**Milestone 2 - NucleUS**

Proposed Level of Achievement:

Apollo 11

Motivation:

Being in university, finding time to pursue your hobby or play a sport which you like has become troublesome. Not only do we struggle to find the time for these activities, but the additional hassle of finding people to do these activities is something which everyone wants to avoid. Being on budget, taking individual cabs to places is an expense which everyone tries to minimize. So to solve this problem and find people with you and for you in the least possible time, we have chosen to develop our application.

# Aim:

We hope to deliver a platform, specific to NUS community members, where people can connect with others around them, view the availability of sports facilities and book them at the same time. On the same website, one can find people to share cabs with based on their convenience and requirements.

Target Audience:

Our website is primarily a booking system for sports facilities of NUS. Hence, the target audience are the all the members of the NUS community.

# User Stories:

* As a user, I want to be able to book sports facilities in NUS and request for more players to join me in an activity if I want.
* As a user, I want to be able to join an activity where there is a request for more people.
* As a user, I want to be able to look for someone to share a cab with such that both of our requirements are met.
* As an administrator, I want to keep check on the usage of facilities and prevent overbooking.

# Scope of Project:

To book a sports facility: We will have a list of all the sports related venues, and venues which can be used for various purposes. According to the user’s choice, they will be redirected to the slot booking page of the respective sport. They can choose their level of proficiency at the sport and select the timing for the booking. They also must indicate if they are looking for people to join them in the activity. Their details such as telegram handle will be stored by our system to be shared with people who might want to join the booking.

To join a booking: Beginning with a page with options of the activity/sport which the user can join, the user’s choices will lead him/her to locations of facilities of the activity and then to the timings for the bookings. Here, the user can see already booked timings which have requests for more people. According to the user’s choice, he/she can choose from among those. The user needs to fit in the proficiency level as required by the individual who made the booking. We also provide a flexibility window of 15 minutes, by providing the telegram username to people who are going to join, in case they want to reschedule the event with the same host by maximum 15 minutes.

To search for a cab booking: Beginning with the form for cab searches, using a keyword such as destination look for a request which has already been made. Otherwise backtrack and make a new request with the relevant details.

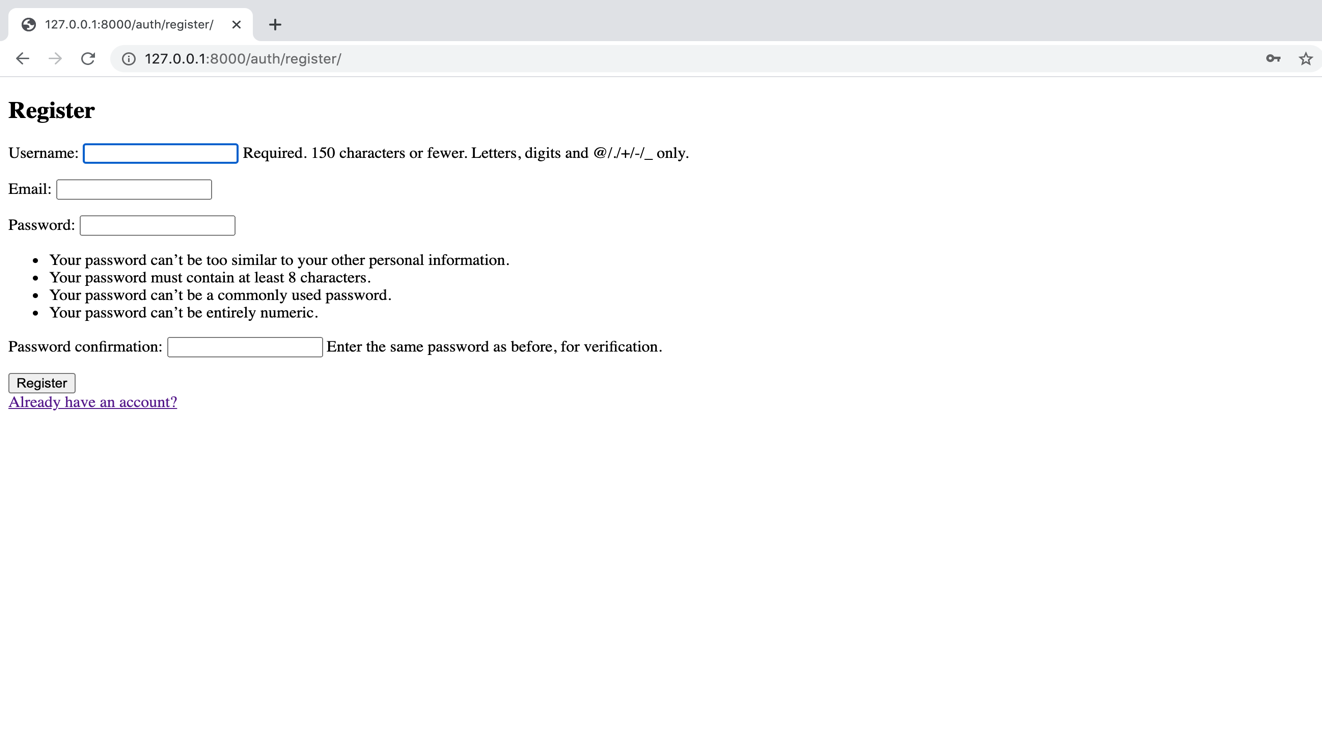
Features completed:

1. Basic logging in system for users :

Users can login from the authorisation page or register with a new account.

A screenshot of a social media post

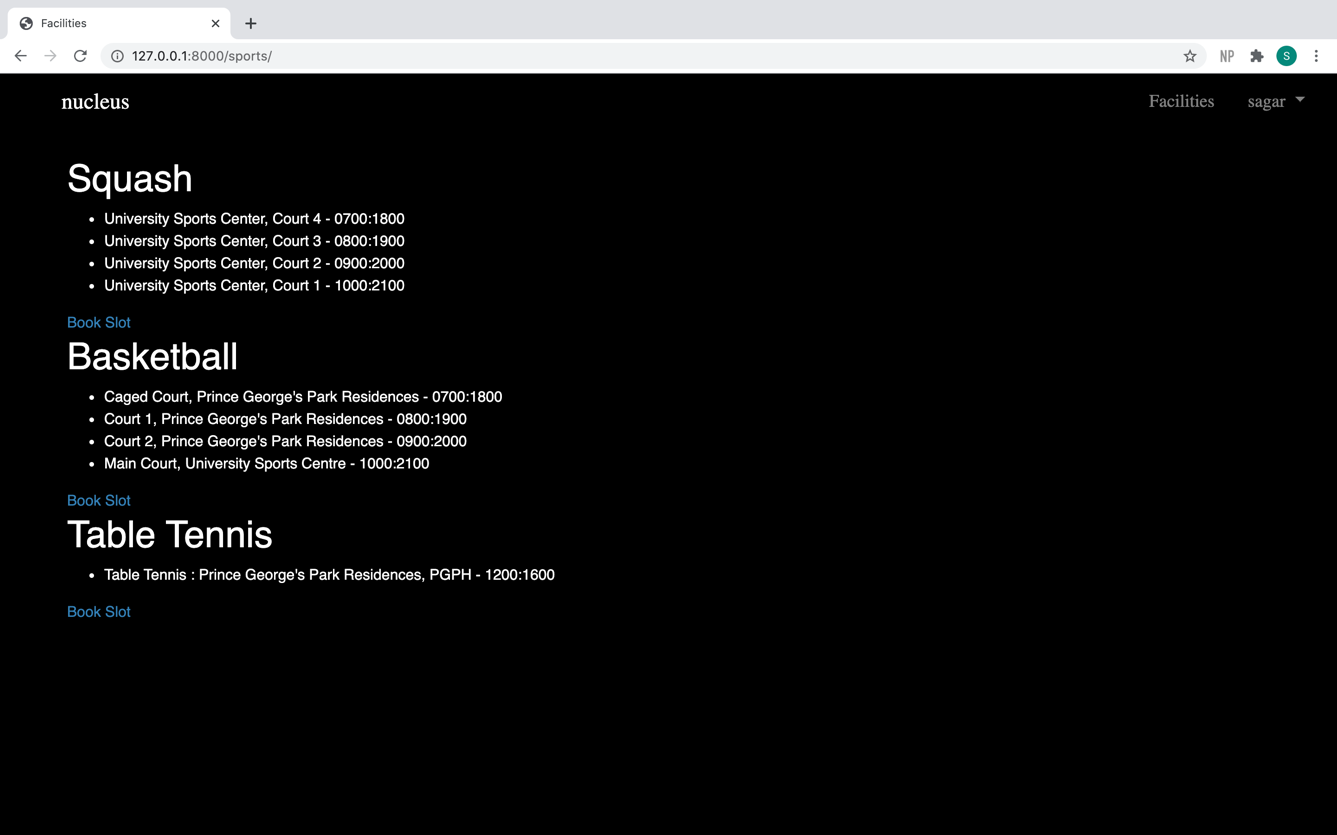
Description automatically generated



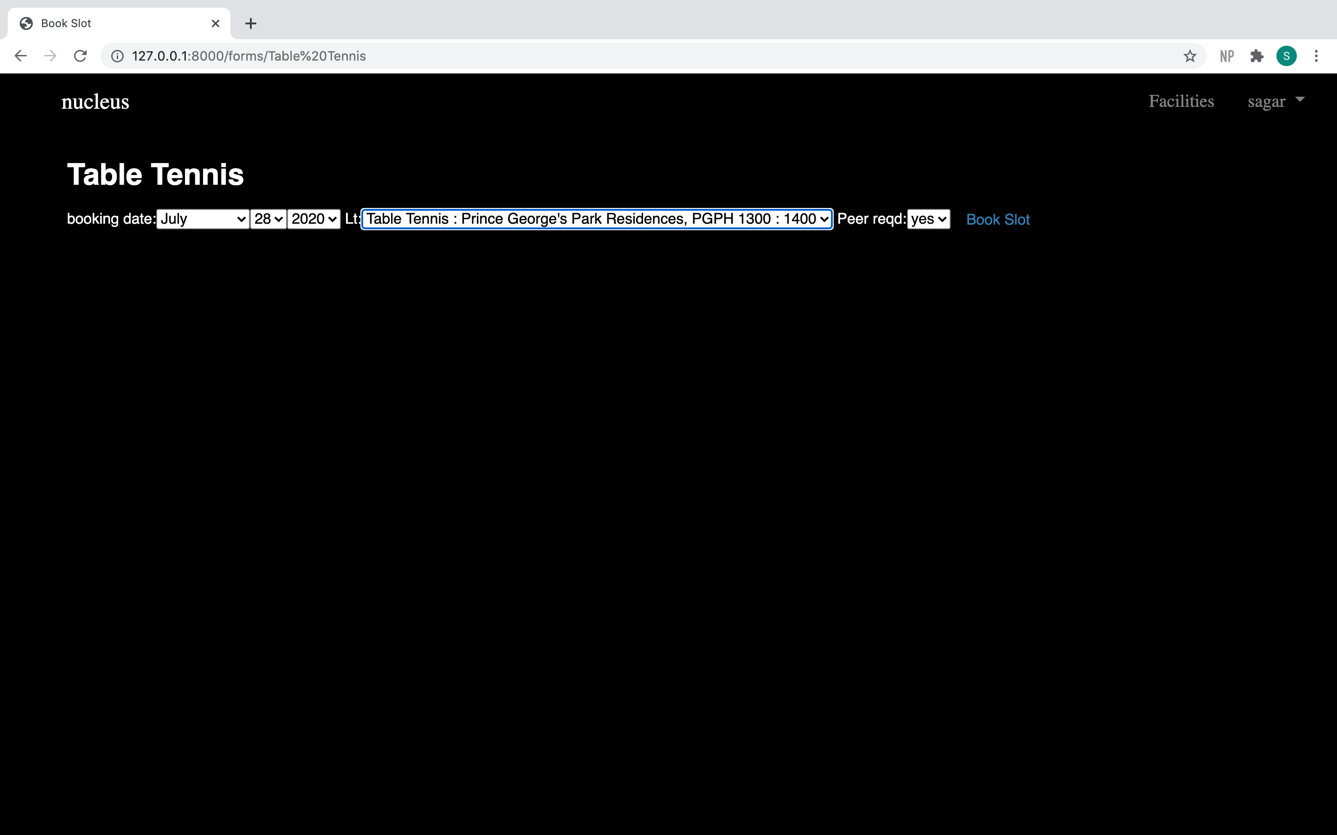
1. Prepare navigation bar and activate buttons to redirect to required pages. Example, home -> facilities.

A screenshot of a cell phone

Description automatically generated



1. Set up the facilities page where the book slot hyperlink takes the user to a page to accept details of the booking.



1. Accept the booking and add to google calendar after user authentication.

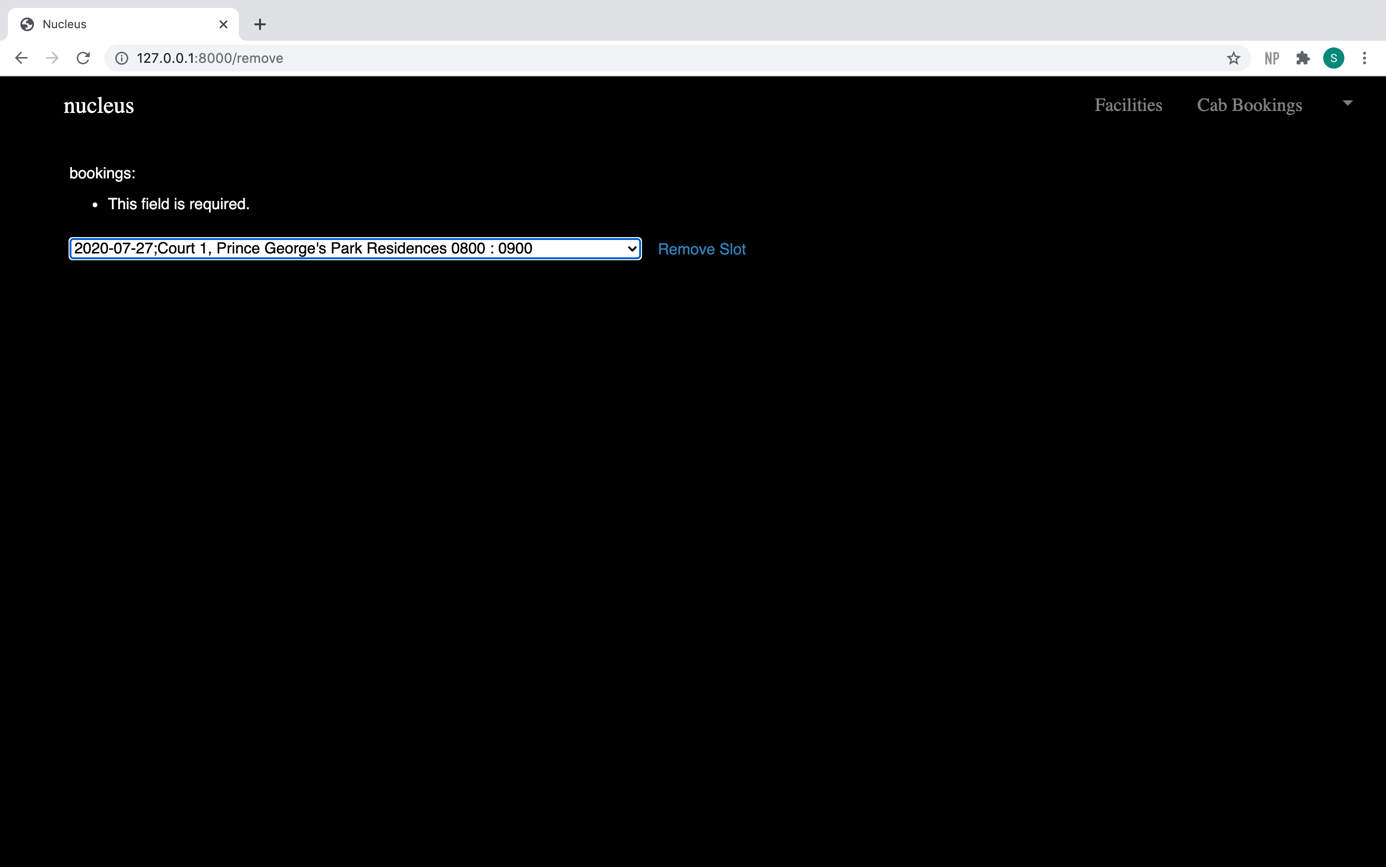
A screenshot of a social media post

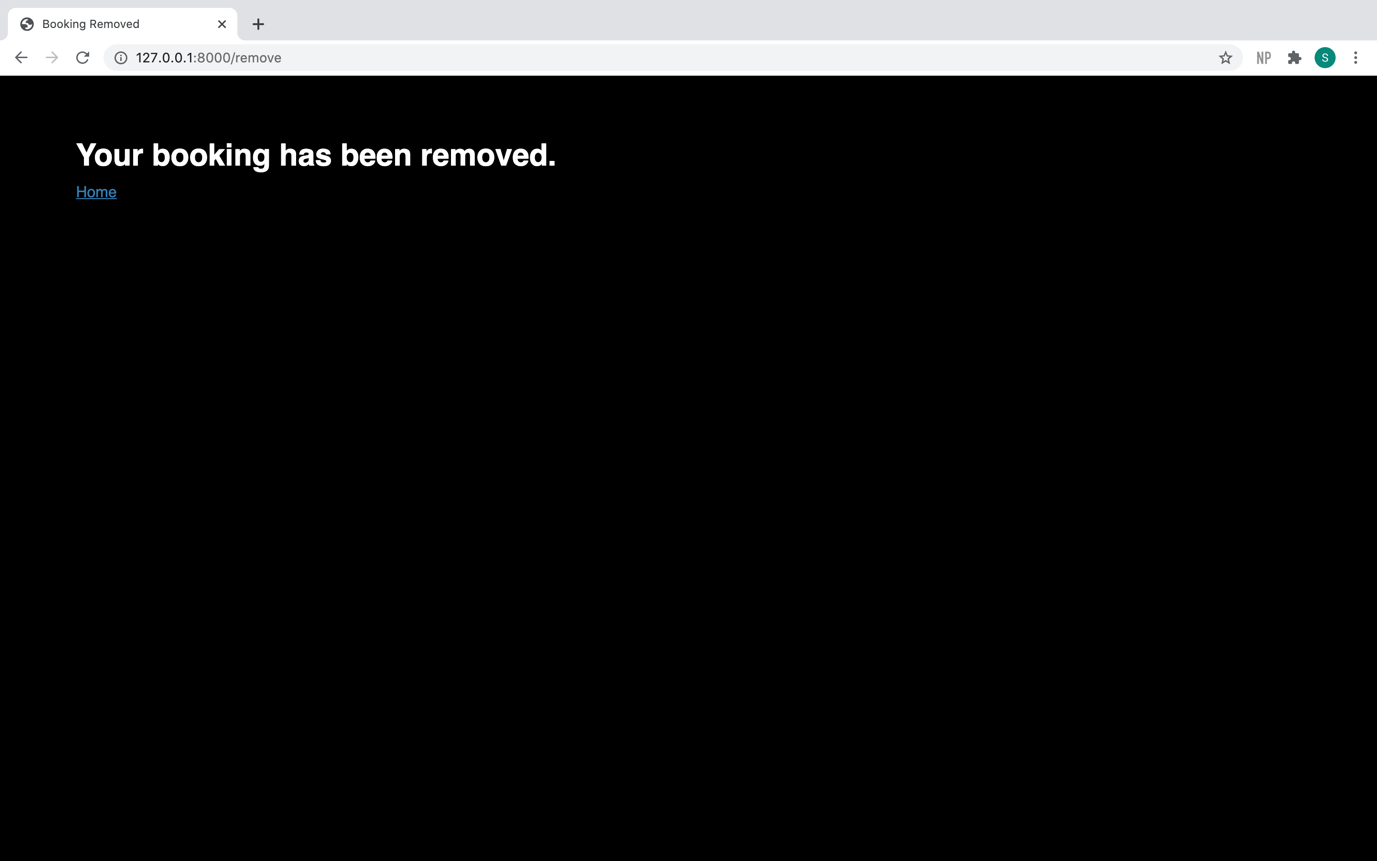
Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. Remove Slot feature takes the user a page where they choose the slot they want to remove.





A screenshot of a cell phone

Description automatically generated

1. Change user password.

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

# How are we different:

* Meetup/Pal- We are specific to the NUS community hence have more credible users and more specific requests. On the same platform, you can also book the facility you want!
* Since it is a community-based project, all the users know that they will be joining with someone from the university, which makes life more convenient for them.

Tech Stack:

Our team proposed to use ReactJS framework and firebase for database. However, we were unable to install the JavaScript library on the laptops of both the orbitees. After a lot of time was spent on solving the issues arising with the tech stack, we decided to switch over to a different technology.

We are now using Django framework, using SQLite3 database. Our application is run through PyCharm interpreter, on which we have installed a virtual environment. This enables us to have different packages installed locally for every project, hence satisfying the dependencies of modules independently. It ensures that the project stays isolated from one another, and also the operating system.

Progress and Issues of the first phase:

Our initial Tech Stack was ReactJS and Firebase. In this set up, we worked for a while and managed to get a functioning login/register set up and a working home page. Here, when we tried to put in a Navigation bar, the JavaScript Library issue held us back. Unable to solve this issue after a lot of effort, we finally shifted to Django and PyCharm, after discussing with our advisor.

Progress and Issues of the second phase:

After switching to the new Tech Stack, we did not face any technical issues. We successfully created the tables in the database and a basic layout for our website. We set up a navigation bar for easy use. The UI for the home page and the facilities page was improved and hyperlinks were created to navigate through the pages on the basis of user choice. Even though we had a basic login system created there were some issues with it and there was scope for improvement.

Progress and Issues of the final phase:

After the second phase, while trying to develop the website further, we realised that the structure of our database was much more complicated than required and it was creating unnecessary delay. Hence, we improved the structure of our database and also changed the UI to a comparatively more elaborate and efficient one. While creating the features we realised we would need Dynamic Page Reloading to successfully make what we had in mind. However, after intensive research online, we learned that Django does not support Dynamic Page Reloading in itself. Due to the time constraint, we focused our efforts elsewhere. We managed to link our website to Google Calendar. We faced many issues along the way but eventually the bookings made on our website were being successfully reflected in the user’s google calendar. Finally, we finished it up by adding small but crucial side features to make the website more convenient and functional.

Program Flow:

1. Start the app:

* Login:
* Enter your registered username and password and login.
* On successfully logging in, the user will be redirected to the default home page.
* Register:
* Click on the “Don’t have an account?” hyperlink.
* Enter the details. The email must be your Gmail id. The app uses it to add your bookings to google calendar. The password has to pass certain constraints, for example, not being very similar to your username.
* On successful registration, the user will be redirected to the default home page.

1. The user can view his/her upcoming bookings in the home page
2. Book Facility:

* Click on the “Facilities” tab.
* View all the locations and the sub-locations. The user can choose the sport they want and click on the “Book Slot” hyperlink below the location.
* Fill in the details- Date, Location and Time, Peer requirement.
  + - * Book Slot.

If the slot the user has chosen is unavailable, the user will be redirected to a page which conveys the message of booking unsuccessful.

Otherwise, it will redirect you to get Gmail authentication with the same email id as the one given during registration. Once access is provided, the user can close the tab and the webpage will be redirected to a booking confirmation page.

* + - * The booking details will be reflected on the google calendar linked with the specified email id.
      * Click on the “Home” hyperlink to go back to the home page.
      * Here, the user can view the users booked under that booking. If another user had previously booked in that slot their username will also me mentioned next to the current user’s username. Even in the other user’s homepage, the username of the new user will be reflected.

1. Remove Slot:

* In the home page, click the “Remove Slot” hyperlink. This will redirect the user to a page to accept the details of the booking to be removed.
* Choose the booking to be removed. Click on remove slot.
* This will redirect the user to the confirmation webpage of the deleted slot.

1. Change Password:

* From the dropdown menu next to the username, choose change password.
* Fill in the old password and the new password. The new password must also satisfy the constraints mentioned.
* Click on “Save Changes” to successfully save change your password. The user will be redirected to the home page.

1. Logout:

* From the dropdown menu next to the username, choose “Logout”.
* The user will be redirected to the login page.

User Testing:

We asked a few of family members to try our website and share their experience and give their inputs. Following is a summary of the user testing we did.

All our users found navigating the website convenient. They were comfortable with the User Interface almost immediately. Even though the booking slot part of the system was okay, the functionality of other users joining a booking could be improved. With additions such as allowing the user to know that they are joining others before they book the slot or mentioning then number of people required while joining the booking. Another improvement they suggested was that when a user removes a booked slot it should also be removed from their google calendar.

Collaboration:

We are using GitHub for version control and collaboration. We made a private repository “nucleus\_orbital” on which we are storing the working versions.

A screenshot of a cell phone

Description automatically generated

Project Summary:

Our team has created a facility booking system for the sports facilities of NUS. Our initial plan was to implement something to allow users to join other users in a booking with the knowledge of how many people are required and the preferred skill level in the activity. Along with that we intended to have a feed for cab bookings where users could look for other users who would like to share a cab with them. Along with this we wanted to link our website to the user’s Gmail account to reflect the bookings made on our website on the user’s google calendar.

Over the course of the project, our team faced many issues, both technical and external. The work we did in the first phase of the project did not come into use. After creating a doing the initial work with ReactJS, we had to migrate to a fresh tech stack with Django. In the third phase of the project, we faced external issues which adversely affected our progress.

Eventually, we successfully created the booking system. In the booking system users can put in their preference about other people joining the booking. They can see the users booked under the slot in the home page. However, they cannot know how many people are part of the booking and what is the preferred skill level. We have successfully linked the website with the user’s Gmail account to reflect bookings in google calendar. Unfortunately, we could not implement a cab bookings feed in the limited time frame.