

Idea

The project - as all projects do - start with an idea. The idea of this SPA is dependant on the criteria though, so I will start from there. The requirements following the spec are as follows;

- Contain an index.html file
- Be an SPA with some form of interactivity
- Be able to read a flat JSON file
- Be able to export a flat JSON file

Given these requirements I have come up with the idea of a calendar. This isn't just because it fits the criteria but because I want to build one. I use my own calendar religiously and would love to be able to create my own version of it. Having the control to make it do what I want and look like what I want is something that makes me look forward to coding it and will push me to do it to the best of my ability.

Research

Researching into calendars there are an infinite amount of options to choose from. The 3 top used calendars are;

- cal.com
- calendar.google.com
- icloud.com/calendar

Looking into these they all are actually very different given they all aim to do the exact same thing. Cal aims to be a one stop solution, saying that it can connect all your calendars, have custom alerts and show when you're busy to other people. Google aims to be a simple usable calendar with information quickly available at a glance and my favourite; Apple. They seem to have a minimalist design while also

retaining import information that is most likely to be needed. They all have there strong points and niches which helps separate them. I aim to create a simple calendar inspired by all the above as well as my own ideas. Simple, clean, customisable, concise. What they all have in common is having a month calendar visible at all times, as well as the current date at the top. New event buttons are always visible and generally there is a lack of colors on the page.

Functionality

Initially there are functions the SPA must include. I will later go on to create a kanban board that gets updated every time a function changes state (e.g. goes from backlog to in progress) but for now I will put my initial requirements below;

- Display the current month with an option to change the month (e.g. go to 2021 October). This is self explanatory, a calendar must be able to see past and future months to be usable.
- Have a persistent banner for todays events. If there are no events on the current day do not show the banner. The calendar must be able to show upcoming events with a different level or urgency compared to other days events
- Be able to export and import your own calendar using JSON. Having a calendar that can go cross device is a very clear must and since I'm not willing to create a cloud infrastructure for this project this is the next best thing.
- Be persistent using local storage and proxies. The calendar would be fairly useless if every time you refresh the page your events disappeared.
- Options for each event. Each event must be able to have options that get reflected on the UI like colors and urgency.
- Live date/time. Having a library to update the time/date into whatever element I want just by in the element tag adding an attribute (e.g. `<p data-replace-with="hour:minute:second"></p>`)

SDLC

The approach I took to creating this project was the agile approach. Because I have a small amount of pre existing knowledge I know certain “challenges” will not be difficult or take time for me. So to combat this I will use the agile method; figuring out what I want to do, designing it, coding it, testing it, maintaining it. This works best for me in this project. Each part (from the kanban board) will be what you could call a sprint (details later about sprints). Going through each process for each feature of the project. These will all be documented in the meetings files on the GitHub repo.

Designs

I like to start my designs off on paper instead of going straight to Figma. I tried for over 30 minutes but images were not getting along with me so I have them all at the bottom of the page. They are all labelled. I went through 3 versions of the designs. First was on paper as stated and the others digital - using Figma.

User Story & UML Models

Below are the UML models for the calendar, highlighting key events and naming the procedures that should follow the events taking place. Also below is my user story graph, This helps me visualise what I should prioritise. Using this graph you can figure out what the user is most likely going to do so you can make sure those areas are better and focus on making them the star of the show

Sprints

There were quite a few sprints and if you would like to read more on each sprint you can do so by reading the meetings in the GitHub repo. Each one of the below is a sprint I did. But when I say sprint I do not mean 2 weeks. There wasn’t

anything that could've taken that long so I mean sprint as in each feature worth mentioning.

Feature	Status
Persistence using local storage/proxies	
Day (Tuesday, Monday) on each date in calendar view	
Ability to change selected month in calendar view	
Create, update, delete events	
Go to current month button if current month != selected month	
Dark mode/light mode based on color scheme	
Import calendar JSON	
Export calendar JSON	
Banner for todays events (hides if 0 events today)	
Events have custom colors and priority	
Live date/time with html attributes	
Define calendar data/calendar event type	

Here is a demo of one of my meetings/sprints

```
# Meeting 15/02/2024
```

This sprint was based off the remaining tasks to do. There are 2; add days of the week to each date on the calendar and use local storage to save the events locally.

Having each date on the calendar display the day of the week was a simple task. I used the 'Date' object (which is already formatted) to get the day of the week and then displayed it on the calendar with a bit of style.

The local storage was a bit more complicated. It has to be simple, readable and understandable. Another big thing is having the code componentised so that it can be reused and if your debugging something it is much easier to figure out where a problem could be and fix it. With the local storage it is there to make sure that the events are saved when the page was closed and when its opened again the events are still there.

As i am all done now here is a link to the final live [[Website](https://comp-1004-spa.vercel.app/)](<https://comp-1004-spa.vercel.app/>)

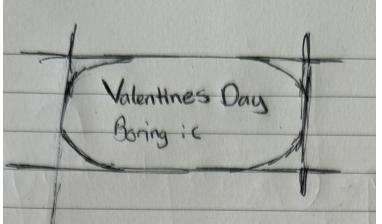
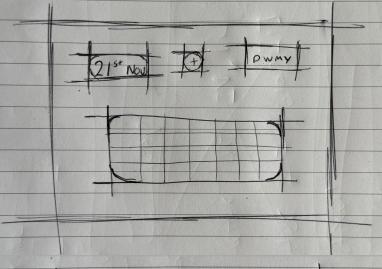
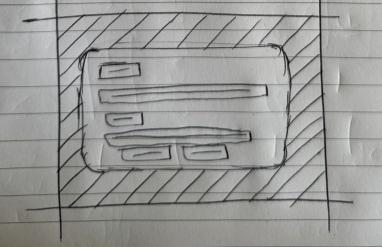
Issues & Challenges

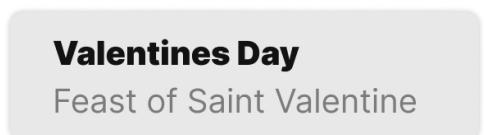
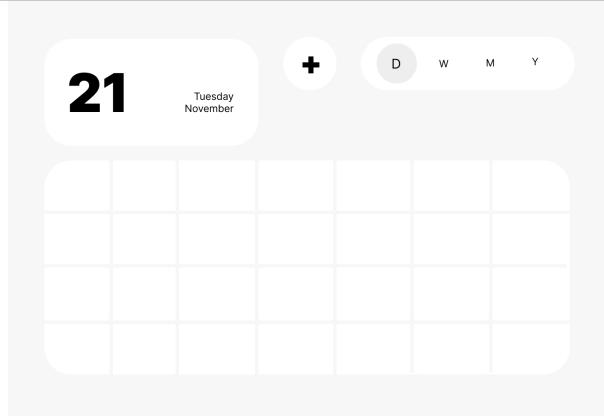
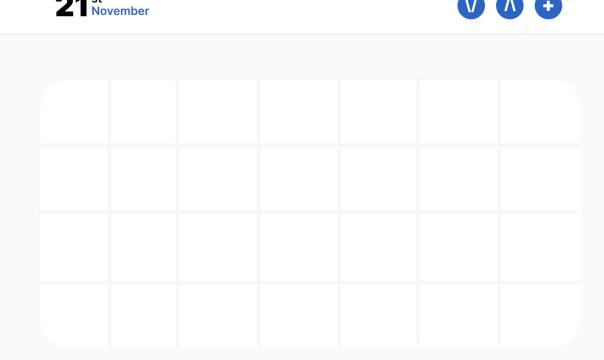
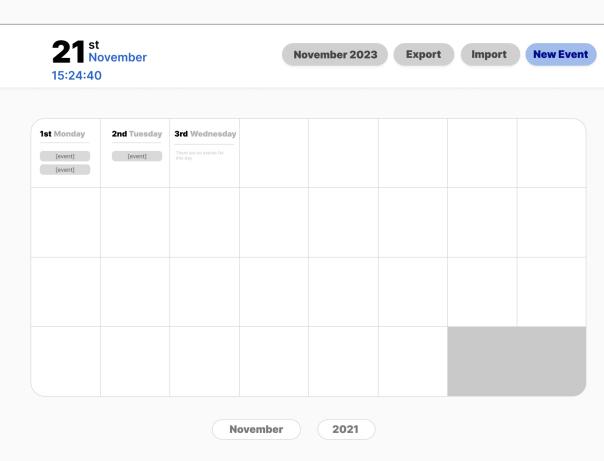
As stated above because of my small but yet pre existing knowledge or web dev I didn't seem to struggle with many things, I did in fact enjoy this project. The most difficult things I faced were;

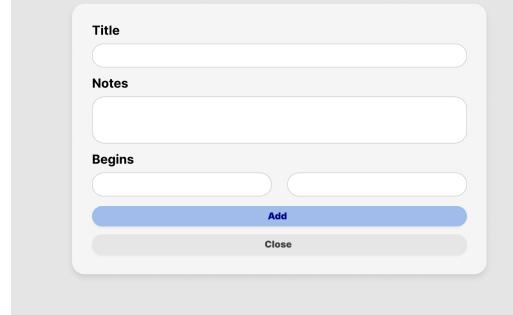
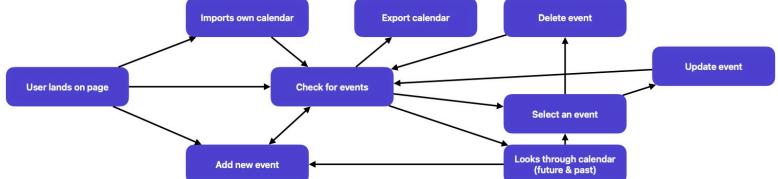
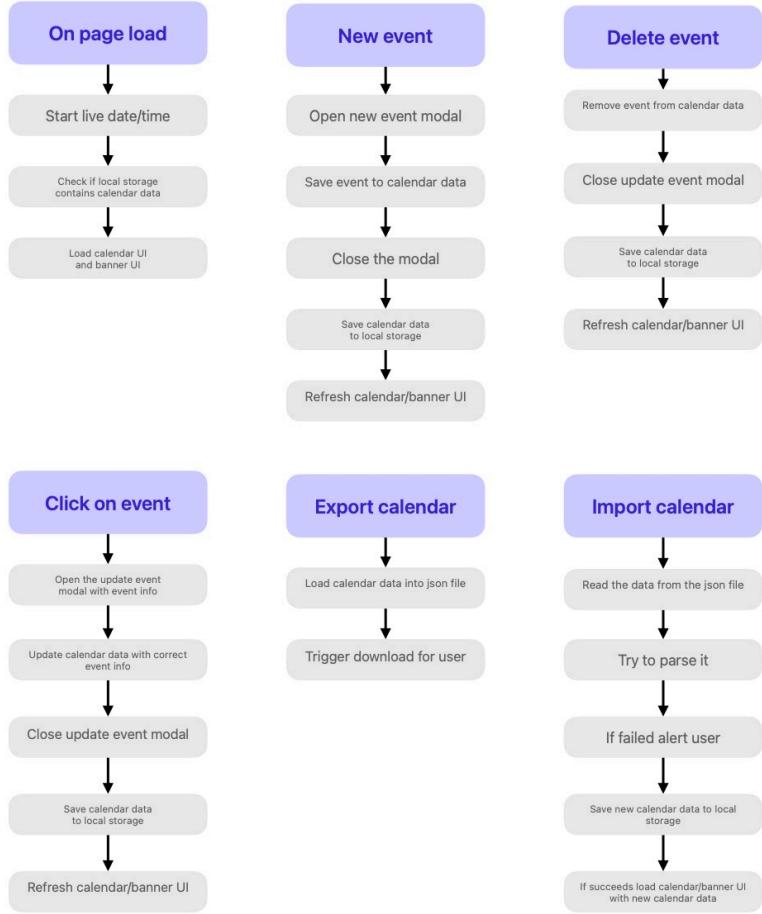
- The calendar data being persistent. This posed quite a challenge as you have to first check if it exists, then deal with getting it. Then you have to save to local storage every time the calendar changes, even the smallest change. Luckily something called proxies exist in JavaScript making this easier to implement.
- Live date/time was also fairly difficult as I had to have a constantly updated and formatted time/date and then on each element that requires it insert the correct information where it should be. Again JS luckily has a built in date formatter then can be used to neatly format not only today and now but any date.

Poster

My poster is below with the other images

Paper Design - V1 - Single Event	 A hand-drawn sketch of a single event card. It features a large oval at the top containing the text "Valentines Day" and "Boring :(".
Paper Design - V1 - Page	 A hand-drawn sketch of a page layout. At the top, there is a header section with three columns labeled "21st Nov", "D+", and "D-MY". Below the header is a grid consisting of two rows and three columns.
Paper Design - V1 - Modal	 A hand-drawn sketch of a modal window. It contains a grid with several rectangular cells, some of which appear to have icons or text inside them.

Digital Design - V1 - Single Event	
Digital Design - V1 - Page	
Digital Design - V2 - Page	
Digital Design - V3 - Page	
Digital Design - V3 - Single Event	

Digital Design - V3 - Modal	
User Story Graph	
UML Models	 <pre> graph TD subgraph OnPageLoad [On page load] Start[Start live date/time] --> CheckLocalStorage[Check if local storage contains calendar data] CheckLocalStorage --> LoadUI[Load calendar UI and banner UI] end subgraph NewEvent [New event] OpenModal[Open new event modal] --> SaveData[Save event to calendar data] SaveData --> CloseModal[Close the modal] CloseModal --> RefreshUI[Refresh calendar/banner UI] end subgraph DeleteEvent [Delete event] RemoveEvent[Remove event from calendar data] --> CloseUpdateModal[Close update event modal] CloseUpdateModal --> SaveLocalStorage[Save calendar data to local storage] SaveLocalStorage --> RefreshUI end subgraph ClickOnEvent [Click on event] OpenUpdateEventModal[Open the update event modal with event info] --> UpdateData[Update calendar data with correct event info] UpdateData --> CloseUpdateModal CloseUpdateModal --> SaveLocalStorage SaveLocalStorage --> RefreshUI end subgraph ExportCalendar [Export calendar] LoadJSON[Load calendar data into json file] --> TriggerDownload[Trigger download for user] end subgraph ImportCalendar [Import calendar] ReadData[Read the data from the json file] --> TryParse[Try to parse it] TryParse --> FailedAlert[If failed alert user] FailedAlert --> SaveLocalStorage SaveLocalStorage --> RefreshUI end </pre>

Poster

The Last Calendar

Welcome to my innovative calendar where organization meets simplicity. This is the last calendar you will ever need! Prioritizing user-friendly UI, designed to streamline your daily activities. Stay on top of your plans effortlessly and discover a new way to manage your time effectively. Unlock the power of efficient planning today!

Features

The calendar itself is designed with ease in mind. Making sure no matter what your doing it will come easy. Speed is another huge priority, you will not experience any lag. Not only is the calendar saved locally on the browser but you can import & export your own calendar too. Easily see the type of event via colors that you can set too whatever you desire.

How Its Made

The calendar is made entirely using vanilla HTML, CSS & JavaScript. In doing this there is no bloatware. Only what I use is loaded in the browser therefore it is small and fast. Use the browser tools to take a look at how its made, the code is clean, formatted and organised.

Bobby Mannino
comp-1004-spa.vercel.app bobby.mannino@students.plymouth.ac.uk