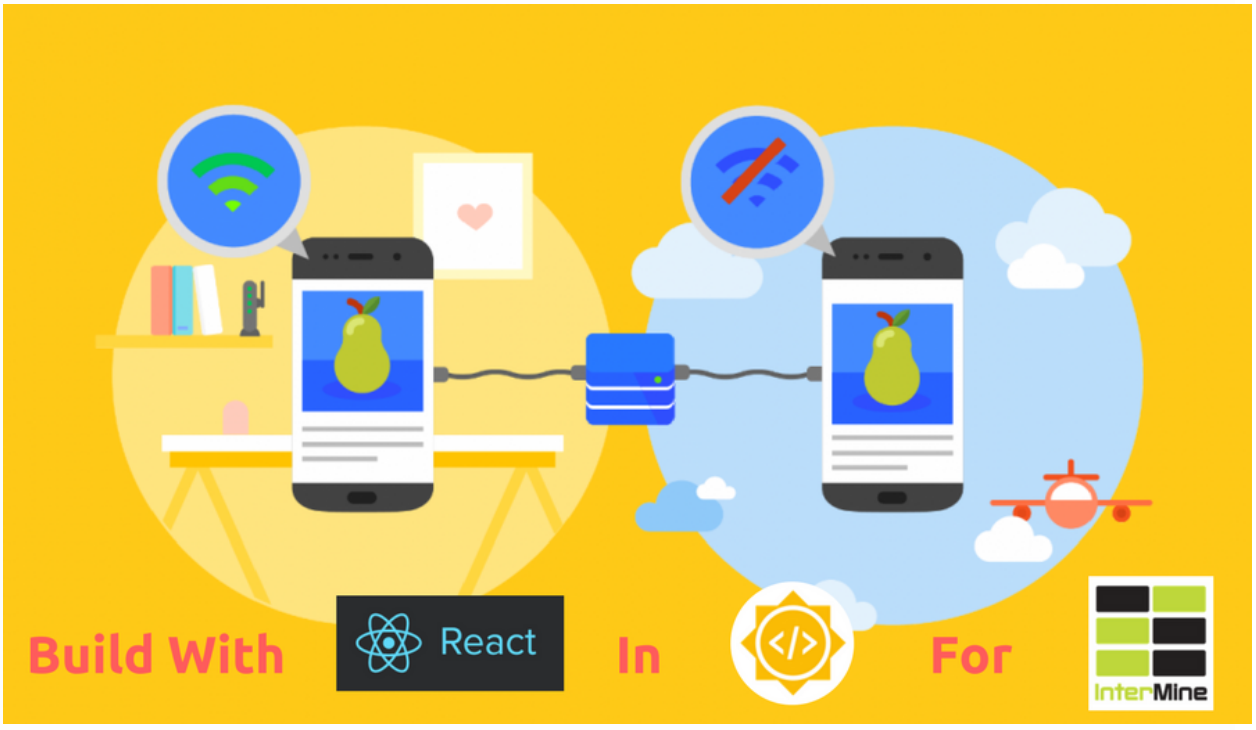
Cross Intermine Search Tool

(as PWA)



Akshat Bhargava (@akshatbhargava123)

About Me

I am Akshat Bhargava, B.Tech (CSE), 1st year, HMR Institute of Technology and Management, New Delhi, India. I am an open source enthusiast and primarily a MEAN and MERN Stack Developer. I love to play around JavaScript and its frameworks. I also have been working into some other fields/frameworks like PWA (ionic-angular), React Native (cross platform mobile apps), NodeJS (express framework), Django, Flask, AI (primarily DL), etc. for more than two years. Apart from all these, I've worked for an NGO as a Hybrid Mobile App and PWA Developer (ionic framework), a development company LogicSquare Technologies as App Developer (MERN Stack) and have many personal projects in same.

Email : akshatbhargavadeepa@gmail.com
Github : github.com/akshatbhargava123

Introduction

From the day organisation list is released, I've been exploring a lot of projects from many different organisations and during this exploration I found Intermine.

I started studying Intermine as 'Cross Intermine Search Tool (abbreviated as CIST further)' project interested me. I contacted the mentor(s) (@nadia#1254) and got to know about the currently made mobile application serving the same search tool consuming different mines' API we now require on web. I have been active in community so, I know what more functionalities we can offer and how we can use the power of PWA to make a performant end user PWA.

NOTE: I've shared a detailed timeline link in *Hacking Timeline* section and end result specs of the project in Project Specs section. Kindly jump if you're already familiar with PWAs. Hacking Strategy here and yes, the UI mockups!

GSoC Project:

Here I'm proposing Cross Mine Search Tool as a Progressive Web App. Below you can see how will making a PWA enhance the quality of this tool.

- **Responsive** - Fits any form factor: desktop, mobile, tablet, or whatever is next.

  Benefits provided to *CIST* by this feature: *We just need to write and deploy code once. It will be easy to manage and upgrade. As our PWA will be a responsive Web App that means, Irrespective of device resolution or size, CIST can be viewed using any type of device. No need to create mobile version separately. I'll be taking care of responsiveness of CIST from start using grid systems wherever needed.*

- **Connectivity independent** - Enhanced with service workers to work offline or on low-quality networks.

  Benefits provided to *CIST* by this feature: *PWA uses power of service worker to cache files and works in the background to provide a smooth experience on every type of internet Connectivity. Even offline. So, any user can access CIST on any type of network. Also, even when fully offline we'll store the search query and send a native (browser or mobile) notification to user whenever the internet connectivity is back with the search results. This is discussed further also with code snippet in project specs.*

- **Re-engageable**- PWA can send push notification, makes re-engagement easy through features like push notifications.

  Benefits provided to *CIST* by this feature: *We can easily generate push notification on events like when the device comes online again and search results are ready, etc on the platform. It will provide the same experience as that of app. It will increase productivity of the tool.*

- **Installable** - Allows users to add apps they find most useful to their home screen without the hassle of an app store.

  Benefits provided to *CIST* by this feature: *When user will first come to CIST then we can provide a option to create a shortcut icon (just like native's app icon) on its device homescreen. So that, next time it can launch its app right from homescreen and can have a native app like experience.*

(I referred Google Developers blog to understand the whole concept of PWA :https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/)

What About Hacking Strategy ?

I am going to use Javascript's most heroic UI development library, **ReactJS** with it's great supporting libraries like **react-router** for navigation system in app, **redux** (global state management), **material-ui-next** for beautiful user interface.

My hacking strategy will include breaking the whole app development process into 3 parts and exploring each field as much as i can:

- **UI/UX:**

  In this part, I will majorly focus on building UI / UX components (react components).

  I'll first design a detailed structuring of components (smart and dumb) and then get to code them by studying how can I maximise the code reuse and use the power of material-ui for beautiful components like search input.

  Specific Technologies that I will be majorly playing with: **Material-UI-Next** and **React's Component API**.

  Useful links:

  1. https://material-ui-next.com/
  2. https://reactjs.org/docs/components-and-props.html

  I'll try to make reusable components as much as I can.

- **Functionalities / Services:**

  Here, I will broadly focus on application navigation system, data flow and developing services that will be able to fetch data from all different mines and rendering it as result.

  Specific Technologies that i will be playing with : **TypeScript**(ReactJS), **Intermine Registry**, JS based libraries like **axios**, **react-router**, **redux** and **redux-saga.**

  Useful links:

  1. http://registry.intermine.org/
  2. http://www.typescriptlang.org
  3. https://github.com/axios/axios

4. https://reacttraining.com/react-router/web/example/basic
5. https://redux.js.org/
6. https://github.com/redux-saga/redux-saga

- **Performance :**

  At the end, I will be focusing on performance of the app and writing unit tests. I will try to reduce the app size as much as I can by using libraries like gzip. I will try my best to get a good lighthouse score. Also, if needed i will use server rendering to make it fast as much as possible (for this i will be using Next.JS).

  I'll follow PRPL pattern for providing the best out of the end app with no point giving harm to UX.

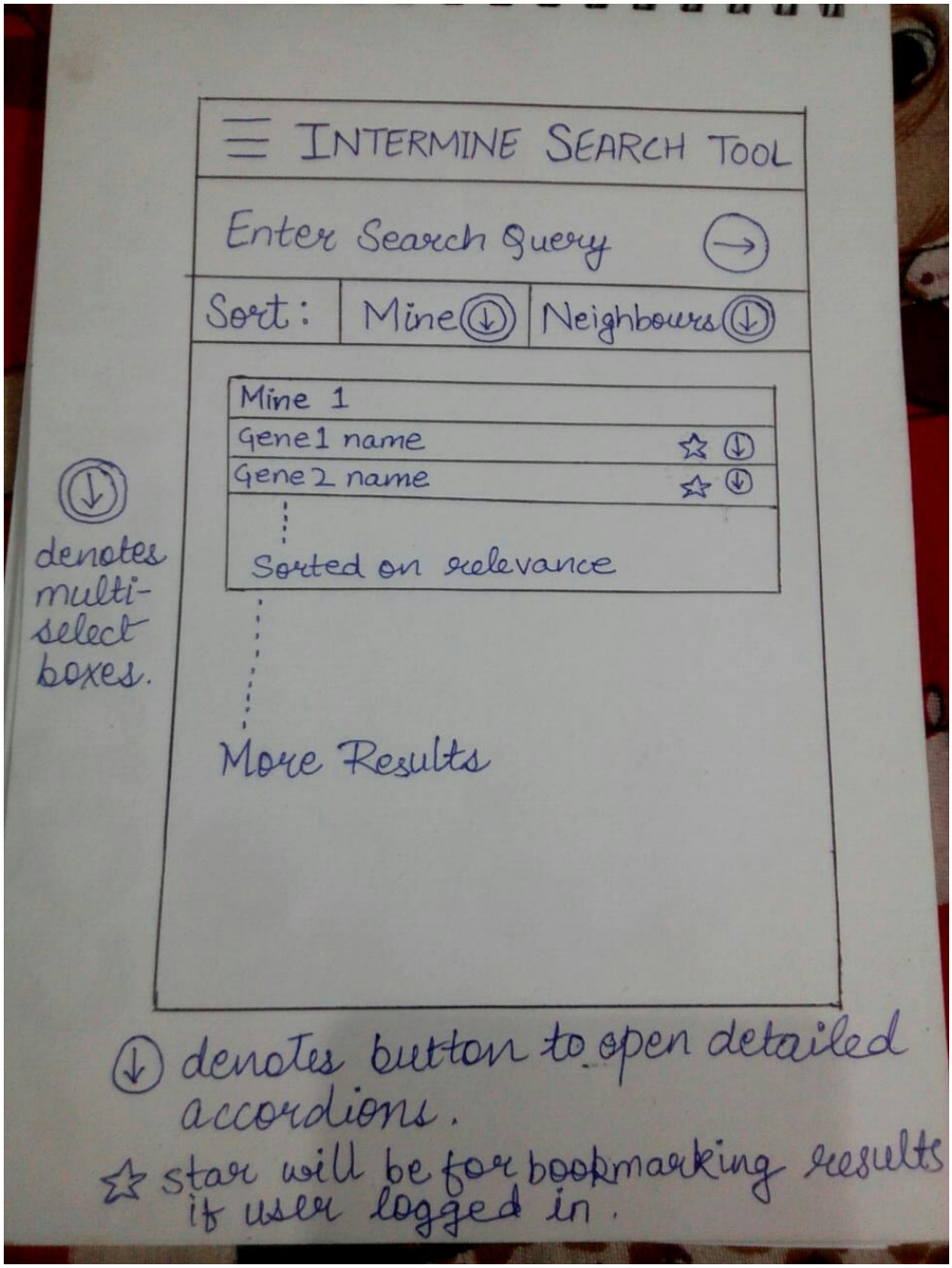
  Specific Technologies that i will be playing with: **GZIP**(for compression), **Next.JS** **(**Server Rendering), **Lighthouse**, **JEST** for writing testing, **PRPL pattern** and much more.

  NOTE: I've follow all best practices of web software development like DRY, PRPL pattern, think and write strategy, etc so we'll have a better end user as well as future developers' experience.

UI Mockup Ahead...

Please go to next page...

UI design basic mockups (Search Page)



Here's a link to flowchart of how exactly things will go:

Detailed Timeline:

https://drive.google.com/file/d/1GmQylHmdsThORJkCAB7aXDH4YegvdxQ1/view?usp=sharing

Let's get specific about major features

- **Search and see results**:

    1. User will be able to enter a search term
    2. The search query runs through all the mine APIs (APIs' base url will be fetched from registry)
    3. The results of each mine start rendering to the list of that specific mine sorted in the order of *relevance (see UI mockup, we'll show the list of each mine separately).*
    4. Giving user accessibility to limit results to a certain *neighbour* by selecting neighbours from a select box which fetches *neighbours* from registry.
    5. Also, we'll provide pagination for user and an option to limit search results in the start even before searching.

Code snippet in nutshell of how will we hit mine APIs using promise based http library *axios*:
`renderToDOM()` is custom method which is given results and name of the mine where results are to be added.

```
axios.get('http://registry.intermine.org/service/instances')
 .then(response => {
  const { instances } = response,
        query = 'BRCA1', // will get this from search input
        endpoint = `service/search?q=${query}`;
  instances.forEach(mineInstance => {
   axios.get(instance.url + endpoint)
    .then(res => {
     const { results } = res;
     this.renderToDOM(mineInstance.name, results);
    });
  });w
```

```
});
```

- **Magical Search (Offline state handling)**:

    1. If application is offline or in very low network conditions, we'll prompt user to notify when the result is ready (when device receives connectivity again).

    2. If user agreed to the same, we'll send user a notification on the relevant time which doesn't require our web app to run even in background due to service workers. Below is a little code snippet of how we will achieve the send notification on search results ready.

```
self.addEventListener('sync', event => {
        if (event.tag === 'searchOffline')
          event.waitUntil(getSearchResultsAndSendNotification());
});
```

`getSearchResultsAndSendNotification()` should return a promise indicating the success/failure of whatever it's trying to do. If it fulfills (we get search results), the sync is complete. If it fails (we fail due to problem in internet connectivity), another sync will be scheduled to retry. Retry syncs also wait for connectivity, and employ an exponential back-off.

Refer: Background Sync API and Web Notifications

- **Addition/Removal of specific mine from mine list:**

    1. We'll provide a user specific page which will list out mines from registry by default.
    2. The user will then be able to select or deselect specific mine they want to include while searching or not.
    3. The user will also be able to add a new mine giving it a custom name and providing a baseurl of the same. That custom mine will be added to user account if they are signed in the app.

Other Similar Project I've Been Working On:

Face Recognition based Attendance:

Idea:

This PWA / hybrid mobile app can register people with their photo and form people groups with those people in it. Then to take attendance of present people, just click a group photo and the image data is uploaded to a self hosted django server which hits AWS' face comparison N times where N is the number of people in that group with each person's id. The id is matched to the Picture storage to retrieve pic of that specific person and then comparison results for each person is sent back to app via server and automatic tick tocks happen.

Technologies used **:**

This app was developed using ionic framework, which helped me to make a PWA out of it as well as hybrid mobile app for pure native functionalities.

Github Link of the Project:
https://github.com/akshatbhargava123/Face_Recognition_Attendance

News4U:

Idea:

This is a very basic app I made during my free time as college project. This PWA / hybrid mobile app lists out News from newsapi.org. When clicked on one of the card, a detailed article on that news' gets open with buttons to add to read later, share and open official news page in inAppBrowser.

Github Link of the Project:
https://gitlab.com/akshatbhargava123/News4U

and many more...

Let's promote GSoC (Open Source Culture) and get some more students on-board:

Contributing to Open Source projects is the best way to put our developing skills into something that matters. We not only make apps / software but we help the whole world by making free, open and world class products. So, being an open source enthusiast it's my duty to promote this culture and as GSoC is giving me an opportunity to contribute to open source community at such a big scale then I would like to promote this program too.
So, here are my plans through which I will be sharing my GSoC experience.

- I will create a blog especially for GSoC that will be updated weekly. In that blog, I will share my last week experience, the challenges that I faced and how I overcome all those hurdles. This will be hosted on **Github Pages (**Open Source Hub**)** using a simple open source template. I will also encourage other fellow GSoCers(in my org and other fellow mates) to do so in the same way and to promote this culture.

Why Should I Be Chosen?

As I am well aware about this project and have researched on what we are going to build and am commenced to follow best coding / software development practices, I can work / take technical decisions myself without disturbing the community / mentors for the most basic stuff. I can be in daily touch with mentor if I get selected to give updates and discussing progress or implementing new ideas.
I've a proper practical timeline which can be easily fulfilled with 35 - 40 hrs of work per week and my whole concentration would be on GSoC only (Open Source :D).

If anytime mid term exams comes in the time period of GSoC then it would take a week max but I would compensate it by working more in the earlier and next week of the exam. I've good past experience doing internships for local companies in the same field of PWAs (so I know best practices to follow in general software development) and also,

I am a member of local groups like Google Developers Group New Delhi, Facebook Dev Circle New Delhi and local meetup groups (JSLovers, etc.) so I have a lot of technical back support if I get stuck somewhere or maybe to get idea for some new features / enhancements to be added. We also do have technical groups in our college so I can also motivate others to work on Intermine Platform. Last but not least, I want to experience how it feels to be a full time open source developer.