



“Front-end and Adaptive Question Rendering”

Table of Content

1. Abstract
2. Goals
3. Design and Implementation
4. Timeline
5. Contact details
6. About me

1. Abstract

This project aims to enhance the user experience and integration in a clinical trial matching application by addressing various goals, including dynamic question rendering, adaptive input formats, a user-friendly interface, integration with the main application, comprehensive documentation, and a prioritization algorithm. Right now the website lacks a good user interface and clear call to action. Having too many options makes it difficult for a user to navigate the website.

The proposed solution plan to redefine the website which involves the elimination of the two scrollbar feature, implementation of a component-based interface in React for organizing patient information effectively by using state management feature of react such as `useState`, `useEffect`, and the categorization of open trials under matched, undetermined, and unmatched options in a more effective way. Additionally, a priority algorithm will be used to dynamically respond to changes in priorities, allowing for a flexible and responsive system. The implementation will also include the rendering of various question types with different input formats and the integration of a customized chatbot to simplify the user experience. The intention of this proposed project idea is to address all those loop holes by implementing a new component-based React UI with enhanced user experience.

2. Goals

Here are some goals for the project-

- ***Dynamic question rendering***
 - It involves creating a ReactJS component that can display questions in a specific order, based on the priority assigned to each question.
- ***Adaptive input formats***
 - The component should handle various question types and input formats, such as multiple choice, free text, and date pickers.
- ***User-friendly interface***
 - Using plain language, providing examples, and avoiding technical jargon. Implement contextual help features, such as tooltips or inline guidance, to provide explanations for complex medical terminology or unfamiliar concepts.
- ***Integration with the main application***
 - To integrate the adaptive question rendering component into the main application, ensuring a consistent and user-friendly interface.
- ***Create comprehensive documentation***
 - Develop thorough and well-organized documentation that explains the purpose, functionality, and usage of the adaptive question rendering component.

- **Prioritization algorithm**
 - To utilize the algorithm that analyzes user responses and assigns weights to questions, prioritizing those that are most relevant to the user's specific situation.

3. Design and Implementation

3.1 Issues with the current website

[\(https://gearbox.pedscommons.org/\)](https://gearbox.pedscommons.org/)

- Homepage is cluttered and overwhelming - The homepage presents a lot of information and options, which can be overwhelming for users. It's not immediately clear what the most important features and resources are.
- Lack of visuals - The website could benefit from more visuals, such as images, icons to make it more engaging and easier to navigate.
- Bad user experience - Two scrollbars on the same page look congested and make it confusing for users to navigate and interact with the page effectively, leading to a poor user experience.
- Overwhelming users with irrelevant questions - Without prioritization, users may be presented with a large number of questions that are not relevant to their specific situation. This can be overwhelming and lead to frustration, especially for users with limited medical knowledge.
- Variable question input formats- The Gearbox website currently supports a limited number of question types, such as multiple choice and free text. This can be restrictive for users who need to provide more complex or nuanced information.
- Limited search functionality - The search bar is not prominent, and there are no options to filter results by category or topic. Filter option is confusing for users as they don't understand on what basis filter is done.

- Inconsistent navigation - The navigation menu changes throughout the website, and some sections have different submenus. This can be confusing for users.
- Lack of clear calls to action - It's not always clear what actions users should take on the website. There are a few prominent calls to action, and some buttons are labeled with generic text.
- Not fully optimized for mobile devices - The website is not fully responsive and can be difficult to use on mobile devices. Some elements are too small or difficult to tap on.

3.2 Enhancements and solutions

- To optimize the user experience, the plan involves eliminating the two scrollbar feature, and implementing a component-based interface in React for organizing patient information effectively. This approach allows clinicians to input necessary details through priority-based filter groups. Upon clicking the search button, relevant open trials will be displayed.
- Open Trials will be further categorized under matched, undetermined, and unmatched options in a horizontal bar format. Clinicians can then interactively explore specific trial data by clicking on each respective option.
- By utilizing React hooks like `useState` and `useEffect` in the implementation of a *priority algorithm*, we can efficiently manage data interactions and enable real-time updates within the components. This approach ensures that the algorithm can dynamically respond to changes in priorities, allowing for a flexible and responsive system.
- To enable the rendering of various question types with different input formats, we can use React's state management and conditional rendering capabilities such as creating separate React components for each question type, each tailored to render a specific input format and utilize state management to track the selected question type.

- Integrating a customized chatbot into our clinical trial matching application, we can provide users, especially those without medical expertise, with a convenient and intuitive way to navigate the complex process of finding relevant trials. The chatbot can offer personalized guidance, answer questions in real-time, and simplify the technical terms and overall user experience, making it more accessible and empowering for all.

3.3 Proposed Implementation

Now that the issues and problems with the website are clear with the proposed solutions, I'll start implementing the features as required by keeping the theme, rules and regulations of the website in mind. Right now the two scrollbars create a lot of confusion and give a bad first impression to a user.

I'll suggest having a component-based interface with disease filter group and basic information like demographics about a patient in the sidebar. This will automatically reduce ample amount of scrolling under patient information section. After choosing a disease from the sidebar, Additional questions and other questions such as "Treatment and Exposure", "Organ Functions" etc will appear on the main interface using the React-based accordion element with the implementation of priority-based algorithm implemented for each heading using react hooks such as useState and useEffect. The options under each question will have various question types with different input formats.

After filling all the required answers, the user has to click on "click to see open trials" button which will then redirect to a different component of "open trials" preferably on the same page or a new page as instructed by the mentors. In the open trial component, a user can click on the given three options - "Matched", "Unmatched", and "Undetermined" and see the results accordingly. Here is the basic wireframe of the website pages.



GEARBox helps clinicians and nurse navigators
find clinical trials for their patients.

Refine your search

Filter by diseases

- ☒ Acute promyelocytic leukemia (APL)
- ☐ Osteosarcoma
- ☐ Yolk Sac Tumor
- ☐ Choriocarcinoma
- ☐ High risk Neuroblastoma
- ☐ Embryonal Carcinoma
- ☐ Down syndrome

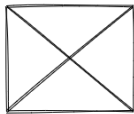
Demographics

Patient Name

Biological sex

Age

Apply filters



ASK ME FOR HELP!

Additional questions related to disease

☐ ☐ ☐ ☐

Treatment and Exposure

☐ ☐ ☐ ☒

Organ Function

☐ ☒ ☐ ☐

Biomarkers

☐ ☐

CLICK TO SEE OPEN TRIALS

Open Clinical Trials

Connect with us



About gearbox

our mission

Data policy

Terms &

Join the community

Researchers

Participants

News

Get

Contact

FAQs

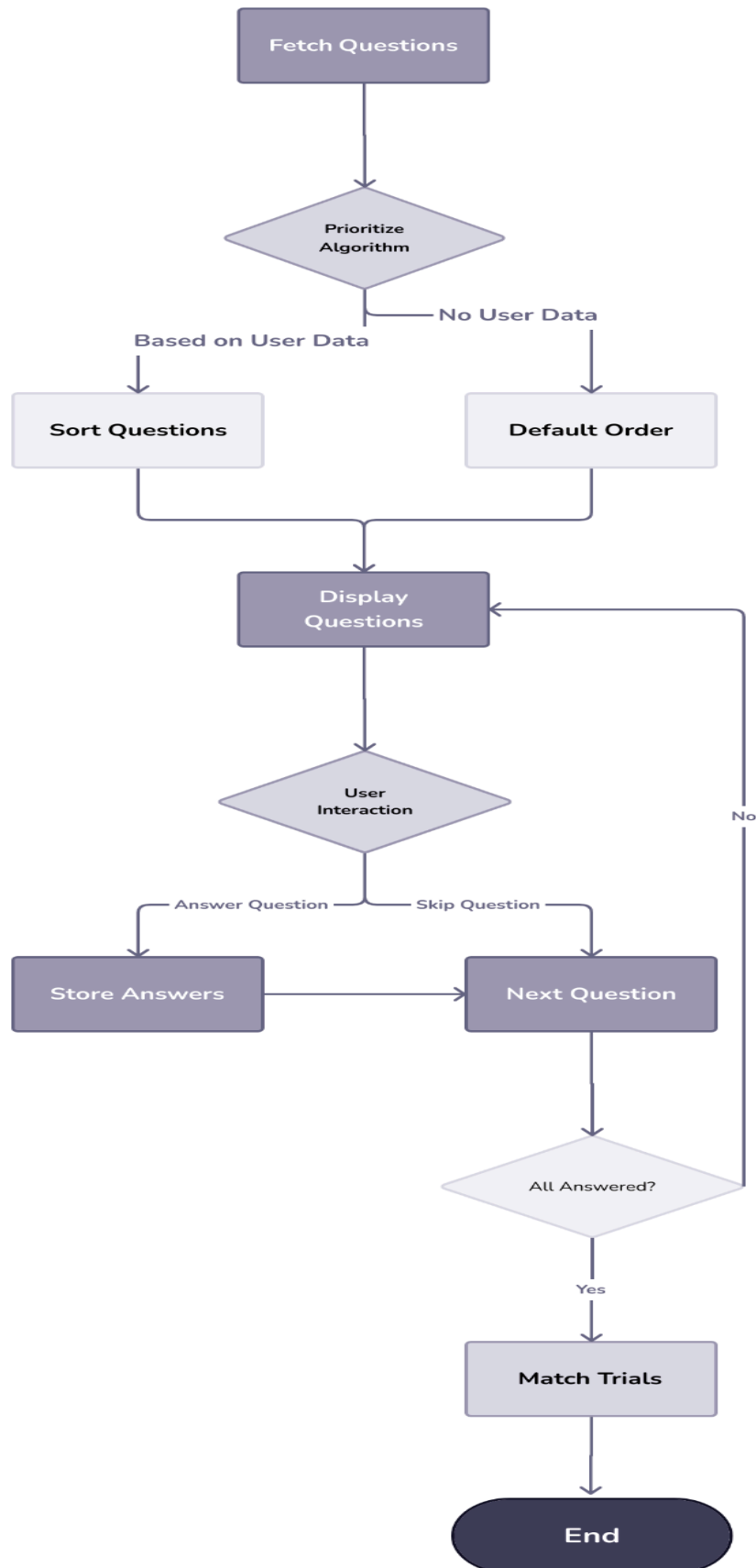
Mobile App



Coming to implementation of the priority algorithm, it will involve following properties:-

- **Data Collection and Prioritization** - Upon inputting patient information, the priority-based filter groups could facilitate the categorization of diseases, determining the priority of each entry based on specific criteria.
- **State Management with React Hooks** - The priority data could be managed using React state hooks, such as `useState`, to track and update the priority status of the diseases as entered by clinicians.
- **Conditional Rendering** - Using React's conditional rendering capabilities, the algorithm could dynamically display the filtered data based on the priority status.

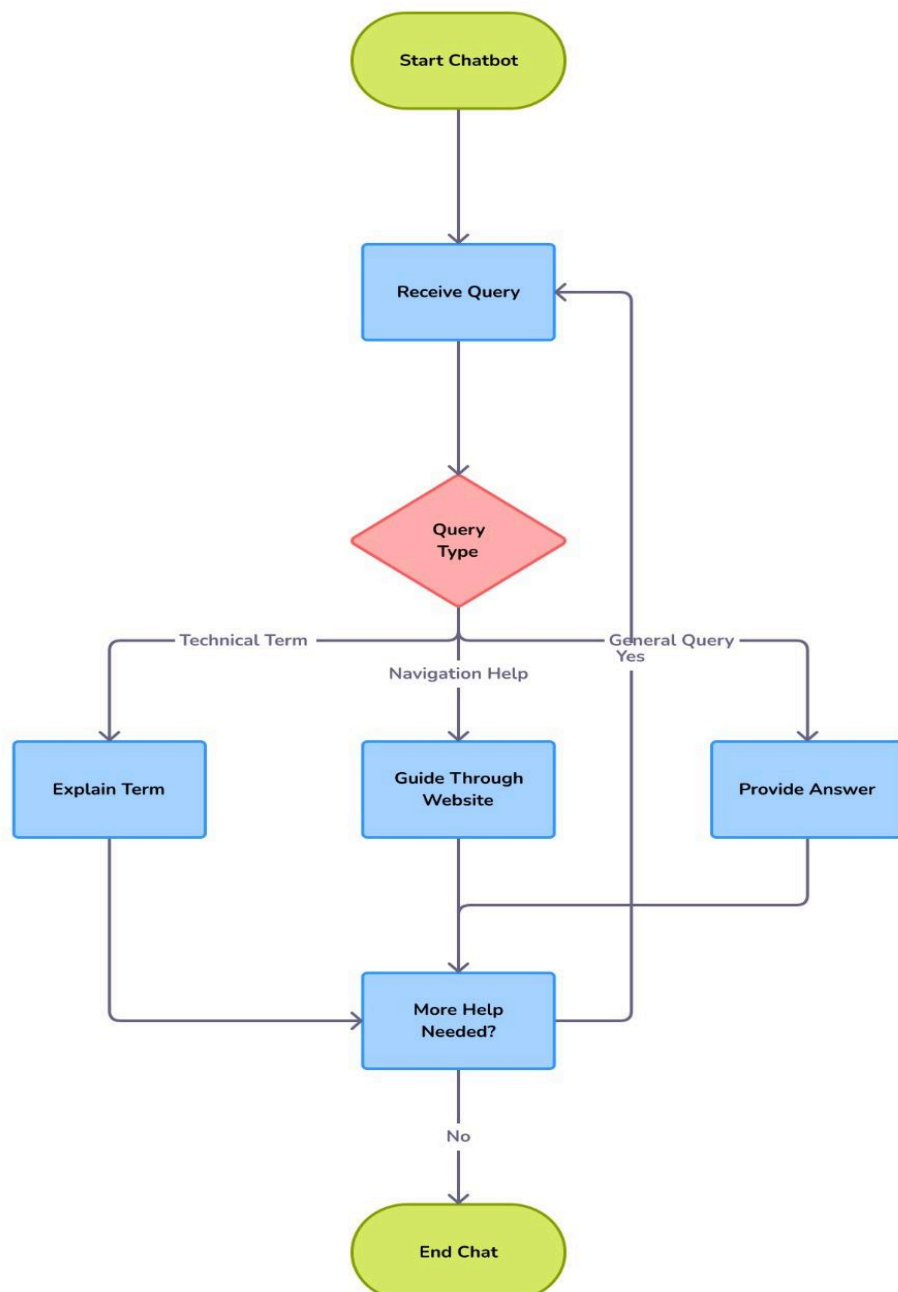
- **Real-Time Updates with useEffect** - The useEffect hook can be deployed to enable real-time updates when the priority of a disease is altered. Here is the explanation of how priority algorithm will work.



Going with the current trend and enhancing the user experience, I also would like to implement a customized chatbot feature for the website which will be used to:-

- simplify the complex technical jargon to understandable language, particularly for individuals without medical expertise.
- Navigate through the website and get an idea about various options and features.
- Give answers to general queries as well.

Here is an explanation of how the chatbot will work.



I have a basic idea about how a chatbot works and how we can implement a chatbot in a website. I am not very well aware of how machine learning algorithms work in chatbots but I am very curious to gain knowledge about it and use it in a real-world project. So, after incorporating all these features we can expect a website like this. This website is subject to change according to the needs of the project and mentors.



Additional questions related to disease

Common observations

Dropdown

Treatment and Exposure

Common observations

Organ Function

Common observations

Biomarkers

Common observations

[CLICK TO SEE OPEN TRIALS](#)



MATCHED

UNMATCHED

UNDETERMINED



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



PEDSHEMAML0008



Title

Relapsed Pediatric AML to Determine the Safety and.....



Open Clinical Trials

Connect with us



About

Our mission

Data policy

Terms &

Join the

Researchers

Participants

News

Get

Contact

FAQs

Mobile App

3.4 What I have done so far

- Analyzed the website properly and realised that the website contains a lot more data than a usual website including info button having information for each trial, various types of questions under disease and other categories, a description of each trial and many more so it requires careful consideration and proper implementation of the features.
- Searched about how we can implement features such that we can compare “Patient information” and “open Trial” dynamically and simultaneously without using multiple scrollbars.
Found these websites useful
<https://ux.stackexchange.com/questions/29609/solve-multiple-scroll-bar-on-one-page>
<https://www.nngroup.com/articles/scrolling-and-scrollbars/>
- Started gaining more knowledge about how we can implement customized chatbot in a website. There are some react based libraries for implementing such chatbots like React Chatbotify or popular NLP (Natural Language Processing) libraries like Dialogflow, IBM Watson, or Wit.ai. enabling language understanding and response generation for the chatbot. These can be integrated with React using their respective APIs.
- Conducted a user research on my college students to know about their user experience. For people with less medical knowledge had a very difficult time figuring out what the website will do especially using two scrollbars made them uncomfortable while using mobile phones.

4. Timeline

I'm able to dedicate 40-45 hours per week on GSoC this summer. I don't have any other commitments except learning and implementing the required work as I also have to increase my skills and knowledge especially related to implementing the chatbot. It will also help me to achieve the goals of the project. I'm ready to put in extra hours if required. I feel a good motivation to contribute to this project as I am really interested in learning to handle so much data at once on a website and also want to explore how a chatbot is implemented from scratch in a website

and how we can train it for medical usecases. I'll try my best to complete my work by 12 September 2024 keeping in mind that I do not need to backtrack the already completed tasks. After every milestone, I'll write a blog about my progress and experience with the project on hashnode. I am more than ready to work well past my committed time if needed.

<p>May 1 - 26 (Community Bonding Period)</p>	<ul style="list-style-type: none">● Get acquainted with mentors and project details.● Get more involved in community and talk to fellow contributors.● Read relevant documentation thoroughly.● Understand the existing codebase and project requirements.● Finalize the website interface after discussing it with mentors.
<p>May 27 - July 8 (Coding officially begins)</p>	<ul style="list-style-type: none">● Week 1-2(May 27 -June 9)<ul style="list-style-type: none">○ Begin the development of the React component for dynamic question rendering.○ Redesign the website and its features.○ Focus on initial implementation of the adaptive question rendering component. Using react properties like useState, props, mapping, I'll focus on the implementing dynamic interface of the website.○ Will implement various input field for questions.● Week 3-4(June 10 - June 23)<ul style="list-style-type: none">○ Refine the dynamic question rendering based on priority and input formats.○ Start integrating the developed components with the main application.○ Conduct an initial round of testing for the integrated components.○ Address any early feedback and make necessary improvements.

<p>July 12 - August 19 (Midterm Evaluation Deadline)</p>	<ul style="list-style-type: none"> • Week 5-6(July 12 - July25) <ul style="list-style-type: none"> ◦ Begin the development and implementation of the prioritization algorithm. ◦ Test and refine the algorithm to ensure accurate prioritization based on user input and its seamless integration into the system. ◦ At this point, my main focus will be on implementing priority algorithm properly and testing within the website. ◦ Gather feedback from mentors for initial refinement, do changes suggested by mentors. • Week 7-8(July 26 - August 8) <ul style="list-style-type: none"> ◦ Address any bugs, performance issues, or usability concerns identified during testing. ◦ Focus on finalizing all project documentation, including user guides and technical specifications. ◦ Prepare for the final submission and evaluation process.
<p>August 19 - September 3 (Final Week)</p>	<ul style="list-style-type: none"> • Submit the final work product for evaluation. Complete the final mentor evaluation and reflect on the overall project journey. • If everything goes well, I would like to start seeing possibilities of working on implementing chatbot
<p>Optional Milestone (Extended deadline)</p>	<ul style="list-style-type: none"> • At this stage I would have learned about the implementation of chatbot as I'll be learning it side by side during previous months, so I'll begin by implementing the feature and set it for specific use-cases related to medical requirements.

5. Contact Details

Name - Vaishnavi Maheshwari

University - National Institute of Technology, Tiruchirappalli, Tamil Nadu, India

Github - [vaishnavi192](#)

LinkedIn - [Vaishnavi Maheshwari](#)

Twitter - [@vaishu_uff](#)

Resume - [Here](#)

Email - vaishnavi19official@gmail.com

Contact number - +918878039669

Location - Tiruchirappalli, India, IST(GMT+5:30)

6. About Me

Hi, I am Vaishnavi Maheshwari, currently a second-year undergrad student at [National Institute of Technology, Tiruchirappalli, India](#). I am a keen learner and an open-source enthusiast with a strong interest in web development.

My skills include Javascript, ReactJS, Python, Version control(Git/Github), MySql, flask, data analysis, sqlite3. docker.

I recently completed Build@Mercari'24 training program conducted by Mercari, Tokyo, Japan. I was among one of the 6 candidates selected from my country for this program where we learned about backend development, docker, CI/CD, workflow and pipeline, API development, data fetching, data analysis etc. I created different routes, API endpoints to fetch data from the database based on HTTP methods, Implemented error handling, Made a website using react and typescript etc.

I also attended Global Open Source Cohort in March 2024 where I interacted with different maintainers of various open-source programs such as Outreachy, Fossasia, MLH, LFX. They were discussing about how data is going to increase exponentially in coming years due to advancements in AI. From there I got interested in gaining knowledge about data management.

I am a proficient web developer and I want to further deepen my knowledge in extending my limits by gaining knowledge about scalability and data management for huge data-centric websites like gearbox.

I am also very curious to learn about implementation of a chatbot and train it on specific usecase of medical trials as I feel it can have very good impact on a

website like Gearbox which has a lot of technical information making it difficult for common people to understand. So, this project perfectly aligns with my techstack.

Here are some of my projects-

- [RC_NITT](#) - Restructured and made the website dynamic using React, javascript.
- [Hacknitte](#) - A website to improve coding culture of college by creating a unified dashboard by integrating leetcode, codechef, codeforces user ids. Used various coding platforms API and successfully implemented for various use cases. Used React, mysql, spline3d, Bootstrap, MantineUI.
- [Youtube Clone](#) - I was experimenting with adding new features in my youtube clone out of curiosity as I am fascinated by how youtube handles data and want to explore the technical side of it. I got to know about various aspects of API implementation by using [Youtube data API](#). Used state management systems like useState and useEffect for handling response and api fetching.
- [Student Internship Portal](#) - Made a website to make it easier for first and second year college students to get alumni-based internships. Used Firebase, Google form API and developed different routes connecting to companies website.
- [Transfinitte](#) - Made a Website for college flagship 42 hrs hackathon. MantineUI, MaterialUI, spline3d, figma etc for designing frontend.
- [Guess word game](#) - Created an app where users have to guess words based on hints provided.
- [Hurdle game](#) - Developed a game where the user has to tap on the screen to avoid hurdles.
- [Settle payment app](#) - Made a payment settling app to make group transactions easier.
- [Decentralised crowdfunding app](#) - Made workable backend of crowdfunding app using solidity.

Apart from this, I am a huge fan of open source in general and how it works by sharing knowledge and collaborating instead of competing with others, I also love to share my knowledge and experiences with the help of my YouTube channel - [@vaishu_uff](#), where I share about my college experiences, technical opportunities and my projects. I aim to make more content related to tech and

open source in future to increase awareness in my country about the possibilities of open source opportunities. I am really fascinated about how YouTube handles so much data per day which made me more curious to gain knowledge in this field.

In general, I am also very active in open source, I try to participate in as many open-source events as possible, college technical events and hackathons. I participated in SIH 2023, DWOC, [Delta Winter of Code](#) (open source competition like GSOC by students of NIT Trichy) and [TRI-NIT](#) Hackathon. Got 1st rank in DSA contest for girls, and 8th rank in contest among 300 students, hosted by WIN-NIT and Spider, coding Club of NIT-T. Contributed to various organizations like processing, oppia, sugarlabs. Completed Girlsript summer of code'23 and hacktoberfest'23. Participated and mentored in DWOC. I contacted over 60 organizations and maintainers to participate in DWOC. Took part in Outreachy'24, Codeheat challenge by Fossasia. Part of Red Panda community. Attended Global Open Source Cohort in March'24.

I am a Microsoft Learn Student Ambassador, helping students of my college and member of Google developers students club (GDSC). These programs enhance students' employability by offering training in skills not usually taught in academia.

Being a coordinator at [Technical Council \(NIT Trichy\)](#), I coordinated in conducting our college flagship hackathon, Transfinitte. Worked on various campus development initiatives like Hacknitte website, Painnitte(open source question papers uploading platform for college). My recent focus is on the TecOS initiative, encouraging the open-source culture in our college. We conducted workshops on open source, Git/Github to improve culture in college.

Thanks!

