# 

# 

# 

# 

# WEforUS: A Community Platform

### Detailed Components Design

### Women Engineers January 2025

**Requirements Analysis doc:** [RAD (Requirement Analysis Document)](https://docs.google.com/document/d/1EOvy3KtNSXthx0VdpciAe1-puTyQEJVCzLI-dmggtAg/edit?tab=t.0)

# Components

## Back-end

### Posts

Currently, the plan is to integrate with a self-hosted, feature-rich backend. Using something open-source like [Lemmy](https://github.com/LemmyNet/lemmy) or [Discourse](https://github.com/discourse/discourse) would reduce the effort needed to build a useful system.  
  
Use cases covered

* Reaching out - this would be the backbone for post-based communication, organised by topic. For both formal and informal communication
* Meet-Up - Can associate an event with a post - enabling rich discussions by default
* Job Board - Implement front-end restrictions to ensure necessary details are provided. Also enables discussion associated directly to a job. We would have to implement a way for the creator to ‘close’ discussion once filled or expired.
* PG Finder: Similar to job board; ‘PG’ posts could be associated with discussions and ratings
* Private Messaging: If available on the platform, it can be used out of the box. If not, there are definitely libraries we can use.

Plan

* Create private topics for ‘Job Board’, ‘PG Finder’ and ‘Meet up’. Filter them out from regular conversation and restrict availability only to the associated pages
* Create unique interfaces for each feature
* Build the front-end for these features from scratch, interacting with the API as needed

Alternatives

|  |  | Lemmy | Discourse |
| --- | --- | --- | --- |
| Ease of development | Forum help | Not as much, but there are more code examples | Extensive, but a lot of the relevant ones seem older (2016 and so) |
| Richness of API | [Extensive](https://join-lemmy.org/api/classes/LemmyHttp.html#getPost) but not as [documented](https://meta.discourse.org/t/why-isnt-the-discourse-api-fully-documented/138265/4) | [Extensive](https://join-lemmy.org/api/classes/LemmyHttp.html#getPost) but could be better [documented](https://lemmy.ml/post/98675). |
| Independent front-end development | [Definitely possible](https://lemmy.ml/post/363093) and brought up often | Not really [recommended](https://meta.discourse.org/t/next-js-application-as-custom-front-end-replacement/196796), but seems possible (see also) |
| Amount of customisation | Some CSS customisation with themes | Customisable (front and back) with [themes and plugins](https://meta.discourse.org/t/how-do-i-go-about-making-a-very-customized-theme/284772/2) |
| Security | Overall | Uses ActivityPub which seems secure | Details [provided](https://github.com/discourse/discourse/blob/main/docs/SECURITY.md) |
| Transparency (to developers) | Simpler codebase | More complex codebase |
| Isolation from other instances (unlike almaconnect) | [Yes](https://lemmy.ml/post/146965#:~:text=Closed%20/%20Private%20instances) | Yes |
| Ease of maintenance | Easy to update base code |  |  |
| Language documentation | Rust + TS; | Ruby + JS; |
| Learning curve | Higher (rust) | [Lower](https://appdeskservices.com/ruby-vs-rust/#:~:text=2.-,Syntax%20and%20Learning%20Curve,-Syntax%20and%20Learning) (ruby) |
| Features available | Topics | ✅(community; like subreddit, or tags) | ✅ |
| Filters | [✅](https://lemmy.ml/post/2922402)(less extensive) | [✅](https://meta.discourse.org/t/searching-for-content-effectively/273328) (more extensive) |
| Discussions (for a post) | ✅ | ✅ |
| Private messaging | [✅](https://join-lemmy.org/docs/users/01-getting-started.html?highlight=bot#:~:text=Send%20notifications%20to%20Email:%20Enable%20to%20receive,read%20the%20community%20rules%20in%20the%20sidebar) | ✅ |

TLDR: Discourse is better used and seems better documented. But Lemmy is built in a modular way that allows us to build an independent front-end. Doing so with Discourse could cause breaks with future updates. The Lemmy codebase is also smaller and less complex.

### General

This includes aspects - not covered by Posts - that we’d have to implement.

Use cases covered

* Hall of Fame: Storing and creating information of members
* Who’s Who: Storing images
* Individual Pages + Profile Information: Storing information per person; provided by them or core information like e-mail
* Announcements: Envisioned as posts that can’t be commented on. These could be something shared by TS (e.g, application dates for upcoming cohorts, forms to be filled, general information seen in mails) or us (Sharing linkedIn & medium posts, project completion, appreciating others within the cohort publicly)

Plan

* Should be covered in a set of tables; possibly NoSQL for images, SQL for the rest
* Should be sufficiently secure
* Should be backed-up regularly

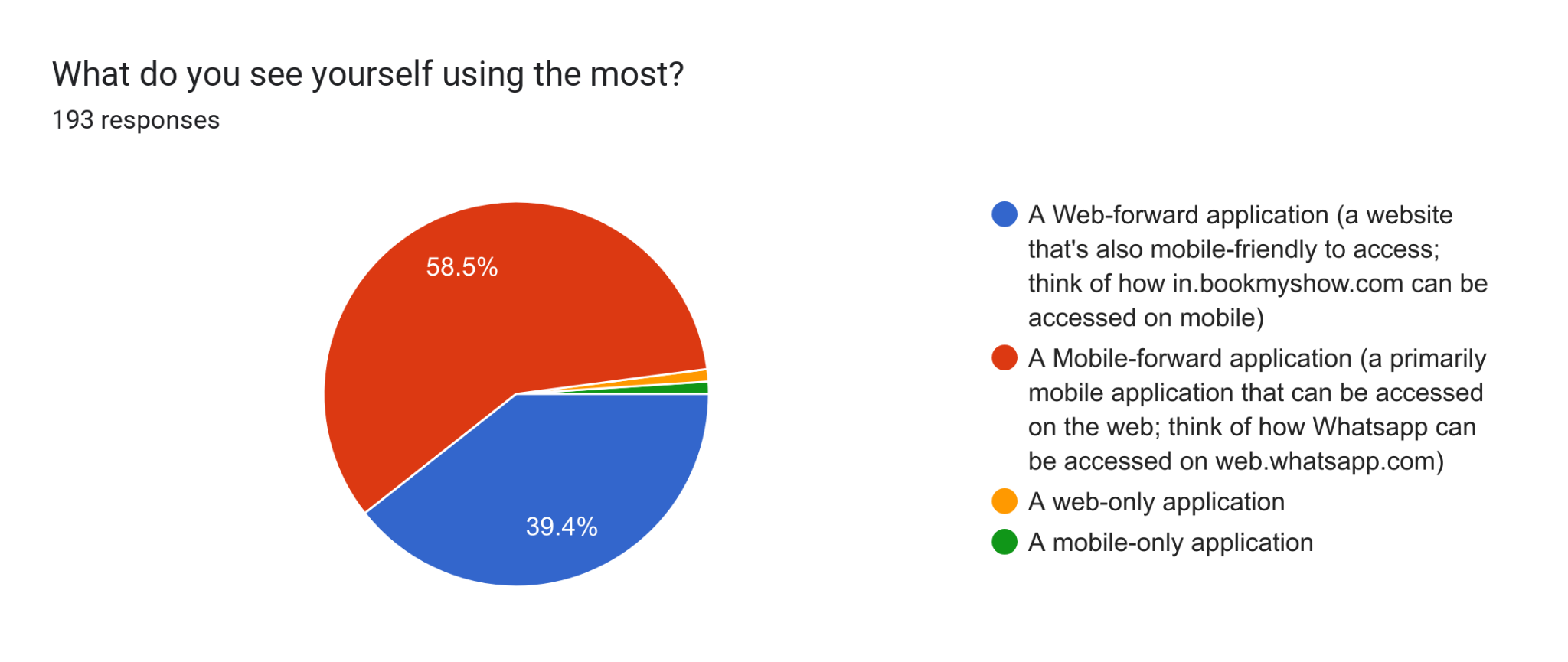
Alternatives

**Choose based on the front-end & back-end frameworks**

Lemmy: PostgreSQL and [pict-rs](https://crates.io/crates/pict-rs) (image hosting; seems to be a wrapper on [ImageMagick](https://imagemagick.org/index.php))

## Front-end

Custom front end to be developed to meet all use cases.



Use cases covered

* All of the core features
* Notifications
* Log-in and Auth - Asokan prefers we use Google SSO **only**, since security is handled much better than storing passwords on our server. We’re going with this approach.

Requirements

* Deployable on all selected platforms with minimal additional effort
* Maintainability of the codebase
* Sufficiently easy to set up templates (based on WE color scheme)
* Will be building a front-end to interact with the post-based back-end. Should be checked if this is a good match

Plan

* Create tuned pages to handle features based on post (Job Board, Meet Up)
* Integrate with LinkedIn to autofill details, Google authentication (checked against emails in our WE database), Google calendar (or similar) to set up and display events, and Maps to show PG locations and compare

Alternatives

|  |  | [Phoenix](https://www.phoenixframework.org/) | [Flutter](https://flutter.dev/multi-platform/web?gad_source=1&gclid=Cj0KCQiAqL28BhCrARIsACYJvkd47_2SuoNcfKauJx6FtZDEfpKR6l6A6ke54HEEiywLoTx6T7HrpL8aApEiEALw_wcB&gclsrc=aw.ds) | [Ionic](https://ionicframework.com/) | [ERP Next](https://docs.frappe.io/erpnext/user/manual/en/introduction) | [React native](https://reactnative.dev/) |
| --- | --- | --- | --- | --- | --- | --- |
| Ease of development | Documentation |  | Good |  | Little to no (Bad) | Good. |
| Integration ease |  | Comprehensive tutorial in documentation. Minimal steps.  [Ref](https://docs.flutter.dev/deployment/flavors) |  | - | Extensive API Integration Tutorials available. |
| Effort for deployment on noted platforms? | Web only | Supports Cross Platform  [Ref](https://docs.flutter.dev/deployment/flavors) | ‘Write once, run anywhere’, on all platforms | Supports Cross Platform  [Ref](https://discuss.frappe.io/t/frappe-erpnext-quick-easy-mobile-app-tested-working/128762) | Supports Cross Platform - [Web/ios/Android](https://docs.expo.dev/) |
| Security | Overall |  | Have to be mindful of certain practices  [Ref](https://www.cossacklabs.com/blog/flutter-application-security-considerations/) |  | Highly Secure (widely used across various businesses)  [Ref](https://frappe.io/security) | No explicit security issues noted, possess the same risk as [any other tech stack.](https://blog.logrocket.com/understanding-security-react-native-applications/) |
| Transparency (to developers) |  | High |  | High | High |
| Ease of maintenance | Documentation |  | Has an active community and is constantly updated. |  | Constantly being updated and has an active community. Parent framework team is very receptive. | Has an active community and is constantly updated. |
| Learning curve | Steep (difficult to learn initially but progresses rapidly with effort) | Exponential (Documentation walks you through your first app. Simple learning) | Easy (for those who’ve worked with web) | Steep (difficult to learn initially but progresses rapidly with effort) | Easy (For Devs who worked with web already) |
| Resource availability (eg stackoverflow) for debugging | Low to no.  Unable to find answers for queries. | High. Lots of code repos. I found some stackoverflow questions with answers. Various youtube tutorials  Uses widgets which can be less intuitive | High; uses common web frameworks to develop | Medium.  Lots of code repos. I found some stackoverflow questions with answers. | High. Medium articles are a good source to help with template codes and stackoverflow/github do have good solutions to blockers. |
| Features available | Scalability | Yes | Yes | Yes | Yes | Yes |
| Customizability |  | High |  | Med-Low | High (Wide range of CSSLibraries available) |
| Deployment (IoS, Android, Web) | Web only | All  IOS, Android, Linus, Mac, Windows | All; but looks like it runs as a [PWA shell rather than native code](https://nisarg-c-metric.medium.com/react-native-vs-ionic-the-best-mobile-app-framework-in-2024-b15e812b29ad) | All | All |

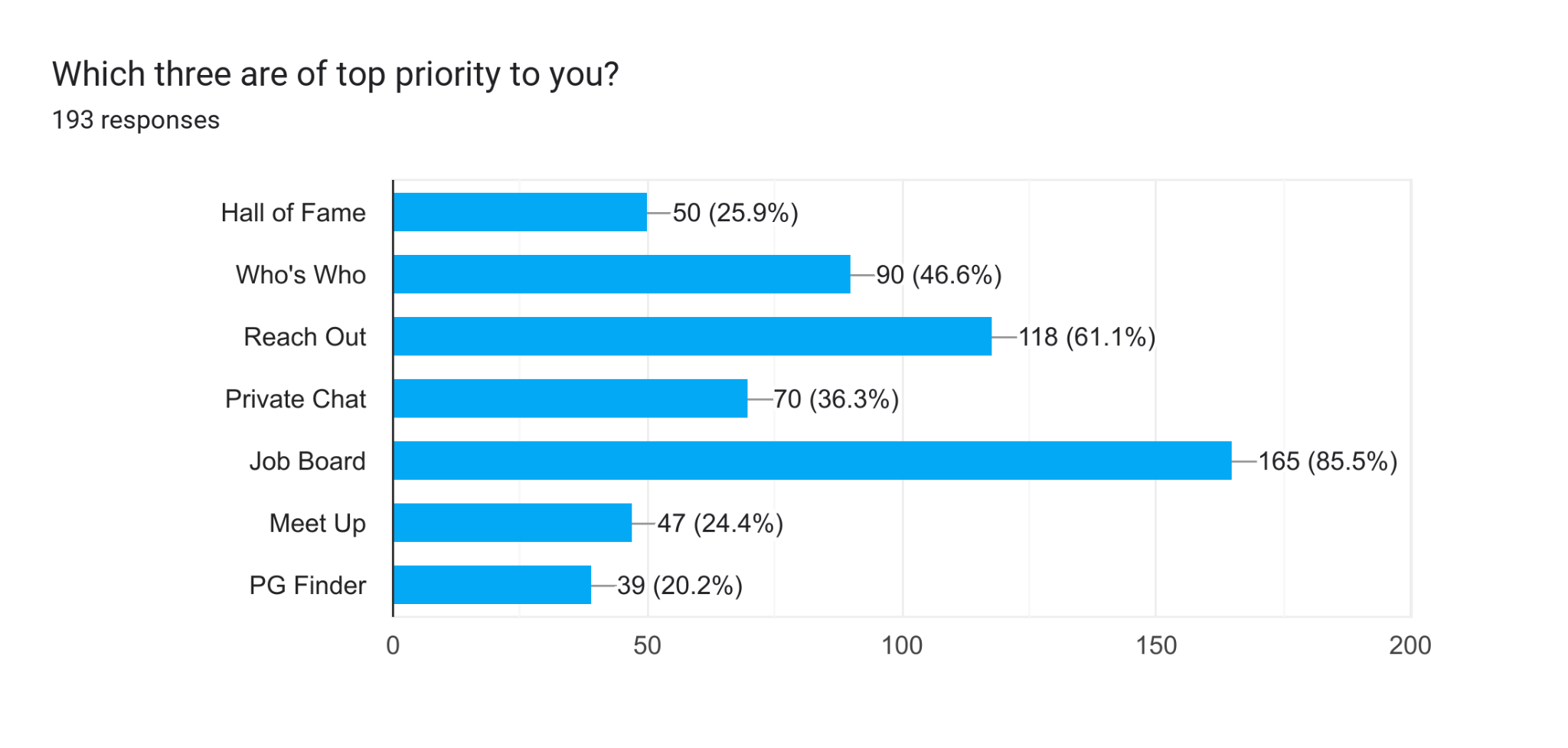
“If native performance and extensive community support are crucial to your business needs, React Native is the go-to framework. On the other hand, if you prioritize cost-effectiveness and have a team skilled in web technologies, Ionic offers an efficient solution.” - [medium](https://nisarg-c-metric.medium.com/react-native-vs-ionic-the-best-mobile-app-framework-in-2024-b15e812b29ad)

[Comparison](https://medium.com/@fmmagalhaes/i-built-the-same-app-with-flutter-react-native-and-ionic-33ff8b358562) working with Flutter, Ionic, React Native

**React JS is chosen because**

* **We believe convenience afforded by mobile is important in enabling impromptu and informal conversations (ruling out web-only platforms)**
* **React is fairly similar to React Native, which will make it much easier to extend deployment onto mobile**
* **React has more documentation and discourse, also it’s easier for contributors to pick up work and maintain long term.**

## Prioritization of features:



Hall of Fame is a minimal priority feature since it raises many questions. (Who decides who is in the Hall of Fame? How is it updated?..) with no easy answer. It can be covered by letting people create announcements about their achievements & show the top ‘liked’ achievements separately.

Also, based on suggestions below, Private Chat is a tertiary priority as well for now.

Additional elements required:

* Auth
* Notification
* Settings
* Landing page - essentially the Reach Out feed, but with only, say, 5 posts displayed. Everything else viewable only once logged in

## Schedule

| Jan | 25 | * Tech stack selected * Prioritisation of features |
| --- | --- | --- |
| 28 | Proofs of concept: Google Auth & Post API (set up private instance and fetch data) |
| 31 | Work begun on top 3 features |
| Feb  ([Saathwika Dhenuvakonda](mailto:dsaathwika15@gmail.com)unavailable from 7th to 14th)  ([Anisha Sharma](mailto:anisharma968@gmail.com)available on weekends) | 7 (Sprint 1) | Completed functionality of features |
| 14 | Work begun on the rest of the features |
| 21 | Completed functionality of features |
| 28 | Make the UI consistent & fine-tune |
| March  ([Tara Ann Lukose URK21CS1022](mailto:taraann@karunya.edu.in)may not be available 3-5 March)  ([Anisha Sharma](mailto:anisharma968@gmail.com)available on weekends) | 7 | Bug fixes |
| 12 | Rigorous Testing + Test writing |
| 14 | Set up on prod server with regular backups of data |
| 21 | Buffer time |

## Stretch Goals:

Some suggestions from the [form](https://docs.google.com/forms/d/19zM-VvBbVbPUgTqrapDRrJrwXjsKbQgWIPjb_GZ7H0o/edit#responses):

1. Hall of fame: have like an insta feed of the most recent *n* posts with captions (old ones archived to who’s who profile)
2. Option of anonymous posts for Reach out ++
3. Referral request form with Job Board ++++
4. Online meet up option
5. Roommate/flatmate finder (can be a community in reaching out), also DSA corner, higher studies & abroad studies
6. Include current TS emails in announcements
7. Discord-style study servers
8. Mentor-mentee or buddy system +++
9. Company & workplace review
10. Chat room with expiration
11. Mentor-offered learning sessions: eg bootcamp, extra classes
12. Mock interview platform; resume review
13. Polls on posts

Concerns:

1. Who chooses people to be in the Hall of Fame?
2. Hall of fame not as important ++++ (one thinks that posting achievements in WE4US would be too similar to linkedin) suggestion: pull recent posts from linkedin
3. Private messages may not be needed: can provide whatsapp or linkedin ++++++
4. Calendar for meet up not needed +++
5. PG finder not as important +++
6. Privacy & member verification. Hide profiles from external users

Suggested goals:

1. Set up production and test instance for future maintenance