Resonate- Flutter App

Personal Information

Name: Akshat Kedia

Email: akshatkedia9@gmail.com

Phone: +919771006409

GitHub Handle: https://www.github.com/Axhaat

Discord Name: Axhaat#8015

LinkedIn: https://www.linkedin.com/in/akshat-kedia-8787531b1

Skype name: live:.cid.c471d3d7fb9daa8f

Proposal: https://github.com/Axhaat/Gsoc-proposal-Resonate

Address: Tiruchirappalli, Tamil Nadu, India

Time Zone: IST (GMT +5:30)

Project: Resonate|AOSSIE

Education

University: National Institute of Technology, Tiruchirappalli, India

Degree: B-Tech in Instrumentation and Control Engineering

Expected Graduation Date: 30 June 2025

Synopsis

Chosen Project: Resonate | AOSSIE.
Project size: 350 hours
Mentor: Jaideep prasad

Abstract:

Resonate is an **open-source social audio application** that allows users to create and join audio conversations in real-time. Like other popular social audio apps like Clubhouse and Twitter Spaces, resonate provides a platform for users to connect with others around shared interests and engage in meaningful discussions. With Resonate, users can create or join "rooms" based on topics of interest and participate in live audio conversations with other users. The app also features the ability to "raise your hand" to speak and a text chat feature, allowing users to interact with each other in a variety of ways. In simple words, resonate is an open source version of clubhouse, which means that the code is freely available for anyone to use, modify, and contribute to. Overall, resonate provides a unique and engaging way for users to connect with others and have conversations on a wide range of topics, all within an open and accessible framework.

Vision (Why Resonate):

The goal behind Resonate, an open-source project of social audio like Clubhouse or Spotify Greenroom would be to provide a platform for users to engage in audio-based conversations in a collaborative and community-driven environment. By making the project open source, it would allow developers and creators to build upon the initial platform, adding features and functionality that may not have been originally conceived. An open-source social audio platform could also offer increased transparency and user control over their data and privacy. Users could potentially have more input into the development and direction of the platform, leading to a more personalized and user-friendly experience. Additionally, an open-source social audio platform could provide opportunities for innovation and experimentation in the audio space, potentially leading to new and exciting use cases beyond traditional social networking. Overall, resonate has the potential to foster a more **collaborative, transparent, and innovative audio community**.

Motivation:

As an enthusiastic developer with a strong passion for open-source projects, I have consistently pushed myself to go beyond my comfort zone and explore new challenges. With over a year of experience in Android app development and contributions to multiple open-source projects, I am confident in my ability to achieve the proposed goals of this project. In addition to completing

various Android development courses, I prioritize staying up to date with the latest best practices in the industry. My skills and experience make me an ideal candidate for this project. I am a committed team player who can learn quickly and adapt to any situation. I have been actively contributing to Resonate for past 3 weeks, and I plan to continue doing so even after the GSoC period. I have already begun working on the project and have even created a prototype for it. I believe that my passion, skills, and dedication make me a strong candidate for this project, and I am excited about the opportunity to contribute to its success.

Detailed Proposal Description

Architecture

Here is a proposed architecture that would be best the Resonate: -

1. Front-End (Flutter):

The client-side of the application will be built using Flutter, a mobile development framework that enables cross-platform development for both iOS and Android devices. Flutter's rich widget library would provide a range of tools to create an engaging and responsive user interface.

2. Real-Time Communication:

Real-time audio communication is the core of the Resonate, and Flutter provides several plugins like Flutter WebRTC or Agora.io or LiveKit to create real-time audio connections between users. These plugins offer features such as low-latency audio streaming, voice and video calling, and screen sharing. We would be working with LiveKit as per the discussion with the mentor.

3. Backend:

For the backend development, we can use a modern serverless architecture that offers high scalability, low maintenance, and cost-effectiveness. You will use services Google Firebase to create a scalable backend for your app. We will use NodeJS to interact with LiveKit and Firebase.

4. Data Storage:

We will need to store user data such as profile information, audio recordings, and user interactions. Firebase offers Firebase Cloud Firestore, a real-time NoSQL database, that can be used to store user data and interactions.

5. Authentication:

User authentication is crucial, and for that purpose, we can use AuthO for authentication of users.

6. Notifications:

Push notifications are vital for user engagement, and we can use Firebase Cloud Messaging (FCM) to send notifications to users.

7. Analytics:

To understand user behaviour and improve app performance, you need to track user interactions and app usage. Firebase Analytics can track user behaviour and app usage, providing insights that can help you improve your app. Furthermore, In-app feedback can be used to obtain feedbacks from the user.

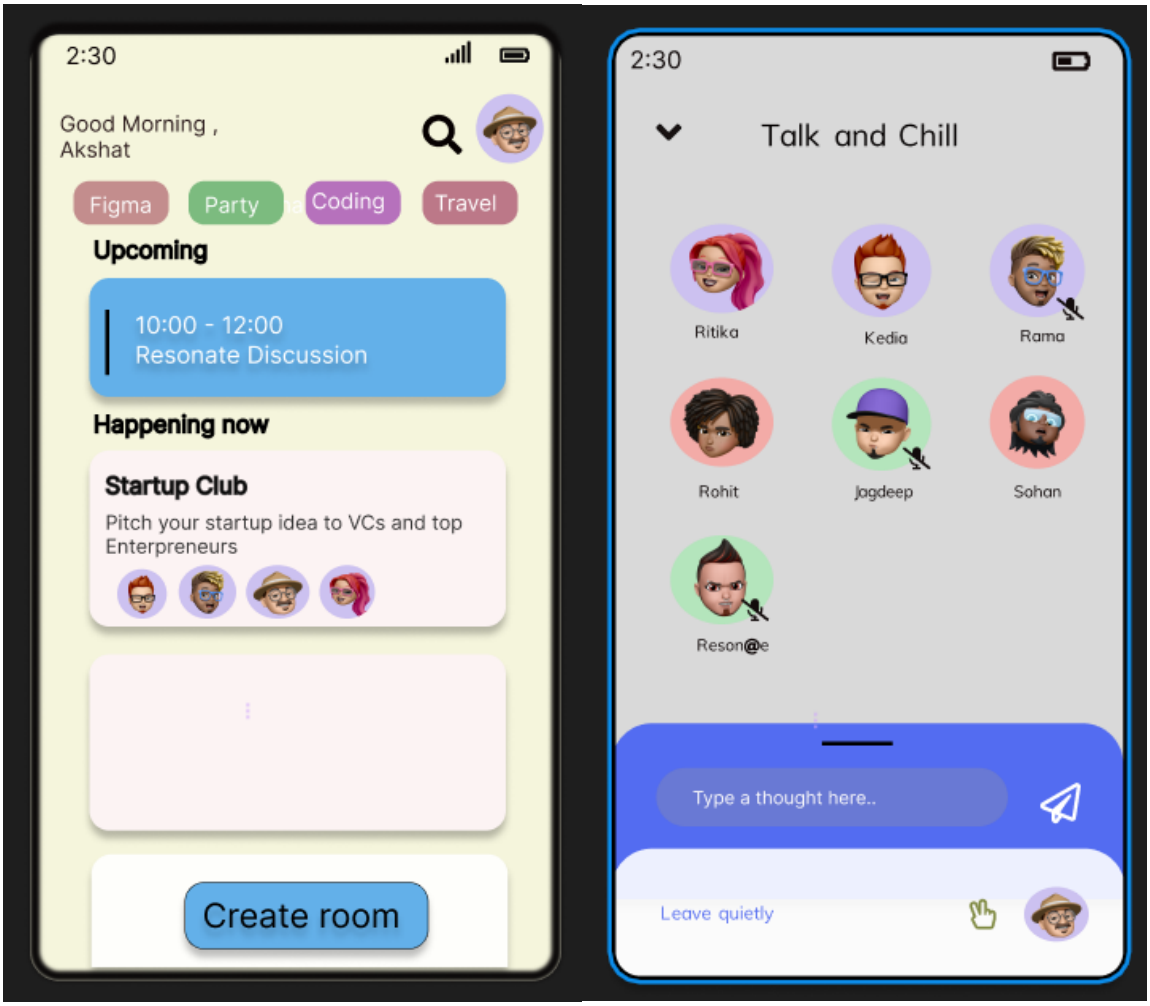
Since we will be building the app from the scratch, here is **list of features** we will be working on:

1. **Registration:**
   - User receives an invitation via mobile number
   - User clicks the link, installs the app, and logs in
2. **User Profile:**
   - User creates a personal profile with real name, photo, and short description
3. **Rooms:**
   - User can create rooms to discuss interesting topics
   - User becomes the moderator of their created rooms
4. **Search:**
   - User can search for rooms by entering keywords in a search bar
5. **Feed:**
   - Feed displays information about rooms or people based on user's search history, preferences, and followed people
6. **Upcoming for You:**
   - Upcoming events planned to join are displayed in this section
7. **Activity:**
   - Users can view updates from their Clubhouse community
   - When a user's contact list joins the app, it will be reflected in the Activity section
8. **Notifications:**

• Users receive notifications when people they follow join new rooms as listeners or create their own rooms.

Other additional features like kicking people out, blocking users, live polls, and inviting speakers will also be added.

Here is the Tentative UI for the app- Home page , Room (I have created a tentative UI for the app, final would be made under the guidance of the mentor)



Here is the goal of the project:

- UI for the App
- we will discuss Firebase integration for authentication and back-end.
- we will review how to embed Agora's voice call functionality into our app.

- Add different functionality like creating club, having different designation in a room like (coordinator, speakers, and visitor)
- Once a feature is completed, I would be adding the unit tests for the implemented feature.
- Setup automatic CI/CD on GitHub
- Setup issues and PR template on GitHub

Timeline/Implementation

App implementation is divided into the following timeline:
(All dates mentioned below are of 2023)

❖ Community Bonding period (May 4 to May 28)

➔ Communicate and bond with fellow students and mentors. Also, I would try to learn about the ongoing projects in the organisation especially those related to flutter.

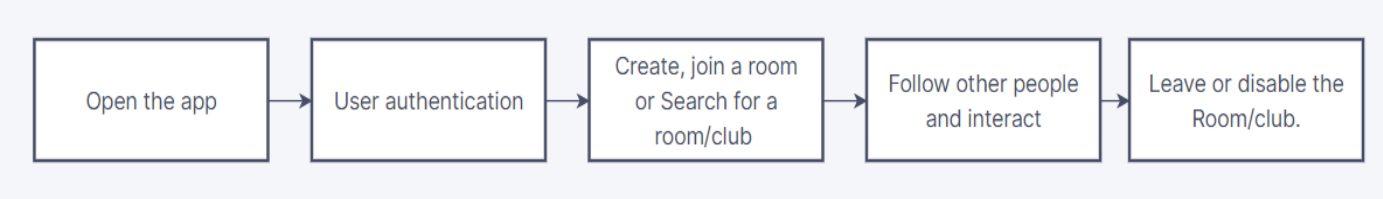
➔ Create specific issues for the project after discussing the **app flow/architecture** with the mentor.

➔ Getting familiar with the app architecture, APIs, and packages.

➔ I would explore more about the tools that can be used to do the project and discuss the final choice of tools/libraries with the mentos.

➔ Since this app would be published on the Google play store, I would start preparing the assets like icons, etc.

➔ Finalise deadline and milestone with the mentor and modify if any need arises.



Deliverable:

➔ Community bonding report and report about the experience, and app architecture

❖ Week-1 (May 29 to June 5)

- ➔ Coding officially begins.
- ➔ Set up the flutter project and repo on GitHub.
- ➔ Create the app's final design/UI or modify the design/UI that has already been designed based on the feedback from the mentor.
- ➔ We will use widgets like Text, Container, Column, Row, RaisedButton, etc. to build your UI.
- ➔ After getting confirmation on the design, I will go on to the coding part of it.

❖ Week -2 (June 6 to June 13)

- ➔ After getting confirmation on the design, I will go on to the coding part of the app.
- ➔ Setup the firebase Authentication

- ➔ Based on the verification method (either with email id of mobile number), we will create a new Firebase Project from the console and integrate it with our Flutter application.
- ➔ After the android and iOS configuration, we will enable the verification method we would be using from the firebase console.
- ➔ Add Firebase authentication, Firebase core and cloud Firestore plugins in pubspec.yaml
- ➔ Enable Firebase user verification.

```dart
import 'package:firebase_auth/firebase_auth.dart';

final FirebaseAuth _auth = FirebaseAuth.instance;

Future<void> register(String email, String password) async {
  try {
    UserCredential userCredential = await _auth.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
  } catch (e) {
    print(e);
  }
```

➔ Google and Facebook OAuth can also be implemented for easy sing-in.

❖ Week-3 (June 14 to June 21)

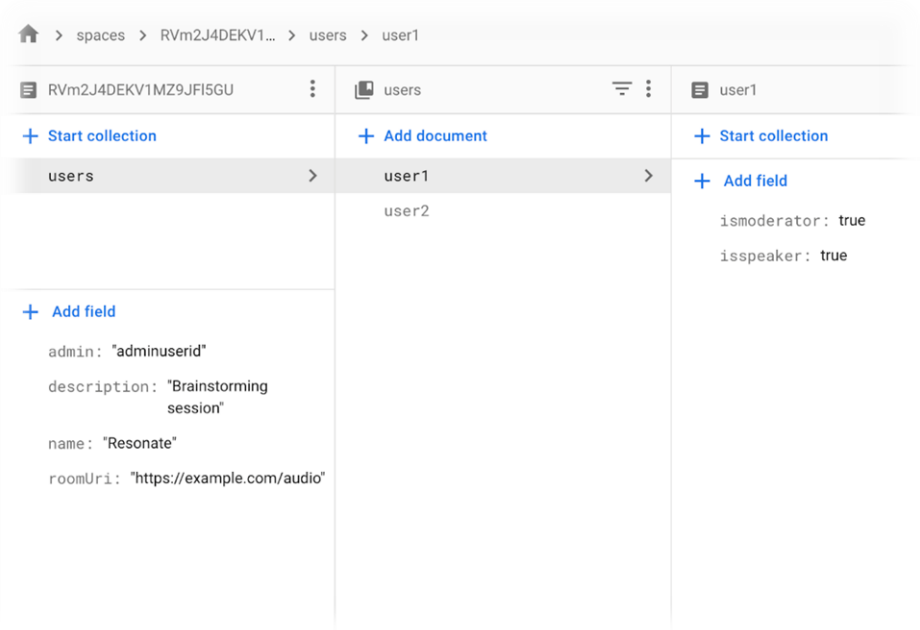
➔ Setting up firebase cloud Firestore

➔ Also, learn about the firestore database and research its querying and scaling abilities.

➔ In our database, we are going to have collections of user, clubs, rooms, user's unique id, participant and followers.

```dart
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Firestore Demo'),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          ElevatedButton(
            child: Text('Add Data'),
            onPressed: () => addData(),
          ),
          ElevatedButton(
            child: Text('Get Data'),
            onPressed: () => getData(),
          ), ],),),);}

Future<void> addData() async {
  try {
    CollectionReference users = firestore.collection('users');
    await users.add({
      'name': 'John Doe',
      'email': 'johndoe@example.com',
      'age': 30,
    });
    print('Data added successfully!');
  } catch (e) {
    print(e);
  } }

Future<void> getData() async {
  try {
    QuerySnapshot querySnapshot = await firestore.collection('users').get();
    for (QueryDocumentSnapshot documentSnapshot in querySnapshot.docs) {
      print(documentSnapshot.data());
    }
  } catch (e) {
    print(e);
  }}}
```

➔ Set up a mock server to test the features of the server.



This is the firebase firestore structure as of now. Final one will be finalized with the mentor. More field will be added based on the finalized structure.

Deliverable:

➔ Documentation of all above modules

❖ Week-4 (June 22 to June 29)

➔ Creating an agora/LiveKit account (as of now LiveKit has been confirmed)

➔ Incorporating LiveKit into the app both for iOS and android.

```
final room = LivekitRoom.connect(
  url: 'wss://your-livekit-server.com/ws',
  token: 'your-access-token',
  participant: LivekitParticipantOptions(
    name: 'Your Name',
  ),
);
```

For creating a room

❖ Week-5 (June 30 to July 7)

➔ Improvements based on the feedback received from mentors, other community members.

➔ Writing unit test for above implemented modules.

❖ Week-6 and Week-7 (July 8 to July 22)

➔ To use Firebase Cloud Messaging to send notification to the user.

➔ Steps:

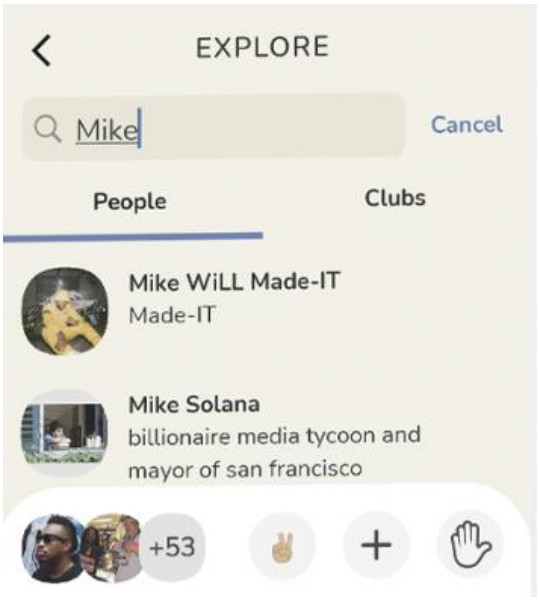
- Setup Firebase
- Add the FCM dependency.
- Request user permission for receiving notifications.
- Get FCM token.
- Configure FCM message handling.
- Send message from Firebase console.

```dart
import 'package:firebase_messaging/firebase_messaging.dart';

final FirebaseMessaging _firebaseMessaging = FirebaseMessaging.instance;

// Initialize Firebase Messaging
void initializeFirebaseMessaging() {
  // Request permission to receive notifications
  _firebaseMessaging.requestPermission();

  // Configure Firebase Messaging
  FirebaseMessaging.onMessage.listen((RemoteMessage message) {
    // Handle incoming message when app is in foreground
    print('Received message: ${message.notification.title}');
  });

  FirebaseMessaging.onMessageOpenedApp.listen((RemoteMessage message) {
    // Handle incoming message when app is in background or closed
    print('Opened app from message: ${message.notification.title}');
  });
}

// Send notification to FCM
Future<void> sendFCMNotification(String recipientToken, String title, String message)
  // Build the notification payload
  final payload = <String, dynamic>{
    'notification': {
      'title': title,
      'body': message,
    },
    'to': recipientToken,
  };

  // Send the notification payload to FCM
  final response = await _firebaseMessaging.send(payload);

  // Check the response from FCM
  if (response.isSuccessful) {
    print('Notification sent successfully.');
  } else {
    print('Notification failed to send.');
  }
}
```

- ➔ Testing the integration by sending messages from the Firebase console
- ➔ Add app to the quick tile of the phone for easy access. Users can enable and disable app for quick settings. This blog can be referred for the same.

❖ Week-8 and Week-9 (July 23 to August 6)

- ➔ Setting up a searching feature allowing user to look for room of their choice.
- ➔ Adding a feed feature that displays information about rooms or people based on user's search history, preferences, and followed people.



❖ Week-10 and Week-11 (August 7 to August 21)

Testing and deploying

- ➔ Final testing on devices and deploying in play store.
- ➔ By this time mentor would have approved the design and logo. Before uploading on the play store, I will finalize the icon, banner, and screenshot, and description needed while uploading on play store.
- ➔ Firebase Analytics would also be setup for app. It provides many analytics tools built-in like crashlytics, logging and many more that would help us identify where users are struggling and try to improve in the following weeks.

❖ Week-12 (August 22 to August 29)

- ➔ Improvements based on the feedback received from mentors and other community members.
- ➔ Writing unit test for above implemented modules.
- ➔ Manual exhaustive testing on different devices, emulators
- ➔ Bug fixes, Writing documentation.

❖ Optional Milestone:

- ➔ **Automatic CI/CD for repo for releasing builds**.
  If I get free time between, I work on setting up CI/CD for the repo by GitHub Actions for automating many chore tasks like releasing builds, singing up Apks, etc.
  There are various tools listed on flutter deployment doc like Codemagic and Fastlane that could be used for setting up the workflow. Appropriate tools and actions would be implemented after a discussion with my mentor.

- ➔ **PR and issue template for bugs, features or feedback**.
  User can report bugs, request, and give feedback by ready-made template on GitHub. These template auto-populates the issues/pull-request description field and provide a skeleton framework that contributors can fill out.They help provide a baseline standard of information quality and organisational rigour.

- ➔ Add some more features like enhancing the user experience of people in the room (e.g. Adding comments to people's profile)

❖ **I plan to continue contributing to this project after the GSoC'23**

My background/ Technical skills

I am Akshat Kedia, 2nd year Undergraduate pursuing BTech degree at National Institute of Technology, Tiruchirappalli, India. I started with QBasic and then learned C++, Java, and Python. For last one year I have been learning app development (both Java and Flutter). I created several apps as a part of competitions and project, including a music player named kMusic. I use git and GitHub every day, and I am well acquainted with using them for version control.Moreover, I am a competitive programmer too. Of late, I have been learning cyber security, Kubernetes, and many more skills to boost my performance.

I am also very active in open source, college technical events and hackathons. I have participated in Hacktoberfest'23 , DWOC (Delta Winter of Code), and TRI-NIT Hackathon.

I am a member of TeCOS, the official open-source community of our college, where we participate and at the same the time encourage college students to participate. Being in this community, I got the work on a project called PaiNITTe, an initiative to help student get access to academic resources. My recent focus is on the TeCOS initiative, encouraging the open-source culture in our college. I am very active in open source. I try to participate in many of the open-source events

happening. I participated in KWOC, DWOC, and Hacktoberfest previously and this is my attempt in GSOC.

Projects

Contribution Projects-

1. PaiNITTe

PaiNITTe is an open-source website that provides academic resources for the students of NIT Trichy. The website is designed to offer a platform for students to access various academic-related activities such as past question papers and class tests. One of the key features of PaiNITTe is its database of CTs (Class Tests) and end-semester question papers. This database is constantly updated with the latest question papers, allowing students to access them easily and use them for exam preparation. Overall, PaiNITTe serves as an excellent resource for NIT Trichy students to access academic materials and prepare for their exams. Its open-source nature and collaborative approach make it an ideal platform for knowledge sharing and learning. I have added contributions to it and have added a few features to it. (click to see the contribution)
PaiNITTe is made using Docusauras 2.

GitHub link: https://github.com/Axhaat/painitte

Personal Projects:

1. kMusic

kMusic is an open-source music player designed for music lovers and enthusiasts. It supports a wide range of audio formats, allowing users to play their favourite music files without any hassle thus making it a great choice for anyone looking for a reliable and versatile music player. I had developed this as one of my personal projects.
KMusic is made on Java.

GitHub link: https://github.com/Axhaat/kMusic

2. Jarvis AI

Jarvis AI Project is a Python-based project that aims to provide personalized assistance to users based on their queries. The project uses libraries of python to understand user queries and generate appropriate responses. The functionality of the project includes features such as setting reminders, playing music, searching the internet, controlling smart home devices, and answering general knowledge questions.
Jarvis AI is programmed in Python.

GitHub link: https://github.com/Axhaat/Jarvis-AI

3. To-do App

To-do app is an open-source project designed for personal use. It is a software application designed to help individuals organize their tasks and responsibilities. The app typically offers a simple and intuitive user interface that allows users to create to-do lists, add tasks to these lists, and provides a check box and option to delete once the task is completed. I am also trying to add few more feature like users can also prioritize tasks based on their importance and track their progress as they complete each item on their list and things like ability to categorize tasks, collaborate with others on shared lists, and integrate with other productivity tools.
It is written in Kotlin**.**

GitHub link: https://github.com/Axhaat/todo-app

Contributions to open source

AOSSIE-Org/Resonate - https://github.com/AOSSIE-Org/Resonate/pull/23
Beacon - https://github.com/CCExtractor/beacon/pull/202
Taskwarrior-flutter - https://github.com/CCExtractor/taskwarrior-flutter/issues/142
openMF/mobile-wallet - https://github.com/openMF/mobile-wallet/issues/1368
openMF/mifos-mobile-cn - https://github.com/openMF/mifos-mobile-cn/pull/235
Hacktoberfest -Successfully completed Hacktoberfest 2022.
Processing-android - https://github.com/processing/processing-android/pull/730
CodingPractice-Hacktoberfest - https://github.com/csubhasundar/CodingPractice-Hacktoberfest22/pulls?q=is%3Apr+author%3A%40me+is%3Aclosed

Commitment:

Weekly Commitment: 40-45 hours

During the first two months of my college semester break, I plan to work full-time on this project. However, August will be a busy month for me as I will be returning to campus and starting my college internship season. Therefore, I have allowed for almost a month to complete milestone 4 to make up for any potential time constraints. Although I do not foresee any other gaps or absences, I will inform my mentor in advance in case of any unforeseeable emergencies.

Anyways, I am more than ready to work well past my committed time if needed.

Other Commitments:

Here are some of the things I would do that may help me prevent personal matters from affecting my GSoC performance:
1.Make sure to allocate enough time for my GSoC project each day or week, depending on my project requirements and timelines.
2.Keeping in touch with my mentor regularly and informing them of any personal matters that may affect my project timeline.