✍🏽

Second Challenge

Welcome to our second React JS challenge 🥳

P.S.***: At the end of this document there are links that can help in the construction of this challenge.***

Introduction

Thinking about a new client that appeared in the market, **Compass UOL** had the idea of creating a planner. This planner will help the client to organize his week and his tasks and at what times they happen.

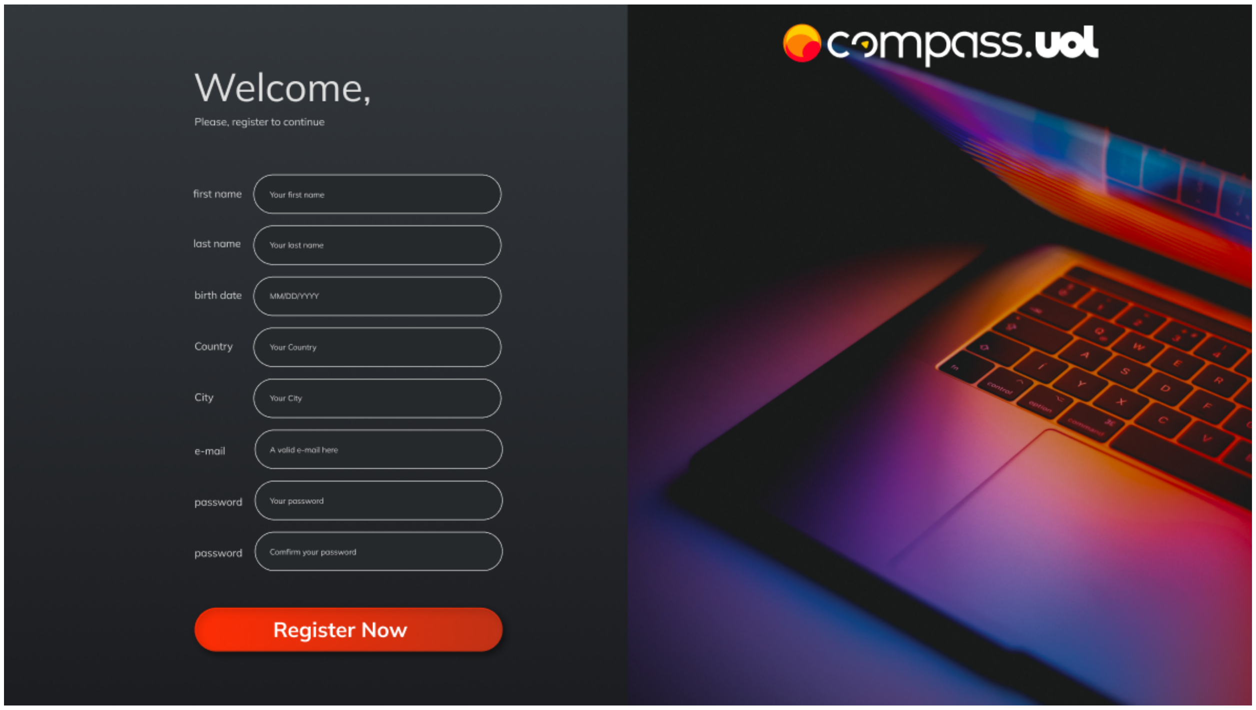
So you have this new mission, to build a new planner for some clients😀

1 . Register New Clients

In this step you create a registration form for new customers.

You have the creative freedom to validate as best you can and also make use of select instead a input where you feel it is necessary.

💡 **Remember to work with field validation!**



Mandatory:

- You need to use **localstorage** to store the users
- You need to follow the layout above

Tip:

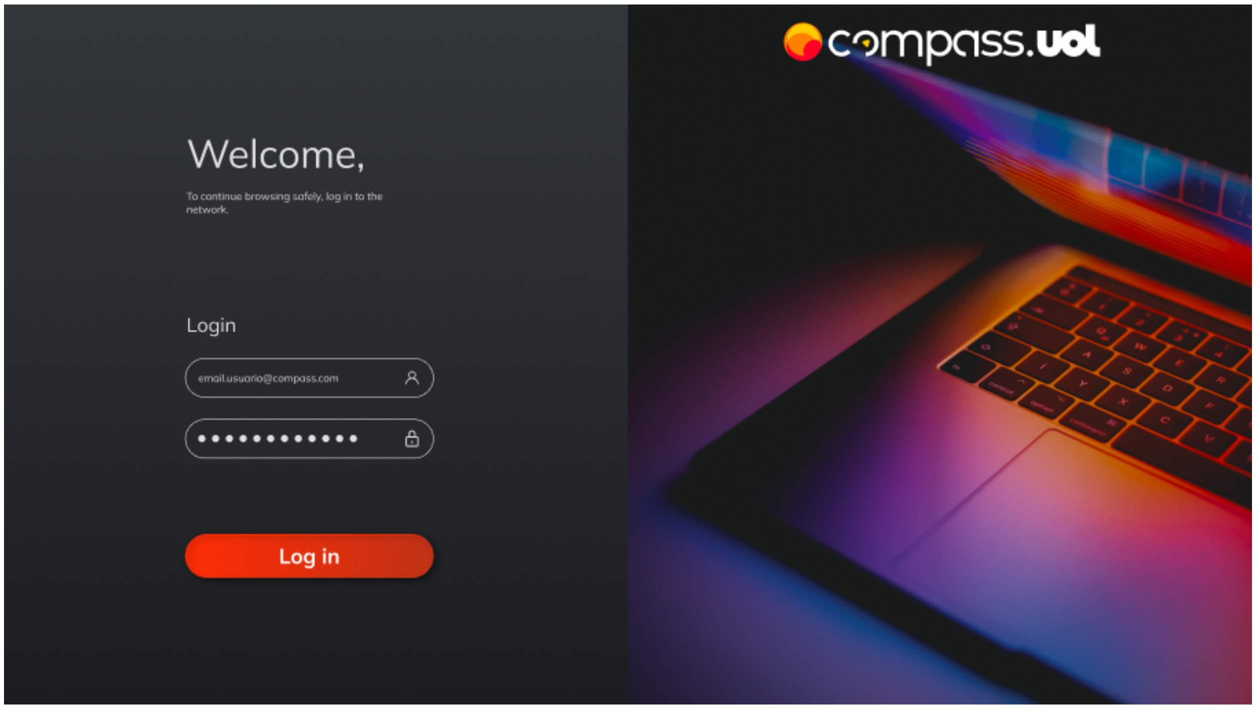
- You can use redux or context-api

2 . Login the new clients

In this step you after you register a new user, you need to login him to the system.

You can find more information at the **Useful links** at the end of this file!

💡 **Remember to work with validation!**


compass.uol
Welcome,
To continue browsing safely, log in to the network.
Login
email.usuario@compass.com
Log in

Tip:

- You can use redux or context-api for the register/login proccess

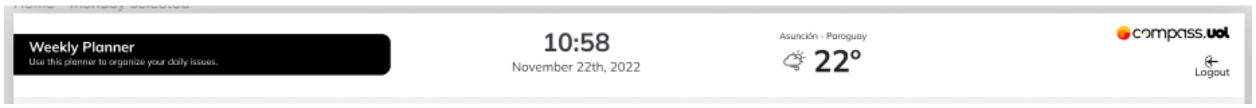
2 . Dashboard

In this step you will build a system dashboard. The user will be able to redirect to it when successfully logging in.

The dashboard is composed of two items:

- Header
- Main content

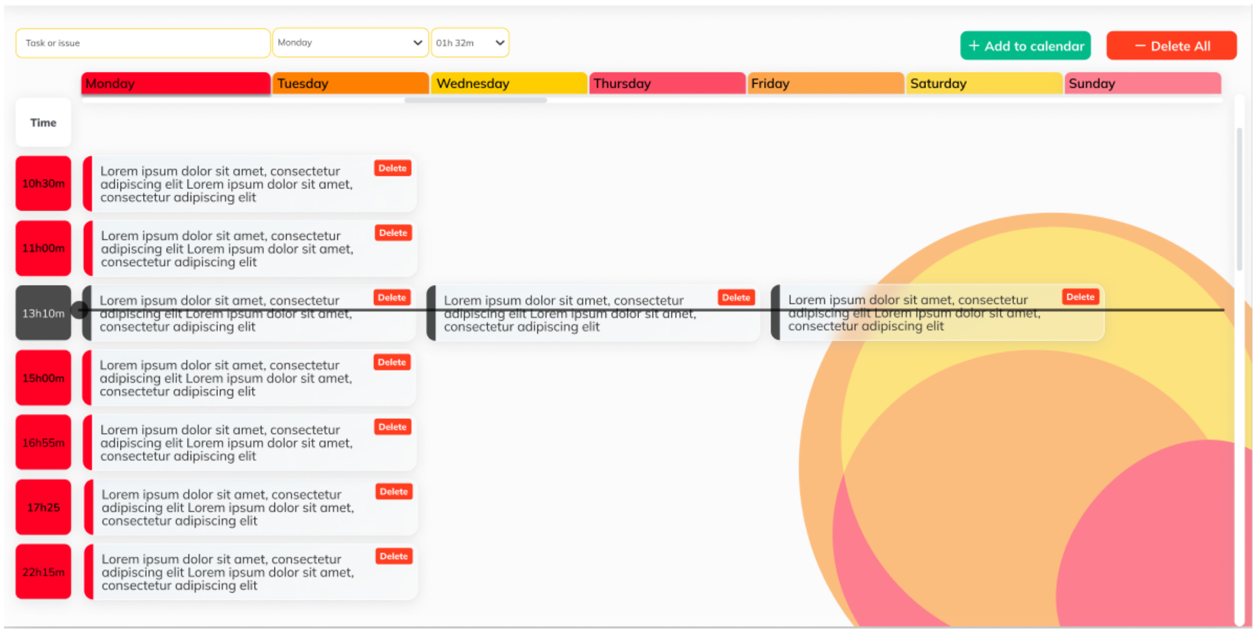
2.2 The Header



In the header you will have the name of the planner, current date and time, how is the weather and a logout button.

- **The logout button:** When the user clicks it, he should be logged out of the application and will no longer be able to access the dashboard.
- **Weather forecast:** The location made in the user's registration in the application must be used.
- **Date and time:** The date must be the current date and current time must to be updated.

2.3 The Dashboard



Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

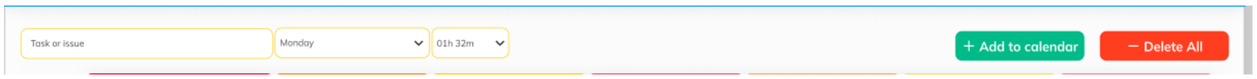
Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

The dashboard is composed by:

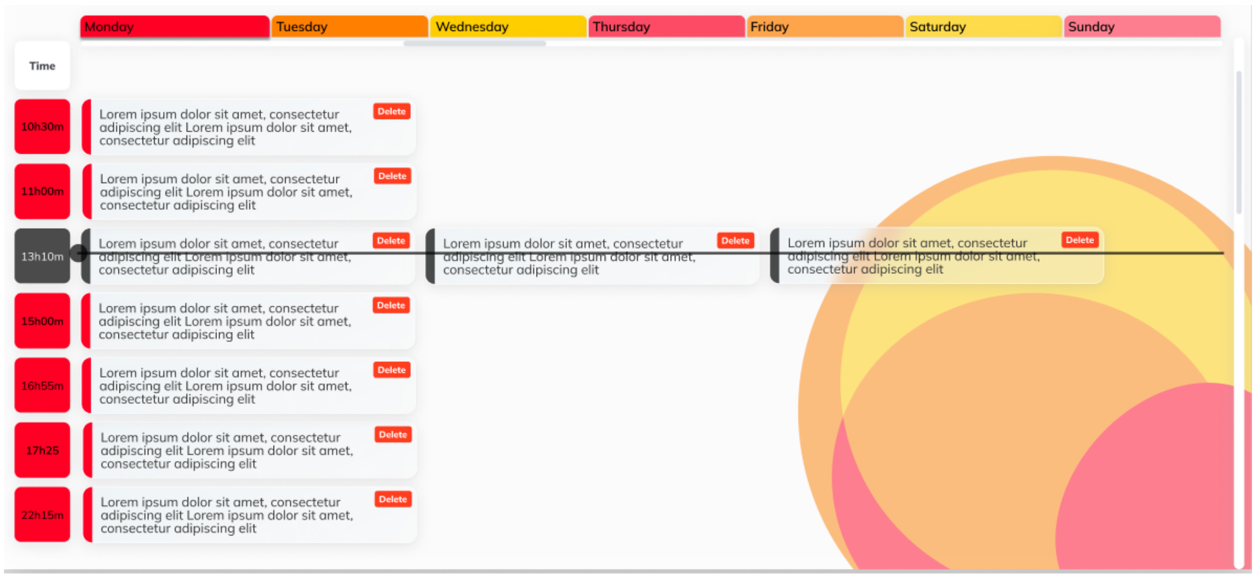
1. **Action Section**



In the action section you must have:

- One input where the use will put the card description
- A select containing the days of the week. (When selected will be the day where de new task you be added)
- A select with the hours, where the user can choose the event time.
- One add button that when clicked, will add the task to the board in the specified day of the week.
- One delete all button that when clicked will erase all data from the dashboard

2. **Board**



Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit Lorem ipsum dolor sit amet, consectetur adipiscing elit

The board will contain all created cards with the day of the week above and the time beside.

- In the board will only be events and the times previously added.
- The cards and the time must be the same color as the day of the week selected.
- The day of the week is clickable. When the user clicks, it will update the board to show the cards of that day.
- The selected card is slitly bigger then the others.

- Each card will have a delete button, that when clicked will delete the card from the board and the day.
- Tasks that already past the current time, will be colored gray and the row will have a line.

Mandatory:

- You need to respect the specified colors.
- Do not use external libraries like bootstrap, material-ui, tailwind and among others.
- Remember to make small commits within your develop.

Optional:

- If you wish, you can make the screen responsive, but at this moment is not mandatory.
- You can use Sass, CSS pure, Less, styled-componets and css-module.

Tip:

- Pay attention to separation of concern.
- You can use scroll in the dashboard.
- Think about reusability when you built the dashboard.
- Use all the knowledge acquired over the weeks.
- You can find the weather api in the useful links.

Useful links

- Figma https://www.figma.com/file/bFCO644LzxRZTqyGSLcgzI/Weekly-Planner?node-id=0%3A1
- Axios https://www.npmjs.com/package/axios

- LocalStorage https://developer.mozilla.org/pt-BR/docs/Web/API/Window/localStorage
- Weather API https://openweathermap.org/api