



UNIVERSIDAD
Panamericana

Web Project Proposal

Andrea Carolina Hinojosa Trujillo

Aemi Rosas Sánchez

José Eduardo Moreno Paredes

Jetzuvely Del Carmen González Gutiérrez

Universidad Panamericana

Web Application Development

Gabriel Castillo Cortes and Bernardo Moya de la Mora

August 28th, 2025

Web Project Proposal

Oceans, seas and coastal areas are an essential part of the Earth's ecosystem and are critical for sustainable development. This is more than 66% of the earth's surface, and contains 97% of the earth's water. They are a pillar for human survival, they sustain our oxygen, climate and absorb a significant part of carbon dioxide emissions. Also, they are also home to an extraordinary biodiversity, so vast that much of it remains undiscovered. However, it is not only important in terms of nature, but also economically, culturally and socially.

Clearly, it is something that concerns us all and, that is why, the UN in a sustainable development agenda that world leaders have adopted, considered it as something crucial for human well-being. This agenda consists of 17 goals that have the objective of eradicating poverty, protecting the planet and ensuring prosperity for everybody by 2030. These goals are: no poverty, zero hunger, good health and well-being, quality education, gender equality, clean water and sanitation, affordable and clean energy, decent work and economic growth, industry, innovation and infrastructure, reduced inequality, sustainable cities and communities, responsible consumption and production, climate action, life below water, life on land, peace, justice and strong institutions and partnerships for the goals.

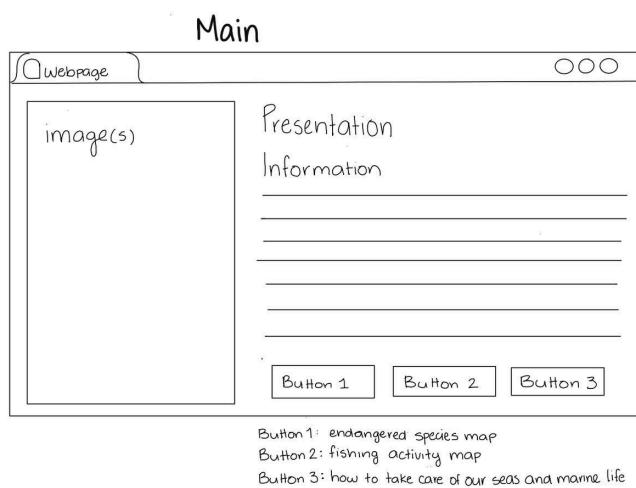
In this context, Goal 14: Life Below Water plays a crucial role, as it focuses specifically on conserving and sustainably using the oceans, seas and marine resources for sustainable development. We believe that this goal is crucial for humanity because, as we can see, the ocean is a resource on which we rely in different areas that are interconnected. For this reason, to protect marine resources and because we are passionate about its biodiversity, we chose to make our contribution specifically to this goal through an awareness website for seafood harvesters.

Protecting life below water is not only about conserving marine species but also about maintaining the balance of entire ecosystems that we need for survival. By focusing on Goal 14, we want to highlight the urgent need for awareness and action, showing that anybody that has contact with marine environments, not just world leaders, can contribute to the health, resilience and long-term sustainability of marine ecosystems. Moreover, we want to demonstrate that it is possible to achieve it through everyday actions that, despite their simplicity, can have a significant positive impact.

The main objective of this project is to reduce the illegal fishing and the extinction of marine species by spreading important information about fishing practices. We aim to provide updated information such as conservation status of marine species and their geographic distribution so that fishers avoid catching them and fishing around those areas, as well as information about the season and locations where it is safe and legal to fish, and about good fishing practices, including sustainable techniques to protect the environment and preserve marine biodiversity.

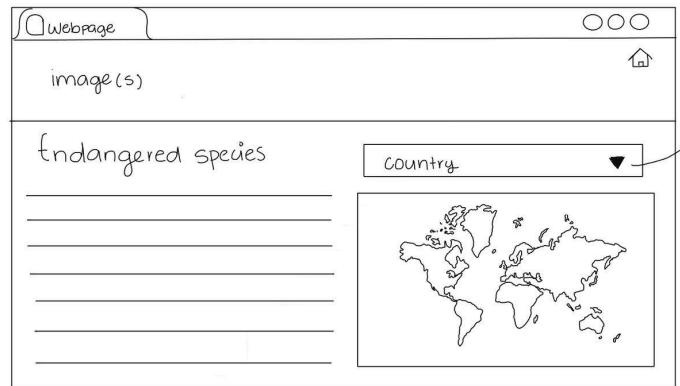
The proposal is to create a Web Application where this information is organized in a user-friendly, interactive and practical way, encouraging fishermen to become interested in contributing to protect oceans and its biodiversity. Each window of this website will be organized in two sections. On one side images or interactive map and, on the other, explanatory text related to the display content. They will also include a menu so that it is easier for the user to navigate between sections.

The main page, on one side, will be conformed by the presentation of the page: our goals, who we are, the importance of protecting the sea, etc., followed by three buttons that will redirect the user to 1) an endangered species map, 2) a fishing activity map and 3) information about how to take care of oceans and marine life. On the other side, there will be images and visual elements related to the information.

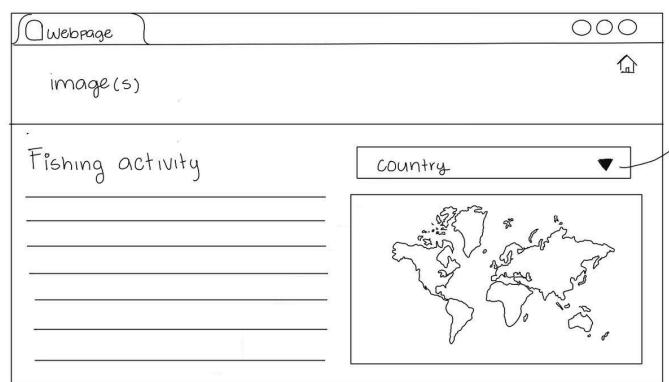


If the user presses button 1, they will be redirected to a window which on one side will show an endangered species interactive map that will contain the geographic distribution of different endangered species represented by dots around the area. Every dot will have a

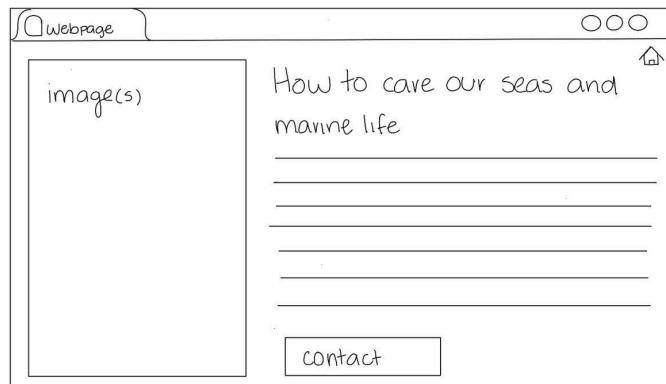
different color, according to the conservation status of the marine species to represent this information in a more visual way. The user will be able to filter the information by area and the map will update dynamically to display only the relevant fishing zones and species distribution. On the other side, at first there will be a general description of the map: its futures, its meaning, and what each color represents. However, when the filter is activated, the information displayed on the map will also be presented in written form.



If the user presses button 2, they will be redirected to another interactive map that will contain information about fishing activity. It will highlight areas protected or restricted due to illegal fishing, as well as the geographic distribution of species allowing the user to identify whether a species they intend to fish is located in a no-fishing zone. This map will also include a filter option by region and in the written part information related to the above will be displayed.



If the user presses button 3, they will be redirected to a section about good fishing practices for responsible fishing. This section will provide guidelines and tips to minimize environmental impact, protect marine biodiversity and comply with local regulations through examples, visual elements and instructions that make it easier for the user to understand and apply these practices. At the end, there will be a mailbox so that people can contact us to learn more about the cause or to share their feedback.



The web application and its technical features will be developed using HTML, CSS and JavaScript. The data will be gathered from free APIs such as the International Union for Conservation of Nature (IUCN) Red List for the species conservation status, the Ocean Biodiversity Information System (OBIS) and the Global Biodiversity Information Facility (GBIF) for their geographic distribution, and the Food Agriculture and Renewable Natural Resources Legislation Database (FAOLEX) for fishing information. Data from these sources will be integrated dynamically to ensure accurate and updated information for users.

The project is expected to have an impact in reducing the capture of vulnerable species and promoting sustainable practices. The information will be presented in an easy, accessible, interactive, and efficient way. Since the data sources are from organizations that specialize in researching the subject, the information will be relevant, clear and transparent. Moreover, since the web page will be in constant contact with the APIs, the information will remain updated. This is especially important nowadays, as it is often difficult for users to find reliable sources.

This project will be a great opportunity for people to get engaged and it could act as a platform for organizations to unite around the cause. Therefore, it contributes not only to achieving goal 17 but also indirectly to other goals. Through the fostering of collaboration, real awareness about the importance of water bodies and responsible fishing, this project seeks to nourish a long-term positive impact. By the integration of a sustainable development approach, that will guide humanity to live in harmony with nature, securing the conservation and sustainable use of the oceans and seas.

In our times, the ocean, seas and marine life face an unprecedented threat, which is a direct endangerment to the health of our planet. Overfishing, pollution, habitat degradation and climate change are causing disruption in fragile ecosystems, placing countless species at risk of extinction. Addressing these threats requires human collaboration from governments, organizations, scientists and communities. By working together, establishing common goals and promoting sustainable development practices, we can decelerate the decline of marine biodiversity and ecosystems to recover it. Steering toward a culture of protection and recovery is important to address today's challenges. Only through collective action can we ensure that the conservation and nourishment of marine ecosystems can be effective and generate an impact world-wide.

Human well-being is undeniably connected to the Earth's ecosystems. Humanity's dependence and its lack of responsibility on natural resources and ecosystem services cannot be overstated. Water bodies are responsible for the most amount of oxygen produced on Earth, as well the production of food for millions of people. Despite this, resources are limited, and we must manage them responsibly. By denying this responsibility, humanity has placed countless habitats in extreme danger and driven numerous species to extinction. Caring for aquatic environments also means caring for our own well-being. The conservation of marine ecosystems is a fundamental key to the survival and prosperity of humankind.

References

United Nations. *Sustainable Development*. (n.d.). <https://sdgs.un.org/>

Union for Conservation of Nature. *The IUCN Red List of Threatened Species*. (2025).

<https://www.iucnredlist.org/>

Food and Agriculture Organization of the United Nations. *FAOLEX Database*. (2025).

<https://www.fao.org/faolex/opendata/en/>

(n.a). GBIF API Reference. (n.d.).

<https://www.gbif.org/es/network/2b7c7b4f-4d4f-40d3-94de-c28b6fa054a6>

UNESCO. Ocean Biodiversity Information. (n.d.). <https://obis.org/>



UNIVERSIDAD
Panamericana

Project prototype: Development log

Andrea Carolina Hinojosa Trujillo

Aemi Rosas Sánchez

José Eduardo Moreno Paredes

Jetzuvely Del Carmen González Gutiérrez

Universidad Panamericana

Web Application Development

Gabriel Castillo Cortes and Bernardo Moya de la Mora

September 9th, 2025

Development Log

Team

Day 1: Distribution of tasks

We started by distributing the tasks to every team member. Eduardo was going to make one map window, Aemi the other map window, Andrea the recommendations window and Jetzuvely the main window.

Day 2-Day 5: Sections development

Everyone did their part separately, but considering a style defined on day 2: font-style, color-palette; and a defined structure: as close as possible to what is stated at the webpage proposal document. Even so, each of us was experiencing coding problems because the code was not working as we expected. At the end, we managed to solve the problem by debugging and looking at the presentations, internet information and IA resources.

During these days, we realized that developing the structure for the two map sections was almost the same. The only thing that changed was the information because the maps are about different topics. That's why we decided to copy one map source code and adapt it for the other map according to their specific needs. Therefore, we decided to assign both map sections to Aemi and let Eduardo put it all together in a webpage. He was in charge of bringing everything together in such a way that it was coherent in style and functionality.

Day 6: Webpage integration

Eduardo integrated our individual pages into a single website, but we had some problems with this part. We didn't know how to use GitHub with different accounts. Eduardo explained to us how to do it and made a mini tutorial, so we managed to bring the maps and recommendations sections together. However, there were still some problems with uploading the main page and we spent hours trying to do so, until Eduardo gathered the files through another means and then uploaded them to our repo's website.

Day 7 (deadline): Documentation and final details

We completed the final adjustments to our website. When integrating the parts together, the webpage stopped working correctly. We solved the problem by debugging the code and we figured out that the problem was in the links, so we wrote them correctly and the webpage started to work again.

Also, we wrote our personal work reflection and the problems/solutions we faced as a team during the development.

Andrea

Personal work reflection

Regarding the website development, I did the recommendations section. I searched for information about sustainable fishing practices, organizations that contribute to the cause, and the place where people can find information about their area's fishing regulations. I attempted to be congruent with the style that my teammates incorporated to the other windows.

I also learnt about the implementation of the bootstrap grid system in the code and how it helps to organize the webpage better, since it is organized by rows and columns and has attributes that contribute to efficient style development. Also, although we didn't implement a dropdown list for the menu because other methods looked nicer, I learnt how to make a dropdown menu with a bootstrap grid system class.

On the other hand, about teamwork, I consider that I actively participated in the progress of this delivery because not only did I send my part, but I also was following the progress that we were making and I let my team know that they could count on me if there were any further problems. Thus, I tried to maintain communication with the team.

I could have improved my organization skills to deliver my part earlier so that my team could have time to work unhurriedly.

Team work reflection

My teammates actively contributed to the assignment by delivering before the deadline their part of the work. We also communicated throughout the development of this delivery and tried to solve problems together.

Aemi

Personal work reflection

For this project, I was responsible for creating the pages related to maps: Endangered Marine Species and Fishing Activity. I worked on the basic structure of the pages using HTML and applied fundamental CSS elements, like defining colors, layout, and the design structure.

In addition, I also did the research to gather accurate and reliable information for both pages, making use of trustworthy sources such as the FAO, NAA, and other recognized organizations. I also contributed ideas to enhance the visual identity of our website, like suggesting the font style for titles and the color palette.

Through this work, I was able to improve my skills in web development, particularly in CSS, and I also learned how to use Bootstrap. Working with my teammates allowed us to build a functional prototype.

I think that my contribution helped to ensure that our website is informative. Although some information is still missing, the essential content and purpose are clear.

Team work reflection

I think that our organization throughout the project was very good. Everyone had a clear role and contributed their part, which made the work more efficient and balanced and we helped each other with our doubts.

Eduardo

Personal work reflection

During this project, my main responsibility was merging all the pages that had been developed. To achieve a smooth merging, I explained to my teammates the basics of GitHub, showing them how to upload their changes and push updates to the main branch. Once all the files were in place, I updated the code references to connect everything properly, ensuring smooth navigation and consistency across the site. This step helped the application feel more cohesive and user-friendly.

I was also responsible for adding headers and footers to every page. To achieve this I used a Header and Footer in html, defining their styles in css. And adding a fade effect to show an introduction image into every page, that once it's scrolled would make the header opacity go higher .The headers and footers helped to connect the pages and make them feel more cohesive. Including these design features made the overall project more polished and aligned with standard web development practices.

Team work reflection

I was quite happy about our team's results, and working dynamic. Each member contributed meaningfully, and together we produced a high-quality outcome. We may need to work a little more into getting to know Git better.

Jetzuvely

Personal work reflection

I'm working on a new idea in a different style because I prefer and love design more, that's why I opt for a 3D style, I do this style with a video tutorial and look for information about things that I considered very important not only for this project but in general because I decided to add the stories and partners and educational videos. I did it with div and img, one scrip for java for the scroll in the

website, videos and some classes for the design and more efficiently, for example, the title class is very comfortable because it is reusable for some parts of the project. My part was the main page and the first page to start, at the beginning they were very difficult because in the css I want something newer and more interactive and for this I used a combination of tools fonts and sizes colors and more the html is easier because it is more information and classes, the real challenge for me is in the css. (I started on the weekend and finished my part on Monday, the first day I started with the html the skeleton of the website and then defined the look and found it in the css design)

Team work reflection

I think my team's work is incredible. Each person contributes different and important things and good ideas to the design. The name and images are very beautiful. My entire team works together perfectly, they are very punctual and communicate well, the codes are very good.



UNIVERSIDAD
Panamericana

Project prototype: Use of AI report

Andrea Carolina Hinojosa Trujillo

Aemi Rosas Sánchez

José Eduardo Moreno Paredes

Jetzuvely Del Carmen González Gutiérrez

Universidad Panamericana

Web Application Development

Gabriel Castillo Cortes and Bernardo Moya de la Mora

September 9th, 2025

Use of AI report

Date: 09/09/2025

Project/Module: Project Prototype

Goals for the documented section:

We planned to ensure that our project would be both functional and visually attractive. Our goal was to have a clear vision of how our website would look by adding buttons, images, and some of the most important information. We also wanted the design to be engaging, so that the users could easily navigate and understand the content.

Task completed:

We worked on the prototype by using essential elements to build the structure of our website. However, we had some problems and challenges during the development process. At first, we didn't understand how to properly use Bootstrap, especially when it came to making specific changes we wanted, like adjusting the position of some elements, for example, moving the map more to the left. Also, we were unsure how to share and collaborate through a GitHub repository. Despite these challenges, we solve the problems by searching for solutions on the internet, using AI tools, and helping each other.

Use of Gen AI:

Prompt: “What is the Bootstrap Grid System?”

Output: “El **Bootstrap Grid System** (sistema de cuadrícula de Bootstrap) es la forma en que Bootstrap organiza y acomoda los elementos en una página web.

👉 Básicamente es una **rejilla** (grid) que divide el espacio de la pantalla en **12 columnas**.

Con esas columnas puedes decidir qué tanto espacio ocupa cada elemento y cómo se distribuyen en distintos tamaños de pantalla (computadora, tablet, celular).”

Its advantages are that its grid system automatically adjusts layout for different devices and screen sizes without extra effort. It also ensures your website looks clean and consistent across different browsers. And you can override Bootstrap's default styles with your own css, or even download custom builds with only the features you need.”

Edits: Added more columns and rows, commented the code so that the line can be more understandable. The output gave specific padding and margin, but we also changed the methods. It was only used as a layout to start the bootstrap grid system.

Reflections:

We learnt that you can put a pre-established background, that it helps us to not have to use CSS from 0, the classes make it easier to establish padding and margin, and it gives structure to the webpage.

Prompt: “*Why is my image on the wrong side?*”

Output: “*El problema no está en la imagen en sí, sino en la estructura del grid de Bootstrap.*”

Edits: We read the solution that AI was giving us and modified it on our code while manipulating it to find out how it affects the functions.

Reflections: We were having problems with the positioning of the elements because, although we were writing on the code that we wanted some element in some place, the element wasn't doing what we expected. After looking for the problem in the code and not succeeding, we asked ChatGPT and helped us with the problem. We learnt that if an image has absolute position, bootstrap won't recognize it.

Resources:

Bootstrap web page: <https://getbootstrap.com/>

Blackboard Presentations

Youtube tutorials

What we learned:

We learnt how to use the bootstrap grid system regarding the general structure of the page, how to use the bootstrap system to make sections that appear as something very relevant, and how to make headers and footers. Bootstrap is a very efficient system because it is very easy to add classes that would be a little more complicated to modify with CSS. Also, it brings the possibility to make layouts faster than making them from scratch with CSS.

Challenges and bugs:

The biggest challenges we faced were when we couldn't accommodate things as we wanted because, no matter how much we changed the code, the element didn't behave as expected. But, with AI tools, we were able to find the problem, understand it and find solutions. It is easier to solve a problem when you know what the problem is. Also, we had problems remembering how to write some lines of code, but we managed to do it by checking the presentations.

Next steps:

After this delivery we can write mistakes that we make and how we solved them, but in a very specific way, so that we can have a “dictionary of resources” and start remembering how to solve the problems in future occasions. In that way, we will be able to rely less and less on AI to find bugs.

Reflection:

As a team, we believe that our collaboration was very good. Each member contributed in different ways, whether through research, coding, design, or problem-solving, which allowed us to combine our skills and complete our objective of this delivery.

One of the most important lessons we learned as a team is that communication is essential. By constantly sharing progress, doubts, and ideas, we were able to stay organized. We also realized the value of patience, since not everything worked as we expected at first, and we needed to act quickly.

This delivery of the prototype helped us to learn and strengthen our skills of HTML and CSS, while also giving us the opportunity to check, share, and validate our research among team members.



UNIVERSIDAD
Panamericana

Beta version: Development log

Andrea Carolina Hinojosa Trujillo

Aemi Rosas Sánchez

José Eduardo Moreno Paredes

Jetzuvely Del Carmen González Gutiérrez

Universidad Panamericana

Web Application Development

Gabriel Castillo Cortes and Bernardo Moya de la Mora

October 21st, 2025

Development Log

Team

Day 1: Distribution of tasks

We started by distributing the tasks to every team member. Eduardo was going to make one map window, Aemi the other map window, Andrea the recommendations window and Jetzuvely the main window.

30/09/2025 Investigation about the best map tool

Andrea and Aemi

We started by exploring different mapping tools. Our first idea was to use Google Maps because it is widely known and offers many features. However, after researching its documentation, we found out that it requires payment once a certain usage limit is reached. Since we needed a completely free option for our project, we decided to discard that idea.

Next, we discovered OpenLayers, an open-source mapping library. We tested it, and we successfully displayed the map on the page. Although we still need to test how it works with an API, this was a big step forward in the development process.

Challenges: It was finding a mapping tool that was both functional and free. Many of the most popular options required a paid plan, which limited our choices.

Solution: We decided to use OpenLayer, but we needed to test it with an API.

02/10/2025 API Research

Andrea and Aemi

We spent the session researching APIs that provide information about endangered marine species. At first, we realized that there weren't many available options, and we also did not fully understand how to access or obtain an API. To keep track of our research, we wrote down several possible options that could be useful to our project.

Since we hadn't learned how to properly get and use APIs yet, we asked for help. Aemi's dad explained to us how the process works and guided us on where to look. Thanks to that, we were able to choose the IUCN Red List API as the main data source for our endangered species web page.

Challenges: It was not knowing how to find or access APIs. This limited our progress at the beginning of today's research.

Solution: We decided to ask someone with more experience in the topic to guide us. With that help, we understood the process better and successfully selected a suitable API.

09/10/2025 - 16/10/2025 Implementation of endangered map

Andrea and Aemi

Goals for the documented session

The goal of this session was to improve our map integration by making it more efficient and user-friendly. We also aimed to filter the API data to make it more manageable and relevant to our project's focus.

Task Completed

We decided to switch the map, since Leaflet is simpler to use and works directly with JavaScript. This change allowed us to handle map functionality more easily and with better performance.

While testing the API, we noticed that it contained a massive amount of information, but the data wasn't well-organized. Because of this, the map took more than five minutes to load when displaying all endangered animals worldwide. To solve this, we decided to limit our map to filter the data to include only species that are in critical conservation status and only in Mexico. This reduced the dataset to just 53 animals, which improved the loading time significantly.

At first, we built the map in a separate project to make testing easier. The challenge came when we needed to integrate it with our Endangered Page, but after some adjustments, we managed to connect everything correctly.

Finally, we used AI to help extract and assign the correct latitude and longitude values for each species, allowing us to accurately plot them on the map.

Challenges

- The original API contained too much unorganized information.
- The map took several minutes to load due to the large dataset.
- Integrating the map project into the main page required additional effort.

Solutions

We reduced the dataset by focusing on critically endangered species in Mexico and switched to Leaflet for simpler and faster map rendering. We also used AI tools to efficiently locate the geographic coordinates for each species.

Use of AI

Prompt: "Give me the latitude and longitude coordinates to each endangered animal"

We needed to find a way to assign accurate latitude and longitude coordinates to each endangered animal so they could be correctly displayed on the map. Doing it manually would have been too time-consuming.

Output: The tool gave us the coordinates.

Edits: After generating the data with AI, we organized it to fit the correct format for our JS map. We then verified that each point matched the right species before adding them to the map.

Reflection: Using AI saved us a lot of time for the coordinates and made the process more efficient.

Resources used

Leaflet

IUCN Red List API

The help of Aemi's dad for specific doubts of API implementation

Next Steps

Improve the visual design and interactivity.

What I learned

I learned how to use Leaflet to create dynamic maps with JavaScript and how to filter large API datasets to improve performance. I also learned how to integrate a separate project into a larger web page and how AI can be used to automate data organization.

10/10/25 – Research on Different Methods for Designing Maps

Jetzuvely Gonzalez

During this day, I spent time researching different ways to design maps for fishing activity. I explored several tools and platforms to understand their capabilities. One of the tools I found most visually appealing was Figma, which allows creating map designs using plugins like Map Maker. I realized that Figma is mainly a design tool and doesn't offer full interactivity, but it is very useful for planning the layout, colors, and overall style of maps.

I also looked into more interactive approaches, like using Leaflet and JSON (GeoJSON) data. Eduardo explained how JSON can be used to structure map data and make it interactive. This approach seemed more practical for real applications because it allows the map to display actual data dynamically without relying on platforms like Google Earth, which can be more complex to use.

Throughout this research, I compared the advantages and disadvantages of each method. Figma is great for aesthetics and visual planning, while JSON + Leaflet provides functionality and real data interaction. I wrote notes about how each tool could be used depending on whether the goal is visualization or interactivity and the use of API, I think is more difficult.

Later, I reviewed all the information I collected and reflected on how these methods could influence the final design of our maps. I focused on aspects such as title placement, layout, and responsiveness for mobile devices. I noted that some designs look good on a computer but overlap or lose clarity on smaller screens.

Although I did not implement any maps myself, this research helped me understand the workflow of designing maps, from aesthetic planning in Figma to functional implementation using JSON and Leaflet. I also considered how structured data can improve map usability and help communicate important information clearly. Overall, this process improved my knowledge of map design methods and allowed me to evaluate different options for future projects.

18/10/25 Fishing Activity Page Enhancement with Leaflet and EJS

José Eduardo Moreno Paredes

Project / Module: Catch the Change - Web Development

Goals for the documented session

- Complete map integration and refactor common components
- transform header and footer into reusable EJS components
- transform htmls into EJS templates
- add map functionality to fishing activity page
- add necessary routes in app.js
- add required dependencies if any
- update of project structure
- creation of commandsRequired.md file
- add devLog.md file for future documentation of development process
- add geoJson files for
- map functionality
- removal of unnecessary files (map.js)

List what you planned to accomplish. (e.g)

- [:)] Complete map integration and refactor common components
- [:)] transform header and footer into reusable EJS components
- [:)] transform htmls into EJS templates
- [:)] add map functionality to fishing activity page
- [:)] add necessary routes in app.js
- [:)] add required dependencies if any
- [:)] update of project structure
- [:)] creation of commandsRequired.md file
- [:)] add devLog.md file for future documentation of development process
- [:)] add geoJson files for
- [:)] map functionality
- [:)] removal of unnecessary files (map.js)

Describe what you actually worked on. (e.g)

In this development session, the project was refactored to improve maintainability and functionality. The header and footer were transformed into reusable EJS components, and all static HTML files were converted into EJS templates.

Full map functionality was added to the Fishing Activity page using GeoJSON data, with corresponding routes updated in app.js and unnecessary files like map.js removed. The project structure was reorganized, required dependencies were added, and documentation files commandsRequired.md and devLog.md were created to support future development.

Document how you used Gen AI (e.g., Copilot, ChatGPT, etc.)

Prompt: "I want to connect my GEOJSONs into my ejs element called map. The geojsons are files that contain the information for the commercial zone in mexico and two geojson about the protected areas"

Output: AI provided two implementation suggestion options:

1. Frontend approach using fetch() to load GeoJSON files directly
2. Backend approach using fs.readFileSync() to pass data through EJS templating

AI recommended Option 2 for better performance and provided complete code examples for both app.js route handler and map.ejs integration using Leaflet.js

Edits:

- Essential css for map height added to map container
- GeoJSON data loaded server-side in app.js using fs.readFileSync
- Added inline styles and CSS classes to legend color indicators for visual distinction
- Moved console.log statements earlier for better debugging

Reflection: The back-and-forth helped me understand:

- How EJS templating variables are passed from Express routes (res.render with data object)
- The <%- %> syntax for injecting unescaped content (necessary for JSON)
- Why Leaflet maps require explicit height styling to render
- The trade-off between server-side file reading vs client-side fetching

What I Learned:

- Server-side data loading pattern: read files → parse JSON → stringify → pass to template → parse in frontend
- Leaflet.js GeoJSON layer integration with styling, popups, and controls
- CSS requirement: map containers need explicit height or they render at 0px
 - EJS variable scope: variables must be passed through res.render() object to be accessible in templates
 - Debugging approach: using console.log to verify data loading before rendering

19/10/25 – Page integration and hosting

Aemi Rosas Sánchez

During this day, I integrated our Endangered Page into the main project. Once the pages were connected, I created the Host Page to display everything in one place. I use Render to make the live link of our project.

However, while testing the site, I encountered an error in the Recommendations Page. After debugging, I discovered that the issue was caused by a capitalization error. One of the EJS files had a capital letter in its filename, while the references in the code used lower case letters. Once I fixed the file name to match, the page loaded correctly, and everything worked.

Challenges: I faced a technical issue caused by inconsistent file naming (uppercase vs lowercase). This small detail caused the page not to render properly, and it took some time to identify the source of the problem.

Solutions: We fixed the capitalization error in the EJS file name to match the references in the project, which resolved the issue and allowed the page to load successfully.

20/10/25 – Final Design and Adjustments

Jetzuvely Gonzalez

On this day, I focused on making the final adjustments to the design of the main page, the maps, and the titles. I reviewed the layout carefully and identified small details that could improve the overall look and user experience.

One of the main tasks was ensuring that the design works well on mobile devices. I adjusted some sections so that text, images, and map elements do not overlap or appear too small on smaller screens. This involved checking spacing, resizing certain elements, and improving alignment.

Although the core design was already defined, these small changes were important to make the page feel polished and professional. This process helped me understand how small adjustments can greatly impact usability and visual balance. In the end, the main page, maps, and titles are more coherent, readable, and responsive for different devices.

20/10/2025 Setting Up The Contact Form

Andrea Hinojosa

Goals for the documented session

- Search for solutions to send emails through the forms so that people can contact us.
- Allow the forms to send emails
- Send a nice message of confirmation

Tasks Completed

- Search for solutions
- Allow the forms to send emails
- Send a nice message of confirmation

Use of Gen AI

PROMPT 1

Prompt: *"How to set up forms in a webpage to send to gmail."*

Output: **Option 3: Backend with Node.js + Nodemailer (For Full Control)**

Edits: I modified some of the given colors to make them clearer and more legible, and modified the font so it didn't look so out of place..

Reflection: It helped me get a general idea of which design suited him best.

PROMPT 2

Prompt: *“Can you explain it to me step by step?”*

Output: Great — your server is tidy and nearly ready. I'll walk you through, step-by-step, exactly what to change/add so your Render app accepts form submissions and sends them by Gmail using **nodemailer**. I'll keep everything in CommonJS to match your code.

// --- Add near top of file, after other requires ---

```
const nodemailer = require("nodemailer");
require('dotenv').config(); // only if using .env locally (safe to keep)
```

// --- Add body parsers if not already added ---

```
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
```

// --- New route to handle form submissions ---

```
app.post("/send", async (req, res) => {
  try {
    const { name, email, message } = req.body;

    if (!name || !email || !message) {
      return res.status(400).send("All fields (name, email, message) are required.");
    }
  }
```

// Configure transporter using Gmail and App Password

```
const transporter = nodemailer.createTransport({
  service: "gmail",
  auth: {
    user: process.env.GMAIL_USER,
    pass: process.env.GMAIL_PASS
  }
});
```

// Build email

```
const mailOptions = {
  from: `Website Contact <${process.env.GMAIL_USER}>`, // shows as from your
Gmail
  replyTo: email, // reply goes to the user who filled form
  to: process.env.GMAIL_USER, // you receive the message
  subject: `New contact form message from ${name}`,
  text: `Name: ${name}\nEmail: ${email}\nMessage:\n${message}`,
  html: `<p><strong>Name:</strong> ${name}</p>
<p><strong>Email:</strong> ${email}</p>
<p><strong>Message:</strong><br/>${message.replace(/\n/g, "<br/>")}</p>`
```

};

// Send mail

```
await transporter.sendMail(mailOptions);
```

```
return res.status(200).send("✅ Email sent successfully.");
```

```
} catch (err) {
```

```

        console.error("Error in /send:", err);
        return res.status(500).send("✖ Failed to send email.");
    }
});

```

Edits: I stopped at the middle of this step because I realized the app somehow would have access to my account and I don't want my information to be exposed.

Reflection: Understanding what chatGPT says to you is important because imagine what would happen if I had simply done what the AI said without being aware that my data could be exposed.

PROMPT 3

Prompt: I am concerned that my data may be exposed with this method.

Output: We can use **Ethereal Email**, a **fake SMTP testing service**. It's designed exactly for development/academic projects:

```

const nodemailer = require('nodemailer');

app.post('/send', async (req, res) => {
    const { name, email, message } = req.body;

    // Validate form fields
    if (!name || !email || !message) {
        return res.status(400).send('All fields are required.');
    }

    try {
        // Create a test account on Ethereal
        let testAccount = await nodemailer.createTestAccount();

        // Create a transporter using the test account
        let transporter = nodemailer.createTransport({
            host: 'smtp.ethereal.email',
            port: 587,
            auth: {
                user: testAccount.user,
                pass: testAccount.pass
            }
        });

        // Define the email content
        let info = await transporter.sendMail({
            from: `Academic Project <${email}>`, // from the form
            to: testAccount.user, // you "receive" it in the test account
            subject: `New message from ${name}`,
            text: `Name: ${name}\nEmail: ${email}\nMessage:\n${message}`
        });
    }
});

```

```

    // Send the preview URL back to the browser
    console.log("Preview URL:", nodemailer.getTestMessageUrl(info));
        res.status(200).send(`✓ Email sent! Preview it here:
${nodemailer.getTestMessageUrl(info)}`);
    } catch (err) {
        console.error('Error sending email:', err);
        res.status(500).send('✗ Failed to send email.');
    }
});

```

Edits: I added this part to the code in the app and adapted it to the real variables that the forms have.

Reflection: I didn't imagine there was a way to do it just as a test.

Prompt: I want the data sent from the form to be saved so that I can see it later because Ethereal only displays it temporarily.

Output://① Save submission locally

```

const submissions = JSON.parse(fs.readFileSync(submissionsPath, 'utf8'));
submissions.push({ name, email, subject, message, date: new Date() });
fs.writeFileSync(submissionsPath, JSON.stringify(submissions, null, 2));

```

Edits: I added this part to the code in the app and added the URL also to the JSON file.

Reflection: I thought it would be harder to send the email, but I'm happy that it would not.

Prompt: I want to open a popup window if the mail is sent, so I have to send the response to the front end, right?

```

Output: const formData = new FormData(this);
const data = Object.fromEntries(formData.entries());
try { //converts the info to json file and sends it to the server
const response = await fetch('/send', {
method: 'POST',
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify(data) });
if (result.success) {
    alert(result.message); // simple popup
    this.reset(); // optional: clear the form
} else {
    alert("Error: " + result.message);
}
} catch (err) {
    alert("Error sending request");
    console.error(err);
}
});

```

Edits: I added this part to the eventlistener part in our code.

Reflection: This change makes the code cleaner and allows the data to be sent to the server in a structured way, without having to create a variable for each field in the form.

Prompt: how can I activate a modal from bootstrap?

Output:**using JavaScript manually** (to control it programmatically).

```
const modal = new bootstrap.Modal(document.getElementById('exampleModal'));
document.getElementById('openModalBtn').addEventListener('click', () => {
    modal.show(); // opens the modal
});
```

Edits: I combined the last prompt with this one. Instead of showing an alert, I showed the modal with `moda.show()`.

Reflection: Sending a modal instead of an alert makes the UI prettier and I learned how Bootstrap can be activated: by Javascript (coding) or html (using `data-bs-toggle` and `data-bs-target..`). The best option was javascript because I don't want to press a link or target to activate it.

Prompt: “[Code] The client prints “Error sending request”, but the server says "Received data”.

Output: Perfect, that means that the server is receiving and processing the info, but the client is failing to read the response... [] Check in the browser console if the request reaches the browser correctly and what the server replies.

Edits: I created a new modal named modal.

Reflection: The browser is very literal in what the problem is and that is very helpful. I should stop to see it before freaking out because it didn't work.

Resources Used

Ethereal Email

Nodemailer

What I Learned

-There are many ways in which you can send emails through a website

-The order in which the modules are set is important for the app to start correctly

-Sometimes there are errors that you have until they are magically solved. I had an error “submissions variable is not defined” and there was not a variable with that name. Suddenly, without any changes in the code, when I started the server, it decided to work. I mean, of course it was not magic, probably the file had not been updated or something like that.

-You have to be very careful with the variables, I usually have mistakes in having different names so they are not recognized by the server and I don't realize.

-I learnt about the node's module that is used to send emails from web pages in an easier way.

-I didn't know anything about SMTP which is Simple Mail Transfer Protocol and allows emails to be sent by the internet. I didn't go that deep into that world, just enough to understand the code, but I had never thought about how emails are sent and now I know it's through this protocol.

-There is a better way to send all the information from the form fields

-There's a better way to send all the information from the form fields, because with `FormData(this)`; everything is sent at once.

-`Object.fromEntries` converts data into a hashtable.

-application/json is the format in which the data is sent to the server. I already knew how to send it as html format, but didn't know how to do it as a json.

Challenges & Bugs

- I didn't want to expose my gmail.
- Used a fake SPTM that is used for tests.
- When modifying the code to add the futures there were errors I didn't understand.
- Debug the code correctly.
- The modal wasn't showing. The client printed an error sending request, but the server printed "received data".
- I asked chat to help me and told me that the problem was not the server, but the client. Since inside the try was more steps than just sending the request, if something else failed, it also sent the error, which was happening. Chat told me to check the browser console and it said "modal undefined", which meant that I didn't have a variable named modal, so I added it.
- The modal wasn't showing in the public website, but it was working well in the localhost.
- It was because sending the fake email interfered with the functionality, so we decided to look for more options to send the email. For now, the info that the user sends to us is saved into a json.

Resources Used

Ethereal Email

21/10/2025 Solving Design Problems

Andrea Hinojosa

Goals for the documented session

- [] Maintain fishing activity map into the bootstrap container
- [] Center the map into the div
- [] Copy the changes of the color of the nav-bar in the main page to the other ones
- [] Make the recommendation page responsive

Tasks Completed

- [] Maintain fishing activity map into the bootstrap container
- [] Center the map into the div
- [] Maintain fishing activity map into the bootstrap container
- [] Center the map into the div
- [] Copy the changes of the color of the nav-bar in the main page to the other ones
- [] Make the recommendation page responsive

Use of Gen AI

PROMPT 1

Prompt: "How to make this container responsive so that the content doesn't go out of it in smaller windows?"

Output:  1. Wrap everything in a responsive container

2. Use Bootstrap's responsive grid correctly

3. Avoid fixed widths or heights in your CSS

Prompt: This is my code, I'm avoiding fixed widths as you said, why is the div not working yet? .rectangle{ background: linear-gradient(white,#b9eff9); border-radius: 10px; padding: 10px; width: 70%; box-shadow: 0px 4px 10px rgba(0,0,0,0.1); margin-top: 10px; margin-bottom: 20px; display: flex; } .contact, .forms{ align-items: center; padding: 20px; width: 50%; }

Output: Perfect  That CSS explains exactly why your layout overflows on smaller screens — and fixing it is super simple. Let's look at what's happening and how to fix it without changing your visual design  So inside `.rectangle` (which is 70% of the page), you have two elements of 50% each side by side. That works fine on wide screens, but on narrow screens (like phones), 50% + 50% still try to sit horizontally → they overflow out of the container.

```
.rectangle {  
    width: 100%;           /* use full width of Bootstrap col */  
    max-width: 900px;      /* optional max width to limit it */  
    flex-wrap: wrap;       /*  allows wrapping on small screens */  
    justify-content: center; /* keeps content centered */  
}  
  
.contact, .forms {  
    min-width: 300px;      /* prevents shrinking too small */  
}  
  
/*  Stack the two blocks vertically on small screens */  
@media (max-width: 768px) {  
    .contact, .forms {  
        width: 100%;  
    }  
}
```

Edits: I added lines to the code so that the content doesn't flow out. Regarding the rectangle, its width changed to 100% so that it covers all the bootstrap containers, I added flex-wrap so that the content goes down in smaller windows, and justify the content center to keep the content centered and added @media.

Reflection: It is important to add

What I Learned

-@media is to set different conditions to the behaviour of that element depending on the screen size.

-flex wrap is for the content to move down when there is not enough space for them to be.

-The prompt that helped me to understand the CSS code of the rectangle that contains the forms helped me to adjust by myself the title in the hero, which was coming out of the blue rectangle.

Individual

Andrea

With each exercise I've done, I can see that everything actually works based on requests and responses. Even if you're doing something unfamiliar and seemingly out of this world, you always start from the basics, and it's not difficult to understand how it works. Also, I'm really impressed with all the things that you can do online. Before this class I didn't picture how everything online was done (meaning how the emails travel through the internet) and I think it is amazing that there are many possibilities that are easy to implement. I'm happy because I understand and implement the topic better and better. Although we couldn't implement the sending and receiving emails for now, I learned a lot and I think that that is the most important thing.

Team work reflection

I'm satisfied with my teammates' work. I think we maintain excellent communication and distribution of tasks. Everyone has done an excellent job with their parts and we have delivered everything on time and with high quality.

Aemi

Andrea and I developed the map of endangered marine animals in Mexico. At first, we explored different map tools and APIs, switching from Google Maps to OpenLayers, and finally to Leaflet for easier use with JavaScript. We used the IUCN Red List API, but since it had too much data, we filtered it to show only critically endangered species in Mexico.

Then I integrated the map into the Endangered Page, later merging it with the main project. Along the way, I fixed errors like a capitalization issue in an EJS file and improved the site's performance and organization.

Team work reflection

I think our team worked well together and maintained good communication throughout the project. We supported each other, shared ideas, and divided tasks.

Eduardo

Personal work reflection

I discovered that the data required for the project is available; however, there is no accessible API to retrieve it. Therefore, I believe we should prepare to generate a dedicated database containing this

information, allowing users to access it without having the data stored directly within the project folder.

Team work reflection

Collaborating as a team introduced diverse priorities and perspectives, enabling us to support one another in tackling challenging tasks and ensuring collective progress throughout the project.

Jetzuvely

Personal work reflection

During this project, I focused on improving my CSS skills and ensuring the layout and responsiveness of the website were functional across devices. I also explored different ways to integrate maps and GeoJSON data. Overall, I feel satisfied with my contributions and progress.

Team work reflection

Working with the team was a positive experience. Communication and collaboration were smooth, and everyone contributed their knowledge effectively. I learned from my peers, shared ideas, and we managed to implement the project goals efficiently. Everything went super well, and the teamwork was excellent.



UNIVERSIDAD
Panamericana

Final version: Use of AI report

Andrea Carolina Hinojosa Trujillo

Aemi Rosas Sánchez

José Eduardo Moreno Paredes

Jetzuvely Del Carmen González Gutiérrez

Universidad Panamericana

Web Application Development

Gabriel Castillo Cortes and Bernardo Moya de la Mora

December 2nd, 2025

Date: 18/10/25

José Eduardo Moreno Paredes

Project / Module: Catch the Change - Web Development

Goals for the documented session

- Complete map integration and refactor common components
- transform header and footer into reusable EJS components
- transform HTMLs into EJS templates
- add map functionality to fishing activity page
- add necessary routes in app.js
- add required dependencies if any
- update of project structure
- creation of commandsRequired.md file
- add devLog.md file for future documentation of development process
- add geoJson files for
- map functionality
- removal of unnecessary files (map.js)

Describe what you actually worked on.

In this development session, the project was refactored to improve maintainability and functionality.

The header and footer were transformed into reusable EJS components, and all static HTML files were converted into EJS templates.

Full map functionality was added to the Fishing Activity page using GeoJSON data, with corresponding routes updated in app.js

and unnecessary files like map.js removed. The project structure was reorganized, required dependencies were added,

and documentation files commandsRequired.md and devLog.md were created to support future development.

Document how you used Gen AI (e.g., Copilot, ChatGPT, etc.)

Prompt: "I want to connect my GEOJSONs into my ejs element called map. The geojsons are files that contain the information

for the commercial zone in mexico and two geojson about the protected areas"

Output: AI provided two implementation suggestion options:

1. Frontend approach using fetch() to load GeoJSON files directly
 2. Backend approach using fs.readFileSync() to pass data through EJS templating
- AI recommended Option 2 for better performance and provided complete code examples for both app.js route handler and map.ejs integration using Leaflet.js

Edits:

- Essential css for map height added to map container
- GeoJSON data loaded server-side in app.js using fs.readFileSync
- Added inline styles and CSS classes to legend color indicators for visual distinction
- Moved console.log statements earlier for better debugging

Reflection: The back-and-forth helped me understand:

- How EJS templating variables are passed from Express routes (res.render with data object)
- The <%- %> syntax for injecting unescaped content (necessary for JSON)
- Why Leaflet maps require explicit height styling to render
- The trade-off between server-side file reading vs client-side fetching

What I Learned:

- Server-side data loading pattern: read files → parse JSON → stringify → pass to template → parse in frontend
 - Leaflet.js GeoJSON layer integration with styling, popups, and controls
 - CSS requirement: map containers need explicit height or they render at 0px
 - EJS variable scope: variables must be passed through res.render() object to be accessible in templates
 - Debugging approach: using console.log to verify data loading before rendering

Date: 20/10/25

Jetzuvely González

Project / Module: Catch the Change - Web Development

Planned goals for the session:

- Make CSS changes to improve design and responsiveness.
- Research how to implement maps using GeoJSON.
- Test how to use the navigation toggle and media queries.

Work actually done:

In this session, I focused mainly on improving the CSS of the website, adjusting elements such as the mission section and the navigation bar to make them more responsive and visually appealing across different screen sizes. I also researched how to use GeoJSON maps to display commercial zones and protected areas in Mexico, although full integration will be

done in future sessions. Additionally, I tested features such as the navigation toggle and media queries to control the visibility and size of certain elements on mobile devices.

Learning and observations:

Adjusting CSS and media queries requires attention to the behavior of each element on different devices.

Understanding the logic of toggles and responsive menus helps improve user experience.

Researching GeoJSON maps gave me insight into available methods and tools, even though full implementation will come later.

I used AI mainly to clarify specific doubts and learn new techniques. I asked for help with:

Implementing the navigation toggle.

Using media queries to make images and sections responsive.

Understanding GeoJSON map integration methods.

AI provided clear suggestions, code examples, and guidance that helped me apply these features correctly. I also complemented this with tutorial videos to better understand the concepts. Overall, AI supported my learning without doing the work for me.

Date: 17/11/25 - 22/11/25

Aemi Rosas Sánchez

Project / Module: Catch the Change - Web Development

Goals for the documented session

- Make the Contact Us page fully functional so that users receive a confirmation email when submitting a message.
- Transform the entire project into React.
- Improve existing pages and fix UI/UX issues.

Describe what you actually worked on.

This week I spent most of my time transforming our entire project into React. At the beginning, it felt extremely complicated because I wasn't fully comfortable with the structure, components, and environment variables. However, with the help of documentation, examples from class, and a lot of trial and error, I was able to migrate everything successfully. I also fixed the recommendations page, made the form fully functional, and implemented an email confirmation system for users.

Document how you used Gen AI (e.g., Copilot, ChatGPT, etc.)

Prompt: "Why in the contactus.js the EMAILJS values from the process.env are not getting retrieved?"

Output:

Reason

- In a React frontend built with CRA (or many bundlers) only env vars prefixed with REACT_APP_ are injected into the client bundle. Plain process.env.EMAILJS_* will be undefined in the browser.
- Also .env must be at project root and you must restart the dev server after changing it.
- Note: putting email service secrets in client-side env exposes them to users — better to send email from your server.

Suggested fixes

- Use REACT_APP_ prefixes for the EmailJS keys.
- Use the correct backend endpoint (/api/contactUs).
- Add runtime checks and clearer error handling.

Edits: I applied all the suggestions: renamed the variables, reorganized the .env file, restarted the server, and updated the EmailJS configuration. It immediately worked.

Reflection: This helped me understand how environment variables work in React and why everything needs to be prefixed. I also learned about security concerns when exposing secrets on the frontend.

Prompt2: "Please help me to fix why I'm not getting the leaflet location markers under the endangered_species section"

Output:

Most likely causes (quick checklist)

- Leaflet CSS not imported → markers/controls won't display.

- Default marker image URLs not resolved by your bundler (CRA/webpack) → invisible markers.
- Map container has zero height or is covered by CSS → markers exist but are not visible.
- Wrong react-leaflet API usage (v3/v4 differences) or markers not added inside mounted MapContainer.

```
// ...existing code...
import L from 'leaflet';
import 'leaflet/dist/leaflet.css';

// Fix default icon URLs so markers show up in React builds
delete L.Icon.Default.prototype._getIconUrl;
L.Icon.Default.mergeOptions({
  iconRetinaUrl: require('leaflet/dist/images/marker-icon-2x.png'),
  imageUrl: require('leaflet/dist/images/marker-icon.png'),
  shadowUrl: require('leaflet/dist/images/marker-shadow.png')
});

// ...existing code...
```

Edits: I added all the required imports, fixed the default icon paths, and adjusted the CSS. I also realized that in React I needed to use the **L.** prefix for every Leaflet function to avoid compilation errors.

Reflection: This helped me clearly understand how Leaflet integrates with React and how import paths break easily.

Resources:

- <https://www.emailjs.com/>
- Additional React & Leaflet documentation

What I Learned:

I learned a lot about how React handles environment variables, component rendering, and external libraries like Leaflet. I also discovered new “tricks” that React requires to make everything work smoothly, especially with dependencies that weren’t originally built for it. Implementing EmailJS taught me how to send emails from the client quickly.

Date: 25/11/25 - 01/11/25

Aemi Rosas Sánchez

Project / Module: Catch the Change - Web Development

Goals for the documented session

- Fix multiple code errors appearing during local compilation.
- Adjust parts of the code so the project would run correctly on Render.
- Make backend modifications required for deployment.
- Continue improving project structure and stability.

Describe what you actually worked on.

During this week, I focused on correcting several errors that appeared when running the project locally. The biggest challenge was preparing everything for deployment on Render, especially the backend, which required multiple configuration changes. I researched build commands, environment variables, and deployment settings to make sure the project would compile and run correctly in production.

Resources:

- How to tell REACTJS to build a project in Production mode? -
<https://www.geeksforgeeks.org/reactjs/how-to-tell-reactjs-to-build-project-in-production-mode/>
- Deploy a Create React App Static Site -
<https://render.com/docs/deploy-create-react-app>
- I keep getting deployment timeouts -
<https://community.render.com/t/i-keep-getting-deployment-timeouts/14817>
- Environment Variables and Secrets -
<https://render.com/docs/configure-environment-variables>

What I Learned:

I learned a lot about how to properly deploy a project into production and all the steps involved, from configuring environment variables to understanding how Render builds React apps. I also gained experience debugging deployment errors and adapting the backend to work correctly on a live server.

Date: 26/11/25

Eduardo Moreno

Project / Module: Catch the Change - Web Development

Goal: Implement persistent authentication and a posts platform with MongoDB Atlas.

Authentication:

- Problem: Sessions didn't persist on reload.

- Solution: AuthContext + localStorage, ProtectedRoute for /posts & /profile, Google OAuth integrated.

Posts Platform:

- Backend: Node.js + Express + MongoDB, full CRUD, only authors can edit/delete, Post model includes author info and timestamps.
- Frontend: React fetches posts from API, auto-fills user info, loading indicators, inline edit/delete.

Contact System:

- Validates email, trims fields, logs operations, GET/DELETE endpoints, frontend handles errors.

Results:

- Persistent auth, CRUD posts functional, secure, improved UX, modular code, detailed logs.

Date: 21/11/25

Jetzuvely Del Carmen González

Project / Module: Catch the Change - Web Development

Goal: Create a section of the publication with navigation bar functionality and design

Functionality:

- unbreakable design

Posts Platform: I couldn't finish giving the page full functionality, so we opted to start from scratch and create a new interface.

Results:

- I completed the design and some of the functionality, but I couldn't complete the database and MongoDB.

General use of AI during the project

CSS Formatting and Indentation: Assisted in properly structuring and formatting CSS code for readability and maintainability.

Debugging Assistance: Streamlined the debugging process by highlighting potential issues and offering solutions.

Component Reusability: Guided the creation of reusable elements, such as headers, footers, and other EJS components.

Project Initialization: Suggested and documented the required commands for setting up and running the project efficiently.