React project development

Final Report of Capstone Project

João Ricardo Ramos Alves

Bachelor of Informatics and Computing Engineering

Tutor in U.Porto: Rui Maranhão
Advisor in the company: Nuno Ribeiro

27/03/2023 - 31/05/2023

Contents

1 Introduction

1.1 Contextualization

Bliss Applications was founded in 2009 and it belongs to the WYgroup. Its purpose is to develop digital products, such as mobile apps and websites, with a creative design that can improve their client's overall digital experience.

The company's headquarters are located in Lisbon, but they also have offices in Porto, where I spent most of my time, and in Boston. Their facilities in Porto are near the faculty of engineering, in UPTEC.

Currently, the company is working on several projects. Some of their most notorious clients are Farfetch, Sacoor, Global Shares, Worten, among others.

In these last few years, the company has experienced a huge growth, especially during the 2019's covid pandemic. At the moment, just the frontend department has around 50 people working on diverse projects, and the whole company has approximately 250 employees.

Bliss Applications was introduced to me through a job fair organized in FEUP last year. The company was presented to me as place filled with creativity and curiosity, as well as a great place to work. As such, when the time came to organize my internship for the Capstone Project, I contacted Bliss. After some exchange of emails and a few interviews, I was accepted into the Bliss Academy.

The Bliss Academy had multiple participants, each assigned to a specific group. The areas of each group were frontend, backend, and QA. I was assigned to the frontend group, due to personal preference. The purpose of this academy was to provide us with a large set of skills as well as to understand how the company worked, through the interaction with its employees.

Throughout this report, I will talk about my experience at Bliss, the challenges I was faced with, and the solutions I found, as well as the skills I acquired during this internship. This report will also help me reflect on the work that I have done and to consolidate any prior knowledge that I might have overlooked.

1.2 Objectives and expected results

The first purpose of this internship was to learn new technical skills that are currently being used by companies, in order to expand my skill set and my value as a potential employee.

Additionally, I wanted to gain insight on how companies operate. I was interested on how companies in my field of study would work, and how the communication between teammates and development teams would be done from an inside perspective.

The internship was divided into 2 main stages. The focus of the first stage was to create a solid knowledge foundation for each of the frontend academy members, and to allow us to discover a little bit more about the company. In the second stage, we started working with React.

The final goal of the academy was to build a successful React website with all the knowledge that we would acquire throughout our time in Bliss. This website should work like expected and take advantages of the tools that React provides.

1.3 Report Structure

In the following section, we will focus more on the overall experience of my internship. I will talk about how the academy was organized, and what activities we had.

In section 3, we will see the tasks and challenges that I was assigned to. We will also take an in-depth look at our final project.

Finally, we will reflect on the positive and negative aspects of this internship.

2 Employed methodology and primary activities

2.1 Employed methodology

Over the course of the academy, the work methodology remained fairly consistent. Most of the days we had a lesson in the morning, where a new topic was introduced to us. The exploration of the new subject was done by company employees, who helped organize the academy. Occasionally, they also shared some of their work experiences, where they demonstrated certain concepts that were covered in the lesson topics. I believe this was a very positive aspect, as it made the lessons more interesting and dynamic.

After the lessons concluded, we had some time to think about the topics. Any doubts were shared among the group and discussed.

In the afternoon, we applied the new concepts learned to our project. By putting into practice what we learned, we could better consolidate the new knowledge we acquired. Moreover, we could explore the subject in order to gain further understanding of it.

I believe the combination of theoretical lessons in the morning and practical application in the afternoon as well as constant available support, created a well-rounded learning experience during the internship.

2.2 Primary Activities

2.2.1 Introduction

In the first day of the internship, I travelled to Lisbon, where the headquarters of the company are located, to meet with the rest of the academy members. After a brief presentation of the company and the people responsible for organizing the academy, the participants shared a little bit about themselves. Following that, we had a team-building moment as a way to get acquainted with each other. Next, we had a detailed presentation of the frontend department and the main projects they were currently involved in.

Just a few days after the trip to Lisbon, I visited the office in Porto, where I worked for the rest of the internship. Here, I got to meet more people from Bliss that helped me throughout the time I was in the academy.

2.2.2 Lessons

Most of the lessons we had in the academy were during the morning, but due to incompatibility with some of the classes in FEUP, I couldn't be present in some of them. However, the lessons were recorded, and made available for all participants.

In Figure 17, it's possible to visualize the main topics of the lessons at any given time during the internship. It's important to note, that during the times that there is no major topic selected, work on the projects was still being carried out. Not only that, but some periods match weekends or holidays where there was a change of topics.

Our first phase lessons included subjects like HTML, CSS, JavaScript, and HTTP that I was already familiar with. However, other new topics such as Search Engine Optimization, Tailwind and SASS were introduced into our lessons. We also took a further in-depth look at HTML accessibility.

We also talked about Figma[2], which is a powerful web-based design tool that allows users to create multiple designs for applications, websites, and other products.

Our next main focus was on JavaScript. During this time, we addressed many topics of this programming language. We began with a basic overview of the language and its history, and then we moved on to more advanced

topics such as data structures, functions, events, and asynchronous execution.

Before the second phase of our project started, we had a lesson about agile methodology. The primary goal of this session, wasn't to have a complete overview of all agile methodologies, but instead, explore the most used solutions inside the company. The ones we discussed were Kanban, for smaller projects, and Scrum, which is used with some variations by Bliss.

We also had a session dedicated to exploring some workflows used with git. The ones talked about were Gitflow, Trunk-based, GitLab and GitHub workflows. Both advantages and disadvantages of each were mentioned.

Then it started the second phase of the internship, and its main phase. Our objective now, was to build a website using React. We started off with the origins of the JavaScript library, and then we moved on to other topics.

Firstly, we explored the structure of React, and how it breaks down our UI into smaller and reusable components. The first lesson also included JSX, which stands for JavaScript XML. It's a syntax extension for JavaScript that allows you to write HTML-like code within your JavaScript code. Following this, we learned new concepts of React, such as props, states, life cycles, and hooks. In order to integrate CSS into React, we also learned how to use styled components[4].

The last step was to explore the use of Redux[10], which is a predictable state container for JavaScript apps [10].

Overall, I consider the lessons were of extreme importance to the development of the projects.

2.2.3 Meetings

Periodically, we had a meeting to check on our progress. During this meeting we could clear any doubts that we might had related to our assignment. We also received useful tips that could improve our project on multiple aspects, such as good practices, better code readability and optimizations. Even though we only had this meeting occasionally, we were able to communicate any time using Slack[12].

For the purpose of having a more realistic experience, mutual cooperation between the members of the academy was also encouraged. This led to some meetings with just academy some participants where we discussed any doubts we had and shared ideas for improvement.

In addition, monthly meetings were made together with the supervisor from Bliss Applications, Nuno Ribeiro, and the advisor from FEUP, Rui Maranhão, to check on the progress of the internship and project.

3 Project Development

Throughout the internship we developed 2 similar main projects utilizing different technologies. One was developed using standard tools, such as HTML, CSS, and JavaScript, and the other was developed using the React library.

Besides these 2 websites, small research for learning purposes was conducted previous to the development. The objective of this task was to use a Chrome extension called Lighthouse[5] to analyse the metrics of well known websites, such as McDonald's[8], Zara[9] and Adidas[6]. The metrics taken into to consideration were the performance, accessibility and SEO. We were also tasked to do an exercise related to HTML accessibility. This way it was possible for us to understand what improvements could be done to those websites, and what we should take into consideration while building our own.

Following this small research and exercise, we got started with the projects.

3.1 Requirements

The goal was to build a website that allowed users to organize their financial information and track both their expenses and incomes. The name given to the website was Bliss Economics, which is obviously both a reference to the company's name and the purpose of the online platform. Users should have multiple features at their disposal, including:

- Adding a new transaction
- Deleting a transaction
- Editing a transaction
- Checking the overall income, expense and balance
- Checking the latest transactions
- Visualizing the transactions in a chart
- Viewing all transactions in an organized way
- Provide a way to hide monetary values for privacy concerns

The UI of the website was delivered to us through Figma. The design had both desktop and mobile versions that should be handled with a responsive

design. Through the Figma UI we can also visualize the interactions that the user can have with the website, as it shows different kinds of modals, as well as popups.

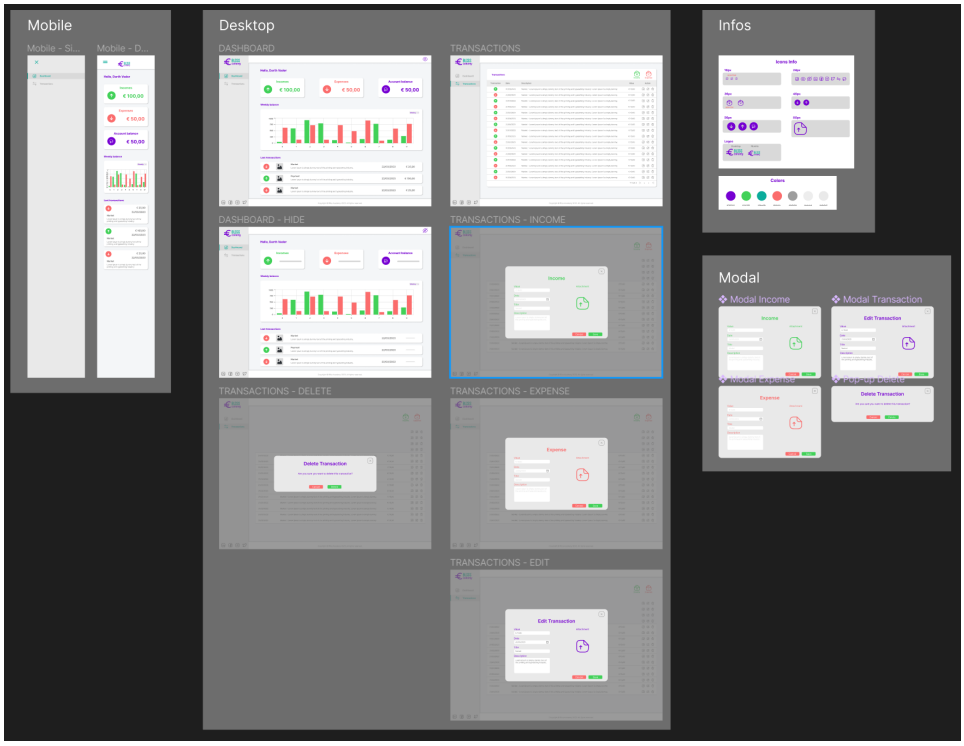

Figure 1: Figma UI of Bliss Economics website

Another requirement is to take accessibility into consideration while building our website.

Finally, there should be an integration with the work developed by the backend team from bliss academy. Even after integration, we should ensure that all components of the project work together as intended.

3.2 Architecture and used technologies

As stated before, the first project was built using HTML, CSS, and JavaScript. However, besides some standard CSS used for colors and responsive design, tailwind was applied for most of the style related work. Another option was to utilize SASS, but I believe the addition of another technology into a single project would make it difficult to maintain, and to keep it organized.

During the development of the first website, we didn't have access to the API made by the backend team, so the solution was to use JSON Server[7], a npm package, to run a local server to which we could simulate our API calls. By using JSON Server we were able to define custom routes, making it easier to continue the development without depending on the backend team's progress.

In the second project we used the React library to build our website. React allowed us to have a component-based architecture, which made our code much more organized and maintainable than in the first project. To style each of our components we decided to use styled components. One of the main advantages of using styled components is that we cam encapsulate the style of each component locally, preventing conflicts, especially as our project scales in dimension.

In addition, we also utilized React Router to handle navigation. This is necessary since React is a single-page application library that relies on client-side routing to ensure smooth navigation and rendering of components without being necessary to make full page reloads.

Furthermore, we use Redux to manage the internal application state related to the API calls. By utilizing this tool, the display of loading or error status to the user was also done, improving overall UX.

The interaction with API itself was done with Axios, a popular HTTP client library.

In order to allow the cooperation between members of the frontend team, all projects were organized in a GitHub[11] repository managed by the company, where all academy members had access to the projects of each other. This allowed for a much better cooperation between participants. Even though each of us were developing individual projects, we were able to provide mutual help and feedback.

3.3 Developed Solution

The Bliss Economics website I developed has 2 pages. The first one is the dashboard page where the user is able to check the overall status of their account.

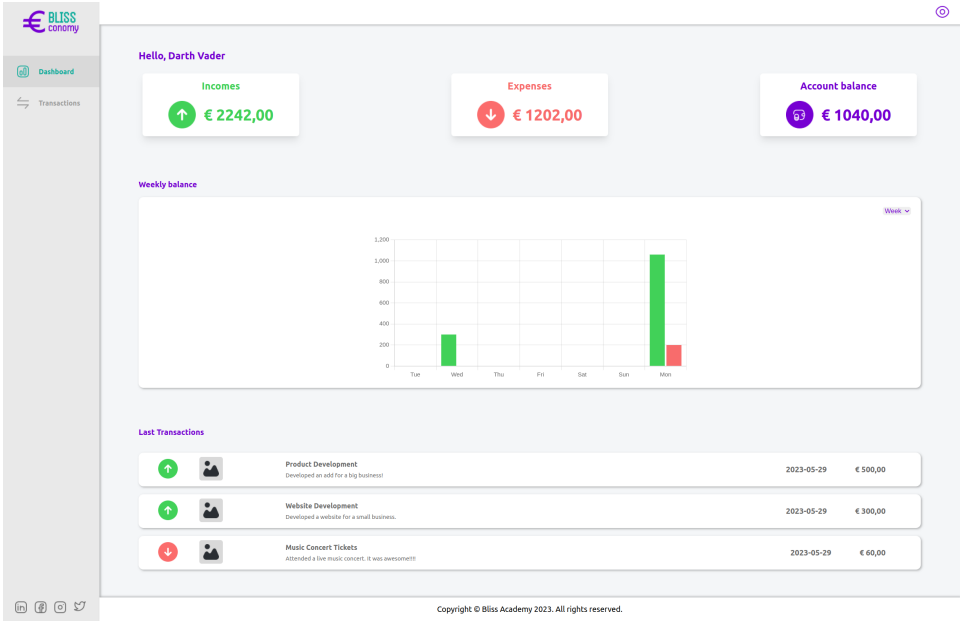

Figure 2: Dashboard Page

On the top of the page there are 3 cards. Each of them indicates the value of the user income, expense and balance, respectively.

Figure 3: Income, Expenses and Balance cards respectively

Below the cards, there is chart showing the transactions done during the selected time period. It's possible to visualize the transactions done throughout the last week, the current month or year. This chart is being generated using a JavaScript library called Chart.js[1]. It was used in both the Vanilla JavaScript and the React project with some required adjustments.

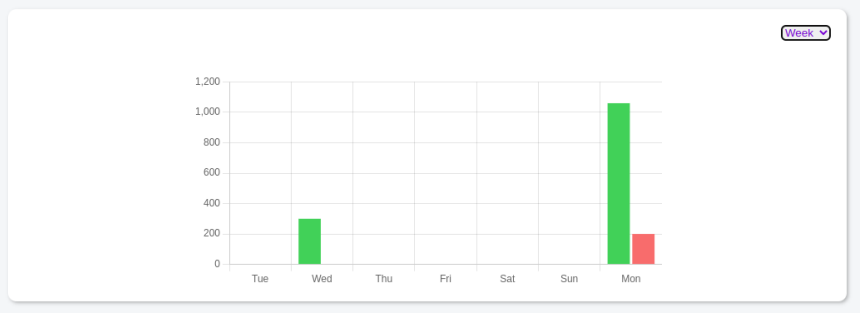

Figure 4: Chart with the information of the transactions from the past week

Right under the chart, in the ending section, it's possible for the user to view the last transactions performed by him. They contain relevant information, such as the title, description, date and value of each transaction.

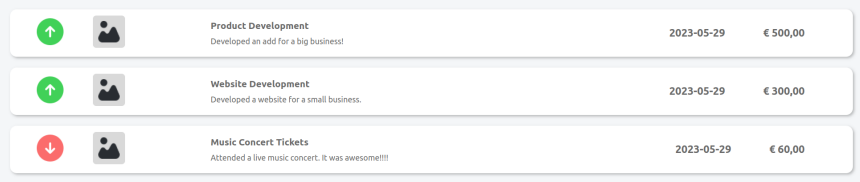

Figure 5: Last transactions performed by the user

It's worth noting that each item adapts quite well to a mobile display.

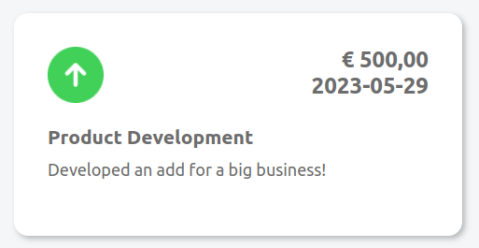

Figure 6: Last transactions performed by the user, while on a mobile display

These transaction items, as well as the cards have a functionality to hide their value when the eye button on the top right of the page is clicked.

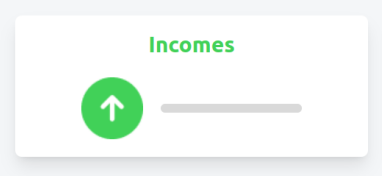

Figure 7: Card with privacy on

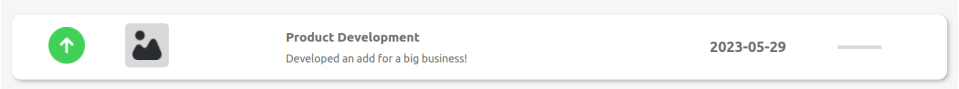

Figure 8: Last transaction with privacy on

Now, moving on to the transactions page.

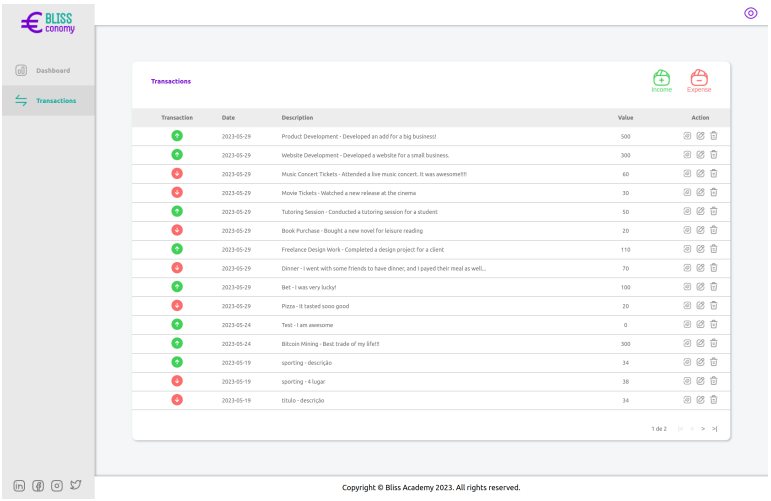

Figure 9: Transactions Page

As we can see in the image above, this page contains a table listing all transactions made by the user. In this table there are a lot of functionalities. In the top part of the table there are 2 buttons that allow the user to add new transactions, which can be either an income or expense.

Figure 10: Buttons to create new transactions

After clicking the buttons, a modal appears. The user can then input all the information for a new transaction to be created.

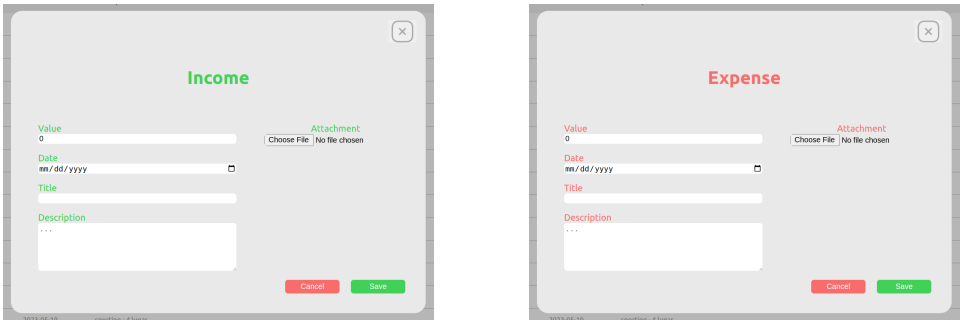

Figure 11: Modal to create new transactions. Income on the left and expense on the right

Besides creating new transactions, it's also possible for the user to edit old ones, since it's likely that a mistake can be made by a user. Each item in the table has an actions section where there is an edit icon. When clicked, it opens up an edit modal, similar to the previous ones.

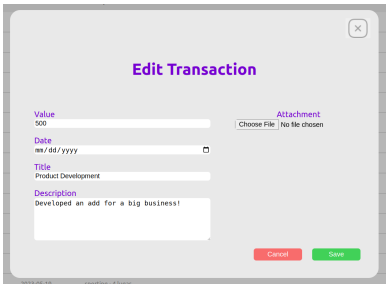

Figure 12: Modal to edit a transaction

Right next to the edit button in each item, there is also a trash icon that deletes the corresponding transaction. To complete this action, the user only needs to confirm it, when the popup appears.

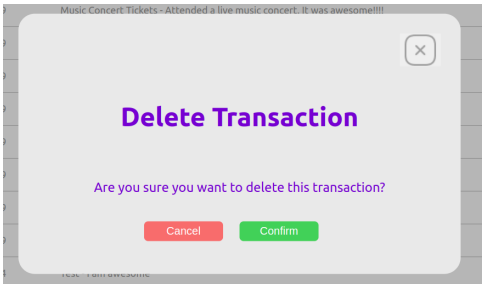

Figure 13: Notice before deleting a transaction

Since it's possible for the user to have a very large amount of transactions, the table doesn't display all of the transactions at once. Instead, pagination was implemented as a way to better organize the information. To navigate through the transactions, the user can use the buttons at the right bottom corner of the table.

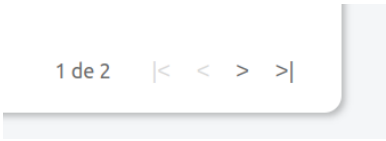

Figure 14: Table navigation buttons

For the user to change page he can use the side menu visible in all pages. While in mobile display, the menu collapses, and can be opened with the button on the page header.

Figure 15: Responsive design of side menu in mobile

Overall, the website provides multiple functionalities that the users can

use to properly manage their finances. The project code and challenges can be found in a GitHub repository[3].

3.4 Validation

To evaluate the implemented solution we decided to use the Lighthouse extension to access multiple metrics. The result can be seen in the image below.

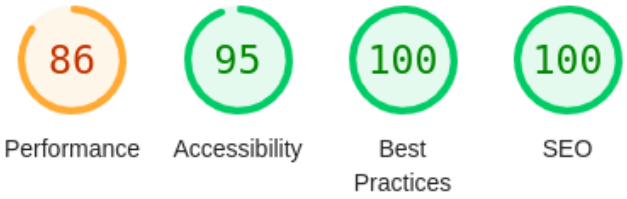

Figure 16: Results of Lighthouse extension on google chrome

In terms of SEO and Best practices, the website obtained a perfect score, which is a good sign. This indicates that the web page has a good and organized structure. Regarding accessibility, the score was very high, and the only negative points discovered were about the low contrast between some colors. Since we were required to follow the design given to us, it wasn't possible to improve on this aspect. Finally, the performance of the website was rated slightly lower than the previous metrics, however, it was still very good. This score could be improved by reducing unused JavaScript, and by minifying it. Other problems are related to the use of cache. All this could be improved when making the deployment of our solution. Nevertheless, I was happy with the overall score.

4 Conclusions

4.1 Accomplished results

The project was an overall success. I implemented all of the features that were in the requirements, even going beyond in certain things, such as the management of the internal state of the program when API calls are made, and displaying that info to the user. However, there are still currently some problems with the backend side of the program, namely, a CORS error. To overcome this problem, what I did was use the JSON Server mentioned

previously to replace the endpoints that weren't working properly. When the backend solves the problem in the future, it will only require a simple change to make the program work as intended.

In conclusion, I believe I was able to accomplish all the objectives I had set for this internship. I acquired new technical skills and insights on how companies of my field of study operate. Furthermore, we were also able to develop a React application with most of the core features that you would find in a regular website. I feel that this project was ideal to try out all these new technologies.

The company also gave me all the support I needed to perform this internship to the best of my abilities. Overall, the time I spent at Bliss Applications was extremely rewarding, even surpassing my initial expectations.

4.2 Lessons learned

I was able to acquire a lot of new useful technical skills by exploring new technologies. Among many of the tools we used, React was the one we spent more time exploring. Because we made two similar projects using vanilla JavaScript and React, it was possible to make a comparison between the two. In my personal opinion, I think React is a great technology that allows you to make a dynamic website in a relatively simple way. However, for me, the main advantage wasn't the ease of use, but instead the organization that it provides. Unlike with vanilla JavaScript, where there is no clear way to organize your JavaScript files, with React, components are organized and reused. Not only that, but they are integrated with HTML when using JSX, which makes changes specific to components much easier to do. This made the project much easier to develop, even as it dimension grew.

I consider that the communication among all frontend members, including participants of the academy and employees of the company, worked very well. However, I think that the communication with the backend team could have been better, and maybe required more effort from both sides to solve the remaining problems of our website.

Besides technical skills, I was able to understand how companies work, and improve my overall communication skills. Moreover, the knowledge acquired throughout this internship will definitely be of great importance in my future career.

4.3 Future Work

In the future, in order to improve our project, there are somethings that we can do. First, solve the CORS problem together with the backend team. Then, we could also review a part of the UI with the design team, to improve the accessibility of the overall website. Finally, once those are solved, we could move on to adding new features. It could also be beneficial to the project, to seek the feedback of real users and to perform more rigorous testing.

A Appendix

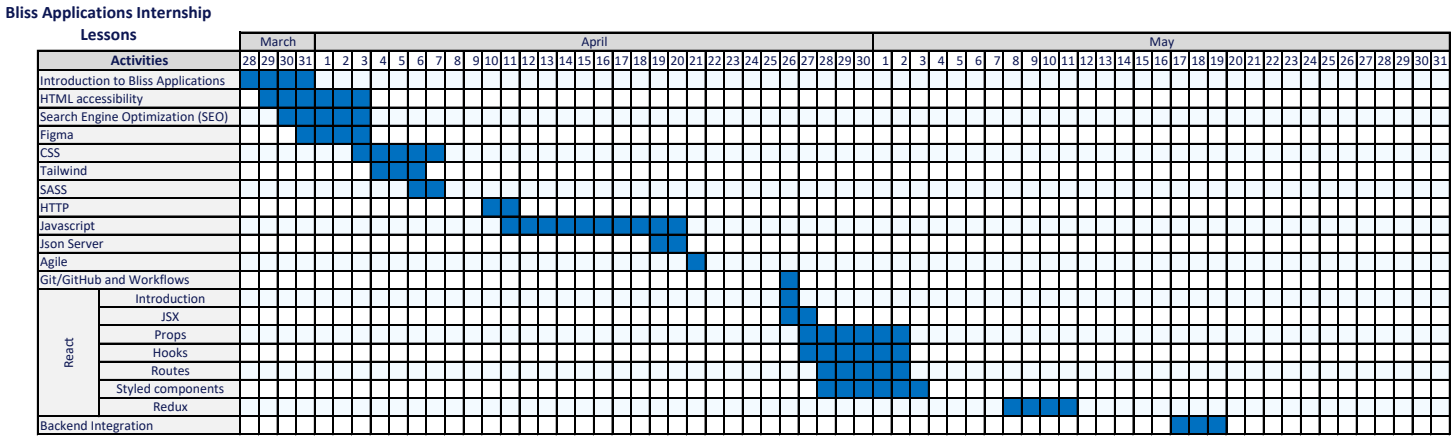

Figure 17: Gant Diagram of Bliss academy lesson topics

References

[1] Chart.js. https://www.chartjs.org/docs/latest/.

[2] Figma. https://www.figma.com/.

[3] Github project repository. https://github.com/Johnny-droid/Bliss-Internship.

[4] Styled components. https://styled-components.com/.

[5] Lighthouse. https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk, 2019.

[6] Adidas. https://www.adidas.pt/, 2023.

[7] Json server. https://www.npmjs.com/package/json-server, 2023.

[8] Mcdonald's. https://www.mcdonalds.pt/, 2023.

[9] Zara. https://www.zara.com/pt/, 2023.

[10] Dan Abramov and the Redux documentation authors. Redux. https://redux.js.org/introduction/getting-started, 2023.

[11] GitHub Inc. Github. https://github.com/, 2023.

[12] Slack. Slack. https://slack.com/, 2023.