

Event Nexus

COP4710- 23 - Spring 2023

Chrystian Orren



Table of Contents

Project Description.....	Page 3
GUI.....	Page 4
ER-Model.....	Page 9
SQL-Model.....	Page 10
SQL-Examples.....	Page 13
Constraint Enforcement.....	Page 18
Conclusion.....	Page

Project Description

This is a web based application created by Chrystian Orren for COP4710 with Professor Vu in Spring 2023.

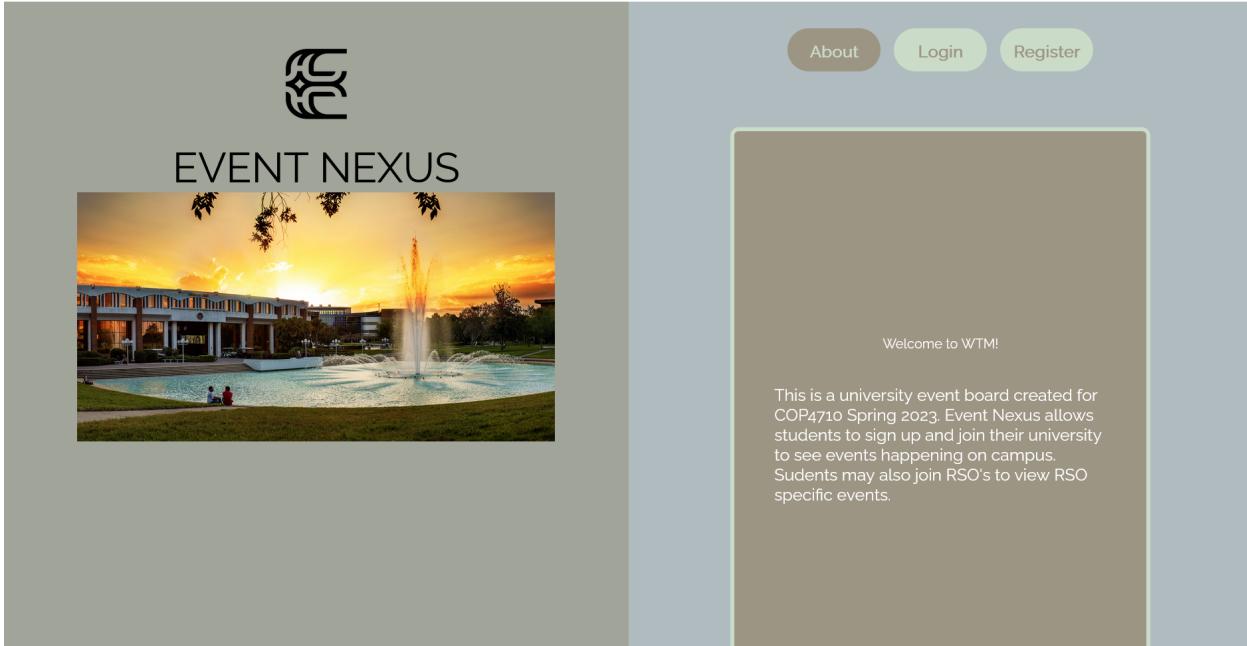
The purpose of this project is to create a web-based application that solves the problem of keeping track of events around university campuses. With this application there are three levels of users: student, admin, and superadmin.

Students are able to login and join any RSO the university offers, they are also able to view public events, private events for the university, and RSO events from which they are a part of. From my interpretation of the project description Admins are students who have created an RSO. Admins are able to create events for their RSO and will only display it to the members of the RSO. Superadmins create the profile for a university and hold the responsibility of approving public events, approving the creation of RSO's, and creating private events. Users can also comment and rate events shown on the main page.

According to the project description document given to us students are not able to create events, however this contradicts public events needing approval since the only other user creating a public event would be the superadmin. So I allowed students to submit a request for a public event and it will only be seen after the super admin approves it.

GUI

Landing Page

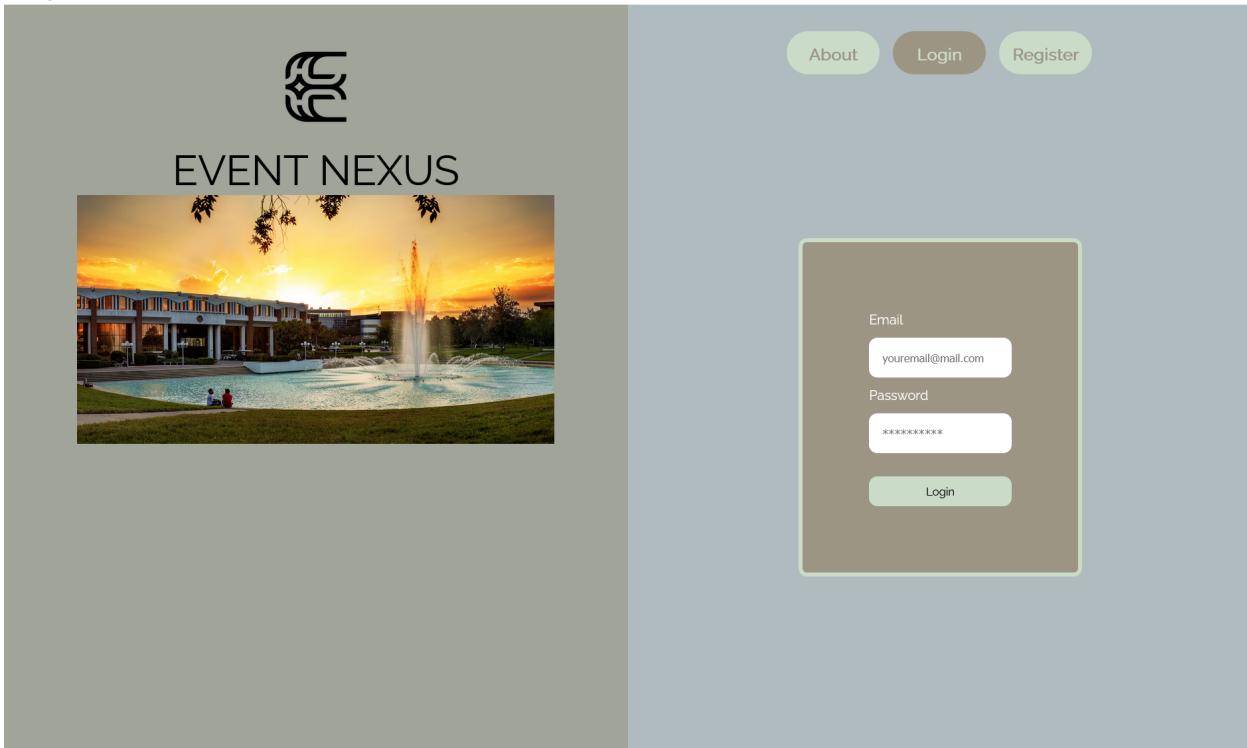


The screenshot shows the landing page of the Event Nexus application. On the left, there is a large image of a university campus at sunset, featuring a fountain and modern buildings. On the right, there is a sidebar with three buttons: "About" (brown), "Login" (green), and "Register" (light green). Below the buttons is a large brown rectangular area containing text.

Welcome to WTM!

This is a university event board created for COP4710 Spring 2023. Event Nexus allows students to sign up and join their university to see events happening on campus. Students may also join RSO's to view RSO specific events.

Login



The screenshot shows the login page of the Event Nexus application. It features the same layout as the landing page, with a large campus image on the left and a sidebar with "About" (green), "Login" (brown), and "Register" (light green) buttons on the right. The main area contains a brown rectangular form with fields for Email and Password, and a "Login" button.

Email
youremail@mail.com

Password

Login

Register

EVENT NEXUS

Email: youremail@mail.com

Password: *****

Confirm Password: *****

User Level: Student

Create Account

Main Page

Find an RSO

Private Public RSO

Cycle Into Spring | Recreation & Exercise

Ring in the spring season by taking part in a month-long challenge to complete at least eight cycling classes at the Recreation and Wellness Center (RWC) in April. Participants can sign up in person at one of the RWC Cycle45 classes prior to April 1, 2023.

Comments

Rate This Event

Event Rating: 4.00

4000 Central Florida Blvd, Orlando, FL 32816, USA

Map data ©2023

Free Coffee and Study Space | Social Event

From 9:00am to 3:30 pm Monday-Friday, the Wesley Foundation space will be open as a study space for students. There is also free coffee (freshly brewed and a Keurig)!

Comments

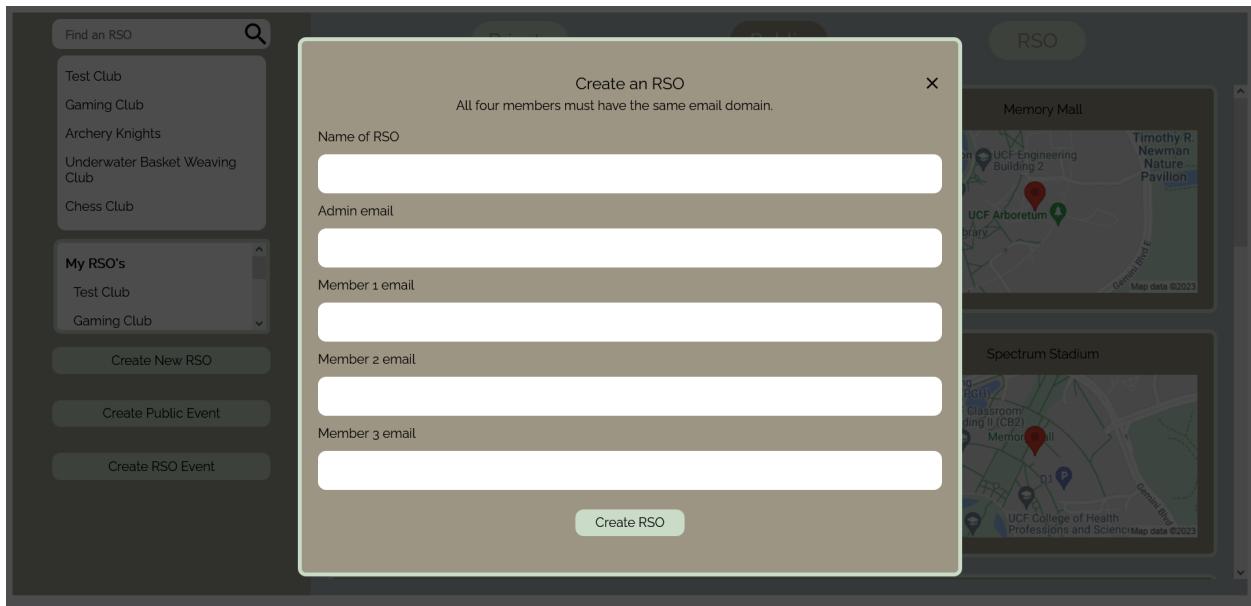
Rate This Event

Event Rating: 5.00

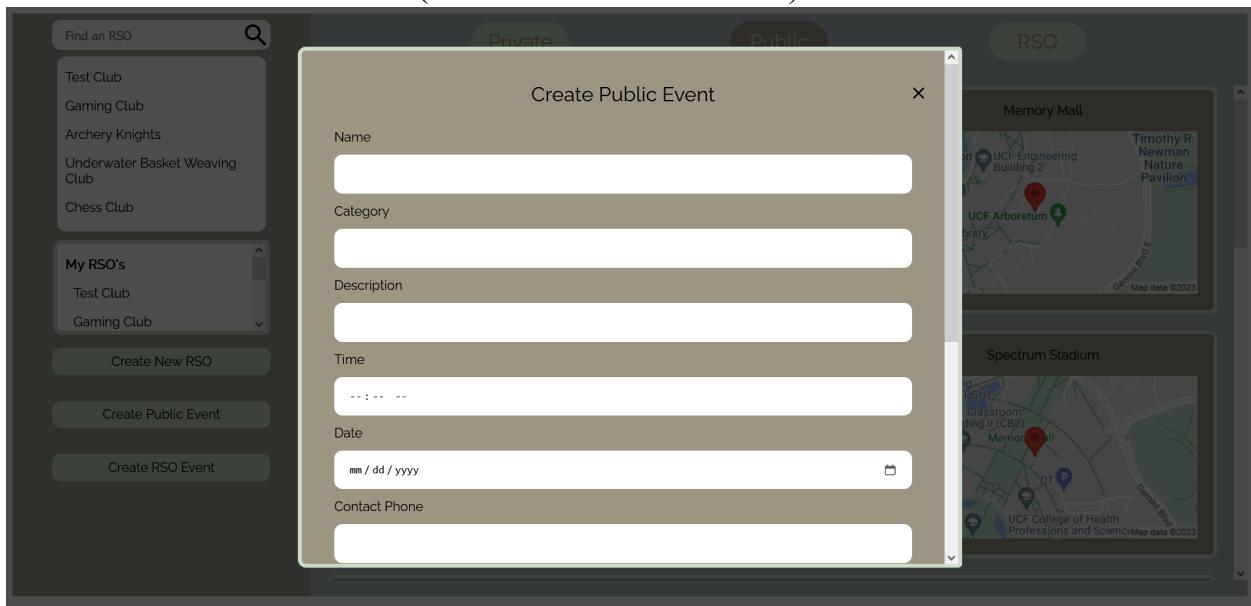
1395 Campus View Ct, Oviedo, FL 32765, USA

Map data ©2023

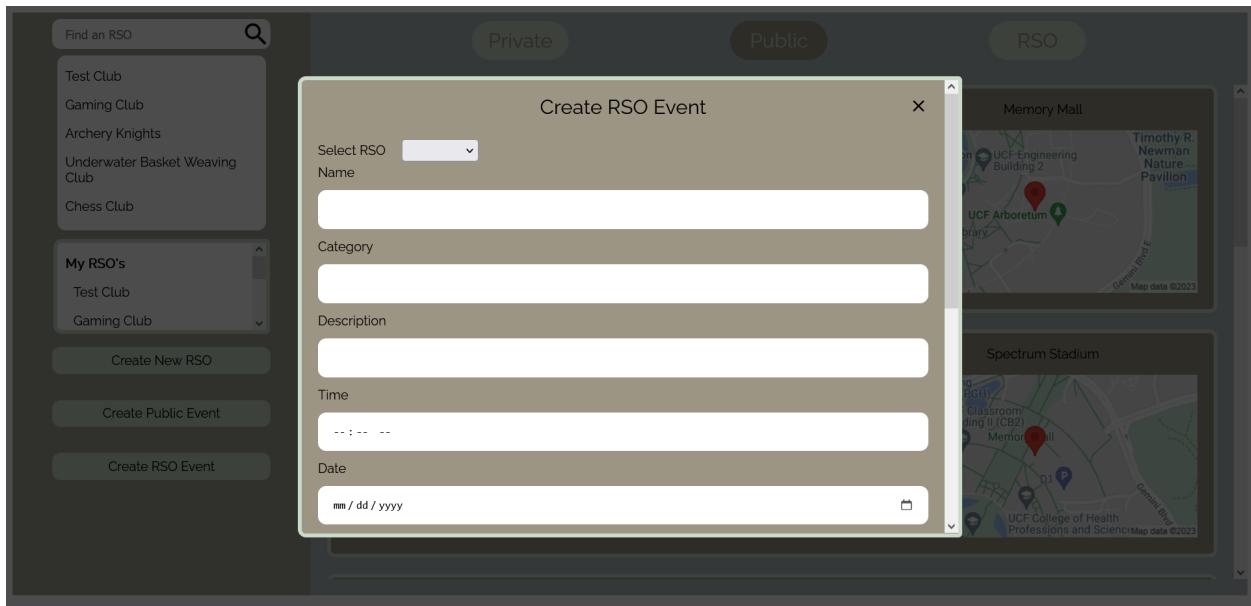
Create New RSO Modal



Create Public Event Modal (Same for Private event)



Create RSO Event



Superadmin Main Page

The screenshot shows the Superadmin Main Page with two main sections: "Events awaiting approval" and "RSO's awaiting approval".

- Events awaiting approval:** Displays an event card with details: Name: Pick up Soccer, Category: Social / Sport, Date: 2023-04-20, Time: 18:00:00, Location: 12491 Gemini Blvd N, Orlando, FL 32816, USA. It includes "Reject" and "Approve" buttons.
- RSO's awaiting approval:** Displays an event card with details: Name: Intramural Indoor Soccer. It includes "Reject" and "Approve" buttons.
- Buttons:** A green "Create Private Event" button is located between the two sections.

Event Comments

The screenshot shows a user interface for managing RSOs (Student Organizations). On the left, there's a sidebar with a search bar labeled "Find an RSO" and a magnifying glass icon. Below it is a list of "Test Club" members: Test Club, Gaming Club, Archery Knights, Underwater Basket Weaving Club, and Chess Club. A section titled "My RSO's" lists "Archery Knights" and "Underwater Basket Weaving". Buttons for "Create New RSO" and "Create Public Event" are also present.

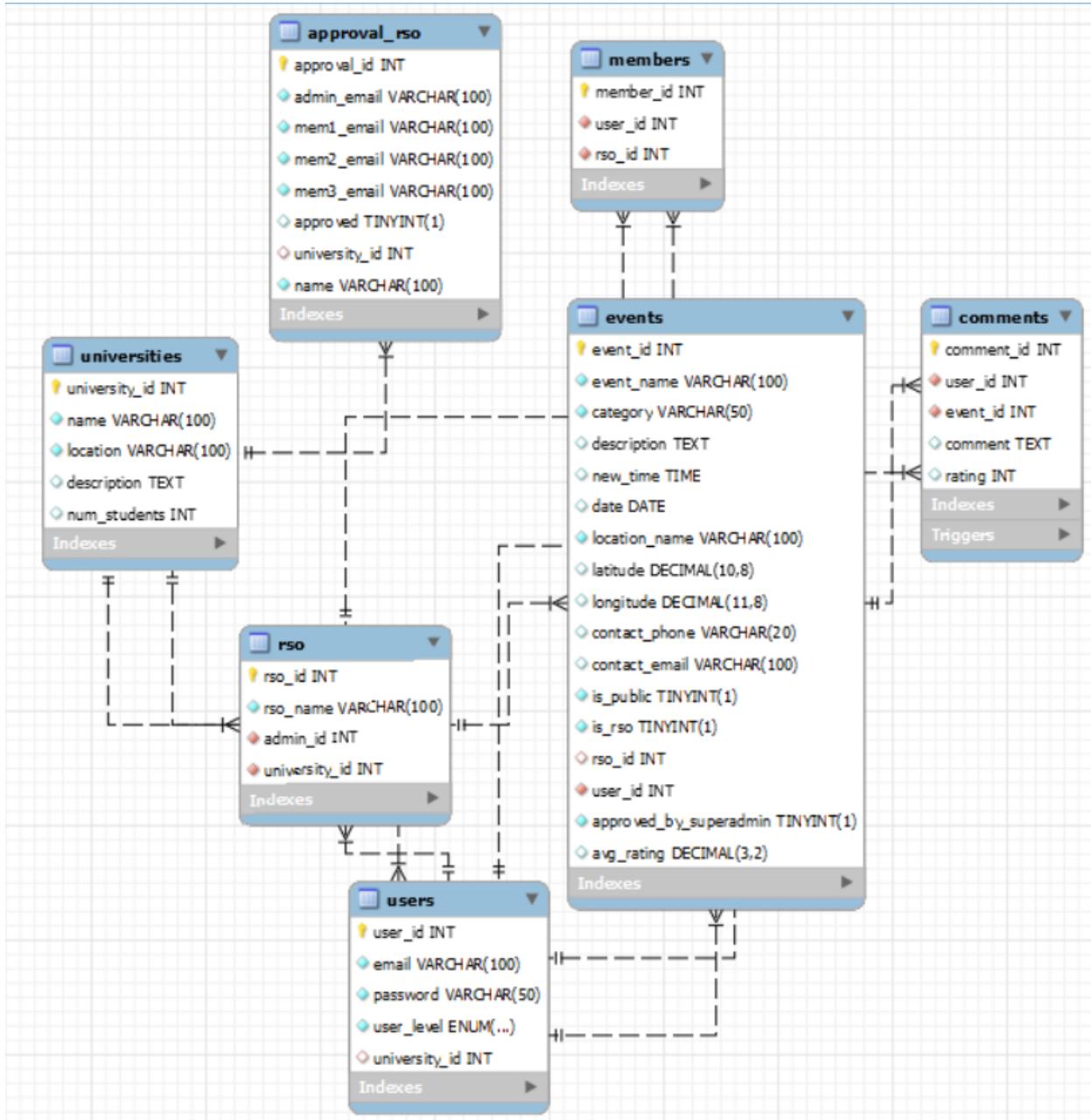
The main area features three tabs at the top: "Private", "Public", and "RSO". A modal window is open, centered over the "Public" tab, with a close button "X" in the top right corner. The modal has a header "Write a comment" with a right-pointing arrow. Inside, there are three text input fields:

- "How long is the route?"
- "Anyone got an extra bike I can borrow?"
- "I'll bring the Gatorade!!"

Below these fields is a section titled "My comments" containing the same three messages, each with edit and delete icons. At the bottom of the modal is a large, empty text input field with a placeholder "Type your comment here..." and a "Send" button.

On the right side of the screen, two maps are displayed. The top map is titled "Memory Mall" and shows locations like UCF Engineering Building 2, Timothy R. Newman Nature Pavilion, and UCF Arboretum. The bottom map is titled "Spectrum Stadium" and shows the stadium itself along with other campus buildings like Classroom Building II (CB2) and the College of Health Professions and Sciences. Both maps include a copyright notice "Map data ©2023".

ER-Model



SQL-Model

```
CREATE DATABASE event_app;
```

```
USE event_app;
```

```
DELIMITER //
CREATE TRIGGER update_avg_rating AFTER INSERT ON comments
FOR EACH ROW
BEGIN
    UPDATE events
    SET avg_rating = (SELECT AVG(rating) FROM comments WHERE event_id =
NEW.event_id AND comment IS NULL)
    WHERE event_id = NEW.event_id;
END //
DELIMITER ;
```

```
ALTER TABLE events
ADD CONSTRAINT unique_event_location_time_date
UNIQUE (location_name, new_time, date);
```

```
CREATE TABLE approval_rso (
    approval_id INT NOT NULL AUTO_INCREMENT,
    admin_email VARCHAR(100) NOT NULL,
    mem1_email VARCHAR(100) NOT NULL,
    mem2_email VARCHAR(100) NOT NULL,
    mem3_email VARCHAR(100) NOT NULL,
    approved BOOLEAN DEFAULT 0,
    university_id INT,
    name VARCHAR(100) NOT NULL,
    PRIMARY KEY (approval_id),
    FOREIGN KEY (university_id) REFERENCES universities(university_id)
);
```

```
CREATE TABLE universities (
    university_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    location VARCHAR(100) NOT NULL,
    description TEXT,
```

```

num_students INT,
PRIMARY KEY (university_id)
);

```

```

CREATE TABLE users (
    user_id INT NOT NULL AUTO_INCREMENT,
    email VARCHAR(100) NOT NULL,
    password VARCHAR(50) NOT NULL,
    user_level ENUM('superadmin', 'admin', 'student') NOT NULL,
    university_id INT,
    PRIMARY KEY (user_id),
    FOREIGN KEY (university_id) REFERENCES universities(university_id)
);

```

```

CREATE TABLE events (
    event_id INT NOT NULL AUTO_INCREMENT,
    event_name VARCHAR(100) NOT NULL,
    category VARCHAR(50) NOT NULL,
    description TEXT,
    new_time TIME,
    date DATE,
    location_name VARCHAR(100) NOT NULL,
    latitude DECIMAL(10, 8),
    longitude DECIMAL(11, 8),
    contact_phone VARCHAR(20),
    contact_email VARCHAR(100),
    is_public BOOLEAN NOT NULL,
    is_rso BOOLEAN NOT NULL,
    rso_id INT,
    user_id INT NOT NULL,
    avg_rating DECIMAL(3,2) DEFAULT NULL,
    approved_by_superadmin BOOLEAN NOT NULL,
    PRIMARY KEY (event_id),
    FOREIGN KEY (rso_id) REFERENCES rso(rso_id),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

```

```

CREATE TABLE rso (
    rso_id INT NOT NULL AUTO_INCREMENT,
    rso_name VARCHAR(100) NOT NULL,

```

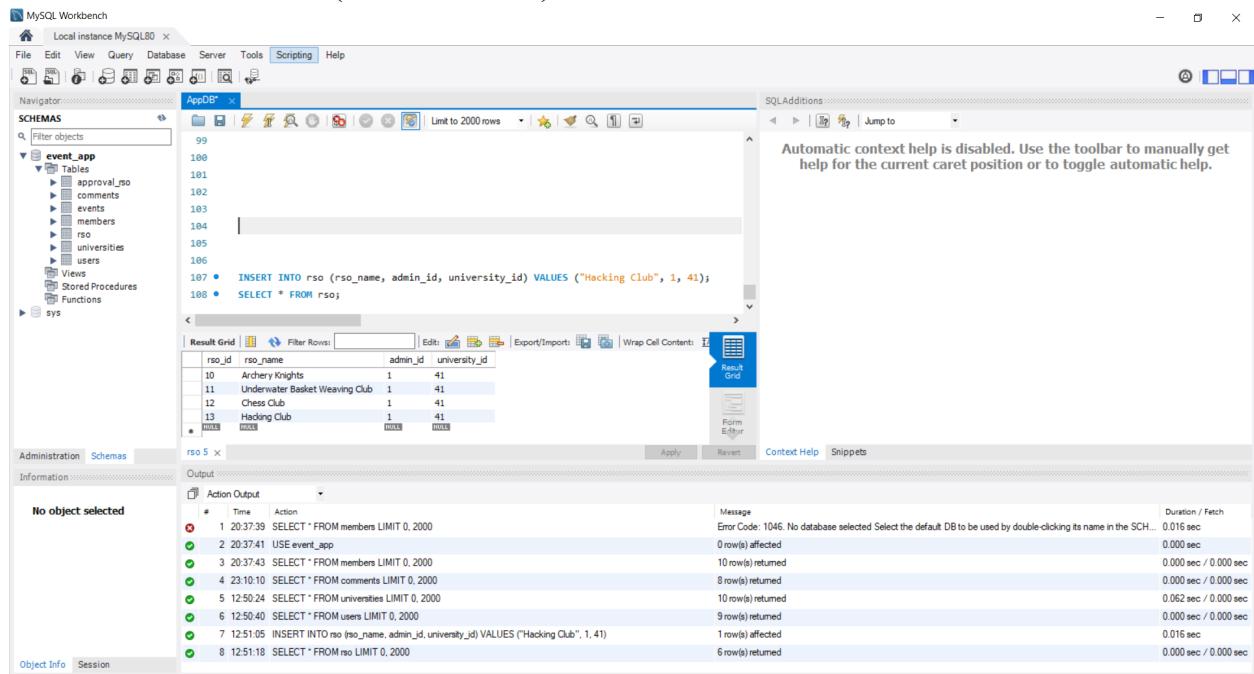
```
admin_id INT NOT NULL,  
university_id INT NOT NULL,  
PRIMARY KEY (rso_id),  
FOREIGN KEY (admin_id) REFERENCES users(user_id),  
FOREIGN KEY (university_id) REFERENCES universities(university_id)  
);
```

```
CREATE TABLE members (  
    member_id INT NOT NULL AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    rso_id INT NOT NULL,  
    PRIMARY KEY (member_id),  
    FOREIGN KEY (user_id) REFERENCES users(user_id),  
    FOREIGN KEY (rso_id) REFERENCES rso(rso_id)  
);
```

```
CREATE TABLE comments (  
    comment_id INT NOT NULL AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    event_id INT NOT NULL,  
    comment TEXT,  
    rating INT,  
    PRIMARY KEY (comment_id),  
    FOREIGN KEY (user_id) REFERENCES users(user_id),  
    FOREIGN KEY (event_id) REFERENCES events(event_id)  
);
```

SQL-Examples

Insert into RSO table (Create RSO)



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** event_app
- Tables:** approval_rso, comments, events, members, rso, universities, users
- Scripting:** Shows the SQL code used to create the RSO record:


```

107 • INSERT INTO rso (rso_name, admin_id, university_id) VALUES ("Hacking Club", 1, 41);
108 • SELECT * FROM rso;
      
```
- Result Grid:** Displays the newly created RSO record:

rso_id	rso_name	admin_id	university_id
10	Archery Knights	1	41
11	Underwater Basket Weaving Club	1	41
12	Cheer Club	1	41
13	Hacking Club	1	41
- Output:** Shows the execution history with the following log entries:

Action	Time	Action	Message	Duration / Fetch
1	20:37:39	SELECT * FROM members LIMIT 0, 2000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.016 sec
2	20:37:41	USE event_app	0 row(s) affected	0.000 sec
3	20:37:43	SELECT * FROM members LIMIT 0, 2000	10 row(s) returned	0.000 sec / 0.000 sec
4	23:10:10	SELECT * FROM comments LIMIT 0, 2000	8 row(s) returned	0.000 sec / 0.000 sec
5	12:50:24	SELECT * FROM universities LIMIT 0, 2000	10 row(s) returned	0.062 sec / 0.000 sec
6	12:50:40	SELECT * FROM users LIMIT 0, 2000	9 row(s) returned	0.000 sec / 0.000 sec
7	12:51:05	INSERT INTO rso (rso_name, admin_id, university_id) VALUES ("Hacking Club", 1, 41)	1 row(s) affected	0.016 sec
8	12:51:18	SELECT * FROM rso LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec

Insert into members table (Join RSO)

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

event_app

Tables: approval_rso, comments, events, members, rso, universities, users

Views, Stored Procedures, Functions

sys

AppDB* X

Limit to 2000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

members 8 X

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

members 8 X

Output

Action Output

No object selected

Time Action Message Duration / Fetch

5 12:50:24 SELECT * FROM universities LIMIT 0, 2000 10 row(s) returned 0.062 sec / 0.000 sec

6 12:50:40 SELECT * FROM users LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec

7 12:51:05 INSERT INTO `rso_name, admin_id, university_id` VALUES ('Hacking Club', 1, 41) 1 row(s) affected 0.016 sec

8 12:51:18 SELECT * FROM rso LIMIT 0, 2000 6 row(s) returned 0.000 sec / 0.000 sec

9 12:53:45 SELECT * FROM users LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec

10 12:54:04 SELECT * FROM members LIMIT 0, 2000 10 row(s) returned 0.000 sec / 0.000 sec

11 12:54:14 INSERT INTO members (user_id, rso_id) VALUES (9, 13) 1 row(s) affected 0.000 sec

12 12:54:17 SELECT * FROM members LIMIT 0, 2000 11 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

Insert into events table (Create Event)

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

event_app

Tables: approval_rso, comments, events, members, rso, universities, users

Views, Stored Procedures, Functions

sys

AppDB* X

Limit to 2000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

events 10 X

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Context Help | Snippets

events 10 X

Output

Action Output

No object selected

Time Action Message Duration / Fetch

8 12:51:18 SELECT * FROM rso LIMIT 0, 2000 6 row(s) returned 0.000 sec / 0.000 sec

9 12:53:45 SELECT * FROM users LIMIT 0, 2000 9 row(s) returned 0.000 sec / 0.000 sec

10 12:54:04 SELECT * FROM members LIMIT 0, 2000 10 row(s) returned 0.000 sec / 0.000 sec

11 12:54:14 INSERT INTO members (user_id, rso_id) VALUES (9, 13) 1 row(s) affected 0.000 sec

12 12:54:17 SELECT * FROM members LIMIT 0, 2000 11 row(s) returned 0.000 sec / 0.000 sec

13 13:00:26 SELECT * FROM events LIMIT 0, 2000 11 row(s) returned 0.000 sec / 0.000 sec

14 13:00:58 INSERT INTO events (event_name, category, description, new_time, date, location_name, latitude, longitude, contact_phone, contact_email, is_public, is_rso, user_id) VALUES ('UCF Football Game', 'Sports', 'Come watch the UCF Knights play!', '14:30:00', '2023-09-16', 'Spectrum Stadium', '28.6055', '-81.1995', '555-1234', 1) 1 row(s) affected 0.016 sec

15 13:01:01 SELECT * FROM events LIMIT 0, 2000 12 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

Insert into comments table (Post comment)

The screenshot shows the MySQL Workbench interface with the database 'AppDB' selected. In the 'Tables' section of the 'event_app' schema, the 'comments' table is highlighted. A query window displays the insertion of a new comment:

```

107 • INSERT INTO comments (user_id, event_id, comment) VALUES (9, 34, "WOOOO HOOOO GO KNIGHTS");
108 • SELECT * FROM comments;

```

The result grid shows the newly inserted row:

comment_id	user_id	event_id	comment	rating
15	1	29		4
16	9	10	Anyone got an extra bike I can borrow?	NULL
26	9	10	I'm being the Gatorade!!	NULL
27	9	34	WOOOO HOOOO GO KNIGHTS	NULL

The 'Action Output' pane shows the log of recent actions, including the insertion of the comment.

Update comments table (Edit comment)

The screenshot shows the MySQL Workbench interface with the database 'AppDB' selected. In the 'Tables' section of the 'event_app' schema, the 'comments' table is highlighted. A query window displays the update of a comment:

```

107 • UPDATE comments SET comment = "Edit this comment" WHERE comment_id = 27;
108 • SELECT * FROM comments;

```

The result grid shows the updated comment entry:

comment_id	user_id	event_id	comment	rating
15	1	29		4
16	9	10	Anyone got an extra bike I can borrow?	NULL
26	9	10	I'm being the Gatorade!!	NULL
27	9	34	Edit this comment	NULL

The 'Action Output' pane shows the log of recent actions, including the update of the comment.

Viewing public events

MySQL Workbench Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: AppDB*

SCHEMAS event_app

Tables: approval_rso, comments, events, members, rso, universities, users

Views, Stored Procedures, Functions, sys

Limit to 2000 rows

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid: SELECT * FROM events WHERE is_public = 1 AND approved_by_superuser = 1;

event_id	event_name	category	description	new_time	date	location_name	latitude	longitude
14	UCF Art Exhibit	Art Exhibit	Explore the latest art installations at the UCF Art ...	11:00:00	2023-10-31	UCF Art Gallery	28.65	-81.35
30	Napercise	Health	The best way for muscle recovery, building up y ...	13:30:00	2023-04-11	Knights Plaza, Gemini Blvd N, Oviedo, FL 32765, USA	28.65	-81.35
34	UCF Football Game	Sports	Come watch the UCF Knights play!	14:30:00	2023-09-16	Spectrum Stadium	28.65	-81.35

events 13 x

Action Output:

Time	Action	Message	Duration / Fetch
13:30:26	SELECT * FROM events LIMIT 0, 2000	11 row(s) returned	0.000 sec / 0.000 sec
13:00:58	INSERT INTO events (event_name, category, description, new_time, date, location_name, latitude, longitude)	1 row(s) affected	0.016 sec
13:01:01	SELECT * FROM events LIMIT 0, 2000	12 row(s) returned	0.000 sec / 0.000 sec
13:03:07	INSERT INTO comments (user_id, event_id, comment) VALUES (9, 34, "WOOOO HOOOO GO KNIGHTS")	1 row(s) affected	0.000 sec
13:03:10	SELECT * FROM comments LIMIT 0, 2000	9 row(s) returned	0.000 sec / 0.000 sec
13:04:45	UPDATE comments SET comment = "Edit this comment" WHERE comment_id = 27	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
13:04:51	SELECT * FROM comments LIMIT 0, 2000	9 row(s) returned	0.000 sec / 0.000 sec
13:06:19	SELECT * FROM events WHERE is_public = 1 AND approved_by_superuser = 1 LIMIT 0, 2000	7 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

Viewing RSO events (Chess RSO, and Gaming RSO)

MySQL Workbench Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator: AppDB*

SCHEMAS event_app

Tables: approval_rso, comments, events, members, rso, universities, users

Views, Stored Procedures, Functions, sys

Limit to 2000 rows

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid: SELECT * FROM events WHERE is_rso = 1 AND FIND_IN_SET(rso_id, '9,12') > 0;

event_id	event_name	category	description	new_time	date	location_name	latitude	longitude
32	Smash Brothers Tournament	Social gathering	Stop by the usual spot today in the student uni ...	12:30:00	2023-04-12	12715 Pegasus Dr, Orlando, FL 32816, USA	28.6015505	-81.35
35	Chess Tournament	Social / Competitive	We will be hosting a chess torunament in the St ...	12:00:00	2023-04-20	12715 Pegasus Dr, Orlando, FL 32816, USA	28.6015505	-81.35

events 20 x

Action Output:

Time	Action	Message	Duration / Fetch
13:06:19	SELECT * FROM events WHERE is_public = 1 AND approved_by_superuser = 1 LIMIT 0, 2000	7 row(s) returned	0.000 sec / 0.000 sec
13:12:40	SELECT * FROM rso LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
13:15:41	SELECT * FROM events WHERE is_rso = 1 AND rso_id IN (@ids) LIMIT 0, 2000	0 row(s) returned	0.000 sec / 0.000 sec
13:15:49	SELECT * FROM rso LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
13:16:15	SELECT * FROM events WHERE is_rso = 1 AND rso_id IN ("9,12") LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec
13:16:35	SELECT * FROM events WHERE is_rso = 1 AND rso_id IN ("9,12") LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec
13:16:46	SELECT * FROM events WHERE is_rso = 1 LIMIT 0, 2000	2 row(s) returned	0.000 sec / 0.000 sec
13:17:41	SELECT * FROM events WHERE is_rso = 1 AND FIND_IN_SET(rso_id, '9,12') > 0 LIMIT 0, 2000	2 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

Viewing Private Events

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80, AppDB*
- Navigator:** Schemas (event_app), Tables (approval_rso, comments, events, members, universities, users), Views, Stored Procedures, Functions, sys.
- SQL Editor:** Contains the following SQL query:

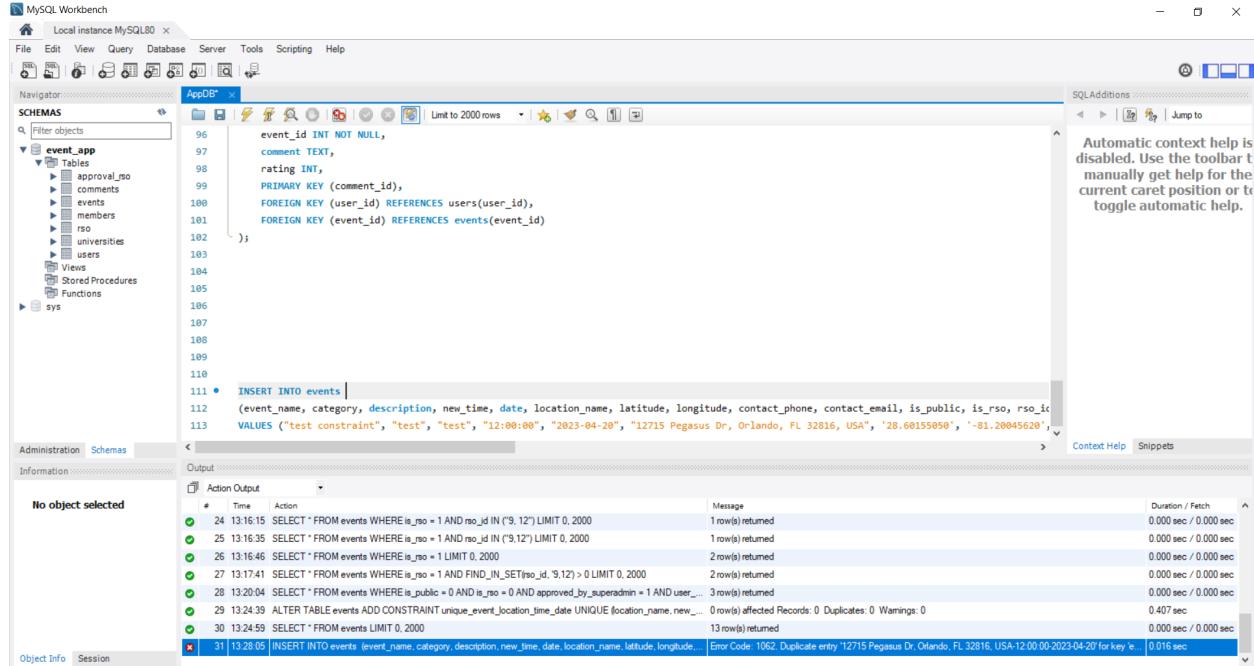

```
100
101
102
103
104
105
106
107
108 •  SELECT * FROM events WHERE is_public = 0 AND is_rso = 0
109 AND approved_by_superuser = 1 AND user_id IN (SELECT user_id FROM users WHERE university_id = 41 AND user_level = "superadmin"));
```
- Result Grid:** Shows a table with columns: event_id, event_name, category, description, new_time, date, location_name, lat, lon. Data rows include:

event_id	event_name	category	description	new_time	date	location_name	lat	lon
27	Cycle Into Spring	Recreation & Exercise	Ring in the spring season by taking part in a mo...	00:00:00	2023-04-28	4000 Central Florida Blvd, Orlando, FL 32816, U...	28.5...	-81.5...
28	Free Coffee and Study Space	Social Event	From 9:00am to 3:30 pm Monday-Friday, the W...	09:00:00	2023-04-11	1395 Campus View Ct, Oviedo, FL 32765, USA	28.6...	-81.5...
29	Coping with Social Anxiety	Workshop/Conference	This workshop will aim to increase participants u...	11:00:00	2023-04-11	12715 Pegasus Dr, Orlando, FL 32816, USA	28.5...	-81.5...
- Output:** Shows a list of actions with their times, messages, and durations:

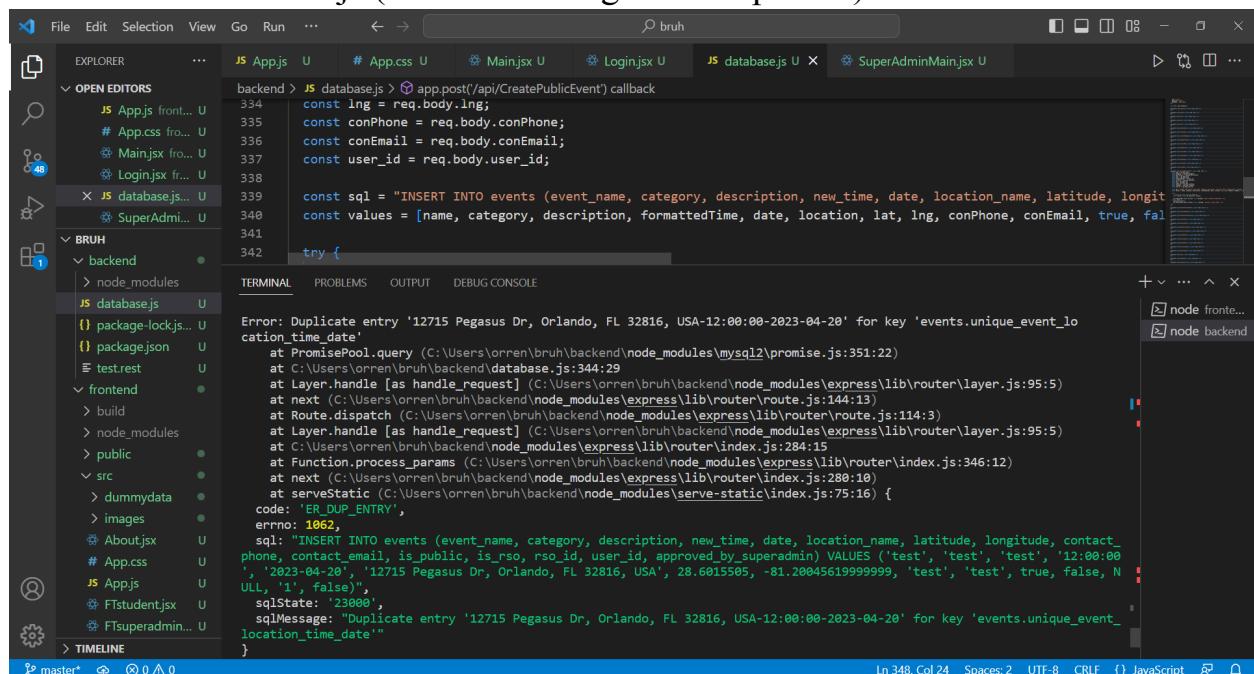
Action	Time	Message	Duration / Fetch
21 13:12:40	SELECT * FROM rso LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
22 13:15:41	SELECT * FROM events WHERE is_rso = 0 AND rso_id IN (@ids) LIMIT 0, 2000	0 row(s) returned	0.000 sec / 0.000 sec
23 13:15:49	SELECT * FROM rso LIMIT 0, 2000	6 row(s) returned	0.000 sec / 0.000 sec
24 13:16:15	SELECT * FROM events WHERE is_rso = 1 AND rso_id IN ("9,12") LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec
25 13:16:35	SELECT * FROM events WHERE is_rso = 1 AND rso_id IN ("9,12") LIMIT 0, 2000	1 row(s) returned	0.000 sec / 0.000 sec
26 13:16:46	SELECT * FROM events WHERE is_rso = 1 LIMIT 0, 2000	2 row(s) returned	0.000 sec / 0.000 sec
27 13:17:41	SELECT * FROM events WHERE is_rso = 1 AND FIND_IN_SET(rso_id, '9,12') > 0 LIMIT 0, 2000	2 row(s) returned	0.000 sec / 0.000 sec
28 13:20:04	SELECT * FROM events WHERE is_public = 0 AND is_rso = 0 AND approved_by_superuser = 1 AND user_...	3 row(s) returned	0.000 sec / 0.000 sec

Constraint Enforcement

Inserting a new event with overlapping location, date, and time.
MySQL:



Terminal for database.js (File containing API endpoints)



Error message shown to user: “Couldn’t create request”

An admin who is not the Admin of the RSO attempts to create an event for that RSO: Show an error message.

My frontend prevents this event from ever happening. When the user logins in an API endpoint runs to get all RSO's that the user is a member of. Then it loops over these RSO's (myRSOs) to check if the user is an admin to any of the RSOs. From there when a user clicks on the "Create RSO Event" button they must select which RSO to create an event for. The only options that appear are the ones returned from checking if the user is an admin to any of their RSOs (adminRSOs). Therefore if you are not an admin of an RSO you don't have the option to create an event for that RSO.

Function that gets all RSOS which the user is an admin of.

```
const checkIfAdmin = () => {
  for(let i = 0; i < myRSOs.length; i++){
    if(myRSOs[i].admin_id.toString() === user_id){
      //console.log("FOUND ONE");
      setIsAdmin(true);
      setAdminRSOs(adminRSOs.concat(myRSOs[i]));
    }
  }
}
```

Active / Inactive RSO's.

For an RSO to be active it must have 5 members. I don't have a column for activity in my RSO table. So I have my frontend check the (adminRSOs) array for 5 members. Therefore the admin can only create events for RSOs they are the admin of and are active RSOs

```
async function checkActivity(rso_id){
  try{
    const response = await api.post('/api/checkActivity', {"rso_id": rso_id});
    if(response.data.length >= 5){
      return true;
    }
    else{
      return false;
    }
  }catch(err){
    console.log(err);
  }
  return false;
}

const getActiveRSOs = () => {
  if( activeRSO.length === 0){
    adminRSOs.forEach(element => {
      if( checkActivity(element.rso_id) ){
        setActiveRSO(activeRSO.concat(element));
      }
    });
  }
}
```

Conclusion

Throughout building and testing the web app the database query response times were great, usually instantaneous. I've learned a lot from making this web app, I had prior experience with React Native from POOSD. Creating this made me learn MySQL since I had 0 prior experience with MySQL and databases in general. Also connecting the database to the react project took some time to learn but once I had it set up it was very easy to create the API endpoints. Styling the website was pretty easy to learn since I had some experience styling the mobile app for POOSD. When it came time to implement the Google Maps API that took me a long time to figure out. In retrospect I would have implemented it more securely by using a .env.local file to store the API key but since this website will only be hosted locally I didn't find it necessary. If I had more time I would've also liked to implement lazy loading when it comes to the events. Right now all events are loaded into the frontend when the user logins in. If there were heavy usage on the site that would be a terrible way to implement the viewing feed for events. Overall I learned a lot of valuable skills pertaining to web development over the course of this project. I also gained a solid introduction to how databases operate and how to use them efficiently.