Milestone 3

Team Name:
StudyPal_2

URL:
https://main--studyypal.netlify.app/login

Proposed Level of Achievement:
Apollo 11

Motivation:

Many students find that studying alone can be challenging, and that having a study partner can provide motivation, accountability, and a different perspective of course material. However, finding a compatible study partner can be difficult, especially for students who are new to a university or who have a busy schedule.

Aim

We hope to address this challenge by connecting students with compatible study partners based on their academic interests, availability, and study preferences. By collaborating with a study partner, students can share knowledge, review course material, and ask questions. This can lead to a deeper understanding of the material and improved academic performance.

User Stories

1. As a university student, I want to find a study partner who is taking the same courses as me and has a similar schedule, so that we can review course material together and prepare for exams.
2. As a student who is studying remotely, I want to connect with other students who are studying in the same field as me, so that I can build relationships and feel more engaged with my studies.
3. As a student who is struggling with a particular course, I want to find a study partner who has a strong understanding of the material, so that they can help me to review and clarify key concepts.
4. As a student who is interested in collaborating on a research project, I want to find a study partner who shares my research interests and has the skills and knowledge to help me with the project.

Scope of the project:

Our web platform aims to provide a comprehensive set of tools and features to enhance the academic experience for students at NUS.

The platform facilitates the creation and discovery of study jios, where students can initiate study sessions and find suitable study partners based on their academic interests, study preferences, and availability.

Additionally, it offers a centralised repository for module resources, including past year papers and textbooks, making it easier for students to access and share these materials.

The platform integrates with Telegram to organise and list module-specific chat groups, streamlining communication among students.

The calendar feature allows students to manage their study sessions, important dates, and deadlines in one place, syncing with external calendar applications.

The project scope encompasses the development of these primary features, with additional features such as gamification, communication tools, and a feedback and rating system as potential future enhancements.

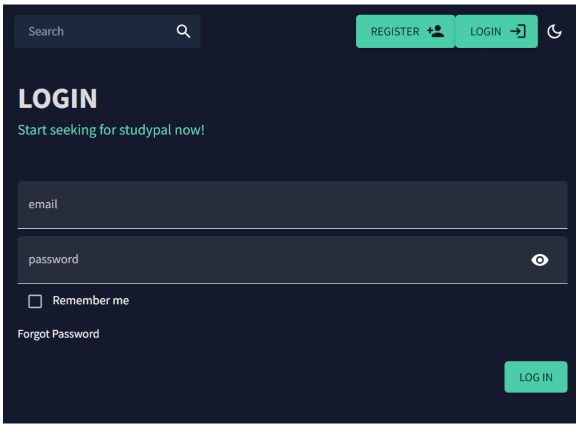
Primary features

1. User profile creation:

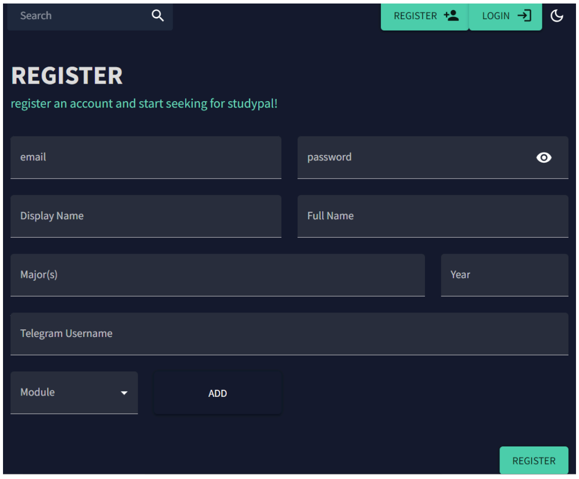
The user would first sign up a profile using his email and password. This function is supported by the firebase authentication function. Then after logging in, the user would be led to another page to fill in their details such as username, major, telegram handle, modules etc immediately after the account was created; this is achieved by using Firestore database.

More sign-in and register methods might be provided, such as gmail authentication and canvas account authentication/ login via NUS. More user's details can be incorporated in the user's profile such as studying styles, average attention span, academic strengths etc, and this will facilitate the recommendation system to provide the user with a potential suitable partner.

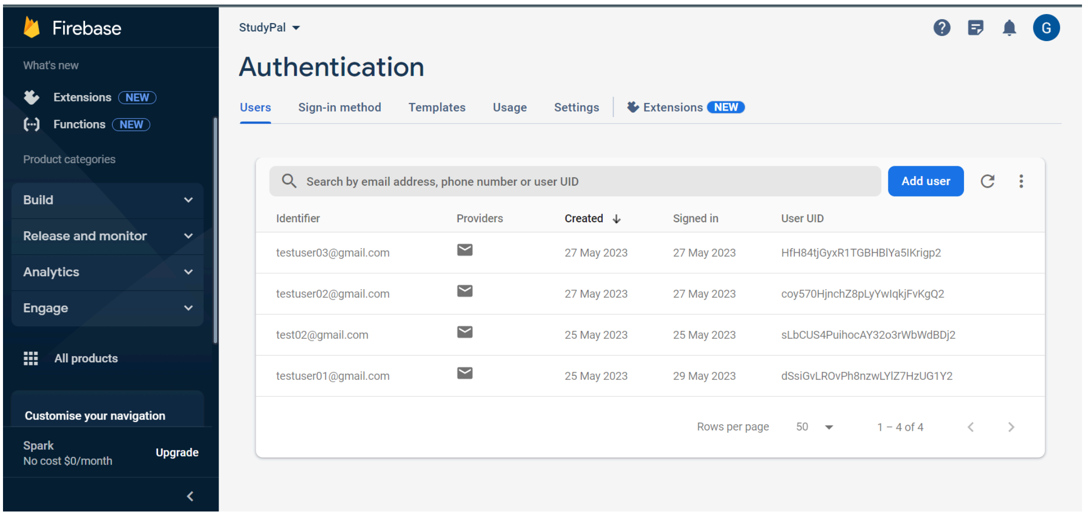
1. The login interface, the button on topbar will enable the user to register a new account



2. The sign up interface

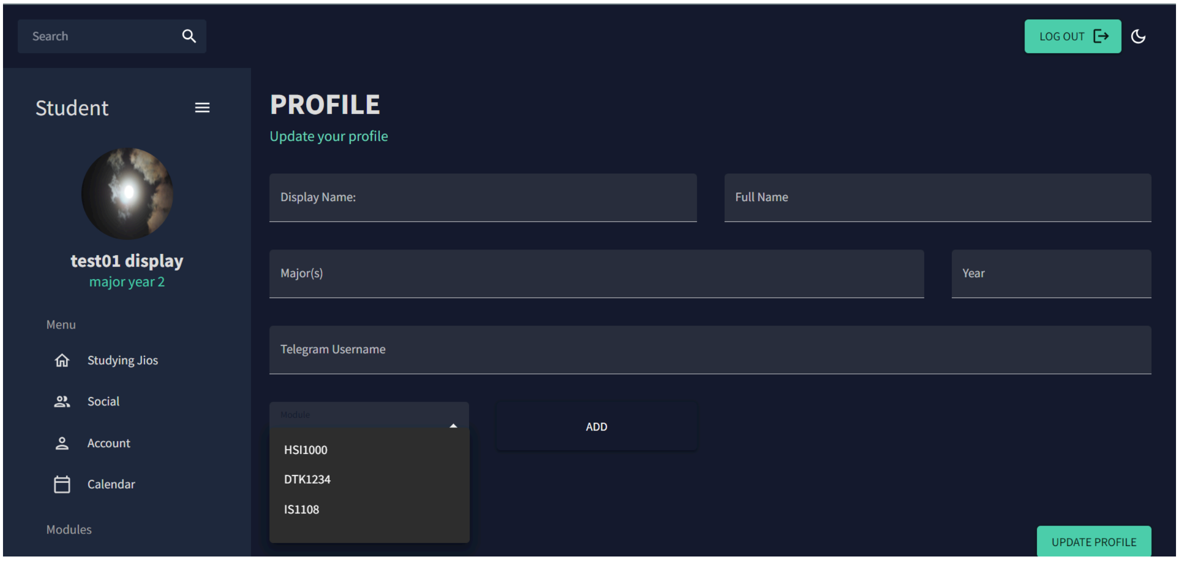


3. The monitor interface of the users' account



Identifier	Providers	Created ↓	Signed in	User UID
testuser03@gmail.com	✉	27 May 2023	27 May 2023	HfH84tjGyxR1TGBHBlYa5lKrigp2
testuser02@gmail.com	✉	27 May 2023	27 May 2023	coy570HjnchZ8pLyYwlqkjFvKgQ2
test02@gmail.com	✉	25 May 2023	25 May 2023	sLbCUS4PuihocAY32o3rWbWdBDj2
testuser01@gmail.com	✉	25 May 2023	29 May 2023	dSsiGvLROvPh8nzwLYlZ7HzUG1Y2

4. The user can update their profile with more details in the account page



2. Social (Connecting with other users):

A page that allows users to view the profiles of other registered users, find their contact details and send/ accept friend requests. This page can also display the user's "friend list", as well as their friends' recent activity (if set to public).

Sort/ Filter Functions:

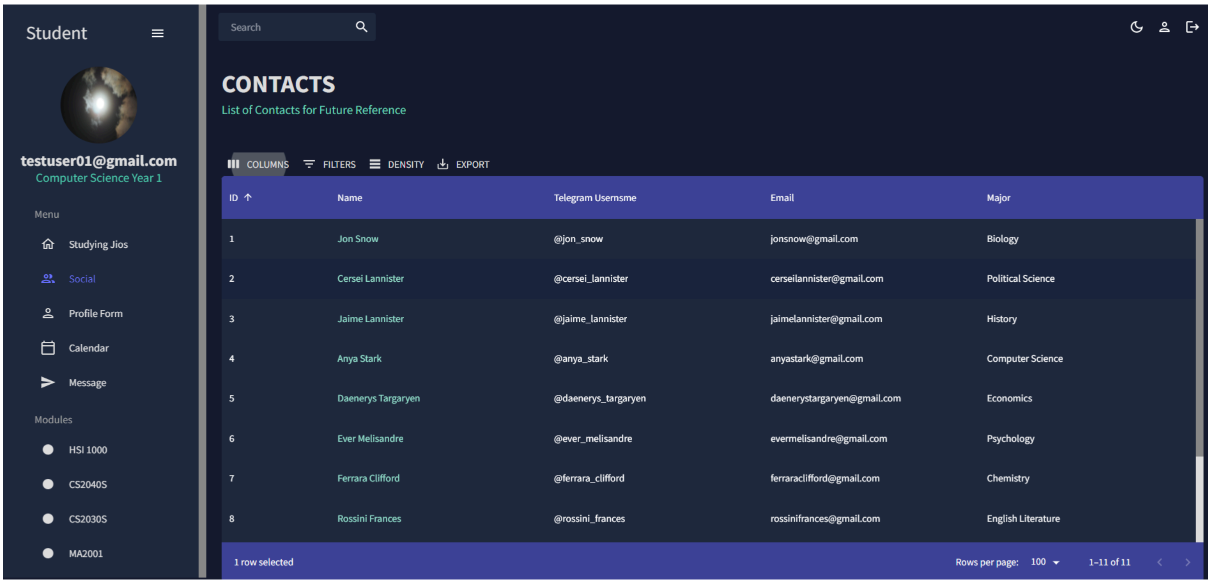
- The header of each column can be clicked to sort the contents of the table by the values in that column
- Clicking multiple column headers will sort the contents of the table by the first column clicked, with the values in subsequent columns being used to resolve any ties in the ranking of entries.

- Entries can be filtered according to individual column values

Layout Customisation:

- Users can select which columns they want to show or hide

- The compactness of each row (and thus the number of entries shown on screen) can be adjusted by the user by picking one of three settings



CONTACTS

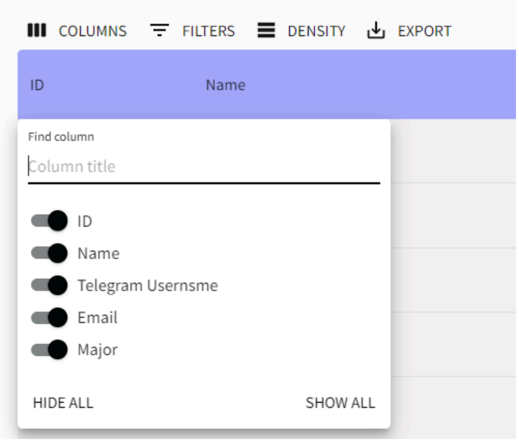
List of Contacts for Future Reference

ID ↑	Name	Telegram Usernsme	Email	Major
1	Jon Snow	@jon_snow	jonsnow@gmail.com	Biology
2	Cersei Lannister	@cersei_lannister	cerseilannister@gmail.com	Political Science
3	Jaime Lannister	@jaime_lannister	jaimelannister@gmail.com	History
4	Anya Stark	@anya_stark	anyastark@gmail.com	Computer Science
5	Daenerys Targaryen	@daenerys_targaryen	daenerystargaryen@gmail.com	Economics
6	Ever Melisandre	@ever_melisandre	evermelisandre@gmail.com	Psychology
7	Ferrara Clifford	@ferrara_clifford	ferraraclifford@gmail.com	Chemistry
8	Rossini Frances	@rossini_frances	rossinifrances@gmail.com	English Literature

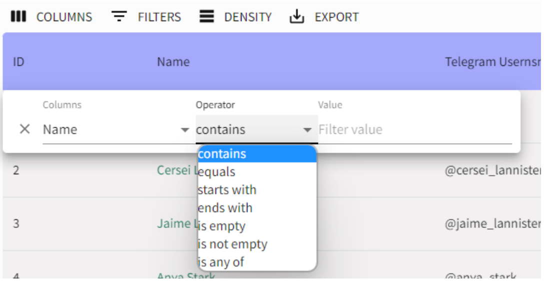
1 row selected   Rows per page: 100   1–11 of 11

Figure: Social Page

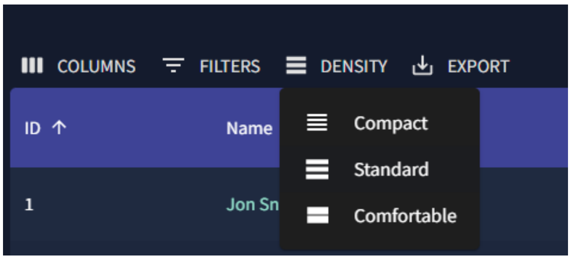
1: Show/Hide the columns selected



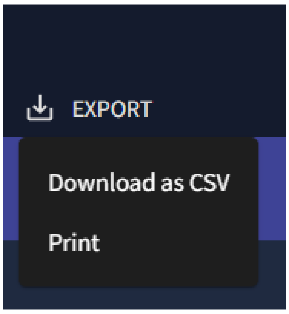
2: Filter according to a particular columns value



3: Adjust the compactness of each row



4: Exporting the table as CSV or PDF(print)

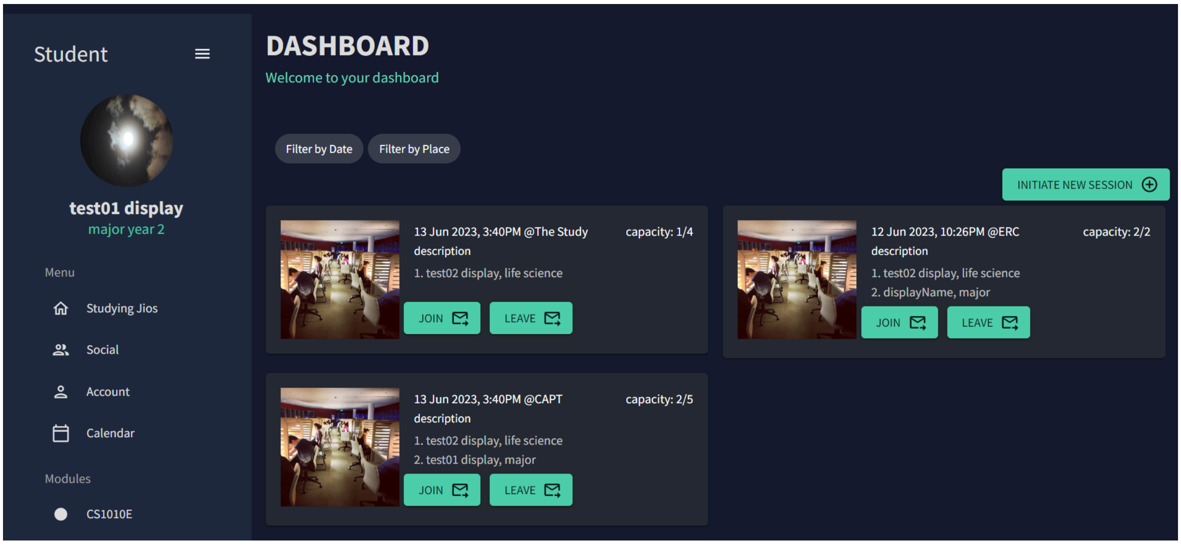


3. Studying Jios

It allows students to initiate study jios and indicate details such as the date/time, venue (physical/ online/ hybrid), capacity and description to establish goals/requests. Once a study jio has been launched, it will be publicly available and other students can find it via the search/filter function and join for the session. The participants can also leave the joined jio if they want to. Currently, the participant can join or leave a jio without seeking permission from the initiator. However, a complete request, accept and decline functionality would be implemented in milestone 3.

Matching preferences & algorithm: Before the filters being implemented, Students would be given recommendations for the potential studying sessions according to their past studying session history.

1. Dashboard of studying jio:

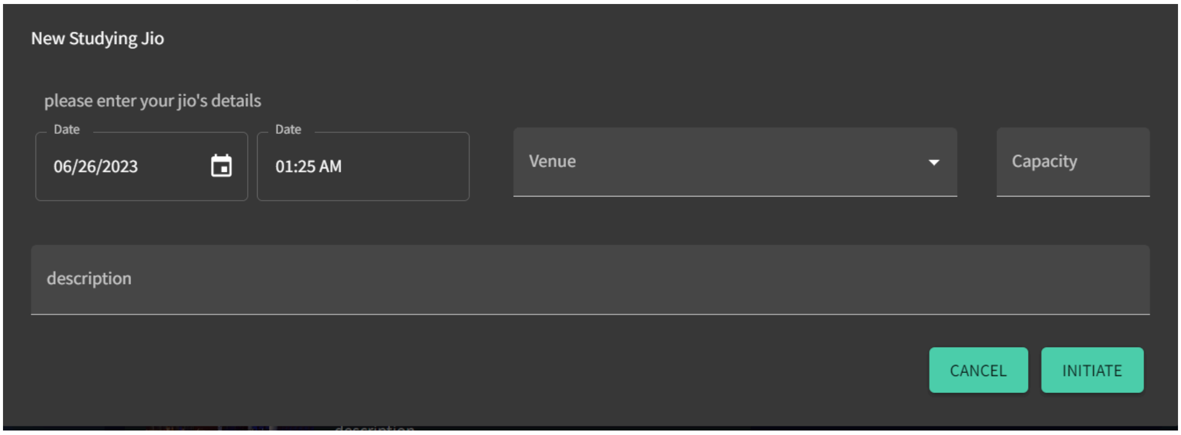


13 Jun 2023, 3:40PM @The Study capacity: 1/4
description
1. test02 display, life science

12 Jun 2023, 10:26PM @ERC capacity: 2/2
description
1. test02 display, life science
2. displayName, major

13 Jun 2023, 3:40PM @CAPT capacity: 2/5
description
1. test02 display, life science
2. test01 display, major

2. The window for initiating a new session

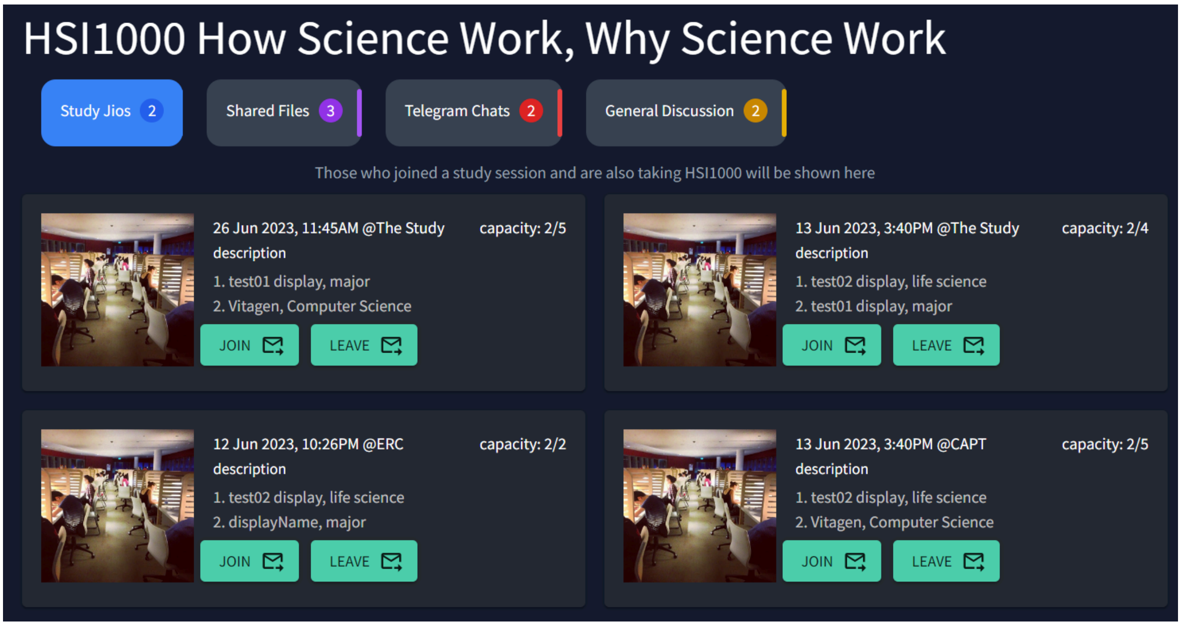


Date: 06/26/2023
Date: 01:25 AM

4. Modules Pages

When the user enrolled in certain modules and declared the modules in these profile, respective modules pages will be shown in the sidebar. Each modules page includes 4 tabs:

a. Study jios: the page will display filtered study jios which have at least one of the jio participants that enrolled in the particular module, thus being able to facilitate collaborative learning between module-mates.



HSI1000 How Science Work, Why Science Work

Those who joined a study session and are also taking HSI1000 will be shown here

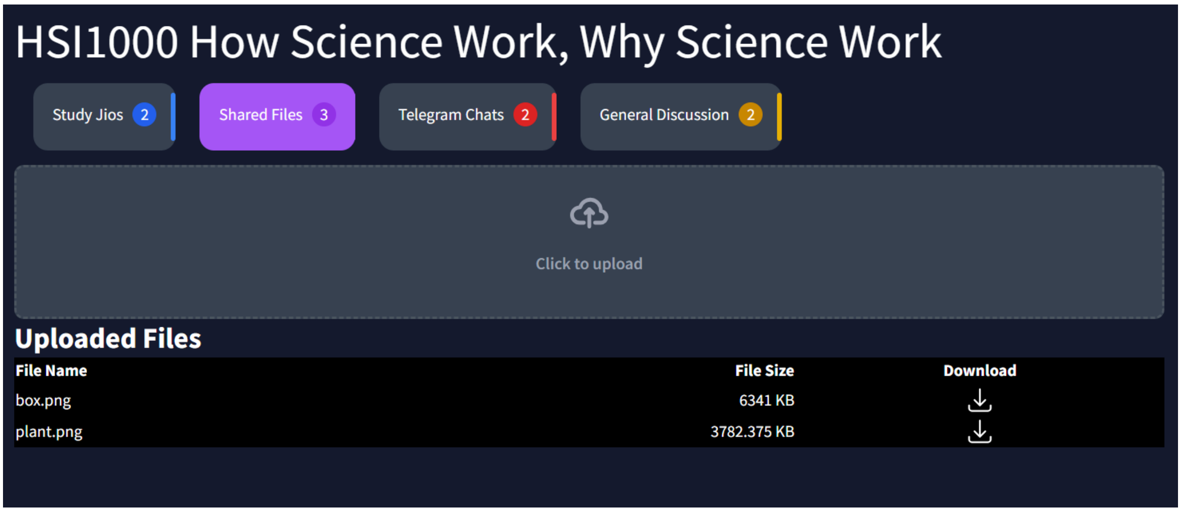
26 Jun 2023, 11:45AM @The Study capacity: 2/5
description
1. test01 display, major
2. Vitagen, Computer Science

13 Jun 2023, 3:40PM @The Study capacity: 2/4
description
1. test02 display, life science
2. test01 display, major

12 Jun 2023, 10:26PM @ERC capacity: 2/2
description
1. test02 display, life science
2. displayName, major

13 Jun 2023, 3:40PM @CAPT capacity: 2/5
description
1. test02 display, life science
2. Vitagen, Computer Science

b. Shared files: The students are able to upload and download files with any kind of file types into this shared space. Therefore relevant resources such as past year questions, tutorials questions and answers could be shared among the students and benefits them as a whole.

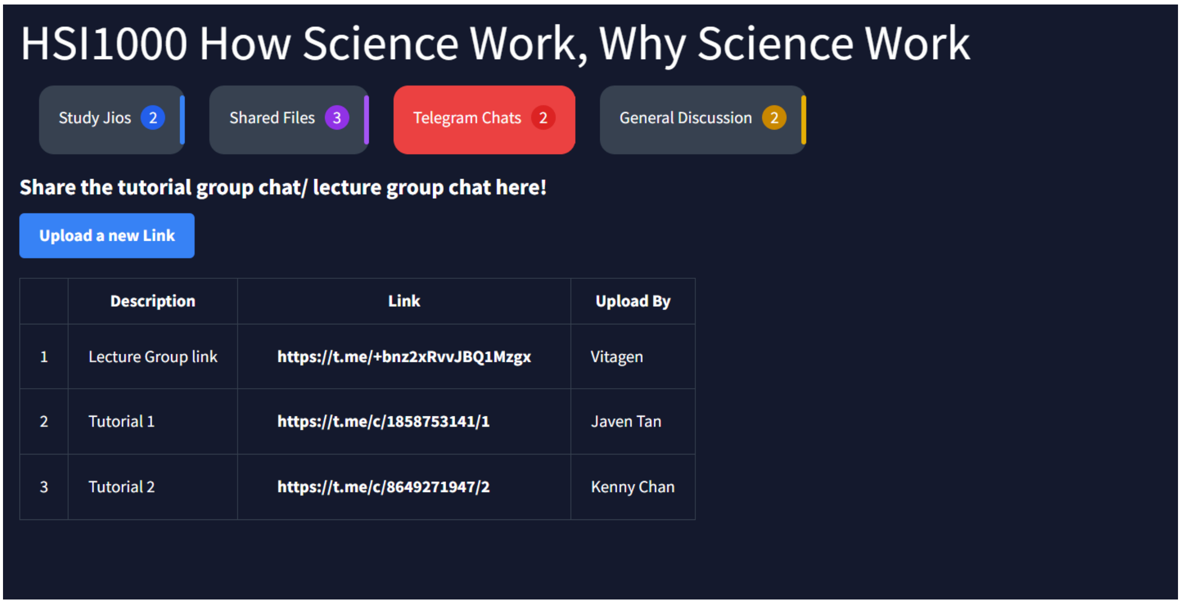


HSI1000 How Science Work, Why Science Work

Uploaded Files

File Name	File Size	Download
box.png	6341 KB	⤓
plant.png	3782.375 KB	⤓

c. Telegram Chat: Similar to the shared files, students, especially teaching assistants, could upload their telegram group link here to provide multiple platform for discussion among the classes.

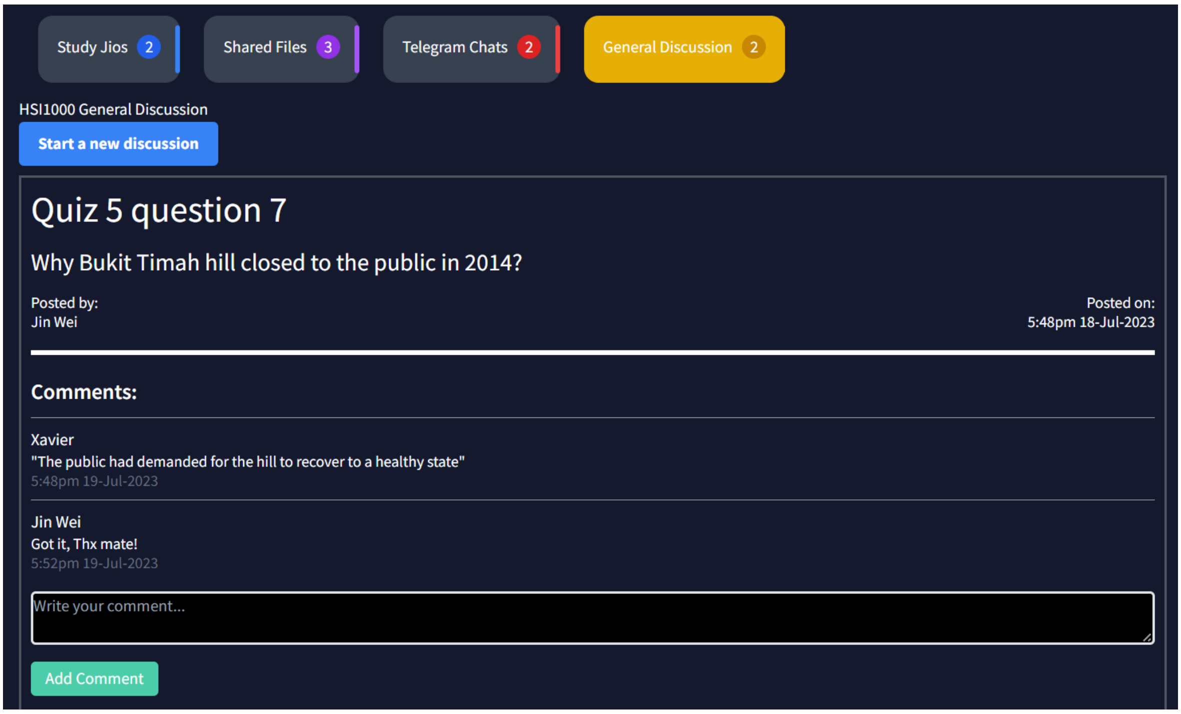


HSI1000 How Science Work, Why Science Work

Share the tutorial group chat/ lecture group chat here!

	Description	Link	Upload By
1	Lecture Group link	https://t.me/+bnz2xRvvJBQ1Mzgx	Vitagen
2	Tutorial 1	https://t.me/c/1858753141/1	Javen Tan
3	Tutorial 2	https://t.me/c/8649271947/2	Kenny Chan

d. General Discussion: A thread that enables students to post questions and queries and seek help. After a post is created, others could comment below the created thread and therefore foster discussion among the students.



HSI1000 General Discussion

Quiz 5 question 7

Why Bukit Timah hill closed to the public in 2014?

Posted by:
Jin Wei

Posted on:
5:48pm 18-Jul-2023

Comments:

Xavier
"The public had demanded for the hill to recover to a healthy state"
5:48pm 19-Jul-2023

Jin Wei
Got it, Thx mate!
5:52pm 19-Jul-2023

5. Calendar

The calendar feature provides a centralised tool for students to organise and track their group studying sessions and important dates, deadlines, and events related to their courses. Calendars can be integrated and synced with external calendar applications such as Google Calendar, Canvas calendar and Outlook. This enables users to access their studying jios alongside their other personal and professional commitments.

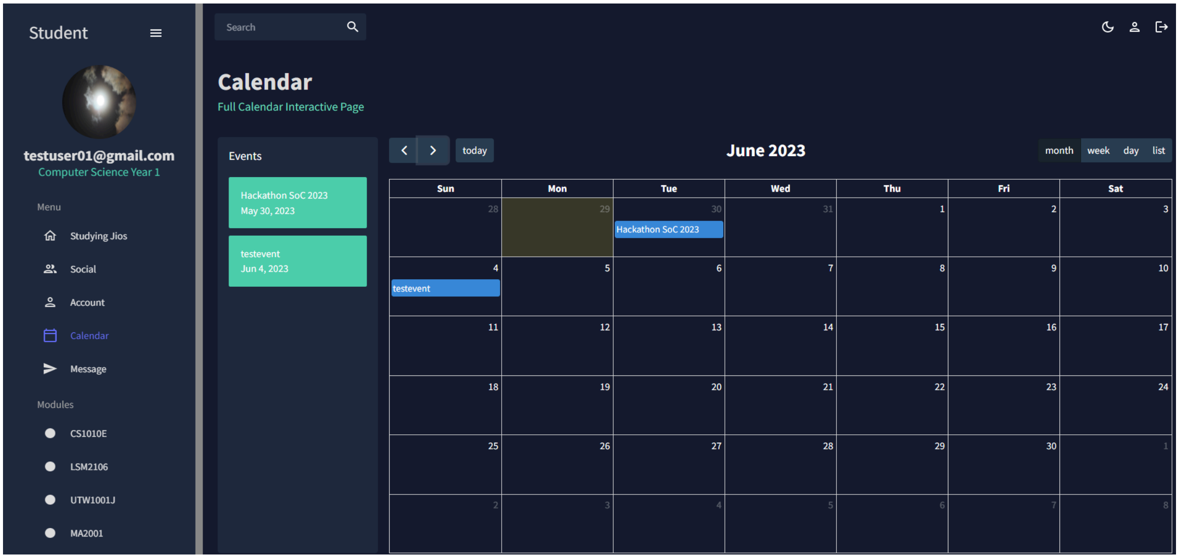
Adding new event:

- New events can be added to the calendar by clicking the corresponding date.
- Created events will be displayed on left sorting according to its precedence

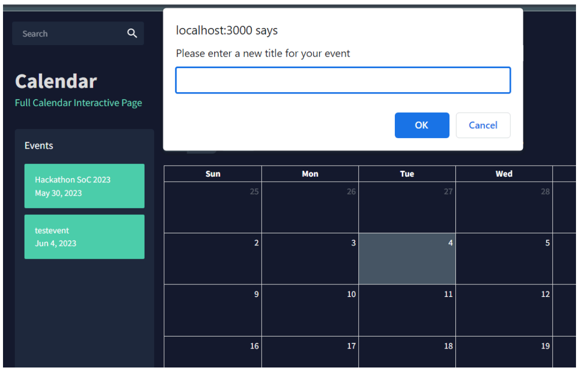
Navigation:

- The calendar has different view mode, i.e. month, week, day, and can be navigated using the arrow on top left

1. The interface of the calendar function, highlighted box shows the current date

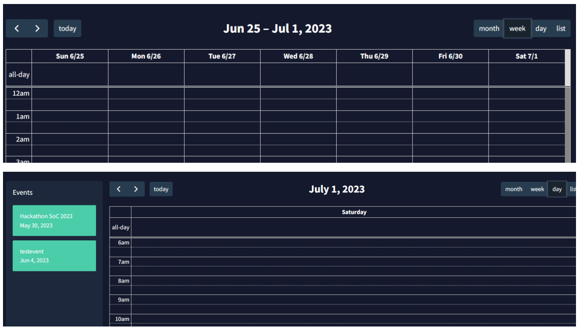


2. Event creation


localhost:3000 says

Please enter a new title for your event

3. Sample showcase of viewing in week and day



6. Community for each modules:

These module tabs are added by the users based on the modules that they are enrolled in. Each module consists of 4 tabs:

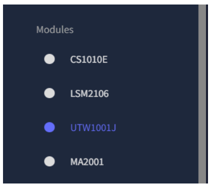
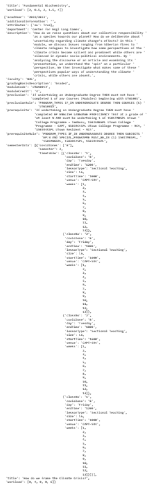
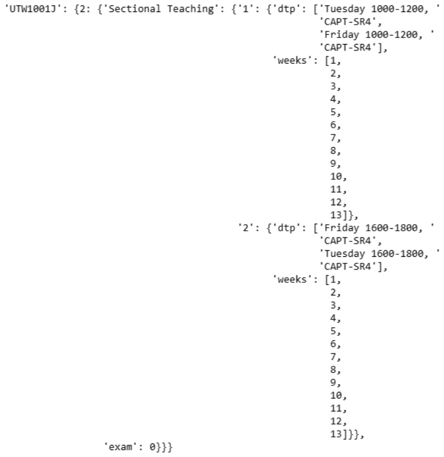
a. Study Jios:
b. Shared Files: users can access and share past and current module resources such as past year papers, cheat sheets, and notes. Files from past semesters will be archived and can be accessed by the current enroller.
c. Telegram Chats: each module will provide telegram chat group links for different

lecture and tutorial groups for the users easy access if they prefer telegram apps more for quick questions.

d. Discussion: Users can post questions here and other could reply to the posts.

Backend data processing:

1. Data collection from NUSmods:

	1. Study Jios 2. Shared Files* 3. Telegram Chats* 4. Discussion
Module Pages tab in sidebar	Tabs for each module page (* indicates that scraped data will be needed to structure the page)
	
Raw data scraped from NUSmods API using custom Python script	Python dictionary condensed_modsData generated and used for downstream processes using same Python script

```
 'title': 'Fundamental Biochemistry',
 'workload': [2, 0.5, 2, 1.5, 4]}
{'acadYear': '2022/2023',
 'additionalInformation': '',
 'attributes': {'su': True},
 'department': 'Center for Engl Lang Comms',
 'description': 'How do we raise questions about our collective responsibility '
                'as a species towards our planet? How do we deliberate about '
                'uncertainty regarding climate change's effects? In this '
                'module, we discuss issues ranging from Siberian fires to '
                'climate refugees to investigate how some perspectives of the '
                'climate crisis become salient and prominent while others are '
                'silenced in dynamic socio-political environments. By '
                'analysing the discourse of an article and examining its '
                'presentation, we understand the "spin" on a particular '
                'perspective. We then investigate what makes some of these '
                'perspectives popular ways of understanding the climate '
                'crisis, while others are absent.',
 'faculty': 'NUS',
 'gradingBasisDescription': 'Graded',
 'moduleCode': 'UTW1001J',
 'moduleCredit': '4',
 'preclusion': 'If undertaking an Undergraduate Degree THEN must not have '
               'completed 1 of any Courses (Modules) beginning with UTW1001',
 'preclusionRule': 'PROGRAM_TYPES IF_IN UNDERGRADUATE DEGREE THEN COURSES (1) '
                   'UTW1001%',
 'prerequisite': 'If undertaking an Undergraduate Degree THEN must have '
                 'completed NP ENGLISH LANGUAGE PROFICIENCY TEST at a grade of '
                 'at least N AND must be undertaking 1 of 1501TMBSPL UTown '
                 'College Programme - Tembusu, 1502JNBSPL UTown College '
                 'Programme - CAPT, 1503RC4SPL UTown College Programme - RC4, '
                 '1503URSPL UTown Resident - RC4',
 'prerequisiteRule': 'PROGRAM_TYPES IF_IN UNDERGRADUATE DEGREE THEN SUBJECTS '
                     'NP:N AND SPECIAL_PROGRAMME MUST_BE_IN (1) 1501TMBSPL, '
                     '1502JNBSPL, 1503RC4SPL, 1503URSPL',
 'semesterData': [{'covidZones': ['B'],
                   'semester': 2,
                   'timetable': [{'classNo': '1',
                                  'covidZone': 'B',
                                  'day': 'Tuesday',
                                  'endTime': '1200',
                                  'lessonType': 'Sectional Teaching',
                                  'size': 16,
                                  'startTime': '1000',
                                  'venue': 'CAPT-SR4',
                                  'weeks': [1,
                                            2,
                                            3,
                                            4,
                                            5,
                                            6,
                                            7,
                                            8,
                                            9,
                                            10,
                                            11,
                                            12,
                                            13]},
                                 {'classNo': '2',
                                  'covidZone': 'B',
                                  'day': 'Friday',
                                  'endTime': '1800',
                                  'lessonType': 'Sectional Teaching',
                                  'size': 16,
                                  'startTime': '1600',
                                  'venue': 'CAPT-SR4',
                                  'weeks': [1,
                                            2,
                                            3,
                                            4,
                                            5,
                                            6,
                                            7,
                                            8,
                                            9,
                                            10,
                                            11,
                                            12,
                                            13]},
                                 {'classNo': '2',
                                  'covidZone': 'B',
                                  'day': 'Tuesday',
                                  'endTime': '1800',
                                  'lessonType': 'Sectional Teaching',
                                  'size': 16,
                                  'startTime': '1600',
                                  'venue': 'CAPT-SR4',
                                  'weeks': [1,
                                            2,
                                            3,
                                            4,
                                            5,
                                            6,
                                            7,
                                            8,
                                            9,
                                            10,
                                            11,
                                            12,
                                            13]},
                                 {'classNo': '1',
                                  'covidZone': 'B',
                                  'day': 'Friday',
                                  'endTime': '1200',
                                  'lessonType': 'Sectional Teaching',
                                  'size': 16,
                                  'startTime': '1000',
                                  'venue': 'CAPT-SR4',
                                  'weeks': [1,
                                            2,
                                            3,
                                            4,
                                            5,
                                            6,
                                            7,
                                            8,
                                            9,
                                            10,
                                            11,
                                            12,
                                            13]}]}],
 'title': 'How do we Frame the Climate Crisis?',
 'workload': [0, 4, 0, 0, 6]}
```

```
'UTW1001J': {2: {'Sectional Teaching': {'1': {'dtp': ['Tuesday 1000-1200, '
                                                      'CAPT-SR4',
                                                      'Friday 1000-1200, '
                                                      'CAPT-SR4'],
                                              'weeks': [1,
                                                        2,
                                                        3,
                                                        4,
                                                        5,
                                                        6,
                                                        7,
                                                        8,
                                                        9,
                                                        10,
                                                        11,
                                                        12,
                                                        13]},
                                        '2': {'dtp': ['Friday 1600-1800, '
                                                      'CAPT-SR4',
                                                      'Tuesday 1600-1800, '
                                                      'CAPT-SR4'],
                                              'weeks': [1,
                                                        2,
                                                        3,
                                                        4,
                                                        5,
                                                        6,
                                                        7,
                                                        8,
                                                        9,
                                                        10,
                                                        11,
                                                        12,
                                                        13]}},
                 'exam': 0}}}
```

Necessary information for downstream processes has been successfully scraped and stored as a Python dictionary, which can be easily converted into a .json file for integration into Firebase.

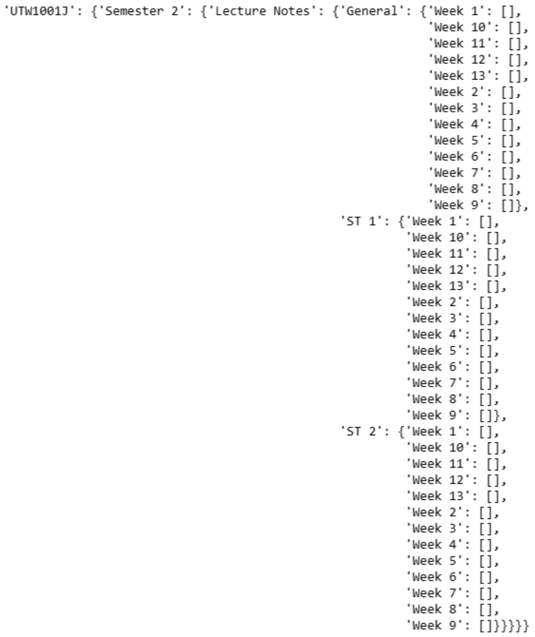
Data is organised as followed

- Module Code
  - Semester
    - Lesson Type
      - Class Number
        - Date, Time and Place (dtp, some classes have multiple weekly sessions)
        - Weeks

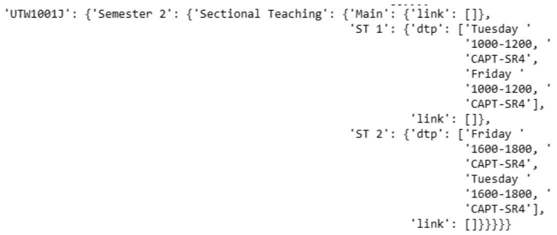
2. Generation of Directory Structure:

“Shared Files” Directory

“Telegram Chats” Directory



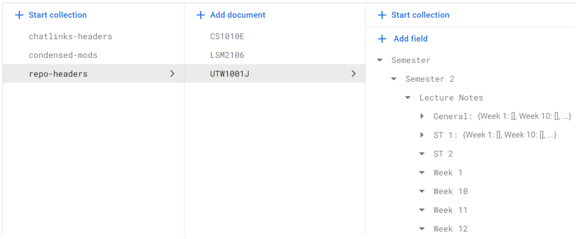
```
'UTW1001J': {'Semester 2': {'Lecture Notes': {'General': {'Week 1': [],
                                                          'Week 10': [],
                                                          'Week 11': [],
                                                          'Week 12': [],
                                                          'Week 13': [],
                                                          'Week 2': [],
                                                          'Week 3': [],
                                                          'Week 4': [],
                                                          'Week 5': [],
                                                          'Week 6': [],
                                                          'Week 7': [],
                                                          'Week 8': [],
                                                          'Week 9': []},
                                              'ST 1': {'Week 1': [],
                                                       'Week 10': [],
                                                       'Week 11': [],
                                                       'Week 12': [],
                                                       'Week 13': [],
                                                       'Week 2': [],
                                                       'Week 3': [],
                                                       'Week 4': [],
                                                       'Week 5': [],
                                                       'Week 6': [],
                                                       'Week 7': [],
                                                       'Week 8': [],
                                                       'Week 9': []},
                                              'ST 2': {'Week 1': [],
                                                       'Week 10': [],
                                                       'Week 11': [],
                                                       'Week 12': [],
                                                       'Week 13': [],
                                                       'Week 2': [],
                                                       'Week 3': [],
                                                       'Week 4': [],
                                                       'Week 5': [],
                                                       'Week 6': [],
                                                       'Week 7': [],
                                                       'Week 8': [],
                                                       'Week 9': []}}}}
```

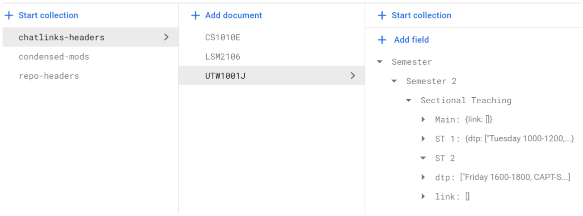


```
'UTW1001J': {'Semester 2': {'Sectional Teaching': {'Main': {'link': []},
                                                   'ST 1': {'dtp': ['Tuesday '
                                                                    '1000-1200, '
                                                                    'CAPT-SR4',
                                                                    'Friday '
                                                                    '1000-1200, '
                                                                    'CAPT-SR4'],
                                                            'link': []},
                                                   'ST 2': {'dtp': ['Friday '
                                                                    '1600-1800, '
                                                                    'CAPT-SR4',
                                                                    'Tuesday '
                                                                    '1600-1800, '
                                                                    'CAPT-SR4'],
                                                            'link': []}}}}}
```

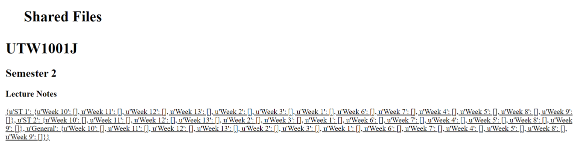
Empty lists denote stored files in the “Shared Files” Directory, and a single link for the “Telegram Chats” Directory

6.1 Python dictionary representations of the directory tree structure generated from condensed_modsData and saved to disk as .json files using a custom Python script





6.2 Firebase representation of the .json files uploaded using a second custom Python script



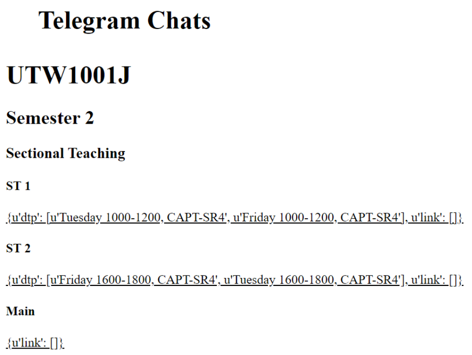
Shared Files

UTW1001J

Semester 2

Lecture Notes

{u'ST 1': {u'Week 10': [], u'Week 11': [], u'Week 12': [], u'Week 13': [], u'Week 2': [], u'Week 3': [], u'Week 1': [], u'Week 6': [], u'Week 7': [], u'Week 4': [], u'Week 5': [], u'Week 8': [], u'Week 9': []}, u'ST 2': {u'Week 10': [], u'Week 11': [], u'Week 12': [], u'Week 13': [], u'Week 2': [], u'Week 3': [], u'Week 1': [], u'Week 6': [], u'Week 7': [], u'Week 4': [], u'Week 5': [], u'Week 8': [], u'Week 9': []}, u'General': {u'Week 10': [], u'Week 11': [], u'Week 12': [], u'Week 13': [], u'Week 2': [], u'Week 3': [], u'Week 1': [], u'Week 6': [], u'Week 7': [], u'Week 4': [], u'Week 5': [], u'Week 8': [], u'Week 9': []}}.



Telegram Chats

UTW1001J

Semester 2

Sectional Teaching

ST 1

{u'dtp': [u'Tuesday 1000-1200, CAPT-SR4', u'Friday 1000-1200, CAPT-SR4'], u'link': []}.

ST 2

{u'dtp': [u'Friday 1600-1800, CAPT-SR4', u'Tuesday 1600-1800, CAPT-SR4'], u'link': []}.

Main

{u'link': []}.

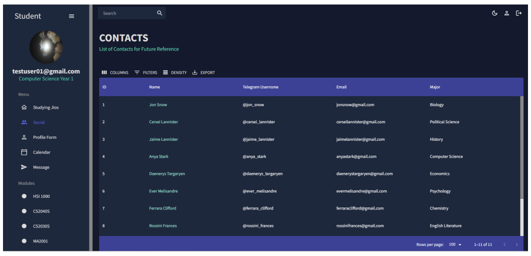
6.3 Rough rendering of header structure on Glitch (https://indecisive-honored-sandalwood.glitch.me/)

Necessary information for downstream processes has been successfully scraped and stored as a Python dictionary, which can be easily converted into a .json file for integration into Firebase.

6. Light/ Dark Mode

The dark and light mode allow users to choose between a dark colour scheme or a light colour scheme based on their preferences or environmental conditions.

Material-UI has been used to implement this function. The switching button is located at the global top bar so that the toggle can happen at any desired page.The colour of the theme can still be adjusted to improve the visibility of the words in certain pages.

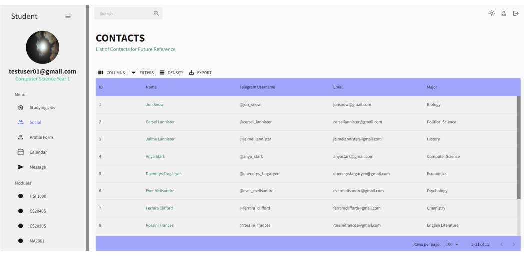


CONTACTS

List of Contacts for Future Reference

ID	Name	Telegram Usernsme	Email	Major
1	Jon Snow	@jon_snow	jonsnow@gmail.com	Biology
2	Cersei Lannister	@cersei_lannister	cerseilannister@gmail.com	Political Science
3	Jaime Lannister	@jaime_lannister	jaimelannister@gmail.com	History
4	Anya Stark	@anya_stark	anyastark@gmail.com	Computer Science
5	Daenerys Targaryen	@daenerys_targaryen	daenerystargaryen@gmail.com	Economics
6	Ever Melisandre	@ever_melisandre	evermelisandre@gmail.com	Psychology
7	Ferrara Clifford	@ferrara_clifford	ferraraclifford@gmail.com	Chemistry
8	Rossini Frances	@rossini_frances	rossinifrances@gmail.com	English Literature

Rows per page: 100 1–11 of 11



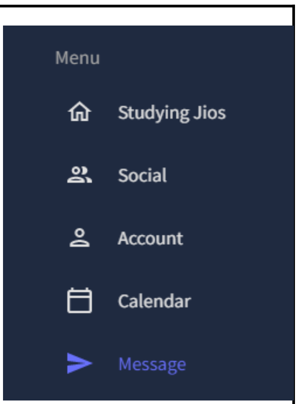
CONTACTS

List of Contacts for Future Reference

ID	Name	Telegram Usernsme	Email	Major
1	Jon Snow	@jon_snow	jonsnow@gmail.com	Biology
2	Cersei Lannister	@cersei_lannister	cerseilannister@gmail.com	Political Science
3	Jaime Lannister	@jaime_lannister	jaimelannister@gmail.com	History
4	Anya Stark	@anya_stark	anyastark@gmail.com	Computer Science
5	Daenerys Targaryen	@daenerys_targaryen	daenerystargaryen@gmail.com	Economics
6	Ever Melisandre	@ever_melisandre	evermelisandre@gmail.com	Psychology
7	Ferrara Clifford	@ferrara_clifford	ferraraclifford@gmail.com	Chemistry
8	Rossini Frances	@rossini_frances	rossinifrances@gmail.com	English Literature

Rows per page: 100 1–11 of 11

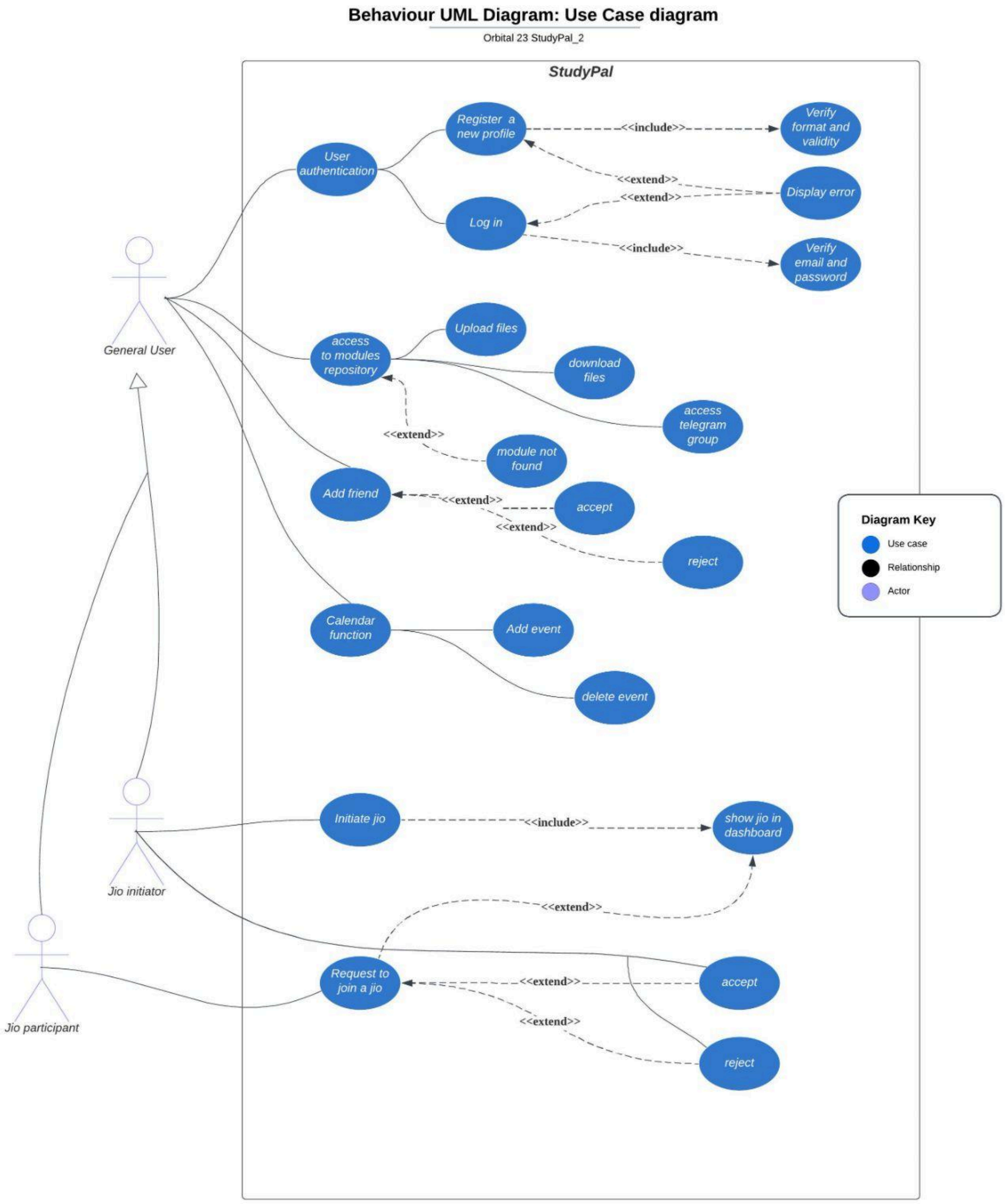
Additional features:

1.	Communication tools: The app will include built-in communication tools such as messaging or video conferencing to allow students to collaborate remotely.	 Menu Studying Jios Social Account Calendar Message
2.	Gamification: The website could include gamification features such as badges, rewards, or leaderboards to incentivize students to use the app and study with their partners.	
3.	Feedback and rating system: The website could include a system for students to rate their study partners and provide feedback on their study sessions, which could help to improve future matches.	

System Design:

- Design diagrams

1. Use case diagram



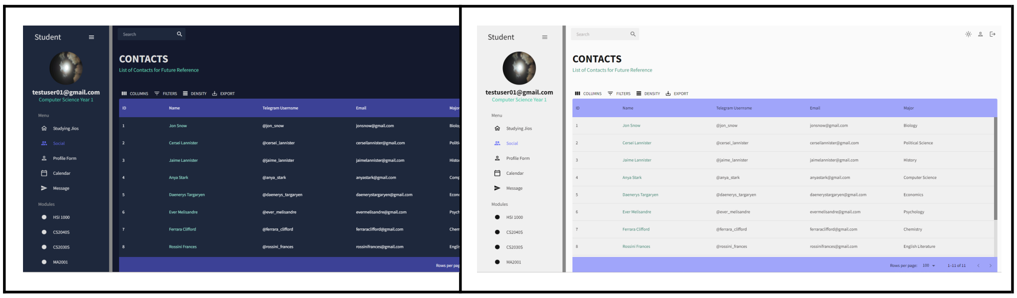
2. Tech Stack:
    a. React (Material-UI, formik, sidebar): frontend control and styling
    b. Html, css & Javascript: frontend control
    c. Firebase: user authentication and firestore database
    d. Github and git: version control
    e. Netlify: Web application deployment
    f. Lucidity: UML diagram drawing

- Design principles

a. User-Centred Design: Prioritise the needs, preferences, and goals of the students at NUS. Conduct user research to understand their requirements and design the platform around their academic experience, making it intuitive and easy to use.

b. Simplicity: Strive for a clean and minimalist design that avoids clutter and complexity. Focus on presenting information in a clear and concise manner, ensuring that users can quickly grasp the platform's features and functionalities.

c. Consistency: Maintain consistent design elements, such as navigation, typography, and colour schemes(dark and light modes), throughout the platform. Consistency helps users develop familiarity and reduces cognitive load when interacting with different sections or features.

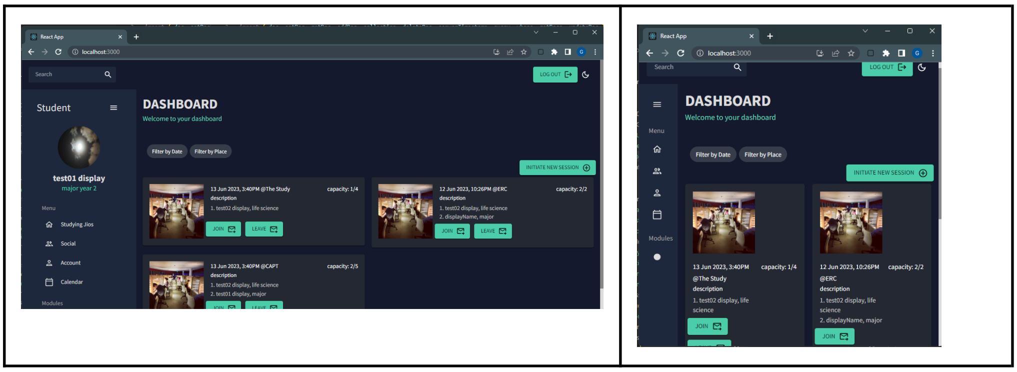


CONTACTS

List of Contacts for Future Reference

ID	Name	Telegram Usernsme	Email	Major
1	Jon Snow	@jon_snow	jonsnow@gmail.com	Biology
2	Cersei Lannister	@cersei_lannister	cerseilannister@gmail.com	Political Science
3	Jaime Lannister	@jaime_lannister	jaimelannister@gmail.com	History
4	Anya Stark	@anya_stark	anyastark@gmail.com	Computer Science
5	Daenerys Targaryen	@daenerys_targaryen	daenerystargaryen@gmail.com	Economics
6	Ever Melisandre	@ever_melisandre	evermelisandre@gmail.com	Psychology
7	Ferrara Clifford	@ferrara_clifford	ferraraclifford@gmail.com	Chemistry
8	Rossini Frances	@rossini_frances	rossinifrances@gmail.com	English Literature

d. Responsive Design: Design the platform to be responsive and accessible across various devices, providing a consistent experience for all users on desktops, laptops, tablets, and smartphones.



DASHBOARD

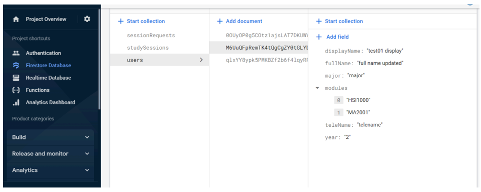
Welcome to your dashboard

13 Jun 2023, 3:40PM @The Study capacity: 1/4
description
1. test02 display, life science

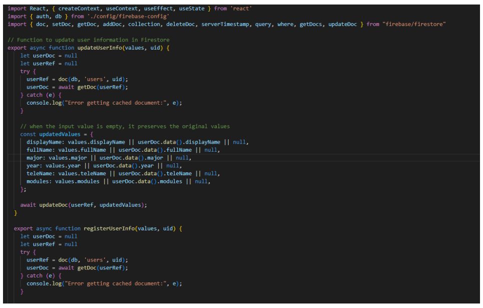
12 Jun 2023, 10:26PM @ERC capacity: 2/2
description
1. test02 display, life science
2. displayName, major

13 Jun 2023, 3:40PM @CAPT capacity: 2/5
description
1. test02 display, life science
2. test01 display, major

e. User Security: User data is protected by the implementation of firebase authentication system and firestore database. The data could only be accessed with the administrator who holds the configuration keys and domain.



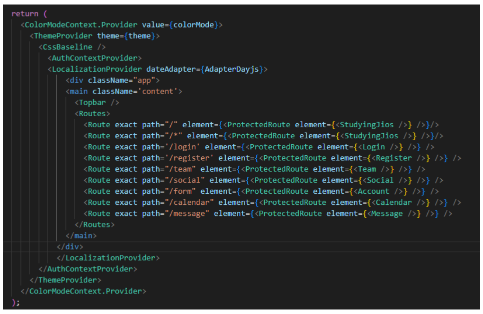
```
sessionRequests
studySessions
users

0OUyOP0g5COtz1ajsLAT7DKUW
M6UuQFpRemTK4tQgCgZY0tGLYt
qlxYY8ypk5PMKBZf2b6f41qyRF

displayName: "test01 display"
fullName: "full name updated"
major: "major"
modules
    0 "HSI1000"
    1 "MA2001"
teleName: "telename"
year: "2"
```



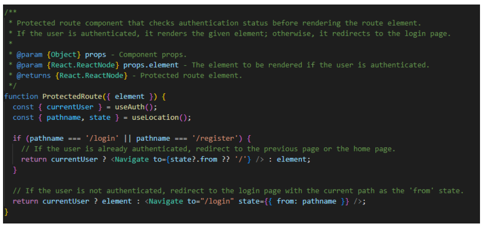
```
import React, { createContext, useContext, useEffect, useState } from 'react'
import { auth, db } from './config/firebase-config'
import { doc, setDoc, getDoc, addDoc, collection, deleteDoc, serverTimestamp, query, where, getDocs, updateDoc } from "firebase/firestore"

// Function to update user information in Firestore
export async function updateUserInfo(values, uid) {
    let userDoc = null
    let userRef = null
    try {
      userRef = doc(db, 'users', uid);
      userDoc = await getDoc(userRef);
    } catch (e) {
      console.log("Error getting cached document:", e);
    }

    // when the input value is empty, it preserves the original values
    const updatedValues = {
      displayName: values.displayName || userDoc.data().displayName || null,
      fullName: values.fullName || userDoc.data().fullName || null,
      major: values.major || userDoc.data().major || null,
      year: values.year || userDoc.data().year || null,
      teleName: values.teleName || userDoc.data().teleName || null,
      modules: values.modules || userDoc.data().modules || null,
    };

    await updateDoc(userRef, updatedValues);
  }

  export async function registerUserInfo(values, uid) {
    let userDoc = null
    let userRef = null
    try {
      userRef = doc(db, 'users', uid);
      userDoc = await getDoc(userRef);
    } catch (e) {
      console.log("Error getting cached document:", e);
    }
```

Console of the project StudyPal on firestore	Firestore operation functions

f. Page Security:

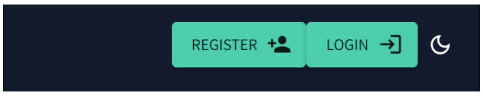


```
return (
  <ColorModeContext.Provider value={colorMode}>
    <ThemeProvider theme={theme}>
      <CssBaseline />
        <AuthContextProvider>
        <LocalizationProvider dateAdapter={AdapterDayjs}>
          <div className="app">
          <main className='content'>
            <Topbar />
            <Routes>
              <Route exact path="/" element={<ProtectedRoute element={<StudyingJios />} />}/>
              <Route exact path="/*" element={<ProtectedRoute element={<StudyingJios />} />}/>
              <Route exact path='/login' element={<ProtectedRoute element={<Login />} />} />
              <Route exact path='/register' element={<ProtectedRoute element={<Register />} />} />
              <Route exact path="/team" element={<ProtectedRoute element={<Team />} />} />
              <Route exact path="/social" element={<ProtectedRoute element={<Social />} />} />
              <Route exact path="/form" element={<ProtectedRoute element={<Account />} />} />
              <Route exact path="/calendar" element={<ProtectedRoute element={<Calendar />} />} />
              <Route exact path="/message" element={<ProtectedRoute element={<Message />} />} />
            </Routes>
          </main>
        </div>
        </LocalizationProvider>
      </AuthContextProvider>
    </ThemeProvider>
  </ColorModeContext.Provider>
);
```



```
/**
 * Protected route component that checks authentication status before rendering the route element.
 * If the user is authenticated, it renders the given element; otherwise, it redirects to the login page.
 *
 * @param {Object} props - Component props.
 * @param {React.ReactNode} props.element - The element to be rendered if the user is authenticated.
 * @returns {React.ReactNode} - Protected route element.
 */
function ProtectedRoute({ element }) {
  const { currentUser } = useAuth();
  const { pathname, state } = useLocation();

  if (pathname === '/login' || pathname === '/register') {
    // If the user is already authenticated, redirect to the previous page or the home page.
    return currentUser ? <Navigate to={state?.from ?? '/'} /> : element;
  }

  // If the user is not authenticated, redirect to the login page with the current path as the 'from' state.
  return currentUser ? element : <Navigate to="/login" state={{ from: pathname }} />;
}
```

Protected route component that checks authentication status before rendering the route element. If the user is authenticated, it renders the given element; otherwise, it redirects to the login page.

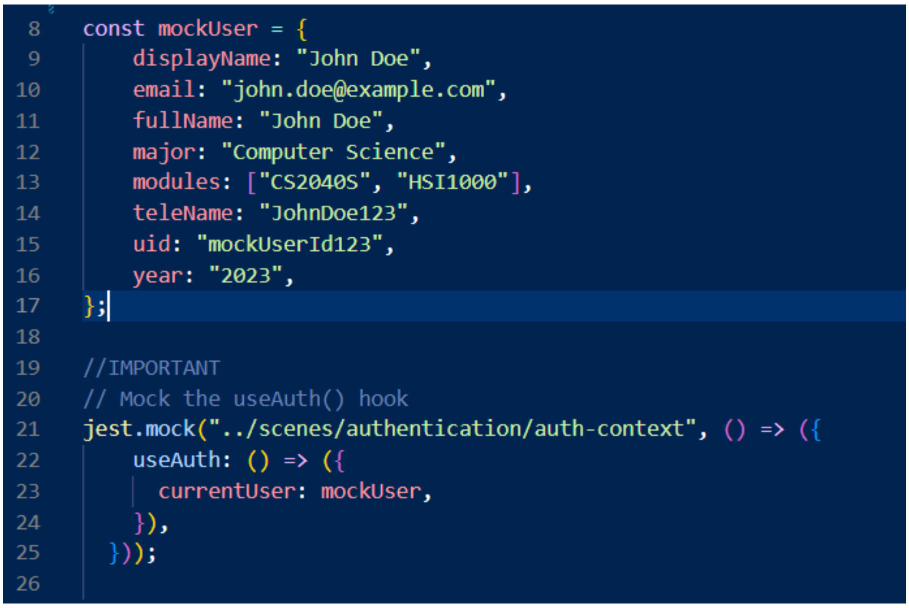
Same goes for top bar, "log out" button is shown when user is authenticated, ""Register" and "Login" buttons are shown otherwise

Evaluation & Testing:

a. **Introduction:** Our project involves testing React components using the Jest testing library, along with the @testing-library/react and @testing-library/user-event packages. The main objective of testing is to ensure the correctness and reliability of the components in various scenarios.

b. **Testing Approach:** The testing approach used in this project is based on unit testing, focusing on testing individual React components in isolation. The primary tool used for testing is Jest, a popular testing framework for JavaScript applications. The @testing-library/react package is employed to render components and interact with them, while @testing-library/user-event is used for simulating user interactions.

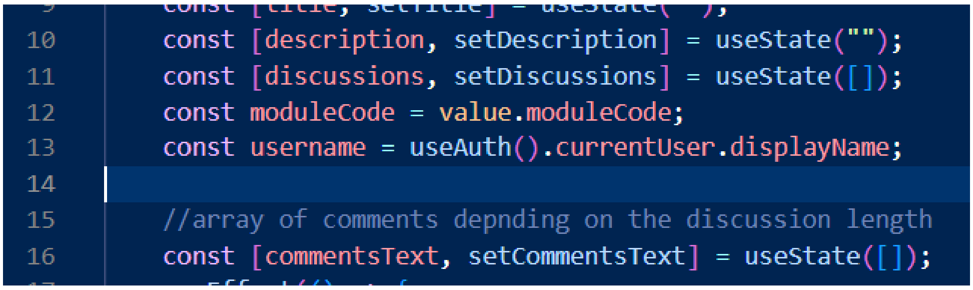
c. **Mock Dependencies:** To ensure accurate and isolated testing of components, certain dependencies need to be mocked. In this project, the `useAuth` hook from the authentication context is an external dependency that is mocked using the Jest jest.mock function. By providing a custom implementation for useAuth, the component can be tested with specific user contexts, allowing for better test coverage.



```
8   const mockUser = {
9       displayName: "John Doe",
10      email: "john.doe@example.com",
11      fullName: "John Doe",
12      major: "Computer Science",
13      modules: ["CS2040S", "HSI1000"],
14      teleName: "JohnDoe123",
15      uid: "mockUserId123",
16      year: "2023",
17  };
18
19  //IMPORTANT
20  // Mock the useAuth() hook
21  jest.mock("../scenes/authentication/auth-context", () => ({
22      useAuth: () => ({
23        currentUser: mockUser,
24      }),
25    }));
26
```

i. At line 8,
During the testing, we will not be able to access to firebase database, so we need to mock user to ensure that the testing can be run properly.The mockUser acts as a user of our project that is going to perform something in our web application.

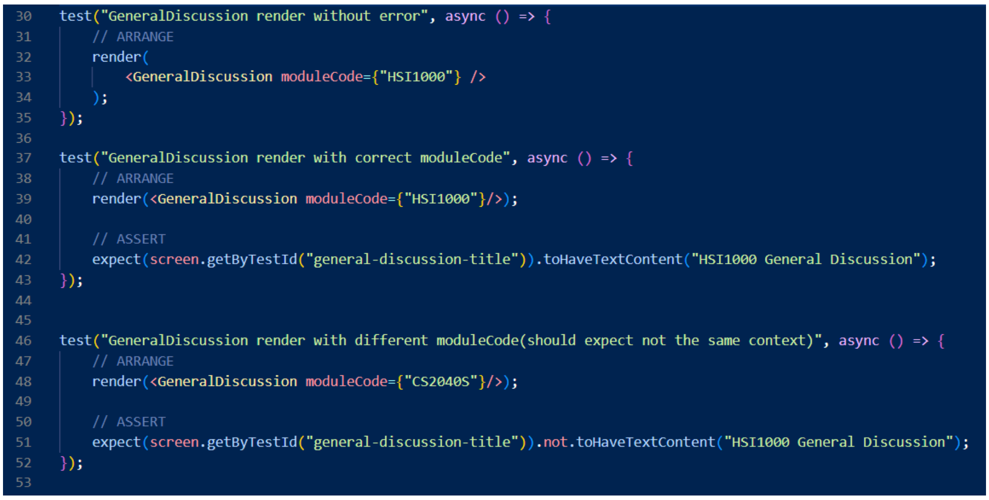
ii. At line 21,
As `useAuth` is not accessible during testing phase as it requires authentication from firebase, we need to mock the useAuth() so that the <GeneralDIscussion /> component can be rendered without error



```
9       const [title, setTitle] = useState("");
10      const [description, setDescription] = useState("");
11      const [discussions, setDiscussions] = useState([]);
12      const moduleCode = value.moduleCode;
13      const username = useAuth().currentUser.displayName;
14
15      //array of comments depnding on the discussion length
16      const [commentsText, setCommentsText] = useState([]);
17        useEffect(() => {
```

Partial code from GeneralDiscussion.jsx:
In line 13, if we did not mock the useAuth() function, the `currentUser` variable will be a `null` value and it cannot be tested properly

d. **Test Suite Structure:** The test suite is organized into multiple test cases, each targeting different aspects of the GeneralDiscussion component. The test cases include:



```
test("GeneralDiscussion render without error", async () => {
    // ARRANGE
    render(
        <GeneralDiscussion moduleCode={"HSI1000"} />
    );
});

test("GeneralDiscussion render with correct moduleCode", async () => {
    // ARRANGE
    render(<GeneralDiscussion moduleCode={"HSI1000"}/>);

    // ASSERT
    expect(screen.getByTestId("general-discussion-title")).toHaveTextContent("HSI1000 General Discussion");
});


test("GeneralDiscussion render with different moduleCode(should expect not the same context)", async () => {
    // ARRANGE
    render(<GeneralDiscussion moduleCode={"CS2040S"}/>);

    // ASSERT
    expect(screen.getByTestId("general-discussion-title")).not.toHaveTextContent("HSI1000 General Discussion");
});

```

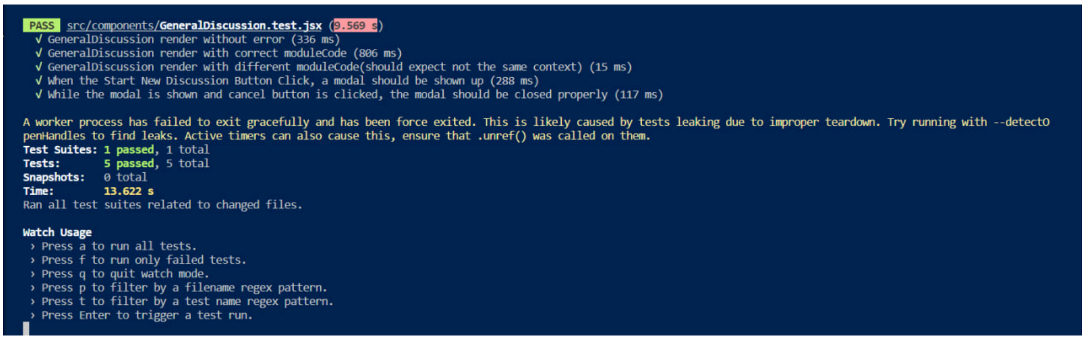
i. GeneralDiscussion render without error: This test case ensures that the GeneralDiscussion component can be rendered without any errors. It is a basic check to verify if the component is correctly rendered.

ii. GeneralDiscussion render with correct moduleCode: This test case checks whether the GeneralDiscussion component displays the correct title based on the provided moduleCode. It helps ensure that the component behaves as expected when rendering different discussions.

iii. GeneralDiscussion render with different moduleCode (except not the same context):This test case verifies that the context changes appropriately when rendering the GeneralDiscussion component with different moduleCode. It ensures that the component's behaviour is dependent on the provided moduleCode.

iv. When the Start New Discussion Button is Clicked, a modal should be shown up: This test case simulates a user clicking the "Start New Discussion" button and verifies whether the corresponding modal is displayed as expected.

v. While the modal is shown and the cancel button is clicked, the modal should be closed properly: This test case confirms that when the modal is visible and the user clicks the "Cancel" button, the modal is closed correctly and is no longer present in the DOM.

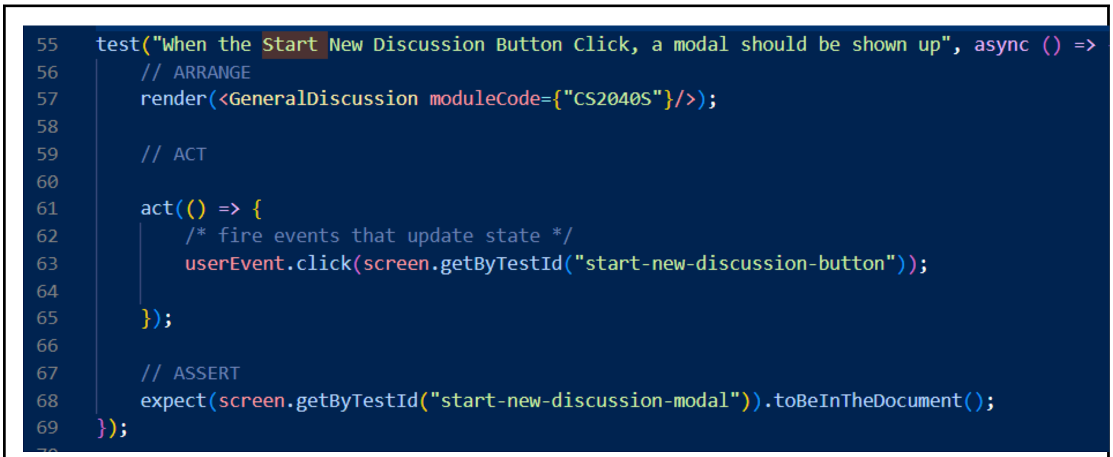


```
 PASS  src/components/GeneralDiscussion.test.jsx (9.569 s)
  √ GeneralDiscussion render without error (336 ms)
  √ GeneralDiscussion render with correct moduleCode (806 ms)
  √ GeneralDiscussion render with different moduleCode(should expect not the same context) (15 ms)
  √ When the Start New Discussion Button Click, a modal should be shown up (288 ms)
  √ While the modal is shown and cancel button is clicked, the modal should be closed properly (117 ms)

A worker process has failed to exit gracefully and has been force exited. This is likely caused by tests leaking due to improper teardown. Try running with --detectO
penHandles to find leaks. Active timers can also cause this, ensure that .unref() was called on them.
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        13.622 s
Ran all test suites related to changed files.

Watch Usage
 › Press a to run all tests.
 › Press f to run only failed tests.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

Successful test for the component GeneralDiscussion

Similar approaches applied to other components as well.

e. Test Implementation



```
test("When the Start New Discussion Button Click, a modal should be shown up", async () =>
    // ARRANGE
    render(<GeneralDiscussion moduleCode={"CS2040S"}/>);

    // ACT

    act(() => {
        /* fire events that update state */
        userEvent.click(screen.getByTestId("start-new-discussion-button"));

    });

    // ASSERT
    expect(screen.getByTestId("start-new-discussion-modal")).toBeInTheDocument();
});
```

For each test case, the following steps are generally followed:
i . Arrange: The test case begins by rendering the GeneralDiscussion component with the necessary props and configurations. Any initial setup required for the specific test is also performed in this step.

ii. Act: This step involves simulating user interactions or any actions that trigger state updates in the component. In some cases, it may involve waiting for asynchronous operations to complete.

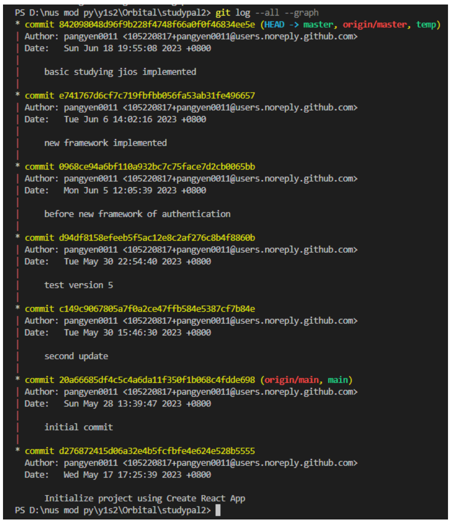
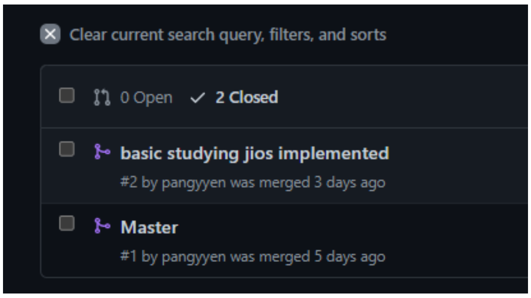
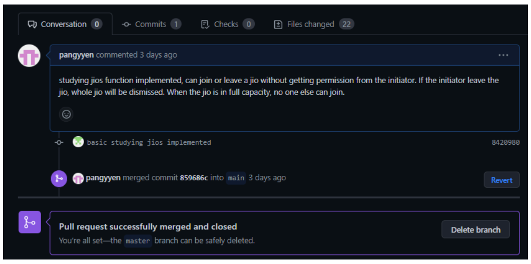
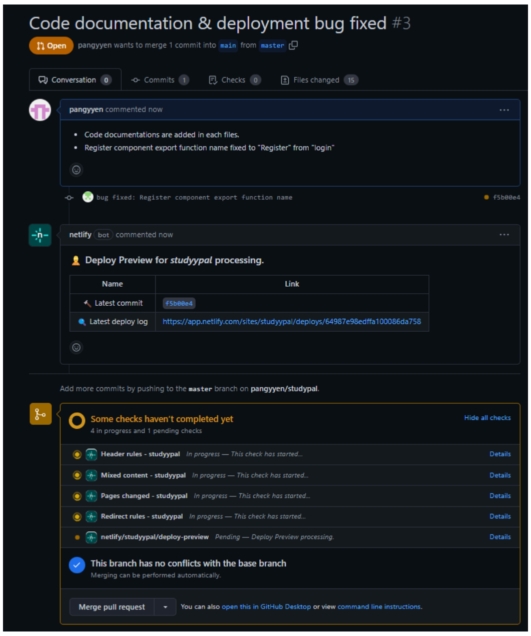
iii. Assert: The final step is to make assertions using Jest's expected statements. The @testing-library/react package is utilised to query the component's rendered output and verify if it meets the expected conditions.

f. **Conclusion:** By following a well-structured testing approach and writing test cases that cover different scenarios, the project ensures that the components in our project functions correctly and behaves as expected. Regularly running these tests as part of the development process helps catch and fix issues early, ensuring the stability and reliability of the project as a whole. Additionally, the use of mocked dependencies enables isolated testing, making it easier to identify specific sources of potential failures.

Software Engineering Practices

1. Planning and Project Management: Project is broken down into several phases and each consists of multiple smaller tasks. A development plan and timeline is created to organise the whole workflow of the developing process. (development plan shown in next section)

2. Version Control: Version control system is handled by adapting Git and github to track changes to the codebase. This allows collaboration between teammates, reversion to previous versions if needed, and management of different branches of development.

	  
Example of using Git in local computer to track and monitor major checkpoints throughout development process	Merges and pull requests are created on github when a major checkpoint is reached.

```
PS D:\nus mod py\y1s2\Orbital\studypal2> git log --all --graph
* commit 842098048d96f9b228f4748f66a0f0f46834ee5e (HEAD -> master, origin/master, temp)
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Sun Jun 18 19:55:08 2023 +0800
|
|     basic studying jios implemented
|
* commit e741767d6cf7c719fbfbb056fa53ab31fe496657
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Tue Jun 6 14:02:16 2023 +0800
|
|     new framework implemented
|
* commit 0968ce94a6bf110a932bc7c75face7d2cb0065bb
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Mon Jun 5 12:05:39 2023 +0800
|
|     before new framework of authentication
|
* commit d94df8158efeeb5f5ac12e8c2af276c8b4f8860b
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Tue May 30 22:54:40 2023 +0800
|
|     test version 5
|
* commit c149c9067805a7f0a2ce47ffb584e5387cf7b84e
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Tue May 30 15:46:30 2023 +0800
|
|     second update
|
* commit 20a66685df4c5c4a6da11f350f1b068c4fdde698 (origin/main, main)
| Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
| Date:   Sun May 28 13:39:47 2023 +0800
|
|     initial commit
|
* commit d276872415d06a32e4b5fcfbfe4e624e528b5555
  Author: pangyen0011 <105220817+pangyen0011@users.noreply.github.com>
  Date:   Wed May 17 17:25:39 2023 +0800

      Initialize project using Create React App
PS D:\nus mod py\y1s2\Orbital\studypal2> █
```

☒ Clear current search query, filters, and sorts

□ 0 Open ✓ 2 Closed

□ **basic studying jios implemented**
#2 by pangyyen was merged 3 days ago

□ **Master**
#1 by pangyyen was merged 5 days ago

Conversation 0 · Commits 1 · Checks 0 · Files changed 22

pangyyen commented 3 days ago

studying jios function implemented, can join or leave a jio without getting permission from the initiator. If the initiator leave the jio, whole jio will be dismissed. When the jio is in full capacity, no one else can join.

`basic studying jios implemented` 8420980

pangyyen merged commit **859686c** into `main` 3 days ago — Revert

Pull request successfully merged and closed
You're all set—the `master` branch can be safely deleted. — Delete branch

Code documentation & deployment bug fixed #3

Open — pangyyen wants to merge 1 commit into `main` from `master`

Conversation 0 · Commits 1 · Checks 0 · Files changed 15

pangyyen commented now

- Code documentations are added in each files.
- Register component export function name fixed to "Register" from "login"

`bug fixed: Register component export function name` f5b00e4

netlify bot commented now

Deploy Preview for *studyypal* processing.

Name	Link
Latest commit	f5b00e4
Latest deploy log	https://app.netlify.com/sites/studyypal/deploys/64987e98edffa100086da758

Add more commits by pushing to the **master** branch on **pangyyen/studypal**.

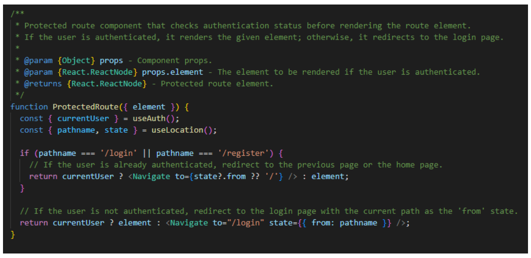
Some checks haven't completed yet — Hide all checks
4 in progress and 1 pending checks

- **Header rules - studyypal** *In progress — This check has started...* Details
- **Mixed content - studyypal** *In progress — This check has started...* Details
- **Pages changed - studyypal** *In progress — This check has started...* Details
- **Redirect rules - studyypal** *In progress — This check has started...* Details
- **netlify/studyypal/deploy-preview** *Pending — Deploy Preview processing.* Details

✓ **This branch has no conflicts with the base branch**
Merging can be performed automatically.

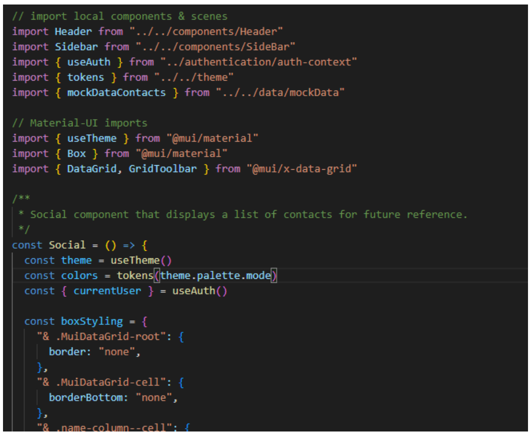
Merge pull request ▾ — You can also open this in GitHub Desktop or view command line instructions.

3. Code Documentation: Inline comments and ReadMe files Document your code thoroughly to make it easier for others (including your future self) to understand and maintain it.



```
/**
 * Protected route component that checks authentication status before rendering the route element.
 * If the user is authenticated, it renders the given element; otherwise, it redirects to the login page.
 *
 * @param {Object} props - Component props.
 * @param {React.ReactNode} props.element - The element to be rendered if the user is authenticated.
 * @returns {React.ReactNode} - Protected route element.
 */
function ProtectedRoute({ element }) {
  const { currentUser } = useAuth();
  const { pathname, state } = useLocation();

  if (pathname === '/login' || pathname === '/register') {
    // If the user is already authenticated, redirect to the previous page or the home page.
    return currentUser ? <Navigate to={state?.from ?? '/'} /> : element;
  }

  // If the user is not authenticated, redirect to the login page with the current path as the 'from' state.
  return currentUser ? element : <Navigate to="/login" state={{ from: pathname }} />;
}
```

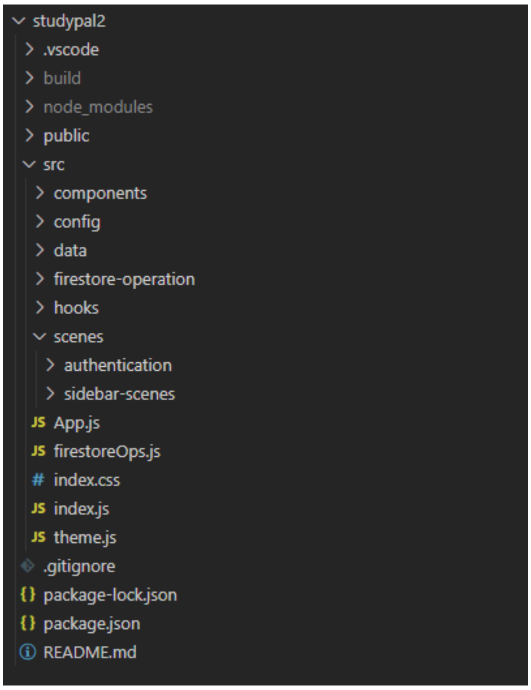
Functions are well explain



```
// import local components & scenes
import Header from "../../components/Header"
import Sidebar from "../../components/SideBar"
import { useAuth } from "../authentication/auth-context"
import { tokens } from "../../theme"
import { mockDataContacts } from "../../data/mockData"

// Material-UI imports
import { useTheme } from "@mui/material"
import { Box } from "@mui/material"
import { DataGrid, GridToolbar } from "@mui/x-data-grid"

/**
 * Social component that displays a list of contacts for future reference.
 */
const Social = () => {
  const theme = useTheme()
  const colors = tokens(theme.palette.mode)
  const { currentUser } = useAuth()

  const boxStyling = {
    "& .MuiDataGrid-root": {
      border: "none",
    },
    "& .MuiDataGrid-cell": {
      borderBottom: "none",
    },
    "& .name-column--cell": {
```

Each component is documented properly

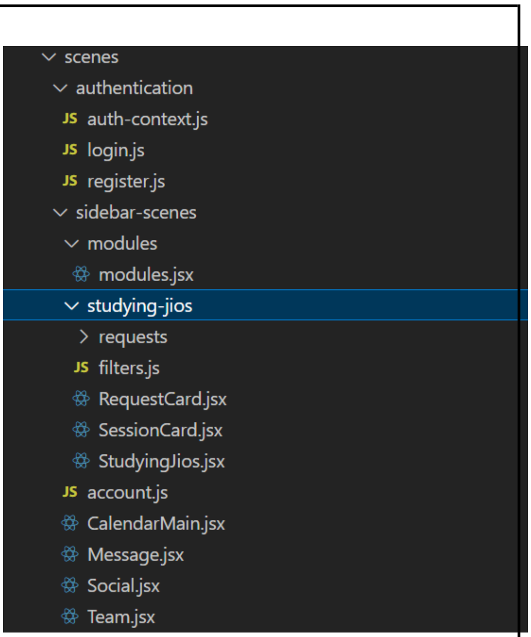
4. File organisation: Logical organisation structure is followed by separating different components of the project into distinct files and folders. This practice improves code maintainability and scalability. Each file was dedicated to a specific module or functionality, making it easier to locate and modify relevant code. By compartmentalising code into smaller units, code reusability is fostered and this ensured that changes made to one component did not impact others unnecessarily.



```
studypal2
  .vscode
  build
  node_modules
  public
  src
    components
    config
    data
    firestore-operation
    hooks
    scenes
      authentication
      sidebar-scenes
    App.js
    firestoreOps.js
    index.css
    index.js
    theme.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

The overall structure of the project:

- components : global components like topbar and sidebar, and reusable module



```
scenes
  authentication
    auth-context.js
    login.js
    register.js
  sidebar-scenes
    modules
      modules.jsx
    studying-jios
      requests
      filters.js
      RequestCard.jsx
      SessionCard.jsx
      StudyingJios.jsx
    account.js
    CalendarMain.jsx
    Message.jsx
    Social.jsx
    Team.jsx
```

Scenes are further separated into sidebar scenes and authentication.

- Sidebar scenes: most of the feature pages are accessed via

like Header - Data: to store mock data - Firestore-operation: globally usable operation - Hooks: globally usable hook, named under the context React - Scenes: components organised according to their appearance in respective scenes	a global sidebar, thus organised in this folder. For “Modules” and “Studying Jios” which are more complicated and consist of more sub-functions, helper functions and sub-components, they are further abstracted into more files within respective folders. - Authentication: Authentication functions are implemented via firebase, similar logic and API invocation are abstracted and bundled together, and passed down to each component with useContext hook. Immediate usage of those functions are seen in login and register pages, thus placed under same folder

Development timeline:

S/N	Tasks	Description	In Charge	Date
1.	Finalising ideas	Finalising user stories and corresponding features Sketching a wireframe and designed a user interface Poster and video design	Pang Yen Javen	6 - 15 May
Lift-off				15 May
2.	Crafting initial framework, Testing out NUSMOD API	HTML, CSS, javascript learning Finalising and learning of framework for UX/UI – React Setting up skeleton of the website	Pang Yen	15 - 24 May
		HTML, CSS, javascript, git & github learning Data retrieval from NUSMOD to provide technical proof	Javen	
3	User account Authentication Modules data retrieval and hosting	Firebase User authentication, login & register pages setup, Milestone 1 documentation	Pang Yen	24 - 29 May
		Uploading data to cloud, then hosting retrieved data in Glitch Milestone 1 documentation	Javen	
Milestone 1: Ideation				29 May
4.	CRUD operations for user information and modules enrolled UI improvement	User profile update during registration Online profile update	Pang Yen	30 May - 12 June
		UI improvement	Javen	
5.	Studying Jios	Studying jio initiation, joining and leaving	Pang Yen	12 - 20 Jun

	Calendar & Reminder function Modules repository	Link external calendar and initiated or participating jios to calendar		
		Dynamic modules pages	Javen	
6.	Testing & Evaluation setup Telegram group link Documentation	Testing & Evaluation setup using Jest and React testing library Documentation for milestone 2	Pang Yen	20 - 26 Jun
		Telegram Group link embedment in module repository Documentation for milestone 2	Javen	
Milestone 2: prototyping				26 Jun
7.	Request & accept component and functionality Discussion thread	Social Function, jio joining requests	Pang Yen	26 Jun - 8 July
		Discussion thread under each module page	Javen	
8.	Recommendation algorithm Feedback for other users	Matching algorithm, prioritise most relevant jio	Pang Yen	8 - 20 July
		Feedback functions	Javen	
9.	UI refinement and testing Documentation	UI refinement and testing Documentation for milestone 3	Pang Yen	20 - 24 July
			Javen	
Milestone 3: Extension				24 July