1. API Choice

* Browse through the [provided list](https://github.com/appbrewery/public-api-lists) and choose an API of interest. This choice should be guided by the potential to retrieve, manipulate, and present data in a meaningful and interactive way. I recommend choosing an API that does not require authentication and is CORS enabled. ([What is CORS?](https://medium.com/@electra_chong/what-is-cors-what-is-it-used-for-308cafa4df1a))

2. Project Planning

* Think through your project, researching the chosen API, its features, what data it will provide, and how it will be used in your web application.

3. Project Setup

* Set up a new Node.js project using Express.js.
* Include Axios for making HTTP requests.
* Include EJS for templating.
* Ensure that the project has a structured directory and file organization.

4. API Integration

* Implement at least a GET endpoint to interact with your chosen API.
* Use Axios to send HTTP requests to the API and handle responses.

5. Data Presentation

* Design the application to present the retrieved data in a user-friendly way. Use appropriate HTML, CSS, and a templating engine like EJS.

6. Error Handling

* Ensure that error handling is in place for both your application and any API requests. You can console log any errors, but you can also give users any user-relevant errors.

7. Documentation

* Include comments throughout your code to explain your logic.

8. Code Sharing

* Use what you have learnt about GitHub to commit and push your project to GitHub so that you can share it with other students in the Q&A area, I'd love to see what you've build too! You can tweet at me @yu\_angela
* Include a Readme.md file that explains how to start your server, what commands are needed to run your code. e.g. npm i  and then nodemon index.js