Alter Ego – Project Report

1. Description

The alter-ego web app is a gamified daily calendar. The game's premise is that the user's productivity reflects on the main character of the game- an astronaut.

The app begins with a calendar where users can enter their day's schedule. The tasks can be of two types - health and work. If the user fails to do a task, the astronaut's or the spaceship's health is affected negatively (depending on the type of task). And while the user is engaged in a task, they can see the astronaut character doing corresponding work in the app. At the end of the day, the astronaut's mission will be affected by how the user performed on that day. The game has four endings depending on how well the user managed their own day.

The motivation for this project was to create an effective calendar app that is engaging to use, and also a productivity tool. The idea is that the users will be motivated to complete their daily tasks, when it is gamified with a dramatic story and beautiful visuals. Since their own productivity is linked to a character in a game, it will be a compelling force to be productive in their own lives.

I personally struggle with time management and am always on the lookout for an effective scheduler app. This app is designed for anyone who needs motivation in their daily lives and is designed to make their chores more interesting.

2. How to use

VIDEO DEMO (4 minutes long) - https://youtu.be/FCCDgeGc5dI

1. Click on "Start your Day" button, and an empty calendar pops up.

2. Enter your schedule for the day. You can enter a task my clicking or dragging on the calendar. You can also drag, resize, update and delete a task after adding it.
(Note: enter atleast one task to see the "Confirm Schedule" Button, Any task entered earlier than current time is ignored)
(TIP: Enter a task which starts one minute after current time for testing)

3. Click the "Confirm Schedule" button and the game starts (if it is a valid schedule). You can now see the calendar and the time till next task.

4. Click on the accessibility button on the bottom left if you want a high contrast view.

5. When the time arrives for a task to start, you will see a popup on the screen. You can choose “Yes” if you are doing it, or decline it by pressing “No”
(Clicking “Yes” shows the astronaut working on corresponding task, and “No” will reduce the points in the game as shown”

5. During a task, you can choose to end it early by pressing the button on bottom left. During no task you can choose to end the day by pressing the same button.

6. When all tasks have passed, the game ends and you are taken to the results page with one of the four endings and your scores.

7. At any point you may click the reset button on bottom right to start over.

3. Tools used

- I used the library Toast UI Calendar.

- Since one of the main component of the app is the calendar, I chose to use a library to implement it, so that I can work on the other aspects of the app. Dealing with all the edge cases of a complete calendar is quite tedious (time zone, daylight savings, leap year etc), which is why this library was used.

- I imported it with npm and referenced it like a standard library, and used my own CSS to override the styling and javascript to customize the behavior (showing only daily view, update/delete events, events smaller than 30 minutes, etc.)

- The calendar is the primary component in this application since it is a daily scheduler. It provides the main framework of the gamified day planner.

4. Development

I was quite conservative in my original plan, since I was unsure if I will be able to find a suitable library for this app. However, over multiple iterations I added more features than I originally planned, like multiple dramatic endings, accessibility mode, supporting complex schedules with overlaps, and various stylistic changes. Further, I constantly iterated on the design over the past weeks, which made the UI more polished than my original mockups.

5. Challenges

The project was quite challenging since it involve various cases like - overlapping calendar events, switching between tasks, ending the day, empty calendar etc., which all involved nuanced changes in the game. The other main challenge was debugging- since I had to wait for several minutes for a task to start or end.