Write Up

Webapp link: https://pvegired.github.io/life-in-numbers/#/
Github repository link: https://github.com/pvegired/life-in-numbers/tree/master
Video folder link:
https://drive.google.com/drive/folders/1Cp3pGNVvfS83zJVanJ4WHwpemOoRS1Vi?usp=sharing

Project Description:

“Day Quantifier” is a personal analytics dashboard project. It quantifies your day by giving weighted scores to help you focus on the things that matter. There are 4 different variations of todos with explanations of what they are. The division is based on the initial study that I conducted with students on their perception of productivity. The division is intended to solve their problem concentrating on things that matter. Oftentimes students think they are productive but end up doing something unnecessary for their immediate goals. So in the day quantifier they can add the tasks that they have in different sections and calculate the score for their day based on the importance of each bucket which they give through weightages.This way they have a perfect reflection of how their day went with respect to their immediate goals and expectations from them.

In this final project, I primarily focused on the backend part of the application. One major challenge is to sync across sessions and todo so I used firebase as the backend tool. I fetch the and display the data through real time APIs. Every minor addition by the user gets added to the server in real time, so that whenever he/she signs out, he could sign in back to the dashboard with where he/she left off.

I added the time left for the day to help users give a gamified sense of mission. And whenever they finish a task the score in that particular division updates with delightful progressbar animation so that in the future when we work on corresponding insights on how each of the sections went, they could view them. It also gamifies each block. This day quantifier will help students quantify their day productivity.

Bulleted List to Describe how a user would use your website:

1. Login / Create an account / Sign In with google → to help transfer their data across sessions
2. Add multiple tasks in a Task List of their liking
3. Check them off when finished → Updates score and progress bar of that task list
4. View the time left in the day for more gamified mission like seriousness
5. View the score to see how their day is going
6. Sign out whenever they want
7. Sign in and find the dashboard at exactly where they left off
8. Loop of [steps 3 to 7]

View the video, for more clarity

External tool:

Name of Tools Used:
I used react for the front end and primarily used firebase for the backend.

Why these tools:
My primary motivation for choosing React is because there are a lot of repeating items in a todo list and for the scope of this project I wanted it to be a single page dashboard and not have multi-page functionality.

My primary motivation to choose firebase for backend is to have continuity across sessions and real time updation of multiple data elements in server which is a very crucial aspect in an application saying that it tracks and quantifies your day.

How did I use them:
I created.a realtime database in firebase and copied the api details and pasted it in react as a separate js file and started calling the db whenever required based on the keys, ids, names in the database.

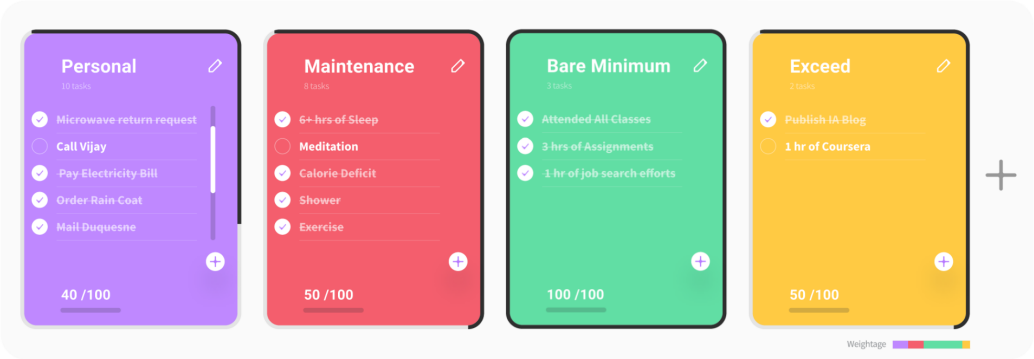
I also used external packages like font-awesome for icons and datepicker for picking the date and to display and calculate the time left in the day function.

What does these add:
Real time reflection of where the user left across different sessions. And a lot of ease of effort in writing the whole javascript functionality of todos and scores.

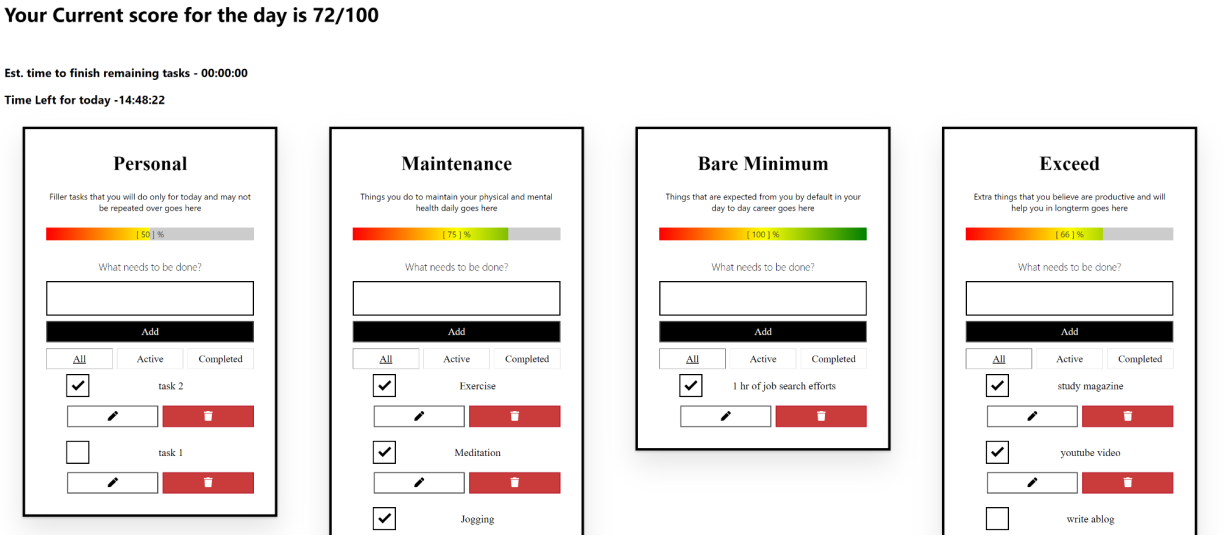
Iteration on HW7 Mockups: 2 pts
Primary change from HW7 mockup is the task list container.When I asked students who are target users around me, they liked the quantification aspect but they also wanted to view the active and completed tasks separately instead of scrolling through them. And they asked if they could delete them as well.

Fixed size too was changed to more of free flowing to avoid the in task container scrolling which makes some of the tasks go hidden.

Before:



After:



Challenges experienced in implementing the website:

I lost track of the amount of challenges I had with making the backend work. Primary challenge is the real time updation instead of session end updation based approach. Every item added by the user has to be stored in the database and then reflected from there to be displayed. That took a lot of time.

Faced major issues with hosting the application too as initially I had forgotten to have a .gitignore file and it started uploading all the node_modules and it started throwing multiple errors because of long names. Issue was gone when I added the .gitignore file.

Other issues that are coming into my mind are real-time score calculation (For which I wrote loops before but when it didn't work, created a new place in database to store them outside the loop and fetch them instead), making the sign in with google page work by adding O auths both before and after hosting on github, changing the structure of database for it to be more date centric so that it's easy to work on insights in future etc.,

Immediate next steps to Implement:

1. On change of date show the corresponding tasks of only on those days
2. Have some task containers which are recurring and unfinished todos to be carry forwarded to next day in others