Terp-To-Do: Your Time Management Assistant

Overview

Terp-To-Do is an app designed to assist you in managing your tasks and organizing your time effectively. Procrastination is a common issue among college students, and many of us have experienced challenges with time management. The stress of college studies often results in an overwhelming workload, making it crucial to have a reliable system for task management. Terp-To-Do was created to help college students tackle their busy schedules more effectively. As students ourselves, we know firsthand how overwhelming it can be to manage classes, assignments, and other commitments. That's why we developed Terp-To-Do to provide a simple yet powerful tool for staying organized and on top of tasks.

During our brainstorming sessions for this project, we all recognized our own shortcomings in time management. We believe this class offers us an excellent opportunity to utilize our programming skills to help others and engage in meaningful projects. Additionally, when discussing the MBTI personality test, we discovered that we all fall under the "sensing" category, preferring structured to-do lists to alleviate feelings of anxiety. Personally, I've had a habit of maintaining handwritten to-do lists since elementary school, but I find it inconvenient to make changes and sometimes forget where I noted down certain tasks.

Hence, the idea of developing an app for organizing to-do tasks emerged among us. This app would allow users to add descriptions and write reflections to aid in recalling important information. Furthermore, the trend towards a "paperless" lifestyle is gaining popularity among young people. Moreover, given that mobile phones have become an integral part of our daily lives, it is more convenient for most college students to use an app on their phones rather than carrying around a physical notebook. Additionally, there are many functionalities that we can add to our app, such as a timer and sending notifications, which couldn’t be easily achieved with notebooks.

Goals

We would like to set up three different sections: creating tasks, working on tasks, and after the task is completed.

Firstly, users should be able to add tasks and input basic information. For each task, they should be able to add a title, description, and deadline.

Secondly, we aim to improve the task execution process by adding a timer feature. After the task is created, when the user swipes the selected task to the left, there will be an option to set up a timer for focus. After entering the desired time, the user will be directed to the timer page.This will help users track their progress and manage their time effectively while working on tasks. Additionally, we are committed to designing the user interface with care to ensure easy navigation and access to important features.

When the user completes a task, we don’t want to simply discard it. Users will be able to write reflections on the task and take pictures to record their progress. For example, users can jot down challenges faced during the task and learn from them. Additionally, if they encounter a similar task in the future, they can recall the reflection, making the process more efficient. Including image functionality would be particularly helpful, as visuals can often convey more than words alone.

Furthermore, we aim to create a user-friendly interface, including a history feature. This allows users to review tasks completed in the past. helping them track their progress and identify areas for improvement over time.

As part of our stretch goals, we envision each user having a profile where they can earn experience points (EXP) upon completing tasks. Additionally, users can view their current level and completed tasks on their profile. This feature serves as motivation for users to engage with the app.

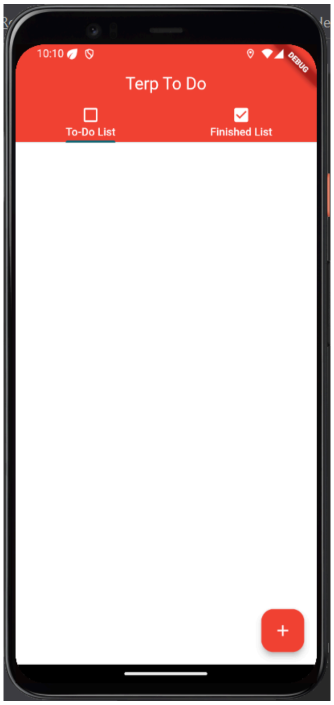
Additionally, to further improve task management, we aim to add organizational features such as categories, tags, or priority levels for tasks. This will allow users to categorize and prioritize their tasks effectively, ultimately boosting their productivity and time management skills.

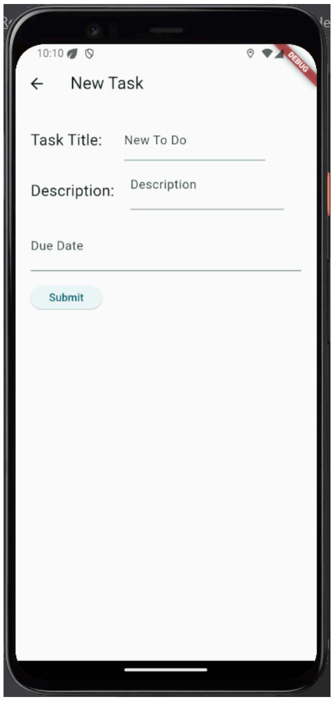
User Guide

After launching the "Terp To Do" application, you will be presented with the to-do list page. There is also a tab for finished tasks and we will talk about that later. You can click on the plus button at the bottom right corner of the page to add a new task. After clicking on the button, you will be directed to the new page, but you can go back by clicking on the arrow button at the top.

Then you can enter the title and description and set a due date. A calendar will show up when you are setting a new date, and you can click on the date you want. You can cancel that or click "OK" to set the date.

Finally, click on the submit button, and you'll be returned to the main page where your new task will be displayed in the to-do list.

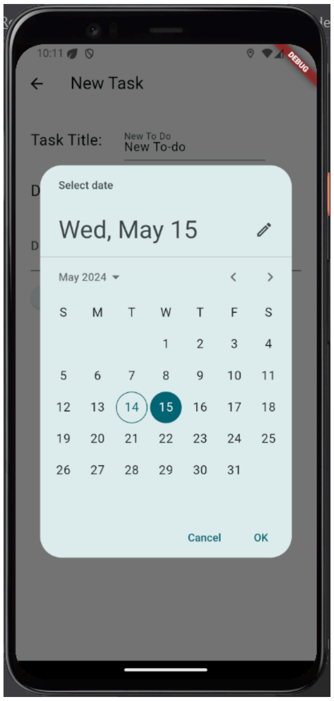




Task Title: New To Do

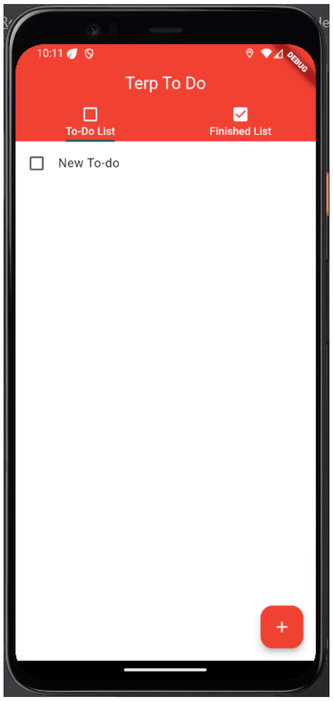
Description: Description

Due Date



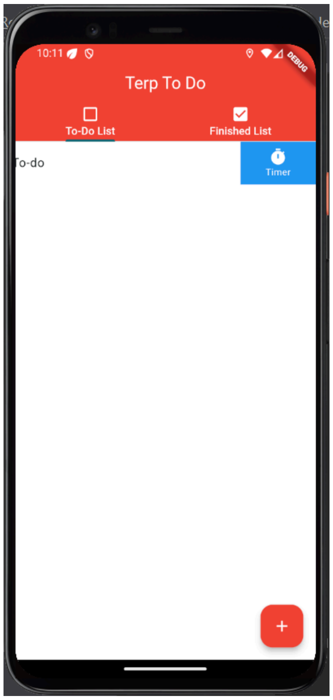
Task Title: New To Do New To-do

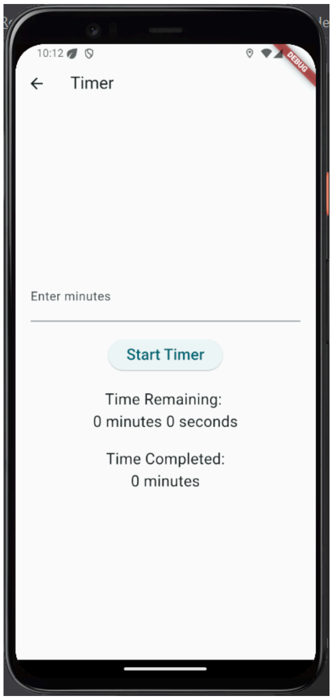
Wed, May 15



□ New To-do

If you swipe the an incomplete to do to the left, you will see a clock icon, clicking on that will direct you to the timer page



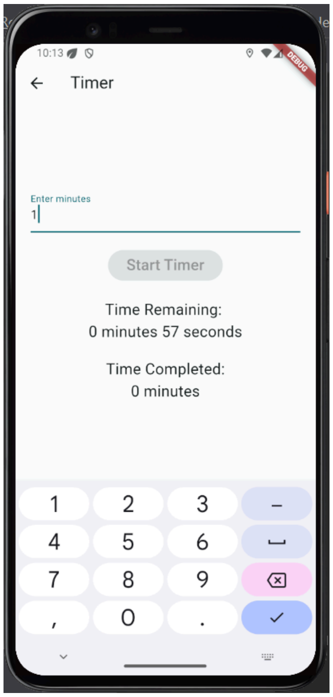


Enter minutes

Time Remaining:
0 minutes 0 seconds

Time Completed:
0 minutes

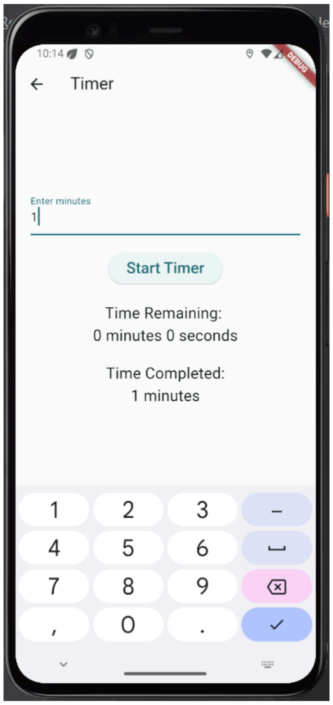
You can enter how long you would like to focus for the current task. The time page will not automatically turn off, and the timer will break if you leave the current application to prevent you from being distributed by another app on your phone. Once you finish the timer, it will be accumulated and be shown under the timer.



Enter minutes
1

Time Remaining:
0 minutes 57 seconds

Time Completed:
0 minutes

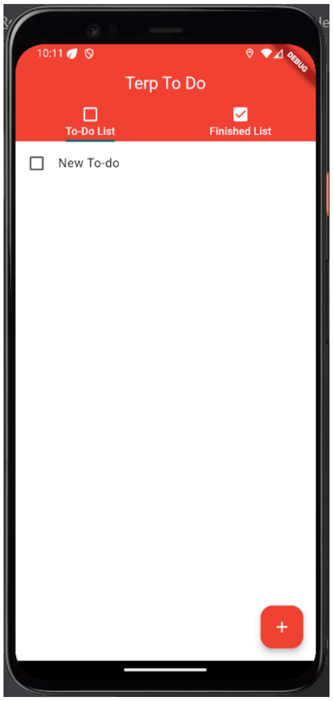


Enter minutes
1

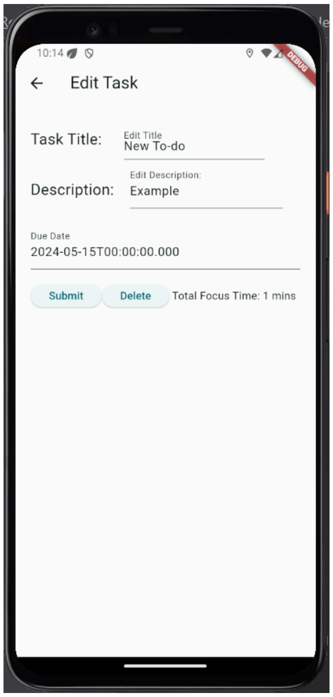
Time Remaining:
0 minutes 0 seconds

Time Completed:
1 minutes

You can also edit the task if you want. Long pressing on the todo that you would like to edit will direct you to the edit task page.



☐ New To-do



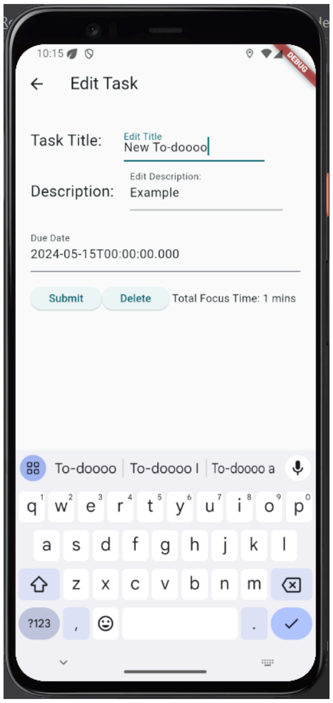
Task Title: Edit Title New To-do

Description: Edit Description: Example

Due Date 2024-05-15T00:00:00.000

Total Focus Time: 1 mins

After making changes and submitting them, you will be able to see the information change as desired.

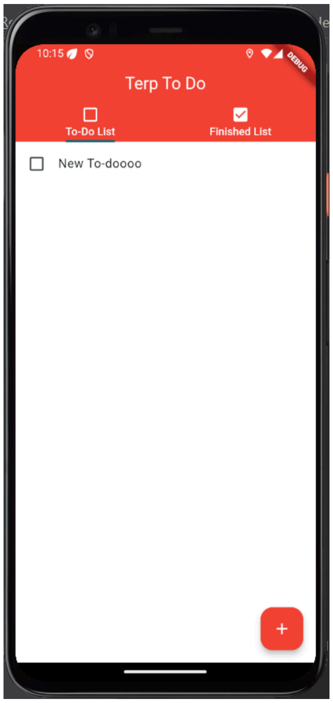


Task Title: Edit Title New To-doooo

Description: Edit Description: Example

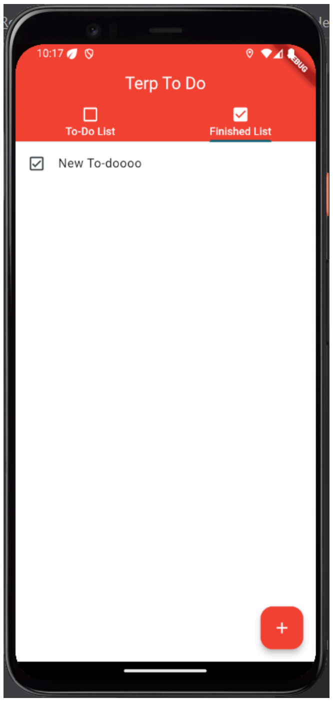
Due Date 2024-05-15T00:00:00.000

Total Focus Time: 1 mins



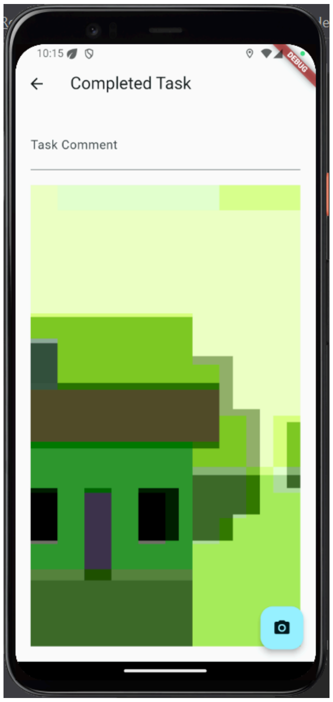
☐ New To-doooo

To complete a task, simply check it, the completed to do will now be in finished list



☑ New To-doooo

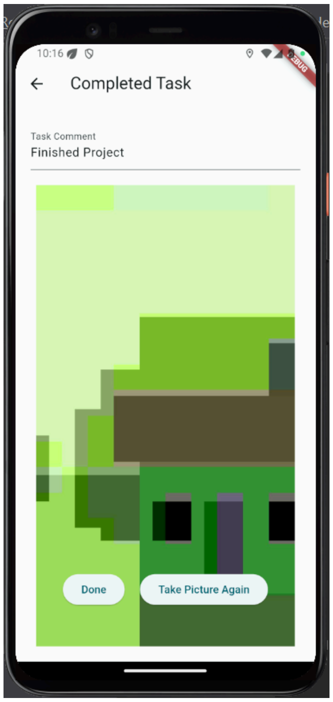
For completed tasks, you can add a comment and a picture. We are using the virtual image generated by the emulator, so it looks like a Duolingo owl. You can change your emulator settings and use the camera of your laptop or a different camera. You can click on the camera button to take a picture. If you are unsatisfied with the image, you can click on the "Take Picture Again" button. Otherwise, you can click on the "Done" button. After that, there will be a confirmation page showing you the information you just entered. You can go back to the main page by clicking on the button at the bottom of the screen.



10:15

Completed Task

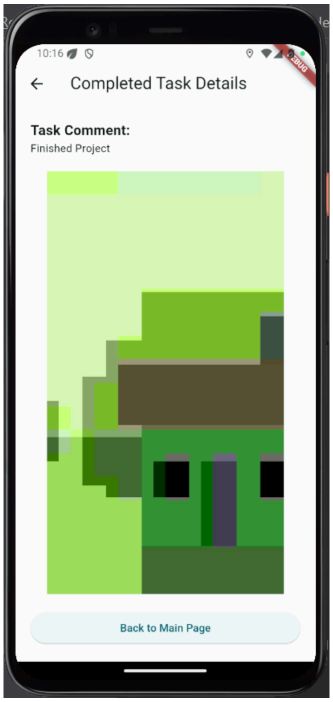
Task Comment



10:16

Completed Task

Task Comment
Finished Project

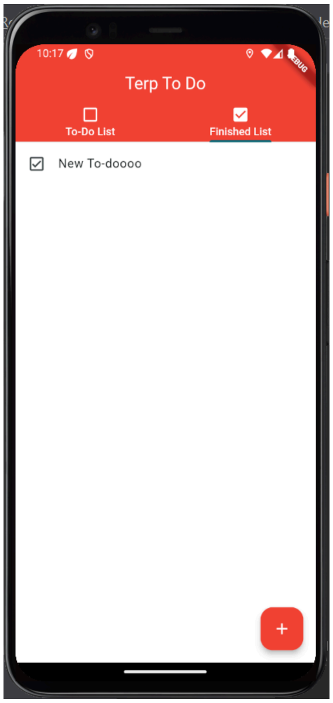


10:16

Completed Task Details

Task Comment:
Finished Project

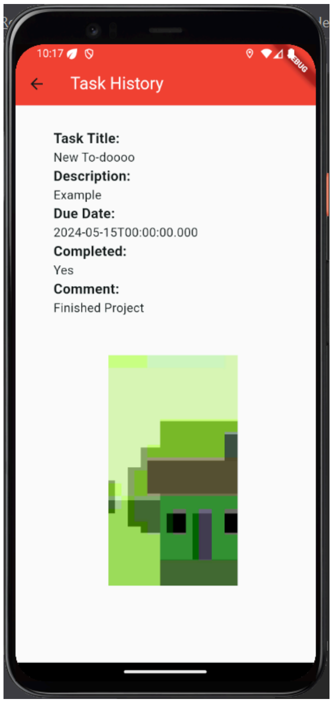
After you check the box in front of each task name, it will be moved to the "Finished List" page. You can click on the task to view its history, including the title, description, due date, status, comment, and an image. All the data will be persistent. If you uncheck the box, it will be considered an incomplete task and will show up on the "To-Do List" page.



10:17

Terp To Do

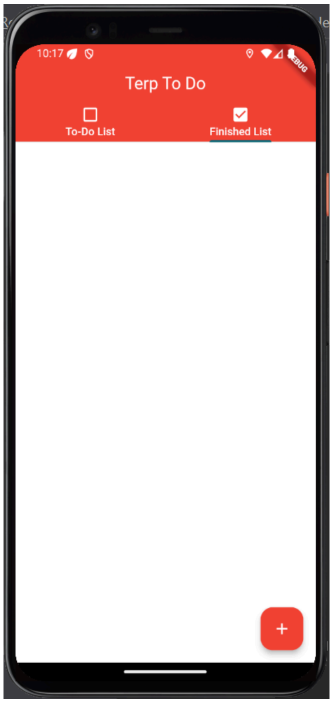
☑ New To-doooo



10:17

Task History

Task Title:
New To-doooo
Description:
Example
Due Date:
2024-05-15T00:00:00.000
Completed:
Yes
Comment:
Finished Project



10:17

Terp To Do

Discussions

One significant challenge we encountered was task assignment and integration. To overcome this hurdle, we instituted regular group meetings to discuss progress, offer suggestions, and ensure equitable task distribution. This practice facilitated fairness and workload balance among team members. During coding sessions, each team member operated on an individual branch. However, particularly in the initial stages, merging proved problematic. Extensive modifications to fundamental elements such as the home page and basic classes often led to conflicts during merging, necessitating substantial rewrites.

When tackling the challenge of persisting data effectively in milestone 2, we encountered several issues that resulted in a comprehensive overhaul of our approach. Firstly, it became apparent that the existing data structure was not optimized for persistence, leading to inefficiencies in accessing and saving data. To address this, we restructure the way existing data were organized and create a new class. Additionally, we need to make sure how other widgets of our project access the persistent variables are updated accordingly.

As the project progressed, we adapted by creating more of our own classes and files, which mitigated conflicts to a considerable extent. Additionally, during milestone3 and stretch goals, we implemented coordinated coding efforts to prevent conflicts. Communication regarding individual work schedules ensured that only one team member modified the code at any given time, facilitating smoother integration. Furthermore, during group meetings, we strategized how to implement new functionalities and identified areas of code requiring modification. This proactive approach streamlined the code merging process and minimized conflicts, enhancing overall efficiency and productivity.

In the second milestone, figuring out how to add the camera feature proved challenging, as it wasn’t a topic covered in class. Consequently, we conducted extensive research to address this issue. Additionally, ensuring proper storage of images for persistence posed another hurdle. During testing of the camera feature, we initially encountered an issue where the camera displayed a pre-recorded video of a cartoon character. Furthermore, the captured images appeared identical, leading to extensive debugging efforts. Subsequent research revealed that this anomaly was due to the emulator we were using, which relied on virtual images instead of the actual camera on our laptops. After updating the emulator settings to allow access to the actual camera, the functionality worked as expected.

For the third milestone, we have a conflict between expectation from the instructor and the existing implementation. For the project proposal, Dr. Marsh explicitly stated he expects that the timer will continue to work reliably if the screen turns off. However, since the intention of using a timer is to prevent users from leaving the application and force them to focus on the current tasks, we would like the timer to break if the user leaves the application. In milestone 2, we successfully satisfy Dr.Marsh requirement on the timer to run even when the screen turns off, however, adding the application will break once users leave the timer on such implementation is harder than we think. We have used multiple packages to see if we can have the timer to run if the screen turns off but stops when the user leaves the application and unable to resolve the problem. Therefore, we contacted TA about whether preventing the timer from automatically turning off can be an alternative solution, and worked on it once we have confirmation on that.

Following the addition of a page allowing users to add comments and images, we needed to ensure proper storage of this information. We also considered adding a confirmation page to display before submission. Although this was not included in our initial proposal, we believed it would be beneficial to allow users the option to retake images and revise comments before finalizing their submission. Initially, we planned to store the images as part of milestone3, with the addition of a history page in our stretch goal. However, we realized that after the user confirmed the information, we could lead them to a page displaying the details, effectively serving as the history page. As a result, we worked on these two tasks concurrently.

For stretch goals, the initial plan for the history page was simply a page that displayed the title of all completed tasks, and completed tasks would disappear from the to-do list. However, since the user might mark something wrong accidentally, the initial plan will prevent the user from undoing the actions. We later decide to separate the completed todo and the incomplete todo, allowing the user to uncheck a todo if they make a mistake. Meanwhile, we also provide a detailed history page for the completed task where users have provided reflection and take pictures.

Potential future directions

Unfortunately, we had numerous ideas we wanted to incorporate into our app, and there were many functionalities of Flutter that we wished to explore. However, due to time constraints, we were unable to implement them this semester.

In the future, our aim is to introduce the functionality to "create an account," allowing users to log in with their email or username. With their own accounts, users can earn experience points (EXP) and coins upon completing tasks, transforming the app into an engaging game-like experience. Furthermore, we would like to enhance gamification by providing a game-store that will allow the user to customize their profile through the coins they received, provide more feedback and motivation.

Additionally, we plan to include organizational features such as categories, tags, or priority levels for tasks. As the number of tasks grows, finding specific ones becomes challenging, making organization increasingly important.

Another feature that we like to do is one-time notification when the due date that user set for compete

We intend to store user data on an online platform, such as Redis, enabling users to recover accidentally deleted tasks or accounts by sending requests. Moreover, we aim to implement measures to protect users' personal information. We may need to consult professionals about security and privacy terms.

Given that the app is named Terp-To-Do, originally designed for UMD students, it would be beneficial to integrate it with ELMS or other course platforms. This integration would allow students to import assignment information from course descriptions, eliminating the need for manual entry. For example, if students can log in to Terp-To-Do with their school email, when professors assign a new task, the same task will show up on Terp-To-Do.

Furthermore, we aspire to enhance the user interface to impart a more professional appearance or incorporate additional animations to improve user engagement. It would be helpful if we could include images like our Terpeiz project, which would make the app more engaging to look at.