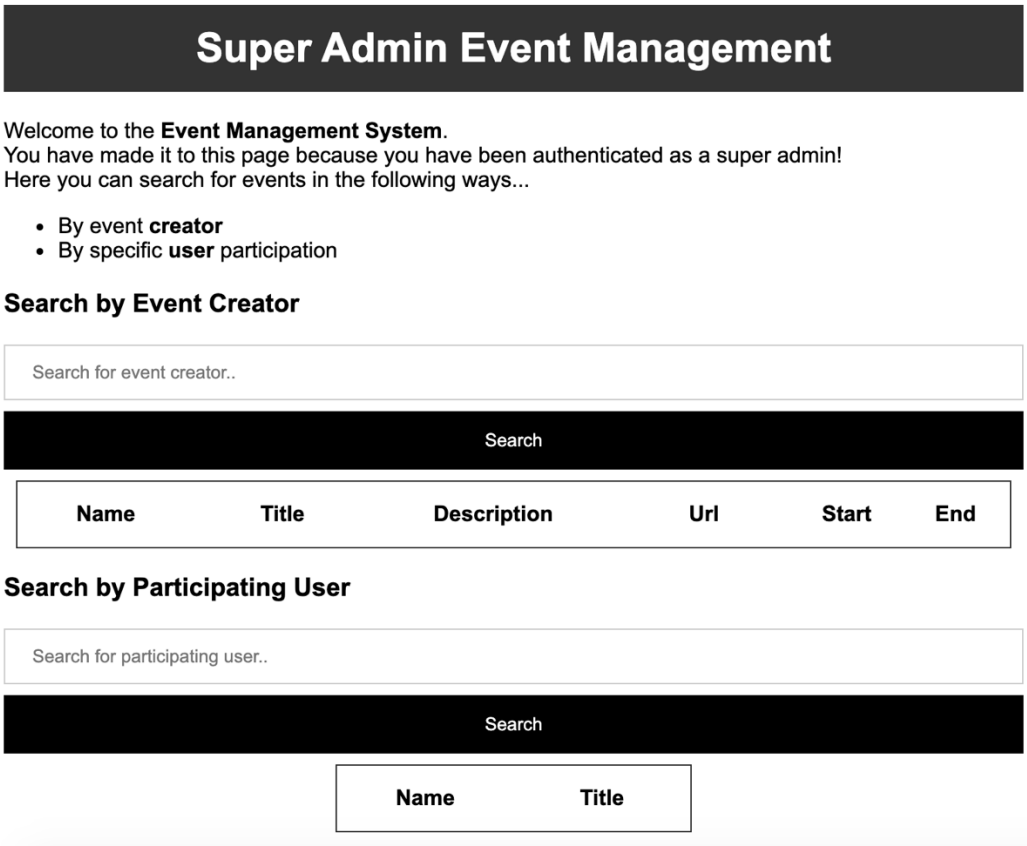
# *EVENT MANAGEMENT SYSTEM*

COP4710 Final Report

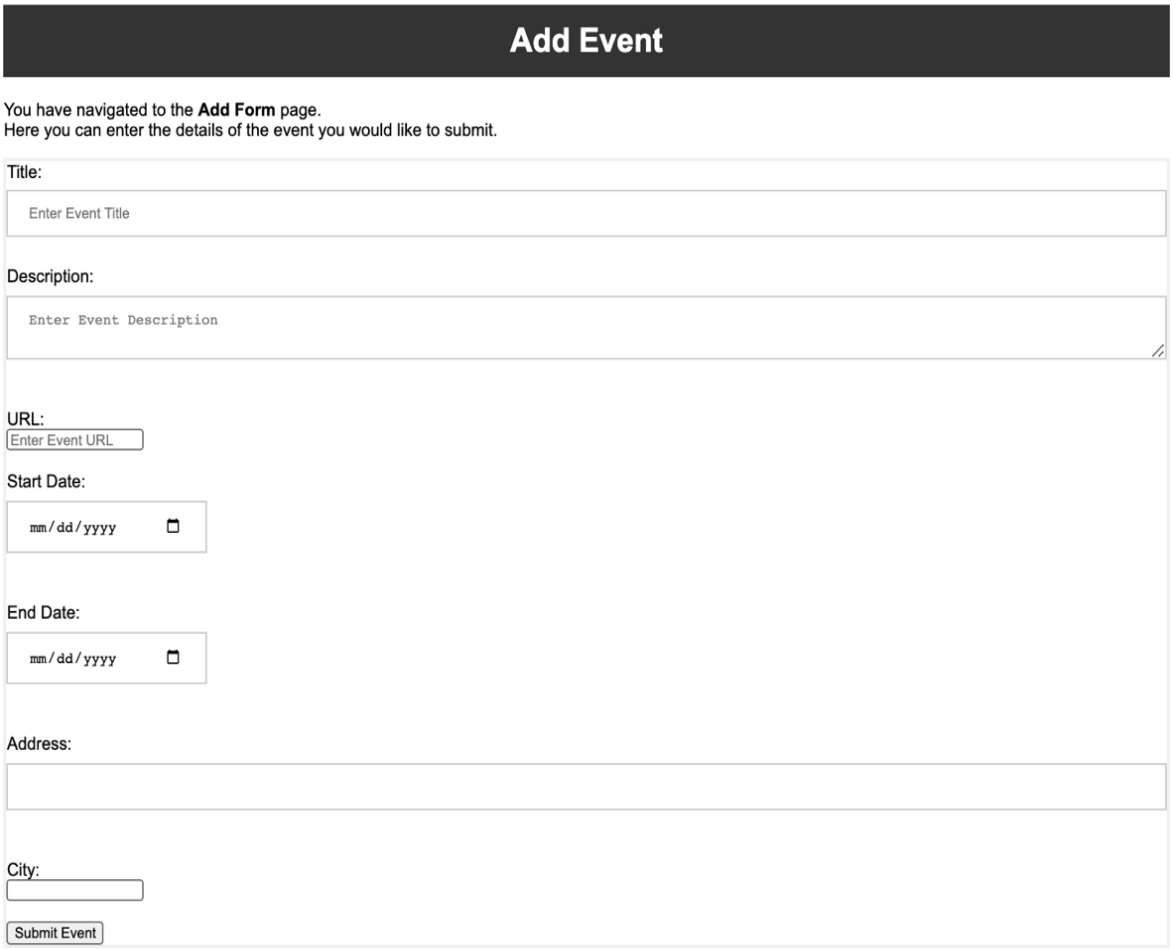
*Izabela Olejnik, Alex Ogilbee, Brett Parker, Mark Pedroso*

The Event Management System we have created for our final project has 3 main users: Super-Admin, Admin, and Participant and 2 main entities: Events and Users. The Super-Admin has the highest level of access to the Event Management System database. The Super-Admin has the ability to query the event database based on a particular admin and to list the events a particular user has participated in. The Admin has the ability to create events and query events, while finally the Participant has the ability to query events.

The below images are the interfaces that each of the user types are able to see when they are authenticated. To begin, the super-admin can query events based on the event creator or by the name of a user that is participating in the event. The query results will be shown in the tables.

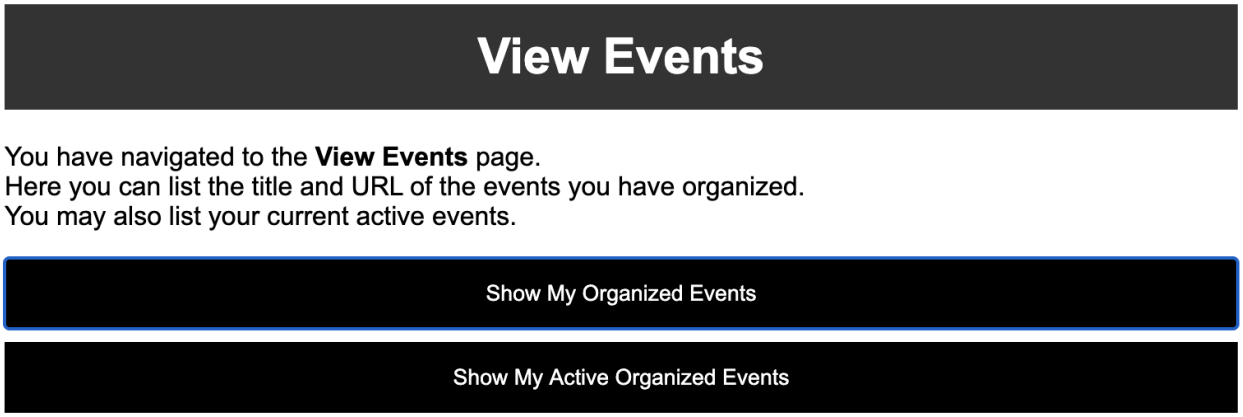


The admin, or event creator, can create events and view events. As the admin creates events, he must enter information about the particular event. The form requirements can be seen below. The event creator will fill in the form about the event they would like to add and then click submit. Upon clicking submit, the data will be sent to the database for participants to be able to register for. After submitting an event, the admin can then look at events.

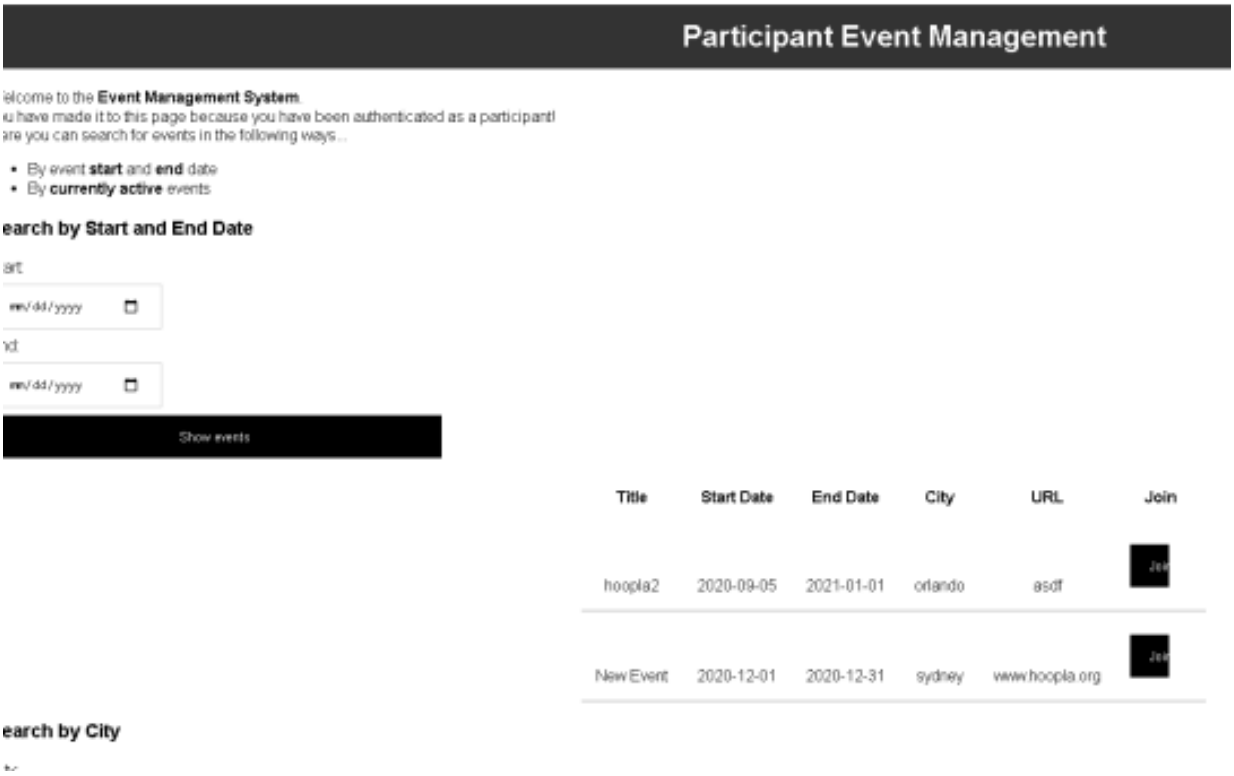



The admin can query events based on events they have created as well as currently active events. The admin can do so from selecting between one of the two buttons shown below. The options are “Show My Organized Events” and “Show My Active Organized Events”.

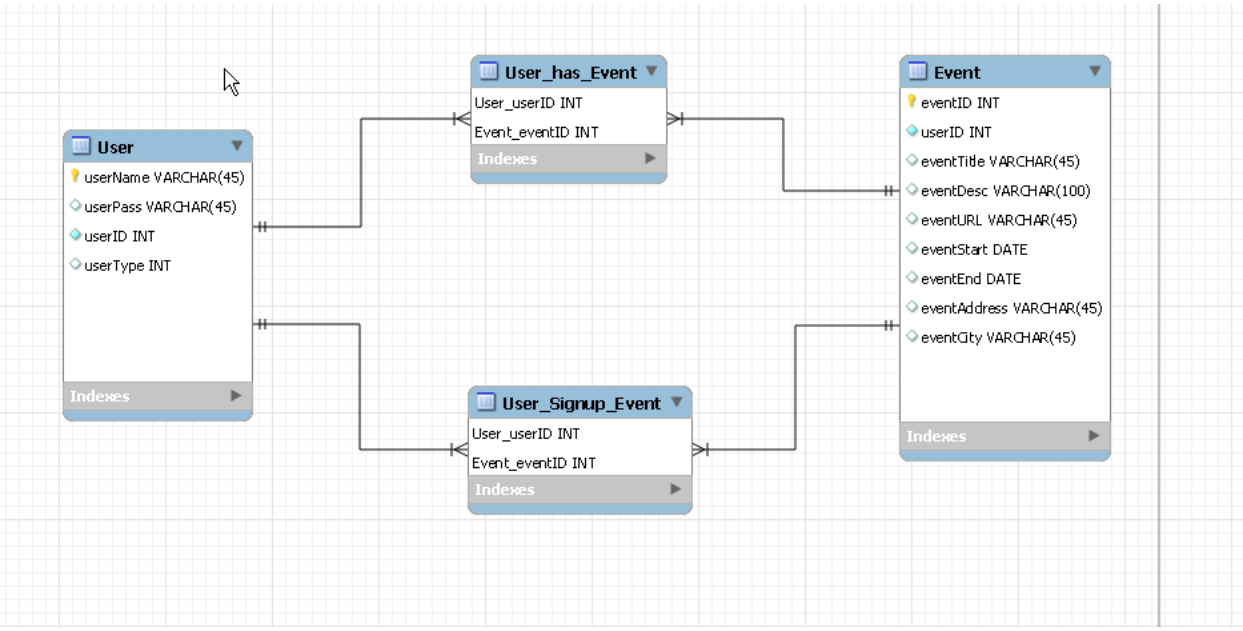
After clicking an option, the admin will be shown the title and URL of the respective event. The information will be pulled from the database to show to the admin.




Finally, a participant will have the option to query events based on certain criteria as well. The participant can query interesting events based on the start and end date received from the database event table. On the other hand, the participants can also query the event database based on the currently active events in a particular city.



Below is an ER diagram that displays the tables we have used in developing our database. The two entities that are shown are User and Event. The users and events are related to each other based on the user having an event or the user signing up for events. The users that are able to have an event, in other words create the event, is the admin. The users that can sign up for events are the participants. The users are distinguished through the userType int value contained in the user table. Through this value, the application is able to distinguish between the users. The super-admin does not create events or signup for events, they are simply able to query event information. The user table also contains the user's username and password in order to authenticate the user on the application. Additionally, the userId is contained in the table to be able to distinguish between the users in the relationships. For example, the user_has_event table contains information needed regarding the relationship between user and event. The user_signup_event also contains information needed regarding the relationship between user and event. Both relationships contain the userId and the eventID. The event table has all the information that is required by the admin to enter about an event that they are entering. The table includes: event and user ID, event title, description, url, start date, end date, address, and city.




To create our application, we divided our work into front-end, back-end, and database design. The front-end component of our application was completed utilizing HTML and CSS as well as JavaScript. The back-end component of our application was completed utilizing PHP. Additionally, the database was completed using MySQL and was run on AWS. In order to run the backend locally, we used XAMPP and Wampserver. Our web application was hosted locally using localhost and our projects file path.

The breakdown of the workload is listed below.

HTML pages (izabela - DONE)

Js doc (izabela - DONE)

CSS style sheets (izabela - DONE)

Report (izabela - DONE)

signup (mark - DONE)

login (brett - DONE)

(SA) lookup events by admin (alex - DONE)

(SA) lookup events by specific user participation (alex - DONE)

(A) create an event (brett - DONE)

(A) list self-created events (alex - DONE)

(A) list self-currently active events (alex - DONE)

(U) lookup event based on date (mark - DONE)

(U) lookup event based on city (mark - DONE)

(U) join an event (mark/brett/alex - DONE)

(U) make a user become an admin (mark/brett - DONE)