**Description**

In the second project for the General Assembly Software Engineering Immersive course, we were asked to **build a React application** that consumes a **public API**. This was a group project.

**You will find the deployed app here:** [FishWatch](https://fish-watch.netlify.app/)

**Getting started/Code Installation**

1. Download source code via the 'Clone or download' button in GitHub.
2. In the CLI navigate to the root and run ***npm i*** to install dependencies for the application
3. Finally, run ***npm run start*** to run the program in your local environment.

**Timeframe & Working Team**

* Timeframe:
  + 1 week
* Working Team:
  + Luke O’Brian

**Technologies used**

* HTML
* CSS
* JavaScript
* React
* Bulma
* Bootstrap

**Brief**

* Needs to Consume a public API.
* The app should include a router -and consist of several "pages".
* Have semantically clean HTML.
* Be deployed online and accessible to the public.

**Project Overview**

* Pair Programming.
* We built a React application using an API which had detailed information about fish stocks in the Atlantic Ocean.
* We decided to show all the different species in an index page and have a sustainable eating section where we offered alternatives to the more popular overfished species.

**Planning**

We wrote down several ideas and highlighted the pros and cons for each project. We also discussed the ideas with our teacher to help us make our final decision.

A picture containing chart

Description automatically generated

We decided that we would build a React application that highlights the effects of climate change and overfishing on fish species in the Atlantic Ocean, we chose this project because the API data was of excellent quality and we both had an interest in the subject matter. We then produced wireframes showing how we wanted our site to look.

Diagram

Description automatically generated

Graphical user interface

Description automatically generated

We did the technical setup together, making sure that all the dependencies were installed before we began work, this entailed, setting up the router and routes for our React application and creating our CSS files for styling.

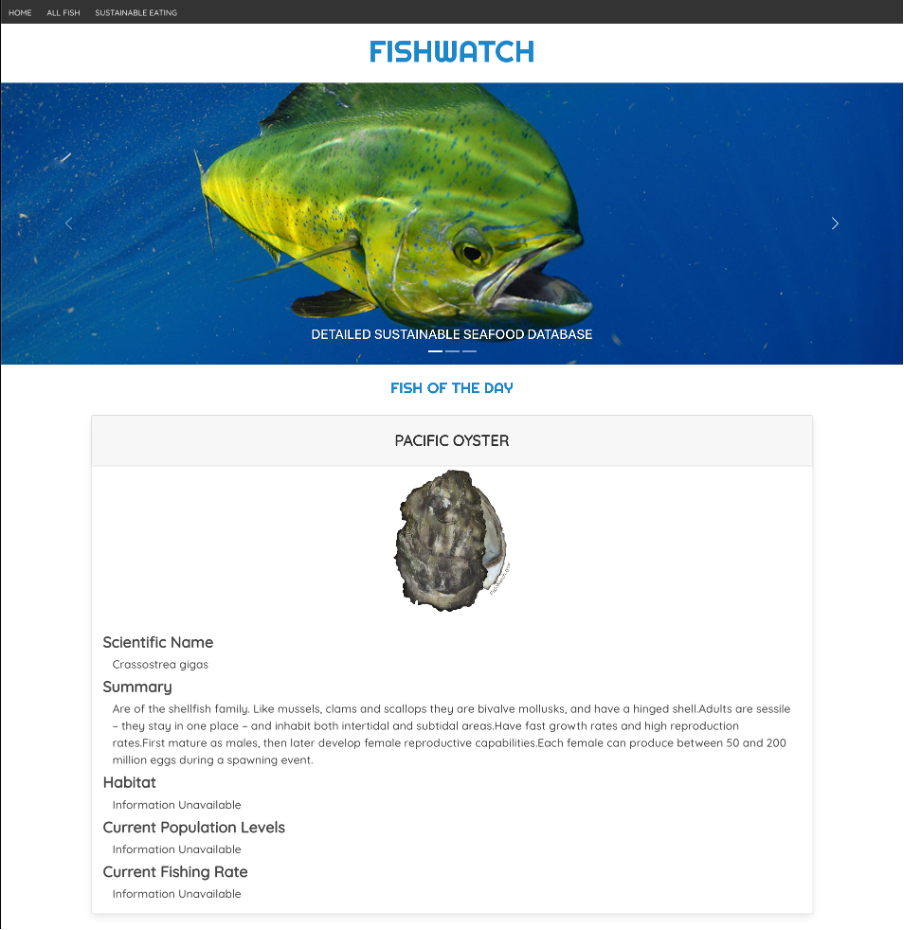
We then decided that I would create the homepage, fish index and the individual fish pages and Luke would work on the fish of the day and the recipe sections. Throughout our project, we made the decision to have daily stand ups to discuss our progress and blockers; if we needed to communicate, we used Slack and Zoom to keep in touch**.** If we encountered any blockers or bugs, we would troubleshoot the issues on Zoom.

**Build Code Process**

We did the technical setup together, making sure that all the dependencies were installed before we began working on our individual sections.

Homepage

* The homepage consists of a Bootstrap carousel and a fish of the day component.



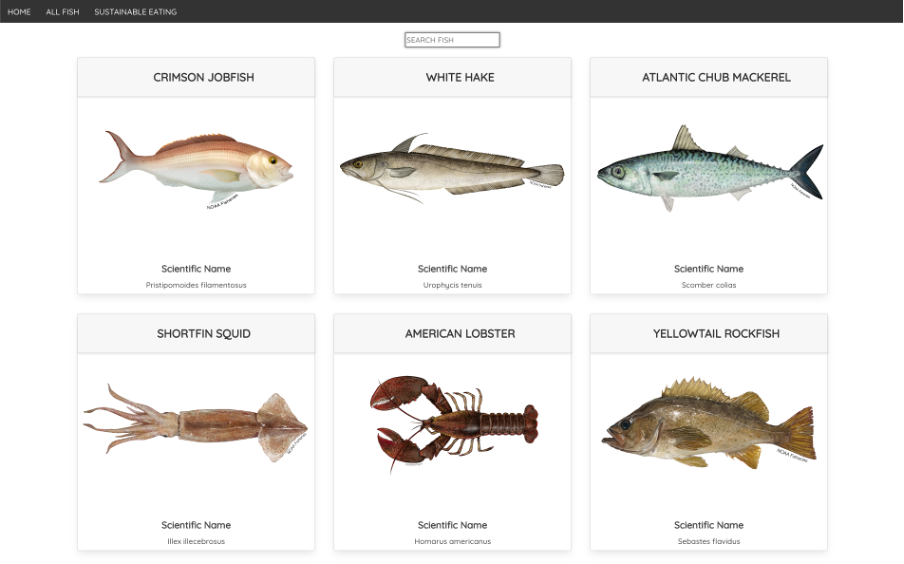
***Featured Code - Setting the fish of the day***

* Firstly, today's date needed to be selected and then stored in localStorage.
* The displayFish state was created using useState.
* The GetRandom() function is then run, this selects a random fish from our API and stores the data in the displayFish state, this is then added to the localStorage.
* The information is sent through as props to the FishDay component so that it can be displayed on our homepage.



Fish Index Page

* The index page consists of a search box and cards showing all the individual species of fish from the public API.



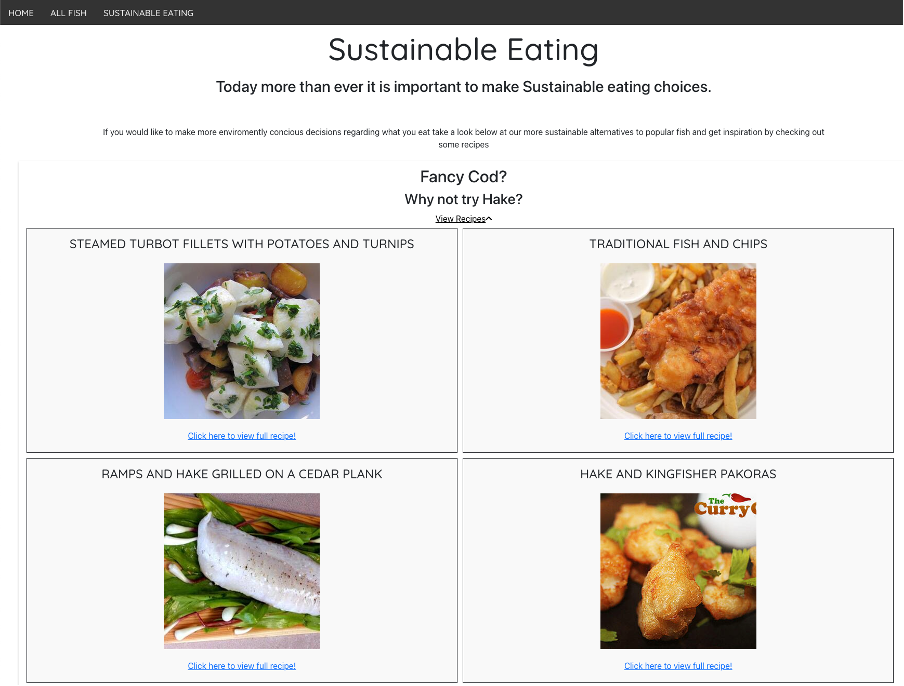
***Featured Code - Searching for specific fish species***

* Search state was created using useState.
* As values are entered into the search input the search state is updated in real-time.
* The filterFish function uses the filter method to return an array which only contains fish that match the search variable.
* This array is then iterated through using the map method, therefore, it only displays fish that match the entered text in the search input, if there is no text it displays all fish.



Sustainable Eating Page

* The sustainable eating page used a recipe API to find delicious alternatives to more popular overfished species.



**Wins**

* Pair-coding was beneficial; I enjoyed creating logic and troubleshooting blockers together. It was great to bounce ideas off Luke, I was used to coding solo, so it was refreshing to have someone who could offer a different perspective and new ideas.

**Key Learnings/Takeaways**

* **React** – This was my first React app, I learnt a lot about how useState and useEffect worked and how to pass properties between components.
* **API’s –** I learnt how to access and use public APIs in an application.

**Bugs**

* Issues with the view alternative recipes in the Sustainable Eating section, it doesn’t always expand and show recipes, could be an issue with the API.

**Future Improvements**

* Make the site responsive.
* Add an additional public API so we can expand the amount of species that the site contains.
* Add a message board or comments section so our users can communicate.