



Arena Access

A **Flutter** app made by **Subhajeet Lahiri** for the **Zense Recruitment Drive 2022**

Try the app now !!!

https://iitbac-my.sharepoint.com/:u:/g/personal/subhajeet_lahiri_iitb_ac_in/EcM7OGtn7sxGmW5Ev3xwINMBSBRffEf6zhxOnGUdLuuOQ?e=ddCvHt

Getting Started

You can download the .apk file from the above link.

Otherwise :

- 1) Clone the repo/download the zip file and extract the contents on your system
- 2) Install flutter with support for android development if not done already
- 3) Hook your android device to your system/ get an emulator up and running.
Your device : Enable USB debugging and plug the device in.
Emulator : Install android studio and start an emulator.
- 4) Run 'flutter pub get' inside the root arena access directory to install the necessary packages
- 5) Execute 'flutter run' on the terminal
- 6) Make sure you are connected to the internet so that the app can fetch booking data/news.

The last two steps are illustrated as follows :

```
HelioSpook@DESKTOP-J1GEE3J MINGW64 ~/Desktop/arena_access-master/arena_access-master
$ flutter pub get
Running "flutter pub get" in arena_access-master... 3.6s

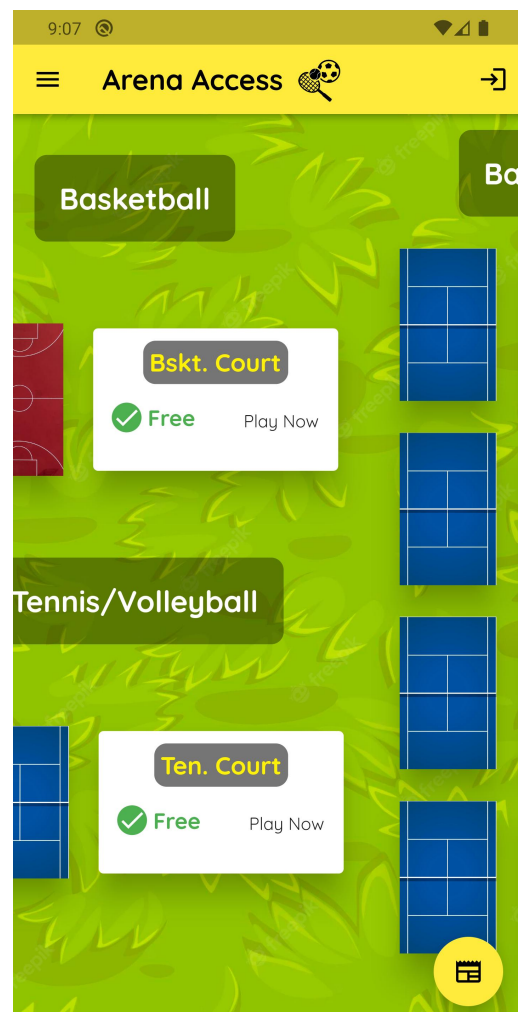
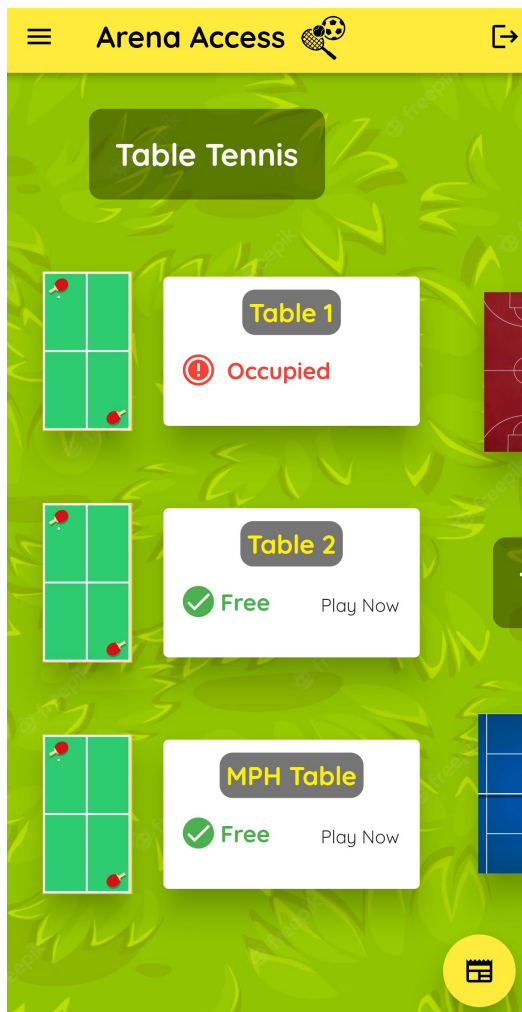
HelioSpook@DESKTOP-J1GEE3J MINGW64 ~/Desktop/arena_access-master/arena_access-master
$ flutter run
Using hardware rendering with device Android SDK built for x86. If you notice graphics artifacts, consider enabling software rendering with "--enable-software-rendering".
Launching lib\main.dart on Android SDK built for x86 in debug mode...
Running Gradle task 'assembleDebug'... 100.2s
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app.apk... 7.8s
Syncing files to device Android SDK built for x86... 460ms
```

The app may take a minute or two to build.

Features / The App

Arena Access is a solution to track everything related to **on-campus sports** .
You can:

- ✧ **Track whether arenas(grounds, courts, tables) are free or occupied right on the home screen.**



I used a horizontal scroll-view and a lively color-scheme to keep the UI fresh and in-line with the app's purpose.

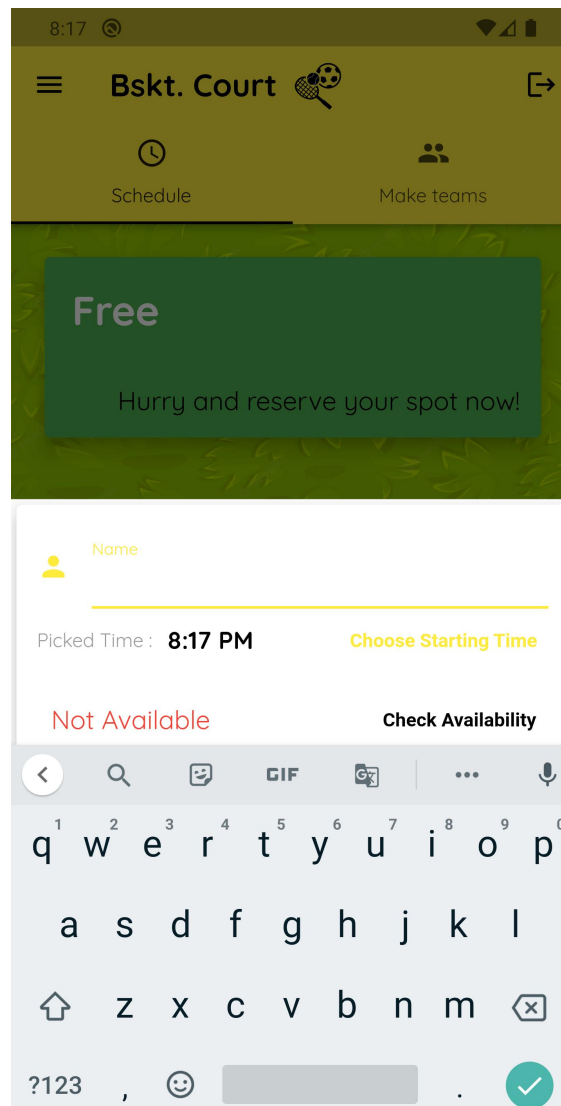
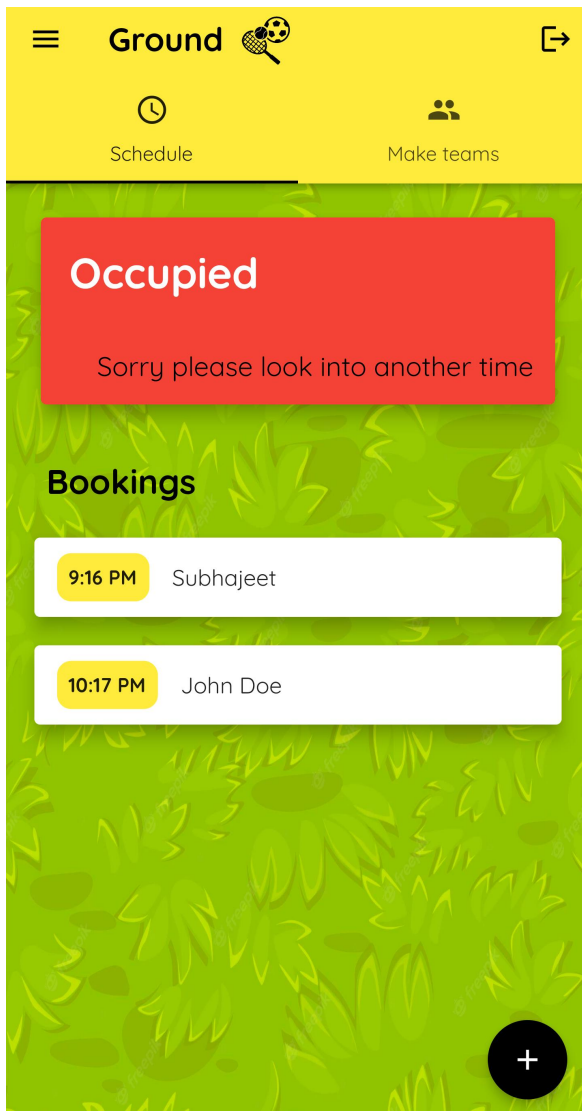
The bookings for each arena are stored in a Real-time Firebase database while the occupied status logic is handled on the device itself.

The app bar action button is for logging users in / logging them out.

The app drawer allows quick access to all the sections of the app.

The Floating Action Button leads users to the news section.

✧ View/Add bookings for 30 min by clicking on the respective arena

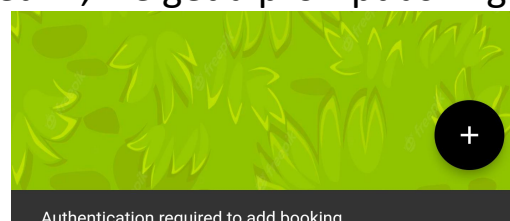


On clicking on an arena , we are lead to its specific screen. We can view the current bookings for the arena on the same day.

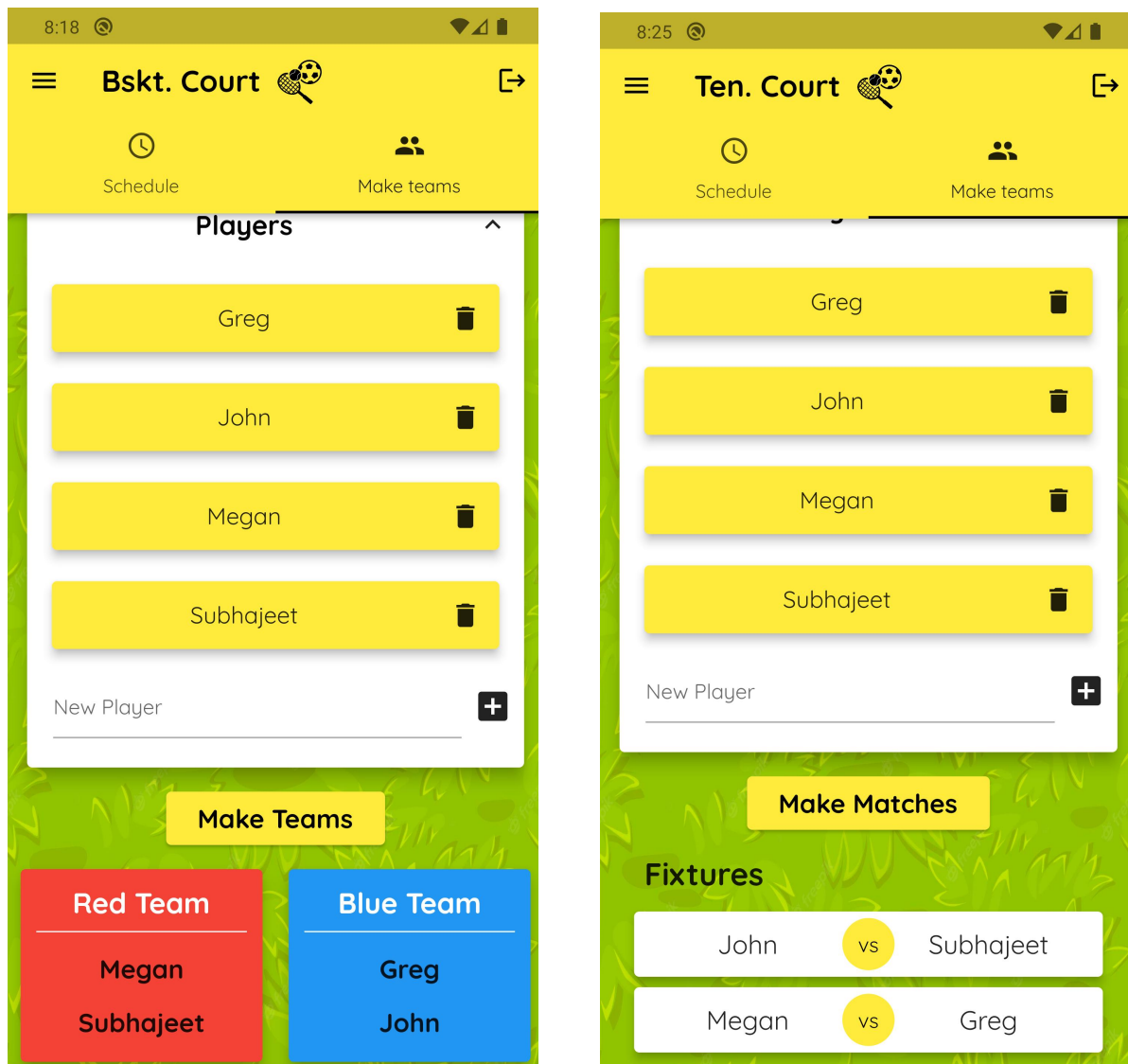
If we are logged in, we can add a booking of our own by giving our name and start time. Before adding one , we need to verify the availability of the court at the given time using the 'Check Availability' Button.

If the arena is available, the booking is added and the arena status is updated.

If we are not logged in, we get a prompt asking us to do so.

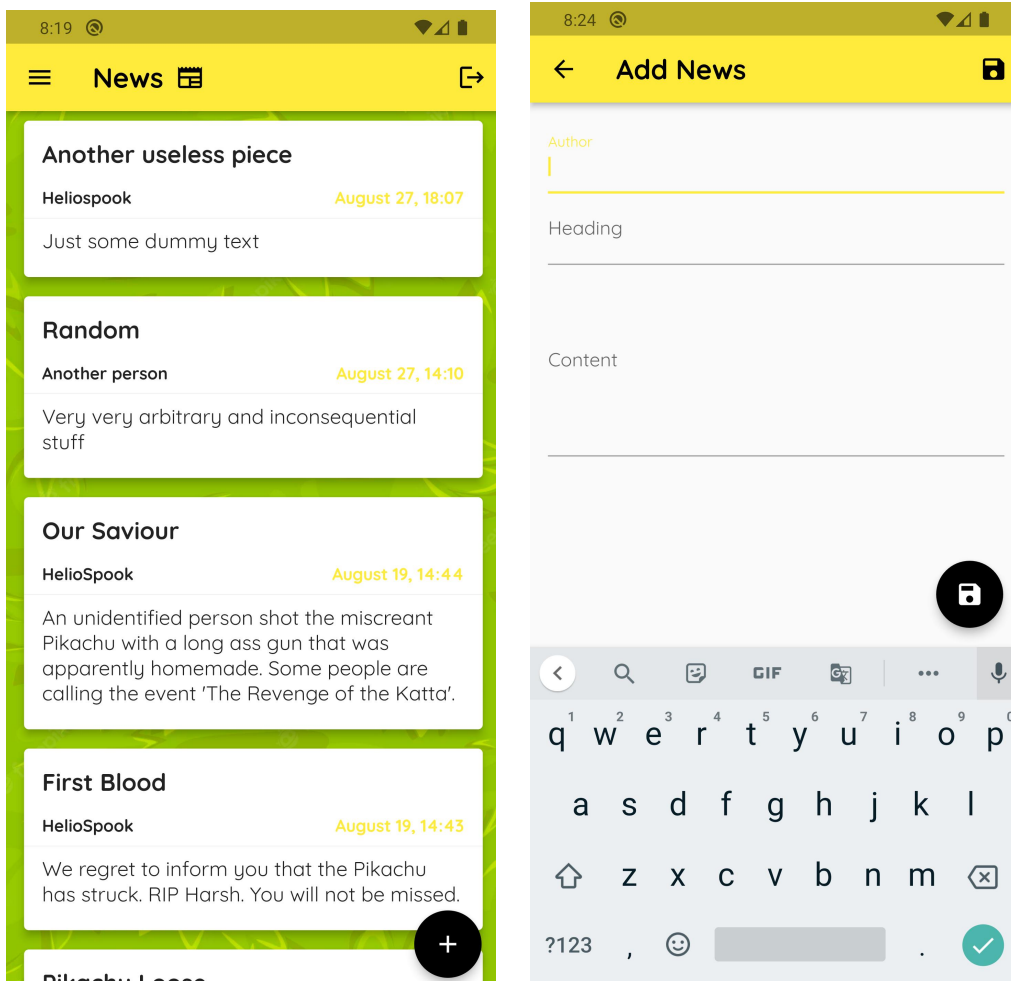


✧ Add Players and make random Teams / Fixtures



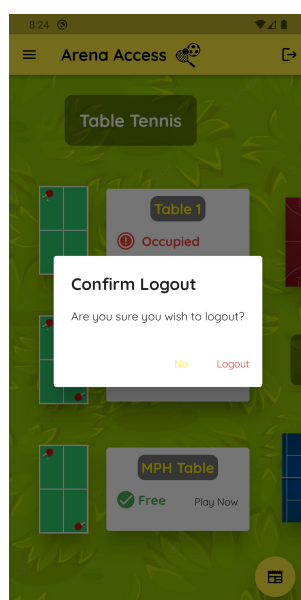
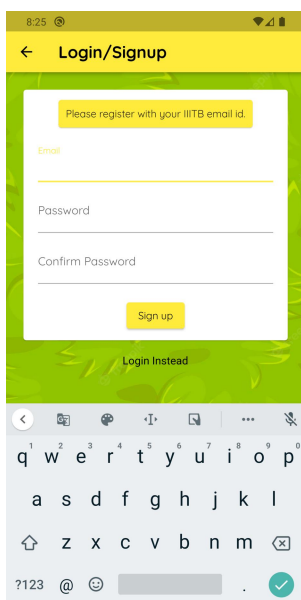
In the other tab (Make Team) for each arena, we can make teams/fixtures. The drop-down button lets us add players and then we can make matches at the tap of a button. The player data is consistent throughout the application as it is provided by a single class and is locally stored.

✧ Follow and add news/announcements

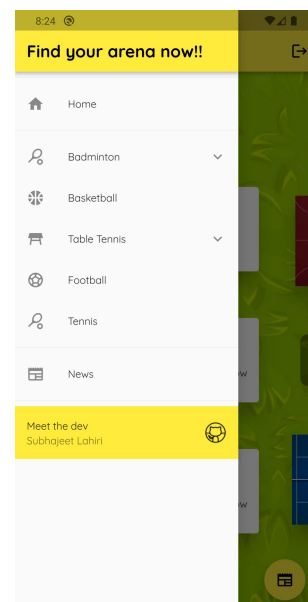


News data is stored on the database and the user needs to be authenticated to be able to add news.

Login/Sign-up



App Drawer



Building the App

The app has been built using the Flutter framework with Firebase providing the Real-time database and the authentication services.

There are three folders in the lib directory of the project besides the main.dart file :

Screens : Their namesake; the widgets containing Scaffold

Widgets : The smaller re-usable widgets

Providers : The classes which provide data to various parts of the app. There are 4 main providers :

- Arena/ArenaCollection : For maintaining information related to bookings and syncing data with the web.
- TeamData : For adding players, making teams/tournaments.
- Auth : For user authentication
- News : Same as ArenaCollection but for news.

The Firebase Real-time database has two folders for Arena data and News data.

Problems Faced :

- ✧ Combing through the bookings and showing the relevant ones was a chore. I implemented some functions in the Arena.dart provider and others in the ArenaCollection.dart provider. So adding the providers and consuming data at the correct places took some time.
- ✧ Authentication was added last minute and due to complications introduced by me in the fetchandsetdata function for the arenas , I couldn't add more user specific functions in time.

In-future :

The arena provider could be cleaned up and user-specific functions like deleting booking/news and a booking quota can be added.