

Developer Journal Starting Pitch

Team 25: Git Gremlins

Problem

- What problem are we solving?
 - productivity
 - carrying a physical journal
 - wellness? mental health?
- How can we solve the problems using this web?
 - easily accessible
 - keeps track of tasks/reminders
 - organizes thoughts and emotions
 - promotes self-reflection and growth because you can see previous entries

In today's digital age, maintaining a productive lifestyle and balancing mental wellness poses a significant challenge. As technology remains omnipresent in each of our lives, it becomes harder and harder to disconnect each day, making these goals seem almost impossible.

Moreover, traditional methods for managing these challenges, such as carrying a physical journal, are becoming outdated. I'm not sure anyone wants to carry one around all day with them, or at least has the convenience of doing so. While personal and tactile, physical journals do not seamlessly integrate with other digital tools that dominate our daily routines. If one chooses to carry one, they are easy to forget or misplace, and do not offer reminders or analytics that can enhance self-understanding and efficiency.

Additionally, the process of mentally tracking tasks, appointments, and personal reflections is time-consuming and often overwhelming. Someone might use a planner or reminders to help them, but this can detract from mental wellness, as individuals may neglect self-care and reflection in the face of mounting daily tasks.

These issues highlight a clear need for a more integrated, efficient, and accessible way to manage personal productivity and mental health in the context of our modern, digital-driven lifestyles.

Appetite

We will have about 6 weeks until the project finishes, so we will need to constrain ourselves to only implement certain core features. Some features that we've decided to omit are:

- Task deadlines
- Task tagging
- Task sorting

These are due to us trying to balance time limits with the main goal of our product. We decided on intending the task list to be a limited side functionality with more simplified controls, thus removing the more complex capabilities that would take too long to implement. We also intend for users to focus more on the journaling aspect of the app, rather than treating it as a planner or to-do list.

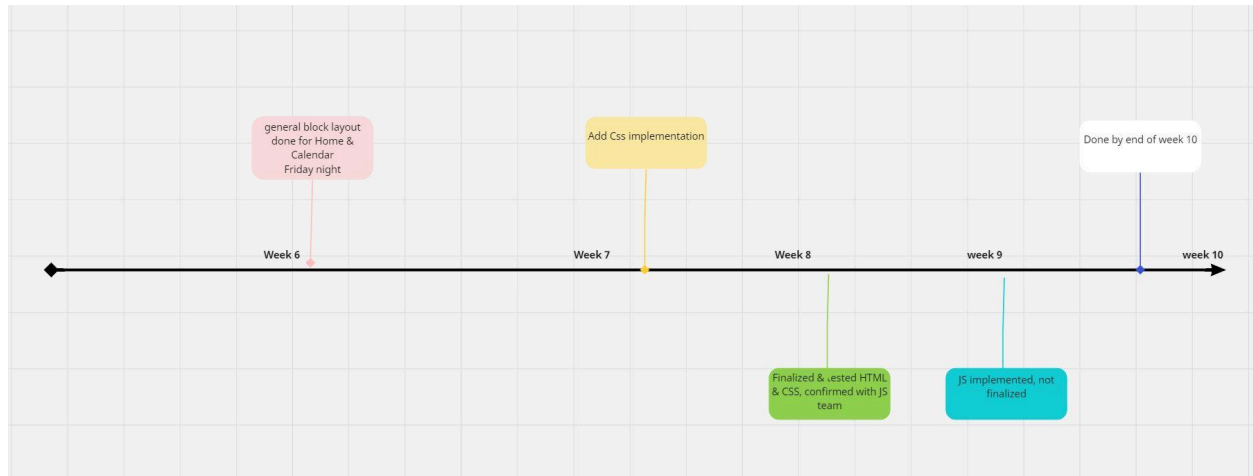
As for technologies we plan to use for this project, we've decided to keep it as simple as possible primarily only using the default html, css, and javascript. We believe that this is what everyone is most familiar with, so that we can proceed without everyone learning how to use a new piece of technology. Also with these limitations, we can keep our application very lightweight.

So our finalized main features that will be split into batches are the following.

Big Batch Main Features:

- Journal
 - Basic HTML and CSS
 - Backend linking with storage
 - Mental and Productivity Rating
- Calendar
 - Basic HTML and CSS
 - Past completed task display
- Task List
 - Basic HTML and CSS
 - Add and delete tasks
 - Link with calendar days

Timeline



2 Main Phases

Phase 1: Frontend structure and Backend research and planning

- 1.) Plan and build HTML & CSS rough layout for Home page and Calendar
- 2.) Build task list, past week side view, and add details to homepage and calendar
- 3.) Add aesthetic features with CSS and finalize frontend

Phase 2: Frontend finalization and Backend implementation

- 1.) Research technologies necessary and outlines the functionality that will be needed. Plan backend storage capabilities.
- 2.) Implement basic functionality for journaling, tasks, and calendar buttons and interactions.
- 3.) Implement storage of user data.

Solution

Features & Functionality:

Journaling - Users can make journal entries for each day; these entries are stored and associated with that day and can be viewed later.

Sentiment and Productivity Rating - Users can rate how they feel and how productive they were each day; these selections, much like the journal entries, are stored and associated with that day and can be viewed later.

Tasks - Users can assign themselves tasks; tasks do not have deadlines nor notes. Tasks are meant to be brief, these are more like small reminders or a scratch pad of things to get done.

Tasks can be completed, edited, and deleted. Tasks completed on a specific day are then stored and associated with that day, much like the journal entries.

Calendar - Users use the calendar primarily to view past days and see the entries of that day.

Calendar also gives a preview of the tasks completed and the sentiments of that day to offer users a quick summary of their past work life.

Technical Features & Decisions:

Database

We would prefer using local first by storing the data in the user's computer with LocalStorage and storing all the information we need as strings in a JSON file. We plan to organize it with one main file for all journals, with each entry being for each day. We will also store tasks in another JSON file.

Testing Framework

HTML & CSS

- Test varying screen sizes to ensure responsiveness of interface.
- Use LightHouse to test performance, speed, and accessibility of webpage.
- Manually test abuse of interface, i.e. long journal entries, misclicks, user errors

JavaScript

- Unit Testing via Jest continuously as development progresses.
- Testing for each JavaScript function and app features

Non-Functional Features

Responsiveness

We want to focus mainly on our users being able to access and use the functions properly and responsively.

Performance

Since our app's functionality does not require heavy data processing, we will use the user's local storage as the database. Because of this, performance shouldn't become a significant issue, so we plan to prioritize user experience.

Accessibility

We plan to use LightHouse to test performance and accessibility. To ensure the best accessibility for users, we will ensure good HTML practice with standardized HTML tags so that navigation on the website is possible via screen-readers, only keyboard, etc.

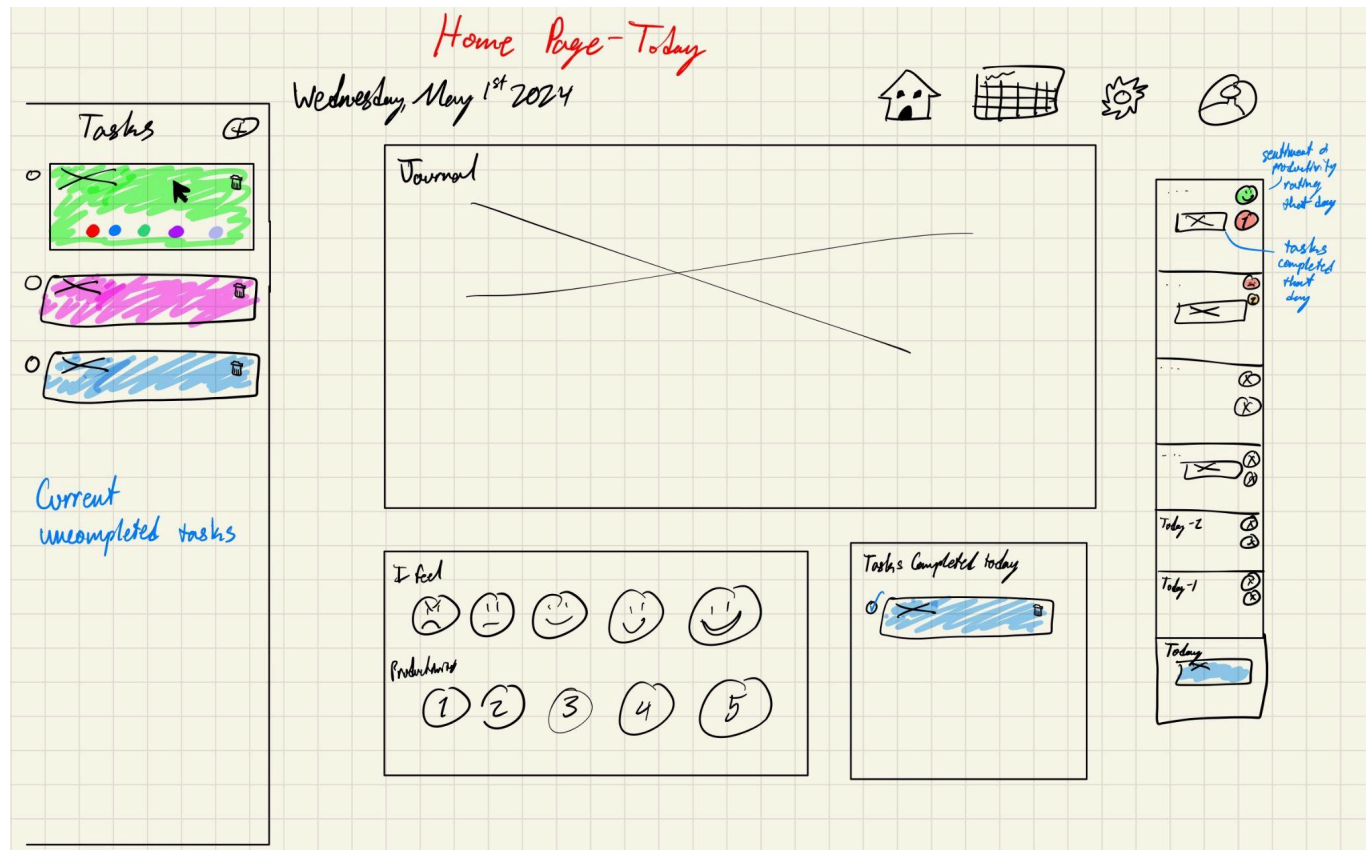
Usability

We plan to design UI elements according to common standards to keep functions intuitive to users.

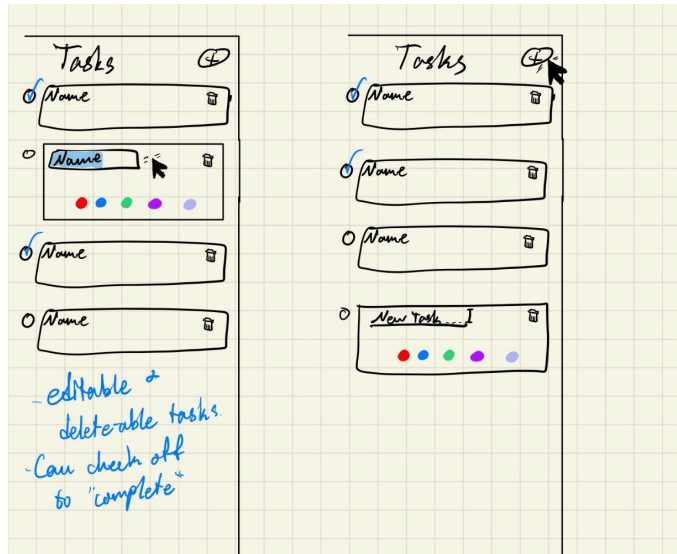
Satisfaction

We will spend appropriate time designing our app to be aesthetic, natural, and convenient for our users to ensure satisfaction.

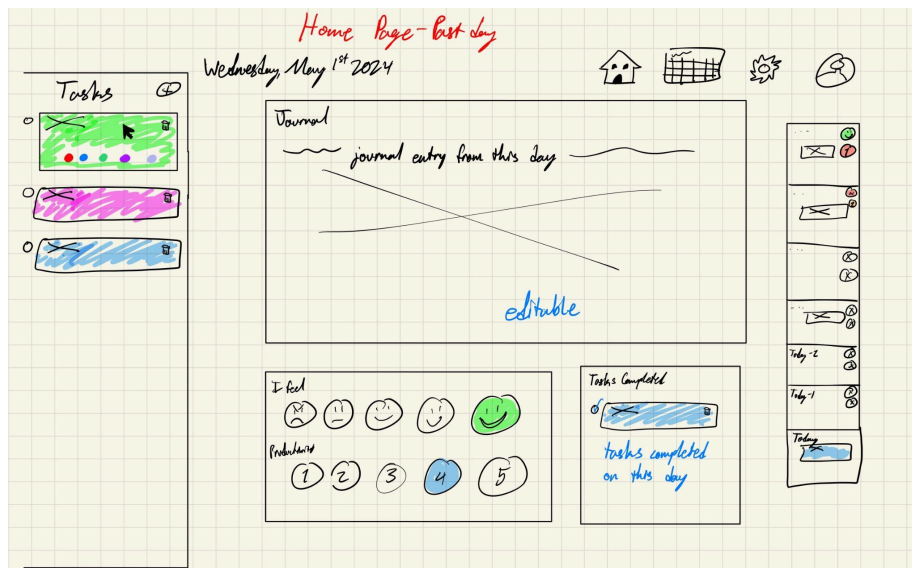
General Design



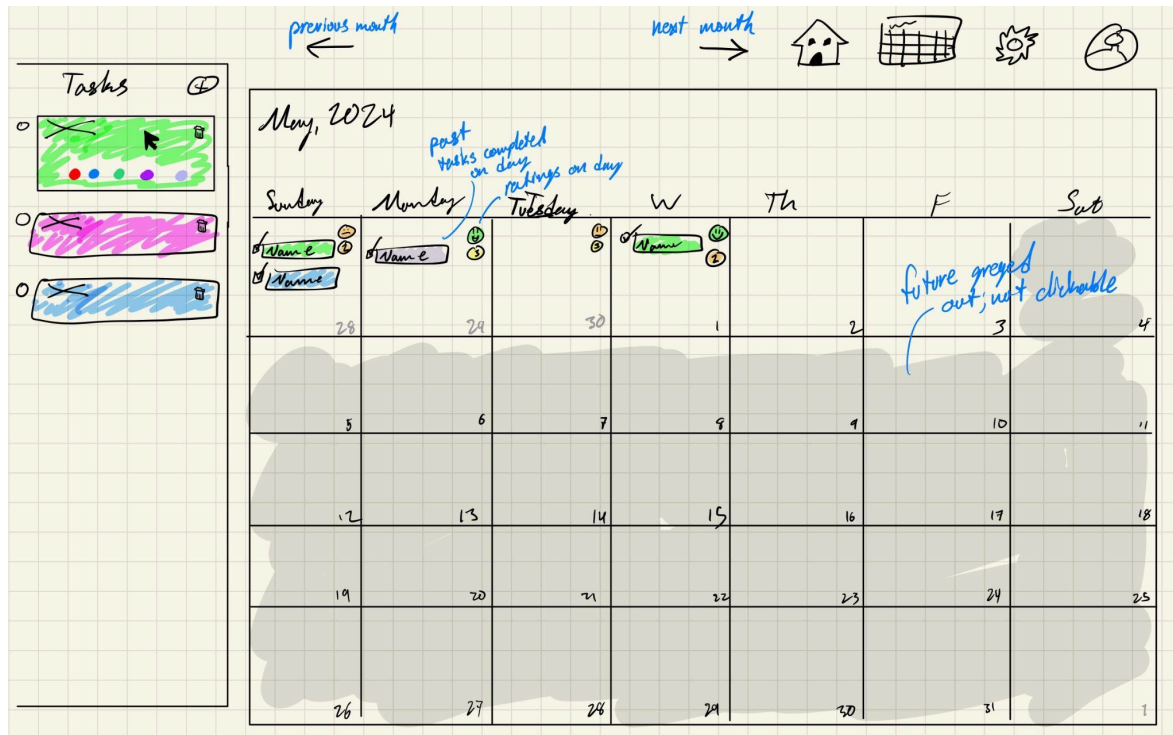
Here is a basic design of our home page. The top of the middle is the Journal Entry box which the user will write their journal entry for the day inside. Below that on the left is the Sentiment and Productivity Rating Widget where the user can select an emoji and number describing how they feel and how productive they were that day, respectively. The Task List is the tall box on the left and displays all uncompleted tasks the user has. Each task is editable, deletable, and can be moved around relative to other tasks. The Tasks Completed Today box is at the bottom right of the middle. When a user “completes” a task by checking the box to the left of each task, that task moves from the Task List to this box and is associated with that day. The rightmost box is the Past-Week Preview, displaying the current day and the past 6 days. The tasks completed and the sentiment and productivity ratings for that day are previewed on each of these days



Here is another view of the Task List feature. If a user clicks on a task, they will be able to edit the name. They will also be given a few colors to choose from which will highlight the task with the selected color. The Add Task button is in the top right and upon clicking, it will create a new task at the bottom which the user will name and select a color for.



This is the Home Page of a past day. It is identical to the Home Page of the current day but displays the journal entry, the tasks completed, and the sentiment and productivity rating all of that particular, past day. The Task List box and the Past Week Preview are still visible and display the same information as with the Home Page of the current day.



Here is the calendar page. It is accessed by clicking on the calendar icon at the top. It displays one month at a time and has forward and back arrows so the user can also view future and past months. Each past day will display a preview of the tasks completed and the ratings from that day. The user can click on past days which will take them to the Home Page of that day. Future days are grayed out and aren't clickable. The Task List is included on the left of this page as well. Clicking on the home button will send the user back to the home page of the current day.

Scope

Certainties: Everything mentioned above in Solution will be implemented.

Maybes: We have also talked about a full-scope search function. This search would take text input and search through all task names and journal entries the user has entered. We would implement this last if time permits.

Rabbit Holes:

We want to be able to save changes made to our calendar, but GitHub Pages doesn't seem to have that functionality. Rather, instead, we plan to explore alternatives to save the data within the website itself, something like Firebase or some other kind of hosting service.

No-gos

During the design process, our group specifically chose to avoid a few features that would either increase the scope too wide or lead to other bugs regarding functionality. The main focus of the webpage is as a developer journal therefore the main functionality should be for the user to type an entry every day about what they did and submit a quick survey of their productivity and mental state.

Hand-drawn sketch of an "Add Task" form on a grid background. The form includes three input fields: "Name", "Deadline", and "Details". To the right of the "Details" field is a vertical list of six colored circles (red, blue, green, purple, orange, teal) under the heading "Color". At the top left of the form is the text "Add + Edit Task".

Initial Add Task design

The main group of features that we avoided was limiting the scope of the tasklist. Our initial design included a text box for details and deadline date. The problem with this idea is that it would require many features to be added in the calendar page because: If the user is able to assign deadlines to tasks, then they should be able to see those tasks in the calendar and add tasks for future days. Therefore the user may use the webpage more so as a planner than a journal. To combat this, the user can only enter a title and assign a color to the tasks. This will shift the purpose of the tasklist to act more as a “reminder” list and focus the attention more on the journal capabilities.