



NUS ORBITAL 2024

Milestone 3

**Wong Sheen Kerr
Nicholas Cheng De Fei**

Table of Contents

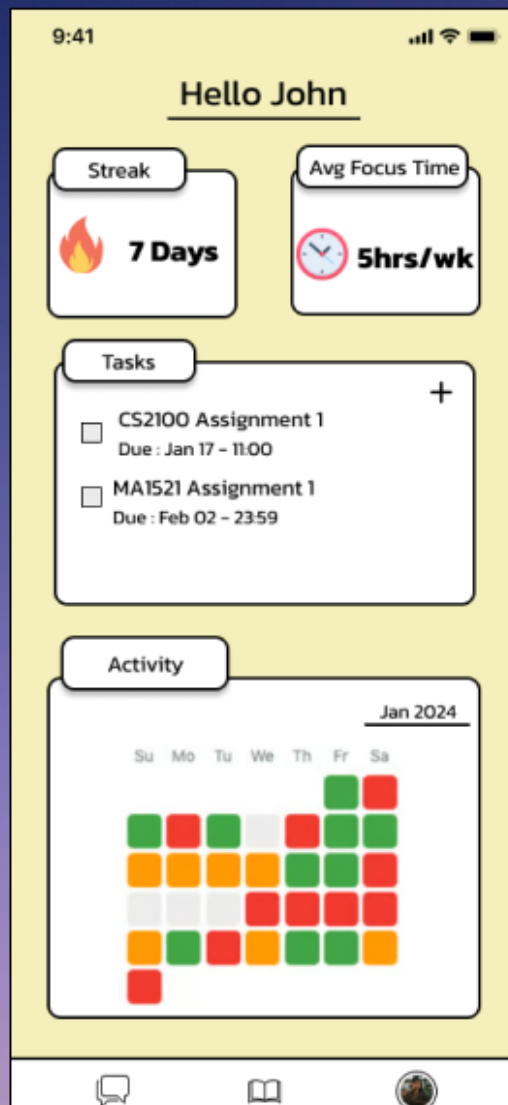
Posters	2
Proof-of-Concept	4
Video demonstration	4
Time Log	4
Deployment	4
Proposed Level of Achievement	4
Goal & Vision	5
User Stories	5
Project Scope (One-Sentence Version)	6
Project Scope (Descriptive Version)	6
Tech Stack	6
Features	7
Authentication System	7
Description	7
Email Verification Extension	7
Onboarding Screen Extension	7
Challenges Faced	8
Images	9
Timer	17
Description	17
Challenges Faced	17
Images	18
Streaks	19
Description	19
Challenges Faced	19
Images	20
Task List	21
Description	21
Challenges Faced	21
Images	22
Heatmap	23
Description	23
Challenges Faced:	23
Images	24
Gamification	25
Description	25
Challenges Faced	25
Images	25
Thread	27
Description	27
Loading Screen	27
Challenges Faced	27
Images	28
Activity Flow Diagram	33
Use Case Diagram	34

Database Model Diagram	35
Firebase Firestore	35
Database Rules	36
Development Plan	37
Github Usage	39
Scrum Board	40
Milestone 1	40
Milestone 2	40
Milestone 3	41
Version Control	41
Pull Requests	42
Quality Control	43
User Testing	51
User Interface	51
Functionality	54
Timer Feature	56
Task List Feature	58
Gamification Feature	60
Thread Feature	62

Posters

Distraction is mantlers

Say goodbye to distractions



Main Features



Focus Mode

Help users focus by blocking distractions



Study Groups

Find peers taking similar mods for help



Activity Tracker

Visualize usage on the application



Study Helper

Provide tips for user on better studying habits

Additional Features



Task List

Allows users to keep track on submissions or important tasks



Motivational Reminders

Send reminders to users on when they intend to study



Gamified Focus Mode

Make studying more rewarding by incorporating gamification concepts

Makers: Wong Sheen Kerr & Nicholas Cheng De Fei

Powered By:



Firestore

Liftoff / Milestone 1 Poster



Distractionismantlers

Powered By:

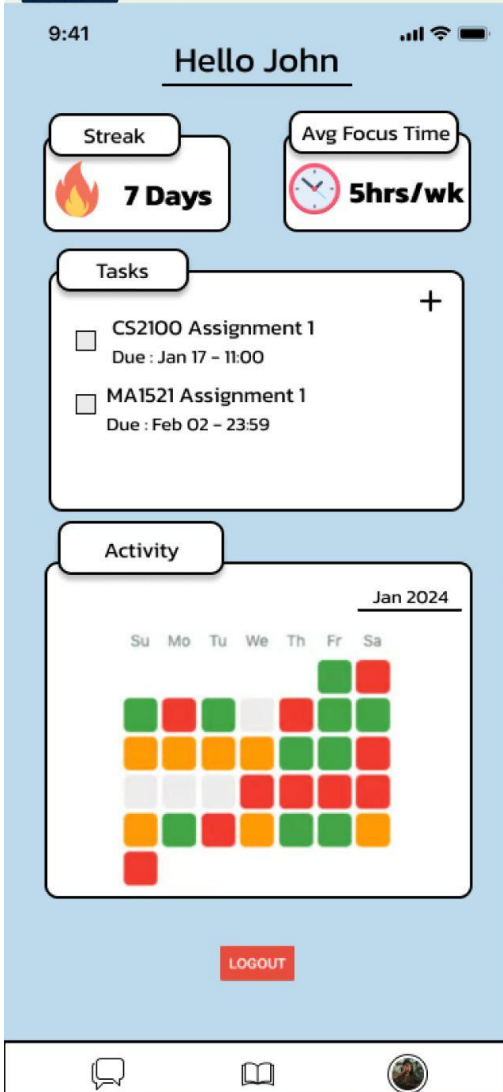


React Native
Firebase



Team 6089

Wong Sheen Kerr
Nicholas Cheng De Fei



Main Features



Focus Mode

Help users focus by blocking distractions



Task List

Allows users to keep track on submissions or important tasks



Activity Tracker

Visualize usage on the application



Timer

To motivate users by allowing them to track their study timings

Additional Features



Motivational Reminders

Remind Users to study



Study Groups

Find peers taking similar courses for help



Study Helper

Provide tips for user on better studying habits



Gamified Focus Mode

Make studying more rewarding by incorporating gamification concepts

SWE Practices

Version Control

Git is used for version control, enabling collaboration, tracking code changes, and resolving conflicts.

Agile Development

Agile is adopted for iterative development, providing flexibility to changes, including regular stand-ups, sprint planning, and retrospectives.

Code Reviews

Code changes are reviewed before merging to maintain quality and share knowledge.

Continuous Integration/Continuous Deployment (CI/CD)

A CI/CD pipeline is established for automated build, test, and deployment, aiding early problem detection and faster software delivery.

Test-Driven Development (TDD)

TDD is adopted, writing tests before minimal code and then refactoring.

Security Practices

Application security is ensured with best practices, secure coding to prevent vulnerabilities, and regular updates to avoid risks from outdated libraries.

Milestone 2 Poster

Proof-of-Concept

Video demonstration

- 1) Milestone 1: 📺 Milestone 1 Recording.mp4
- 2) Milestone 2: 📺 Milestone 2 Video.mp4
- 3) Milestone 3: 📺 Milestone 3 Video.mp4

Time Log

📅 Orbital Time Log

Deployment

Link to the APK file: [Click Here](#)

Link to the code base: [Click Here](#)

Proposed Level of Achievement

Apollo 11

Goal & Vision

Distraction dismantlers aim to provide a solution to student's distraction issues preventing them from being productive. This application provides the necessary functionalities to help and assist in a student's day to day activities and keep them focused on their studies.

Currently in the digital age, students often find themselves constantly engrossed on their phones not realising that they are being distracted from what they are doing. Now with a myriad of social media platforms for entertainment purposes, not only students but many people find themselves spending too much time on their electronic devices. The constant engagement and fixation can lead to distractions, especially when they face academic challenges.

Recognising this, our project aims to provide a tool that will assist students in their academic journey at NUS by providing :

- A healthy way to keep students focus on their work for a set period of time
- A easy and simple way to keep track of the deadlines that are due
- A friendly platform to find other students to study with or to seek help from them when encountering difficulties

User Stories

1. As a student, I get easily distracted by my phone and want to have a tool that can help me focus on my studies.
2. As a student, I want to have a tool that can help me reach the flow state more easily when studying.
3. As a student, I want to be able to connect with other students who are studying the same subjects so that we can help each other out.
4. As a student, I want to compete with my peers in a healthy way, so that I can improve my focus and study habits.
5. As a student, I want to see my progress visually, so that I can understand how much I have achieved.
6. As a student, I want to compete with my peers in a healthy way, so that I can improve my focus and study habits.
7. As a student, I want to have access to a community of learners, so that I can share my experiences and learn from others.

Project Scope (One-Sentence Version)

Create a platform that assists NUS students in maintaining focus on their studies by providing gamification features and facilitating peer interaction.

Project Scope (Descriptive Version)

Our goal is to introduce functionality and gamification features that will help students who are easily sidetracked to concentrate on their studies. We also aim to create a social media-style platform where students can interact, seek help, and find study partners or project collaborators. This platform will not only aid them academically but also foster a sense of community, enhancing their overall academic experience.

Tech Stack

Distraction Dismantlers is a mobile application built using React Native for both the user interface and backend. We utilize Firebase for our database needs and Expo CLI for running and debugging the application, ensuring compatibility for immediate testing on mobile devices.

To enable users to save their timers and manage their usernames in the forums, we have implemented an authentication system that requires users to sign in before accessing Distraction Dismantlers.

With Distraction Dismantlers, users can seamlessly navigate through various study management features via the navigation bar located at the bottom of the screen. These features include the user profile, forums page, and study resources page, each of which will be detailed in the following sections.

Features

Authentication System

[Completed]

Description

Distraction Dismantlers is a personalised focusing app with social media elements like posing on forums, every user must be authenticated with a unique account tied to their email.

The main sign in method used will be their email and a password. Utilising Firebase authentication SDK, a user can do the following actions:

1. Sign in with a email address, set their username as well as a password
2. Login with an account that is tied to their email address

For both login and sign up, user actions are thoroughly checked before making the relevant calls to Firebase authentication. For instance when registering an account if any field is left blank or if the password does not match error messages will notify the user.

Firebase authentication SDK also has inbuilt functionality for checking for duplicate email addresses and password regex to prevent weak passwords based on what the user has keyed in.

Only upon successful login or account registration will the user be routed to the home page. And if the user registered an account it will be reflected and stored in the Firebase authentication.

Email Verification Extension

[Extension Completed]

One additional feature added as an extension was the addition of email verification. Upon a successful registering of an account the user will need to validate his or her email before the next login attempt.

The sending of emails will be handled by Firebase Authentication SDK as well as the tracking of the validation of the email. And if the user does not verify their email, they will not be able to login and they will be notified as well.

However if they do validate their email they can login as per usual in the future.

Onboarding Screen Extension

[Extension Completed]

Another additional feature added was to add an onboarding screen which will only show once when the user opens our app for the first time. This acts as a simple way of communicating to our users and letting them know what our application does and what it can provide.

However subsequent opening of the app will not show this screen and will then direct the user to the login page immediately.

Challenges Faced

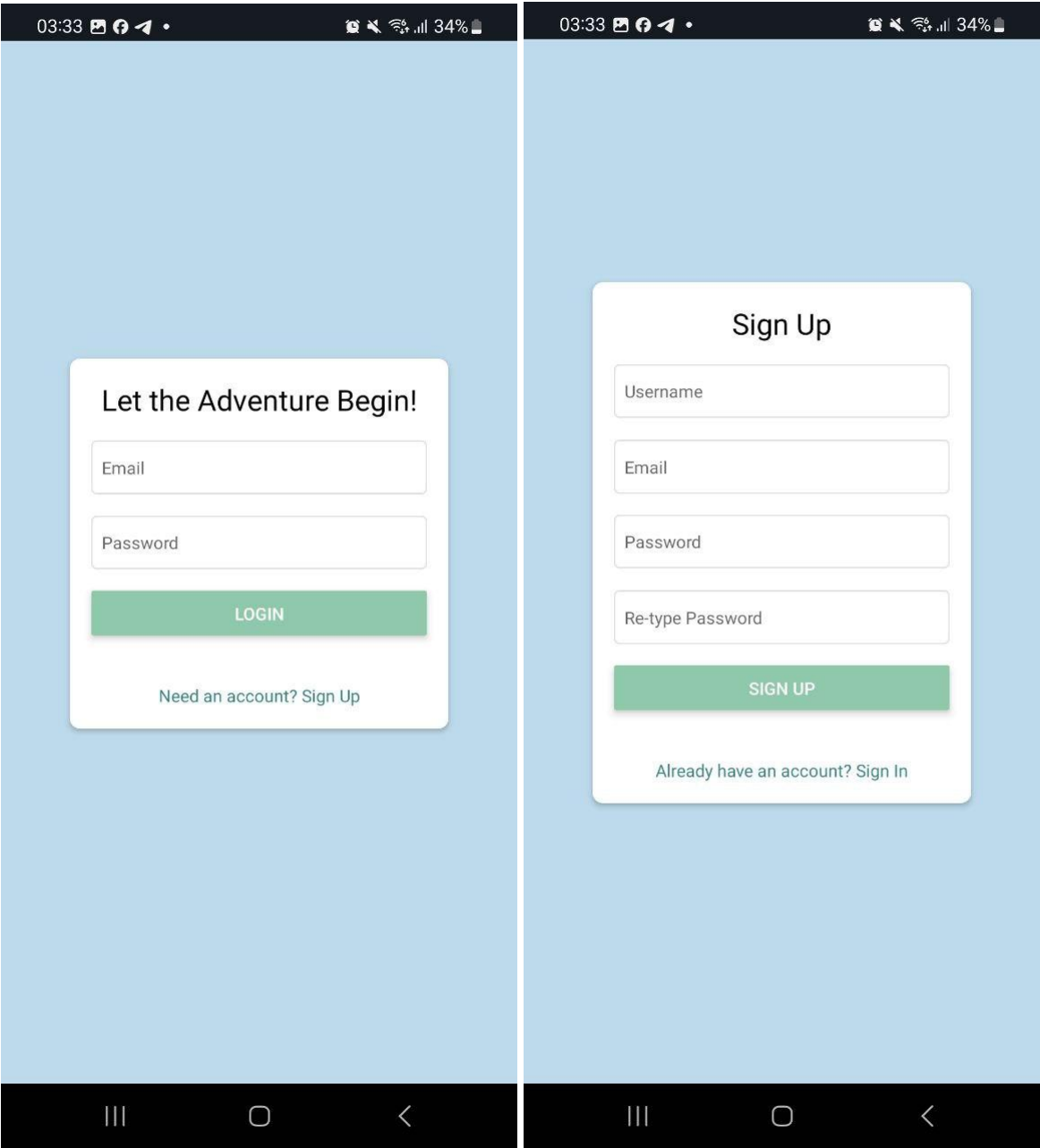
The key challenges for the authentication system are the navigation between the login, register and home pages as well as the usage of Firebase authentication. This is because it was the team's first time using Firebase authentication as well as react native and in the initial phase of the project it can be confusing as to know what we need to set up on the Firebase authentication as well as on react native and know what the various functions does to correctly implement the functionality needed for this feature.

As for navigation, unlike web applications where navigation to another page is simply by redirecting to another html link, for mobile applications it was different but after a bit of research, react has some libraries which helps in navigation on mobile apps by using tabs.

As for the email verification feature the team discovered the "emailVerified" field, which indicates whether a user's email has been verified. Additionally, we researched how to use Firebase authentication to send verification emails.

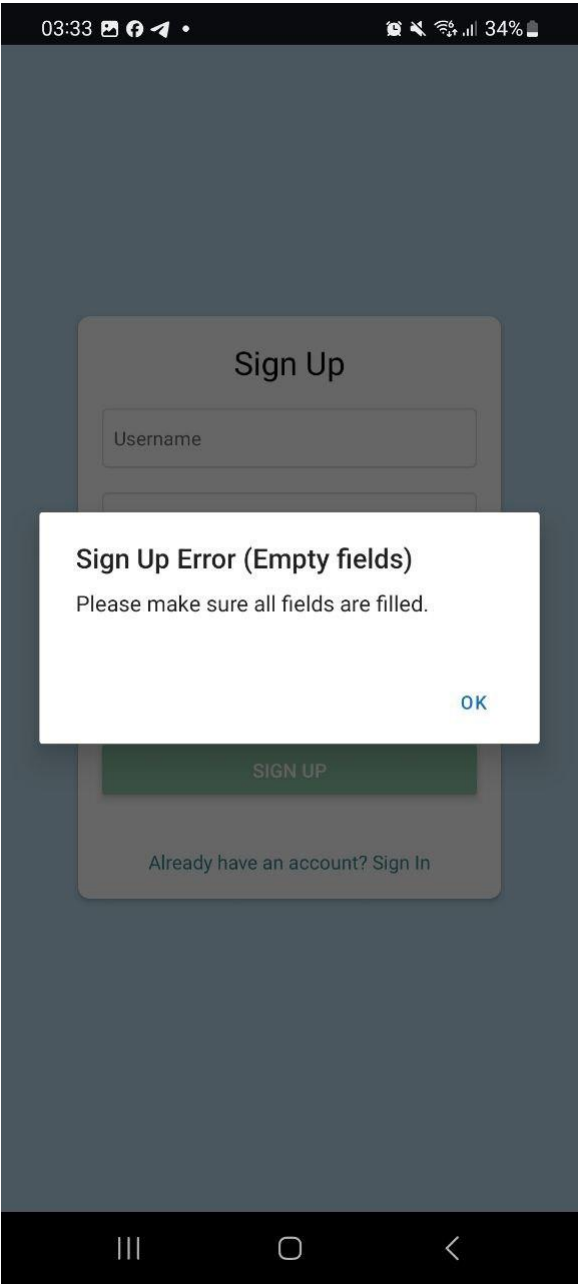
For the onboarding page, the main issue was the front end development. This is because the feature was added early on in the development and the knowledge of react native was limited at the time.

Images

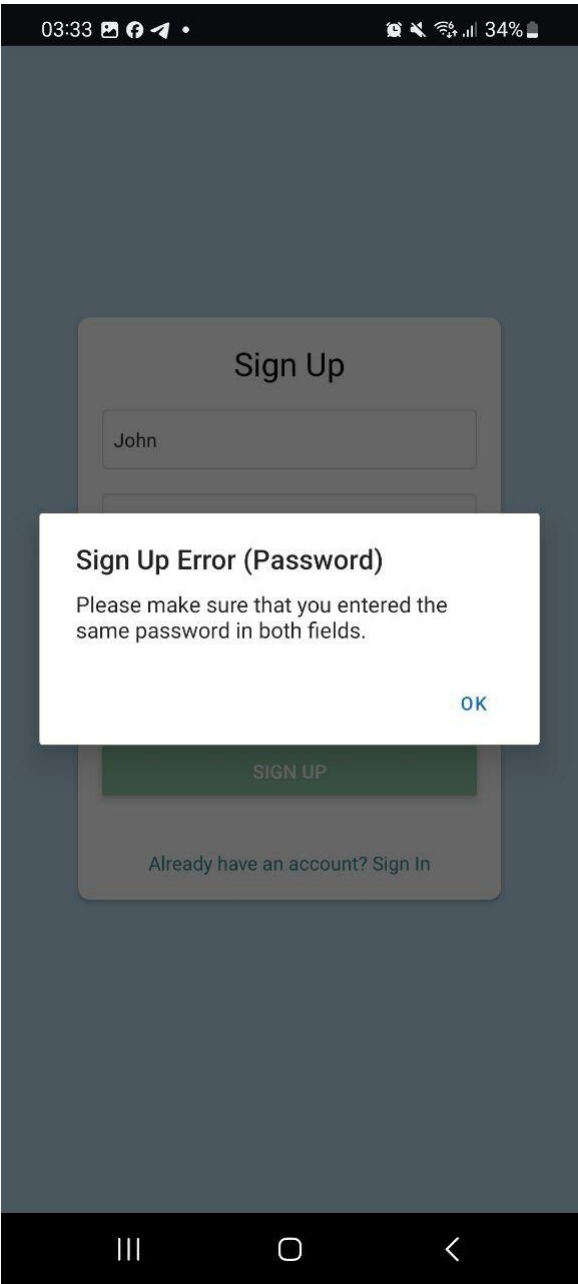


Error Messages :

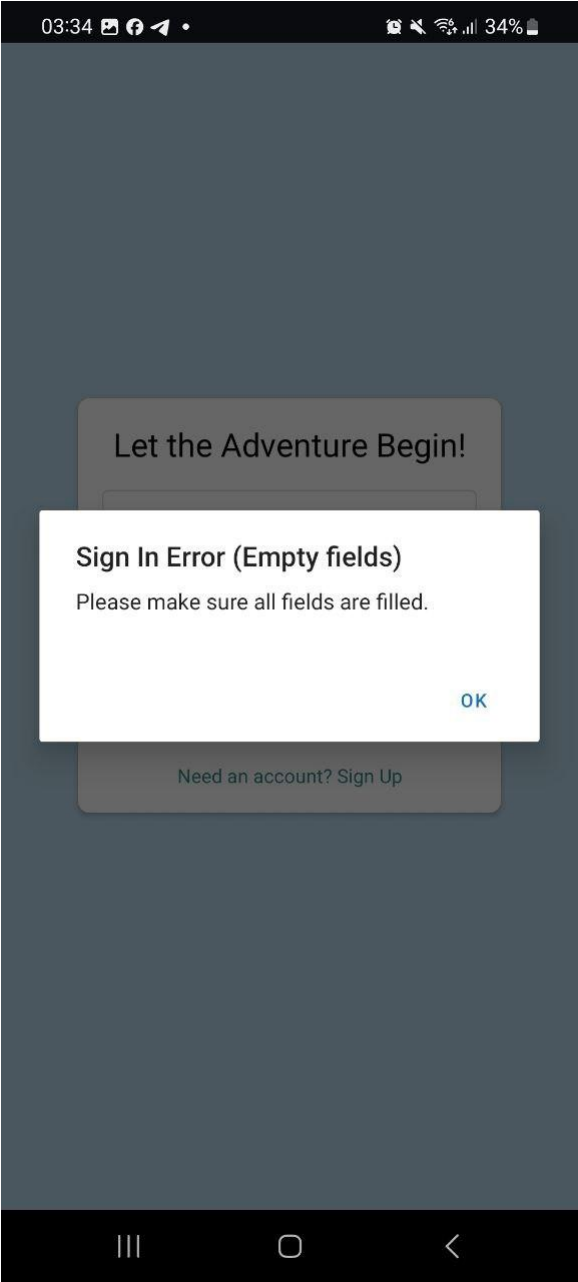
- 1) Sign Up: User left at least one of the fields empty



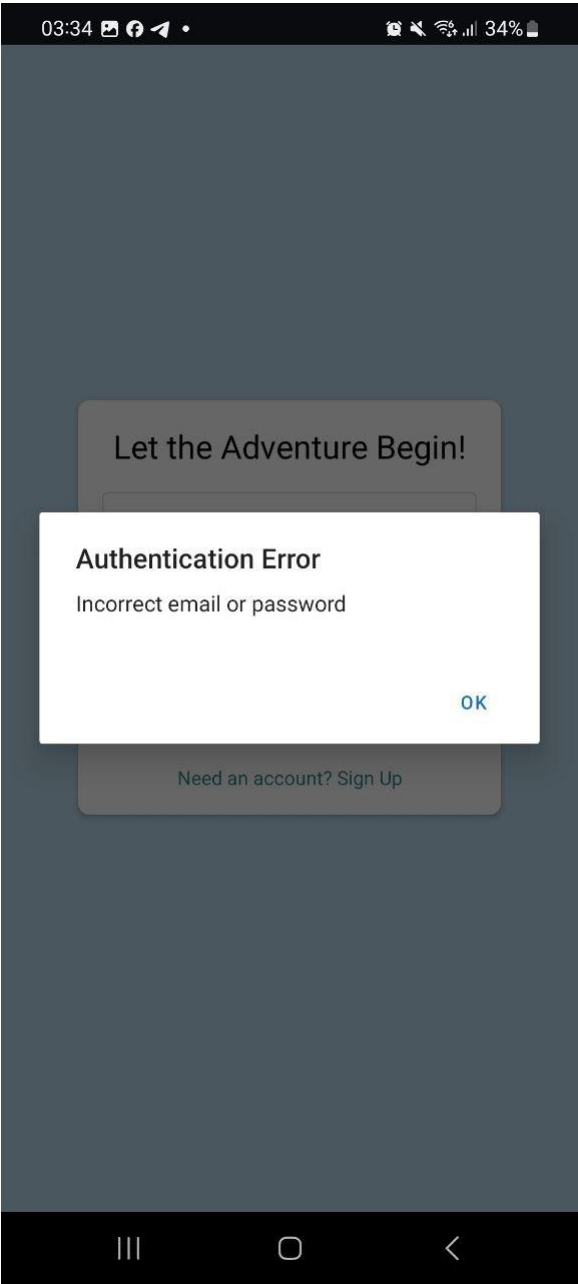
2) Sign Up: User did not key in the same password in both password and password confirmation fields



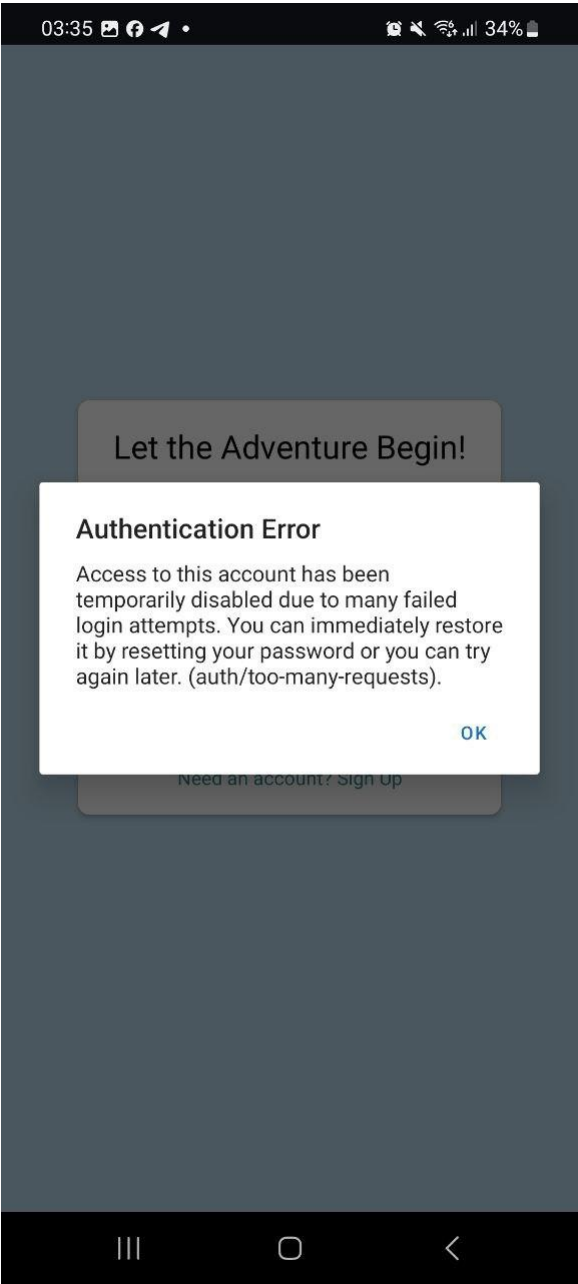
3) Sign in: User left either their email or password field empty



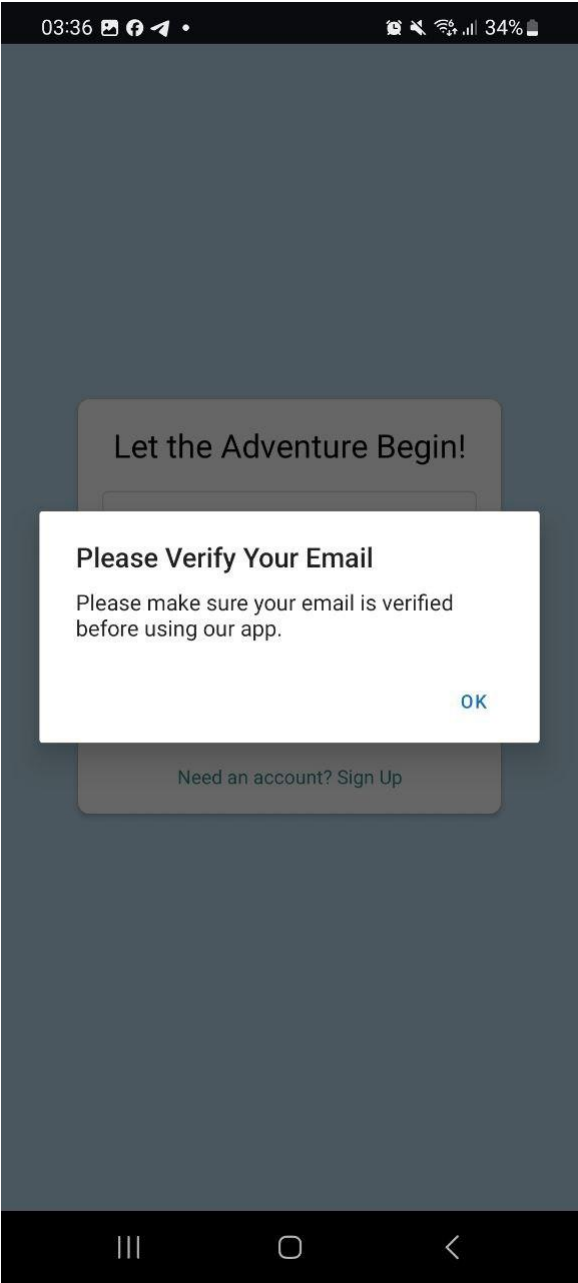
4) Sign in: User did not sign in successfully (incorrect email or password)



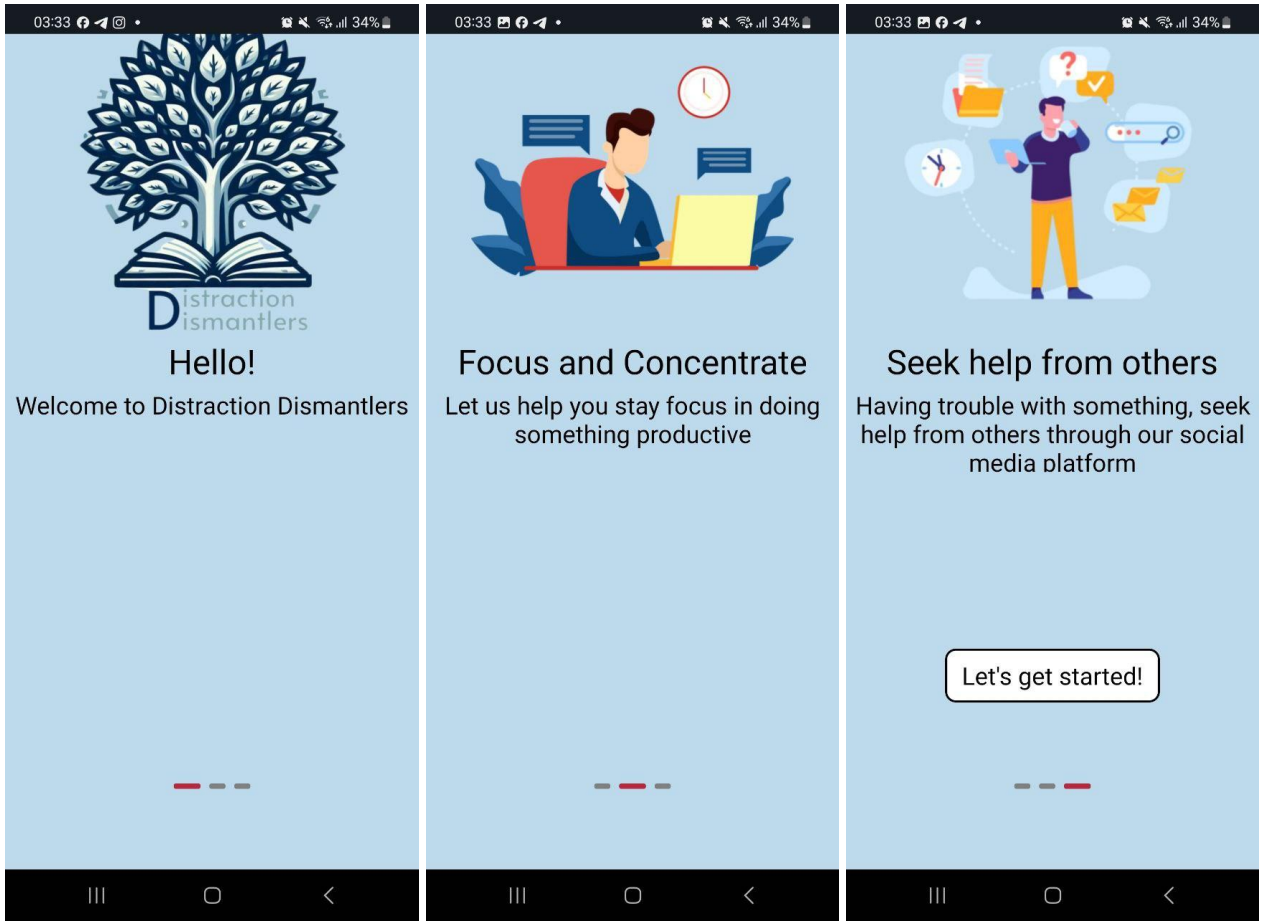
5) Sign in: User tried to sign into the same account too many times



6) Sign in: User did not validate their email



Onboarding Page:



Timer

[Completed]

Description

Upon logging in successfully, the user will be brought to the timer page which will be the home page of the application. On this page it will house the timer feature and its main goal is to help students focus for a set duration specified by the user.

How it works is that there will be 2 scroll bars indicating the time to be set, in increments of 5 minutes or 1 hour. The user can scroll to set the time on how long they want to focus. Here the user can do the following:

1. Set the time for the timer to count down
2. Start the timer by pressing the start timer button on screen
3. The user can wait for the timer to end which will then bring the user back to the timer page
4. The user can also preemptively stop the timer which will then also bring back to the timer page

There is also some back end functionality as well. For each successful study session which is defined as the user studying until the timer has reached 0, the session will be recorded into the Firebase firestore database which will store the date and the duration of each study session. This will then be used for later features like the profile and streak feature. Also the time that the user has set will be remembered even when a session has ended or the user navigates to other pages.

Challenges Faced

One issue faced here was that early on during the development, firebase testing mode expired and thus the read write access to the database was all blocked. As such, we need to learn and write the necessary permissions to allow the users to write and read the database for records owned by the user. This required the users unique identification id which was handled by firebase authentication and as such we need to pass the id from authentication to our application and then to our firestore database to read and write to the database.

Another issue faced was writing the front end for the timer element as since it was not a static element it has to be updated every time the timer goes down by 1 second. In addition, rather than displaying a static color wheel, having a gradient can also help indicate progression throughout the study session.

Images

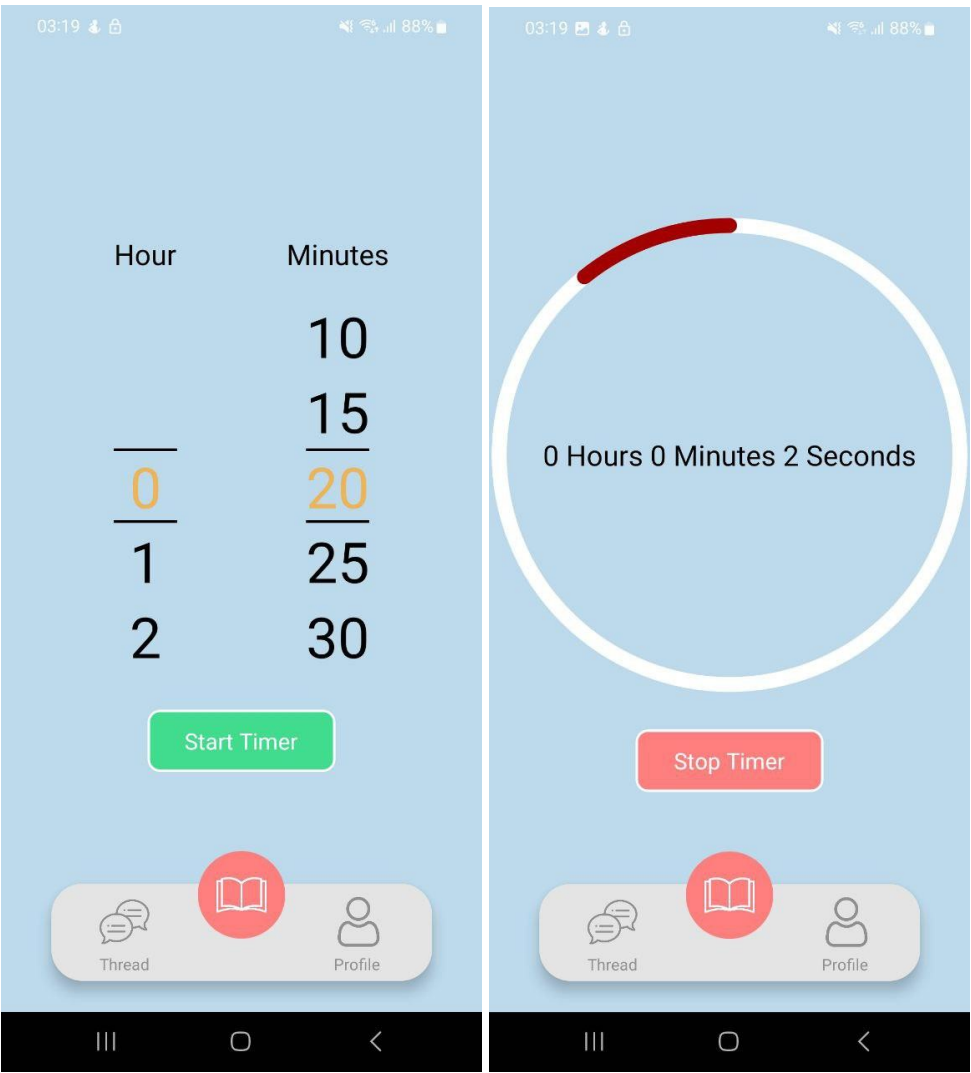


Image : Elements in the Timer Feature

Streaks

[Completed]

Description

Inside the profile page houses the **streak feature**. This is one of the gamification features which will display the number of consecutive days a user has completed at least once. In the event a user did not complete 1 session for a day the streak will be reset to 0 once midnight passes.

The streak feature works as follows:

- Upon registering an account the counter will be 0
- Once the user has completed 1 study session it will increment by 1
- Afterwards any subsequent sessions in the same day will not increase the counter
- In the even a user forgets to complete 1 session before the end of the day it will reset to 0

Another tracker implemented along the streak feature is the **average study time**. This tracker displays the average study time for the week for each individual user. The formula to calculate the average study time is as follows:
Average study time = Total time studied for the week / day of the week

The day of the week is defined as follows (Sunday being the first day of the week). Monday is 1, Wednesday is 3, and Sunday by 7. Using the day of the week prevents depreciated average times since at the start of the week the total time studied for that week will be 0 and by dividing by 7 initially will undervalue the average time.

Challenges Faced

The main issue faced was that the date time format that firebase uses and javascript are different. In firebase the date time stored is in UTC +8 while for javascript it does not. Therefore formatting the date times such that it will sync with firebase and javascript is important because there are date time comparisons for this feature.

Another issue faced was getting the window of data for a specific week. When getting the total time studied for the week, the range will be at 00:00:000 Sunday, till today at 23:59:999. Though javascript allows manually setting of the time. But due to the different time zones in firebase and javascript, after altering the datetime, it is off or over by 1 day. One work around was to ignore the setting of the millisecond.

Images

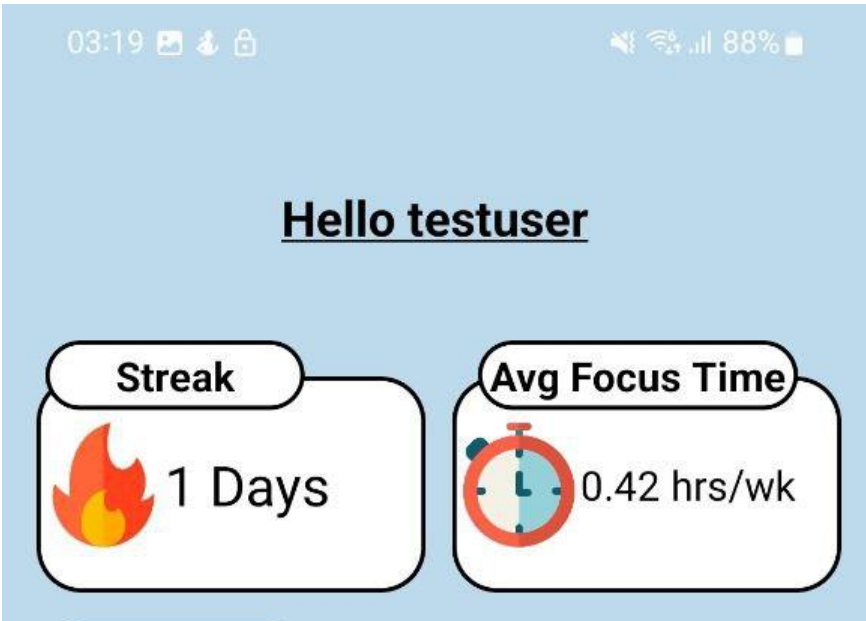


Image : Elements in the Streak Feature

Task List

[Completed]

Description

The motivation behind this feature is to allow users to be able to keep track of assignments or important deadlines with an integrated ecosystem, instead of installing other apps to keep track of things they need to do. Once authenticated, users can access their profile page to view their task list.

On the profile page and on the task list card, a user can do the following actions:

1. Add a task by clicking on the plus icon at the top right hand corner of the card
2. Browse through their tasks by swiping on the task list
3. Mark tasks as complete by clicking on the checkbox on the left of the task
4. Delete tasks by clicking on the delete button on the right of the task

Challenges Faced

Initially, there was confusion between the commands "npx expo run" and "npx expo start," which hindered the proper building of the application. It took half a day to understand the differences between these commands.

Familiarity with Firebase's structure was lacking, especially coming from a SQL background. The concepts of non-SQL documents, collections, and subcollections were new and required additional research before working on the CRUD features of the task list.

Limited knowledge of CSS posed a significant challenge, particularly in styling the task list view to match the design shown in the poster. There was also difficulty in styling the modal for creating new tasks. Additionally, the initial use of a "Pixel 8 Pro" as the emulator led to CSS issues when a team member with a smaller phone attempted recording. The use of absolute numbers in padding rather than relative measurements caused the task list to display incorrectly. This issue required changing the padding to use the device's height and width multiplied by a certain factor.

Images

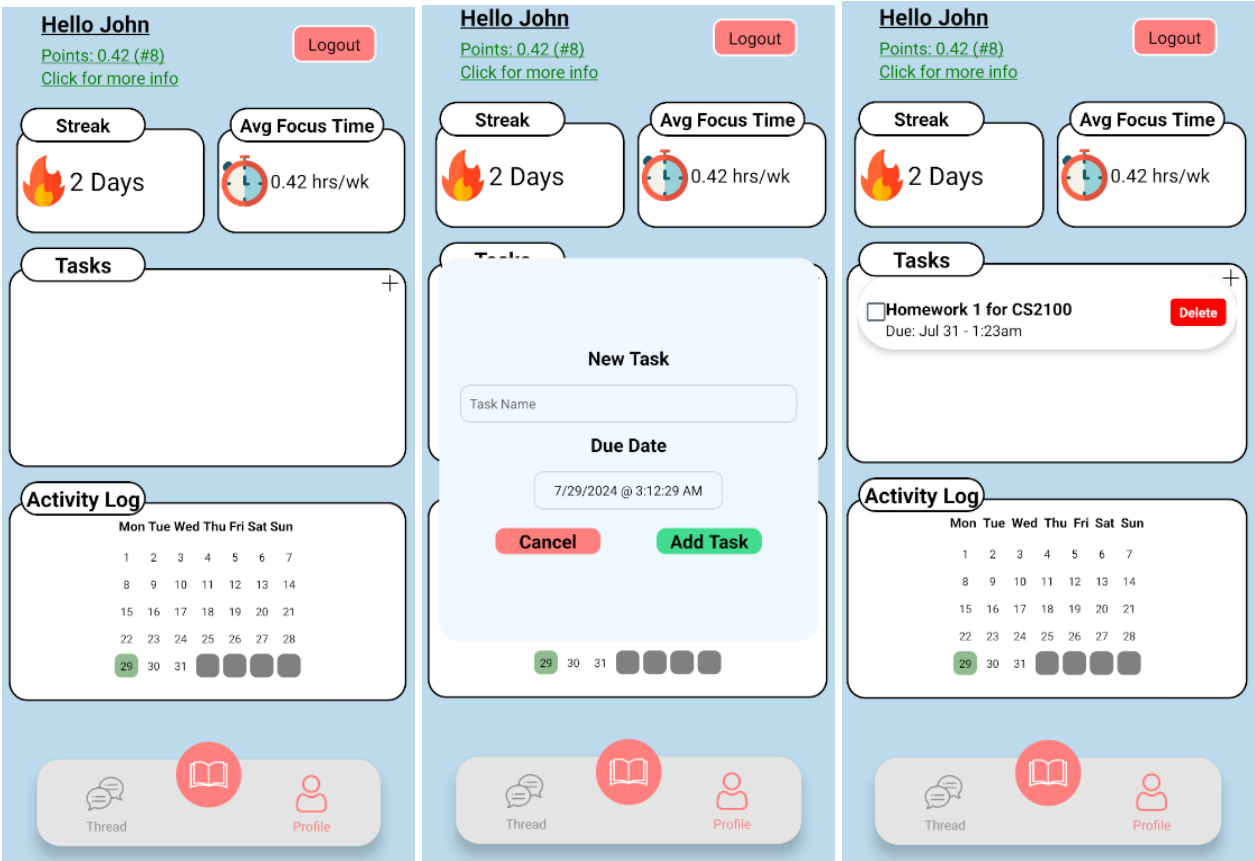


Image : Elements in the Task Feature

Heatmap

[Completed]

Description

The heatmap feature allows users to easily visualize their study activity over the current month, serving as a motivational tool by highlighting their progress. After logging in, users can view a heatmap that displays the number of hours studied each day, with darker colors representing more hours. Users can also click on any day to see the specific number of hours studied.

Challenges Faced:

The team initially lacked knowledge of how to query the Firestore database. We conducted research and examined our database structure to craft appropriate queries for retrieving the monthly study hours for each user. Additionally, coding the user interface presented challenges due to varying month lengths and the fact that months do not always start on a Monday or end on a Sunday. The backend and UI were reworked to ensure accurate data representation.

Images

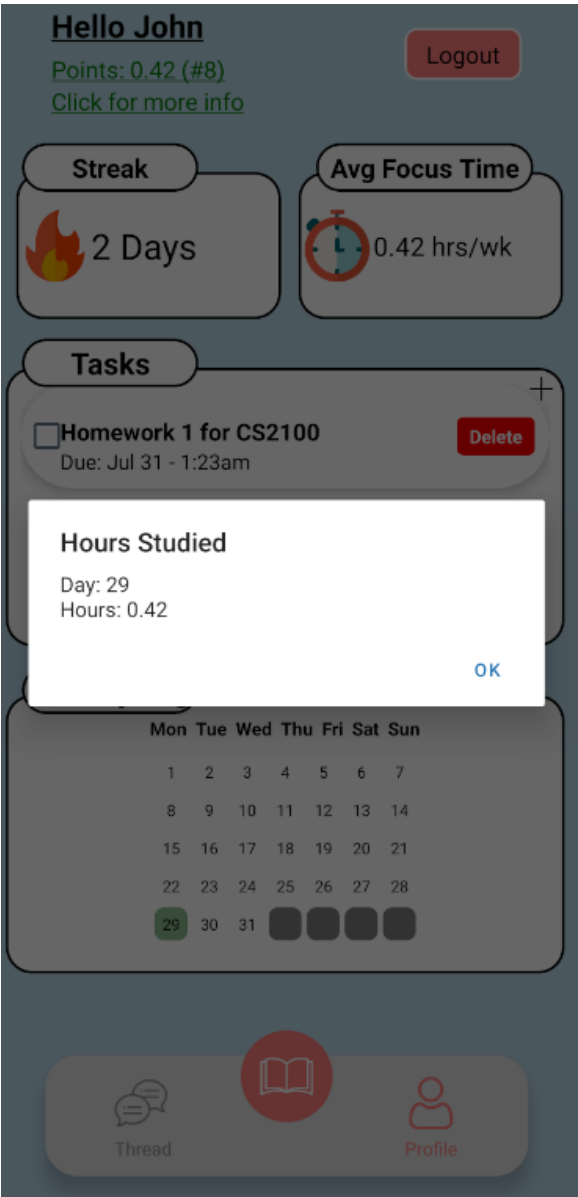
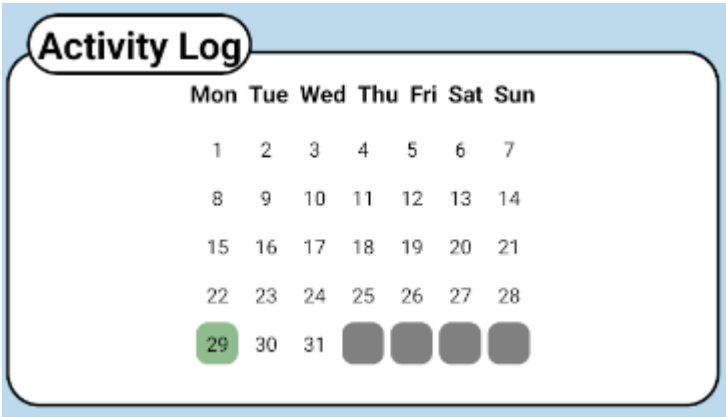


Image : Elements in the Heatmap Feature

Gamification

[Partially Completed]

Description

The gamification feature aims to foster healthy competition among users, encouraging them to study more. After logging in, users can see their rank and accumulated points. Clicking on their rank allows them to view the overall leaderboard. The point system is straightforward, with each hour studied equating to one point, providing a clear incentive for users to aim for the top rank. Initially, we planned to implement more gamification elements, but time constraints limited our progress.

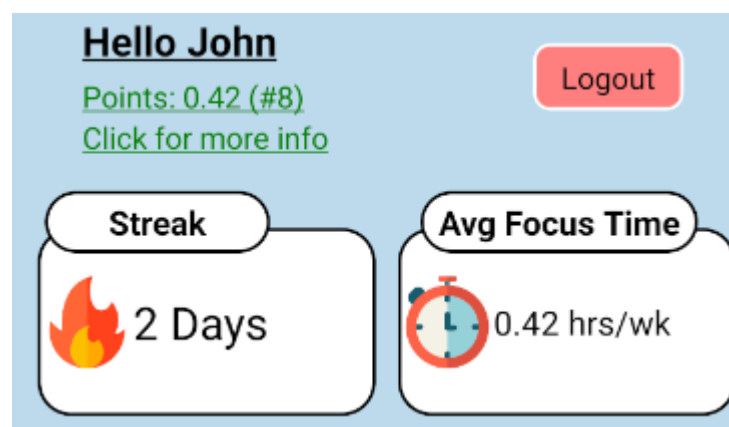
Challenges Faced

Initially, we considered calculating points by aggregating all study sessions for each user, but this approach was inefficient. We ultimately decided to create a separate database for points, similar to how streak statistics are implemented, to reduce the number of database reads while slightly increasing writes.

Handling asynchronous functions also posed challenges. The `await` keyword could not be used directly within the Profile function, as it would delay loading the entire profile until the database query completed. The team had to learn more about React props and utilized the `useEffect` hook to manage asynchronous data fetching.

Lastly, implementing routing was challenging because it required navigation to the leaderboard without having it explicitly listed in the navigation bar. We researched React routing techniques and ultimately implemented nested navigation within the existing navigation structure.

Images



4:38

←

Leaderboard (1 point = 1 hr)

1	Johan	35.67
2	Ben	14
3	Nora	3
4	Dexter	1.83
5	Sheen	1
6	Peter	0.67
7	Max	0.5
8	John (You)	0.42
9	Joel	0.17
10	Sean	0.17
11	Aaron	0.17
12	Hughie	0
13	Bob	0
14	Sally	0

Thread



Profile

Image : Elements in the Heatmap Feature

Thread

[Completed]

Description

The idea for this feature is to provide students with a simple and accessible platform to seek help from others with difficulties faced. Users can ask questions, seek information about different courses and look for group mates. Each post will be dedicated to a specific module code for ease of use.

Users can also “subscribe” to specific modules that they are taking for the semester or something that they are interested in and get a personalized feed based on their subscriptions. Users can also view posts from other modules without subscribing by searching it in the search bar and finding the module.

Users can also post and reply to threads and all this is achieved using firestore firebase and CRUD.

On the thread page, the user can do the following actions:

1. Using the search bar and find posts based on the course code
2. Subscribe to specific courses to get a personalised feed
3. Write posts to a course that the user is subscribed to
4. Reply to other user's posts
5. View their own posts to check for replies

Loading Screen

[Extension Completed]

One additional feature that was added was a loading screen, with much data needed to be queried and fetched and read, the time to render components can be very long which results in displaying an empty screen or some incorrect components before rendering the correct ones. To counteract this, a loading screen will be displayed to inform the user that the app is loading and once all the information has been gathered and processed, only then the thread page will be displayed. This was done by using Reacts use effect and use state to determine if the page can be rendered or not.

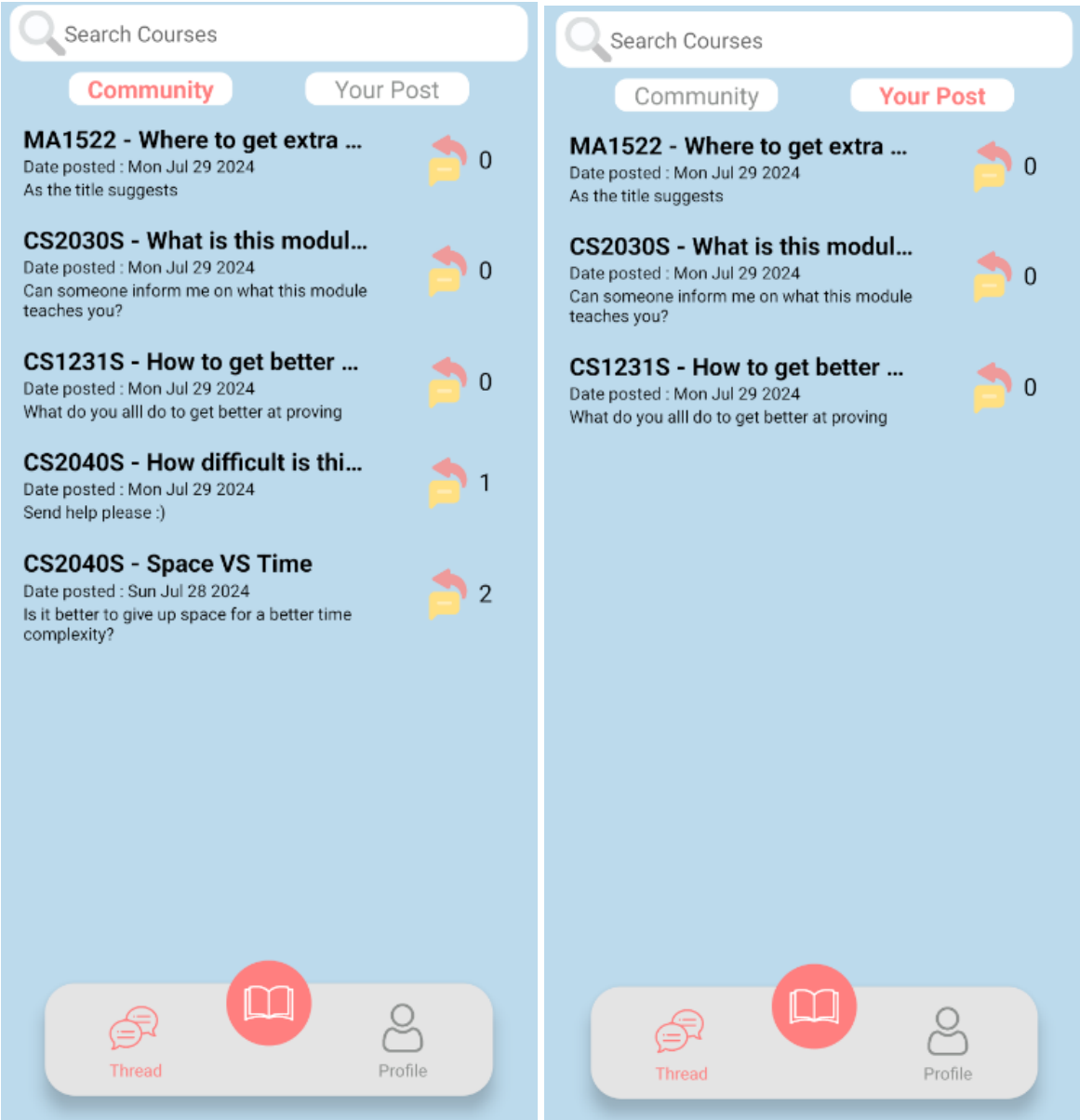
This feature was not only implemented for the thread page but for all other pages as well in the event of slow loading times.

Challenges Faced

Navigation between the different pages was an issue faced early on when implementing the feature. Since depending on the page the user was last seen, when going back the user should be directed to the last screen the user was at. This poses a problem as react natives tab navigation does not remember where the user was previously and thus a different approach for navigation was taken. A way to solve this issue was to store a state to determine which component/screen should be loaded onto the screen. Another use state was used to remember the last page the user was at.

Another challenge faced was implementing a search bar with an auto complete functionality. In essence, the idea was to display potential courses that the user might be searching for based on what they have typed. Since react native did not provide a suggestions feature for their text input, the team has to come up with our own suggestion feature for the text input. This was done using a filter function to find courses with similar abbreviations.

Images



CS

CCS5101

CCS5102

CCS5103

CCS5104

CCS5201

CCS5202

CS1231S - How to get better ...

Date posted : Mon Jul 29 2024

What do you alll do to get better at proving

0

CS2040S - How difficult is thi...

Date posted : Mon Jul 29 2024

Send help please :)

1

CS2040S - Space VS Time

Date posted : Sun Jul 28 2024

Is it ...

2

Thread

Profile

CS

CAN

VS

q w e r t y u i o p

a s d f g h j k l

z x c v b n m

?123 , : ; ' " < > . ,

←

Unsubscribe

MA1522 - Linear Algebra for Computing

This course is targeted at students from School of Computing. It introduces basic concepts in linear algebra that are routinely applied in computer science, in the context of the Euclidean spaces, and will develop basic skills in computing with vectors and matrices with and without using computational software (MATLAB). Students are expected to acquire computational facilities and geometric intuition about vectors and matrices. Major topics: Systems of linear equations, matrices, determinants, Euclidean spaces, linear combinations and linear span, subspaces, linear independence, bases and dimension, rank of a matrix, inner products, eigenvalues and eigenvectors, diagonalisation, linear transformations between Euclidean spaces, applications.

MA1522 - Where to get extra ...

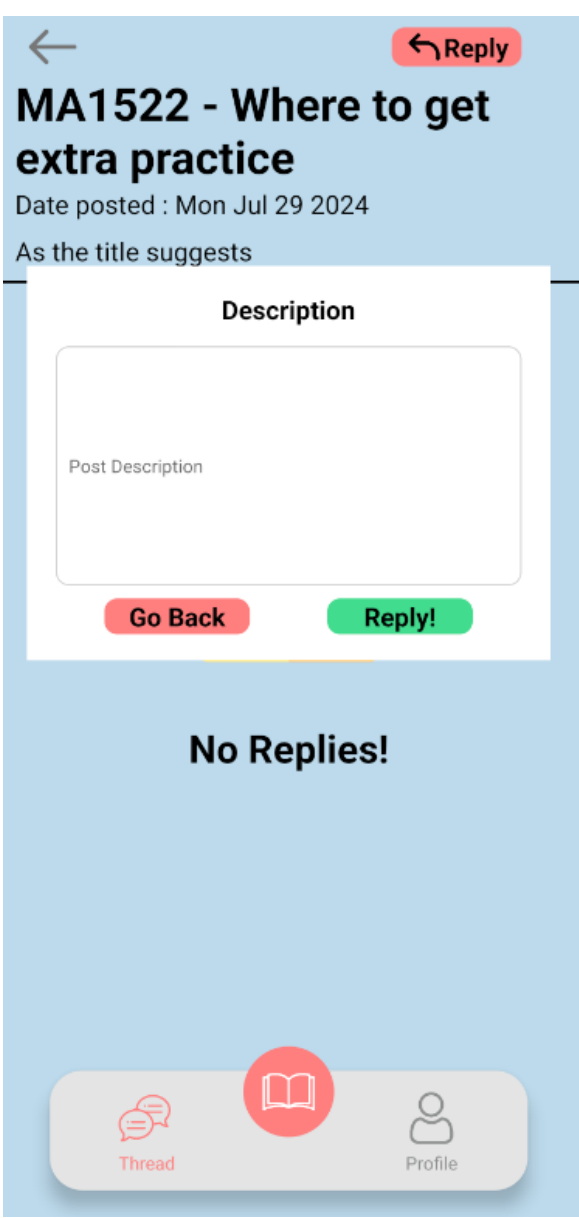
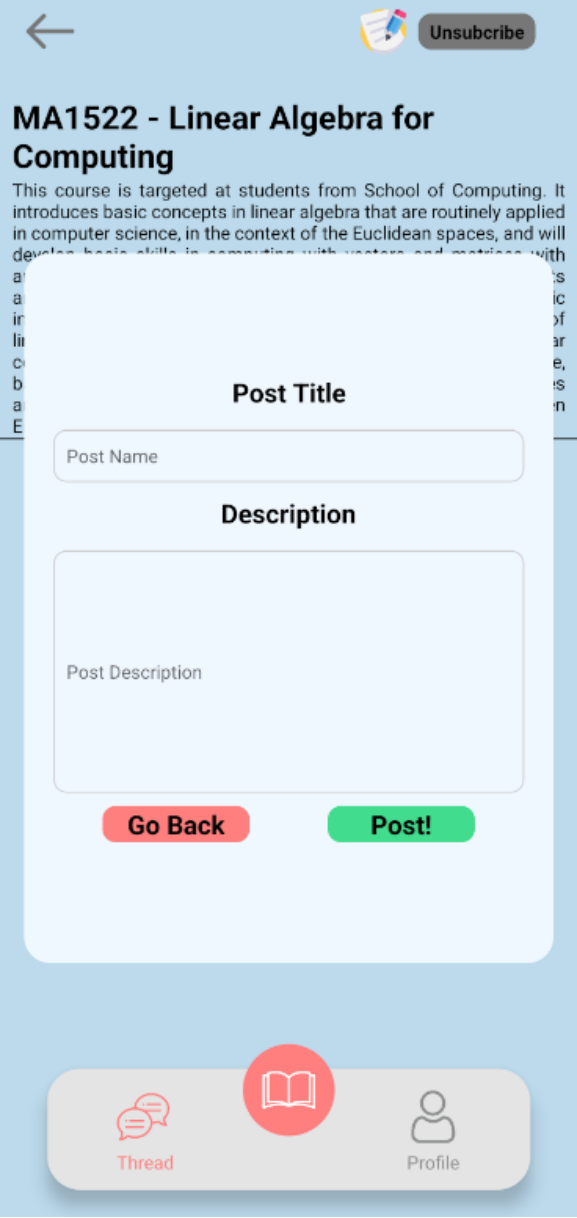
Date posted : Mon Jul 29 2024

As the title suggests

0

Thread

Profile





Date posted : Mon Jul 29 2024



Thread



Profile



Date posted : Sun Jul 28 2024

Date posted : Mon Jul 29 2024

Date posted : Mon Jul 29 2024



Thread



Profile

Subscribe

ACC1701X - Accounting for Decision Makers

The course provides an introduction to accounting from a user perspective. Financial reporting is covered from the viewpoint of an external investor. The focus is on how accounting can help investors make better decisions. Book-keeping and preparation of financial statements are also covered at an introductory level, as investors need to be aware of how the financial statements are derived.

So quiet here post something to start something!

Thread

Profile

Search Courses

Community

Your Post

Start following courses and see them appear here!

Thread

Profile

Search Courses

Community

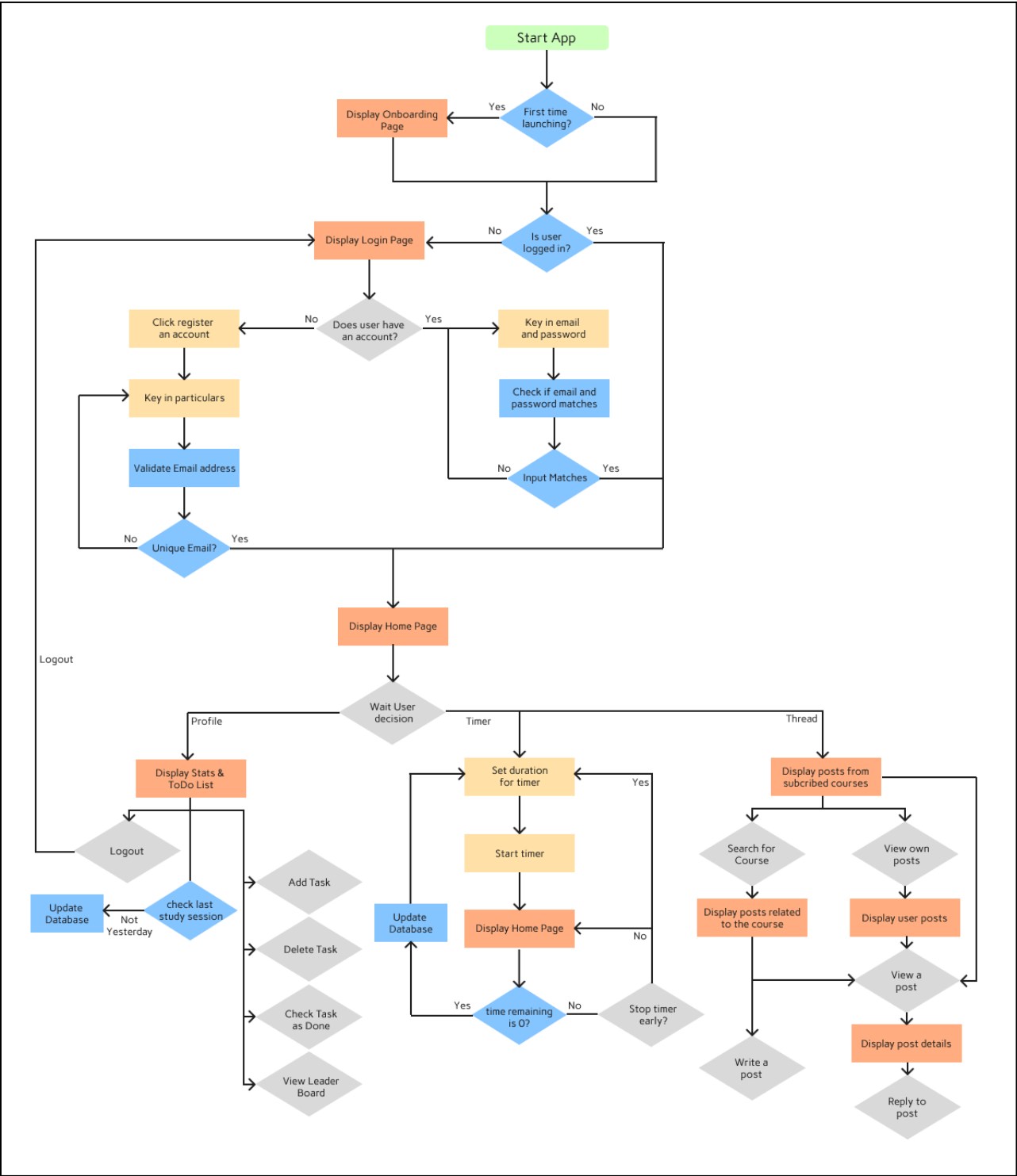
Your Post

Post something and see them appear here!

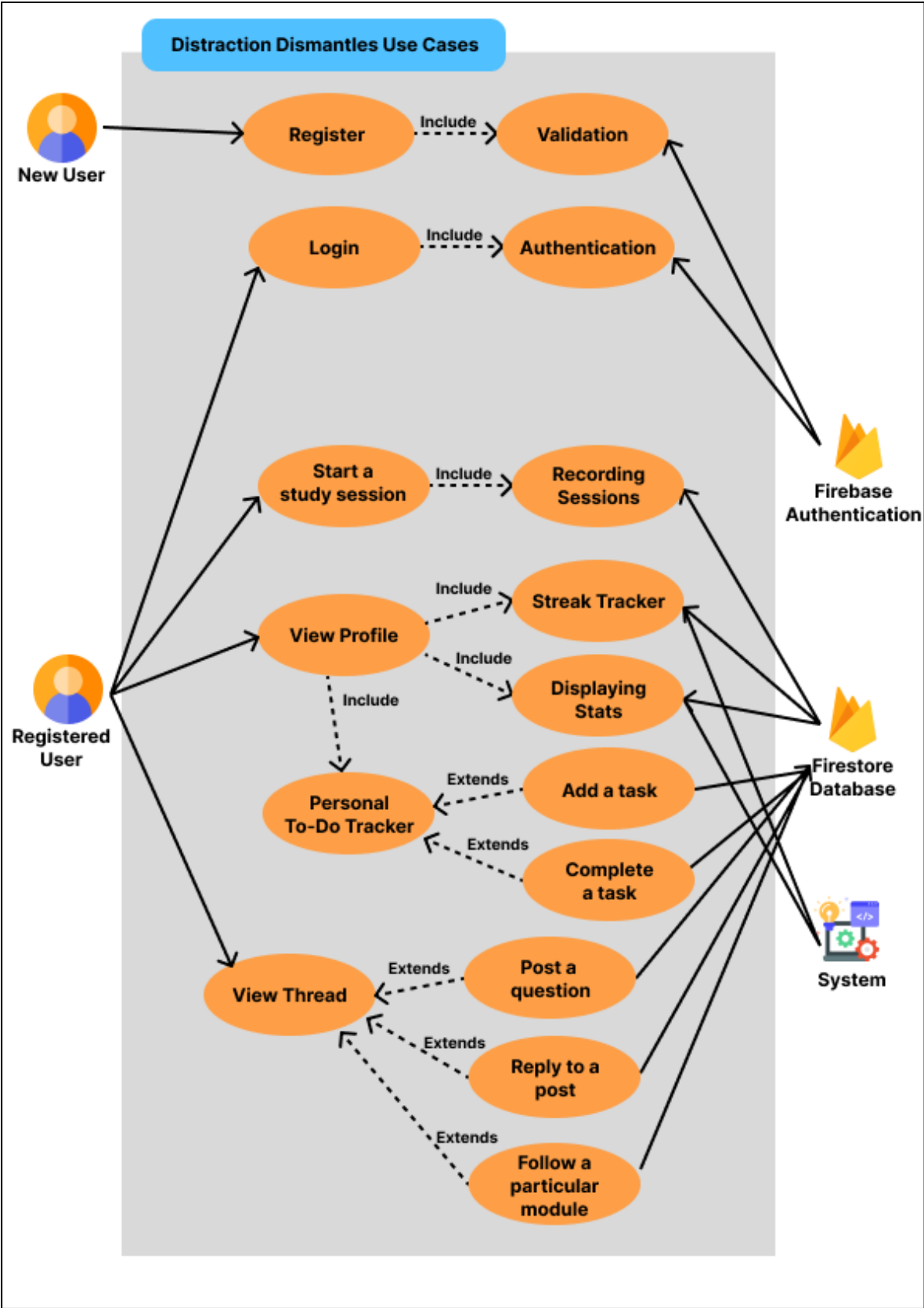
Thread

Profile

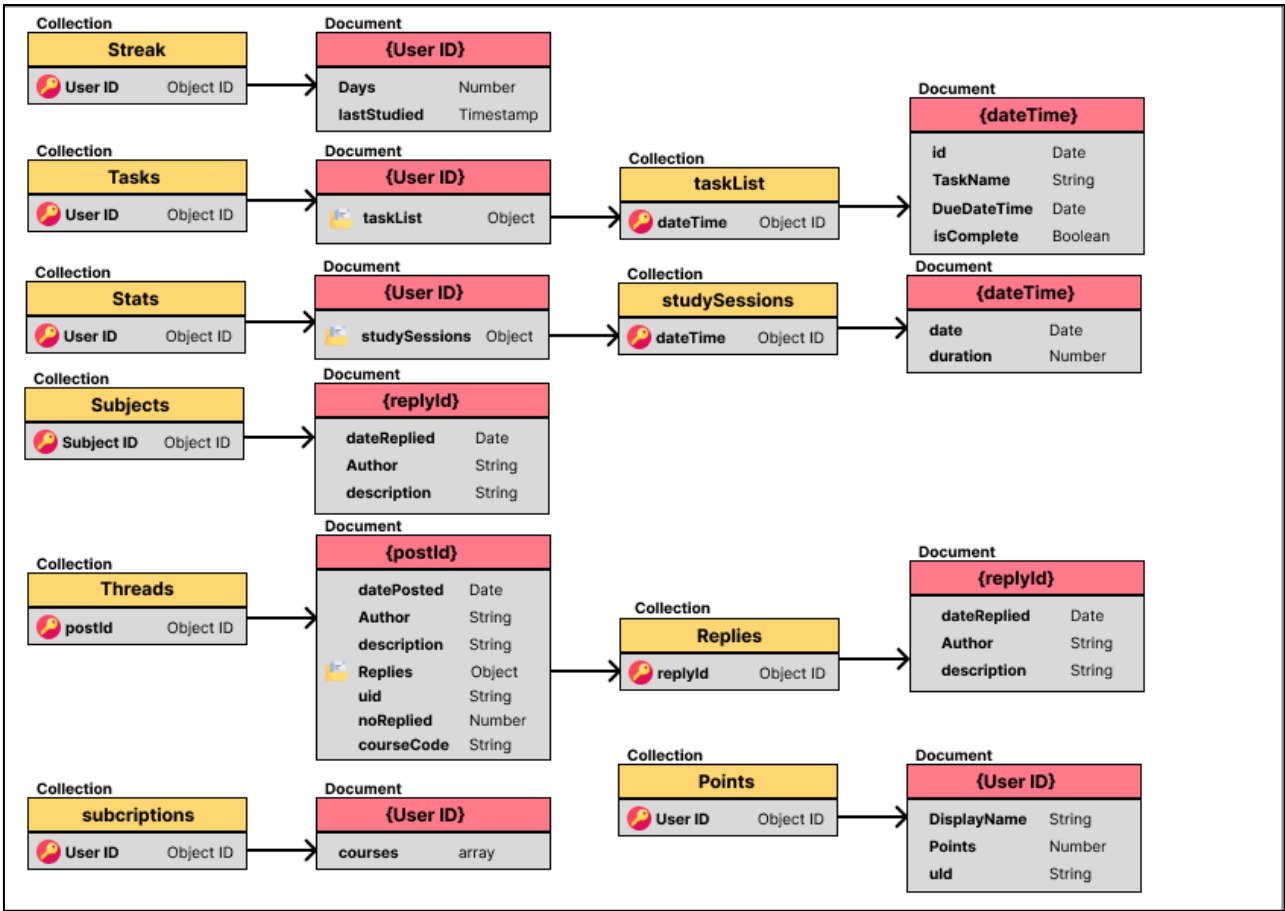
Activity Flow Diagram



Use Case Diagram

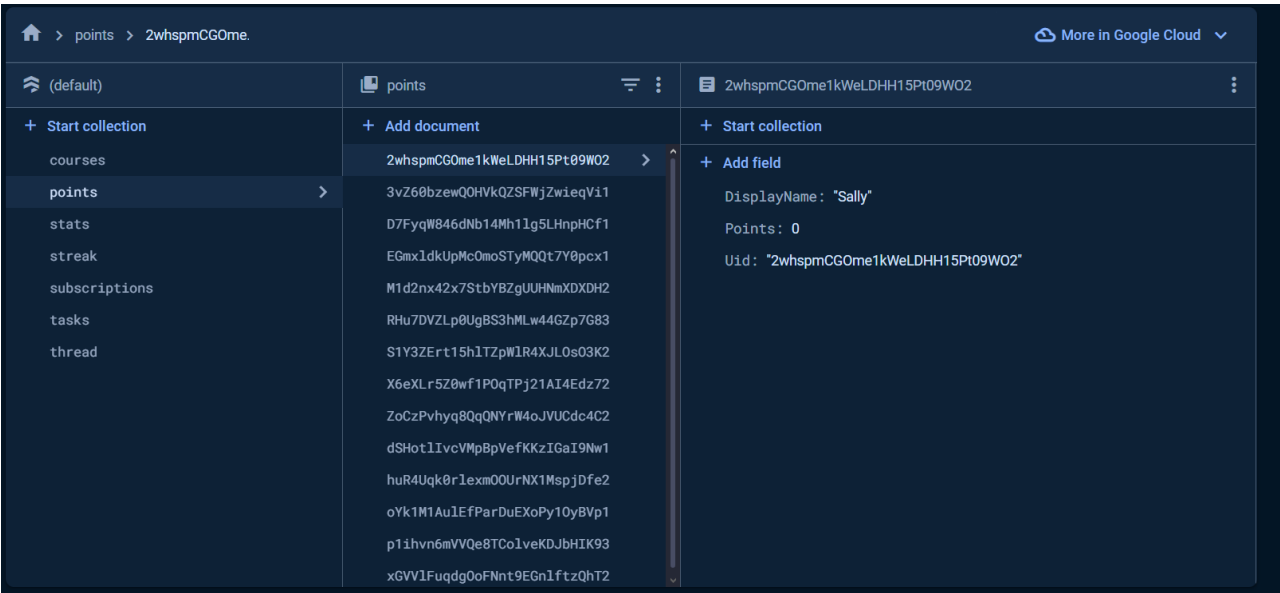


Database Model Diagram



Firestore

Here is a image of how our database looks like in firestore



Database Rules

Here are the rules that were written to manage CRUD requests to the database. Essentially, users are not able to read, write or update other people's data and to do that, the document Id will match the user's Id.

Jul 27, 2024 • 1:58 AM

Jul 27, 2024 • 1:39 AM

Jul 23, 2024 • 1:53 PM

Jul 23, 2024 • 1:52 PM

Jul 23, 2024 • 1:52 PM

Jul 22, 2024 • 5:42 PM

Jul 22, 2024 • 5:34 PM

Jul 22, 2024 • 5:34 PM

Rules Playground

Experiment and explore with Security Rules

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {

    match /stats/{userID}/{document=**} {
      allow read, write: if request.auth != null && request.auth.uid == userID;
    }

    match /streak/{userID} {
      allow read, write: if request.auth != null && request.auth.uid == userID;
    }

    match /subscriptions/{userID} {
      allow read, write: if request.auth != null && request.auth.uid == userID;
    }

    match /tasks/{userID}/{document=**} {
      allow read, write: if request.auth != null;
    }

    match /thread/{userID}/{document=**} {
      allow read, write: if request.auth != null;
    }

    match /courses/{courseID} {
      allow read: if request.auth != null;
      allow write, delete: if request.auth != null && request.auth.uid == "Admin";
    }

    match /points/{Uid} {
      allow read: if request.auth != null;
      allow write: if request.auth != null && request.auth.uid == Uid;
    }
  }
}
```

Development Plan

Tasks	Description	In Charge	Date
Prepare Liftoff Poster and Video <i>Completed</i>	Create Liftoff Poster and slides, record presentation pitch.	Nicholas Sheen	13 - 20 May
Pick up necessary skills <i>Completed</i>	Pick up necessary skills of: - React Native Main framework that is used to build the application - Firebase Hosting, database and authentication method	Nicholas Sheen	21 May - 1 June
Create the database & link to the application <i>Completed</i>	Create a firebase firestore database and link it to react native application	Nicholas	21 - 24 May
Prototype Creation <i>Completed</i>	Implement a UI design and finalise the features and pages	Nicholas	22 - 24 May
Implement Authentication Method <i>Completed</i>	Integrate Firebase authentication to the React Native application.	Sheen	24 May - 2 June
Implement Timer <i>Completed</i>	Implement Timer Feature	Nicholas	
Milestone 1:			3 June
Redesign Navigation Bar <i>Completed</i>	Work on prototype design of navigation bar	Nicholas	3- 12 June
	Learnt how navigation works in react		
	Restyle current Navigation Bar		
Add Logo to app	Redesign Logo	Sheen	12 - 30 June
	Add Logo to app		
Fully Implement Timer Feature <i>Completed</i>	Finish the timer feature such that users can save their timers and the amount of time they have studied into firebase	Nicholas	13 - 26 June
Fully Implement Streak Feature <i>Completed</i>	Finish Streak Feature such that users can see their completed streaks of studying per week	Nicholas	13 - 26 June
Fully Implement Profile	On this page the user can see their	Nicholas	13 - 26 June

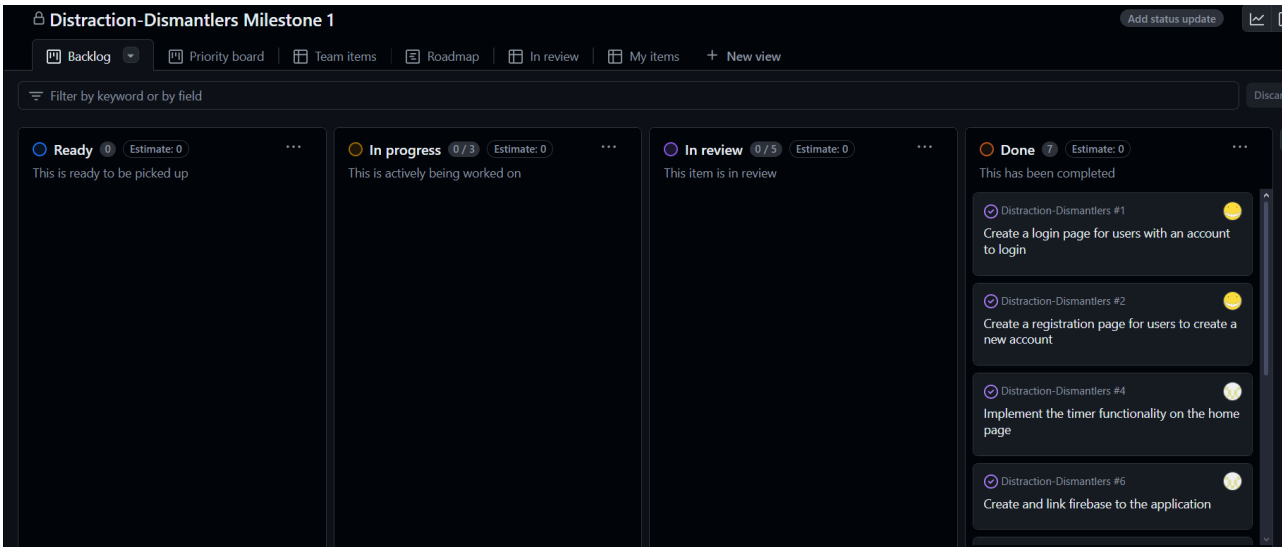
Feature <i>Completed</i>	streak as well as an indicator of the average weekly hours studied based on completed study sessions		
Implement CRUD for Task List Feature <i>Completed</i>	Learn how does a Non-SQL database work	Sheen	16- 30 June
	Design the database to store the task list feature		
	Implement a working logical control flow for Task Page, ensuring user data can be written to and retrieved from Firestore		
Testing and Debugging <i>Partially Completed</i>	Provide basic testing in preparation (Timer) <i>Completed</i>	Nicholas	25 - 30 June
	Provide basic testing in preparation (Task) <i>Incomplete (Test Cases coded but not working with firebase)</i>	Sheen	
Output App to APK <i>Completed</i>	Ensure app logo is updated and output as an android APK	Sheen	30 June
Milestone 2: Working Minimum Viable Product			1 July
Implement Email Verification <i>Completed</i>	Implement email verification for Sign Up page	Sheen	1 - 3 July
Implement Motivational Reminders <i>Incomplete due to lack of time</i>	Implement firebase notification reminders to allow users to add reminders and for the app to periodically remind the user to study	Sheen	3 - 7 July
Implement Gamified Focus Mode <i>Partially Completed</i>	Implement gamification elements to the focus mode, such as points, badges, leaderboards, and customizable themes or avatars.	Sheen	7 - 15 July
Implement Thread Feature (CRUD) <i>Completed</i>	Implement Thread Page to allow users to CRUD posts about modules as well as to navigate around the thread page	Nicholas	15 - 20 July
	Implement the search bar to search for the respective module pages as well as a auto complete feature based on NUS offered modules	Sheen	

Implement Heatmap Feature to Profile Page <i>Completed</i>	Add a heatmap of 30 days to which gets darker the longer hours a user has studied for that day	Sheen	
Further testing and debugging <i>Completed</i>	Perform in-depth unit, integration and system tests for the MVP	Nicholas Sheen	20 - 28 July
Milestone 3: Extension Phase			29 July
Fix any issues or bugs	Allocate time to fix any issues that arise during the deployment of the MVP	Nicholas Sheen	29 July - 28 August
Splashdown			28 August

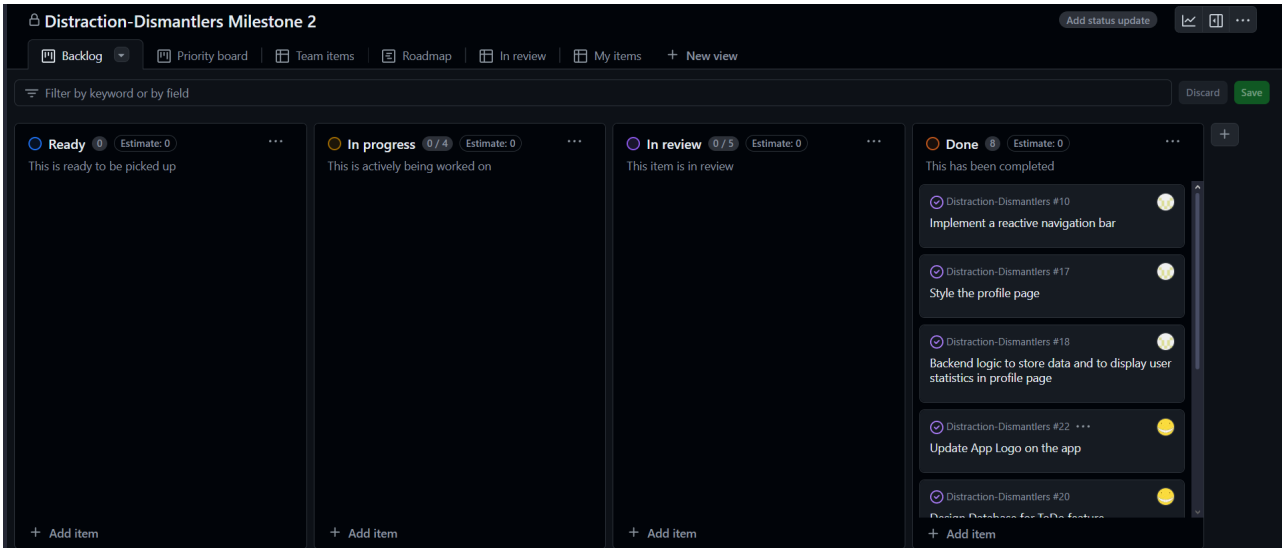
Github Usage

Scrum Board

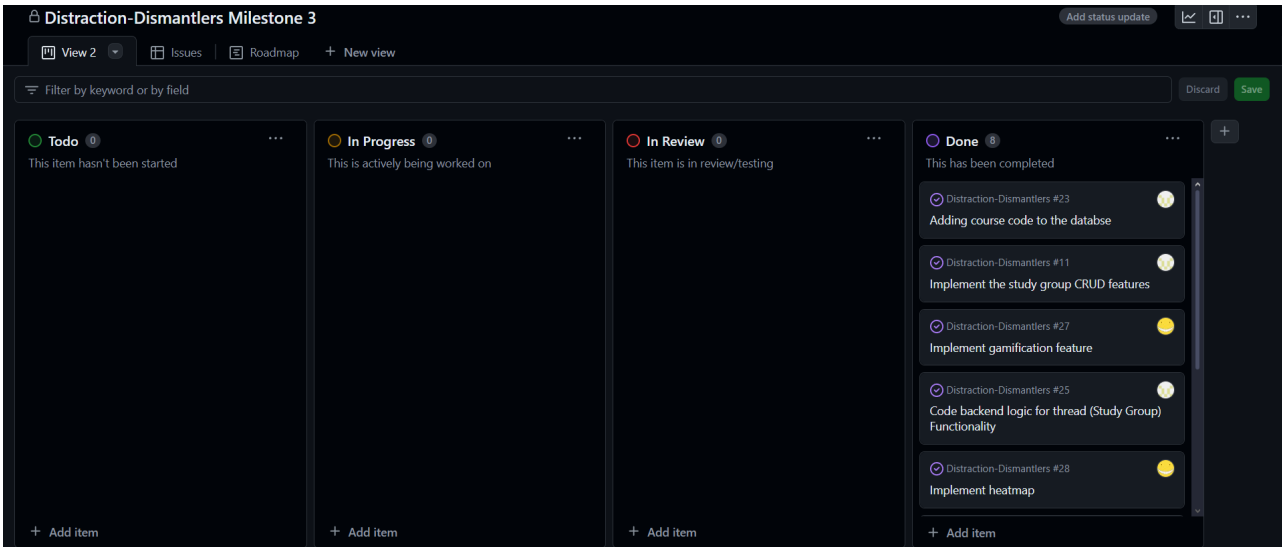
Milestone 1



Milestone 2



Milestone 3

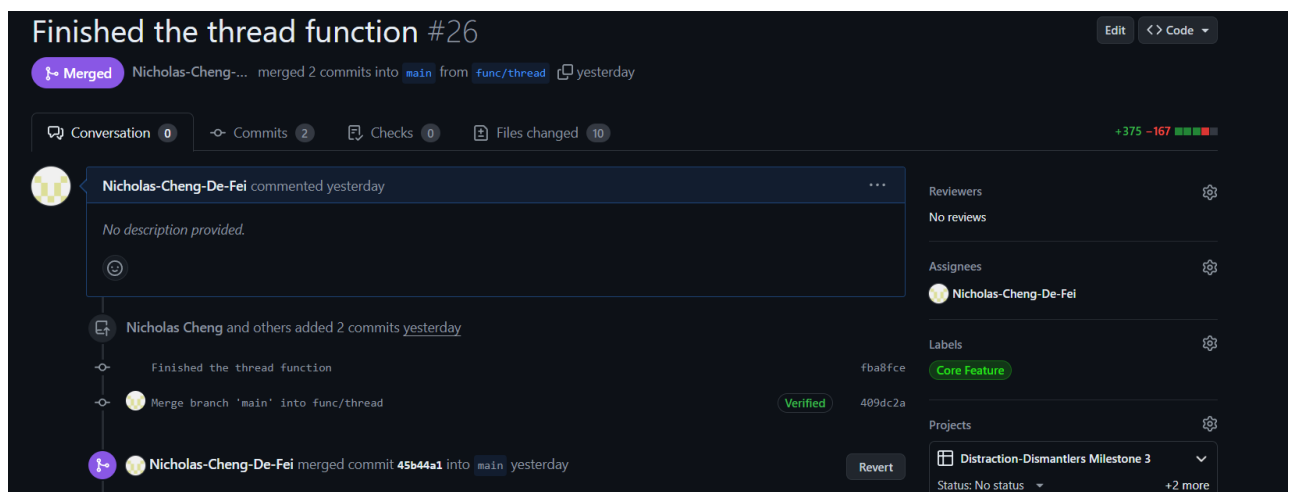


Version Control

Default					
Branch	Updated	Check status	Behind	Ahead	Pull request
main	26 minutes ago				Default
Your branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
func/thread	yesterday		8	0	#26
feature/navBar	last month		36	1	
feature/Timer	last month		39	0	#13
feature/onboarding-page	2 months ago		47	1	
Active branches					
Branch	Updated	Check status	Behind	Ahead	Pull request
func/thread	yesterday		8	0	#26
feature/navBar	last month		36	1	
feature/Timer	last month		39	0	#13
feature/onboarding-page	2 months ago		47	1	

For version control of the code base, the team used github to facilitate and control the different stages/versions of the code throughout the development process. The main branch or the master branch will contain a deployable working code base at any point in time. When working on a new feature, the team will pull from main and work the function in separate branches and once the function is completed and tested a pull request will be made to merge the code version in the branch to the main branch. Any conflicts will be manually handled by the team or by github if possible.

Pull Requests



Only when a team member has finished a function and has tested it, can they create a pull request which the other user will test before accepting the pull request and merging into the main function. Before the merging is done any code conflicts must be settled before a successful merge to the main branch.

Quality Control

For quality control, the team tried and use JEST which was a react native testing library which can mimic human interaction through various elements in our mobile application, the team tried to implement certain interactions using JEST for our test cases, however after multiple attempts it does not work and due to time constraints, the team resorted to doing manual testing for the application.

However in the early phases of testing, the team did attempt to create some test cases using JEST for automated testing.

Testing the Timer Feature

Test cases have been written using JEST to validate the functionality of the timer. Tests include verifying the creation of arrays for possible hours and minutes, as well as ensuring correct rendering of the timer interface.

```
PASS components/__tests__/timer.test.tsx (7.19 s)
  Timer external function
    ✓ Check if the function creates the array of possible hours correctly given a limit (2 ms)
    ✓ Check if the function creates the array of possible minutes correctly given a limit (5 ms)
  Timer rendering tests
    ✓ Check if timer renders correctly (2835 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:  0 total
Time:        7.536 s
Ran all test suites.
```

Challenges Faced:

Some problems faced were that certain functions to mimic user interaction with the application was not working as intended when compared to using the application on the emulator as such some test cases ended up failing because of that. In addition JEST does not have in-built functionality for firebase and as such to counteract this issue, is to simply fetch the same document back and check if what was written is correct.

Testing the Task List Feature

Test cases were written using JEST to verify the CRUD functions of the Task List. However, due to numerous errors and time constraints, the test cases were not completed for this milestone.

```
Test Suites: 1 failed, 1 total
Tests:       4 failed, 4 total
Snapshots:  0 total
Time:        0.785 s, estimated 1 s
Ran all test suites matching /tasks/i.
PS C:\Users\Admin\Desktop\Codes\Distraction-Dismantlers>
```

Challenges Faced:

Initially, there was confusion between the commands “npx expo run” and “npx expo start,” which hindered the proper building of the application. It took half a day to understand the differences between these commands.

Familiarity with Firebase's structure was lacking, especially coming from a SQL background. The concepts of non-SQL documents, collections, and subcollections were new and required additional research before working on the CRUD features of the task list.

Limited knowledge of CSS posed a significant challenge, particularly in styling the task list view to match the design shown in the poster. There was also difficulty in styling the modal for creating new tasks. Additionally, the initial use of a “Pixel 8 Pro” as the emulator led to CSS issues when a team member with a smaller phone attempted recording. The use of absolute numbers in padding rather than relative measurements caused the task list to display incorrectly. This issue required changing the padding to use the device’s height and width multiplied by a certain factor.

Unit Test					Result	
Test ID	User Story	Testing Objective	Steps to Take	Expected Outcome	Result	Test Date
1	I am a new user and I will like to register for an account	Check that the onboarding page shows for a new user	1. Fresh install the application 2. Launch the application 3. Go through the onboarding process 4. Restart the application	The onboarding screen will show only once and and in the future it will not show and instead it will direct to the login page	Pass	24/05/2024
		Check that registering a new account properly the user will be brought to the home page	1. Key in all the particulars needed to create an account 2. Press the register button	The user details will be saved on the database and will be redirected to the home page	Pass	02/06/2024
		Check that error message shows when user did not fill up all the inputs	1. Leave out 1 of the input fields 2. Press the register button	An error message will pop up asking the user to fill up all of the fields	Pass	02/06/2024

		Check that error message shows when user did not key in the same password	1. Key in a different password when confirming 2. Press the register button	An error message will pop up telling the user that the password does not match	Pass	02/06/2024
		Check that error message shows when password is too short	1. Key in a password that has length of less than 5 2. Press the register button	An error message will pop up asking the user to use a longer password	Pass	02/06/2024
		Check that error message shows when too many request is sent to firebase authentication	1. Fill up all the field in the register page 2. Click the register button many times	An error message will pop telling the user that too many requests has been sent to firebase authenticator	Pass	02/06/2024
		Check that error message shows when the user did not validate their email address	1. Register an account 2. Logout of that account under profile page 3. Login to the same account	A pop up should show that the user has not validated their email address yet	Pass	24/07/2024
2	I am an existing user and I want to login to my account	Check that if the user keys in the correct email and password they can login	1. Key in a valid email and password 2. Press the login button	The user should be able to login and be redirected to the home page	Pass	02/06/2024
		Check that error message shows when user did not fill up all the inputs	1. Leave out 1 of the input fields 2. Press the login button	An error message will pop up asking the user to fill up all of the fields	Pass	02/06/2024
		Check that error message shows when user key in an in correct password or email	1. Key in the correct email address but a wrong password 2. Press the login button	An error message will pop up saying the email does not exist or the password is wrong	Pass	02/06/2024

3	As a user I want to start a study session	Check that the timer gets the time set by the user and starts counting down	1. Scroll the 2 time pickers to set the timer 2. Press the start timer button	A timer should display and starts from the time set by the user	Pass	02/06/2024
		Check that the timer goes down by 1 second and displays the remaining time	1. Start the timer for 5 minutes 2. Observe that after a set amount of time	The timer should display the timer remaining after the designated time has lapsed	Pass	02/06/2024
		Check that once the timer ends the user will be brought back to the timer selection page	1. Start the timer for 5 minutes 2. Wait for 5 minutes	Once the timer ends it will bring the user back to the time selection page	Pass	02/06/2024
		Check that if the study session is stopped it will stop the timer and redirect to the timer selection page	1. Start the timer for 5 minutes 2. Press the stop timer button once the timer starts	Upon pressing the stop timer button the timer ends and it will bring the user back to the time selection page	Pass	02/06/2024
4	As a user I want to check my activity	Check if the streak counter increments after a successful study session	1. Create a fresh account as their streak count will be 0 2. Start a study session for 5 minutes 3. After 5 minutes proceed to the profile page	The streak counter will show 1 instead of 0 as the user has not completed 1 study session for the day	Pass	26/06/2024
		Check if the streak counter does not increment after a successful study session	1. Create a fresh account as their streak count will be 0 2. Start a study session for 5 minutes 3. Stop the study session	The streak will still remain 0 since the user did not complete a study session	Pass	26/06/2024

			4. Proceed to the profile page			
		Check if the user misses 1 day the streak will reset to 0	1. Create a fresh account and manually set their streak counter to be 1 and the date for the last study session to be 2 days ago on firebase 2. Proceed to the profile page	The streak card will show 0 instead of 1 as the user did not complete 1 study session yesterday	Pass	26/06/2024
		Check if average hours studied for the week is calculated correctly	1. Create a fresh account as their study hours will be 0 for that particular week 2. Manually set a study session for 32400 (In seconds) 3. Head to the profile session	Since on the day of testing it is a wednesday, the average hours studied should show 3 hrs (324000/60/60/3)	Pass	26/06/2024
		Check if the heatmap updates correctly after a successful study session	1. Create a fresh account as their study hours will be 0 for that particular week 2. Manually set a study session for 32400 (In seconds) 3. Head to the profile session	The heatmap, the box with for today's date will be a different color and upon clicking on the box it will show how long they study for the day	Pass	20/07/2024
5	As a user I want to log what I want to do and important deadlines	Check if the display for the task list is correct	1. Create a fresh account as they will have no tasks in their list 2. Manually input 1 task and name the task as "Sample Task" with the due date to be	In the task card the task "Sample Task" should be displayed with the due date being today	Pass	26/06/2024

			today 3. Proceed to the profile page			
		Check if the user can add a task	1. Create a fresh account as they will have no tasks in their list 2. Proceed to the profile page 3. Press the plus icon 4. Key in a title and due date for the task 5. Click on the add task button	The task with the keyed in title and due date should be displayed in the task list	Pass	26/06/2022
		Check if the user can delete a task	1. Create a fresh account as they will have no tasks in their list 2. Create a new task 3. Press the delete button beside the task in the task list	The task should be removed and not appear anymore on the task list	Pass	26/06/2022
		Check if the user can mark the task as done	1. Create a fresh account as they will have no tasks in their list 2. Create a new task 3. Press the check box beside the task in the task list	The task should be blurred and strike off indicating it has been done	Pass	26/06/2022
5	As a user I want to ask questions on the thread based on the modules I am planning to	Check if the user can search for the module they are looking for	1. Login to an existing account 2. Navigate to the thread page 3. Search for a	The search bar should show all possible courses based on what the user has entered into the text input box.	Pass	28/07/2022

	take next semester		module called "CS2040"			
		Check if the user can subscribe to the module	1. Login to an existing account 2. Navigate to the thread page 3. Search for a module called "CS2040S" 4. Click onto the option 5. Press the subscribe button	Upon clicking into a module sub the top left will show a button and it will say subscribe if the user has not subscribe yet Once clicked the user is subscribed and the button will now say unsubscribe	Pass	28/07/2022
		Check if the user can write a post for a module that they follow	1. Login to an existing account 2. Navigate to the thread page 3. Search for a module called "CS2040S" 4. Click onto the option 5. Press the write icon 6. Key in the relevant information	A popup will appear after pressing the write button, if the user has subscribed to the module Once the post has been submitted it will be displayed on screen	Pass	28/07/2022
		Check if the user can view other people posts	1. Navigate to the thread page 2. Click on the description of a post	It should show the full post on screen and all the replies to the post	Pass	28/07/2022
		Check if the user can reply to a post	1. Navigate to the thread page 2. Click on the description of a post 3. Press the reply button 4. Key in the relevant	Once the reply button is pressed a pop up will appear to write the reply message. After clicking on the reply button it will be displayed in the post	Pass	28/07/2022

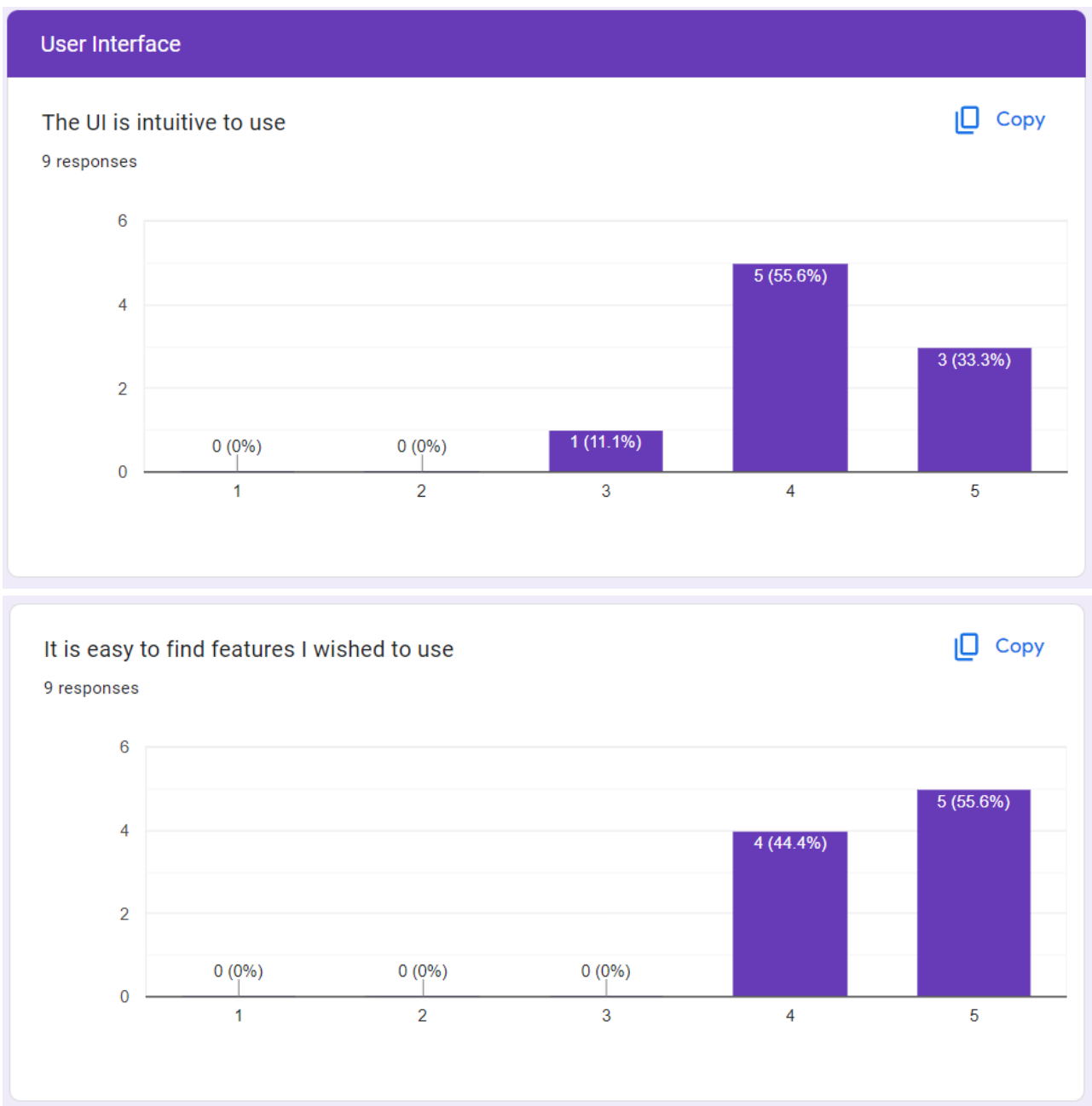
			information			
		Check if the user can see all the posts they have submitted	1. Login to an existing account 2. Navigate to the thread page 3. Press on the your post button at the top	The list should only contain the posts the user has submitted	Pass	28/07/2022
		Check if the user can see all posts based on the module they have subscribed for	1. Login to an existing account 2. Navigate to the thread page	The list should show all the posts depending on the module the user is subscribed to	Pass	28/07/2022
		Check if the go back button bring the user back to the page they were last seen at	1. Login to an existing account 2. Go search for a course 3. Tap on the description of any post 4. Click the back button at the top left	The user should be brought back the the page where it will show all posts based on the course the user searched	Pass	28/07/2022

User Testing

For user testing, the application was sent to people with similar characteristics as our target audience which are students. The participants will receive our application and are told the different features and use it. Once the user has finished using our application they will finish a google form where each question will ask about different aspects of each feature in terms of [functionality](#), [ease of use](#), [simplicity UI](#) and [any bugs encountered](#) as well.

Here are the feedback from the participants:

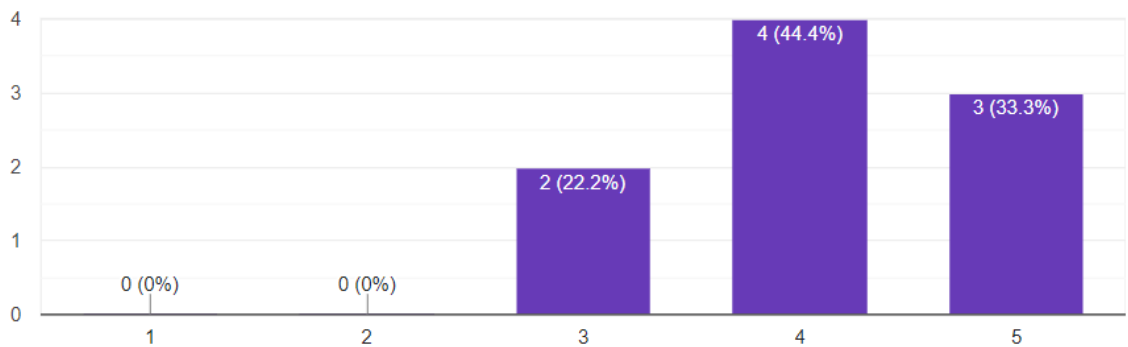
User Interface



The UI is intuitive and well designed

 Copy

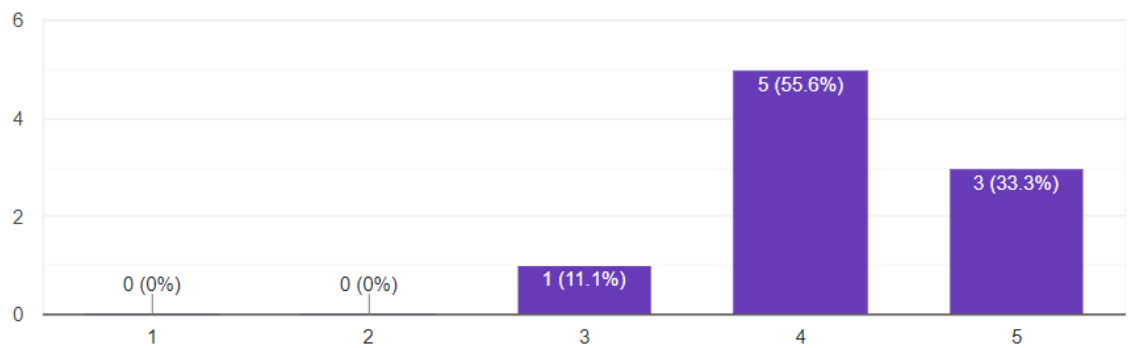
9 responses



I am satisfied with the interface

 Copy

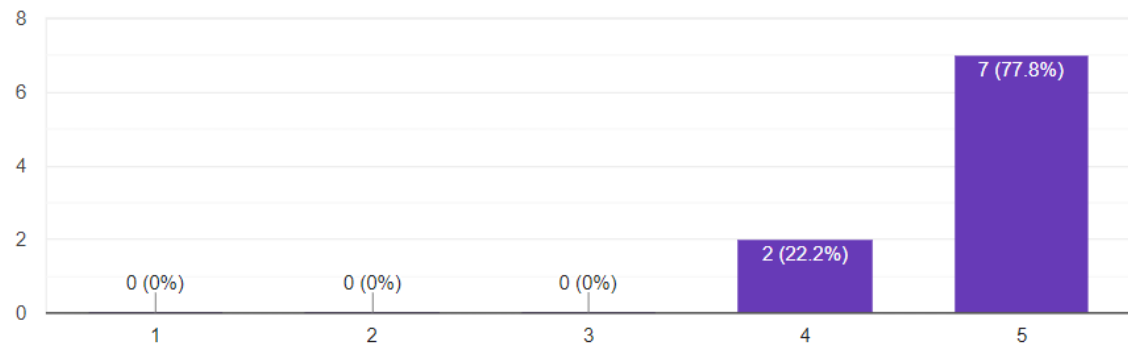
9 responses



I am can easily find what I am looking for

 Copy

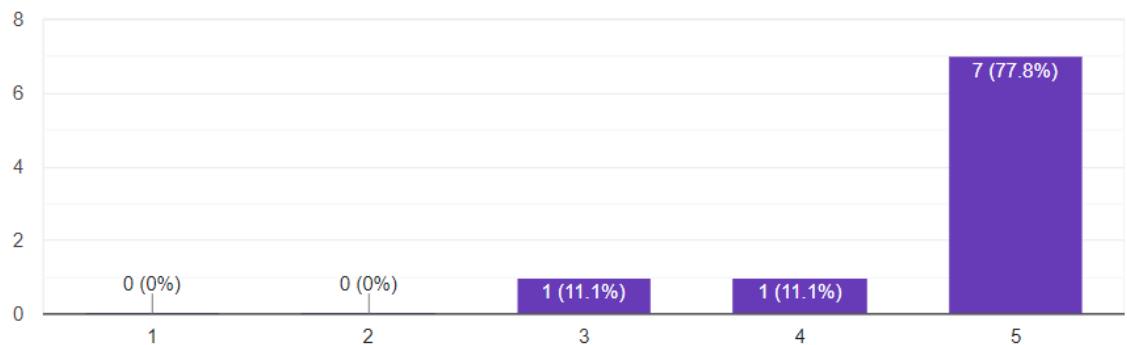
9 responses



Any errors I encountered were clearly communicated to me



9 responses



Any additional feedback on the UI?

7 responses

- Calendar could be better done with the month and proper days.
- The spacing of the tasks between the checkbox and the words
- The oval box for the tasks could be change to squares
- I feel the "write post" for the thread should be in words rather than icon
- Logout can be icon instead of words
- Better to distinguish Hello <name> with different colors/font etc.
- Post under the courses can better distinguished with proper boxes
- Calendar could have the month stated and the current day highlighted or circled.

Search bar in the thread page can be hard to access
The heatmap can remove the grey boxes as it serves no purpose

With the search bar the recommendations box height will be misaligned when the number of options decreases

Having a loading screen was good for slow rendering pages

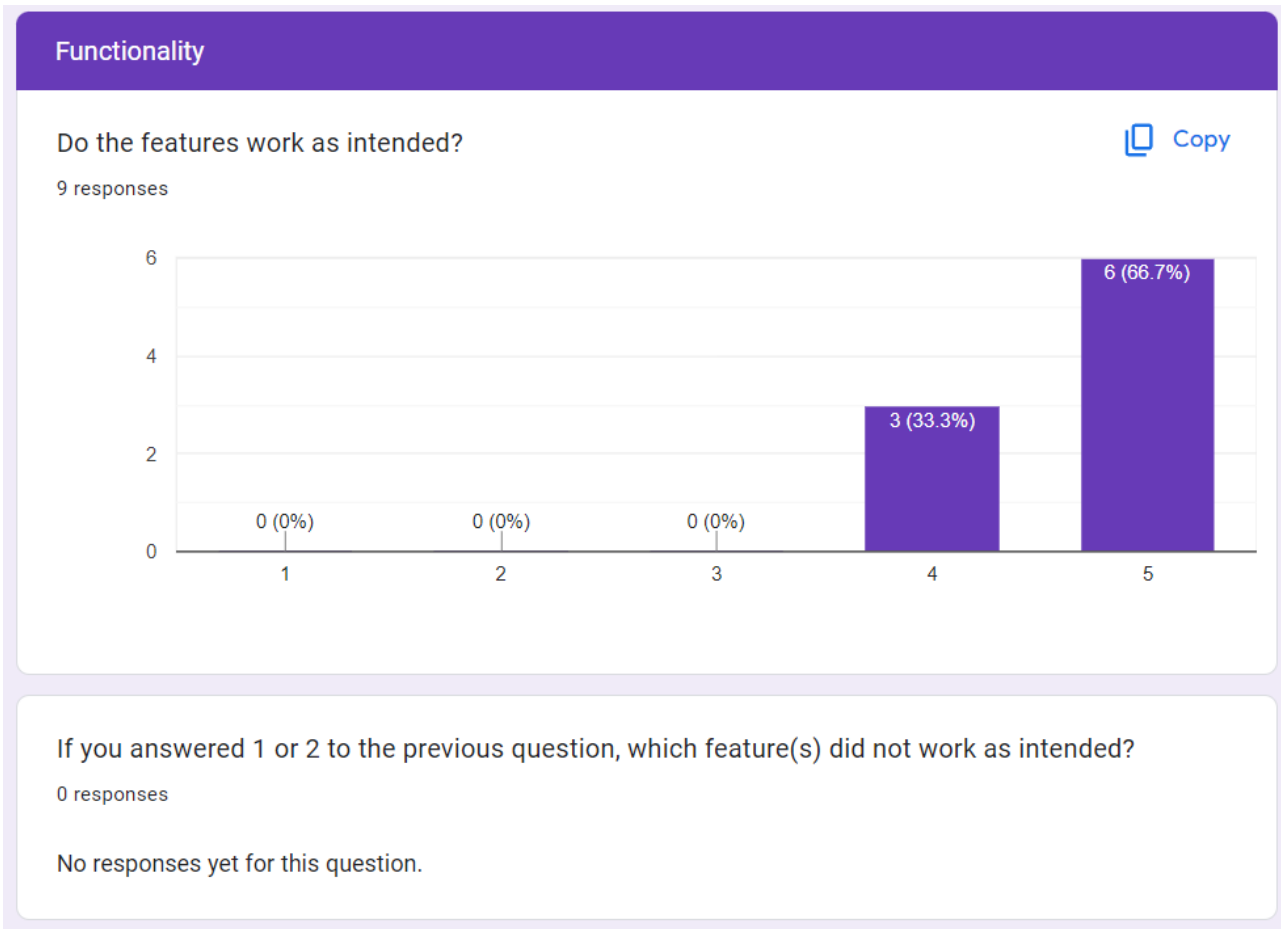
The leaderboard for first, second or third can have like a gold silver bronze color and maybe the top 10 can get a different color as well for incentivization

Any additional feedback on the UI?

7 responses

- For the threads encase each post in a visible box for better segregation of the posts
- Certain UI elements can be done better such as having a background for the leaderboard page
- Maybe show a podium style display for the top 3 people
- The threads can have a boarder between the posts or some display of segregation to distinguish each post
- The leaderboard page is quite plain, possibly let the top 3 have a unique display
- The course thread can be designed better to make it seem less cluttered like hiding part of the description and allowing the user to see more if they want since the description is very long
- The thread page can make posts more individual by having dividers
 - The Profile page can be less cluttered by moving some feature to other pages or let the user scroll to see more
 - The post button for the thread feature should also come with some text instead of just one icon
 - The recommendations for the search bar can include the course name also

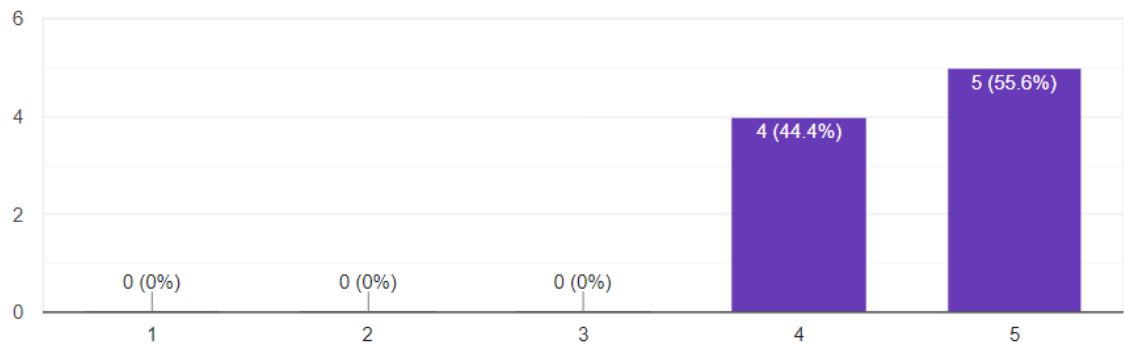
Functionality



Do you understand how to use all features?

 Copy

9 responses



If you answered 1 or 2 to the previous question, which feature(s) did not work as intended?

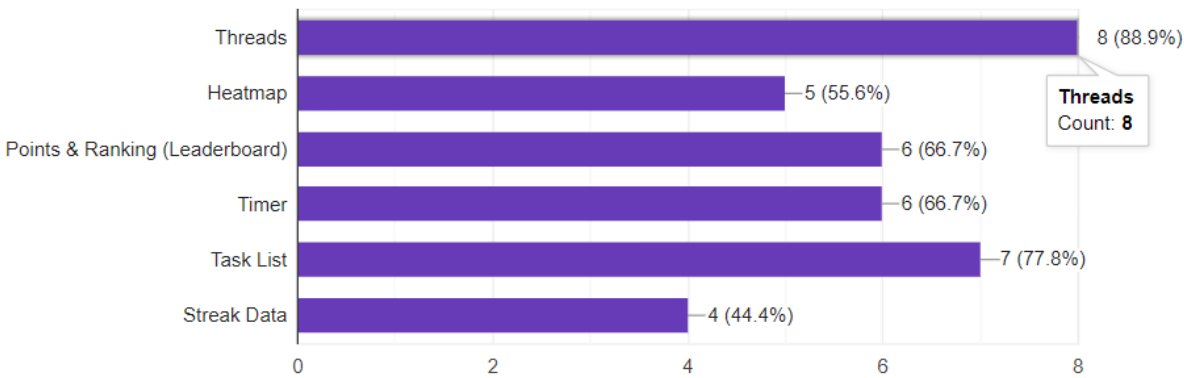
0 responses

No responses yet for this question.

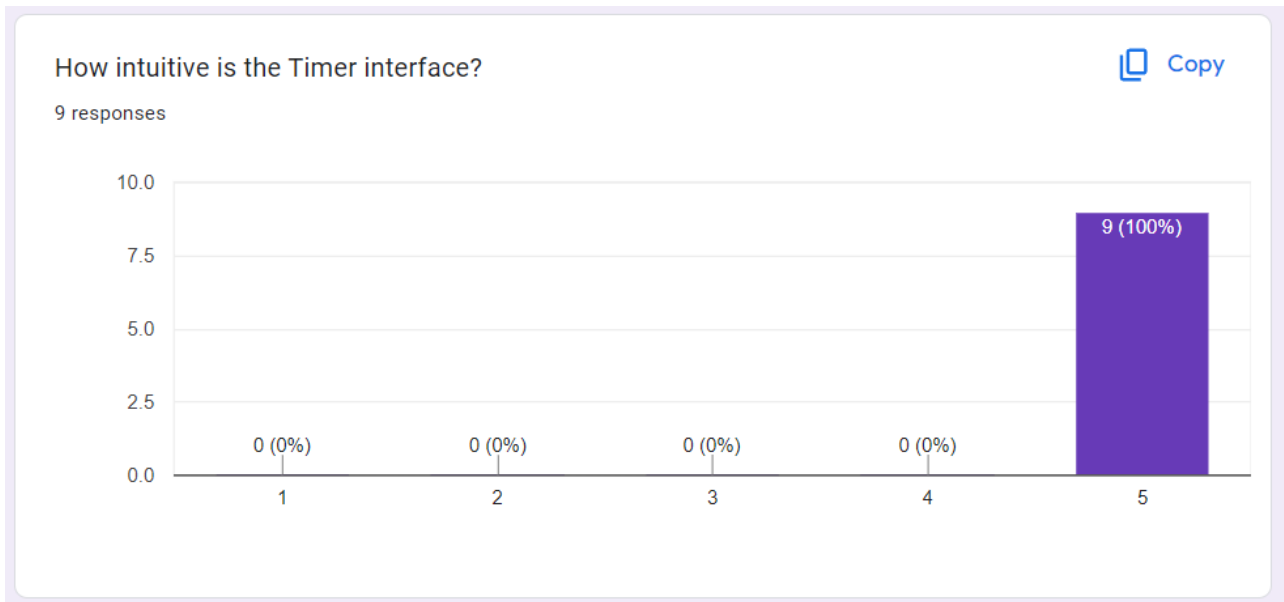
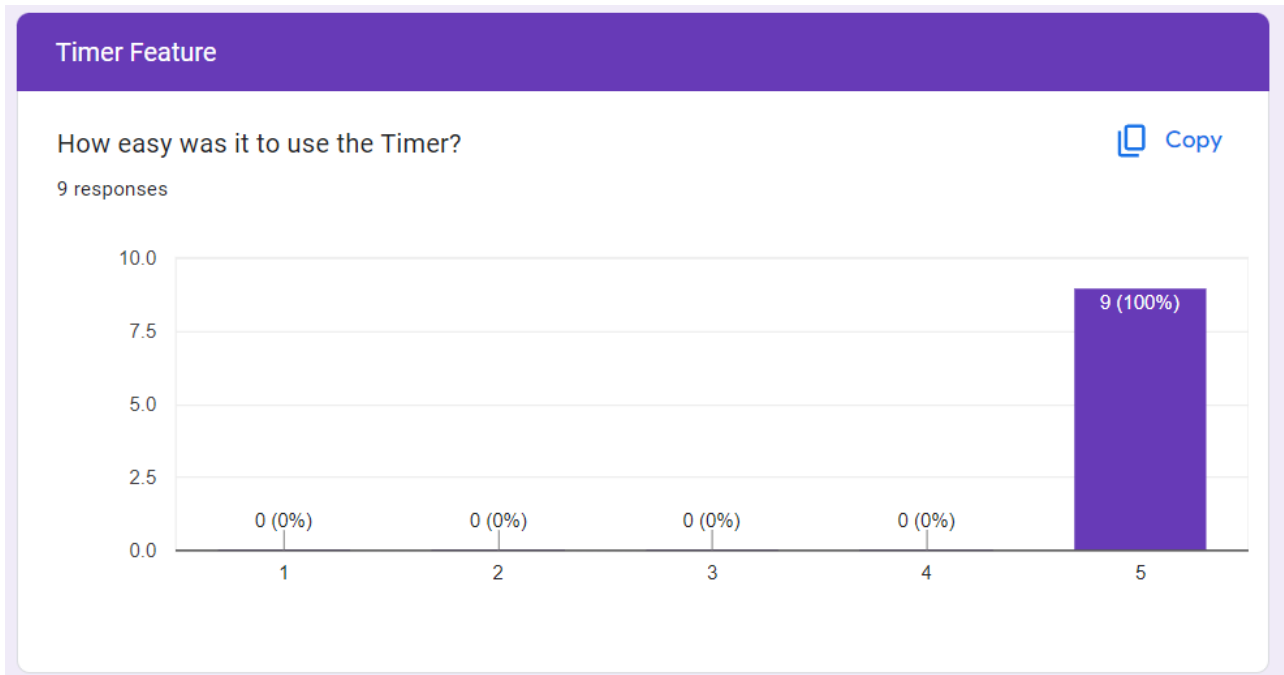
Which feature did you feel satisfied with? (Multiple select)

 Copy

9 responses



Timer Feature



Suggestions on additional function(s) / Bug(s) Found

4 responses

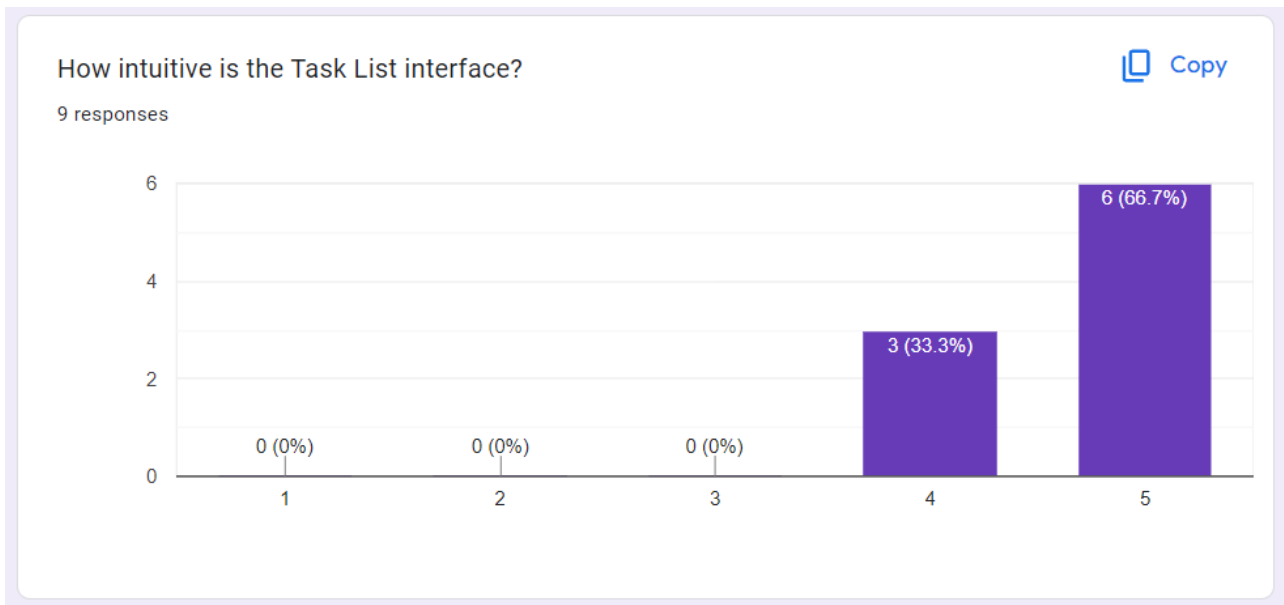
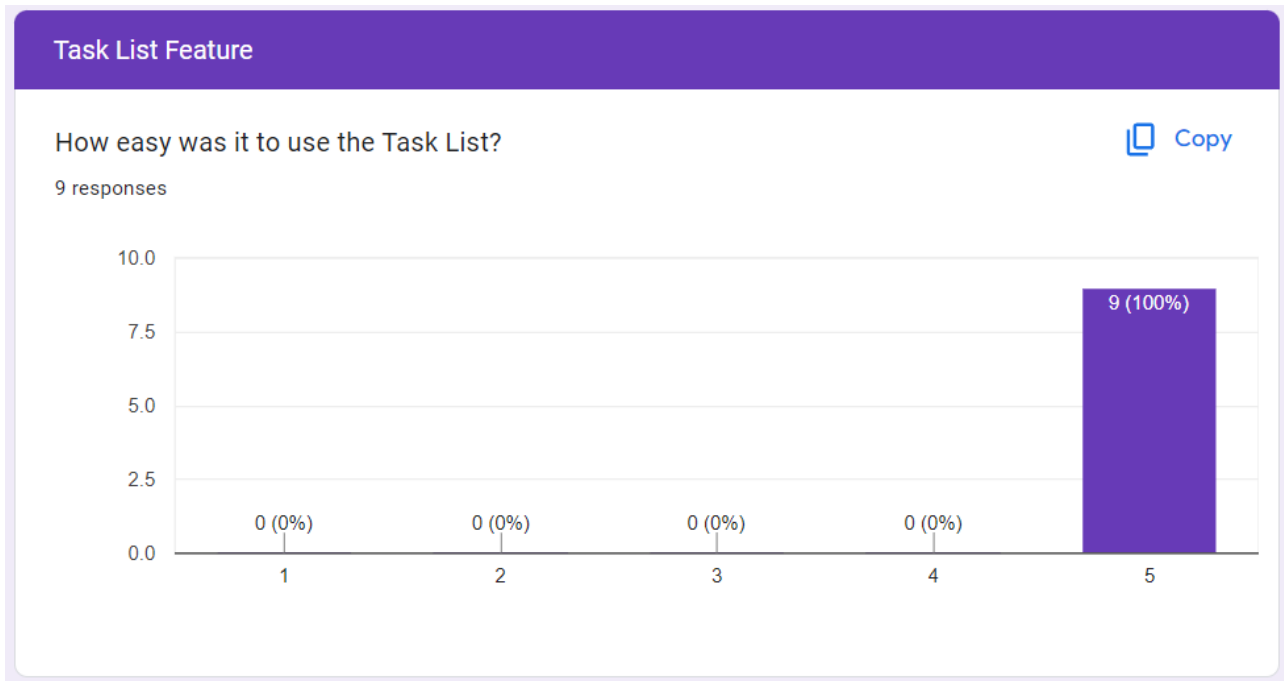
Maybe do not let the user stop the timer as it might defeat the purpose as the user can be tempted to stop at any time making it less effective

Nil

I feel that having the stop button defeats the purpose of the function as the user can be tempted to just stop the timer at any point in the study session

I think just having a animation count down will be better than having a circle

Task List Feature



Suggestions on additional function(s) / Bug(s) Found

6 responses

Possibly add a filter function to filter tasks based on the date range

Possibly have a confirmation button before the deleting the task

Limit the date picker such that it cannot select any dates that has past

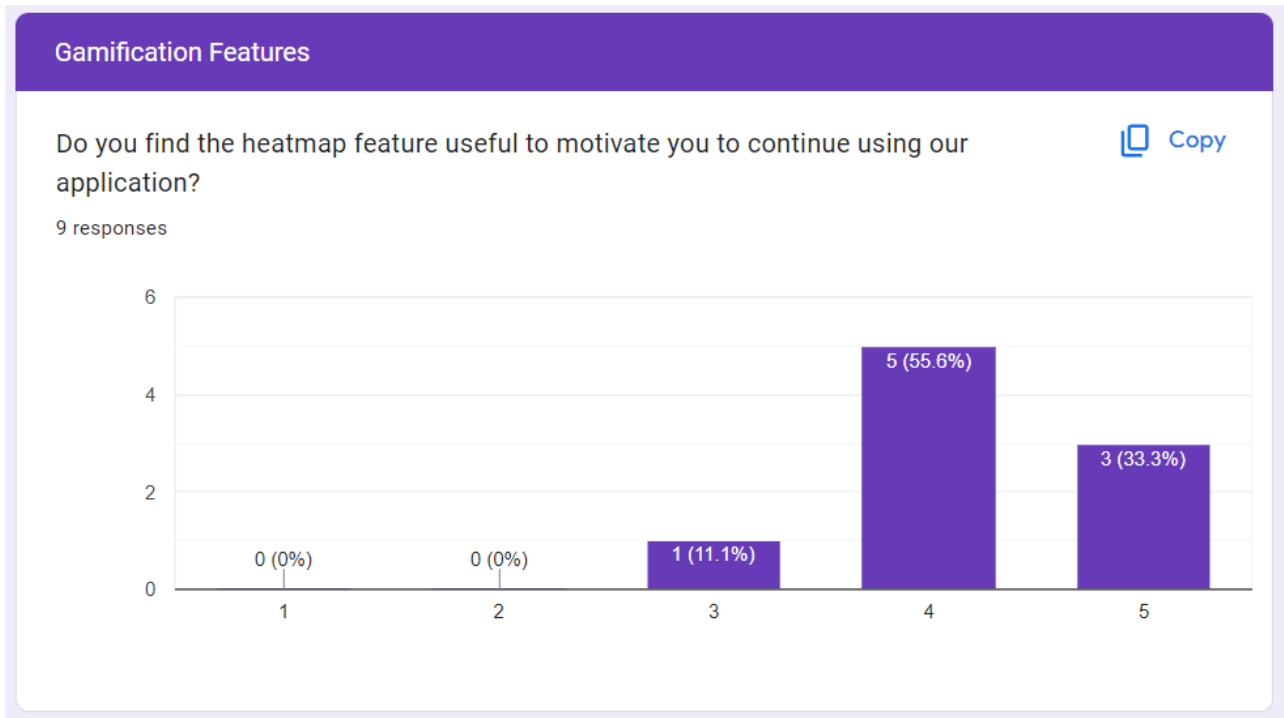
The deletion of the tasks can have a confirmation button to prevent accidentally deleting a task

The spacing between the tasks is a bit cluttered and the date time picker should restrict dates that have already past

Deletion of tasks can have some sort of confirmation

- Extend the view for the task list as it shows at most 2 to 3 tasks

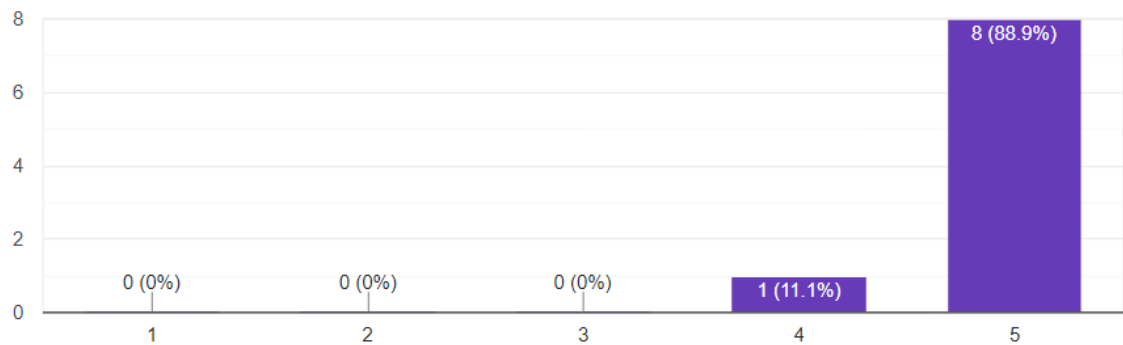
Gamification Feature



Do you find the point system to be intuitive to understand?

 Copy

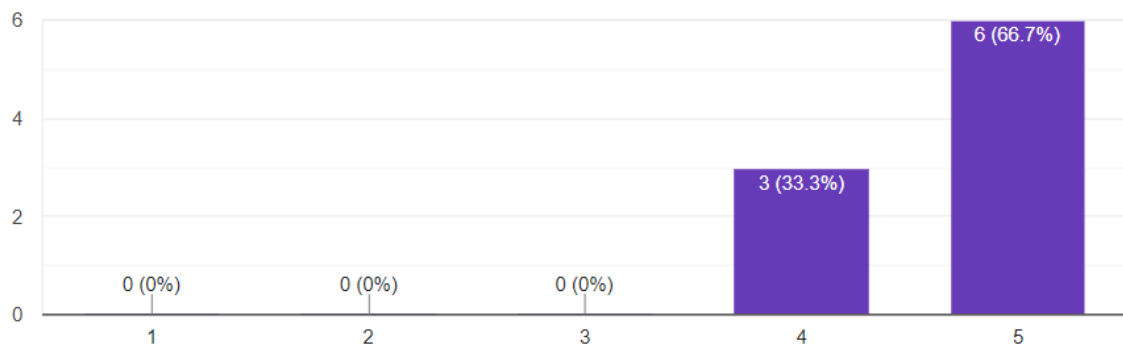
9 responses



Do you find the leaderboard useful to motivate you to continue using our application?

 Copy

9 responses



Suggestions on additional function(s) / Bug(s) Found

4 responses

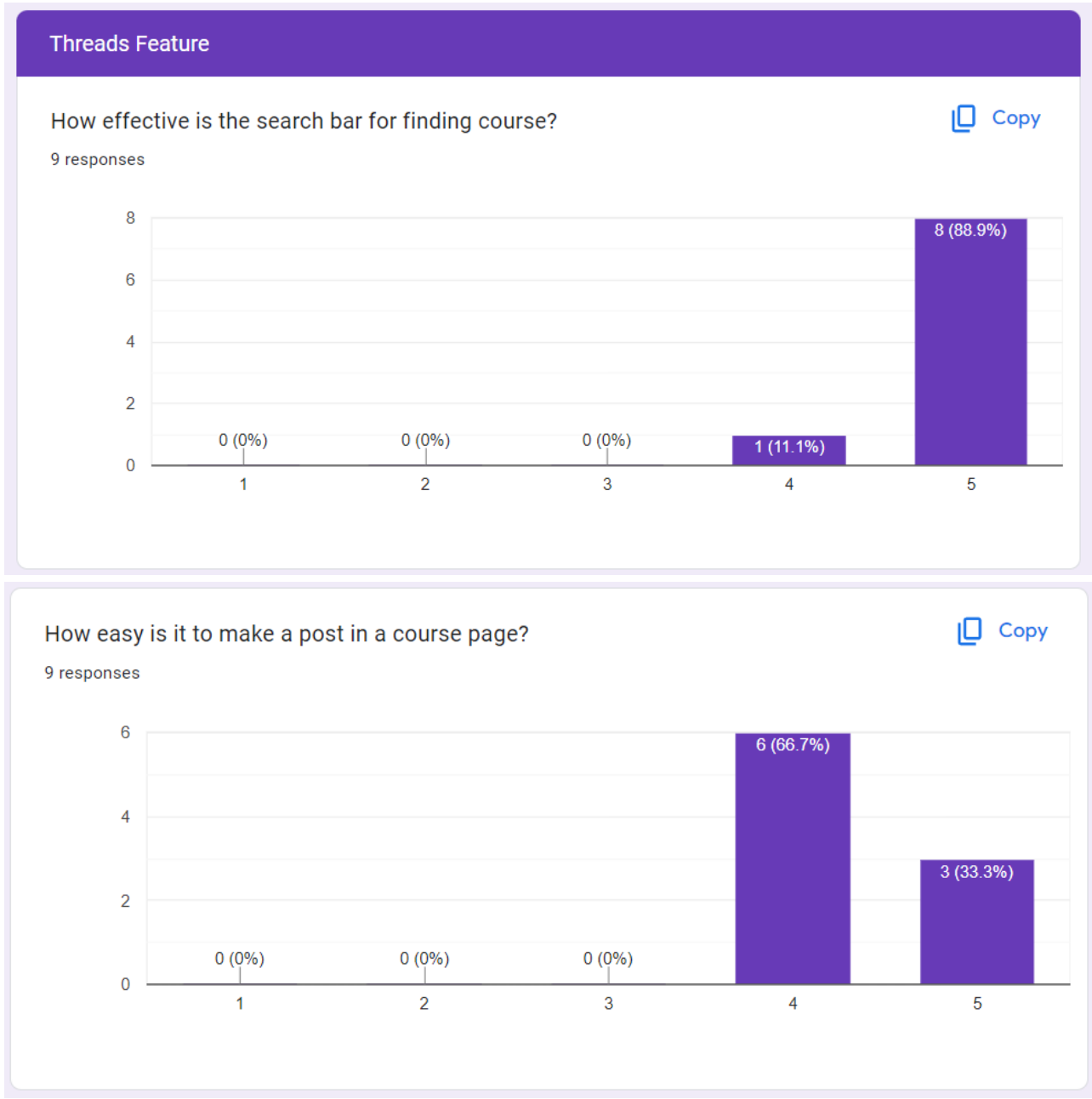
For the point system, maybe the app should award 1 point per hour instead of pro rating based on how long they have studied

Instead of listing every person, possibly show maybe the top 10 then show your ranking afterwards if the user is not in the top 10

The leaderboard can be in its own separate page to prevent cluttering in the profile page

- Can make the streak more interactive by making the flame change color when the streak is broken

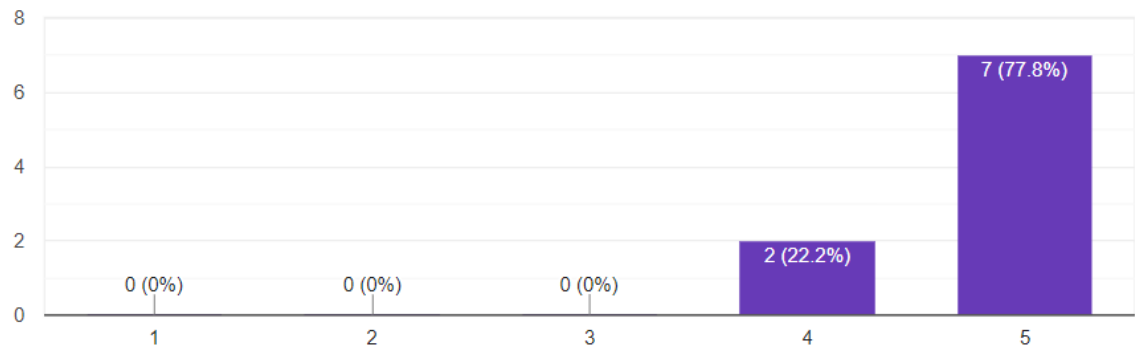
Thread Feature



How easy is it to reply to a post?

 Copy

9 responses



Suggestions on additional function(s) / Bug(s) Found

4 responses

- For posting instead of just an image, it should display a text with a image for better readability
- Possibly include a persisting add post button so that the user does not always need to go into a course forum before posting
- Possibly have a dedicated post option in the navigation bar like reddit and Instagram has for ease of posting to the different threads
- The background for the dropdown for the suggestions when searching for the course has a misaligned height with the number of options