Computer Science Project – SummitSync

Conference Volunteering Management System

Team:
Krishnan Winter
Team lead, developer

Maninder Gill
Scrum master, developer

Jay Choi
Developer

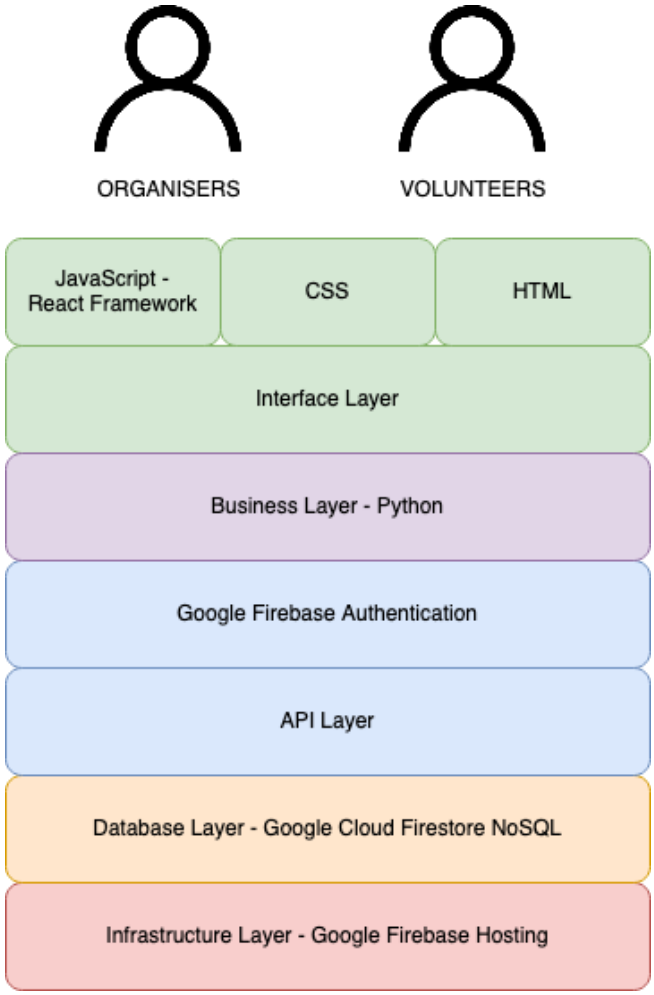
Joshua Wong
Developer

Jon Lai
Developer

Table of Contents

Table of Contents **2**
Overview **3**
Infrastructure Layer 3
Database Layer 3
API Layer 3
Business Layer 4
Interface Layer 4
Functionalities **4**
For Conference Organizers: 4
For Volunteers: 8
For Volunteer Managers (Volunteer Team Leaders): 14
Extra Features 15
All third-party functionalities **16**
Google Location API **16**
Firebase Authentication 17
Firestore Database 17
Implementation challenges **18**
Installation/User document/manual **19**
Installation and running: 19
Backend 19
Frontend 19
User manual: 19
• Account creation 19
• Editing your profile 20
• Dashboard & joining/creating conferences 20
• Tasks & volunteers 20
• Forum 20
• Schedule 21
• Attendance & feedback 21
• Account deletion & resetting your password 21
References **22**

Overview



Infrastructure Layer

We have used Google's Firebase app development platform. Primarily, this is done for the integration between the authentication, database and infrastructure.

Database Layer

For our database layer, we have used Google's Firestore. This is a NoSQL document database that provides performance and scalability for applications.

API Layer

We are leveraging several APIs within our application, one being Google Firebase Authentication API. Firebase authentication supports email and password accounts as well as Google, Facebook, Twitter etc.

We have also used the Google Maps API to implement the functionalities that the users can search nearby conferences based on the location they are at. We use the Google Places API and DistanceMatrix API to calculate distance between two locations.

Business Layer

To implement our backend business layer, we have used Python. Additionally, we use Flask, which is a lightweight framework that makes setting up a web application simple, without the need for bloated libraries and dependencies. Most of our functionalities are delivered through Flask, with data being sent and retrieved from firebase.

Interface Layer

In our interface layer, the React framework enables us to build a dynamic and interactive user interface through its powerful component-based architecture, greatly improving the user experience and simplifying the development process. Alongside React, we utilise CSS for effective styling and layout control and HTML for structuring the web content.

Functionalities

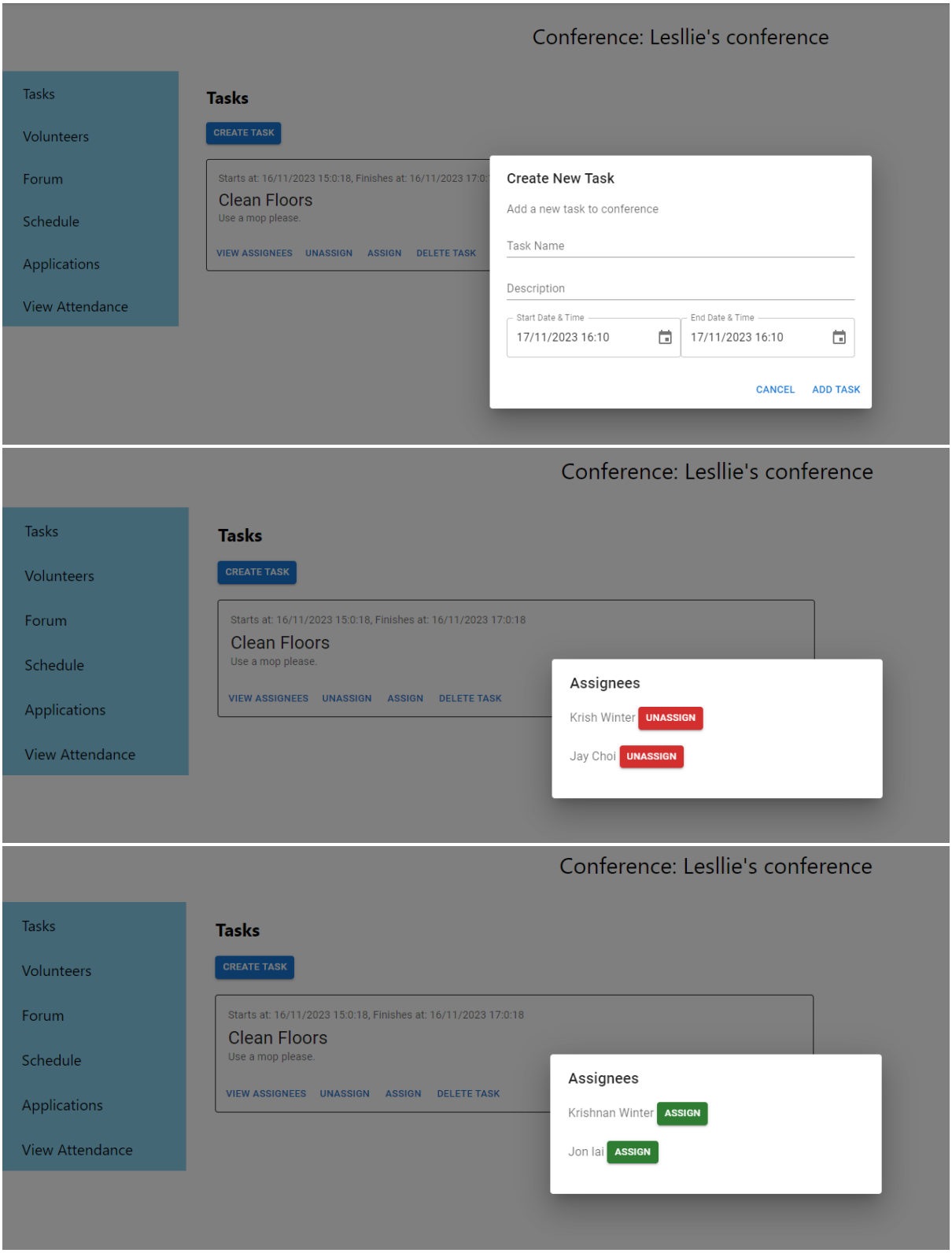
For Conference Organizers:

• *User Management: Organisers can create, edit, and manage volunteer profiles, including contact information, skills, and availability.*

- Unfortunately, we were unable to adequately address this requirement. While organisers cannot edit a volunteer's profile, the volunteers themselves can edit their skills and availability alongside other key information from the 'my profile' page.

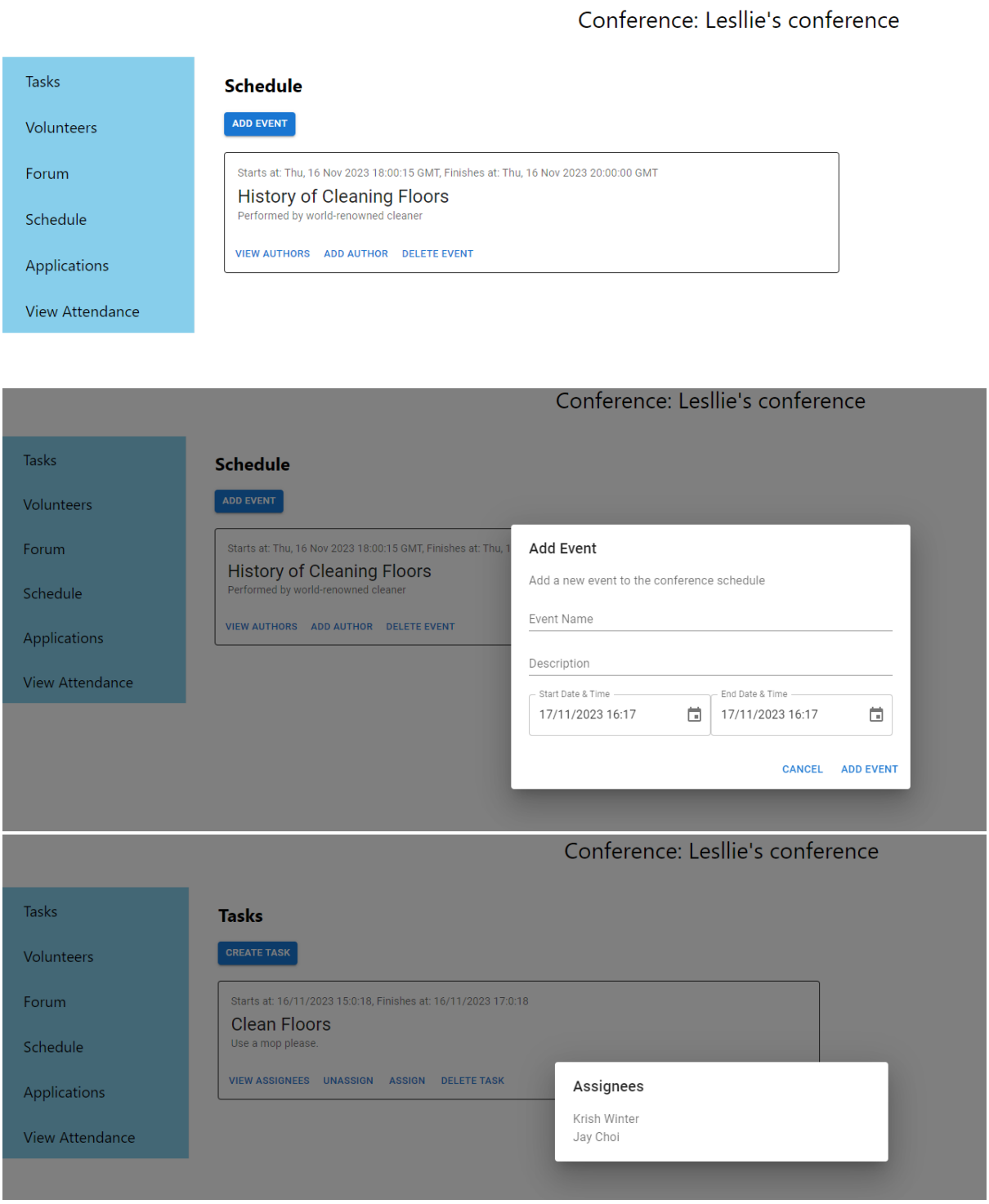
• *Task Assignment: Organisers can create a list of tasks for the conference and assign them to specific volunteers or groups of volunteers.*

- The same organiser account can create tasks in the 'Tasks' page within a conference and assign them to volunteers in the same conference through the 'assign' and 'unassign' buttons on each task.



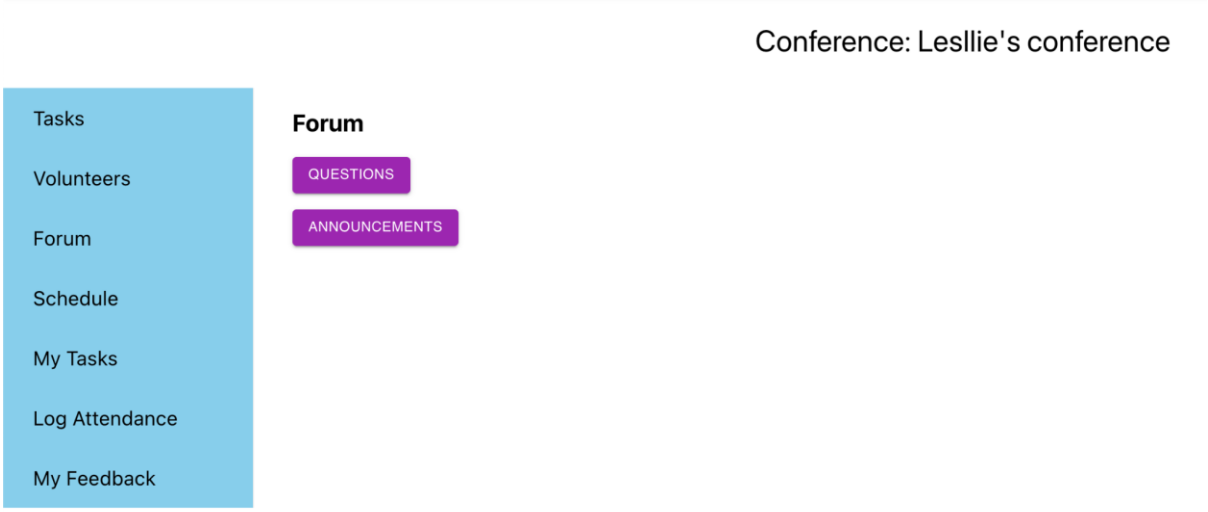
• *Schedule Management: Organisers can create and share the conference schedule, including volunteer shifts and task assignments.*

- Organiser accounts once created along with a conference can create schedules which involve the creation of events, assignees and tasks that all other account types can see within the conference information addressing this requirement

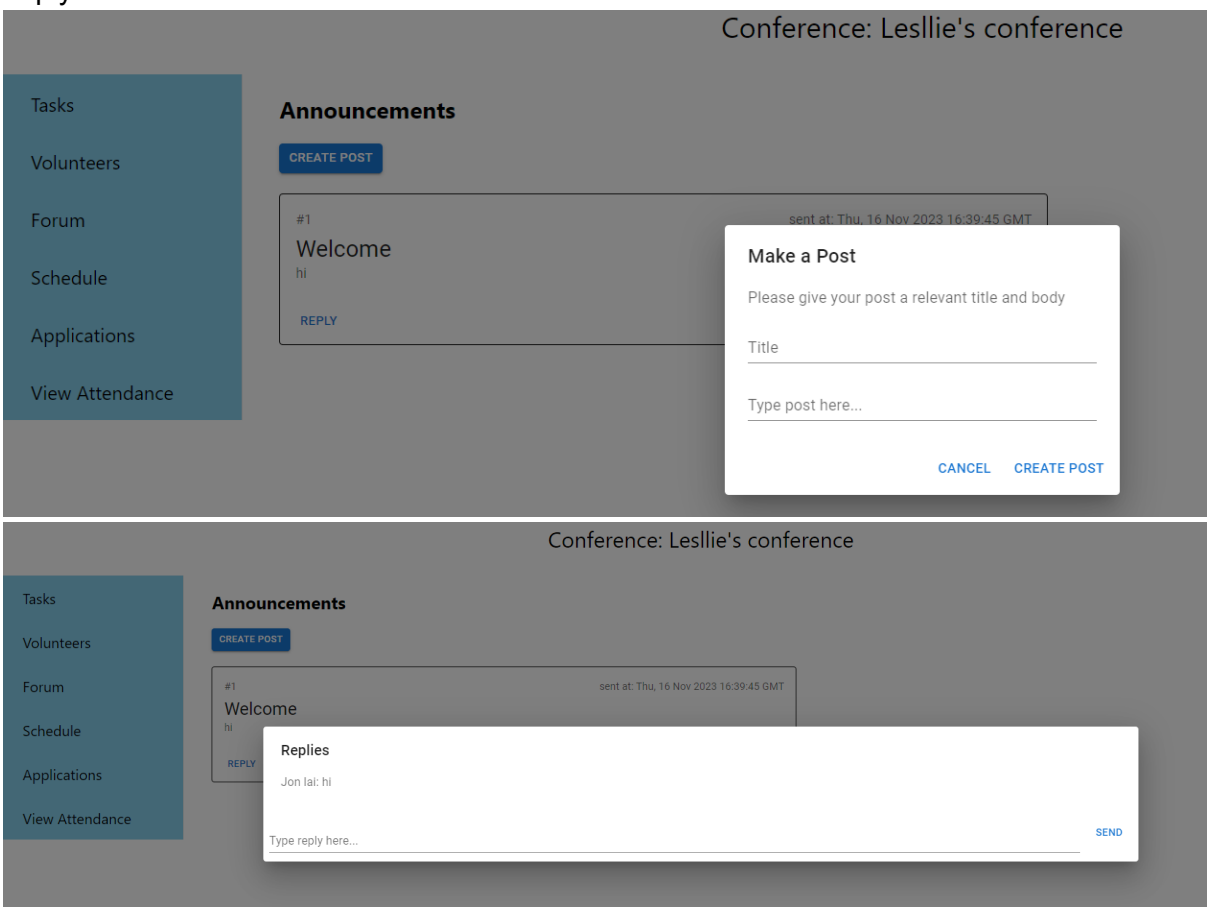


• *Communication: A built-in messaging system allows organisers to send messages and updates to volunteers and receive responses.*

- Organisers can send messages and updates through the forum tab on a conference page.

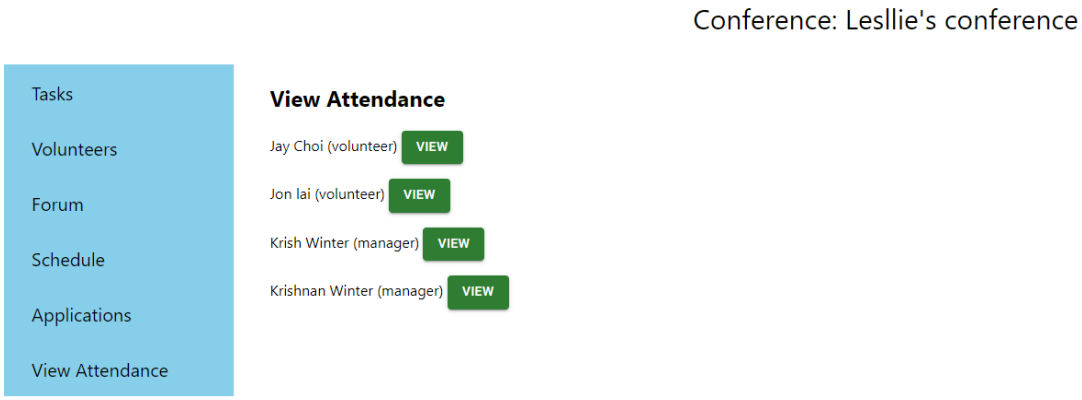


- Organisers have the ability to post announcements in the appropriately titled ‘announcements’ forum through the ‘create post’ button. Volunteers and volunteer managers are unable to make any posts here, but can send a response through the ‘reply’ feature.



• Attendance Tracking: Organisers can track volunteer attendance and hours worked for record-keeping and acknowledgment purposes.

- There is a ‘view attendance’ tab on the conference page, from which the organiser is able to access attendance info for all the volunteers and managers in the conference.



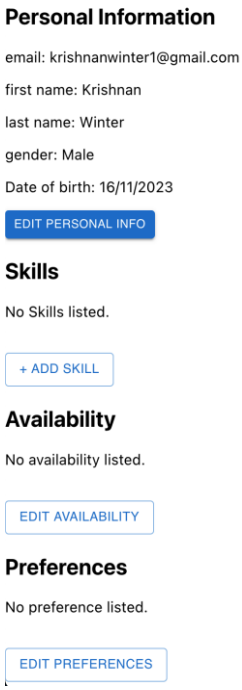
- Organisers can see each conference members’ approved and pending attendance requests. They can click either ‘approve’ or ‘decline’ on pending requests.

For Volunteers:

• *Profile Creation: Volunteers can create and update their profiles, providing information about their skills, availability, and preferences.*

- As a volunteer, you can create an account, including information such as name, date of birth and location. This is the initial information required to create your account

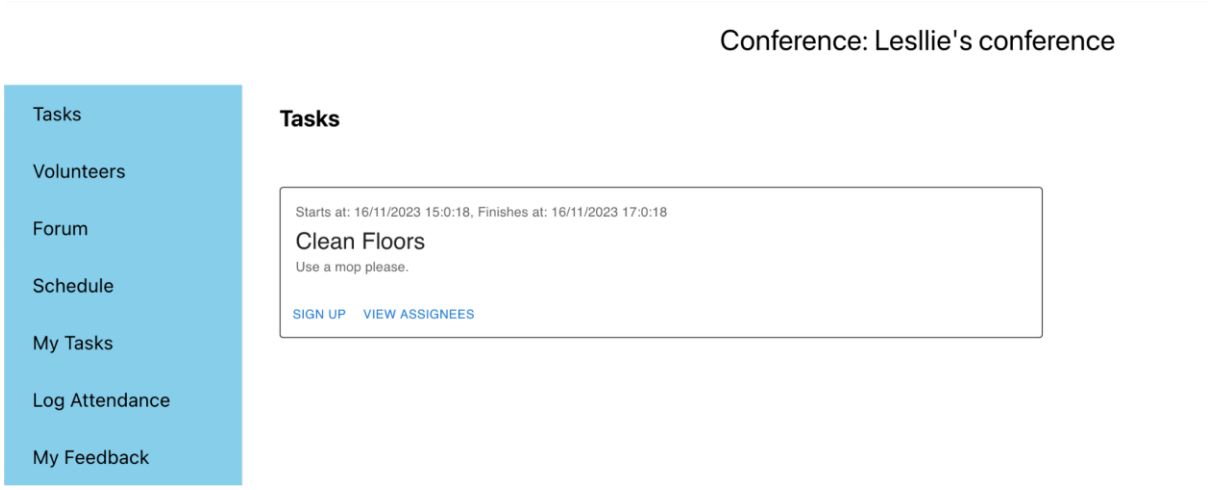
- After creating your account, you can edit it from the ‘My Profile’ page, which can be accessed from the header bar. You can edit personal information that you supplied when you created your account, and you can also add skills, availability as well as preferences for tasks that you would like to volunteer for. The organisers can see this information, and decide what tasks to assign to different volunteers, or even if they should be accepted as a volunteer based on skill requirements.



Task Selection: Volunteers can browse through available tasks, view task details, and sign up for tasks that match their interests and skills.

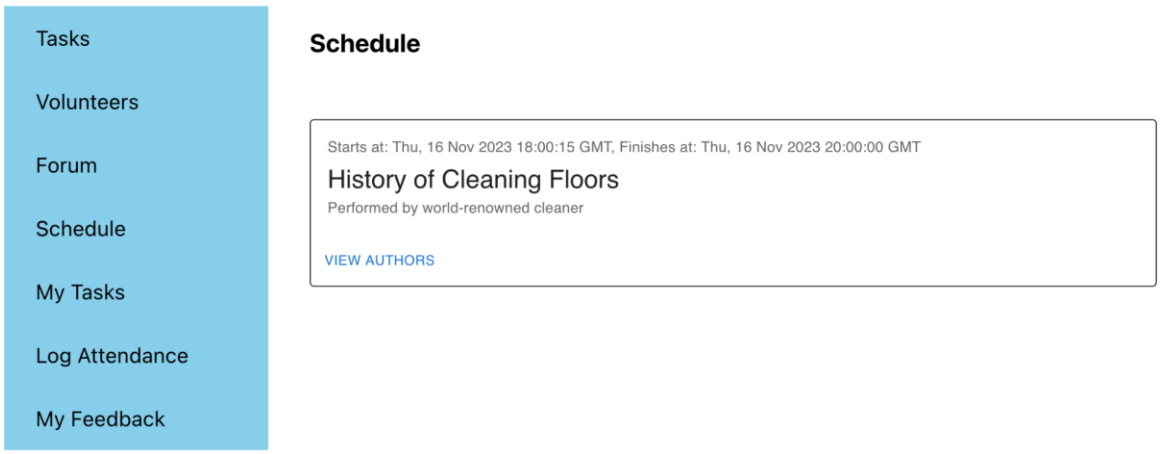
- When viewing a conference that a volunteer has joined, they can navigate to the tasks tab from the menu bar on the left. From here, they can view available tasks,

see their title and description as well as start and end times. They can also see which other volunteers have signed up, as well as sign up for themselves. When you sign up, you are added to the assignees list.

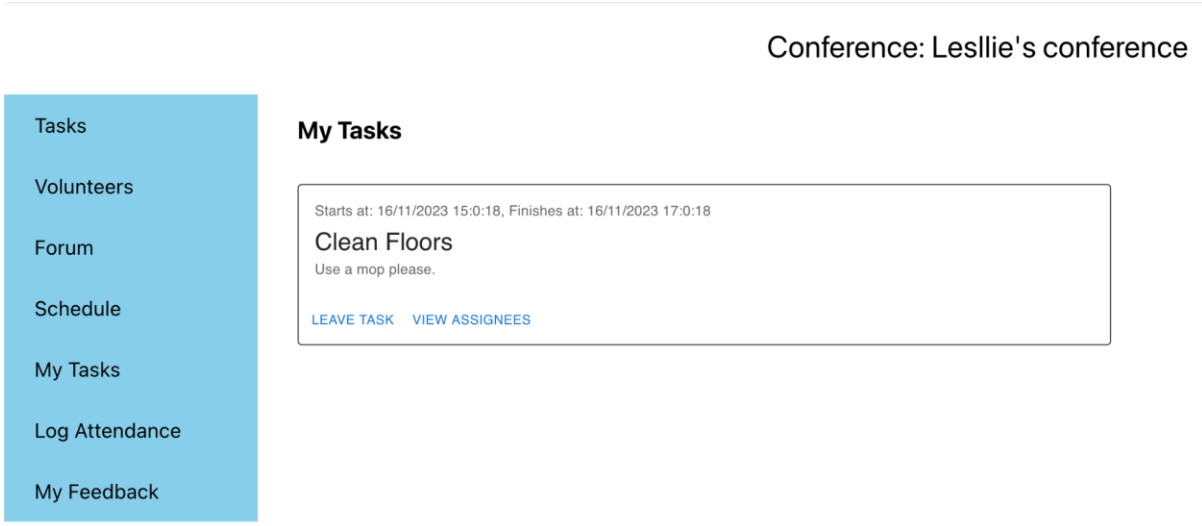


• *Schedule Access: Volunteers can access their assigned tasks, shifts, and the overall conference schedule through their accounts.*

- To see the schedule of the entire conference, a volunteer can navigate to the schedule tab on the menu bar. This will bring up sessions that have been created by the organiser. They can see start and end times, as well as the title and description of the talk. They can also see the authors that are attributed to this certain event.

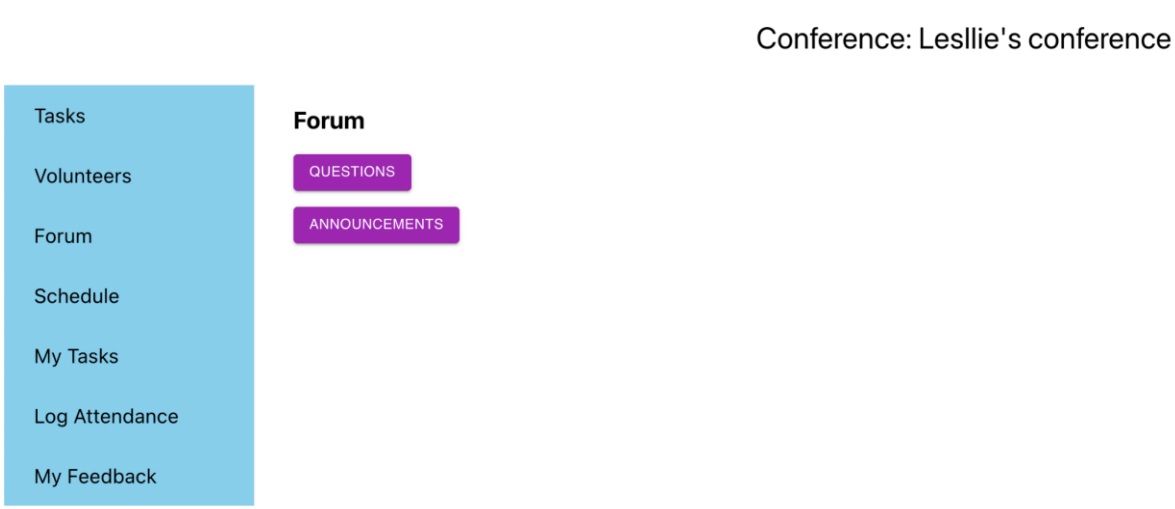


- Volunteers can also see their tasks from the ‘My Tasks’ tab on the menu bar. This will show all the tasks that that particular volunteer has signed up for or been assigned to by the organiser. From here, they can also see other volunteers on that particular assignment, as well as leave the task if they no longer wish to participate in it.

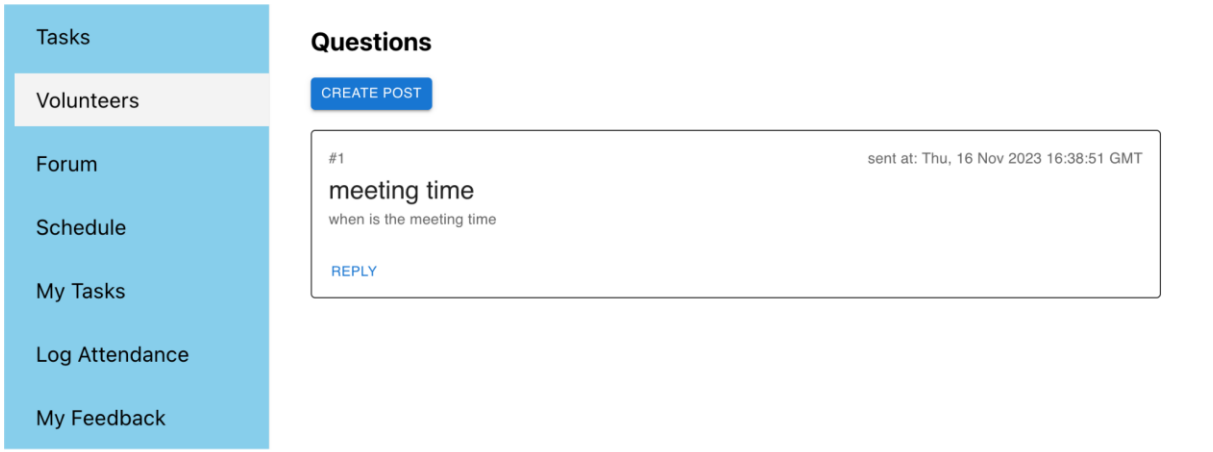


• *Communication: Volunteers can receive messages and updates from organisers and respond if necessary.*

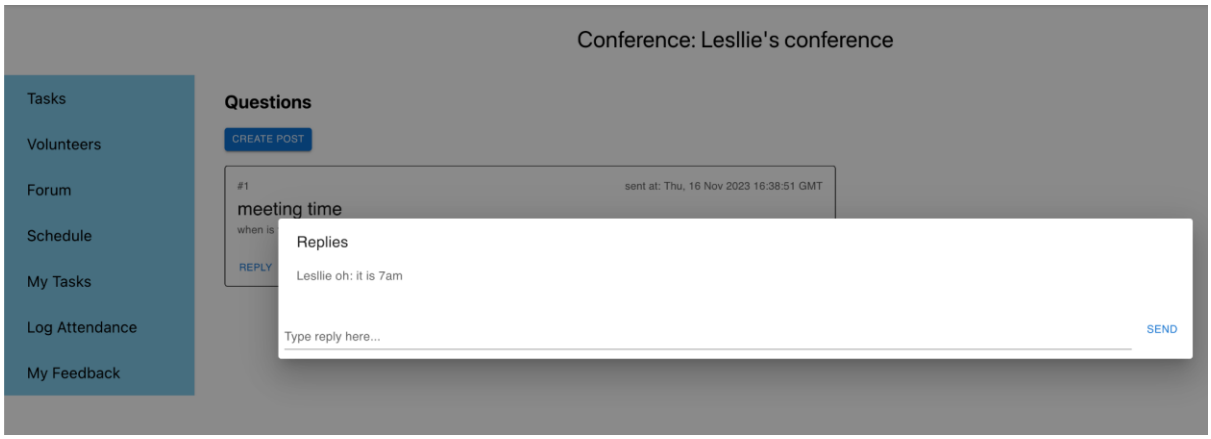
- To facilitate communication, we have implemented a forum. This can be found from the menu bar under the ‘Forum’ tab. Here they can select two options, questions, which allows volunteers to post questions, and announcements.



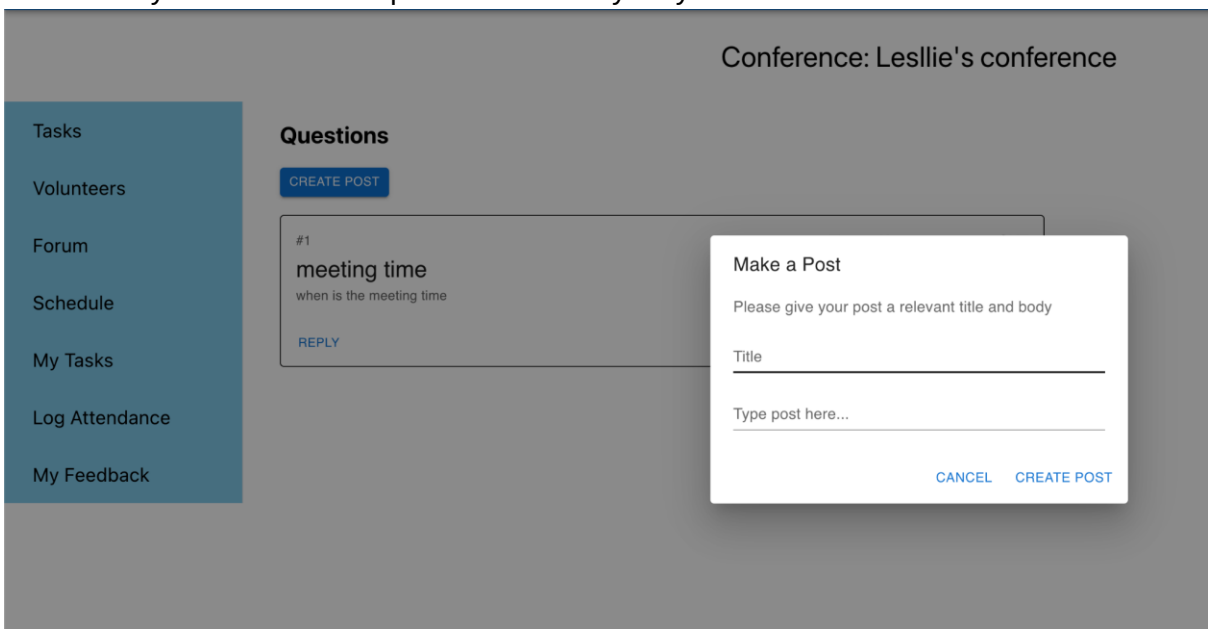
- On the ‘QUESTIONS’ option, volunteers can see all other questions that have been posted by other users.



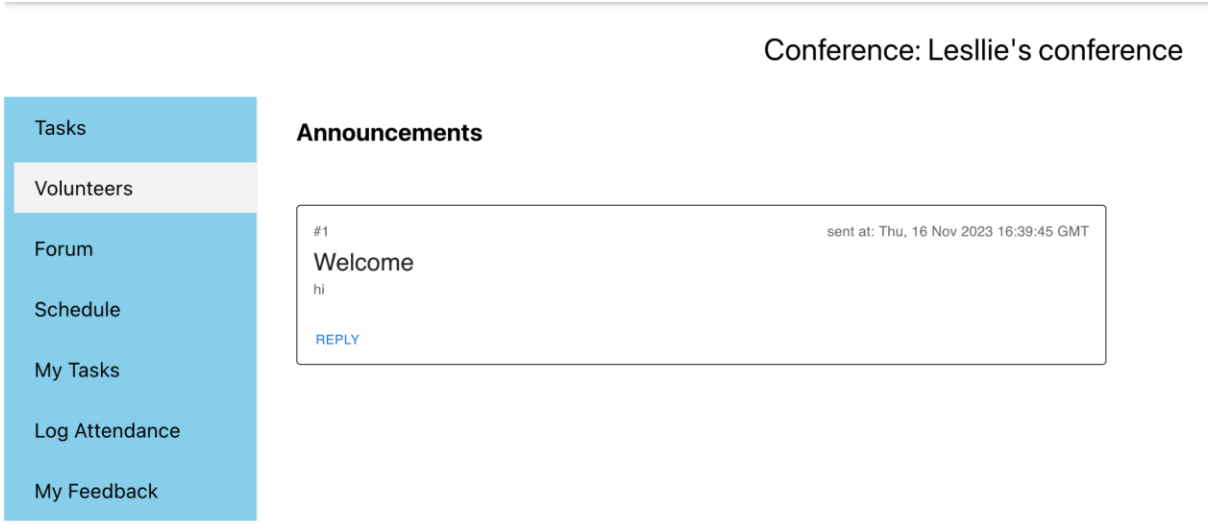
- They can then reply to these questions in the reply thread.



- They can also create questions that they may want to ask as well.



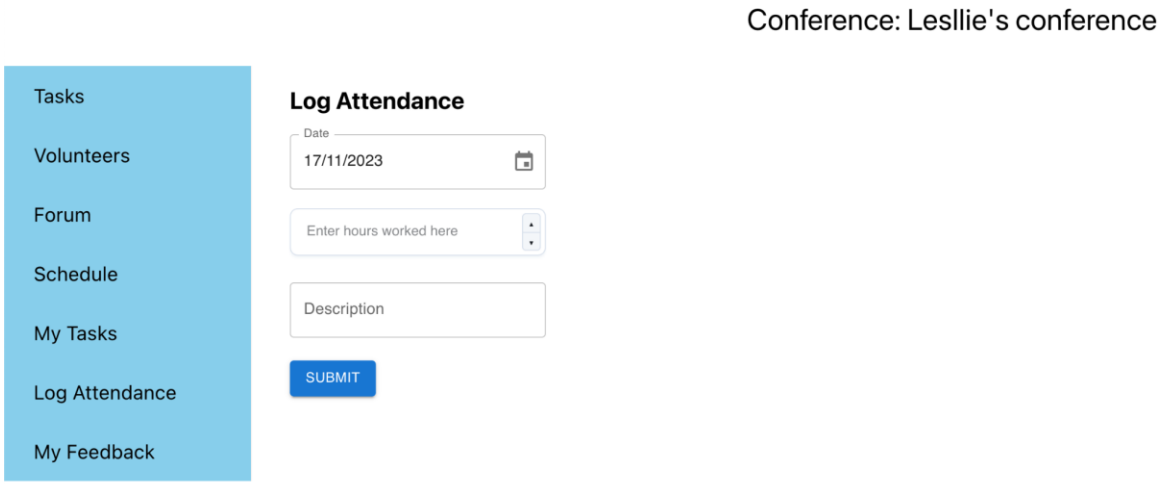
- They can also view announcements that have been made by organisers (but volunteer's cannot make such announcements).



- But they can reply in the same thread-like manner as questions.

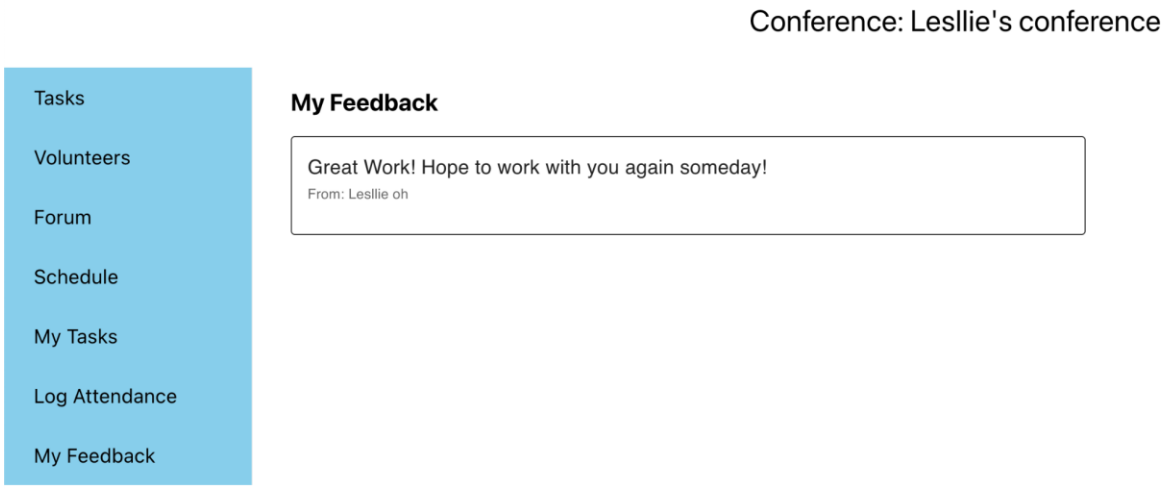
• *Attendance Logging: Volunteers can log their attendance and hours worked, which organisers can review and approve.*

- On a conference page, in the attendance tab, volunteer accounts can log attendance and hours worked which then can be seen by organisers. An attendance log includes a date, hours as well as a description of the work that has been conducted during that given session.



• *Viewing feedback: Volunteers can also view feedback provided to them. (Not in specification)*

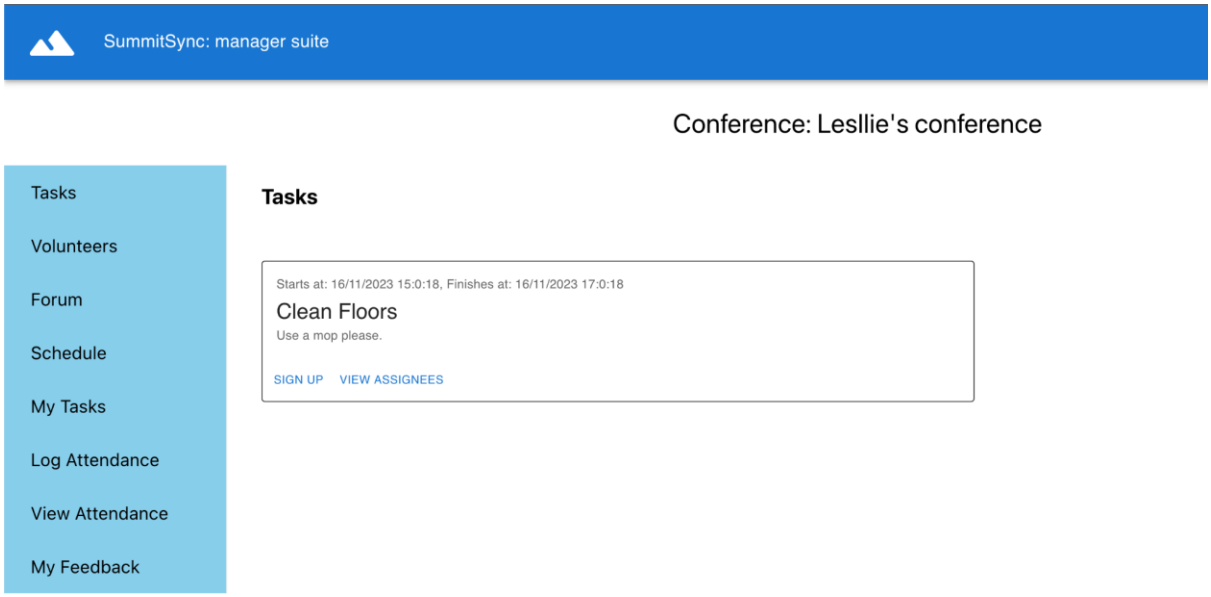
- Volunteers can also view the feedback that is given to them by organisers as well as volunteer managers. This feedback is localised by conference, and is accessible via the ‘My Feedback’ tab on the menu bar.



For Volunteer Managers (Volunteer Team Leaders):

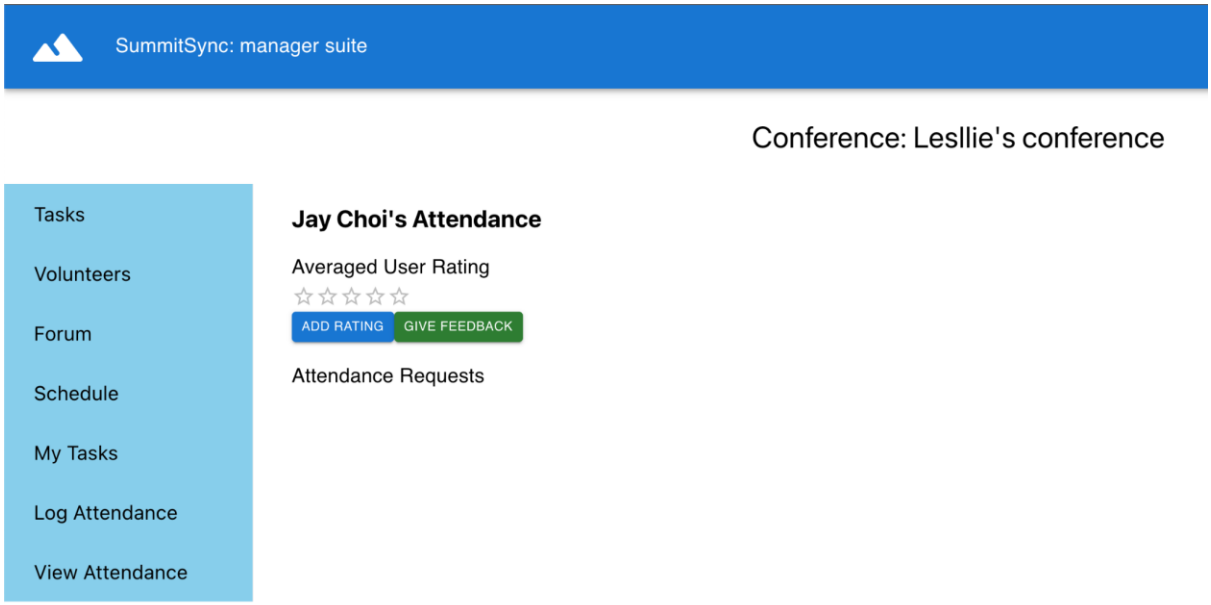
• *Task Overview: Managers can view an overview of all tasks, their status, and the volunteers assigned to them.*

- Managers can view the tasks tab on a conference page seeing all tasks, statuses and volunteers assigned addressing this objective



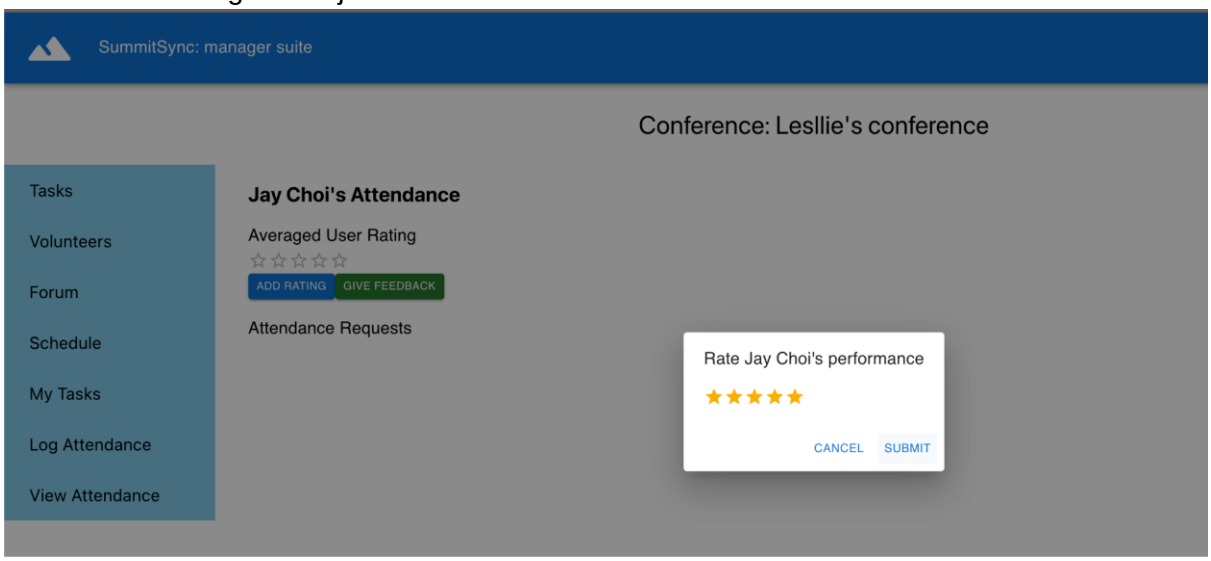
• *Volunteer Performance: Managers can log and track volunteer performance and fill in feedback from attendees or other volunteers.*

- Managers can view the attendance tab on a conference page and see volunteer performance and give feedback to volunteers addressing this objective



• *Select Best Volunteers: Based on the volunteer performance, Managers can vote for the potential best volunteers.*

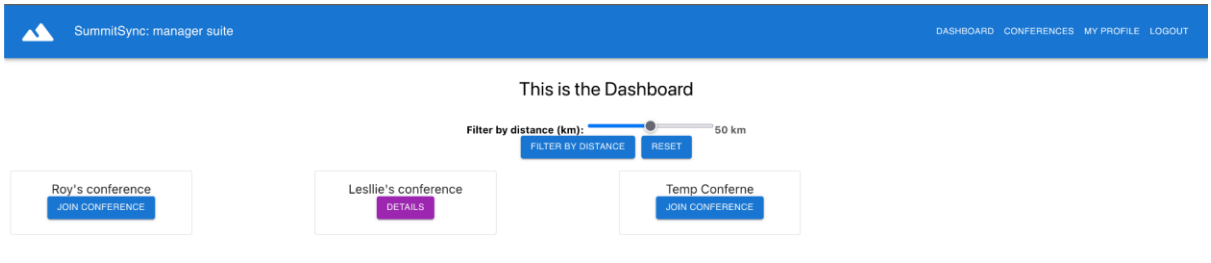
- Managers can rate volunteers out of 5 which gets added to the volunteers profile addressing this objective



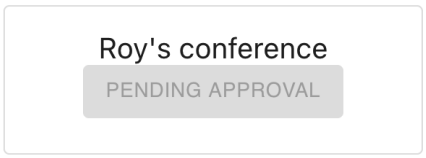
Extra Features

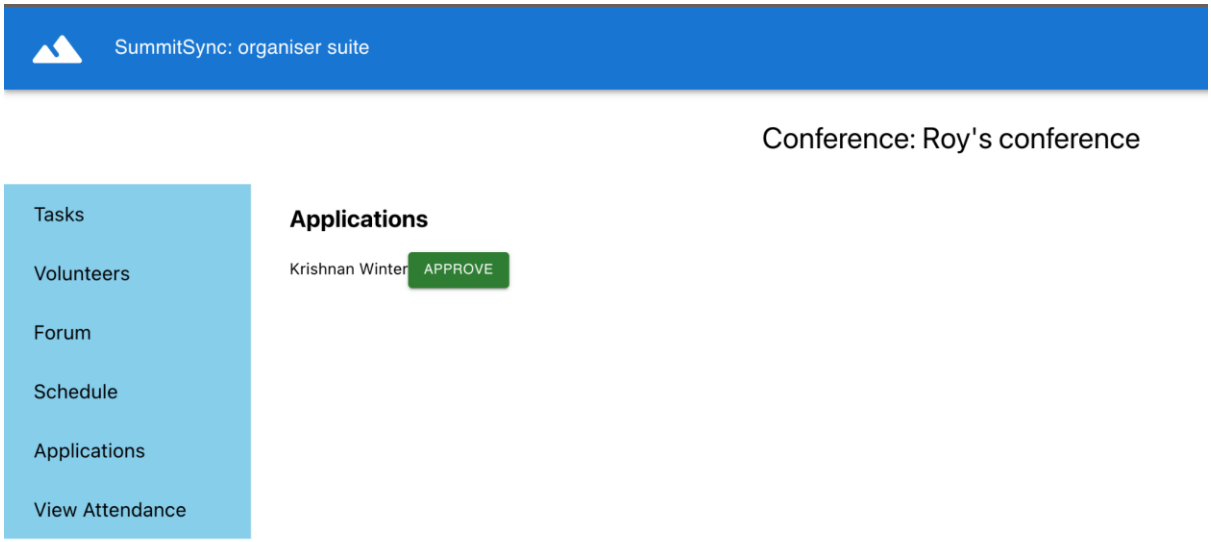
• *Nearby Conferences*

- Users can browse nearby conferences, and specify a radius in which they want to search. This is based on the location they entered when signing up, and the location that the conference organiser imputed when they created the conference. To use this feature, move the slider, and click the filter by distance. It defaults to a 50km radius.



• *Requesting to join*
  - Users can also request to join a conference. From the dashboard, if they see a conference that they wish to volunteer for, they can click on the ‘Join Conference’ button. This will add a request to join that the organiser can see, and approve or ignore.





All third-party functionalities

Google Location API

One of the API’s that we leveraged was the Google location API (Google, “Overview | Geolocation API”). We use this to implement the functionalities that the users can search nearby conferences based on the location they are at. We first use the Google Places API to convert an address into a Place ID. This is done when creating a user or a conference. We then use the DistanceMatrix API to calculate distance between two locations.

We particularly chose Google's Location API for this purpose as it is well documented, with lots of resources available online for guidance. Additionally, as we are using alot of other Google products, we already had a Google development account, and did not have to pay/start another membership to another service. Google also provides a Python library for the Location API, making it even easier for us to integrate into our backend.

However, we are using the free trial of Google cloud, meaning that we only have a total of 3 months free, but are able to utilise all the features on offer. In the future we would potentially have to pay for every request under a pay-as-you-go pricing model, which would increase the cost of the project. More pricing details can be found at: https://developers.google.com/maps/documentation/geolocation/usage-and-billing.

Firebase Authentication

Another Google service that we are using is Firebase Authentication (Google, "Firebase Authentication | Firebase"). Firebase authentication is an extremely flexible solution, supporting email and password accounts as well as Google, Facebook, Twitter etc. There are a host of other features that they also offer, such as multi factor authentication. This authentication solution is easy to use and set up, and well-integrated with our database layer.

Additionally, Firebase Authentication makes it extremely easy for users to reset passwords, generating and emailing links and provides an interface for them to enter new passwords.

Firebase Authentication comes as an SDK that can be easily integrated into the front end by providing an API library as well as a UI library, as well as the backend with a Python library.

Furthermore, Firebase Authentication handles session tokens, meaning that we can easily query Firebase to retrieve the user token of the user currently signed in from the frontend, which we can convert to a user id in the backend.

We are also using a free trial of Firebase Authentication for this project, so its inclusion did not impact the results of this project. In the future however, each user past the 50,000 monthly active users amount will cost a certain amount determined by which tier you are using. More details about pricing can be found here: https://cloud.google.com/identity-platform/pricing.

Firestore Database

Finally, for our database layer, we chose to use Firestore Database (Google, "Cloud Firestore | Firebase"). This is a NoSQL document database that provides performance and scalability for applications. Although NoSQL might not be the best suited schema for our application, where something like PostGres may have been a better fit, its integration with our infrastructure layer as well as Firebase Authentication make this a convenient option. Additionally, the NoSQL makes the data structure easy to understand, and removes the need for the design of a complex relational database.

Additionally, similarly to the Places and Authentication APIs, Firestore Database comes with a convenient and easy-to-use SDK. This includes a Python library that can be used to easily fetch from the database, as well as insert and modify.

Firestore Database is offered as part of the Google Cloud services, so falls under the same account as both Firebase Authentication and the Google Places API. Our current usages for our application fall under the free tier, however, after exceeding quotas for data storage, reads, writes, deletes and network egress, Google starts charging based on location of servers. More details on the pricing model of Firestore Database can be found at: https://cloud.google.com/firestore/pricing.

Implementation challenges

The Firestore database structure was a small learning curve at the start. It stores data in collections and documents, a document is similar to a JSON object, while a collection is a group of documents. There also exists subcollections which are attached to a document. So a collection consists of documents and documents can also have subcollections and this is how the database is set up. For our application we have a collection of conferences and in those is a number of conference documents containing JSONified information and subcollections for tasks, forums, etc and the tasks subcollection contains a document of tasks and it cascades in this fashion. So to access any subcollection you must first go through the higher document first, unlike an SQL database where you can connect any table to another. We cannot access any subcollection directly, this required us to pass back and forth the ids for conferences and tasks or forums to be able to get any information we needed. We also had to convert the data from the object output by the database to a dictionary that Python understands.

Deleting collections or documents was not trivial either as according to their documentation Firestore does not delete subcollections when deleting a collection, so first you have to delete all documents in a subcollection before deleting that subcollection, then you can delete the surrounding collection and other documents. This required burrowing down into the database and deleting from the bottom up.

We also found it difficult to manage both implementing features in the frontend on time, and also maintaining a good user interface. This was due to frontend coding being outside of our main areas of expertise, and as a result we found ourselves often having to learn and implement at the same time. As a solution, we decided to stick to a minimalist and simple design, consisting mainly of cards, buttons, and menus. We also kept a simple colour scheme of mostly white and blue, with a few extra colours for special buttons and elements. These early decisions helped us in making sure features were delivered on time during sprints, while not significantly harming user experience.

Installation/User document/manual

Build and running instructions can be found within the Readme in the Github repository. To begin, please clone the git repository.

Installation and running:

Backend

1. Please ensure that you have Python3 installed. More information can be found at: https://realpython.com/installing-python/.
2. Now we will install the dependencies required for the backend. These can be found within the root project directory, in the `requirements.txt` file. To install these dependencies we will need to use PIP, the package installer for python. This should be included with most recent Python releases, but if not installed, please find more information at the following link: https://pip.pypa.io/en/stable/installation/.
3. Run `pip install -r requirements.txt` to download all dependencies listed in `requirements.txt`.
4. Then, please navigate to the backend directory.
5. To start the application, please run `python3 app.py`.
6. Note: The default port for the application is '8001'. If you wish to change the port of the backend, please open `app.py` in a text editor, and navigate to the end of the file. In the main method, please modify `port` in the `app.run` method call. However, all the fetch requests in the frontend are hard-coded to use the default port of 8001, so you would have to also modify all of those.

Frontend

1. Please ensure that you have Node.js and NPM installed, as these are required to manage packages used by the frontend, as well as running. More details can be found at the following link: https://docs.npmjs.com/downloading-and-installing-node-js-and-npm
2. Second, navigate to the frontend directory.
3. Run `npm install` to download the required frontend dependencies.
4. To start the frontend, please run the following command in the same directory: `npm start`.

User manual:

Note: Screenshots included in 'Functionalities' section

• Account creation

- Login: This is the first screen you will see when you load the application, and have not logged in to your account. Here, you must enter your account's email and password to login. If you do not have an account, click 'Sign up!' to navigate to the registration page.

- Register: To create an account, you will need to enter a name, email, password, date of birth, address, and gender. You must also choose which suite you are making account for (volunteer, manager, or organiser).

• Editing your profile

- My Profile: The ‘My Profile’ button in the header bar takes you to your profile page. Here, you can edit your personal information (name & gender), skills, availability, and conference preferences.

• Dashboard & joining/creating conferences

- Dashboard: Users can view conferences in the system from the dashboard, which can be accessed through the ‘Dashboard’ button in the header bar. On this screen, conferences can be filtered by their distance from your location. If you are using the volunteer or manager suites, you will be able to send a join request to a conference on this screen. If you are an organiser, you will be able to approve these requests on the ‘applications’ page of your conference.

- Conferences: This screen can be accessed from the ‘Conferences’ button in the header bar. Here, organisers can see their created conference, and volunteers & managers can see their joined conferences. Users can also navigate to the conferences listed here.

• Tasks & volunteers

- Tasks: In a conference, an organiser can create tasks from the ‘Tasks’ page. After creating a task they can view, assign, and unassign volunteers/managers, or delete the task by using the appropriate buttons. Volunteers and managers can only join tasks or view assignees.

- My Tasks: Volunteers & managers can view their assigned tasks from this page. They can also choose to leave a task they have joined from this page.

- Volunteers: Users can view all the volunteers and managers in a conference. If you are the organiser of the conference, then you can choose to promote a volunteer to a manager on this page.

• Forum

- Forum: On this page, users can access the announcements or questions forum for a conference.

- Announcements: organisers can make announcements on this page using the ‘create post’ button. All account types are able to respond to announcements using the ‘reply’ button.

- Questions: all users are able to ask questions on this forum using the ‘create post’ button. All account types are able to respond to posts using the ‘reply button’.

• Schedule

- Schedule: Organisers can add events to the conference schedule using the ‘add event’ button. These events cannot have overlapping times. For each event, organisers can view authors, add authors, or delete the event. Volunteers and managers are only able to view the schedule, events, and authors.

• Attendance & feedback

- Log Attendance: Volunteers and managers can send attendance requests to the organiser on this page. They must enter a date, a number of hours worked, and a description for the request.

- View Attendance: Here, organisers can access the attendance of all members of the conference, and managers can access the attendance of all volunteers. Organisers and managers can rate and give feedback to a user after clicking ‘view’. Organisers can also view, decline, and approve a user’s attendance request. Managers can only view an attendance request.

- My Feedback: In each conference, a volunteer or manager is able to access a ‘My Feedback’ page, where they can see all the feedback they have received in that conference.

• Account deletion & resetting your password

- From the ‘My Profile’ page, users can delete their account by clicking ‘Delete Account’. Users must re-enter their password to complete the process.

- On the ‘Login’ page, users can click ‘reset password’ if they have forgotten their password. On this next screen, you can send a password reset request to an email.

References

1. Google. “Cloud Firestore | Firebase.” *Firebase*, 2019, <firebase.google.com/docs/firestore>.
2. Google. “Firebase Authentication | Firebase.” *Firebase*, 2019, <firebase.google.com/docs/auth>.
3. Google. “Overview | Geolocation API.” *Google Developers*, <developers.google.com/maps/documentation/geolocation/overview>.