Real Time Automatic Crowd Sourcing

Presented By

Hassan Abdallah

15-12-2019

Introduction

I will review the Idea of this Project, application architecture and Database Modeling by showing Conceptual and Physical Data Models. Although, I will show some services with associated interfaces. And finally, the project implementation.

This project is about Crowd Sourcing Which aim to extract brilliant Ideas from a group of people. There are three types of users:

- A **Monitor User** Creates a Workshop and share an entry key to whom will participate in this workshop.
- A **Participant user** takes the key and participates in this workshop which he can write down his idea and rate other's ideas, ideas are shuffled automatically at the end of each round. Best rated ideas will be automatically chosen and groups are created for each chosen idea to work on it and participants can enter and exit groups, the monitor can kick anyone from a group.
- An **Admin User** manages users their roles and privileges and workshops preferences.

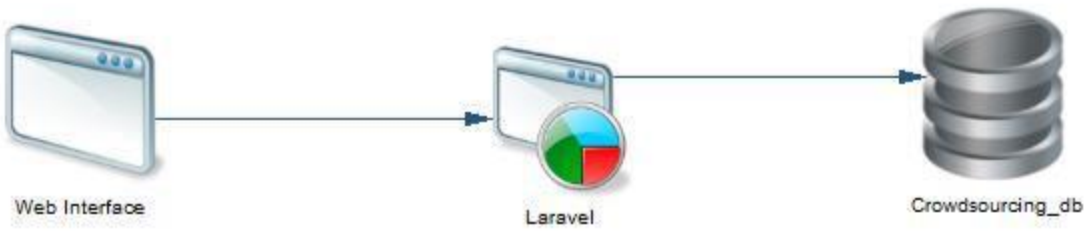
One of the strength of my project, is that the data are always updated automatically in front of the users (the ideas, scores, groups), this is performed using ajax calls, which enable a complete real time application without the need of page refresh or reload at all.

Chapter 1: Application Architecture

MySQL DBMS was used in this web app.

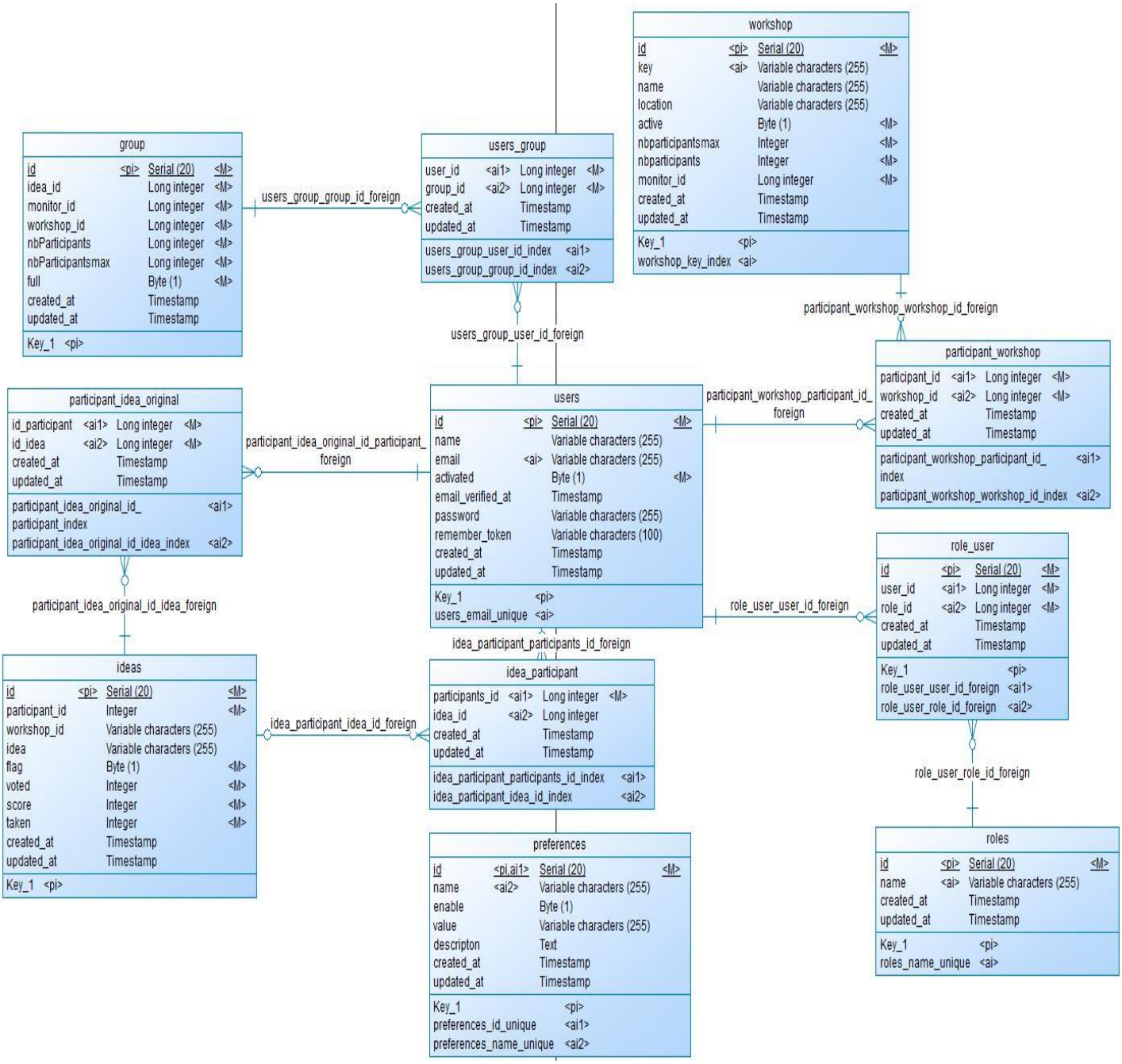
In the application layer Laravel framework (PHP based) is used and contains a query builder (Eloquent).

They are connected using MySQL Driver.

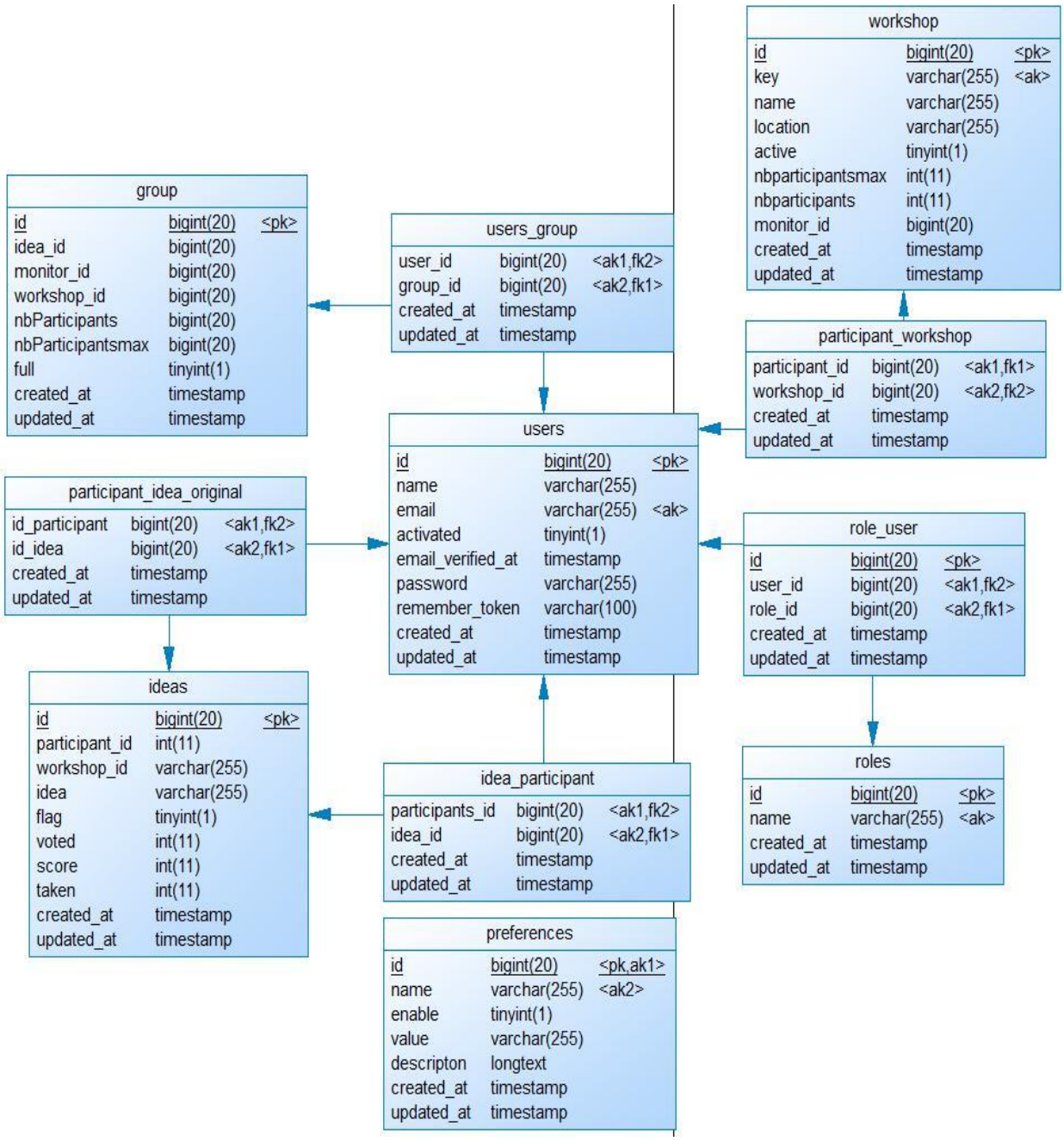


Chapter 2: Database

Conceptual Data Model:



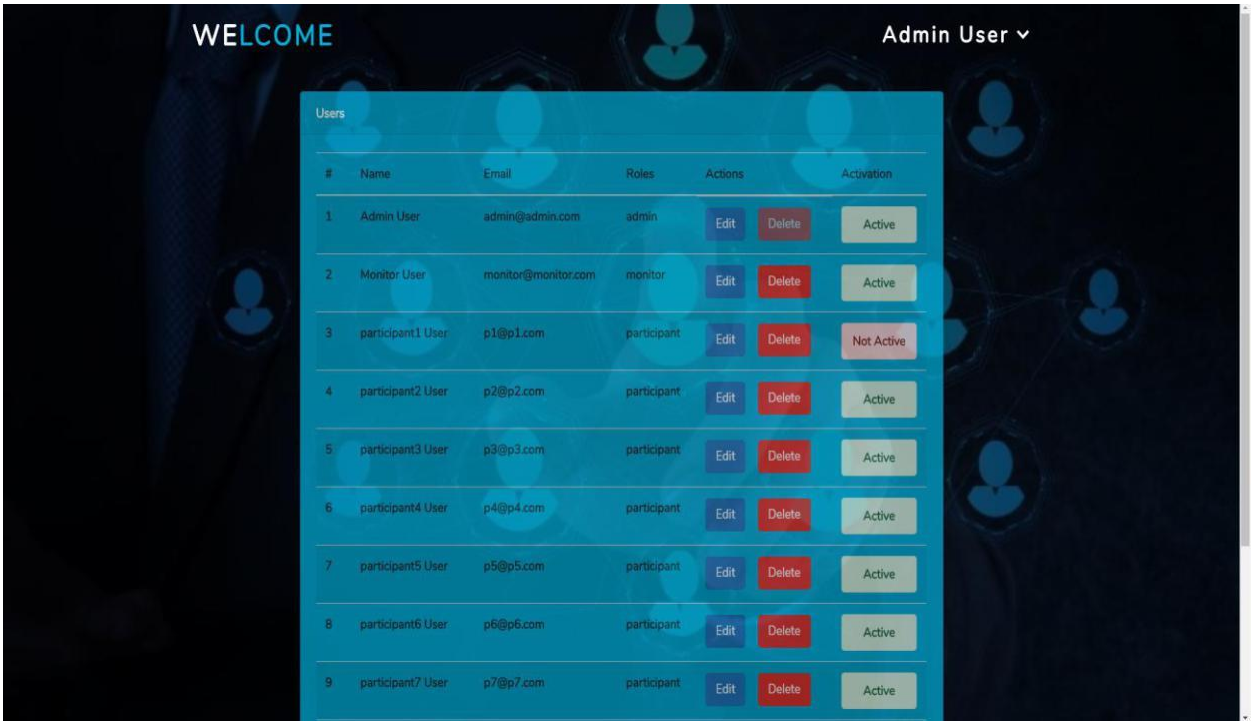
Physical Data Model:



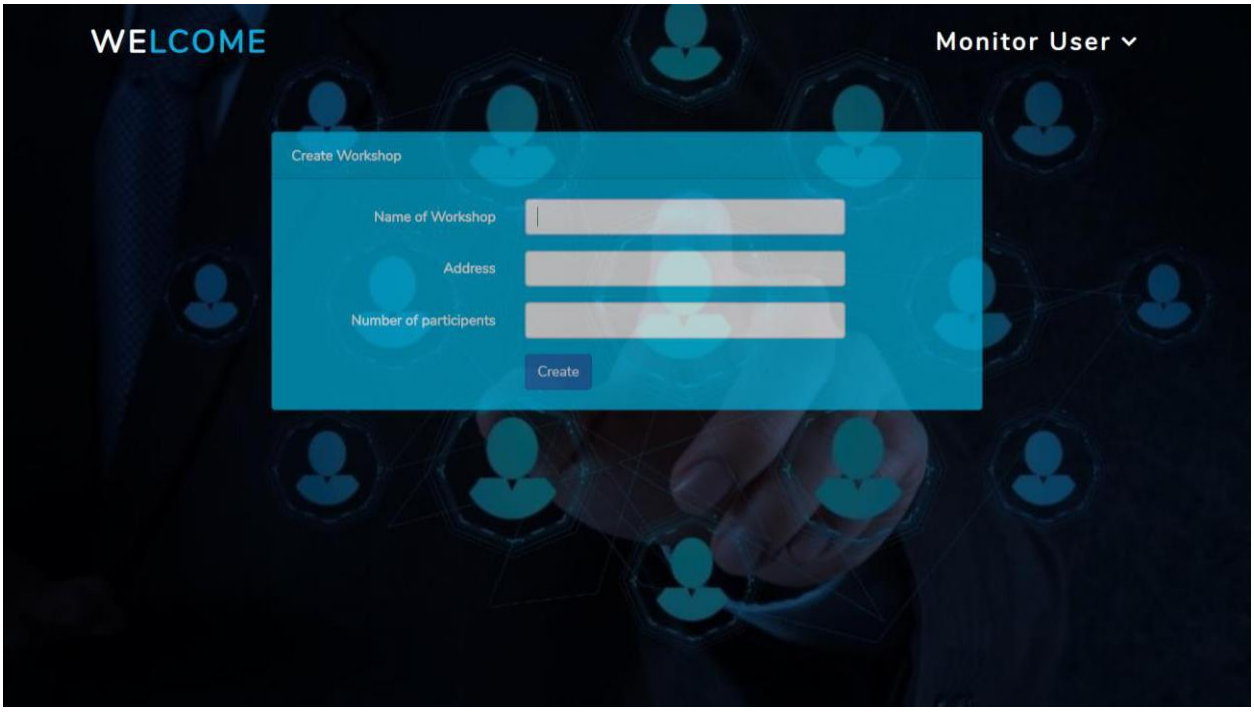
Chapter 3: Services And Interface modeling

- Users Registration and Login.

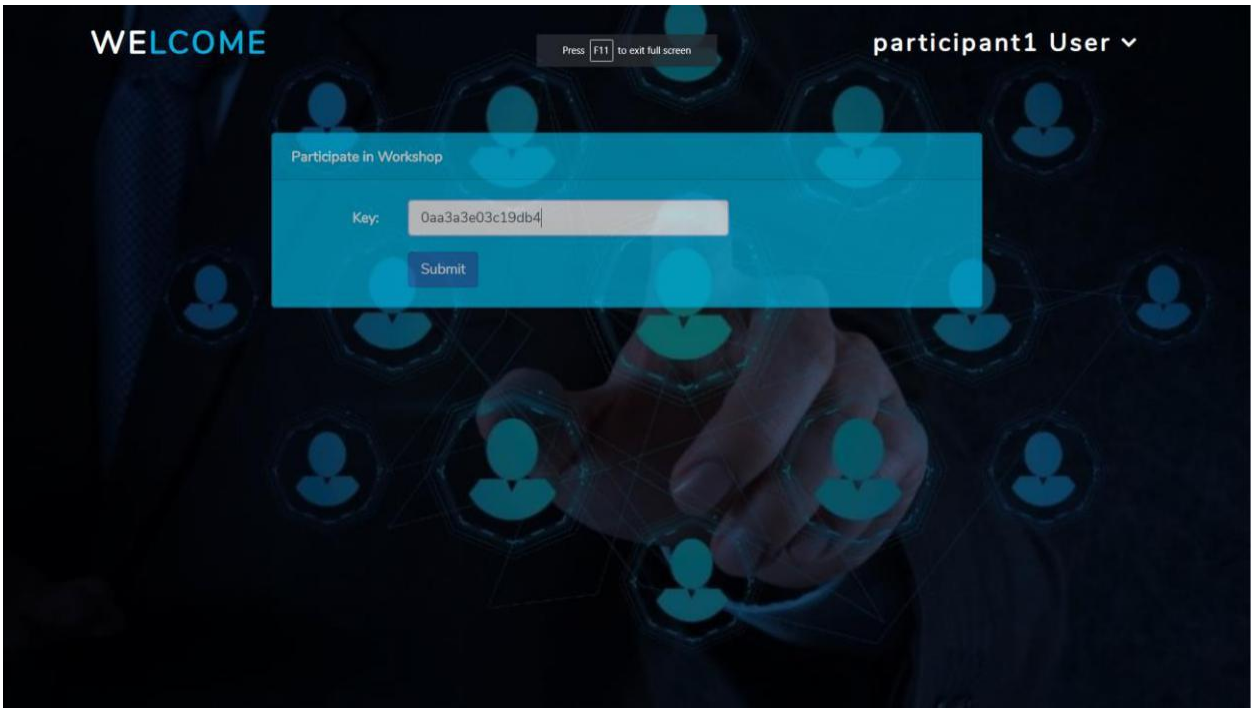
- Users Management (Admin):



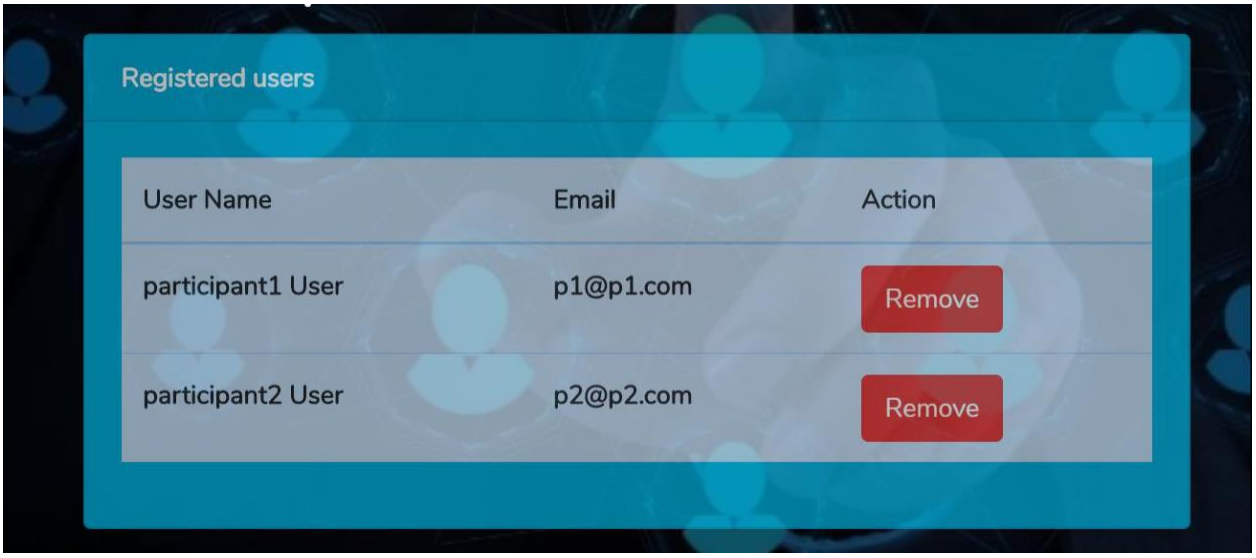
- Create Workshop (Monitor):



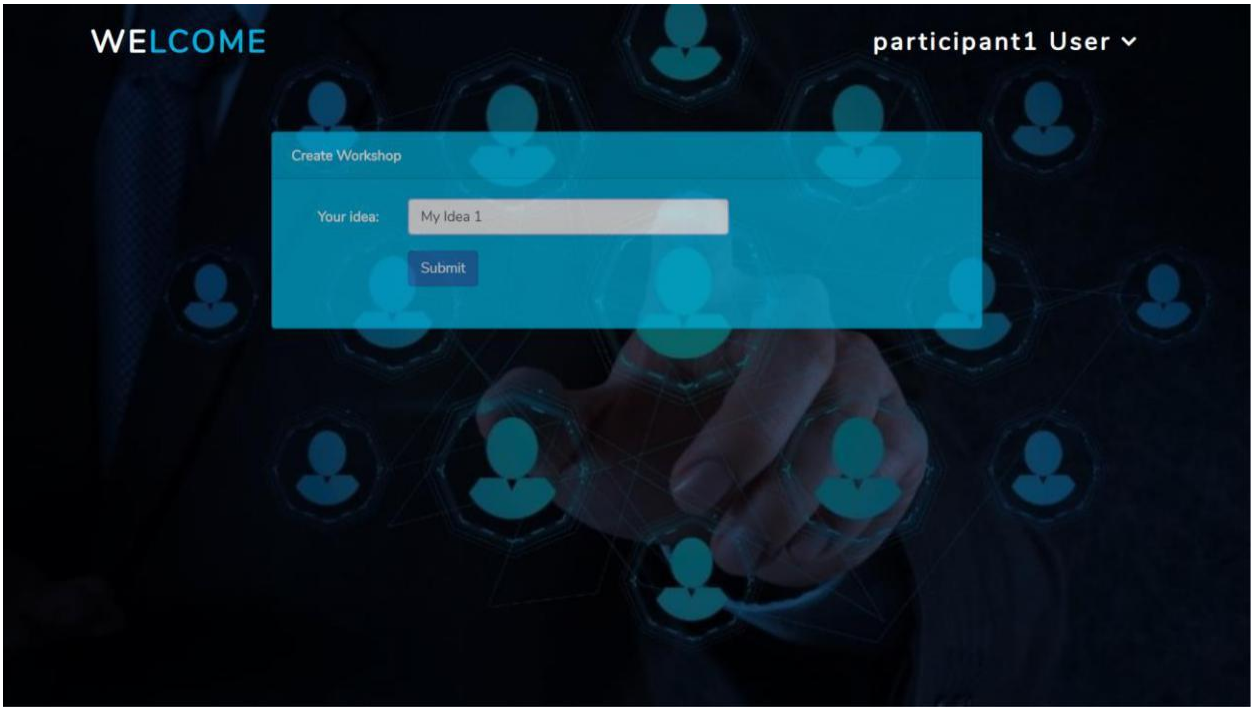
- Participate in Workshop (Participant):



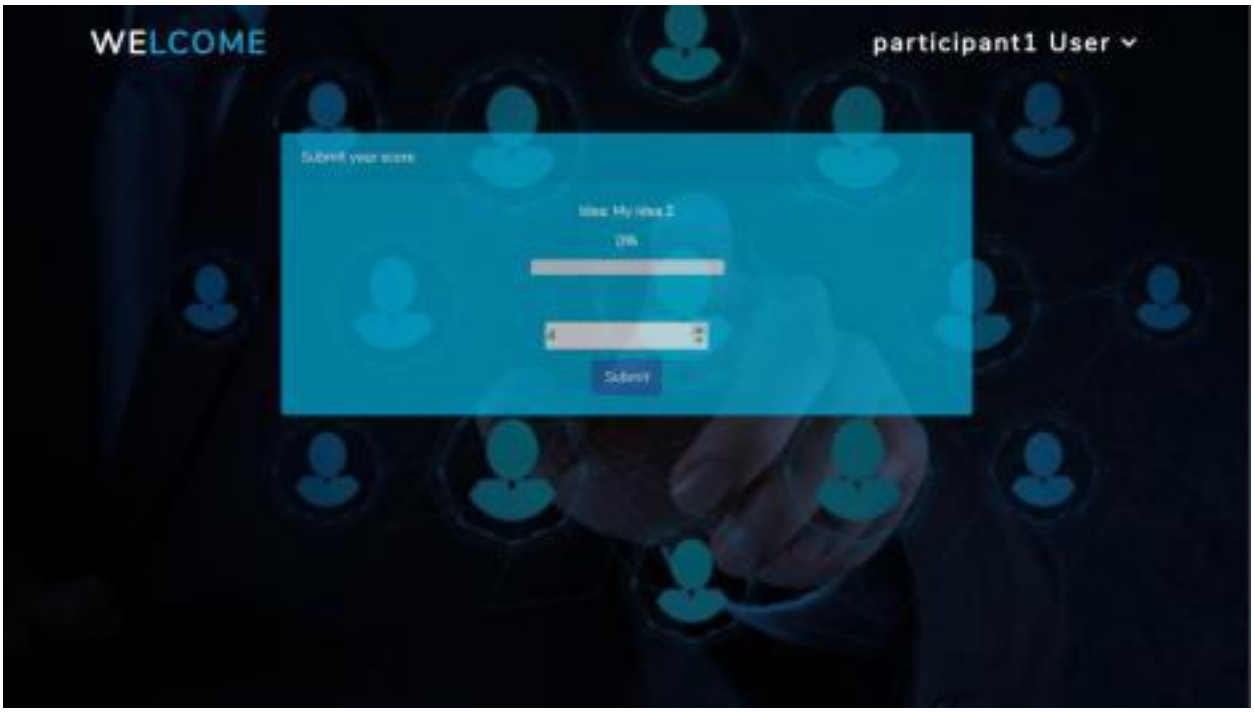
- Review And Remove Users From Workshop (Monitor):



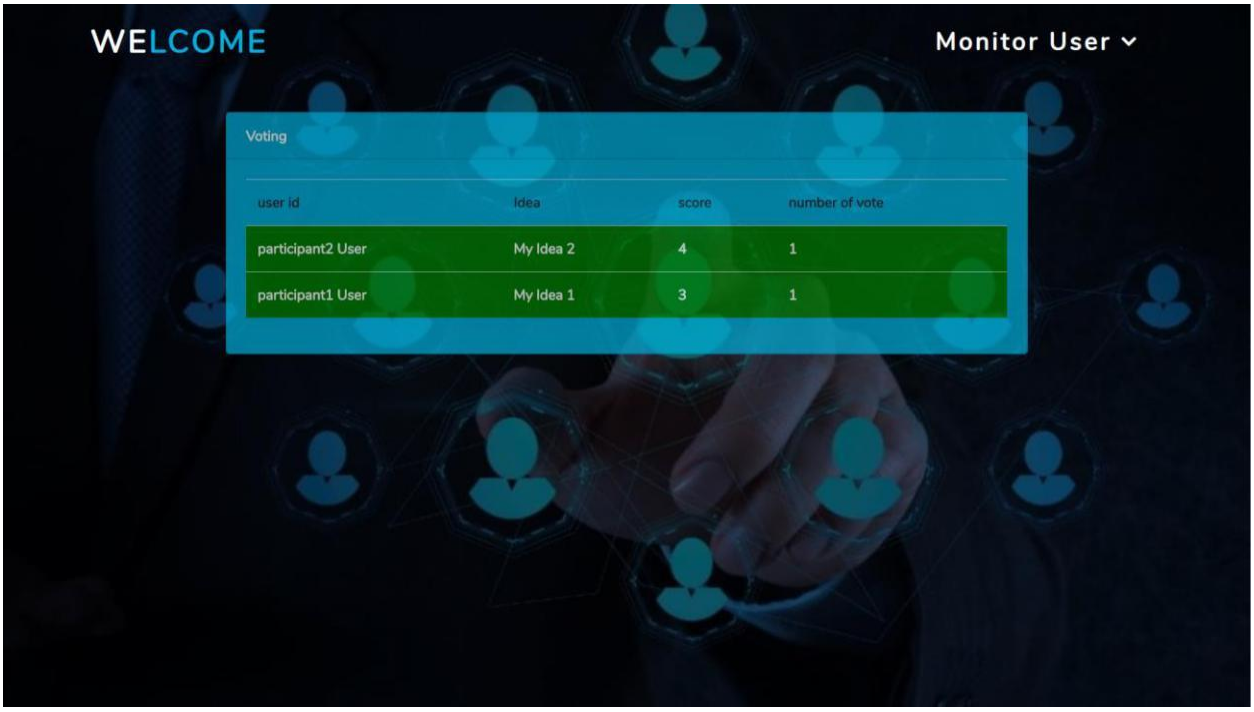
- Submit Idea (Participant):



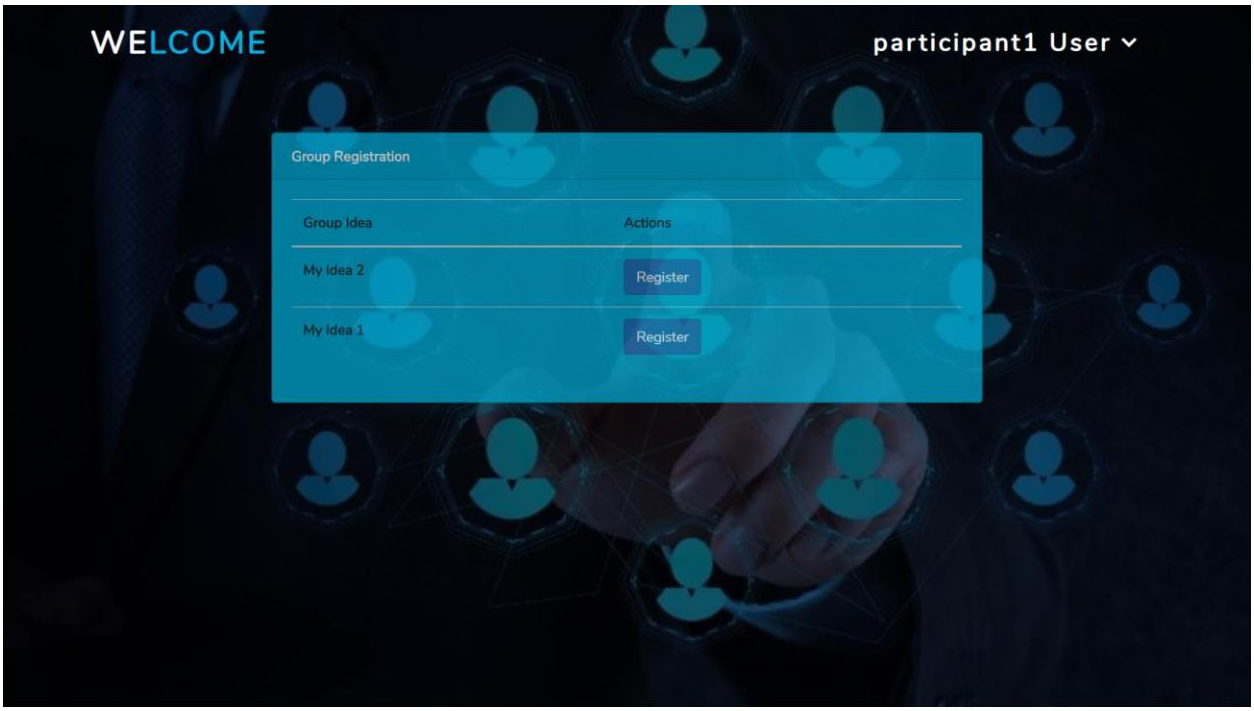
- Rate Idea (Participant):

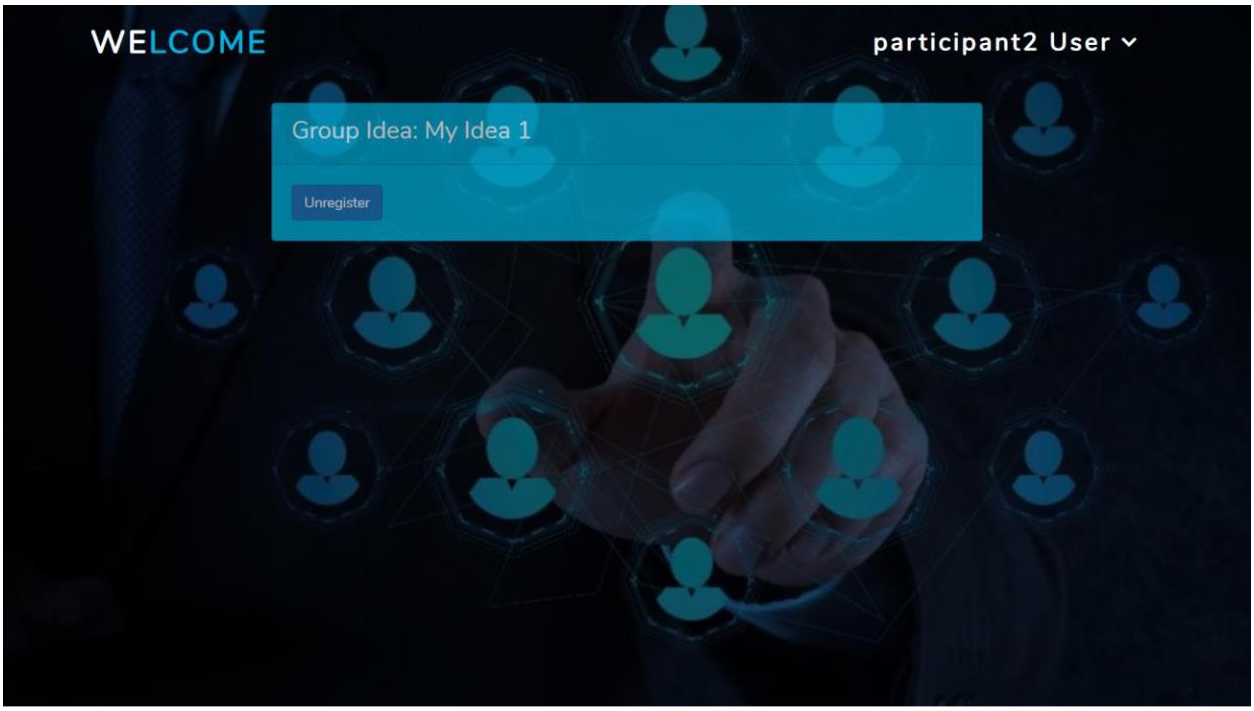


- Real Time Ideas Rating and Top ones Review (Monitor):

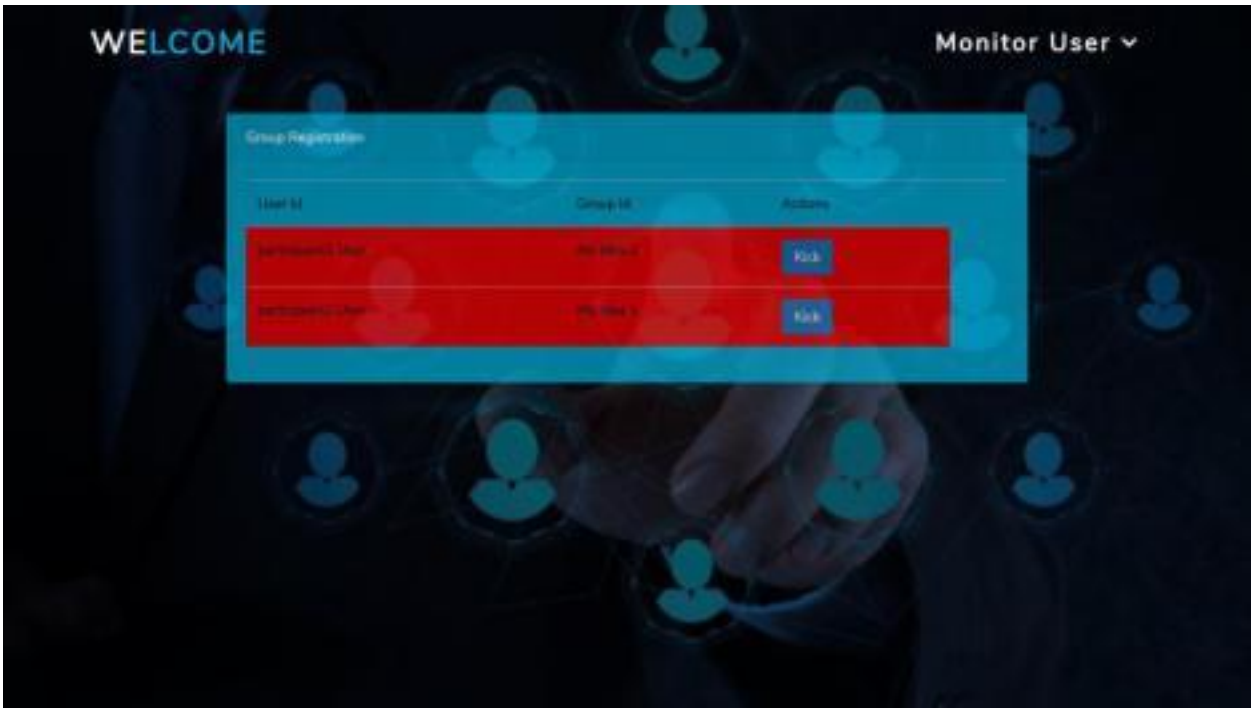


- Group Registration (register/unregister)(Participant):

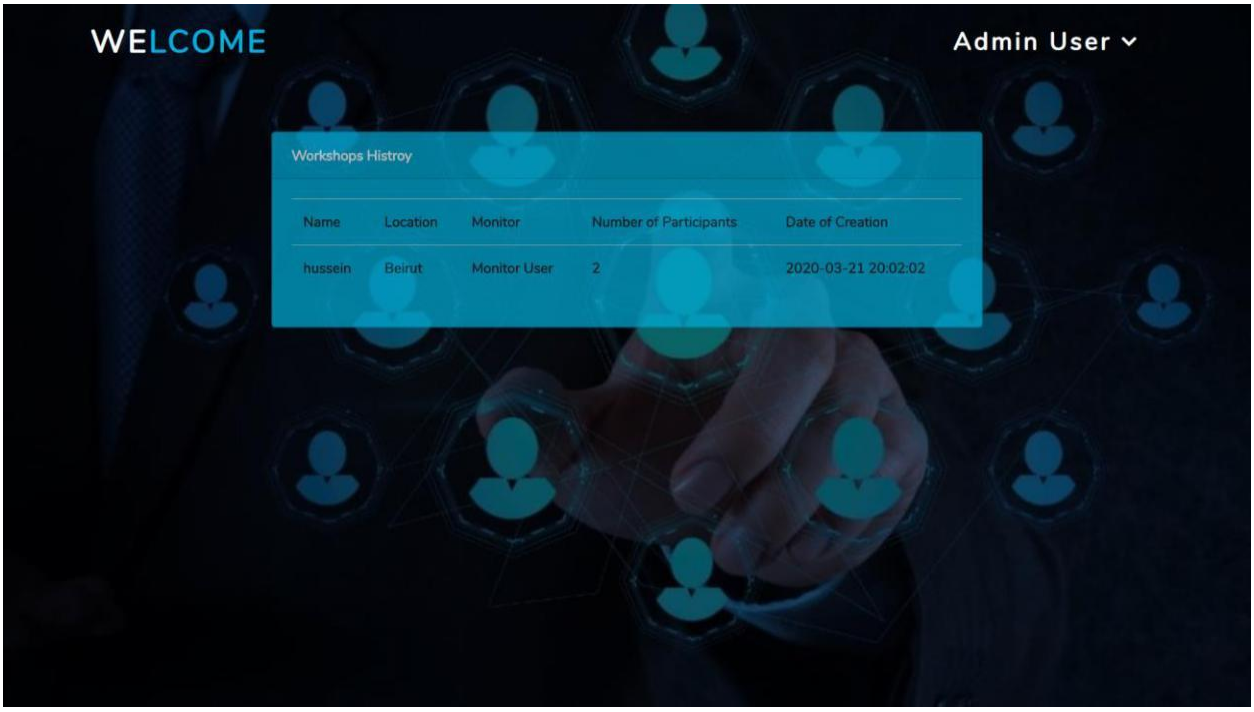




- Group Registrations and remove (Monitor):

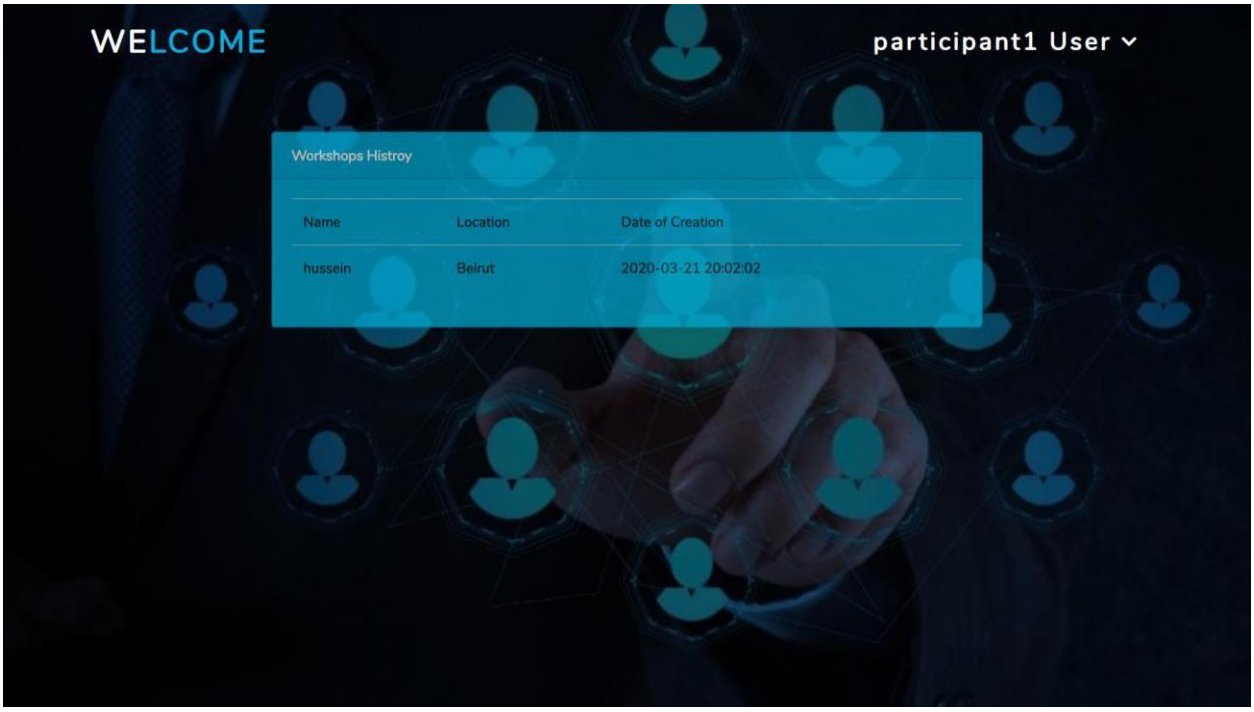


- Workshops History (Admin View):

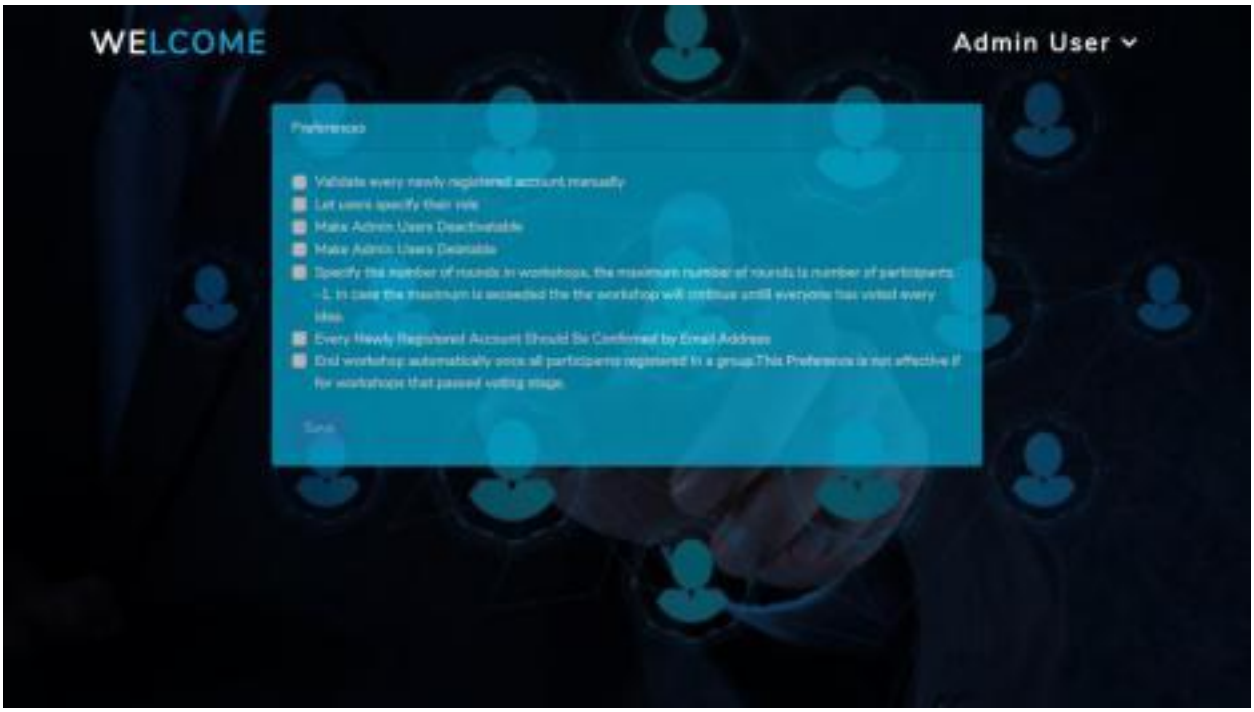


- Workshops History (Monitor View):

- Workshops History (Participant View):



- Preferences (Admin):



Chapter 4: Implementation and Testing

I used MySQL Workbench for database implementation and testing.

I “debugged” triggers printing values in MySQL Workbench to make sure everything works and executes in the right time, such as:

- Deactivation trigger which automatically deactivates workshop at a certain stage (admin can enable/disable this trigger from preferences).
- Creating groups to work on chosen ideas.
- Automating increment/decrement operations on number of participant field in workshop and group tables on insert/delete operations.

I created appropriate indexes and foreign keys (on delete cascade) to ensure referential integrity.

I created 3 database users admin, monitor and participant and granted each user permissions associated with his need only.

The application was developed using laravel framework ver. 6.9.0 which is a MVC web Applications framework based on PHP, HTML, JQuery and CSS.

Conclusion

This course helped me develop a secure, consistent and powerful application whatever was the technology or interface used I learned to control and secure the app from a database point of view.