

Group Report  
**College Event Website**

COP 4710  
Fall 2018  
Group 5

Jonathan Ponader  
Barath Tirumala  
Baran Barut  
Tiger Sachse

<b>Project Description</b>	<b>4</b>
<b>Database Design</b>	<b>4</b>
ER-Model	4
Relational Data Model	4
Create Table Script	4
Triggers	5
Assertions	5
Checks	5
SQL	5
View Events	5
Join RSO/Organization	6
New RSO/Organization	6
New Event	6
Commenting	7
<b>GUI</b>	<b>7</b>
Login Page	7
Dashboard	7
View Events	8
View Event	10
Create Event	11
Create RSO	12
Join RSO	12
Admin Tools	13
<b>Implementation</b>	<b>13</b>
Front End	13
Back End	13
Database	14
Server	14
<b>Constraint Enforcement</b>	<b>14</b>
Event occuring at same time and Place	14
Creating event for Organization Which you aren't an Admin For	14
Number of Members needed to have an RSO	14
<b>Sample Data</b>	<b>14</b>
Python Script	14
Event Table	15

User Table	15
Organization Table	15
University Table	15
<b>Advanced Features</b>	<b>15</b>
Mapping	15
Facebook	16
Availability	16
<b>Conclusion</b>	<b>17</b>
Database Performance	17
Additional Features	17
Problems encountered in project	17

# Project Description

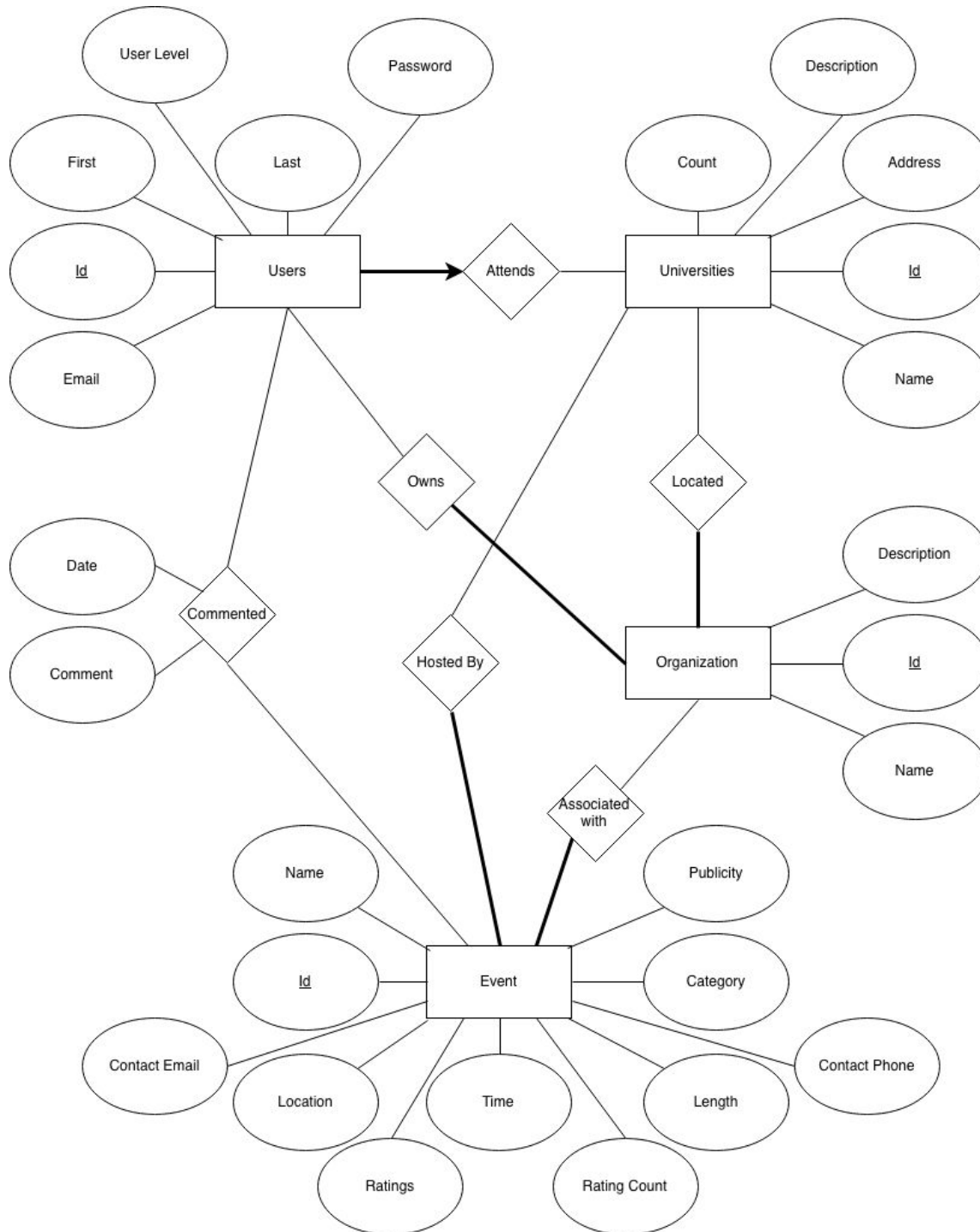
The College Event Website is a website for universities and student organizations to post information about upcoming events into a large, public feed for other students to discover. This is a powerful platform to connect and tie students in various universities together as well as serve as a front for organizations to post and update events which can be directed to the audience.

On which, you can view the following:

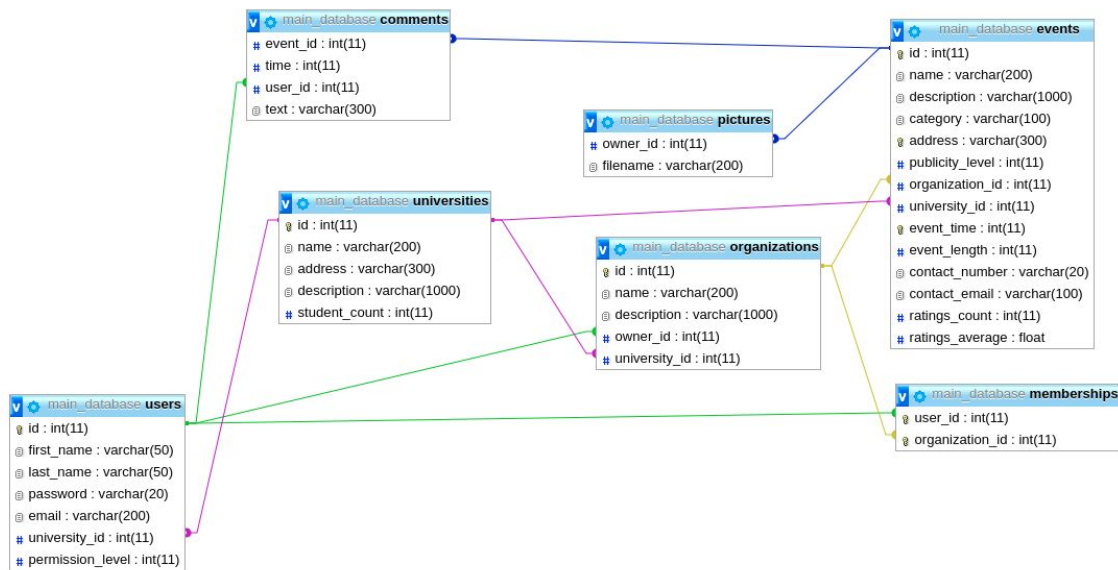
- Organizations: Create and view organizations and clubs at your local university!
- Events: View and create events for local organizations, and register for your favorites!
- Universities: Everything is tied to your university, so you'll always know what's going on around you!

# Database Design

## ER-Model



# Relational Data Model



## Create Table Script

This project includes an SQL script to quickly and easily build a new database with the tables described above. Each table contains several fields, and most of the tables contain foreign keys and a primary key.

## Triggers

There are several triggers included in this project. These triggers are written in the create table script. The triggers mainly focus on the university count field of the universities table. When a new student is added, the appropriate university's university count is incremented. If that student is ever removed, the appropriate university's university count is decremented. Finally, if a student changes universities, the count for the respective universities will change to match.

## Assertions

This project did not require any assertions.

## Checks

This project did not require any checks.

## SQL

Throughout the project we used a series of queries to be able to gather the needed information and present it to the user. Through php we used prepared statements, but we have condensed the statements into pure SQL for the ease of explanation.

### View Events

The view all event page has two main SQL queries. The first one simply looks for all event open to the public, publicity level 0, or event open to all student of a university, publicity level 2. The second query has to pull in the membership table to see if an event is only open to members and is the use a member of the organization. We had to separate these into two separate queries to ensure we got all of the wanted events without duplication.

```
SELECT DISTINCT * FROM `events` E where (E.publicity_level = 0) OR (E.publicity_level = 1  
AND E.university_id = '$_SESSION[univ]')
```

```
SELECT DISTINCT *  
FROM `events` E, `memberships` M  
WHERE(E.publicity_level = 2 AND E.university_id = '$_SESSION[univ]' AND M.organization_id  
= E.organization_id AND M.user_id = '$_SESSION[id]')
```

To view events also relies on some secondary queries, as organization id and university id are stored in the table not the actual name of the organization or university, which is what we want to show. To solve this we added our secondary queries.

```
SELECT name  
FROM organizations  
WHERE organizations.id = '$row[organization_id]'
```

```
SELECT name  
FROM universities  
WHERE universities.id = '$row[university_id]' LIMIT 0,1";
```

## Join RSO/Organization

To join an organization is dependent on 2 queries. The first query enforces the constraint that only members of the university can join an organization associated with that organization. This is by only showing organizations associated with the university to the user and giving them the ability to join. The second query simply inserts the membership into the membership using the organization id and the user id.

```
SELECT name,id
FROM organizations
WHERE organizations.university_id ='$_SESSION[univ]
```

```
INSERT INTO memberships (user_id, organization_id)
VALUES (:user_id, :organization_id')
```

## New RSO/Organization

Creating a new RSO only requires the user of one query. This query simply fills out all of the required information to create a new organization. The id of the organization is auto incremented therefore does not need to be incremented. The university id is also inserted, but does not need to be queried from the users profile because we have it stored in the session variable.

```
INSERT INTO organizations (name, owner_id, university_id, description)
VALUES (:name, :owner, :university, :description'))
```

## New Event

Creating a new event uses a simple query, that takes the input from the form and inserts it into the table. There are some supporting queries that help populate the date in the form. There is a query that looks ups all of the organizations that the user is an admin for and can create a event for that organization. The university is automatically populated, by the session variable.

```
INSERT INTO events (name, description, category, address, publicity_level, organization_id,
event_time, contact_number, contact_email, university_id, event_length)
VALUES (:name, :description, :category, :address, :publicity_level,
:organization_id,:event_time, :contact_number, :contact_email, :university_id, :event_length'))
```



## Commenting

Commenting is dependent on three queries, the first query shows all of the comments related to a event on the view event page. The second query allows for the user to edit a comment, the user can only access this feature if they own the comment.. They also have the option to delete the comment if they own the comment.

```
SELECT * FROM `comments` c WHERE c.event_id = $_GET[event]
```

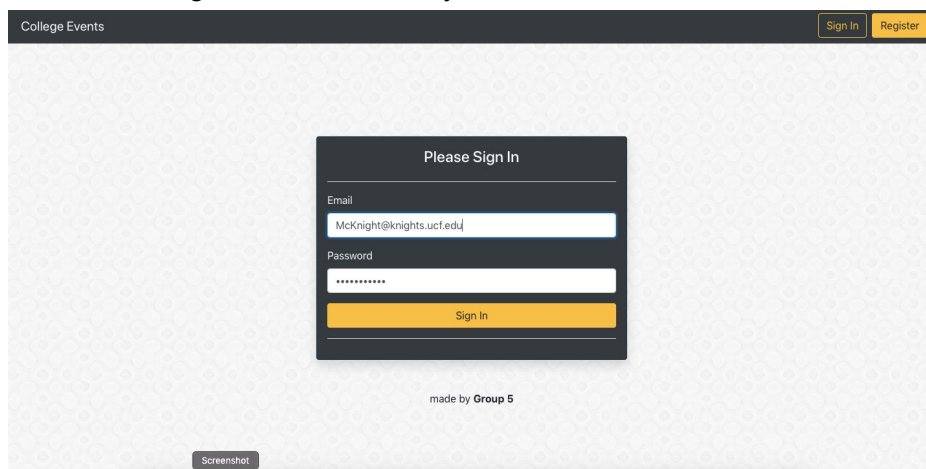
```
UPDATE comments  
    SET comment = '$comment'  
    WHERE id = '$_GET[comment]'
```

```
DELETE FROM comments  
    WHERE id = '$_GET[comment]'
```

## GUI

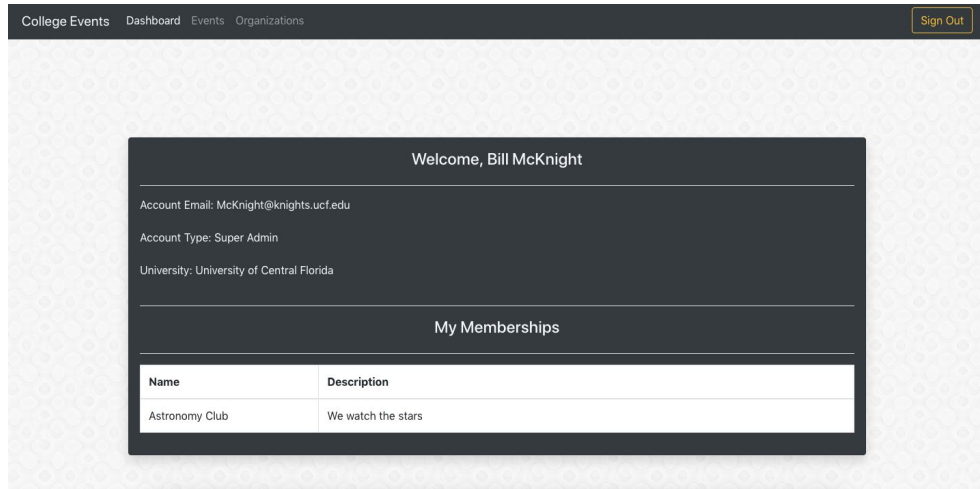
### Login Page

Upon entry to the site, you start at the Login page, this page allows for the user to log into the website and start viewing events. If the user does not have an account there is a link on the page to take them to a Register screen so they can create and account.



# Dashboard

Upon successful logging in the user is presented with the dashboard. This gives the user a quick run down of information about them, such as the university they are registered with and their user level. This page also shows the users memberships to organizations, this gives the user a quick way to make sure they will see events relevant to them



## View Events

The view event page shows all of the events that are open to all users, open to members at your university or open to organizations in which you are a member of. This ensures no events show up here that you are unable to attend. Each event has tons of information. This information includes the name, a description, the time and date, a pin on a map representing the location, as well as contact information for the organizer.

College Events

Dashboard

Events

Organizations



Sign Out

Add Event

Remove Event

Filter

Filter

Name	Category	Description	Date Time	Length	Location
Paint the Moon	Educational	Paint the Moon this wednesday night with us	11-07-18 09:45 pm	0	
Solar Eclipse Party	Fun	Celebrate Another Great Semester	11-30-18 08:00 pm	2	

College Events

Dashboard

Events

Organizations



Sign Out

Add Event

Remove Event

Filter

Filter

	Phone	Email	University	RSO	Publicity
	407-877-6542	McKnight@knights.ucf.edu	University of Central Florida	Astronomy Club	University Students Only
		McKnight@knights.ucf.edu	University of Central Florida	Astronomy Club	RSO Members Only

# View Event

If a user see an event they want to see more information on or wants to comment on, they can click on the event, and it will show up on its own page. This page shows all the information available on the event. This page also allows for user to leave comment on events. Users can edit and delete their own personal comments after they are left, however they cannot edit or delete other users comments. From this page the user can also share the event on facebook, this allows for them to get the word out without the need for having all users creating an account.

College Events

Dashboard

Events

Organizations

Sign Out

Share

Event

Edit Event

Name

Paint the Moon

Event University

University of Central Florida

Event Category

Educational

Event Publicity

University Students Only

RSO

Astronomy Club

Description

Paint the Moon this wednesday night with us

03:33 pm

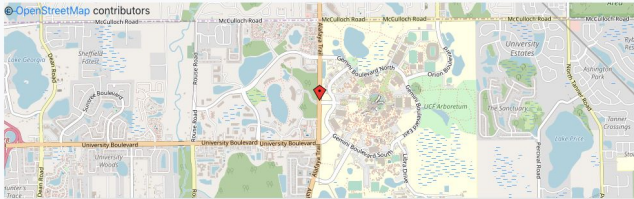
2

Date (mm/dd/yyyy)

12-07-18

Location

OpenStreetMap contributors



-9039998.427 3325163.565

Contact Phone

Contact Email

Jeremy.Barnett94@bellsouth.net

Back

Screenshot

Comments

Comment

Comment

Name	Comment	Modify
Jeremy	fun	<div>Edit</div> <div>Delete</div>

## Create Event

An admin or super admin can create an event for their owned organization. They can select the publicity of an event to control who can see there event, and which RSO/Organization its associated with. The user can use the map to pinpoint the exact location of the event for easy communication. These events are automatically associated with the university that the user is associated, making sure only students that attend a university can own a events at said university.

New Event

Name

Database Presentation

Event Category

Religious

Event Publicity

Open For All

RSO

Description

See our cool project in Action

Time

05:10 PM

Event Length (hours)


.15

Date (mm/dd/yyyy)

11/06/2018

Location

© OpenStreetMap contributors



-9038964.775 3325174.314

Contact Phone

239-313-9027

Contact Email

Mckinght@knights.ucf.edu

Back

Save Changes

## Create RSO

An admin or super admin can also create organizations. These organizations are automatically associated with the university that the user is associated, making sure only students that attend a university can own a organization at said university.

The screenshot shows the 'New Organization' form within the 'College Events' application. The form has a title 'New Organization' and two input fields: 'Name' with the placeholder 'Name of Organization' and 'Description' with the placeholder 'We do...'. At the bottom of the form are two buttons: 'Back' and 'Save Changes'. The application's navigation bar at the top includes 'College Events', 'Dashboard', 'Events', 'Organizations', and a 'Sign Out' button. A 'Screenshot' label is visible at the bottom of the image.

## Join RSO

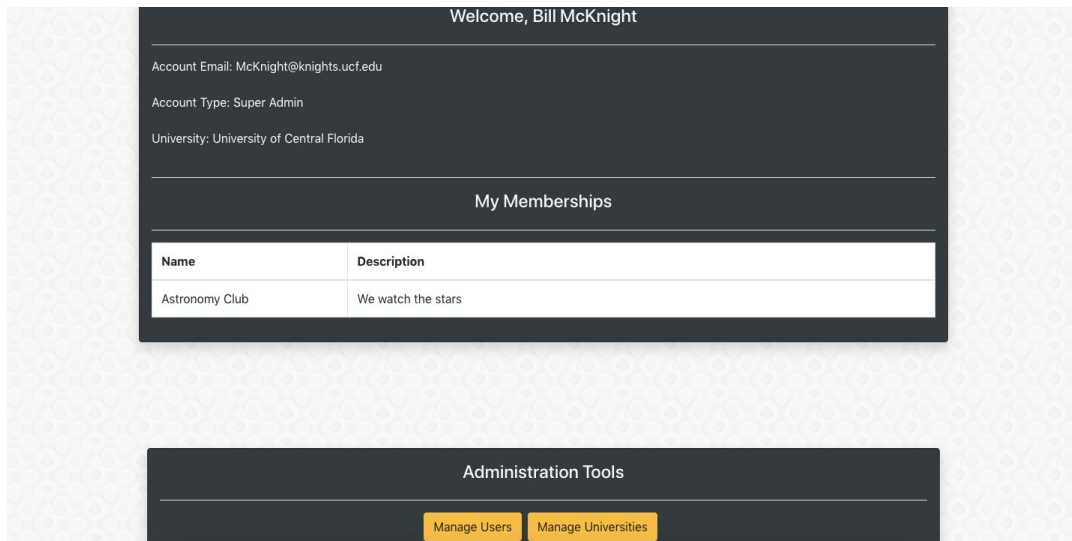
Any student that attends a university that an organization is associated with has the ability to join the organization. This can be done with a simple click. Once an organization is joined it all events associated with organization will show up in the users event feed.

The screenshot shows the 'Join Organizations' page within the 'College Events' application. The page has a title 'Join Organizations' and a 'Back' button. Under the 'Organizations' section, there is a list of organizations with a checkbox and a 'Join' button. The first organization listed is 'Astronomy Club' with the description 'SAE'. Below this, there is a section titled 'My Memberships' which contains a table with two columns: 'Name' and 'Description'. The table has one row with the values 'Astronomy Club' and 'We watch the stars'.

Name	Description
Astronomy Club	We watch the stars

## Admin Tools

Users who are admin automatically have access to the admin tools. Any admin can add users, but only super admins can add universities. All of these tools are available from the bottom of the user's dashboard.



## Implementation

### Front End

The front end of the website was constructed with a HTML and CSS styling. This was the most efficient way to get a nice looking website in the shortest amount of time. There is also some uses of Javascript to improve the look and feel of the website, as well as facilitating the mapping solution. The mapping solution uses Openlayer.js a open source mapping solution. This was the best choice for the project as it did not cost to implement and can be easily be traded out for other options latter on. To give the website the ability to act dynamically the website backend also echos some of the html to change the layout and view based on the user. We also used the Bootstrap library to help with the brunt of the CSS work, as Bootstrap makes website look very nice, very easily.

### Back End

The backend of the website uses php to communicate with the website. This was a new technology for many of us, and was a great learning experience. The php let it self to be very useful as it allowed us to make our website very dynamic by being able to echo new html with

variables set based on the user, as well as allowing us to pass variables via the url. PHP also gave us a simple way to query the database. Making the act of accessing data very easy.

## Database

For this project, we stuck with the tried and true DBMS MySQL. We had originally chosen MariaDB for our DBMS which is a free and open-source fork of MySQL more open to privacy concerns and community. MariaDB caused us to have some issues, as it was a new technology for all of us. After a couple weeks of messing around with MariaDB, we chose to instead go with MySQL as we were much more familiar with it. Due to the DBMS being the core part of this project (can't have a website without a database), it was crucial that we got the database software up and running as soon as possible. MySQL is very reliable and can handle rows of data in the tens of millions. Overall our database implementation was extremely standard. We made no tweaks to MySQL besides allowing users to login to the root user account with a password. MySQL proved to be extremely useful for this project as it had no gimmicks or quirks, and no issues whatsoever. Our DBMS simply worked, and the database was implemented using MySQL swimmingly.

## Server

For our backend server we chose to run linux using Ubuntu 18.04 LTS. We chose Ubuntu because those helping to set up the server were very familiar with the OS. DigitalOcean (our hosting provider) allowed a "one-click install" of Ubuntu with the entire LAMP (Linux, Apache, MySQL, PHP) stack already installed. In terms of operating systems, Ubuntu - just like MySQL - is very standard and basic, and does everything we need it to do. After making sure all of the LAMP components were properly installed and configured, we implemented a couple of firewall rules and were up and running in minutes. We used Apache2 as our web-server, as it came pre-installed with the LAMP stack. There isn't much of a difference between Apache2 and NGinx, so we just stuck with Apache2. Installing modules for PHP7 was as simple as a one-line command. With Digital Ocean we are guaranteed consistent uptimes and on-demand support in the event that we had an issue. We couldn't have asked for an easier time setting up our server.

## Constraint Enforcement

### Event occurring at same time and Place

Events cannot occur at the same time and place, as per the assignment description. To ensure that this never happens, in the events table, the primary key for each entry is the location crossed with the time. Event times are stored in Unix Time, and locations are stored as coordinates. This will ensure that no two entries occur at the same time and same place.



## Creating event for Organization Which you aren't an Admin For

This constraint is enforced by the GUI, as the user does not have the option to create event for admins they are not admins for. The create event button is only shown if the users user level is an admin or super admin. Then from there we run a query to see if the user is the owner of the organization. Only then can they create an event associated with the organization.

```
SELECT name,id  
FROM organizations  
WHERE organizations.owner_id ='$_SESSION[id]'
```

## Number of Members needed to have an RSO

This constraint is not enforced as we see this as a tool for organizations and not a organization management page. This is designed as a way to get the word out about an event. Also the only way to get the privilege to have an event is to be an admin, meaning that you have been upgraded from a regular user to admin or super admin. Meaning you have been approved by some higher power.

## Sample Data

### Python Script

This project generates a huge heap of sample data using Python scripts. These scripts create objects for events, universities, users, etc. and populate variables for each field in the respective table. These variables are partially generated with the random Python package, and partially generated with the faker Python package. The faker Python package generates natural language random strings for all sorts of things, like names, emails, and addresses. Random numbers can simply be generated using the random Python package for things like phone numbers, coordinates, and times. All script data is then written into SQL scripts, which can be run by MySQL and inserted into the tables.

### Event Table

Given the sample data we have made, below is a screenshot of some of this data as it actually appears in our Database. Below you can see the events and their corresponding attributes as relevant to the website.

Showing rows 0 - 24 (1003 total, Query took 0.0004 seconds.)

SELECT \* FROM 'events'

Profiling | Edit inline | [Edit] | [Explain SQL] | [Create PHP code] | [Refresh]

1 > >> Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options		id	name	description	category	address	publicity_level	organization_id	university_id	event_time	event_length	contact_number	contact_email	ratings_count	rating
<input type="checkbox"/> Edit Copy Delete	1004	Paint the Moon	Paint the Moon this wednesday night with us	Educational	-9038233.846 3325145.650	1	4002	1001	1541627100	NULL	407-877-6542	McKnight@knights.ucf.edu	NULL		
<input type="checkbox"/> Edit Copy Delete	1006	Database Presentation	See our cool project in Action	Religious	-9038964.775 3325174.314	1	4002	1001	1541524200	9	239-313-9027	McKnight@knights.ucf.edu	NULL		
<input type="checkbox"/> Edit Copy Delete	1005	Solar Eclipse Party	Celebrate Another Great Semester	Fun	-9039772.142 3324811.238	2	4002	1001	1543608000	120		McKnight@knights.ucf.edu	NULL		
<input type="checkbox"/> Edit Copy Delete	712	Church of John Meeting	Quis ad sit expedita neque. Aut amet dolores dolor...	fugit	-9075682.456041075 3539694.2605946697	1	2081	748	1543478063	660	701-799-1617	brooksvictor@hotmail.com	923		
<input type="checkbox"/> Edit Copy Delete	210	Party for Heather	Deleniti vitae nisi sed voluptas. Esse dolore enim...	nulla	-9075749.464631962 3102791.328871661	2	2827	39	1541056228	720	04191460128	michael05@yahoo.com	33		
<input type="checkbox"/> Edit Copy Delete	349	Church of Eric Meeting	Voluptates dicta iure tenetur tenetur quod autem...	maxime	-9075782.442989059 3457678.386052326	2	2519	175	1541713509	240	393-864-1304	devinjohnson@gmail.com	595		
<input type="checkbox"/> Edit Copy Delete	649	Wake for Nicole	Itaque saepe possimus repellendus delectus eligend...	odit	-9075787.5235771 3044286.9146907874	2	1615	258	1542941776	600	1-801-243-0480x590	ajohnson@hotmail.com	345		
<input type="checkbox"/> Edit Copy Delete	15	Church of Enka Meeting	Cumque id natus dolores a exercitationem temporibu...	aspematur	-9075840.441278566 3406857.179630951	1	686	925	1541216441	600	1-842-601-9619x55862	michaelkelly@yahoo.com	153		
<input type="checkbox"/> Edit Copy Delete	241	Church of Madison Meeting	Impedit officia ea tempora officia officia. Fugit...	odio	-9075908.72444919 3224922.521600972	2	1966	706	1541510704	420	550.319.7668x03691	brownpamela@gmail.com	268		
<input type="checkbox"/> Edit Copy Delete	52	Party for Melissa	Expedita qui perferendis	debitis	-9075914.833277375 3116630.158399511	1	915	388	1540935996	300	039-848-4174x4843	ktucker@yahoo.com	310		

## User Table

Given the sample data we have made, below is an screenshot of some of this data as it actually appears in our Database. Below you can see the sample Users currently registered at the time. The relevant attributes tied to each of the users are also shown below.

```
SELECT * FROM `users` ORDER BY `university_id` DESC
```













> >>

25 ▼

Sort by key:

university\_id (DESC) ▼

 $\leftarrow T \rightarrow$ 

			id	first_name	last_name	password	email	university_id	1	permission_level
	 Edit	 Copy	 Delete	10002	Bill	McKnight	test	McKnight@knights.ucf.edu	1001	2
	 Edit	 Copy	 Delete	9773	Justin	Lee	n2owqi7_ycrmaJx	Justin.Lee39@outlook.com	999	2
	 Edit	 Copy	 Delete	8957	Paul	Jennings	y!2flN6ujxrvjh	Paul.Jennings41@yahoo.com	999	2
	 Edit	 Copy	 Delete	8694	Victoria	Montgomery	okb3r!rsoG8oitbm	Victoria.Montgomery98@yahoo.com	999	0
	 Edit	 Copy	 Delete	8384	Hannah	Hinton	9nybmsJon6v.erat	Hannah.Hinton34@yahoo.com	999	0
	 Edit	 Copy	 Delete	4127	Anthony	Andersen	ks!lhv4tXihdca8kxv	Anthony.Andersen01@gmail.com	999	2
	 Edit	 Copy	 Delete	3907	Mark	Campbell	oe7law5Wg?sklt	Mark.Campbell50@outlook.com	999	0
	 Edit	 Copy	 Delete	3439	Lisa	Archer	n2a!rNtpbuo8jaxo	Lisa.Archer61@gmail.com	999	2
	 Edit	 Copy	 Delete	2966	Lisa	Jones	Ub?lin9kf2uzs	Lisa.Jones84@bellsouth.net	999	2
	 Edit	 Copy	 Delete	2889	Julia	Ibarra	?smkcp6rwy2doptoIE	Julia.Ibarra98@bellsouth.net	999	2
	 Edit	 Copy	 Delete	2420	Patricia	Hancock	m!akdn!rjh4gnpp3bN	Patricia.Hancock81@aol.com	999	2
	 Edit	 Copy	 Delete	1506	Laura	Valencia	iqduih.jpvaRh85r	Laura.Valencia74@bellsouth.net	999	1
	 Edit	 Copy	 Delete	887	Richard	Stout	ph9.leh!jh4Gi	Richard.Stout91@gmail.com	999	0
	 Edit	 Copy	 Delete	792	Christopher	Alvarez	fat7hfm.rph9zgP	Christopher.Alvarez39@outlook.com	999	2
	 Edit	 Copy	 Delete	322	Taylor	Long	Ykjgba5!zdapga6btrc	Taylor.Long80@outlook.com	999	2
	 Edit	 Copy	 Delete	8845	Dawn	Barber	byjk2kkgnoelq?Dvei	Dawn.Barber16@aol.com	998	2
	 Edit	 Copy	 Delete	5443	Amanda	Mcdonald	mw9lcEI4oz!	Amanda.Mcdonald48@aol.com	998	2
	 Edit	 Copy	 Delete	5244	Dylan	Wagner	ni9uuwzuwcv07qk!l	Dylan.Wagner54@bellsouth.net	998	0
	 Edit	 Copy	 Delete	4876	Becky	Smith	re0ewargzw1ezxEe.	Becky.Smith82@gmail.com	998	0
	 Edit	 Copy	 Delete	4226	Robin	Sanders	l8jYylwko!bckom6	Robin.Sanders38@gmail.com	998	0
	 Edit	 Copy	 Delete	3747	Alexander	Miller	9atbMdl6v?m	Alexander.Miller65@yahoo.com	998	2

Showing rows 0 - 24 (4003 total, Query took 0.0003 seconds.)

SELECT \* FROM `organizations`

1
>
>>

Number of rows:
25

Filter rows:

Sort by key:
None

+ Options

				id	name	description	owner_id	university_id
<input type="checkbox"/>	Edit	Copy	Delete	1	Cardenas-Hernandez	Soluta ab nemo quibusdam quaerat aliquam cum. Ut t...	2873	276
<input type="checkbox"/>	Edit	Copy	Delete	2	Sparks, Marks and Cox	Dolores sunt nemo exercitationem. Blanditiis excep...	2154	702
<input type="checkbox"/>	Edit	Copy	Delete	3	Wolfe, Ortega and Scott	Blanditiis quas doloribus dolores nisi blanditiis ...	6934	417
<input type="checkbox"/>	Edit	Copy	Delete	4	Berry Inc	Officiis reprehenderit laborum voluptatum earum. C...	5203	20
<input type="checkbox"/>	Edit	Copy	Delete	5	Taylor, Delgado and Jackson	Dignissimos quas dignissimos quas.	5098	28
<input type="checkbox"/>	Edit	Copy	Delete	6	Anderson, Higgins and Jacobs	Accusantium nobis dolores error tenetur amet magni...	6229	659
<input type="checkbox"/>	Edit	Copy	Delete	7	Green Ltd	Libero deserunt ipsam. Occaecati repudiandae tenet...	4131	350
<input type="checkbox"/>	Edit	Copy	Delete	8	Richardson, Howard and Freeman	Provident quod necessitatibus ad architecto. Magna...	9474	840
<input type="checkbox"/>	Edit	Copy	Delete	9	Rhodes-Petty	Quia dignissimos expedita nisi. Ipsum sint earum. ...	7881	629
<input type="checkbox"/>	Edit	Copy	Delete	10	Shepherd PLC	Optio sint ab harum natus ipsam architecto unde. P...	9252	688
<input type="checkbox"/>	Edit	Copy	Delete	11	Clark-Anderson	Dolore atque a voluptate quasi nobis minus. Totam ...	5431	724
<input type="checkbox"/>	Edit	Copy	Delete	12	Waters-Aguilar	Rerum dicta incidunt optio. Magnam alias reiciendi...	2756	63
<input type="checkbox"/>	Edit	Copy	Delete	13	Johnson, Larson and Wilson	Illo nemo ullam harum ullam facere. Corrupti ipsam...	6235	805
<input type="checkbox"/>	Edit	Copy	Delete	14	Santos-Cuevas	Cupiditate nesciunt voluptas mollitia pariatur exc...	1051	684
<input type="checkbox"/>	Edit	Copy	Delete	15	Werner, Griffith and Rasmussen	Ex quas magni soluta voluptas.	6126	302
<input type="checkbox"/>	Edit	Copy	Delete	16	West Ltd	Laudantium quis commodi nesciunt iusto inventore. ...	4171	366
<input type="checkbox"/>	Edit	Copy	Delete	17	Graham, Wilson and Garcia	Tenetur similique veniam atque. Ducimus numquam od...	2070	518
<input type="checkbox"/>	Edit	Copy	Delete	18	Carroll-Torres	Velit quis magni nulla quas atque ad. Id similique...	7754	408
<input type="checkbox"/>	Edit	Copy	Delete	19	Colon, Cameron and Rivera	Eligendi magni enim assumenda distinctio sapiente ...	6387	375
<input type="checkbox"/>	Edit	Copy	Delete	20	Rowe, Bell and Jones	Id vitae ullam pariatur asperiores magni. Quis mag...	5135	574
<input type="checkbox"/>	Edit	Copy	Delete	21	Taylor, Salazar and Horne	Dolores tenetur omnis nostrum. Dicta repellendus s...	223	535

## University Table

Given the sample data we have made, below is an screenshot of some of this data as it actually appears in our Database. Below you can see the universities currently registered in the database. The name, address, description, and total student population are also shown.



Showing rows 0 - 24 (1000 total, Query took 0.0005 seconds ) [id: 1000... - 976...]

SELECT \* FROM `universities` ORDER BY `id` DESC

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ]

1 > >> Number of rows: 25 Filter rows: Search this table Sort by key: PRIMARY (DESC)

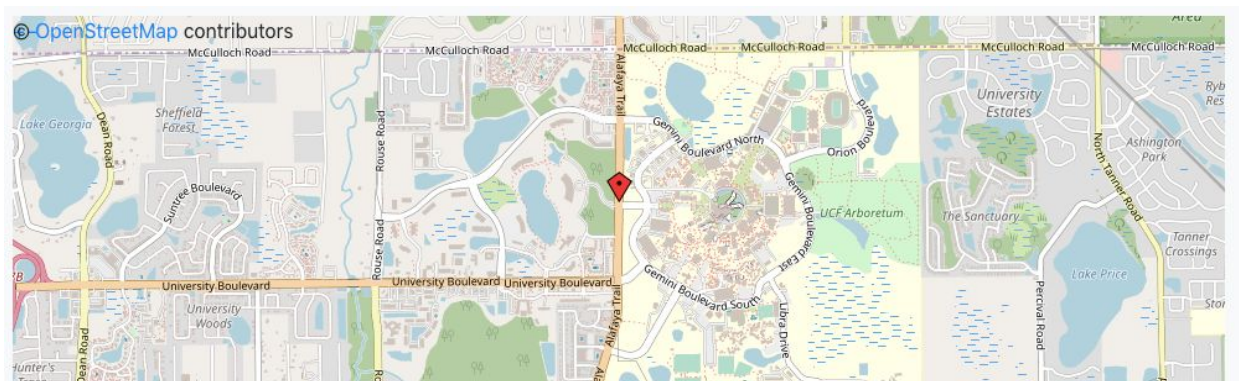
+ Options			id	1	name	address	description	student_count
<input type="checkbox"/>	Edit Copy Delete		1000		Lewis State University	3624 Lewis Court, Port Victorchester, SD 34617	Asperiores occaecati corrupti delectus eligendi. I...	0
<input type="checkbox"/>	Edit Copy Delete		999		University of Rivera	743 Rivera Street, Doylefort, ME 04726	Rem aliquid cupiditate voluptate beatae. Quis pari...	7
<input type="checkbox"/>	Edit Copy Delete		998		Higgins University	33732 Higgins Court, Port Alejandra, TN 07336	Eos omnis pariatur exercitationem. Praesentium arc...	7
<input type="checkbox"/>	Edit Copy Delete		997		Smith Community College	52138 Smith Road, Shannontown, KS 17109	Commodi esse quas sequi voluptatibus placeat. Exer...	7
<input type="checkbox"/>	Edit Copy Delete		996		Mooney Community College	04249 Mooney Road, Lauraville, FL 25256	Tenetur officia provident ipsum totam itaque. Vero...	8
<input type="checkbox"/>	Edit Copy Delete		995		Bennett State College	1341 Bennett Street, South Angelaborough, FL 80423	Saepe voluptate assumenda culpa unde exercitatione...	13
<input type="checkbox"/>	Edit Copy Delete		994		Crawford State University	4564 Crawford Road, Curtisberg, MI 44285	Necessitatibus sed inventore. Ad placeat expedita ...	12
<input type="checkbox"/>	Edit Copy Delete		993		Conner State University	8637 Conner Road, Chelseyland, WA 61022	Consequuntur doloremque quis. Similique labore ist...	10
<input type="checkbox"/>	Edit Copy Delete		992		University of Lopez	703 Lopez Way, Joshuaberg, DC 34181	Quo ex amet incidunt corporis adipisci. Debitis sa...	10
<input type="checkbox"/>	Edit Copy Delete		991		Park State University	564 Park Street, Michaelberg, MT 41838	Quis atque similique non amet aliquid eius. Corpor...	9
<input type="checkbox"/>	Edit Copy Delete		990		Crawford State College	439 Crawford Road, Port Charles, VT 05639	Possimus at labore iusto. Illo est quibusdam ipsam...	15
<input type="checkbox"/>	Edit Copy Delete		989		Terry University	254 Terry Road, Port Tara, NH 06314	Sapiente assumenda veritatis numquam occaecati lab...	6
<input type="checkbox"/>	Edit Copy Delete		988		University of Espinoza	30101 Espinoza Street, Lake Nancyville, MS 33926	Sed tempore voluptates assumenda natus debitis dol...	18
<input type="checkbox"/>	Edit Copy Delete		987		Arellano Community College	7765 Arellano Court, Danielport, NC 33591	Consequatur unde enim sit repellat. Totam nobis pr...	12

Console

## Advanced Features

### Mapping

The mapping was done using Openlayer.js an open source library, that uses open street view for the source of it mapping data. This was a great solution for this project as it was a no cost solution and can be easily swapped out for other technology latter on. The mapping gives the user to pinpoint the location of their event. While also giving the user the ability to change the location if needed. The location is then stored in the database as a coordinate pair for easy retrieval. Another benefit of this is the ability to easily compare the locations of two events, to ensure they are not occurring in the same location.



## Facebook

The facebook integration allows the user to easily share any event on facebook. This allows for events to get more publicity, and be seen by a greater audience. This integration is a very easily done integration as facebook has their own api, and provides all the needed code to simply add the feature to your website. We did have to modify the code a little to have it change dynamically, as the view event page it is on changes based on what event id is stored in the URL. To do this we added some php that will echo the proper address for the facebook link to point to.

The screenshot shows a web interface for sharing an event. At the top, there's a navigation bar with 'College Events', 'Dashboard', 'Events', and 'Organizations', and a 'Sign Out' button. Below this is a 'Share' form titled 'Event' with an 'Edit Event' button. The form contains several input fields: 'Name' (filled with 'Paint the Moon'), 'Event University' (filled with 'University of Central Florida'), 'Event Category' (filled with 'Educational'), 'Event Publicity' (filled with 'University Students Only'), 'RSO' (filled with 'Astronomy Club'), and 'Description' (filled with 'Paint the Moon this wednesday night with us'). Below the form is a blue bar with the Facebook logo and the text 'Share on Facebook'. Underneath that is a button labeled 'Share on Your Timeline' with a dropdown arrow. At the bottom, there's a blue square icon and the text 'This is my event'. Finally, there's a grey bar at the very bottom containing the IP address '206.189.228.110' and the text 'College Events - Event'.

## Availability

DigitalOcean (our server hosting provider) offers “High Availability” server systems for users with critical systems. When ordering a high availability server, DigitalOcean guarantees you 99% availability in a period of one year (up to 1% downtime). These downtimes include scheduled and unscheduled maintenance, so usually the downtime is even less. DigitalOcean also has many tutorials on increasing availability using software like “heartbeat” and “pacemaker”. We

opted-in to DigitalOcean's "Floating IP" feature, which allows our server to change IPs in the event that a load-balancer needs to switch the server's IP. If there is an increased amount of traffic or load on a specific IP, DigitalOcean automatically switches our IP to another, to help prevent downtime. On top of all of this, in the event that our server does go down, DigitalOcean has extremely competent support that is available around the clock.

## Conclusion

### Database Performance

Using PHPMYAdmin, we were able to manage our database and its performance very well. PHPMYAdmin provides a GUI front-end, and as a result it is slightly slower than MySQL's command-line interface. However, as seen in the picture below, even using PHPMYAdmin, we are able to select, sort, and order by over 10,000 rows in our "users" table in under 0.0006 seconds. This being said, fetching data through PHP yielded select query response times in less than 0.0005 seconds. We would suggest using an index such as a B+ index to quickly slim down specific queries by factors such as first letter of the first or last name instead of iterating over something like 10,000 rows. Overall, due to the small size of our database and the nature of the project, we foresee absolutely no issues with database performance, and are seeing speeds of down to .5-.10 ms for some of our more complicated queries.



### Additional Features to Consider

#### Security Plans

- Escape all user input using htmlspecialchars to prevent XSS and SQLi
- Harden the web-server using fail2ban to prevent against unauthorized SSH access
- Hashing and Salting passwords

#### Plans to Broaden Project

- We can broaden the project to allow access to entities other than universities
- Advertise the project to other students to get the word around
- Scrape and insert data from UCF event log at regular intervals using Python and BeautifulSoup

## Recovery Plans

- Backup database once a day as an offline copy
  - Rotate out backups weekly to another, separate server

## Plans to Enable Different “Views” for Accounts

- Allow people who haven’t logged in to access our website up to a certain extent without having to create an account

## Problems encountered in project

### Learning new languages

- This project required us to interact with and learn many languages that some of us were not familiar with. It challenged us to broaden our programming background while also further progressing our personal development skills. In the end, we have background in new technologies that we would not have had the opportunity to work with before.





