Google Summer of Code 2020
CarbonFootprint-Mobile

Contents

Me at a Glance

Basic information

Name: Harshit Jain
Gender: Male
Nationality: India | IST (UTC + 5:30)
Resumé: Link
Address: D-13, Transport Nagar,
Damoh Naka, Jabalpur(M.P.)- 482001

Educational Information

Institute: Shri G.S. Institute of Technology and Science
Degree: B. Tech.
Major: Civil Engineering
Graduation: May' 2022

Contact Information

Email address: harshitjain1309@gmail.com
Secondary Email: harshitjain9014@gmail.com
Phone number: +91 9479756888
Gitter: @iharshit009
Website: iharshit009.github.io

Social Media

GitHub: iharshit009
GitLab: iharshit009
LinkedIn: Harshit Jain
Blog: Medium
Quora: Harshit-Jain-160

Commitments and Availability

The commitments I would have during the GSoC period is:

Looking over the current scenario, my university is closed and examinations would start in May for a **period of 15 days.** I would be contributing around **4 hours daily** during the exam days. Later on, I would spend my time according to preferred working hours(in **IST**):

- **Weekdays** : 04:30 pm - 08: 00 pm, 10:30 pm - 01:00 am = **6 hours**
- **Weekends** : 10:00 pm - 12:30 pm, 04:00 pm - 06:00 pm, 09:30 pm - 12:30 am = **7:30 hours**

I can easily devote **45 hours per week** when my college is closed and at least 32-36 hours per week after that, with more hours during the weekends. I have only 24 academic working hours per week. I have no internships or vacation plans. I will be comfortable and ready to re-plan the timings of my schedule as per the **mentor/admin** availability or any unknown requirements (if any) in the future. I strongly opine that working on this project for this summer will help me learn and explore. My work will help make this app a **production-ready app**, fully ready to be released on the **Play Store and App Store.**

Chosen Idea: CarbonFootprint-Mobile

Project Description

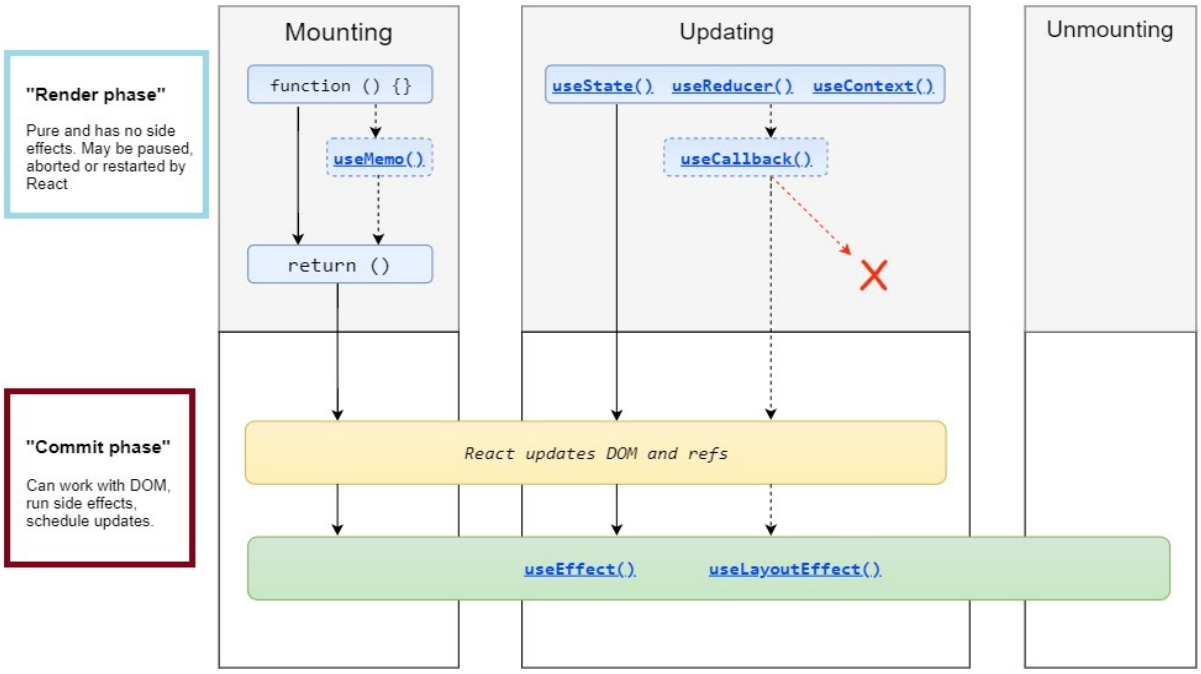
Abstract

The project aims to make a better and fairer hybrid **Mobile Application** to make it easier for every individual to calculate the carbon footprint value, Emission of CO_2, Activity recognition of the user, giving him push notifications on his/her activity. The global aim of this project is to make users aware of their CO_2 usage. Out of many available metrics, one such is a carbon footprint that measures the amount of greenhouse

gas produced by an individual, organization, country. **The carbon footprint allows individuals, organizations, and governments to quantify the contribution of their activities to global warming.** The App is completed and is now in Beta Version, but there is still a requirement for upgrading the existing features, to ease the UX flow and improve the overall UI of the app. **IOS version of the app also** needs to be kept up to date with that of Android.

Detailed Proposal

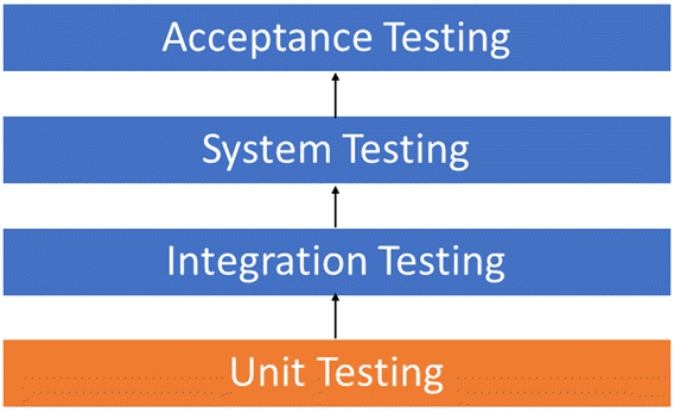
The project currently uses a JS library for Mobile Development that is **React Native** for frontend. The project currently uses **Firebase** for backend support and Google **Maps API** for location services, and the other crucial functionalities include **Oauth** via Facebook, Google, and Twitter. Furthermore, the testing is done with a popular open-source testing library **Jest**. The App is on the urge to be completed with its development phase but major issues/features aren't working properly as expected. This is the lifecycle method for CarbonFootprint App.



In view of the motivation for recreating and **intensifying features** to enhance the project, I would like to propose the following:

1. Debugging the App

The current state of the application has almost completed the project but the major features aren't working well. This summer, we propose to **debug** the complete application, testing it through methodologies like **Unit and Functional Testing**, looking for the lack of user interactivity.



For Unit and functional testing, I would like to propose the use of **Jest** and **Enzyme**, the current scenario of the project does use Jest but could not test the app completely due to many missing test cases and functions. My work with **Jest** can be found here and below is the implementation of **Enzyme** for a "pair of a button" (code snippet from **button.test.js**)

```
import React from 'react';
import {shallow} from 'enzyme';
import Button from './Button';

describe('Button', () => {
    describe('Rendering', () => {
        it('should match to snapshot - Primary', () => {
            const component = shallow(<Button label="test label" primary />)
            expect(component).toMatchSnapshot("Primary button snapshot")
        });
        it('should match to snapshot - Secondary', () => {
            const component = shallow(<Button label="test label" primary={false} />)
            expect(component).toMatchSnapshot("Secondary button snapshot")
        });
    });
});
```

Using ESLint/Prettifier and **husky** for pre-commit tests that will help in resolving syntactical errors for **better commits.**

```
"scripts": {
    "test": "jest",
    "pretty": "prettier --write --tab-width 4 \"src/**/*.js\"",
    "precommit": "lint-staged && npm test"
},
"lint-staged": {
    "*.js": [
        "npm run pretty",
        "git add"
    ]
}
```

An example of the **husky** is shown below for the project Ovuli.

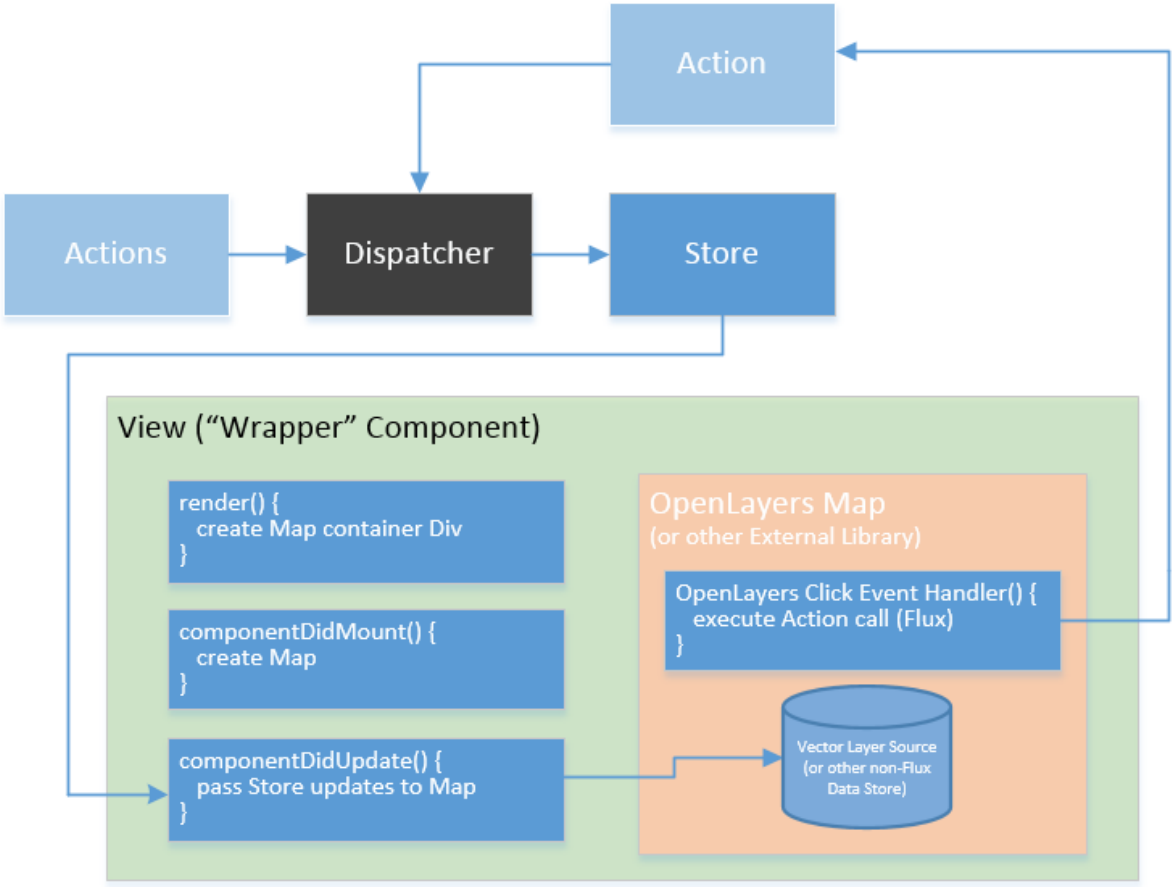
```
harshit@devdell:~/Desktop/projects/ovuli$ git commit -m "test1"
husky > pre-commit (node v12.16.1)
```

```
  ✔ Preparing...
  ✔ Running tasks...
  ✔ Applying modifications...
  ✔ Cleaning up...
[master beed67be] test1
 1 file changed, 1 insertion(+), 1 deletion(-)
```

2. Integrating a free open-source Maps API replacing Google Maps API

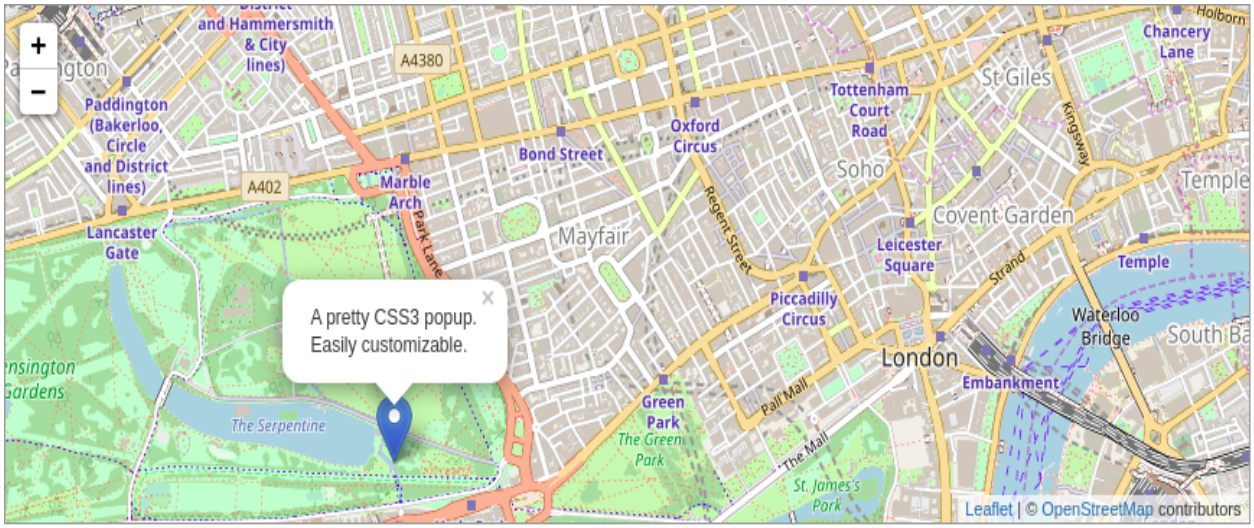
Google Maps API is limited to use for the free tier, so there is an urgent need to replace it with Open source API. Some open-source API's as mentioned in issue #154 is:

1. Openlayers - (Completely free, just mapping services)
I found it very interesting to use due to its amazing storage and view component



2. MapMyIndia- Includes Distance Matrix with Predictive ETA(Completely free)

3. But a much better option for documentation and implementation with Javascript libraries seems to be Leaflet. I have worked with it in one of my projects mentioned here.



The leaflet is used in various implementations that include heatmaps, Route360[0] and various others.

```
import React from "react";
import { render } from "react-dom";
import { Map, Marker, Popup, TileLayer } from "react-leaflet";

const position = [51.505, -0.09];
const map = (
  <Map center={position} zoom={13}>
    <TileLayer
      url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
      attribution="&copy; <a href= # >CarbonFootprint</a> AOSSIE"
    />
    <Marker position={position}>
      <Popup>
        A pretty CSS3 popup.
        <br />
        Easily customizable.
      </Popup>
    </Marker>
  </Map>
);

render(map, document.getElementById("map-container"));
```

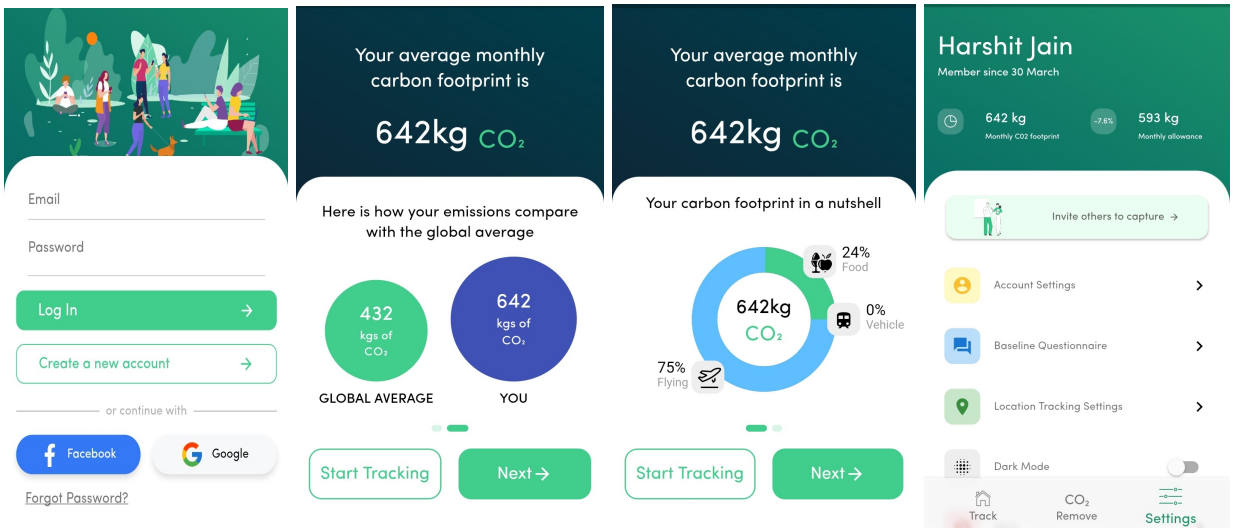
Note- Here, LocationIQ and Mapbox are some other good Map APIs as discussed in #154 but as they are paid APIs, therefore, the above three free versions of these Map APIs are proposed.

3. Synchronization of features with IOS and Android

As we use different react-native packages for implementation of different features in the app, a particular package may work well in Android but the same package may not give desirable outcomes in the IOS and vice versa. Hence, ensuring proper functionality of these features in both the platforms is at the highest priority.

4. Proposing a Better UI to the App for better Interactivity

The current UI seems to be very simple and unappealing, hence there is a wide scope of improvement. Further on discussing with the mentors and can build new designs. Some of the **UI mockups**(ideas from similar other apps) are:



Points that Could be included in the App:

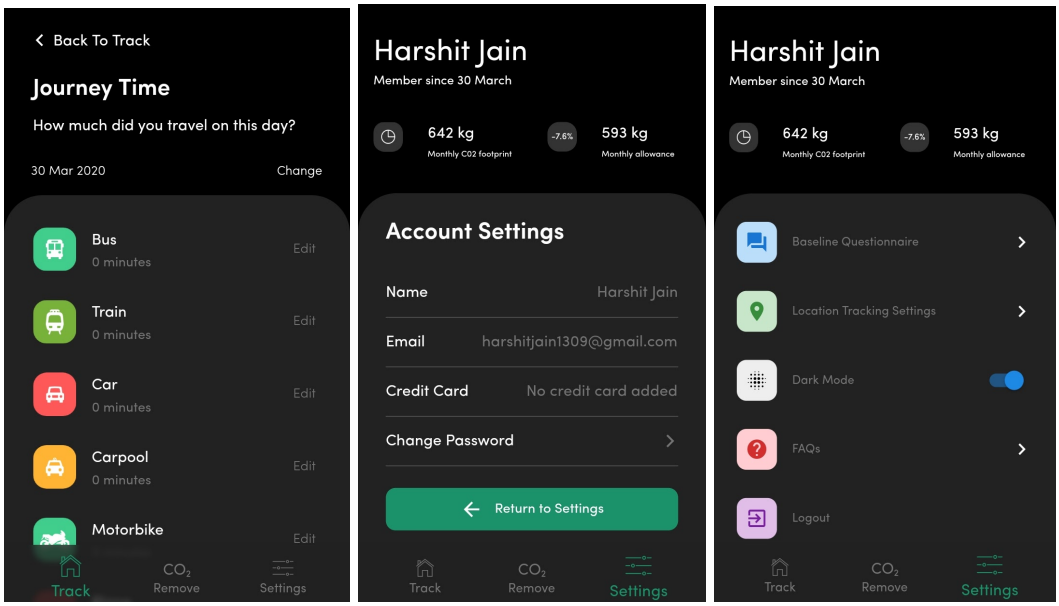
1. **Implementing IFTTT(IF This Then That) in the App(Idea)**
   To get notified over mail or SMS is more friendly than opening the app every time. I propose to use **IFTTT** service(third party application) provides a variety of features through which a user

would get an instant update on his mail/SMS regarding a warning/alert for a **threshold value** of carbon emission**,** could update more according to discussion with the mentor.

2. **Fixing the Oauth integration(Debug)**
   It seems that the current **Oauth** is not working for any of the services. There seems to be a disadvantage as a new user could not register via Oauth services and has to sign up the classic way. Implementation of this may assure more crowds of new users to use the App.

3. **Autocomplete Feature for Adding Friends(Debug)**
   The user needs to type in the complete mail-id of his/her friend, instead, there must be an autocomplete/suggestion feature for easy usability. Regarding the display of searched users, we can use the same component **Flatlist**, but instead of a fixed array, we need to pass the redux state as its data state, so that it can update as soon as new users are fetched. This will also be helpful for identifying different users with the same names, as the **Flatlist** component also displays users' profile pic, from which different users can be distinguished.

4. **Introducing Dark theme to the app(Idea)**
   It has been seen that the majority of users have a great affinity for **dark themes** on their phones, therefore our app needs to provide an option to switch from light theme to dark theme. Some of the examples of proposed dark modes are(only for reference for third-party apps):



Here is a brief on my timeline that will more or less revolve upon the following key points:

Key Points

- To make an appealing and **impressive UI** Idea for the app.
- To **synchronize** the existing features of the IOS with the Android.
- To improve the **Oauth Integration**.
- Proper testing and **Debugging** of the App.
- Publishing the **production-ready app** to the Play Store and App Store.
- Integrating an **open-source Map's API.**
- **Documentation of project**

Note: All the implementation mentioned in this proposal is tentative. I will present a more detailed form of implementation during the community bonding period.

Timeline

Community Bonding Period **May 4 to June 1**	
Week 1 May 4 to May 10	• Interact with the mentors and admin of the projects
Week 2 May 11 to May 17	• Get familiar with the work culture and flow of the project. • Setting up Trello to manage the timeline effectively.
Week 3 May 18 to May 24	• Review existing tickets, make some new ones, and submit patches. • Discuss and finalize Redux implementation.
Week 4 May 25 to May 31	• Update the project to the **latest** versions(IOS and Android)
Phase 1 **June 1 to June 28**	
Week 1 June 1 to June 7	• Discuss the open-source Map API and its implementation in the app
Week 2 June 8 to June 14	• Integrate the Map API with the backend
Week 3	• Develop an intuitive working model of this feature for IOS

June 15 to June 21	and Android
Week 4 June 22 to June 28	• Improve and fix Oauth Integration
Phase 1(**EVALUATIONS**) June 29 to July 3	
Phase 2 **June 29 to July 26**	
Week 5 June 29 to July 5	• Fix the UI Issues and develop a better UI experience
Week 6 July 6 to July 12	• Writing the unit test cases for functionalities in the app (Jest/Enzyme)
Week 7 July 13 to July 19	• Add Dark theme feature for IOS and Android
Week 8 July 20 to July 26	• Implement IFTTT
Phase 2(**EVALUATIONS**) July 27 to July 31	
Phase 3 **July 27 to Aug 23**	
Week 9 July 27 to Aug 2	• Research about different optimization techniques to improve app performance. • Improve the performance of the app.
Week 10 Aug 3 to Aug 9	• Refactor the code. • Buffer period to complete

	any remaining tasks or fix bugs or any unexpected delays
Week 11 Aug 10 to Aug 16	• Extensive debug to all the developed features in the app
Week 12 Aug 17 to Aug 23	• Publish app to Play Store and App Store • Document the project
Final Evaluations	
Week 13	• Writing article on my GSoC experience • Final Submission

I realize that open source is all about **communication** and **collaboration** and I like it that way. Hence, I will be in continuous contact with my **mentor**(@madhavgaba) throughout the program describing the progress and problems I find.

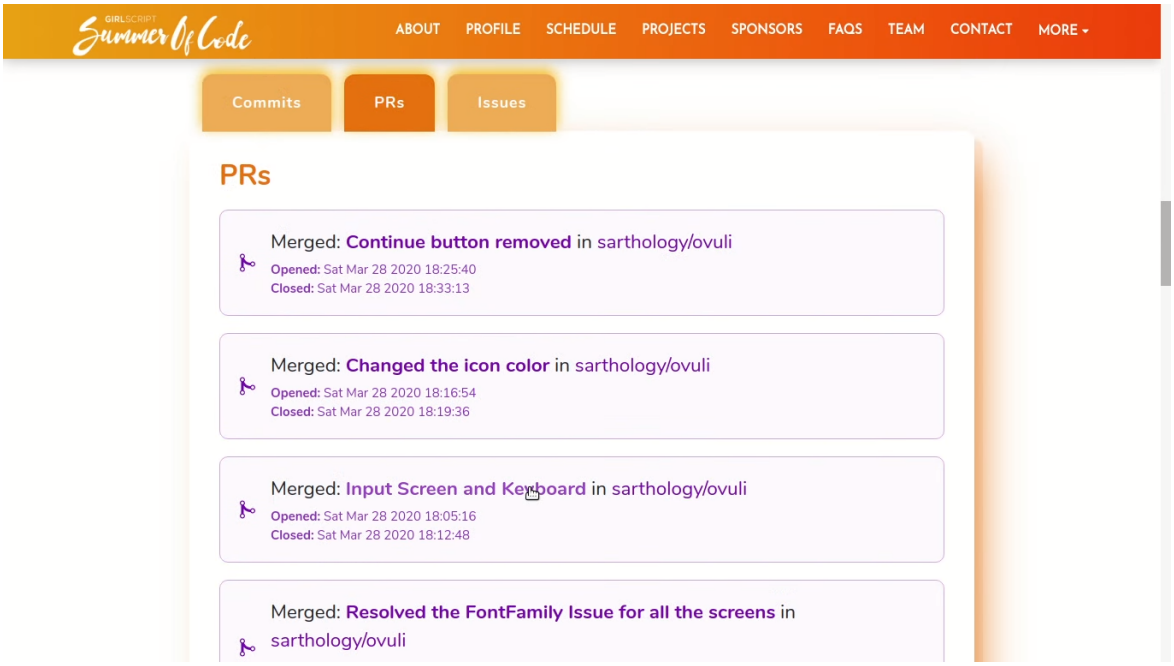
I will prepare a detailed and more fine checklist, which I will host as a **Github gist** during the community bonding period. Thus, in this way, a checklist will be available to everyone with easy accessibility. It will also become easier for a mentor to **add tasks** directly to that checklist and assess them.

Distribution of Total Time (~12 weeks)

Type of Task	Total Time (Weeks)	The fraction of Total Time (%)
Design	1.5	~12.5
Development	5.5	~45.8
Testing	4.5	~37.5
Documentation	0.5	~4.2

Why am I the best person to execute this proposal?

I have also participated in open-source programs previously like GirlScipt summer of code(GSSoC), Summer of Innovation(OSS), etc. So I have measurable experience in contributing to open source projects, working with **Git(VCS),** communicating with **mentors**, committing the reviewed/suggested changes. Here is an image of my contributions from GSSoC:



Contributions to AOSSIE

- **Issues and Merge Requests**
  1. #196: It's very common for API to not give desirable results when it comes to **Autocomplete Searching.** Google provides a Search API but seamlessly there is a mistake in implementing/Activation of Autocomplete Search API. Therefore there can be two approaches:

     A. Use a favourable Open Source API for Auto-Complete that maybe MapMyIndia.

     B. There might be an issue with exceeding the usage limit of Google Search API, incomplete implementation of API or not activating the API Key which could not fetch results.

2. #197: A very common approach if the user is to select the image to upload and then **adjust/crop/resize** the image, but the user can't do either so using a proper upgraded React Native library or using the correct library would give the better results. React-native-photo-Upload an open-source library will give us the required outcomes. This library gives similar experience in both IOS and Android.

3. #199: An unexpected behaviour is observed when the user wants to input a number on the profile page. The keyboard hides the input field, hence creating an issue.

- **Collaboration and Communication Activities**
  - https://gitlab.com/aossie/CarbonFootprint-Mobile/-/issues/184#note_313441517
  - https://gitlab.com/aossie/CarbonFootprint-Mobile/-/issues/174#note_313442778
  - https://gitlab.com/aossie/CarbonFootprint-Mobile/-/issues/154#note_313681927

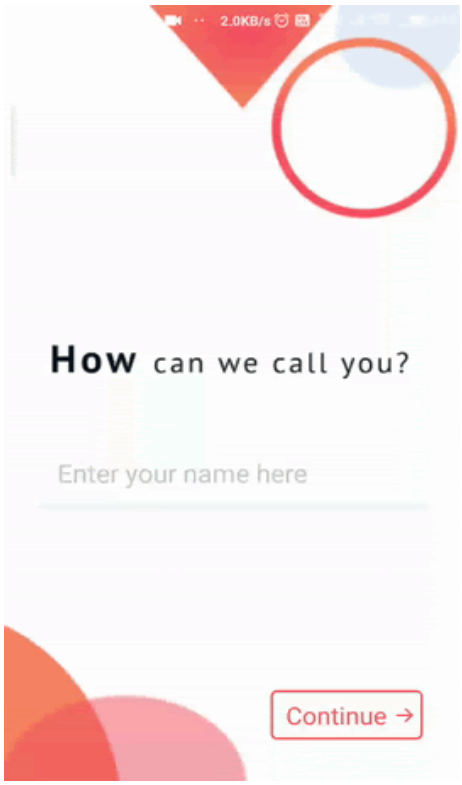
Relevant Projects/Internship I worked on

---

1. **Ovuli**:

Ovuli is a hybrid open-source mobile app built for females to measure the ovulation cycle, so it's an Ovulation Calculation and calculator mobile App built with React Native for **20 Languages**. This app involves the following key points :

a. Developing an **Attractive UI** with Figma and **Oauth** with Google.
b. **Asynchronous Local Storage** of the input data and calculating the Average cycles.
c. **Testing** and **debugging** this application has been an interesting challenge. The testing of this app is done with an Open-source JS testing library **Jest**.



Technology used :

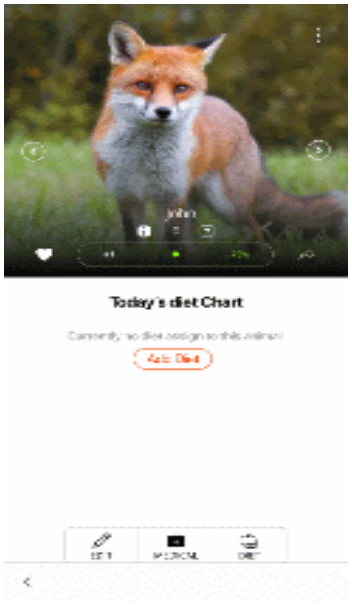
- Frontend
  - React Native **CLI**
- Debugging
  - **Jest**

2. **FaunaNation**:

FaunaNation is a National Level Project under the **"Build for Digital India"** initiative to care, enrich and enhance the current scenario of flora and fauna. I worked as a React Native Developer Intern focusing on the app-building for **IOS**. This app was built under **The Design Institute (TDI)** where I worked with a team consisting of 4

members developing a **Production-level App**. My job included the following:

A. Proposing a **UI Development** Ideas
B. Improving **User interactivity** and Experience
C. Developing a **scalable** and **responsive** App for both IOS and Android
D. **Developing the app** with **Test Driven Development** Approach.
E. Adding **Validations** and API linking with **Redux**



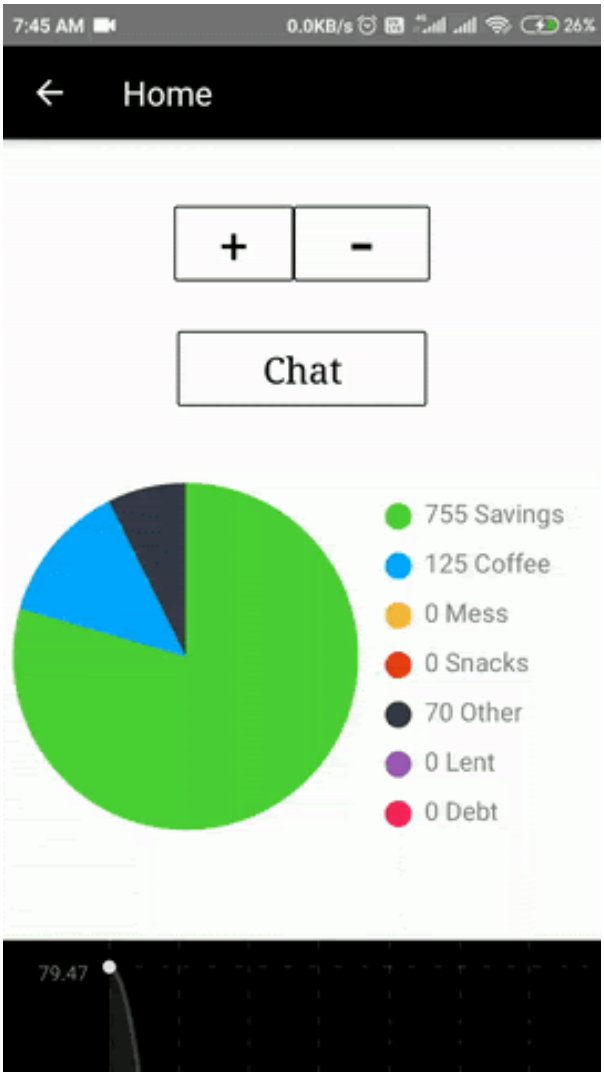
Technology used:

- Frontend
  - **React Native**, Redux
- Backend
  - NodeJS
- DataBase
  - MongoDB
- Target Device
  - **IOS**

3. Artificial Advisor:

Artificial Advisor is your Personal Assistant for tracking your expenses. It's similar to an Expense Tracker but **more than that**. It includes a **chatbot(Assistant)** made with DialogFlow which notices all the user activity related to its **expenses** credited/debited. It has a suggestion module which records, **analyses** the expenses and plots appealing **graphs** for you.

The main feature of this app is its location tracking to capture the **location** of the user, which on analysis can sort expenses according to category and location.



Technology used:

- Frontend
  - React Native
  - Native Base
- Backend
  - Firebase
- Location Services API
  - MapsMyIndia
- Chatbot Assistant
  - DialogFlow

Relevant courses attended and my performance

I am currently pursuing my **Bachelor's** from the **Department of Civil Engineering and Applied Mechanics** at Shri G.S. Institute of Technology and Science, Indore(M.P.). I am currently in my sophomore year of college. I have been working on Computer systems and software programming since my freshman year. The courses studied at my university are:

1. Introduction to C Programming
2. Object-Oriented Programming

Development Skills

Software Packages and Environment

Databases	:	SQLite, MongoDB
OS	:	Ubuntu 18.04, MacOS 10.14
Editor	:	Visual Studio Code, Vim
VCS	:	Git
Design	:	Figma, Adobe Illustrator

Why Open-Source with AOSSIE?

AOSSIE(**The Australian Open Source Software Innovation and Education**) has been constantly working with open source developers around the globe chanting for a change in today's society. Working with developers firmly has gained my concentration on its projects. Out of

which CarbonFootprint Mobile App seems to be very delousing to me. Working in Open source with AOSSIE will help me develop and explore more skills in diverse fields creating an impact globally in the society.

Why GSoC with AOSSIE?

As I am in my sophomore year, planning to gain a lot of knowledge, experiences, and mentorship. Therefore GSoC seems to fill my words and help me get all those things in an amusing way. Working with AOSSIE provides me with an excellent opportunity to utilize my skills to work on an innovative real-world project that can bring global climate change. This project is related to my field of work: React Native, Redux, Firebase so I would be the ideal candidate to work on the CarbonFootprint project. After completion of GSoC, I would like to implement more features to the app and include more projects under AOSSIE and would love to collaborate in other existing projects too.
