

# Group Report

College Event Website

COP4710-0001 SPRING 2023

Solo group: Maahee Abdus Sabur

## Table of Contents

Project Description.....	2
Additional assumptions.....	3
GUI.....	4
ER Model.....	15
Constraints captured in ER model:.....	15
Constraints captured outside ER model:.....	16
Relational Data Model.....	17
Populating Tables with Sample Data.....	22
Note.....	25
SQL examples and results.....	25
Constraint Enforcement.....	30
Advanced Features.....	32
Conclusion.....	33

# Project Description

## Problem:

Most universities in the country host events around campus and off campus. These events are organized by college students in most cases. Students are clustered RSOs (Registered Student Organizations) by different organizations, clubs, fraternities around campus. These events could be of different types: social, fundraising, tech talks, etc. Now, each university has a website where they post their events for the upcoming weeks. One needs to check the website to add each event to his/her calendar. These events are just official events and not all events around the university are included. Another limitation is that one has no way to track weekly events.

## Project Description

You are asked to implement a web-based application that solves the problems. Any student (user) may register with the application to obtain a user ID and a password. There are three user levels: *super admin* who creates a profile for a university (name, location, description, number of students, pictures, etc.), *admin* who owns an RSO and may host events, and *student* who uses the application to look up information about the various events.

*Admin* can create events with name, event category, description, time, date, location, contact phone, and contact email address. A location should be set from a map (Bing, Google, open street map) with name, latitude, longitude, etc. To populate the database, one can use feeds (e.g., RSS, XML) from events.ucf.edu. Each *admin* is affiliated with one university, and one or more RSOs. A *user* can request to create a new RSO or to join an existing one. A new RSO can be created with at least 4 other students with the same email domain (university), e.g., @knights.ucf.edu; and one of them should be assigned as an administrator.

There are different types of events (social, fundraising, tech talks, etc.). Each event can be public, private, or an RSO event. *Public events* can be seen by everyone; *private events* can be seen by the students at the host university; and an *RSO events* can only be seen by members of the RSO. In addition, events can be created without an RSO (public events). Such events must be approved by the super admin. After an event has been published, users can add, remove, and edit their comments on the event, as well as rating the event on a scale of 1-5 (stars). The application should offer some social network integration, e.g., posting from the application to Facebook or Google.

When logged in, Student should be able to view all public events, private events at their university, and event of RSOs of which they are member. They will not be able to create events, but should be able to rate, comment and edit (update) their comments for any event.

## **Additional assumptions**

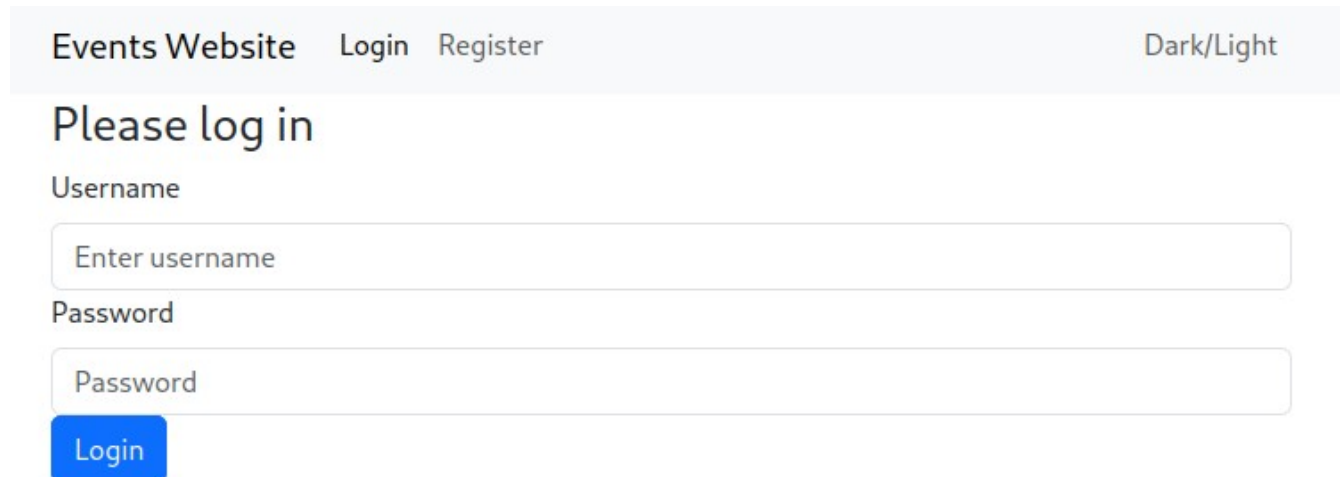
- Superadmin's feed contains ALL events, except for pending public events
- The process of becoming admin is like this:
  1. A student creates an RSO
  2. Superadmin approves the RSO
  3. The student who made the RSO is promoted to admin
- Whether an RSO is active or inactive doesn't affect event creation
- Students see events of RSOs they joined even if RSO admin didn't approve them
- Superadmin approves new RSOs and public events
- Only the student role is registerable
- Superadmin role is manually assigned from database
- Superadmin isn't affiliated with universities/RSOs

# GUI

DBMS: PostgreSQL

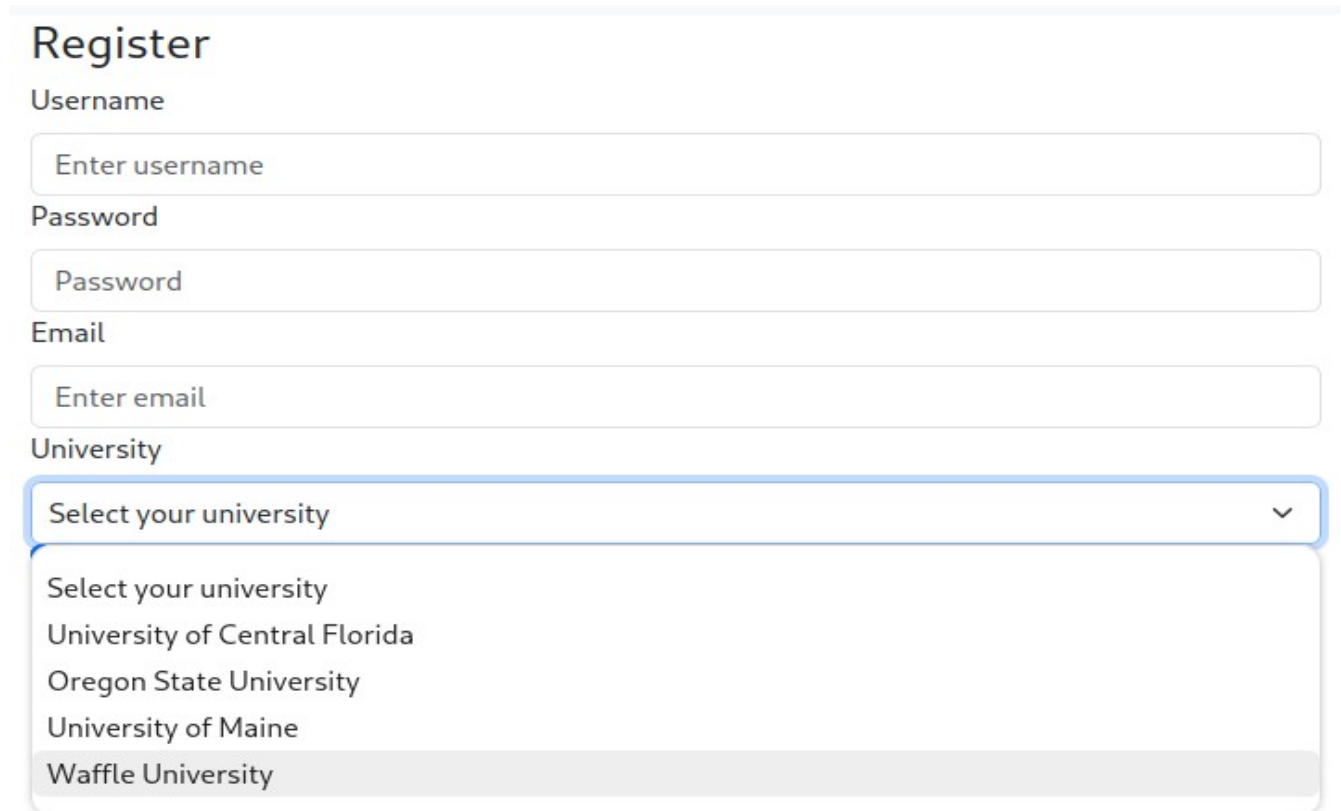
Languages: JavaScript, JSX, HTML, CSS

Platform: NodeJS, React, Fastify, Vite



The login page features a light gray header bar. On the left, it contains the text 'Events Website' followed by 'Login' and 'Register' links. On the right is a 'Dark/Light' theme toggle. Below the header, the heading 'Please log in' is centered. Underneath, there are two input fields: 'Username' and 'Password', each with a placeholder text 'Enter username' and 'Password' respectively. A blue 'Login' button is positioned below the password field.

Figure 1: Login page



The register page has a light gray header bar. Below it, the heading 'Register' is centered. The form consists of four input fields: 'Username', 'Password', 'Email', and 'University'. The 'University' field is a dropdown menu with the placeholder 'Select your university'. The dropdown is open, showing a list of universities: 'University of Central Florida', 'Oregon State University', 'University of Maine', and 'Waffle University' (which is highlighted). The 'University of Central Florida' is also visible above the dropdown list.

Figure 2: Register page



Events Website

Events

Control Panel

Logout


Dark/Light

Welcome, abubakr!


Here are the events:

[University Fundraiser 2043](#)

Join us in raising funds for our university!  
Don't you want to help further education for our future students?


 ليبيا

private


 4/18/2043, 3:44:00 PM

[Coding Workshop](#)

Learn the basics of coding and web development in this hands-on workshop.

 Mali

private

 5/10/2023, 6:00:00 AM

[Guest Speaker: Environmental](#)

[Charity Fundraiser](#)

Figure 3: After logging in: events view

Events Website

Events

Control Panel

Logout

Dark/Light

## Admin controls

New event

## Student controls

Request to join an RSO

Request to create an RSO

View RSOs I am in

View RSO join requests

Figure 5: Student controls

## Superadmin controls

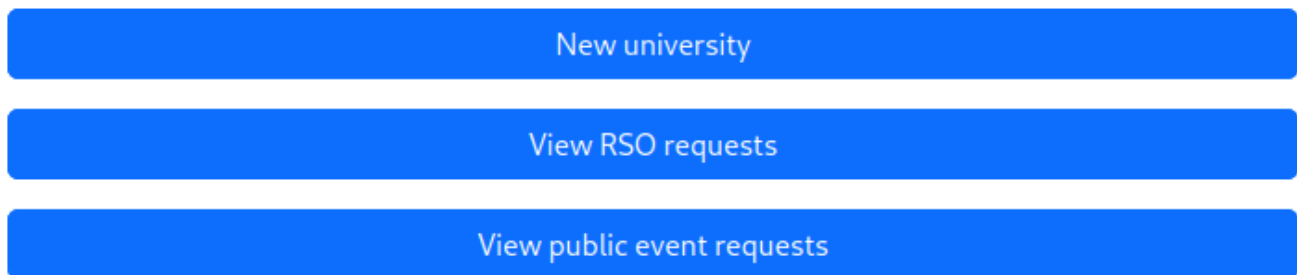


Figure 6: Superadmin controls

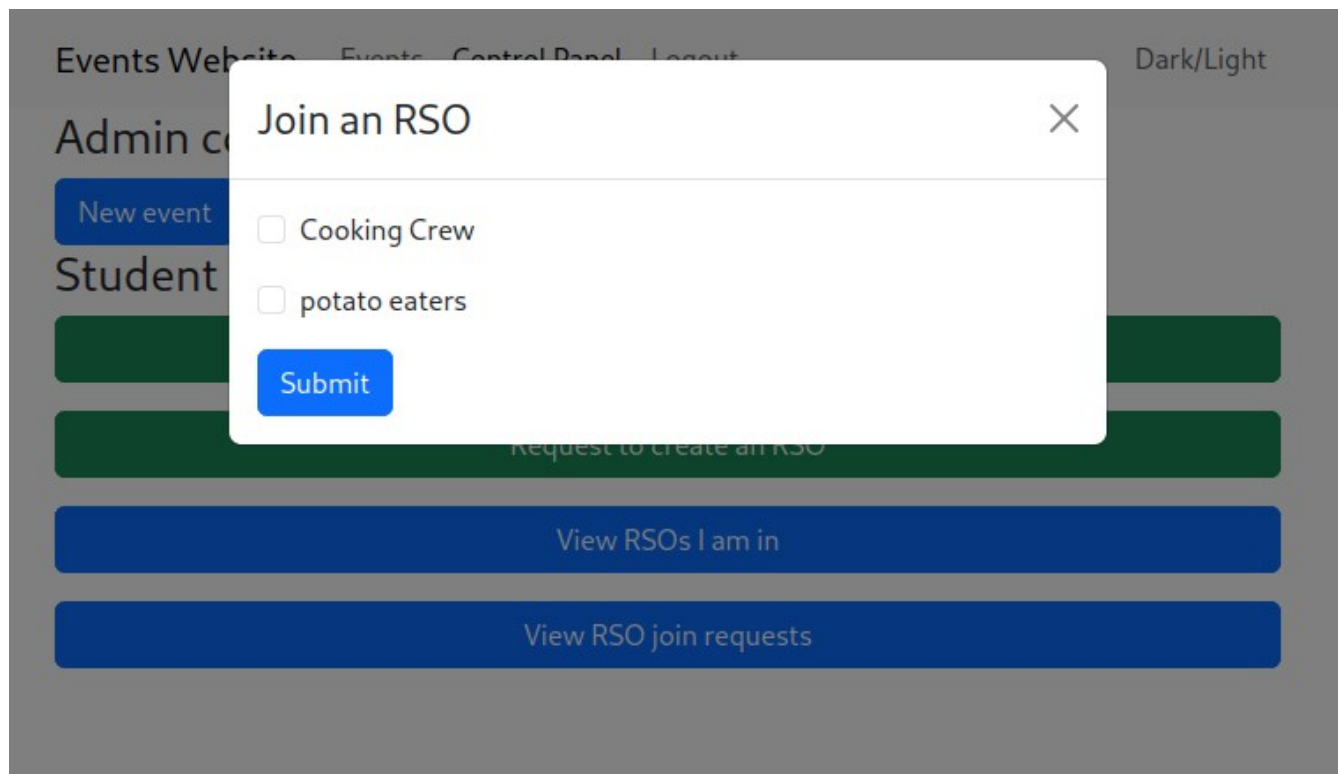


Figure 7: RSO join modal

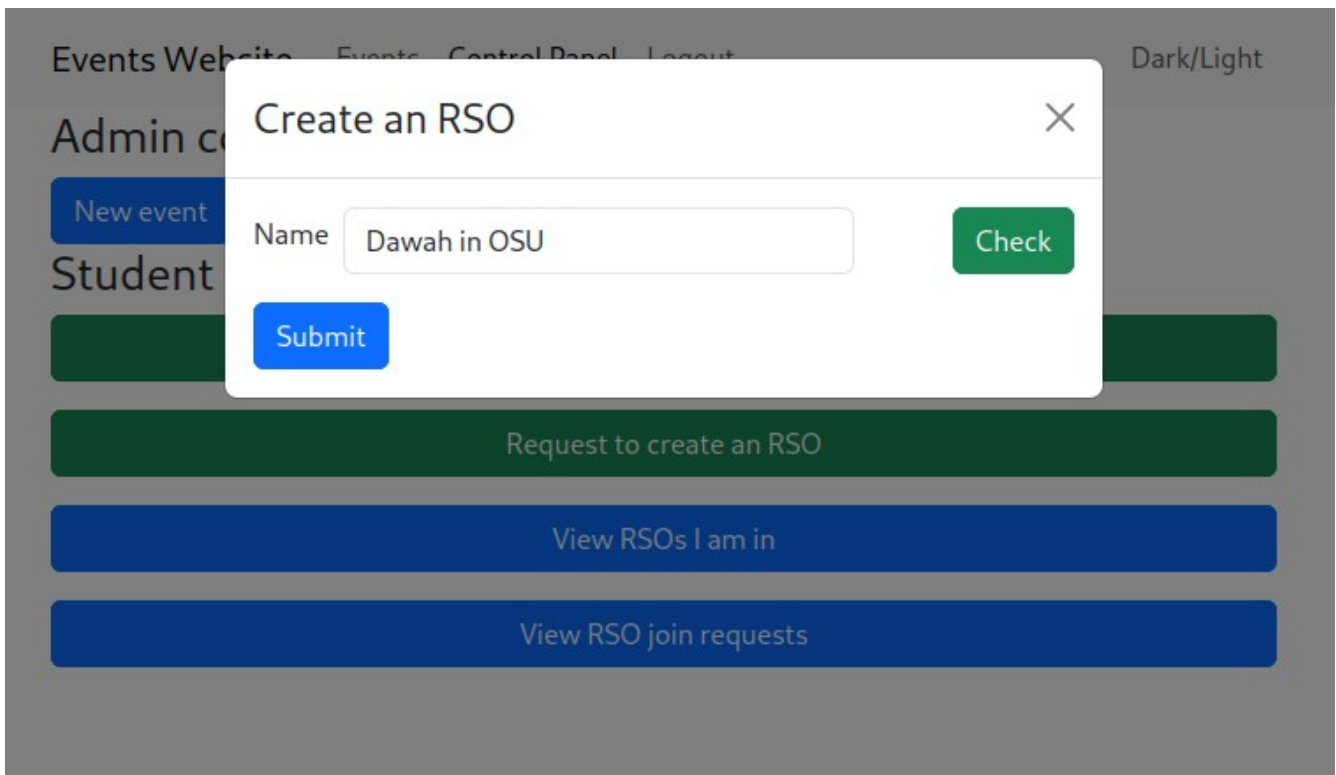


Figure 8: RSO creation modal

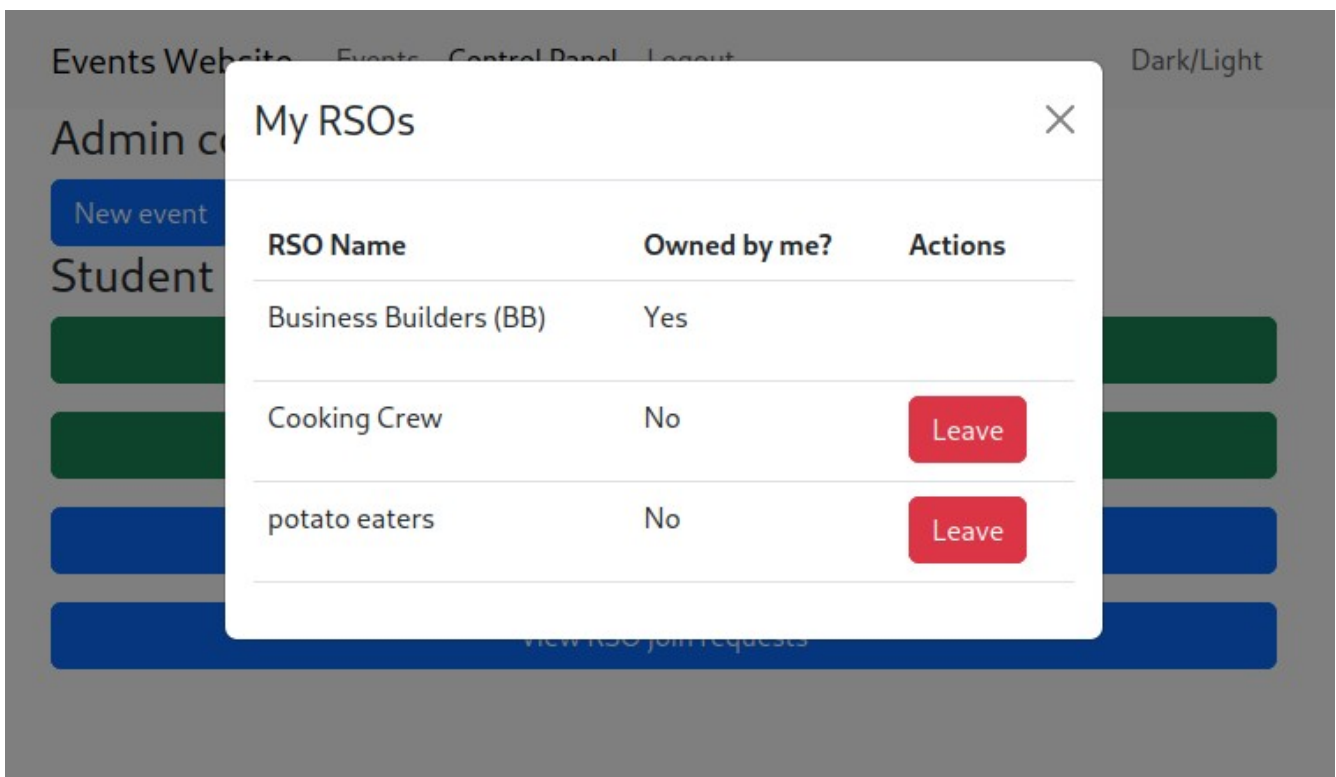


Figure 9: My RSOs modal



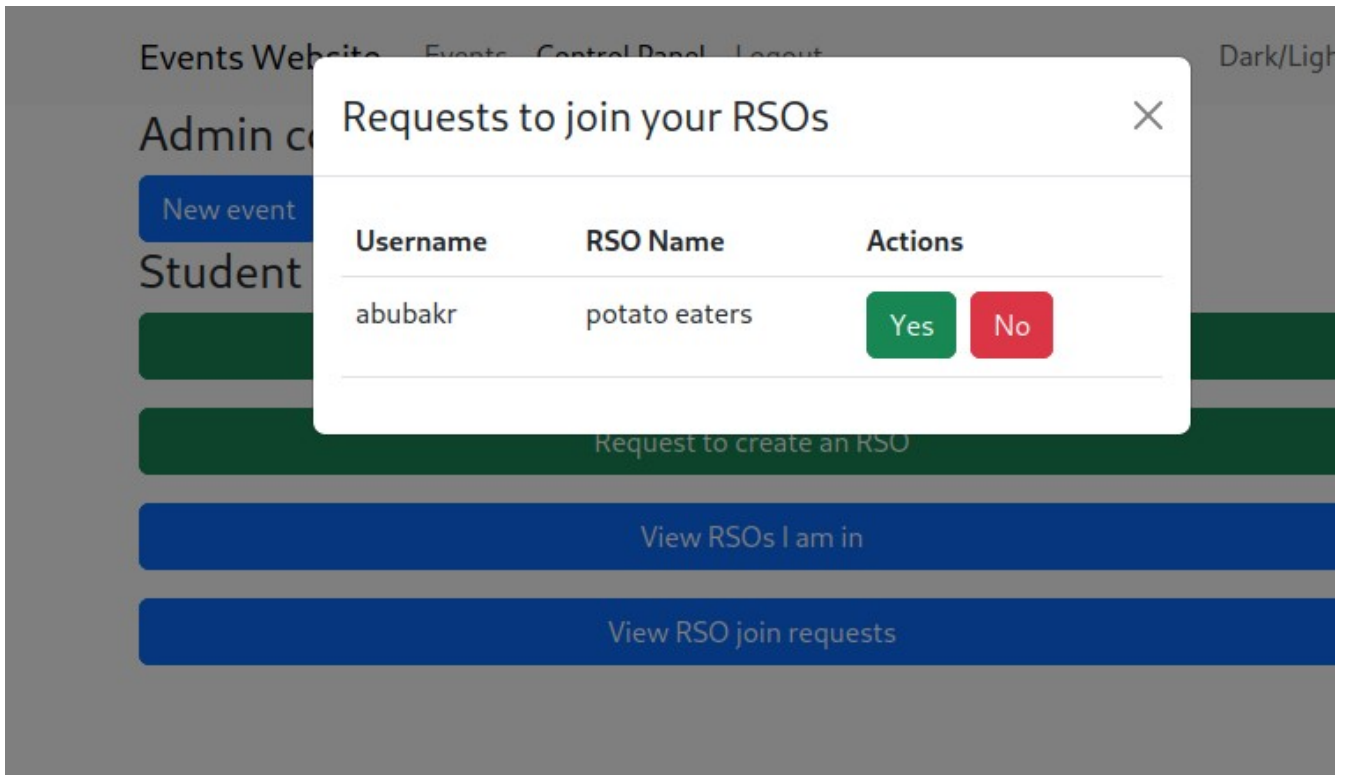


Figure 10: RSO join requests modal

New event

×

Event name

Enter event name

Event type

☒ Public

☐ Private

☐ RSO

Category

Technology

▼

Description

Coordinates

21.4225 39.8261


Radius (in meters)

15

⌵

+

-



Leaflet | © OpenStreetMap contributors

Start time

mm / dd / yyyy, --:-- --

📅

End time

mm / dd / yyyy, --:-- --

📅

Phone number

Enter phone number

Submit

Figure 11: Event creation form / modal

Events Website   Events   Control Panel   Logout   Dark/Light

# Superadmin

## New university

×

Name

Anytown University

Location

Anytown

Description

Number of Students

0

⬆ ⬇ ⬆

Submit

Figure 12: University creation form/modal

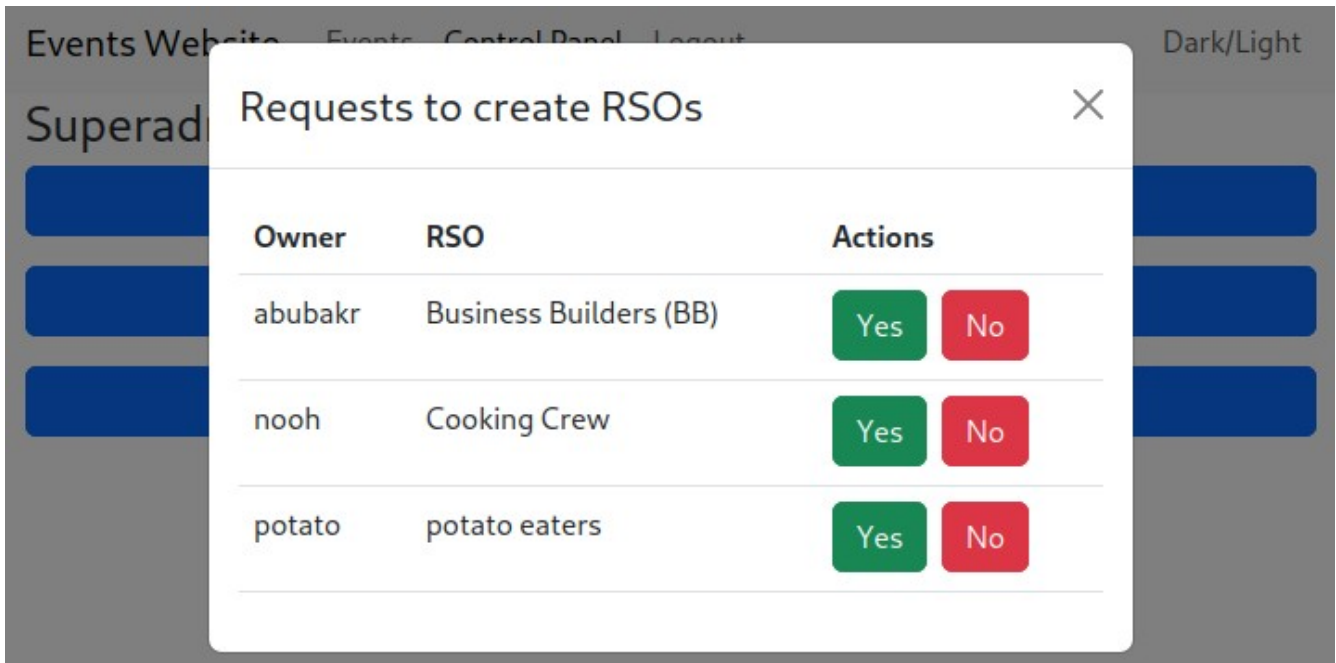


Figure 13: RSO requests modal

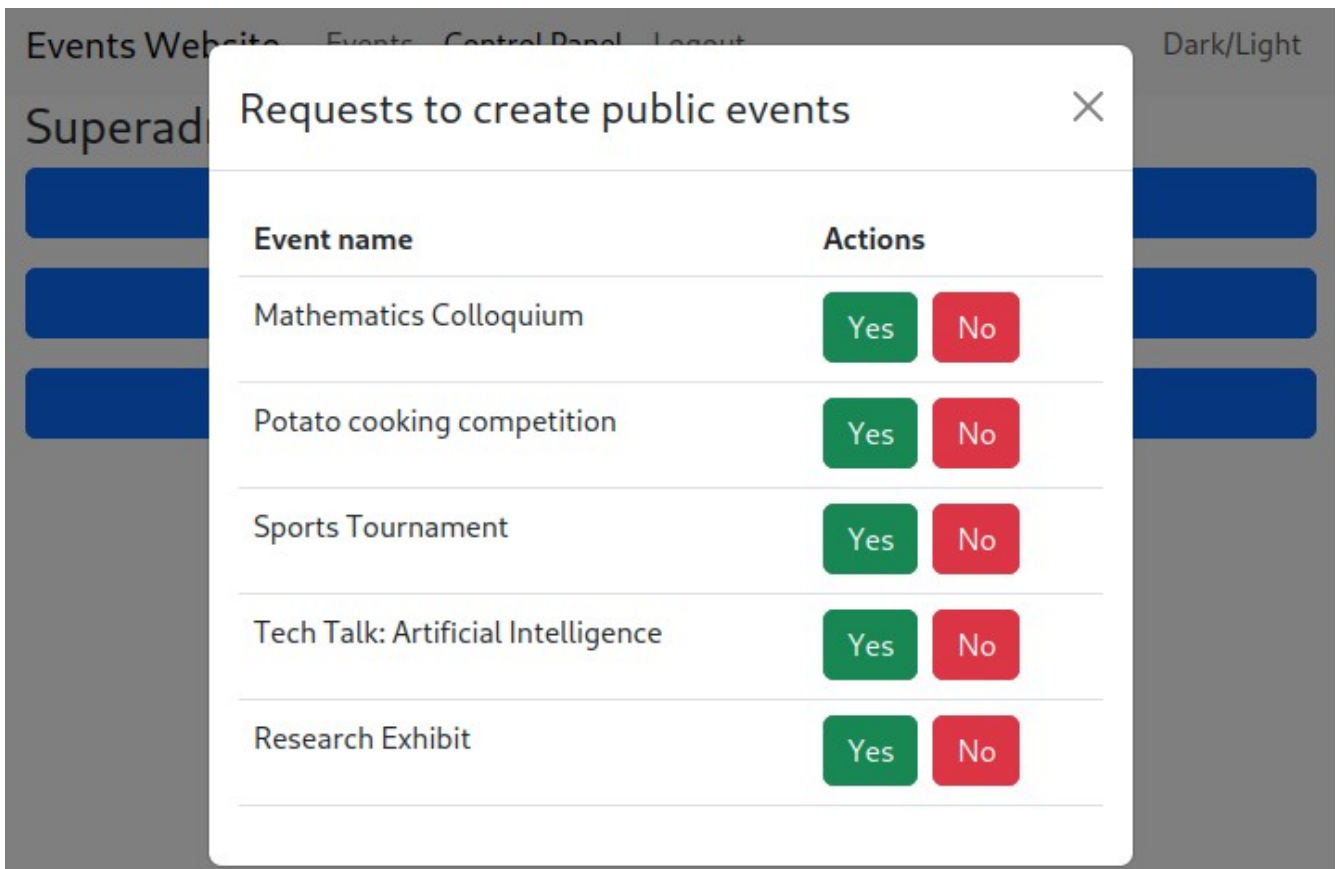


Figure 14: Public event requests modal



## More information

Event type	Public
Event category	Technology
Contact phone	555-1234
Contact email	abdullah@mail.com

## Write a comment:

Your comment

Submit

## Rating:



Number of ratings: 0

## Comments:


abubakr

Edit

Delete

This talk was great. 10/10.

Figure 15: Event details page



Leaflet | © OpenStreetMap contributors

Start time

04/10/2023, 03:04 AM

End time

04/20/2023, 11:43 PM

Phone number

324234

Submit

Event conflicts with existing event: lat,lng,start,end:  
21.4225,39.8261,Mon Apr 17 2023 03:04:00 GMT-0400 (Eastern  
Daylight Time),Mon Apr 17 2023 06:05:00 GMT-0400 (Eastern  
Daylight Time)

Figure 16: Error when creating an event that would conflict with an existing event

Events Control Panel Logout

### Edit your comment

Your comment

Enter a comment

Submit

Figure 17: Editing a comment

# ER Model

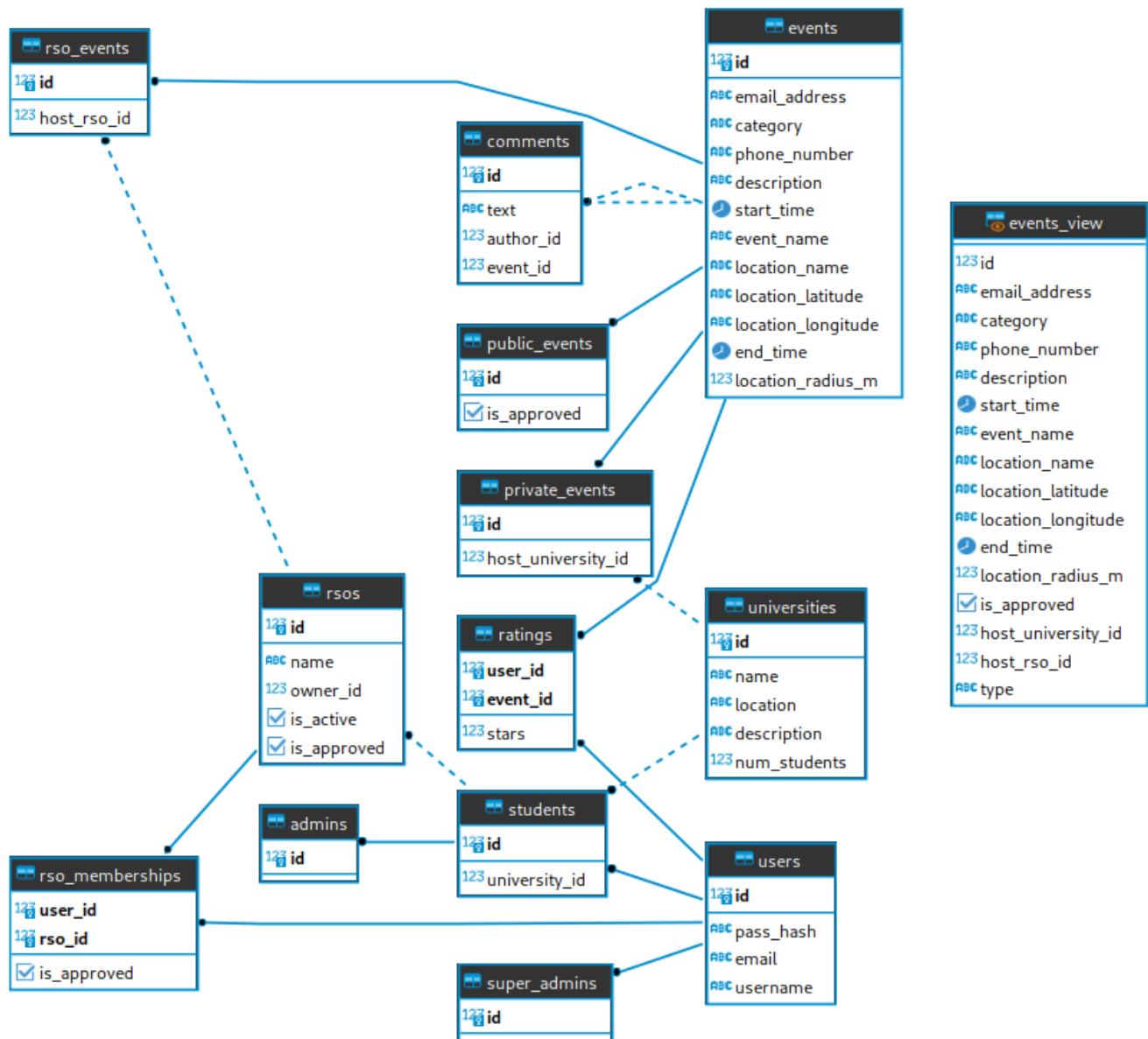


Figure 18: ER diagram

## Constraints captured in ER model:

- Three levels of users
- Three categories of events
- RSO membership many-to-many
- Each RSO has one owner
- User can leave many comments

- User can leave only one rating per event

### **Constraints captured outside ER model:**

- A user must be either student or admin or superadmin
- An event must be either public or private or RSO
- Events can't overlap in both time and location



# Relational Data Model

```
-- public.events definition

-- Drop table

-- DROP TABLE events;

CREATE TABLE events (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    email_address text NOT NULL,
    category text NOT NULL,
    phone_number text NOT NULL,
    description text NOT NULL,
    start_time timestamptz NOT NULL,
    event_name text NOT NULL,
    location_name text NOT NULL,
    location_latitude text NOT NULL,
    location_longitude text NOT NULL,
    end_time timestamptz NOT NULL,
    location_radius_m int8 NOT NULL DEFAULT '15'::bigint,
    CONSTRAINT events_pkey PRIMARY KEY (id)
);

-- public.universities definition

-- Drop table

-- DROP TABLE universities;

CREATE TABLE universities (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    "name" text NOT NULL,
    "location" text NOT NULL,
    description text NOT NULL,
    num_students int8 NOT NULL,
    CONSTRAINT universities_name_key UNIQUE (name),
    CONSTRAINT universities_pkey PRIMARY KEY (id)
);

-- public.users definition

-- Drop table

-- DROP TABLE users;

CREATE TABLE users (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    pass_hash text NOT NULL,
    email text NOT NULL,
    username text NOT NULL,
    CONSTRAINT "User_pkey" PRIMARY KEY (id),
    CONSTRAINT users_username_key UNIQUE (username)
);
```

```

-- public."comments" definition

-- Drop table

-- DROP TABLE "comments";

CREATE TABLE "comments" (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    "text" text NOT NULL,
    author_id int8 NOT NULL,
    event_id int8 NOT NULL,
    CONSTRAINT feedbacks_pkey PRIMARY KEY (id),
    CONSTRAINT comments_author_id_fkey FOREIGN KEY (author_id) REFERENCES
users(id) ON DELETE CASCADE,
    CONSTRAINT feedbacks_author_id_fkey FOREIGN KEY (event_id) REFERENCES
events(id) ON DELETE CASCADE,
    CONSTRAINT feedbacks_event_id_fkey FOREIGN KEY (event_id) REFERENCES
events(id) ON DELETE CASCADE
);

-- public.private_events definition

-- Drop table

-- DROP TABLE private_events;

CREATE TABLE private_events (
    id int8 NOT NULL,
    host_university_id int8 NOT NULL,
    CONSTRAINT private_events_pkey PRIMARY KEY (id),
    CONSTRAINT private_events_host_university_id_fkey FOREIGN KEY
(host_university_id) REFERENCES universities(id) ON DELETE CASCADE,
    CONSTRAINT private_events_id_fkey FOREIGN KEY (id) REFERENCES events(id) ON
DELETE CASCADE
);

-- public.public_events definition

-- Drop table

-- DROP TABLE public_events;

CREATE TABLE public_events (
    id int8 NOT NULL,
    is_approved bool NOT NULL DEFAULT false,
    CONSTRAINT public_events_pkey PRIMARY KEY (id),
    CONSTRAINT public_events_id_fkey FOREIGN KEY (id) REFERENCES events(id) ON
DELETE CASCADE
);

-- public.ratings definition

-- Drop table

-- DROP TABLE ratings;

```

```

CREATE TABLE ratings (
    stars int8 NOT NULL,
    user_id int8 NOT NULL,
    event_id int8 NOT NULL,
    CONSTRAINT ratings_pkey PRIMARY KEY (user_id, event_id),
    CONSTRAINT ratings_event_id_fkey FOREIGN KEY (event_id) REFERENCES events(id)
ON DELETE CASCADE,
    CONSTRAINT ratings_user_id_fkey FOREIGN KEY (user_id) REFERENCES users(id)
);

```

-- public.students definition

-- Drop table

-- DROP TABLE students;

```

CREATE TABLE students (
    id int8 NOT NULL,
    university_id int8 NOT NULL,
    CONSTRAINT students_pkey PRIMARY KEY (id),
    CONSTRAINT students_id_fkey FOREIGN KEY (id) REFERENCES users(id) ON DELETE
CASCADE,
    CONSTRAINT students_university_id_fkey FOREIGN KEY (university_id) REFERENCES
universities(id) ON DELETE CASCADE
);

```

-- public.super\_admins definition

-- Drop table

-- DROP TABLE super\_admins;

```

CREATE TABLE super_admins (
    id int8 NOT NULL,
    CONSTRAINT super_admins_pkey PRIMARY KEY (id),
    CONSTRAINT super_admins_id_fkey FOREIGN KEY (id) REFERENCES users(id) ON
DELETE CASCADE
);

```

-- public.admins definition

-- Drop table

-- DROP TABLE admins;

```

CREATE TABLE admins (
    id int8 NOT NULL,
    CONSTRAINT admins_pkey PRIMARY KEY (id),
    CONSTRAINT admins_id_fkey FOREIGN KEY (id) REFERENCES students(id) ON DELETE
CASCADE
);

```

-- public.rsos definition

-- Drop table

```

-- DROP TABLE rsos;

CREATE TABLE rsos (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    "name" text NOT NULL,
    owner_id int8 NOT NULL,
    is_active bool NOT NULL DEFAULT false,
    is_approved bool NOT NULL DEFAULT false,
    CONSTRAINT "RSOs_pkey" PRIMARY KEY (id),
    CONSTRAINT rsos_name_key UNIQUE (name),
    CONSTRAINT rsos_owner_id_fkey FOREIGN KEY (owner_id) REFERENCES students(id)
ON DELETE CASCADE
);

-- public.rso_events definition

-- Drop table

-- DROP TABLE rso_events;

CREATE TABLE rso_events (
    id int8 NOT NULL,
    host_rso_id int8 NOT NULL,
    CONSTRAINT "Rso_events_pkey" PRIMARY KEY (id),
    CONSTRAINT rso_events_host_rso_id_fkey FOREIGN KEY (host_rso_id) REFERENCES
rsos(id) ON DELETE CASCADE,
    CONSTRAINT rso_events_id_fkey FOREIGN KEY (id) REFERENCES events(id) ON
DELETE CASCADE
);

-- public.rso_memberships definition

-- Drop table

-- DROP TABLE rso_memberships;

CREATE TABLE rso_memberships (
    user_id int4 NOT NULL,
    rso_id int4 NOT NULL,
    is_approved bool NOT NULL DEFAULT false,
    CONSTRAINT rso_memberships_pkey PRIMARY KEY (user_id, rso_id),
    CONSTRAINT rso_memberships_rso_id_fkey FOREIGN KEY (rso_id) REFERENCES
rsos(id) ON DELETE CASCADE,
    CONSTRAINT rso_memberships_user_id_fkey FOREIGN KEY (user_id) REFERENCES
users(id) ON DELETE CASCADE
);

-- Functions (triggers may depend on these)

CREATE OR REPLACE FUNCTION public.check_rso_active()
    RETURNS trigger
    LANGUAGE plpgsql
    AS $function$
BEGIN
    IF (SELECT COUNT(*) FROM rso_memberships WHERE rso_id = NEW.rso_id AND
is_approved = true) > 4 THEN

```

```

        UPDATE rsos SET is_active = true WHERE id = NEW.rso_id;
    ELSE
        UPDATE rsos SET is_active = false WHERE id = NEW.rso_id;
    END IF;
    RETURN NEW;
END;
$function$
;

```

-- Table Triggers

```

create trigger check_rso_active_trigger after
insert
or
delete
or
update
on
    public.rso_memberships for each row execute function check_rso_active();

```

-- public.events\_view source

```

CREATE OR REPLACE VIEW events_view
AS SELECT events.id,
    events.email_address,
    events.category,
    events.phone_number,
    events.description,
    events.start_time,
    events.event_name,
    events.location_name,
    events.location_latitude,
    events.location_longitude,
    events.end_time,
    events.location_radius_m,
    public_events.is_approved,
    private_events.host_university_id,
    rso_events.host_rso_id,
    CASE
        WHEN private_events.host_university_id IS NOT NULL THEN 'private'::text
        WHEN rso_events.host_rso_id IS NOT NULL THEN 'rso'::text
        ELSE 'public'::text
    END AS type
FROM events
    LEFT JOIN public_events USING (id)
    LEFT JOIN private_events USING (id)
    LEFT JOIN rso_events USING (id)
WHERE public_events.is_approved = true OR public_events.is_approved IS NULL;

```

# Populating Tables with Sample Data

-- events

```
INSERT INTO events
(email_address,category,phone_number,description,start_time,event_name,location_name,location_latitude,location_longitude,end_time,location_radius_m) VALUES
('abdullah@mail.com','Academic','555-4321','Join us for a discussion on advanced topics in mathematics.','2023-04-25 10:00:00-04','Mathematics Colloquium','日本','35.6895','139.6917','2023-04-25 12:00:00-04',15),
('abdullah@mail.com','Technology','555-1234','Learn about the latest developments in AI from industry experts.','2023-05-01 09:00:00-04','Tech Talk: Artificial Intelligence','United States','40.7128','-74.006','2023-05-01 11:00:00-04',15),
('abdullah@mail.com','Sports','555-5678','Compete against other teams in a variety of sports, including basketball, soccer, and volleyball.','2023-05-12 05:00:00-04','Sports Tournament','Australia','-33.8688','151.2093','2023-05-12 14:00:00-04',15),
('abdullah@mail.com','Technology','555-3698','Learn the basics of coding and web development in this hands-on workshop.','2023-05-10 06:00:00-04','Coding Workshop','Mali','21.5074','-0.1278','2023-05-10 08:00:00-04',15),
('abdullah@mail.com','Talks','555-1357','Hear from a leading environmental activist about the latest efforts to combat climate change.','2023-04-29 15:00:00-04','Guest Speaker: Environmental Activist','United Kingdom','51.5074','-0.1278','2023-04-29 17:00:00-04',15),
('abdullah@mail.com','Other','555-2468','Join us for a fun evening of food and activities to raise money for a local charity.','2023-05-03 13:00:00-04','Charity Fundraiser','Argentina','-34.6037','-58.3816','2023-05-03 17:00:00-04',15),
('abubakr@knights.ucf.edu','Technology','342343434','Join us in raising funds for our university! Don't you want to help further education for our future students? ','2043-04-18 15:44:00-04','University Fundraiser 2043','','ليبيا',15,'04-17:00:00 18-04-2043','14.0625','29.5352'),
('abdullah@mail.com','Other','555-1357','View the latest papers from up-and-coming researchers in a variety of fields.','2023-04-30 06:00:00-04','Research Exhibit','Deutschland','52.52','13.405','2023-04-30 12:00:00-04',15),
('abubakr@knights.ucf.edu','Talks','3834343234','Ever wondered how to run a business? Join us to learn the strategies used by big businesses.','2023-04-04 03:04:00-04','Common Business Strategies','Canada','57.5158','-112.1484','2023-04-04 05:00:00-04',15),
('pop@tato.edu','Social','(839) 433-3938','Compete in cooking potatoes!','2023-04-05 10:00:00-04','Potato cooking competition','United States','39.6395','-100.5469','2023-04-05 20:00:00-04',15);
INSERT INTO events
(email_address,category,phone_number,description,start_time,event_name,location_name,location_latitude,location_longitude,end_time,location_radius_m) VALUES
('pop@tato.edu','Academic','(324) 434-2394','Join us in raising money!','2023-04-05 20:00:00-04','University Fundraiser 2043','Polska','52.0525','15.8203','2023-04-05 21:00:00-04',15),
('pop@tato.edu','Social','(324) 434-2394','Come to the RS0's opening!','2023-04-05 06:40:00-04','RS0 Opening Ceremony','','السعودية',15,'04-08:00:00 05-04-2023','42.8906'),
('abdullah@mail.com','Sports','555-5678','Watch the university soccer team as they take on their rivals.','2023-04-22 14:00:00-04','University Soccer Match','15','04-16:00:00 22-04-2023','39.1925','21.4858','السعودية'),
('pop@tato.edu','Talks','435454345','Lecture about the virtues of Ramadan','2023-04-17 03:04:00-04','Ramadan Motivation','','السعودية',15,'04-06:05:00 17-04-2023','39.8261');
```

-- universities

```
INSERT INTO universities ("name","location",description,num_students) VALUES
    ('University of Central Florida','Orlando','A university located in Orlando,
    FL. Known for its large student population.',66000),
    ('Oregon State University','Corvallis, OR','Oregon State University is a
    public land-grant, research university in Corvallis, Oregon. OSU offers more than
    200 undergraduate-degree programs along with a variety of graduate and doctoral
    degrees.',35000),
    ('University of Maine','Orono, ME','The University of Maine is a public
    land-grant research university in Orono, Maine. It was established in 1865 as the
    land-grant college of Maine and is the flagship university of the University of
    Maine System. It is classified by Carnegie as among "R1: Doctoral Universities -
    Very high research activity".',11000),
    ('Waffle University','Waffletown','An imaginary university in Waffletown. 39
    million students.',39000000);
```

-- users

```
INSERT INTO users (pass_hash,email,username) VALUES

('$2a$10$9S00knE5bpAxLWqll1q11.XC9LbBSEBvnkziNuYcdL0eesIwaCOE6','abdullah@mail.com',
'abdullah'),

('$2a$10$DJlqPfiPu4dQkymJnDnBmecucNCMG5SNPnobZQdY5geCALlwYg20','abdurrahman@knight
s.ucf.edu','abdurrahman'),

('$2a$10$gF50UZe.vj2bf2z3vzTSG.559UftDBvc6hrMu/BuNyDvm8AlNmETK','abubakr@knights.uc
f.edu','abubakr'),

('$2a$10$Ps8uvDdORiFRoOz7qp6N5u/eNTCoxLzbNl/4NuQ295ohco2QpUnVa','oregonman@osu.edu',
'oregonman'),

('$2a$10$egfIe/05XAmc07uxS4Poe.9p2lLkGkTZecmrhx.NKatKCDw1FrR8W','musa@uni.edu','abu
musa'),

('$2a$10$INmgN55wRDLJGEDpaizqNesMpzLRCGRQESfSIZvDoT19eQQ65CMM.','salman@osu.edu','s
alman'),

('$2a$10$0NbSgAY/NeU08nHQxGjo/.7usbDiwh1lFLAXXP50ozIpMLcB4v6Q.','nooh@osu.edu','noo
h'),

('$2a$10$720YLAWEMlbXBMrSMXLm6uUarjxJG6XsQIckv9FrAyFk126opTh6e','pop@tato.edu','pot
ato');
```

-- comments

```
INSERT INTO "comments" ("text",author_id,event_id) VALUES
    ('I liked this hypothetical event!',4,38),
    ('What if 1+1 was not 2?',4,38),
    ('This talk was great. 10/10.',6,34);
```

-- private\_events

```
INSERT INTO private_events (id,host_university_id) VALUES
    (45,3),
    (40,3),
    (41,3),
    (42,3),
```

```

        (63,3);

--public_events

INSERT INTO public_events (id,is_approved) VALUES
    (36,true),
    (65,true),
    (38,true),
    (62,true),
    (39,true),
    (34,true),
    (35,true);

--ratings

INSERT INTO ratings (stars,user_id,event_id) VALUES
    (5,4,38),
    (5,6,38),
    (5,6,46);

--students

INSERT INTO students (id,university_id) VALUES
    (5,2),
    (6,3),
    (11,3),
    (12,3),
    (13,3),
    (14,3),
    (22,3);

--super_admins

INSERT INTO super_admins (id) VALUES
    (4);

--admins

INSERT INTO admins (id) VALUES
    (6),
    (22),
    (14);

--rsos

INSERT INTO rsos ("name",owner_id,is_active,is_approved) VALUES
    ('Cooking Crew',14,false,true),
    ('potato eaters',22,false,true),
    ('Business Builders (BB)',6,true,true);

--rso_events

INSERT INTO rso_events (id,host_rso_id) VALUES
    (46,16),
    (64,25);

--rso_memberships

INSERT INTO rso_memberships (user_id,rso_id,is_approved) VALUES

```



```
(12,16,true),
(11,16,true),
(13,16,true),
(14,16,true),
(14,17,true),
(6,16,true),
(22,25,true),
(6,17,false),
(6,25,true);
```

## Note

Users' passwords are the same as their usernames. For example, abdullah's password is abdullah.

## SQL examples and results

```
1  INSERT INTO rsos (name, owner_id)
2  VALUES ('Mechanic School', 14);
3
4  select * from rsos
```

id	name	owner_id	is_active	is_approved
17	"Cooking Crew"	14	false	true
25	"potato eaters"	22	false	true
16	"Business Builders (BB)"	6	true	true
28	"Mechanic School"	14	false	false


Figure 19: Inserting an RSO

```
1  insert into rso_memberships (user_id, rso_id)
2  values (22, 17);
3
4  select * from rso_memberships
5  where user_id=22 and rso_id=17;
```

user_id	rso_id	is_approved
22	17	false

Figure 20: Inserting a new student to an existing RSO

```
1 INSERT INTO events (email_address, category, phone_number,
2 | description, start_time, event_name,
3 | location_latitude, location_longitude, end_time,
4 | location_name, location_radius_m)
5 VALUES ('a@a.co', 'Social', '9232332', 'Some description',
6 | '2023-05-10 18:30:00', 'A name', '37.7749', '-122.4194',
7 | '2023-05-10 21:30:00', 'HS1 932', '1000');
8
9 select * from events where event_name = 'A name';
```

Results ▾  ▾ RUN CTRL ↵

id	email_address	category	phone_number	description	start_t
69	"a@a.co"	"Social"	"9232332"	"Some descriptio	"2023-05

Figure 21: Inserting a new event (I omitted some columns from result because of space)

1 -- INSERT INTO comments (event\_id, author\_id, text)  
2 -- VALUES ('63', '14', 'I am in the report!');  
3 |  
4 UPDATE comments SET text = 'New text!' WHERE id = 35;  
5  
6 SELECT \* from comments WHERE id='35';

Results    

RUN CTRL 

id	text	author_id	event_id
35	"New text!"	14	63

Figure 22: Updating a comment

```

1 SELECT id, host_university_id, host_rso_id, event_name, description, location_name,
   start_time, end_time, type
2 FROM events_view

```

Results ▾



RUN CTRL ↵

id	host_univ	host_rso_	event_name	descripti	location_	start_tim	end_time	type
45	3	null	"Universit	"Join us i	"ليبيا"	"2043-04-1	"2043-04-1	"private"
40	3	null	"Coding Wo	"Learn the	"Mali"	"2023-05-1	"2023-05-1	"private"
41	3	null	"Guest Spe	"Hear from	"United Ki	"2023-04-2	"2023-04-2	"private"
42	3	null	"Charity F	"Join us f	"Argentina	"2023-05-0	"2023-05-0	"private"
63	3	null	"Universit	"Join us i	"Polska"	"2023-04-0	"2023-04-0	"private"
39	null	null	"Sports To	"Compete a	"Australia	"2023-05-1	"2023-05-1	"public"
34	null	null	"Tech Talk	"Learn abo	"United St	"2023-05-0	"2023-05-0	"public"
64	null	25	"Rso Openi	"Come to t	"السعودية"	"2023-04-0	"2023-04-0	"rso"
46	null	16	"Common Bu	"Ever wond	"Canada"	"2023-04-0	"2023-04-0	"rso"
65	null	null	"Ramadan M	"Lecture a	"السعودية"	"2023-04-1	"2023-04-1	"public"
38	null	null	"Mathemati	"Join us f	"日本"	"2023-04-2	"2023-04-2	"public"
62	null	null	"Potato co	"Compete i	"United St	"2023-04-0	"2023-04-0	"public"
36	null	null	"Universit	"Watch the	"السعودية"	"2023-04-2	"2023-04-2	"public"
35	null	null	"Research	"View the	"Deutschla	"2023-04-3	"2023-04-3	"public"

Figure 23: Displaying events for superadmin “abdullah”

```

1  SELECT id, host_university_id, host_rso_id, event_name, description, location_name,
2  start_time, end_time, type
3  FROM events_view
4  WHERE (
5      type = 'public' OR (
6          type = 'private' AND
7          host_university_id = (
8              SELECT university_id
9              FROM students
10             WHERE students.id = 5
11         ) OR (
12             type = 'rso' AND
13             host_rso_id IN (
14                 SELECT rso_id
15                 FROM rso_memberships
16                 WHERE rso_memberships.user_id = 5
17             )
18         )
19 )

```

Results ▾



RUN CTRL ↵

id	host_univ	host_rso	event_name	description	location_name	start_time	end_time	type
39	null	null	"Sports To	"Compete a	"Australia	"2023-05-1	"2023-05-1	"public"
34	null	null	"Tech Talk	"Learn abo	"United St	"2023-05-0	"2023-05-0	"public"
65	null	null	"Ramadan M	"Lecture a	"السعودية"	"2023-04-1	"2023-04-1	"public"
38	null	null	"Mathemati	"Join us f	"日本"	"2023-04-2	"2023-04-2	"public"
62	null	null	"Potato co	"Compete i	"United St	"2023-04-0	"2023-04-0	"public"
36	null	null	"Universit	"Watch the	"السعودية"	"2023-04-2	"2023-04-2	"public"
35	null	null	"Research	"View the	"Deutschla	"2023-04-3	"2023-04-3	"public"

Figure 24: Displaying events for user “abdurrahman”

```

1  SELECT id, host_university_id, host_rso_id, event_name, description, location_name,
2  start_time, end_time, type
3  FROM events_view
4  WHERE (
5      type = 'public' OR (
6          type = 'private' AND
7          host_university_id = (
8              SELECT university_id
9              FROM students
10             WHERE students.id = 22
11         ) OR (
12             type = 'rso' AND
13             host_rso_id IN (
14                 SELECT rso_id
15                 FROM rso_memberships
16                 WHERE rso_memberships.user_id = 22
17             )
18         )
19 )

```

Results ▾




RUN CTRL ↵

id	host_univ	host_rso	event_name	description	location_name	start_time	end_time	type
45	3	null	"Universit	"Join us i	"ليبيا"	"2043-04-1	"2043-04-1	"private"
40	3	null	"Coding Wo	"Learn the	"Mali"	"2023-05-1	"2023-05-1	"private"
41	3	null	"Guest Spe	"Hear from	"United Ki	"2023-04-2	"2023-04-2	"private"
42	3	null	"Charity F	"Join us f	"Argentina	"2023-05-0	"2023-05-0	"private"
63	3	null	"Universit	"Join us i	"Polska"	"2023-04-0	"2023-04-0	"private"
39	null	null	"Sports To	"Compete a	"Australia	"2023-05-1	"2023-05-1	"public"
34	null	null	"Tech Talk	"Learn abo	"United St	"2023-05-0	"2023-05-0	"public"
64	null	25	"RSO Openi	"Come to t	"السعودية"	"2023-04-0	"2023-04-0	"rso"
65	null	null	"Ramadan M	"Lecture a	"السعودية"	"2023-04-1	"2023-04-1	"public"
38	null	null	"Mathemati	"Join us f	"日本"	"2023-04-2	"2023-04-2	"public"
62	null	null	"Potato co	"Compete i	"United St	"2023-04-0	"2023-04-0	"public"
36	null	null	"Universit	"Watch the	"السعودية"	"2023-04-2	"2023-04-2	"public"
35	null	null	"Research	"View the	"Deutschla	"2023-04-3	"2023-04-3	"public"

Figure 25: Displaying events for user “potato”

## Constraint Enforcement



Start time

04/10/2023, 03:04 AM

End time

04/20/2023, 11:43 PM

Phone number

324234

Submit

Event conflicts with existing event: lat,lng,start,end:  
21.4225,39.8261,Mon Apr 17 2023 03:04:00 GMT-0400 (Eastern Daylight Time),Mon Apr 17 2023 06:05:00 GMT-0400 (Eastern Daylight Time)

Figure 26: Making new event at same location and overlapping times with existing event

From the frontend, it isn't possible to make events for an RSO which one is not the admin of. Instead, I will show the error message from the backend.

```
→ ~ http post http://127.0.0.1:3000/events "Authorization:Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6InBvdGF0byIsImkIjoimjIiLCJpYXQiOiJlE20DE4MzE4MjJ9.knq75RlEqEgune_MK70QJVRcQz4WUV8vi_48zL7p9qM" name="test" type="rso" category="talks" description="test" latitude="38.002" longitude="29.23" radius="100" startTime="December 23, 2022" endTime="January 2, 2040" phoneNumber="293932" rso="Business Builders (BB)"
HTTP/1.1 403 Forbidden
Connection: keep-alive
Date: Tue, 18 Apr 2023 15:42:32 GMT
Keep-Alive: timeout=72
access-control-allow-origin: *
content-length: 47
content-type: application/json; charset=utf-8
vary: Origin

{
  "message": "You are not the owner of this RSO"
}
```

Figure 27: Making event for RSO other than one's own

User joins an RSO with 4 members

First, rso with id 16 is inactive

<input type="checkbox"/>	 id int8 ▾	name text ▾	 owner_id int8 ▾	is_active bool ▾	is_approv
<input type="checkbox"/>	25	potato eaters	22 	FALSE	TRUE
<input type="checkbox"/>	17	Cooking Crew	14 	FALSE	TRUE
<input type="checkbox"/>	16	Business Builde	6 	FALSE	TRUE

New membership

<input type="checkbox"/>	6	16	TRUE		
--------------------------	---	----	------	--	--

The rso changed to active

<input type="checkbox"/>	16	Business Builde	6 	TRUE	TRUE
--------------------------	----	-----------------	---	------	------

User leaves an RSO with 5 members

I couldn't get the is\_active field to update in this case.

# Advanced Features

Users need to be logged in to access most API routes.

```
→ backend git:(master) http get http://127.0.0.1:3000/events
HTTP/1.1 401 Unauthorized
Connection: keep-alive
Date: Tue, 18 Apr 2023 14:20:49 GMT
Keep-Alive: timeout=72
access-control-allow-origin: *
content-length: 143
content-type: application/json; charset=utf-8
vary: Origin

{
  "code": "FST_JWT_NO_AUTHORIZATION_IN_HEADER",
  "error": "Unauthorized",
  "message": "No Authorization was found in request.headers",
  "statusCode": 401
}
```



## Conclusion

Overall, I was able to implement all of the required features. Some features/functionalities that might be nice in a real life application are: event feed from university websites, social network integration, “forgot password” functionality, and ability to RSVP for an event.

My database performed really well. Queries were completed quickly, and I didn’t notice any significant delays. Perhaps there wasn’t enough data for performance to be an issue.

From this project I learned about Supabase and what it offers. I learned how to use triggers, functions, and views in SQL. I also learned more about the tools I used to develop this full-stack site.

I didn’t really have any problems during the project. After the initial setup, it was going smoothly. In retrospect, I think would be better to work in a team. I also think it would be better to use a framework which comes with features like client routing and API routes so that I don’t have to implement them myself.