)

Using the API you created in **Assignment 1** (the admin site with exposed API endpoints), develop the front-end using React. For example, if you created an API to provide data for your portfolio, develop your portfolio site using React. If you created MongoDB collections (and exposed an API for those) for an e-commerce site or other small business site, develop the site for that using React. (Recall: Previously, in Assignment 1, the only “front-end” pages were the admin pages to make adding to the collections easier. The rest was just the API.) Your site should be styled and be reading the data from your two collections (you can have some hardcoded content to flesh out your pages).

Suggestions:

* To start, you can make a rough wireframe on paper to organize your thoughts as to how you’ll structure your pages.
* Deploy your Assignment 1 API so that you can more easily make requests to it. Make sure your API has cors set up so your API can accept requests from another server.

**Requirements:**

1. Website must be built using React.
2. Website data must be retrieved from a backend API (i.e. your API you created in Assignment 1 using Express). In other words, your React portfolio site will make API requests to your backend API to retrieve the site data.
3. Include important information such as:
   1. If creating a portfolio site:
      1. Your name
      2. Your desired role (e.g. web developer, web designer, full stack developer, etc.)
      3. Your projects including important info about each project such as (for example):
         1. Project name
         2. Screenshot
         3. URL (if applicable)
         4. Short blurb about project
         5. Language listing
      4. Some way to contact you (contact info and/or a contact form).
   2. If creating a different site, consider the important information you’d need and (possible) pages you’d need.
4. Deploy your website. (You can use Vercel. Note that if a portfolio site, using a custom domain is nicer.)

You will be marked on the criteria below. Note that the original rubric is out of 15 (see below) but this has been **scaled up to being out of 18 in Blackboard**.

* **Code quality (5):** Does it work? Does it follow coding best practices? Are there helpful comments? Formatting?
* **Design (5):** Is the design professional? Does the design work for the type of website (e.g. portfolio)?
* **Usability (5):** Is it easy to use? Is it clear what the site is about? Are important pieces of info available?

This assignment is individual and is worth 18% of your final grade. For your code, you can put your React app code on GitHub and just submit your GitHub link so I can view the code. In a README, include the URL to your site online. If you don't manage to get the site online by the due date, include any information I may need to run the site locally.

Due date: **April 18, 2025 @ 11:59pm** (**DO NOT submit this assignment late** as this may result in your assignment not being accepted.)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Excellent**  **(5 – 5 points)** | **Good**  **(4 – 5 points)** | **Satisfactory**  **(3 – 4 points)** | **Somewhat Unsatisfactory (2 – 3 points)** | **Unsatisfactory**  **(0 – 2 points)** |
| **Code Quality** | Code works, is efficient, well-commented and formatted. Code follows best practices. | Code works, is commented and formatted well. Code mostly follows best practices but for a few minor issues. | Code is a little buggy, is not always commented or formatted well but a good attempt. Code not always efficient. | Code is buggy and poorly commented. Some issues following best practices. | Code doesn't work and is not commented. Code has many major issues. |
| **Design** | Design is professional. | Design is mostly professional but with a few minor issues. | A good attempt but feels like student work. | Design is not really complete and is obvious it's student work. | Design not really fleshed out at all. Design is not complete. |
| **Usability** | Website is easy to use and follow and is accessible. It makes sense for the site subject matter (e.g. a portfolio site). | Website is easy to use and follow but with some minor issues. | Website is mostly fine but with many minor issues or at most one or two major issues. | Website has some major issues around usability/accessibility. | Website is confusing/hard to use. Website not accessible. |