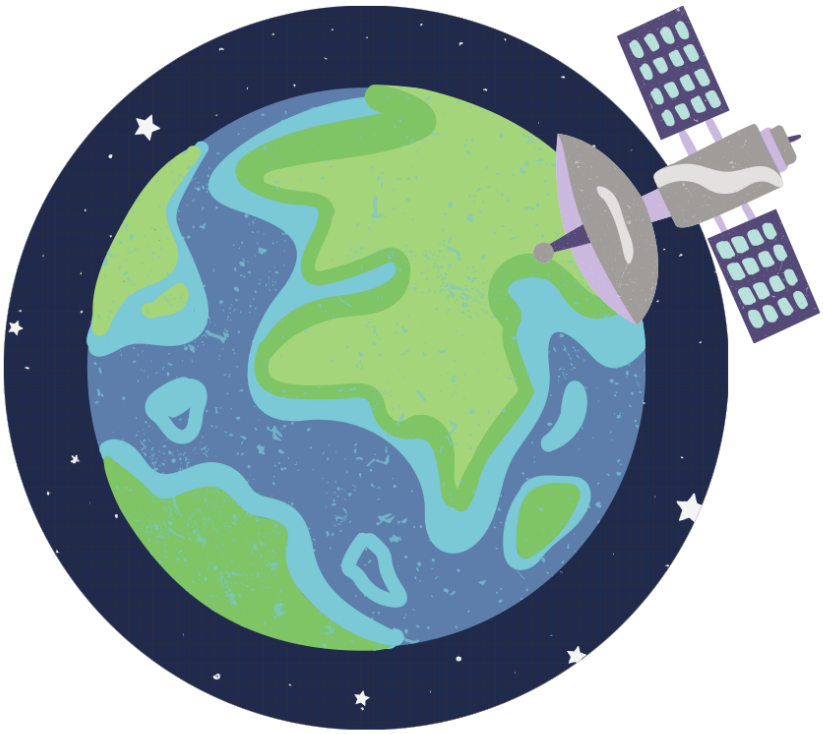
ORBITAL
SKYLAB

AN IMPROVED VERSION!



TEAM INTERNS@ORBITAL
ANG JUN KANG & NG SIMIN
TEAM ADVISOR: RAYNER

Content

Section 1: Introduction **4**
1.1 Motivation 4
1.2 Aim 6
1.3 User Stories 6
1.4 Poster 6
1.5 Video 6
1.6 Code Base 6
1.7 Try it out! 7

Section 2: Features **8**
2.0 User Groups 8
2.1 [All] Main Intro Page 8
2.2 [All] Projects Display Page 9
2.3 [All] Staffs Display Page 10
2.4 [Admin] Dashboard 11
2.5 [Admin] Students Info Page 12
2.6 [Admin] Projects Info Page 14
2.7 [Admin] Submissions Info Page 15
2.8 [Student] Project Profile Page 16
2.9 [Student] Submissions Page 16
2.10 [Student] Student Profile Page 17
2.11 Responsive Design 18

Section 3: Execution **19**
3.1 Development Timeline 19
3.2 Tech Stack & Tools 20
3.3 System Architecture Overview 22
3.4 Entity-Relation Diagram 22
3.5 Git Workflow 23
3.5.1 Source Version Control 23
3.5.2 Workflow 23
3.5.3 Issue Tracking 24
3.6 System Testing 25
3.6.1 Unit Testing 25
3.6.2 Backend Testing 25
3.6.3 Responsiveness Testing 26

3.7 Usability Testing 27
3.7.1 Self evaluation using heuristic evaluation 27
3.7.2 Cognitive walkthrough 29
3.7.3 Actual user focus group 30
3.8 Project Log 31

Section 4: Reflections 32
4.1 Limitations 32
4.2 Future Considerations 32
4.3 Acknowledgements 32

Section 1: Introduction

1.1 Motivation

Orbital is a trademark programme of NUS School of Computing. It provides an opportunity for participants to self-learn software development skills and build their own projects. Yet, for such a prestigious programme, its own platform, Skylab, is not very well designed.

Right from the start, we, as participants of Orbital, experienced great inconvenience because Skylab's **functions are not very comprehensive**. This led to **numerous other platforms being involved and navigating through them all was a pain**. We were first introduced to Piazza, which we used for Q&A and to find groupmates. This platform later became irrelevant as we shifted to Microsoft Teams, which most people rarely checked. This led to advisors starting their own Telegram channels to blast messages. Announcements were also made through Outlook / Gmail, but they were often lost in a sea of other mails. Sometimes, just to get a certain information, such as a dateline or a link, we have to first find the relevant email, click on the attached Google Slides, and then scroll through tens of slides to dig out what we want. It really should not be this hard. Also, Orbital has a Facebook page, but most participants would have forgotten about it by now. Having experienced all of the above ourselves, we want to **unify the functions of the different platforms to create an improved Skylab** so that future Orbital participants will not need to open more than one tab on their browser.

Speaking of Skylab, have we already mentioned how **poorly designed** it is? Crudely put, it is ugly in many aspects and not the most intuitive to use. Also, its loading speed is slow and irritating. We deserve better. We want to provide a **better UI** and a **smoother user experience**.

Ultimately, this is a project from Orbital participants for Orbital participants. We hope that the Orbital team will consider adopting our end product so that our juniors will have a better experience in the future.

Orbital sucks.
We want better!

Our Orbital experience thus far...

Hasn't it been frustrating navigating between numerous platforms and scrolling through tens of slides just to find a single piece of information you need?

Annoyed by Skylab, its ridiculous UX/UI, and its terrible loading speed?

We get you. And we think that we deserve better. Much better.

Our goal:

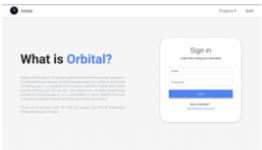
To create a **simple and user-friendly** website to provide an **all-in-one Orbital experience.**

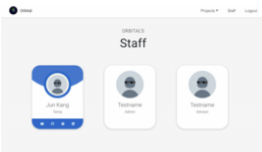
Test ahead!
staff account: orbital@u.nus.edu
student account: angjunkang@u.nus.edu
password: P@ssw0rd

Features

All (including page visitors)

- Main Intro Page
- Log in / log out / sign up
- Projects Display Pages
- Staff Intro Page





Student

- Project Profile Page
- Student Submissions
- Student Profile Page

Admin

- Dashboard
- Users Management
- Projects Management
- Submissions Management

Evaluation

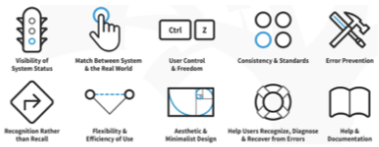
System Testing

React Testing Library

Postman

Chrome DevTools

User Testing



Version Control

Git & Github

Trello

Conclusion

Limitations

- We were unable to implement responsive design to cater to all screen sizes (i.e.,screen sizes with width < 375px).
- Loading speed can be slow at times as we are using a free hosting plan from Heroku.

Future Considerations

- Include displays for past year projects.
- Implement a work bin / forum.

BY TEAM INTERNS@ORBITAL
JUN KANG AND SIMIN

1.2 Aim

To create a **simple and user-friendly website** to provide an **all-in-one Orbital experience**.

1.3 User Stories

Users: Student, Admin, Advisor, Mentor, Tutor

1. As a student, I want to access all of Orbital's information and resources on a single web platform.
2. As a student, I want a forum that is user-friendly to discuss relevant Orbital information.
3. As a student, I want to read about the experiences of students who have taken the modules and view past projects.
4. As a student, I want to find someone to partner with using the forum on the website.
5. As a student, I want to submit documents using the same platform without having to use another website to settle additional submissions.
6. As a student, I want to give and receive feedback on Orbital related projects.
7. As a mentor / advisor, I want to view the documents that my mentees have submitted and provide them comments through the chat function.
8. As a facilitator, I want to moderate the users, forum and other possible features on the platform.

1.4 Poster

https://drive.google.com/file/d/16vjRbVzYGFaiYo1tnv240MFu-pIHTwgh/view?usp=sharing

1.5 Video

https://drive.google.com/file/d/1XhJsgamoO8uHr0WZC5yqNdZQOodsJF3s/view?usp=sharing

1.6 Code Base

https://github.com/lrminrics/orbital-project

1.7 Try it out!

1. Head to https://orbital-skylab.herokuapp.com/
2. Test the admin features using the testing staff account provided:
    a. Email: orbital@u.nus.edu
    b. Password: P@ssw0rd

    or
3. Test the user features by either:
    a. Use the testing user account provided:
        i. Email: angjunkang@u.nus.edu
        ii. Password: P@ssw0rd

        or
    b. Create your own user account:
        i. Log in using the testing staff account and head to the users section
        ii. Proceed to add user with the plus button beside the search bar and perform a single add
        iii. Add your credentials in with your valid NUS email to the system
        iv. Once done, head to the login page and activate your account using the OTP provided.
4. If there is no existing data in the database, download the utils file in the repository and use the respective csv files to upload/generate new data.
5. Test ahead!

Note: If you are planning to test out the user team and submission features on your newly created account, remember to add yourself into a team using the admin account provided.

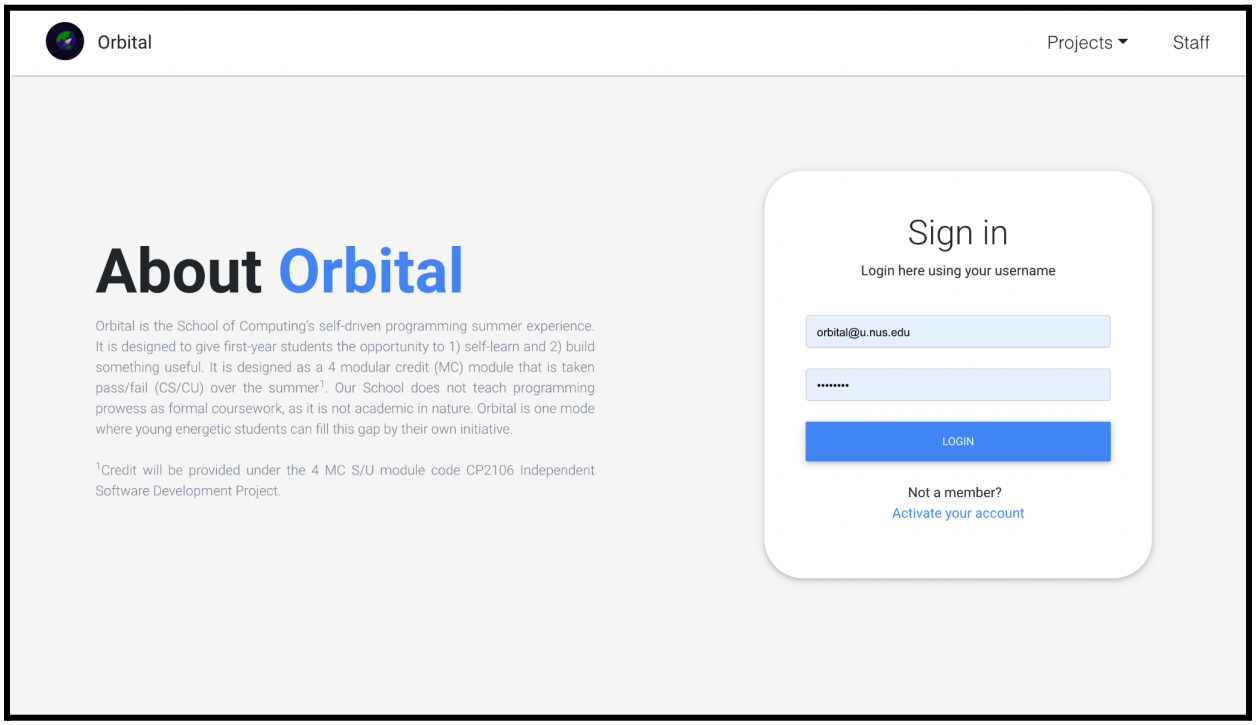
Section 2: Features

2.0 User Groups

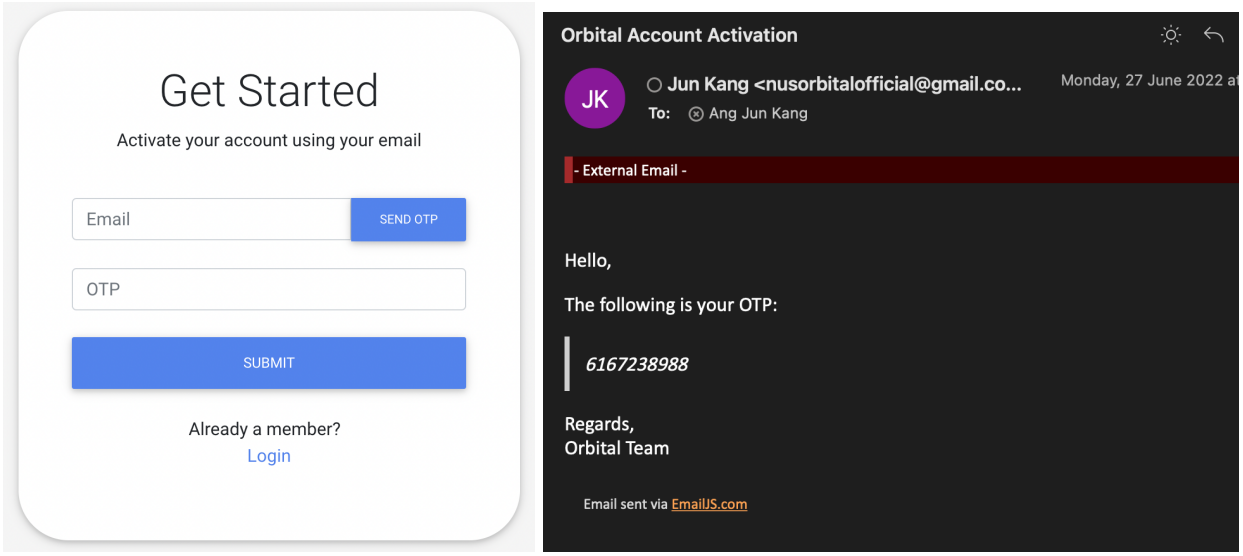
The square brackets indicate which user group the feature is meant for.

User group	Consists of
All	Everyone including page visitors and non-orbital persons.
Student	All Orbital student participants.
Admin	Orbital admin (i.e., facilitator).

2.1 [All] Main Intro Page



At the top of the page is the navigation bar which users can use to navigate to different sites. On the left of the page is a short introduction paragraph about Orbital. On the right, existing users can log into their Skylab account by entering their email and password.



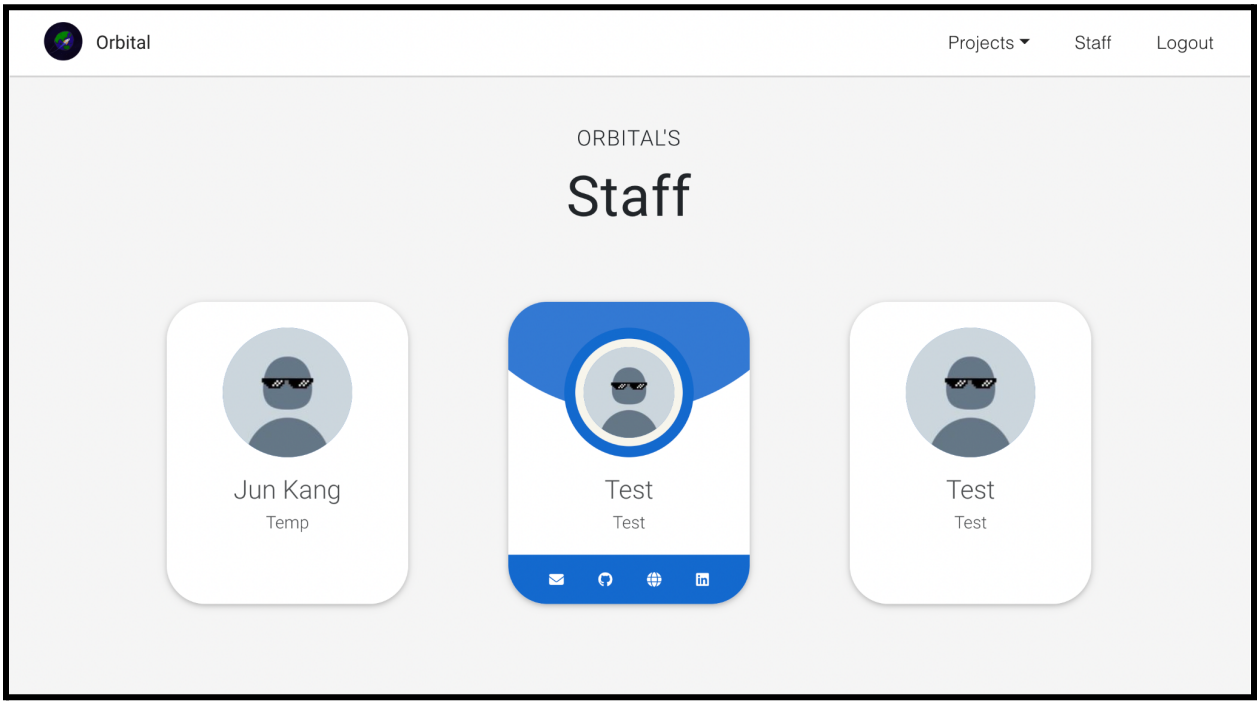
New users can activate their account by getting a one time pass using their emails (refer to section 1.7 for detailed steps).

2.2 [All] Projects Display Page

There are four different projects display pages, one for each achievement level. Every orbital team will have a project card that displays their team name, team ID, team members, and team advisor.

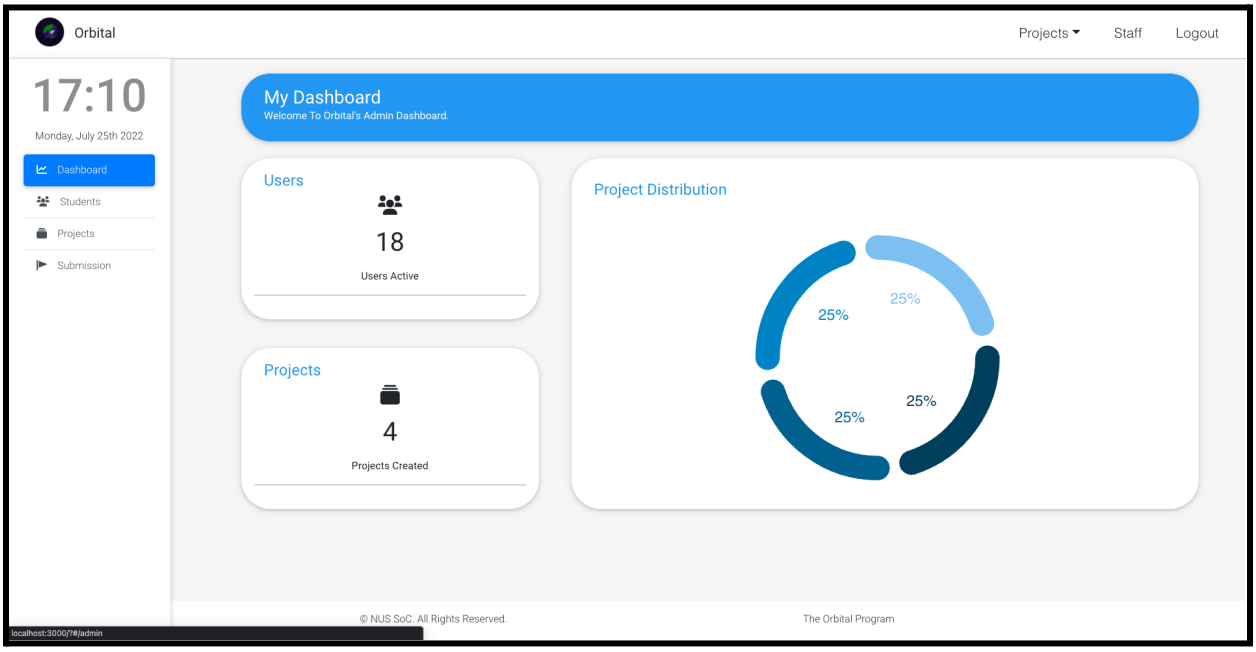
The project card will also include links to the team's project poster and project video if they have submitted the items.

2.3 [All] Staffs Display Page



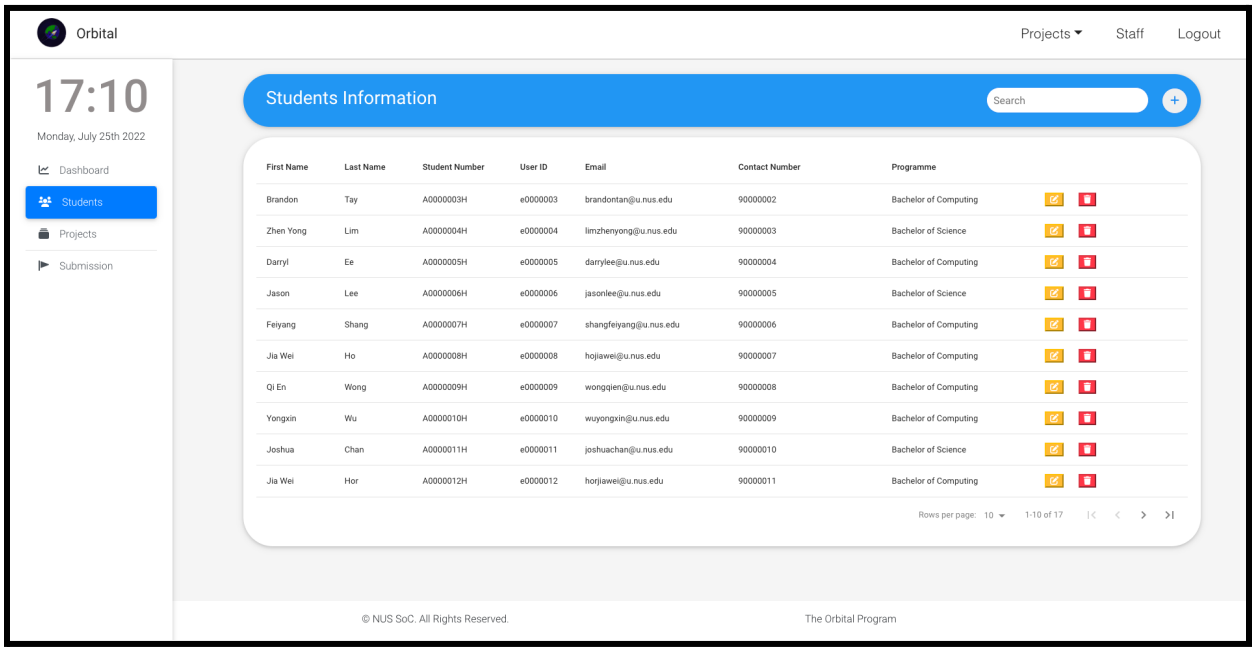
On this page, each staff member will have a staff card that displays his or her name, staff position, and photo. Users can click on the relevant icons at the bottom of each card to be linked to the respective pages. Clicking on the email icon creates a draft email to the selected staff. Clicking on the Github icon brings the user to the staff's Github account. The globe icon links to the staff's personal website and the Linkedin icon opens the staff's Linkedin page.

2.4 [Admin] Dashboard

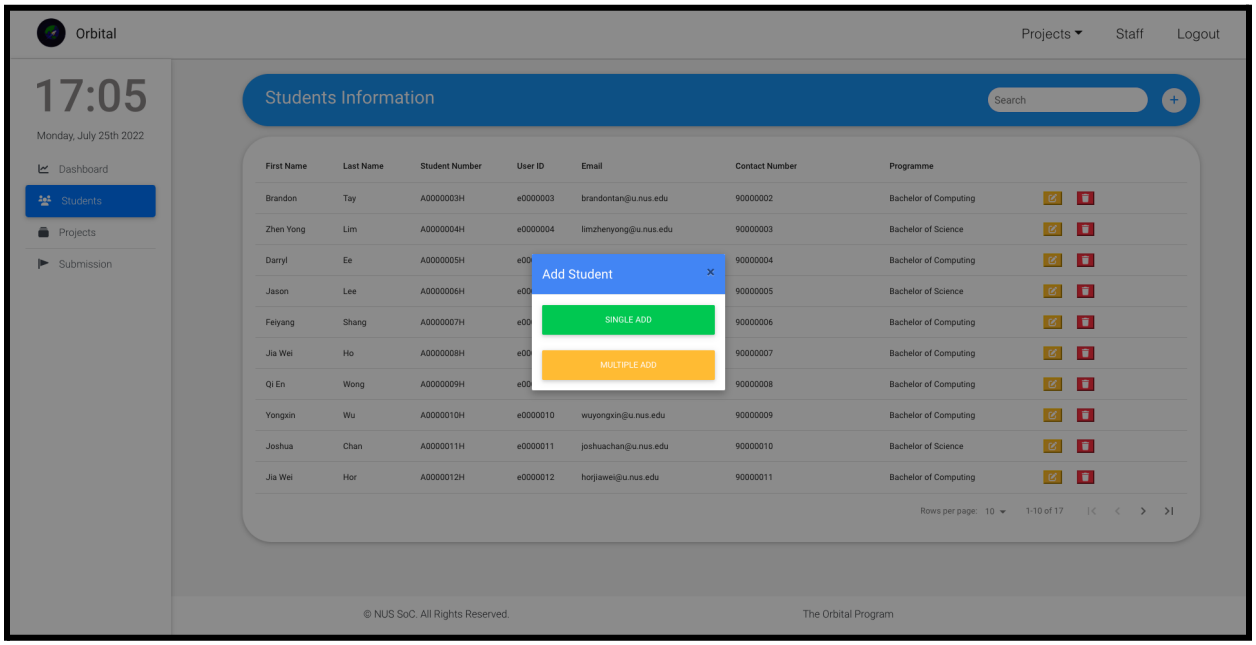


The admin dashboard can only be accessed by the Orbital Admin. It gives an overview on the number of users, the number of projects, and the achievement distribution of the projects. On the left of the page is a navigation sidebar that can be used to navigate to different pages of the admin account.

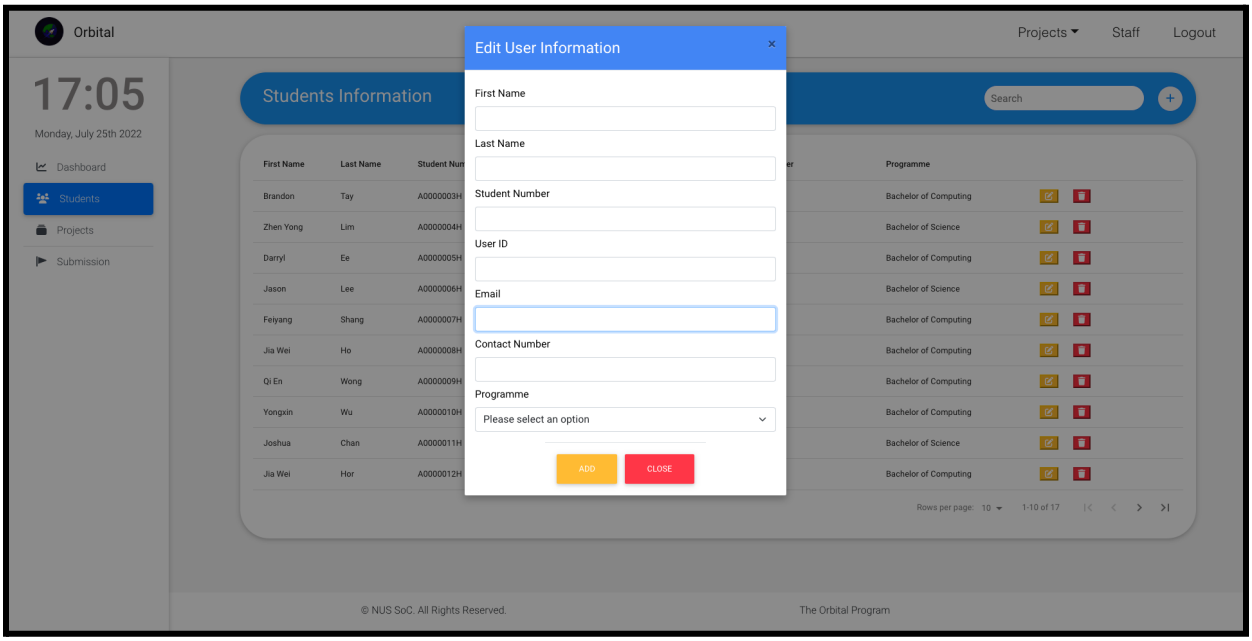
2.5 [Admin] Students Info Page



This page displays the details of all registered Orbital Students. There is a search bar that can be used to narrow down any student. Additionally, the page can be customised to show up to 10, 15, 20, 25, 30, or all users on the same page.

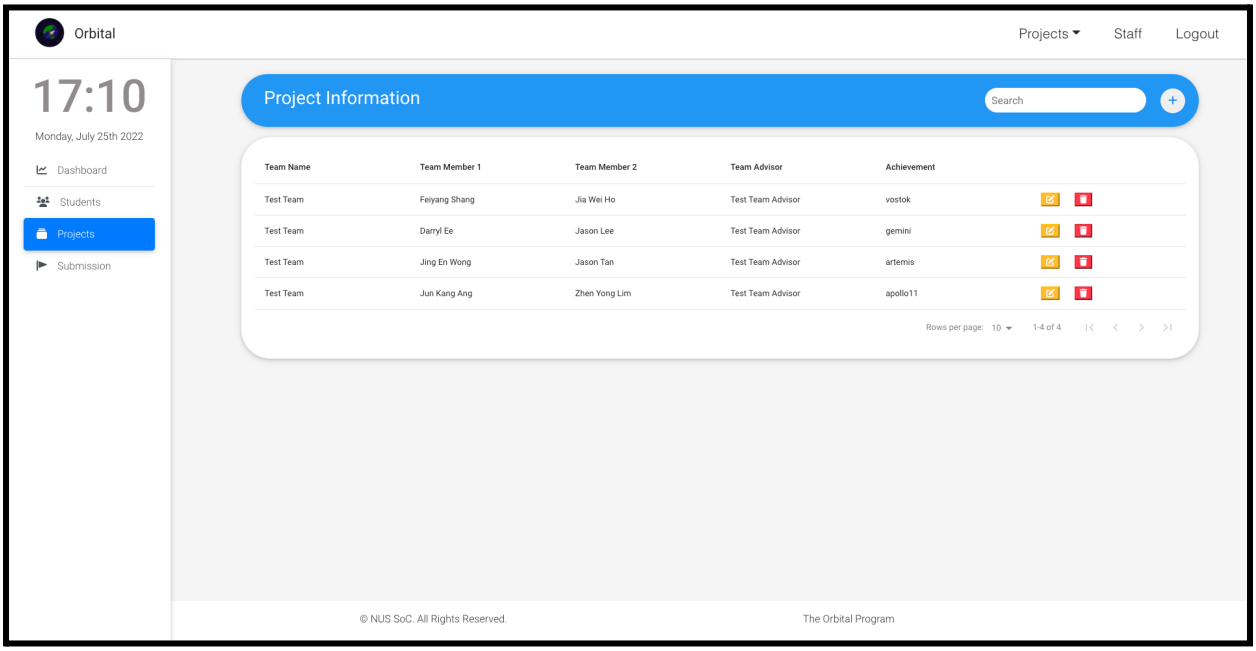


Student details can be mass uploaded through importing an excel file containing the relevant data. Alternatively, in the event that a student needs to be added individually, the Orbital admin can do so by entering the data of that student manually.

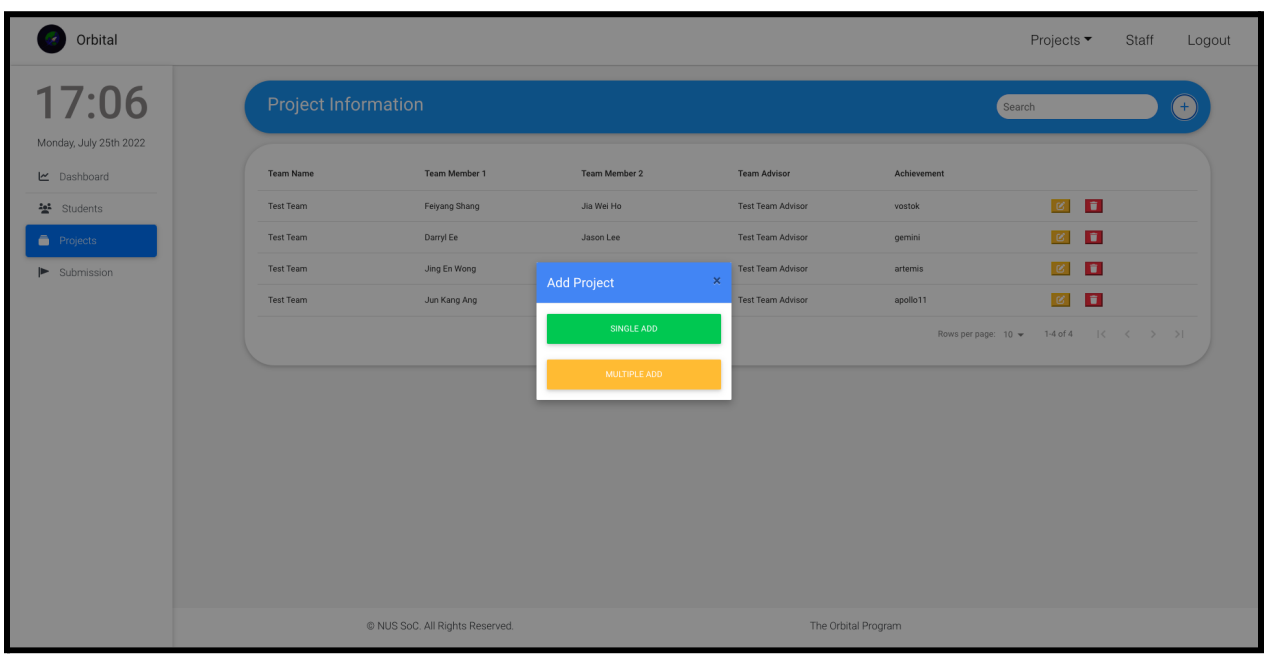


Once added, the information of an individual student can be edited by clicking on the orange / yellow edit button at the relevant row. A pop-up window will appear and the information can be edited through it. The Orbital admin can also remove a student by clicking on the red bin button at the relevant row. A pop-up window will appear to reconfirm the decision.

2.6 [Admin] Projects Info Page



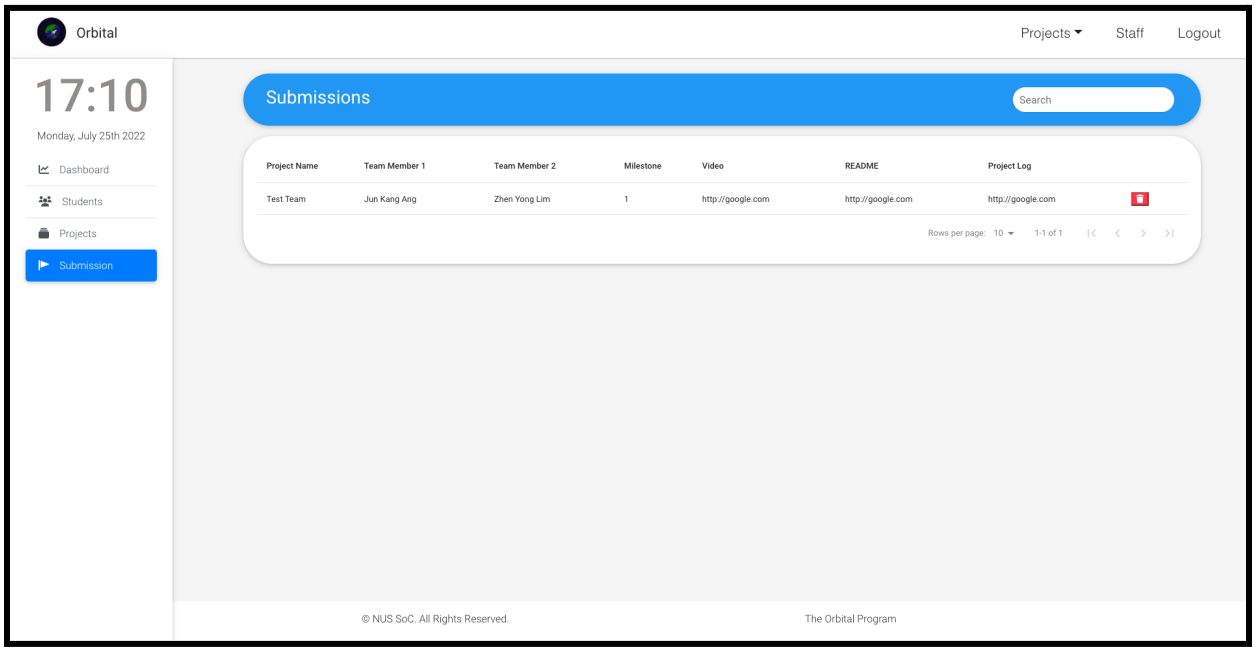
This page displays the details of all projects. Details include team names, team members, team advisors, and project levels. There is a search bar that can be used to narrow down any team. Additionally, the page can be customised to show up to 10, 15, 20, 25, 30, or all teams on the same page.



Team details can be mass uploaded through importing an excel file containing the relevant data. Alternatively, in the event that a team needs to be added individually, the Orbital admin can do so by entering the data of that team manually.

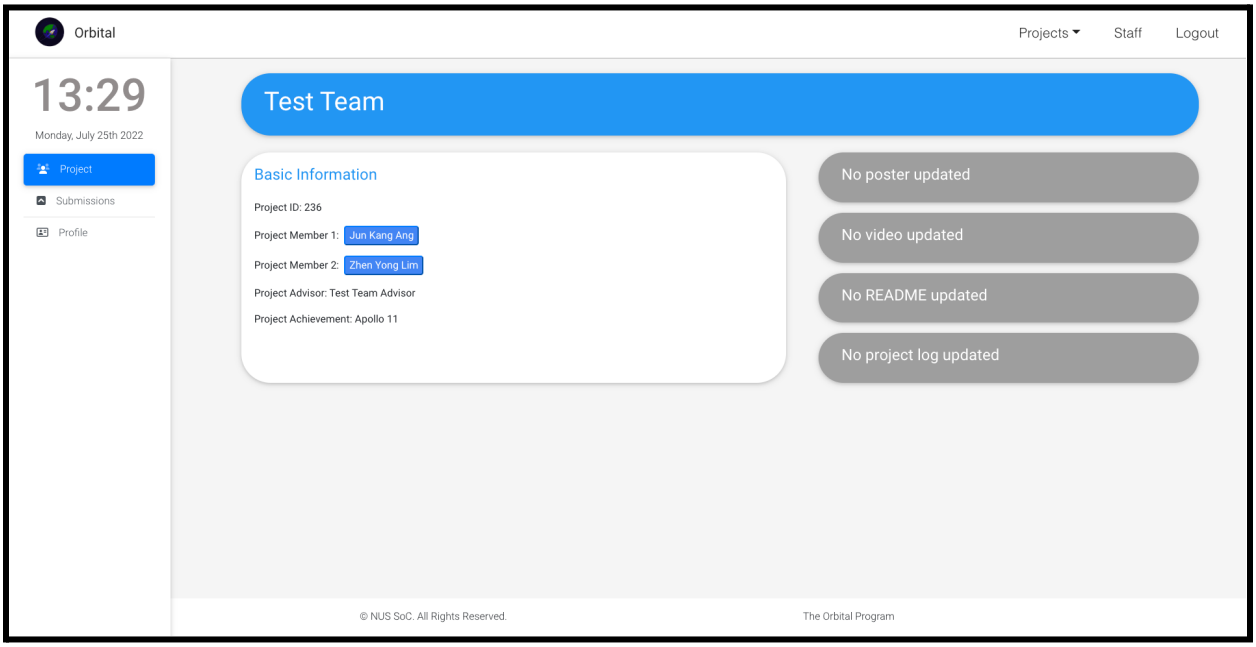
Once added, the information of an individual team can be edited by clicking on the orange / yellow edit button at the relevant row. A pop-up window will appear and the information can be edited through it. The Orbital admin can also remove a team by clicking on the red bin button at the relevant row. A pop-up window will appear to reconfirm the decision.

2.7 [Admin] Submissions Info Page



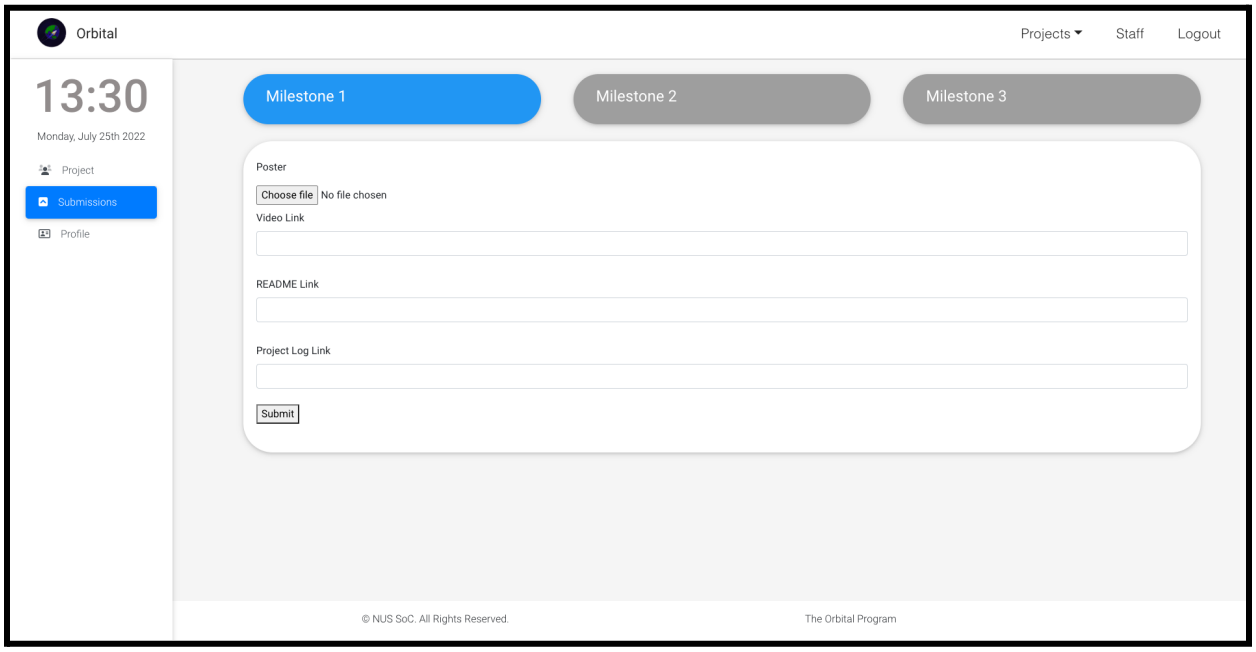
Admin can view details of submissions on this page.

2.8 [Student] Project Profile Page



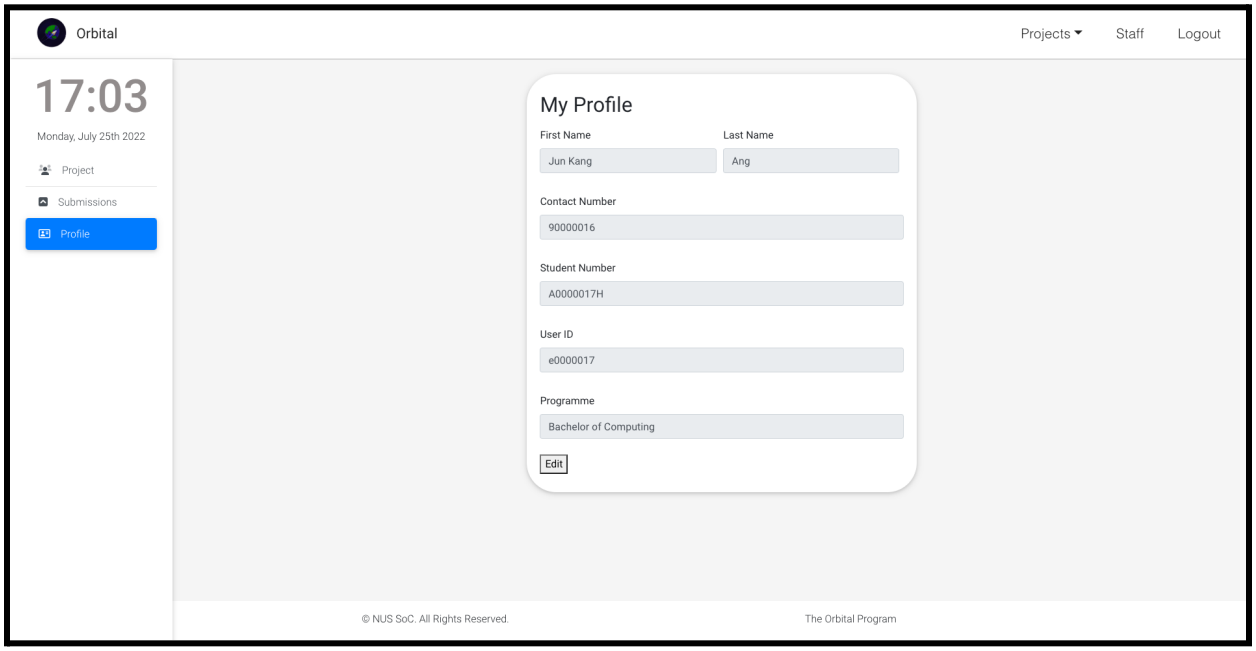
This page shows an overview of the student's team information. Details include the project ID, project members, project advisor, and project achievement level. Students can also access their latest submissions by clicking on the four elongated bars on the right of the page.

2.9 [Student] Submissions Page



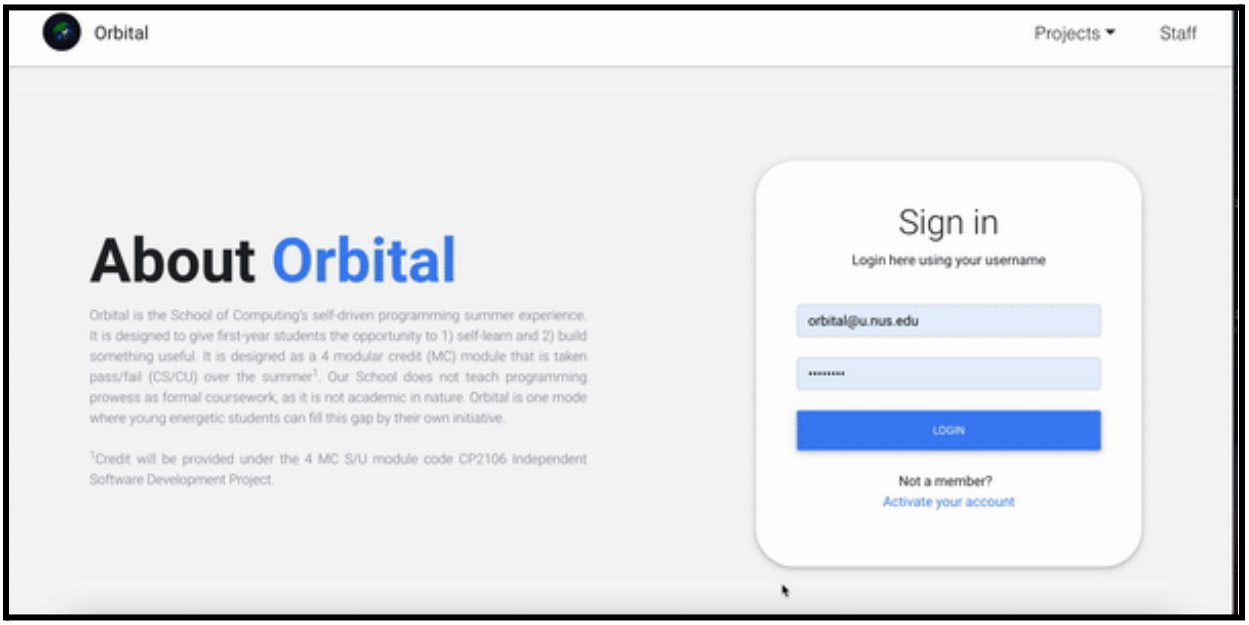
This page allows students to submit their work for the respective milestones.

2.10 [Student] Student Profile Page



This page allows students to view and update their profiles.

2.11 Responsive Design



Responsive design allows the contents of a website to adjust to different screen sizes. This is important as it makes the website accessible across different devices, in turn improving user experience. Responsive design has been implemented for the Main Intro Page, the Projects Display Page, and the Staff Display Page.

Section 3: Execution

3.1 Development Timeline

Date	Deliverables
09/05	**Start**
09/05 - 15/05	• Design Main Intro Page • Design Project Displays Pages
16/05 - 22/05	• Implement Login / Logout Functions • Design Staff Dashboard
23/05 - 29/05	• Design Staff Dashboard • Implement Users Info Page + Projects Info Page
30/05	**Milestone 1**
30/05 - 05/06	• Implement Users Info Page + Project Info Page • Work on backend for Staff Intro Page
06/06 - 12/06	• Website deployment • Come up with UI / UX of Staff Intro Page
13/06 - 19/06	• Implement Team Info Page • Responsive design of Main Intro Page
20/06 - 26/06	• Work on backend of Team Info Page • Responsive design of Main Intro Page
27/06	**Milestone 2**
27/06 - 03/07	• Polish UI / UX of Student Team Info Page + Submission Page
04/07 - 10/07	• Design Student Dashboard Page

11/07 - 17/07	• Project posters and videos backend work • User testing
18/07 - 24/07	• Work on feedbacks from Milestone 2
25/06	**Milestone 3**

3.2 Tech Stack & Tools

1. **React (Frontend Web Framework)**

   React is one of the hottest frontend frameworks used in the industry. Its optimised development interface and coding language are some of the primary reasons for its use in web development. Thus, achieving a hassle-free and rapid development workflow is achieved through lightweight React API's. Its components and concepts are also simple to figure out so there is not much learning curve.

2. **Express (Backend)**

   ExpressJS is hands down the most famous NodeJS framework. It is a prebuilt NodeJS framework that can help in creating server-side web applications faster and smarter. Simplicity, minimalism, flexibility, scalability are some of its characteristics and since it is made in NodeJS itself, it inherited its performance as well. In short, ExpressJS did for NodeJS what Bootstrap did for HTML/CSS and responsive web design and is truly a must have.

3. **PostgreSQL (Database)**

   PostgreSQL has always been one of the few popular databases to use. It has the ability to support all the needed data types such as documents, primitives, geometry, structures etc. It also provides our data integrity by introducing constraints and regulating data we add. Its parallelization of read queries, powerful indexing methods, multiversion concurrency control helps to boost and optimise its performance. It also supports international character sets and enables full-text search to speed up the finding process and integrates case-insensitive and accent-insensitive collations. Last but not least, from the latest update,

it now supports JSON, XML, Hstore and Cstore documents which turns it into a NoSQL database.

Add all that together and we have a powerful data platform that is hard to beat especially for smaller projects that need a reliable choice without ton of effort or cost

4. **NodeJS (Backend)**

   NodeJS is one of the most used backend frameworks. It is easily employed as a server-side proxy where it can handle a large amount of simultaneous connections in a non-blocking manner. It's especially useful for proxying different services with different response times, or collecting data from multiple source points. It provides vast scalability for application and non-block thread execution makes it even faster and more efficient. It is also cross-platform supported which allows us to create SaaS websites, desktop apps, and even mobile apps, all using NodeJS. Lastly, the most important feature is that NodeJS provides a vast set of open-source packages that are available for our work. This is an easy choice for us since the both frontend and backend can be managed with JavaScript as a single language.

5. **Bootstrap/Material Design/Font Awesome (Design)**

   A website or application without design is like a world without colours. These 3 APIs provide minimalistic and simplistic design and icons which helps to add the colours of our world.

6. **Github (Source Control)**

   Github's primary benefit is a version control system which allows for seamless collaboration without compromising the integrity of the original project. Almost everyone uses this for their software engineering practices so this is self-explanatory.

7. **Visual Studio Code (Text Editor)**

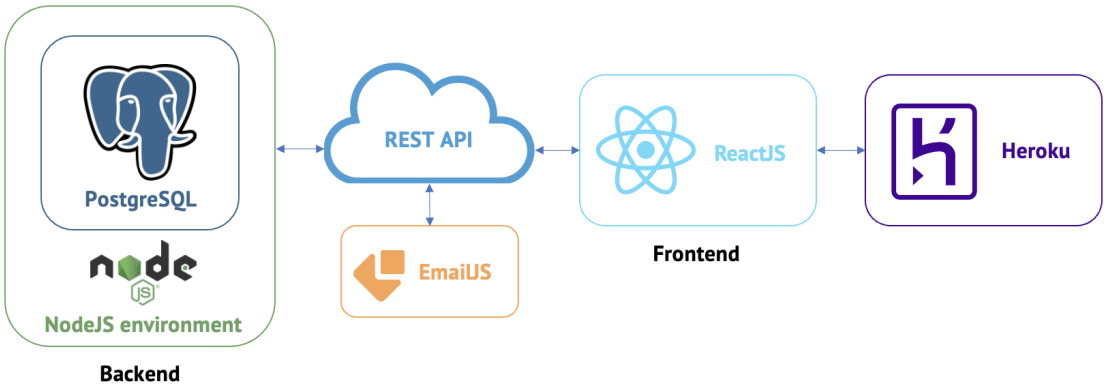
   Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. It is an editor that gets out of our way. The delightfully frictionless edit-build-debug cycle means less time fiddling with our environment, and more time executing on our ideas. It is the best ID without doubt.

8. **Heroku (Hosting)**

Our favourite free hosting platform with a lot of key features. Heroku supports up to eight popular development languages and frameworks. It also has buildpacks to extend the build system to support additional languages or customization. It also has fast app deployment that helps us get our app projects up and running quickly with its ability to deploy on every push to Github or from our own Git, build and push a Docker image, etc. Last but not least, with its integrated data services, we have the ability to install a managed data store such as Heroku Postgres, Heroku Redis, etc to easily persist and interact with and manipulate data.

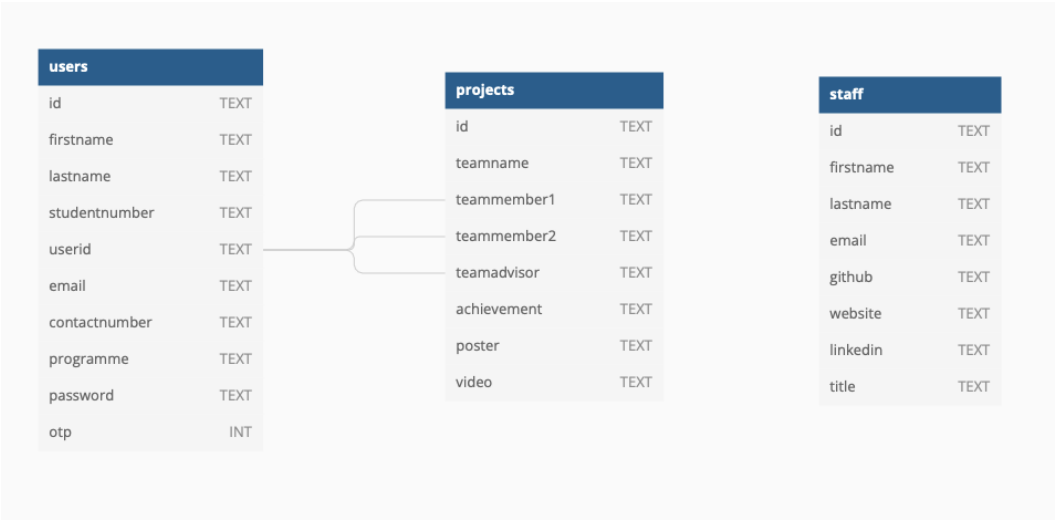
3.3 System Architecture Overview

For our tech stack, we decided to use PERN which consists of a PostgreSQL, ExpressJS, NodeJS backend and a ReactJS frontend. For authentication (e.g, sign in) of users, we use the NodeJS jsonwebtoken package to generate the JWT on the frontend to ensure that authentication is secure and accurate before allowing access to REST API. For email services (e.g, account activation, forgot password), we use the EmailJS to allow for convenient email handling.



3.4 Entity-Relation Diagram

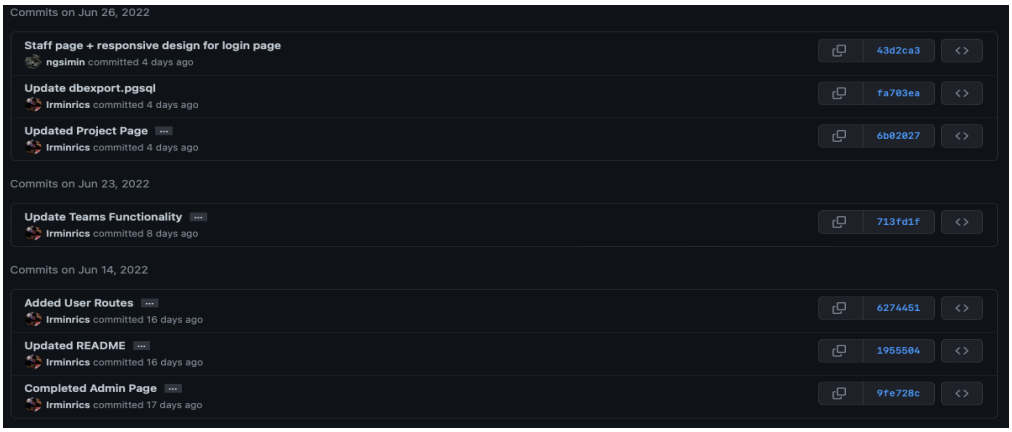
Click here to check out our detailed and interactive entity-relation diagram.



3.5 Git Workflow

3.5.1 Source Version Control

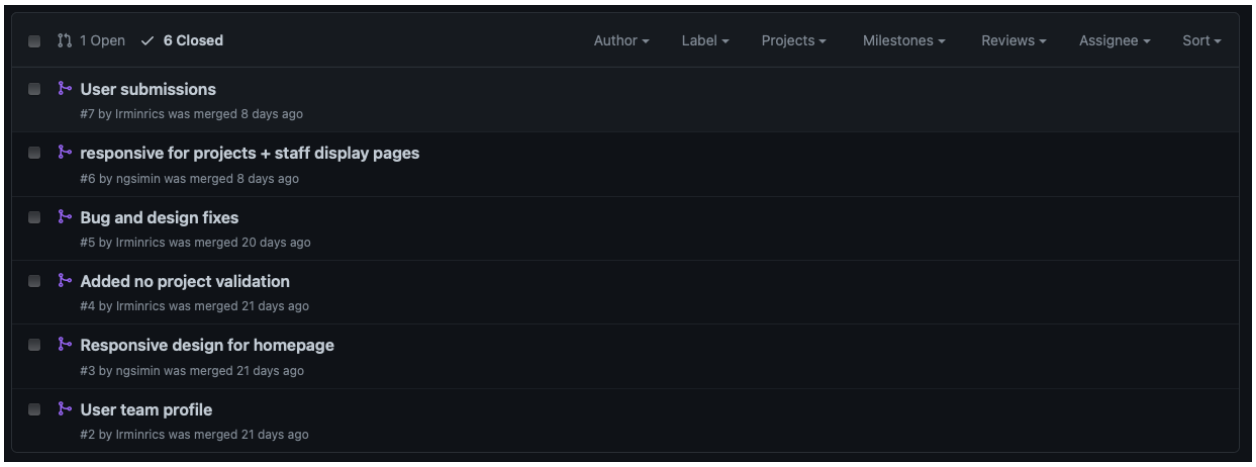
When it comes to source version control, the first thing that comes to any software developer's mind would be Git and Github. Git is a version control system that helps to manage and keep track of our source code history whereas Github is a cloud-based hosting service that lets us manage Git repositories. With these, we used Git and Github to help track and manage changes to our code and this enables us to work on the development of the website at the same time using the different branches.



3.5.2 Workflow

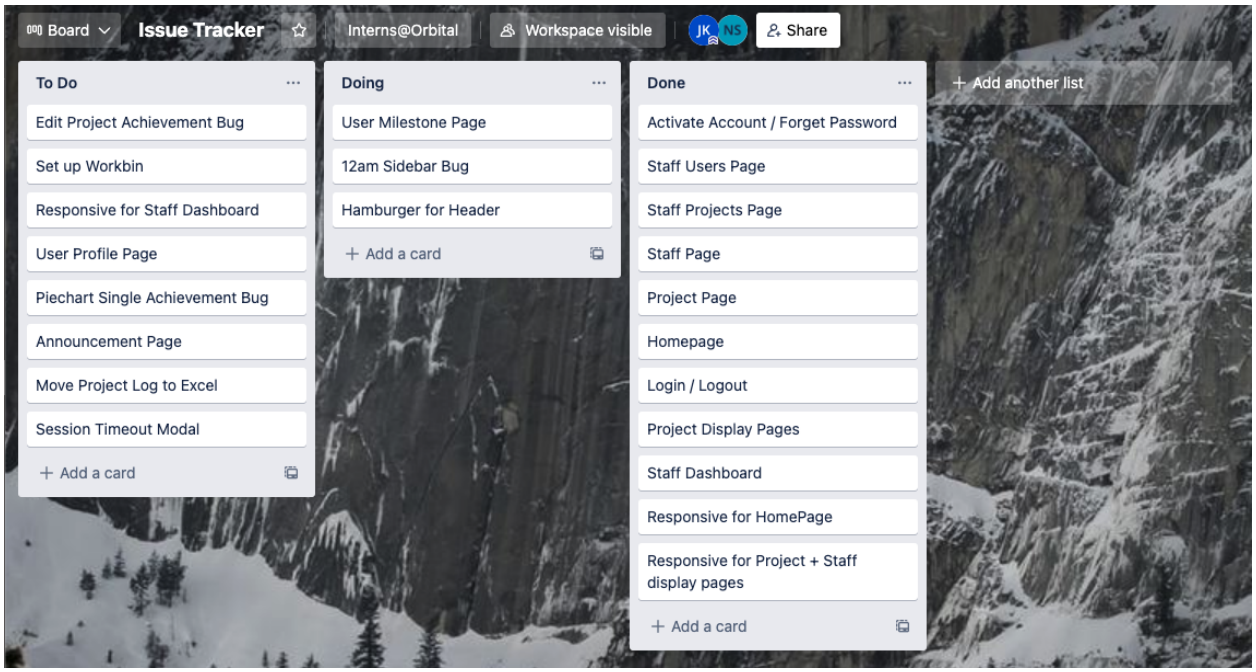
Firstly, we have to have an aim in mind, whether to create a new feature or solve an existing bug. Then, we will create a new branch for it and work on it on our local machine. This helps us to work

on our own features or bugs simultaneously without affecting one another. Once we ensure that the new changes implemented are working as intended, we will submit a pull request for our partner to review. If all goes well, the pull request will be approved by the partner and would be merged into the main branch which would be subsequently deployed onto our website.



3.5.3 Issue Tracking

To track issues, we initially used Github issues to help keep track of issues, features or bugs. However, we moved onto using a google document excel sheet to track manually issues which we find quite cumbersome. After milestone 2, our team has decided to use Trello to help keep track of issues that are either not touched, pending or completed.



3.6 System Testing

3.6.1 Unit Testing

Jest and Enzyme are the two unit testing tools that we looked up upon and decided to use. However, upon further testing and coding, we decided that it was a little too complex to learn as we had many other features to work on. We ended up using the old school console logging method to debug each of the test cases slowly. With more time to learn how to use these unit testing tools more effectively and efficiently, we are sure we will be able to test and craft up better and more detailed test cases.

3.6.2 Backend Testing

Testing Software Used	Reason
POSTMAN	An API client that makes it easy for developers to create, share, text and document APIs A HTTP client that test HTTP requests, utilising a graphical user interface, through which we obtain different types of responses (GET, POST, PUT, DELETE) to test if the API is

Postman	working as intended before deployment It has the ability to create a dummy JSON files to send to the back-end through API calls, to test if the database is properly updated

Postman is a very useful tool for us to test the different types of HTTP requests. With this tool, we are able to test for API calls without an actual user interface. It can also be used to test and validate data provided to the database. In any case of a failed HTTP request through our website, this tool can be used to directly test if the HTTP request is the main problem or if it is due to other factors in the website itself.

3.6.3 Responsiveness Testing

Testing Software Used	Reason
DevTools **Chrome DevTools**	One of the most used browsers and has a tool that is easily accessible which allows us to change the window size easily. Simulate and use virtual mobile devices in Chrome's device mode to build mobile-first websites.

Chrome DevTools are web authoring and debugging tools built right into the Google Chrome browser. From all the browsers and all DevTools variations, Chrome DevTools are the most popular. It provides developers deeper access in our web application and the browser. With this tool, we are able to do everything from testing the viewport on a mobile device to editing a website on the fly, and even measuring the performance of an entire website or individual asset. One special mention for this tool is its ability to freeze the state of the elements, which allows us to test the hover behaviour of some components on our website.

3.7 Usability Testing

3.7.1 Self evaluation using heuristic evaluation

We are utilising Jakob Nielsen's 10 general principles for interaction design. These principles help us to define important points in the composition of interfaces and should be considered when creating layout. We obtained a template from Figma and followed their format.

Severity Rating

(3 factors to consider: frequency, impact and persistence)

0 = I don't agree that this is a usability problem at all

1 = Cosmetic problem only; fix if time is available

2 = Minor usability problem; fixing this should be given low priority

3 = Major usability problem; important to fix, given high priority

4 = Usability catastrophe; fix this before product can be released

S/N	Heuristics	Violation	Recommendation	Severity
1	Visibility of system status	Each of the pages in our system clearly indicate where the user is currently at.	Not applicable	0
2	Match between system and real world	We follow standard UI/UX design practices which can be observed on numerous widely-used platforms. As such, our systems will be familiar to most users.	Not applicable	0
3	User control and freedom	Our system ensures that features like deleting and editing are equipped with a confirmation to ensure	Not applicable	0

		that users are able to comeback from an accident misclick		
4	Consistency and standards	Our system uses team and project interchangeably and might be misleading for users even though they are on a one-to-one relationship	We have decided to synchronise the terms to “project” (Resolved)	2
5	Error prevention	Empty validation is not done for the admin site when the admin decides to edit a user or projects.	We will solve this by the next milestone. (Resolved)	3
6	Recognition rather than recall	Our system is clear and ensures information required to use the design is visible and easily retrievable when needed. This helps to minimise the need for users to recall what the next step is.	Not applicable	0
7	Flexibility and efficiency of use	Although our system cannot be customised by users, all the required features are still clearly shown on the navigation bar.	Not applicable	0
8	Aesthetic and minimalist design	With the help of bootstrap and material, our system adopts a minimalistic	Not applicable	0

		design for each page by ensuring that only necessary information is provided for each feature that the user wants to use.		
9	Help users recognize, diagnose and recover from errors	Our error messages are concise and non-technical, which makes it easy for users to understand what went wrong.	Not applicable	0
10	Help and documentation	We do not have any documentation or pipeline for new users and would require users to understand how our system works.	Create a section for new students to understand how to use the system properly (Resolved)	2

3.7.2 Cognitive walkthrough

In order to understand how potential users navigate our website, we conducted cognitive walkthroughs since they can quickly reveal any glaring usability issues.

Findings	**How to resolve it**	**Status**
Bugs regarding users and projects	Added several validation checks such as checking if the user is already in a project before adding them to a new one.	Resolved
Slow loading from hosting site	Added a loading spinner to allow the users to know that the site is still loading up the required information	Resolved

Admin table should include search functions with pages to make it more user friendly	Changed to datatable with pagination and implemented a search bar	Resolved

3.7.3 Actual user focus group

It was our goal to collect as many valuable opinions about our website as possible from a group of actual users both before and after testing. Considering that SOC students from NUS are most likely to reflect the input of actual users in the future, we decided to invite them to evaluate the usability of our product. They were given access to both the admin and users accounts to raise issues and suggest changes that could help improve the usability of our product.

Suggestion	Status	Remarks
Admin User & Projects List		
Ability to add individual user and projects	Implemented in Milestone 3	We were considering if there is a need for an individual add for users and projects and the focus group solidified our decision and told us it would be useful for software testing
User Features		
Standardising the term projects and teams	Implemented in Milestone 3	We have decided to stick to the term projects in order for it to be less confusing.
Sign Up Page		
Create a sign up page instead of asking the user to activate their account	Rejected	We decided that since orbital is not a site where anyone can sign up for, there is no need for a sign up page as only verified users participating in the module would have the rights to sign up and activate their account.
Profile Page		
Possibility to upload user profile picture	Rejected	Due to time constraints, this feature is a good to have and not a must to have. So we decided not to go with it.

3.8 Project Log

Project Log

Section 4: Reflections

4.1 Limitations

During the course of completing our project, we encountered two main limitations. The first was the need for a responsive site for various platforms. We noticed that different platforms like mobile phones and tablets had different needs and it was difficult to cater to all screen sizes. Some parts of our UI, when adapted to fit a mobile phone screen, looked less aesthetic on monitors with high resolution. As a result, some parts of the UI must be sacrificed to meet the needs of users from other platforms. Our second limitation is that we are using a free hosting site like Heroku for our project. While it meets our basic needs for deployment, the loading speed can be slow at times, as we're using a free plan.

4.2 Future Considerations

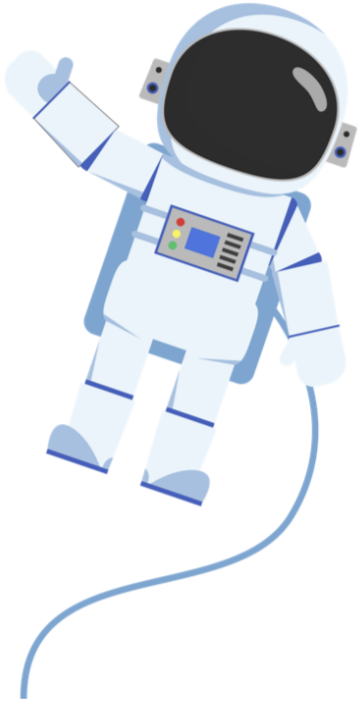
As our current site is a one-year version of Orbital, users will not be able to view past orbital projects. The site can be improved to include past year projects in order to be used consistently throughout the years. The structure of Orbital Skylab can be extended to accommodate more roles as users have different roles. Currently, our site only offers user and admin roles, but this can be enhanced with other users like staff and advisors. It is also possible to add additional features such as a workbin and forums to allow users to access all the required materials of Orbital on a single site. Last but not least, to enhance existing features, admins should be able to set milestone dates and lock milestone tabs for users.

4.3 Acknowledgements

We would like to thank our project advisor **Loh Jia Ming, Rayner** for his guidance throughout the programme. His detailed evaluations have given us many insights, which allowed us to improve on our work. He has also been very responsive and has addressed our queries as soon as he could. Additionally, we would like to thank teams **Maybe<Stocks>**, **LoaNUS**, **Jermit**, and **FT25** for their precious feedback after each milestone.

THE END

THANKS FOR READING!



TEAM INTERNS@ORBITAL
ANG JUN KANG & NG SIMIN
TEAM ADVISOR: RAYNER