Title: Group Report

Project: College Event Website

Course: COP4710 Database Systems, Spring 2017

Group: 17

Group Members:
Brian Nguyen
William Sawran
Daniel Truong
Camilo Junoy

Table of Contents:

Project Description:

Our project is a web application for universities in Florida that allows users to create an account, login and view/attend events depending on what university or organization they belong to.

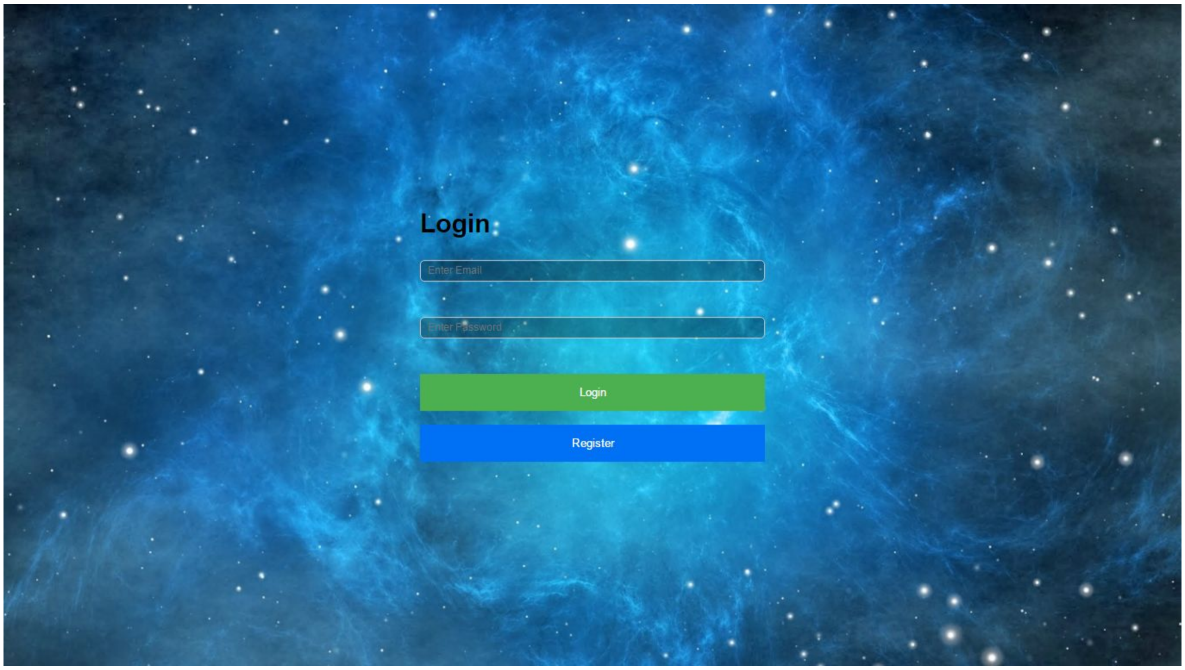
Users will be able to view a detailed event page for more information and be able to rate/comment on that particular event.
Users will also be able to create or join an RSO as long as they meet the constraints to create/join one. Admins of an RSO may create an event and Super Admins will be able to create profiles for universities.

GUI:

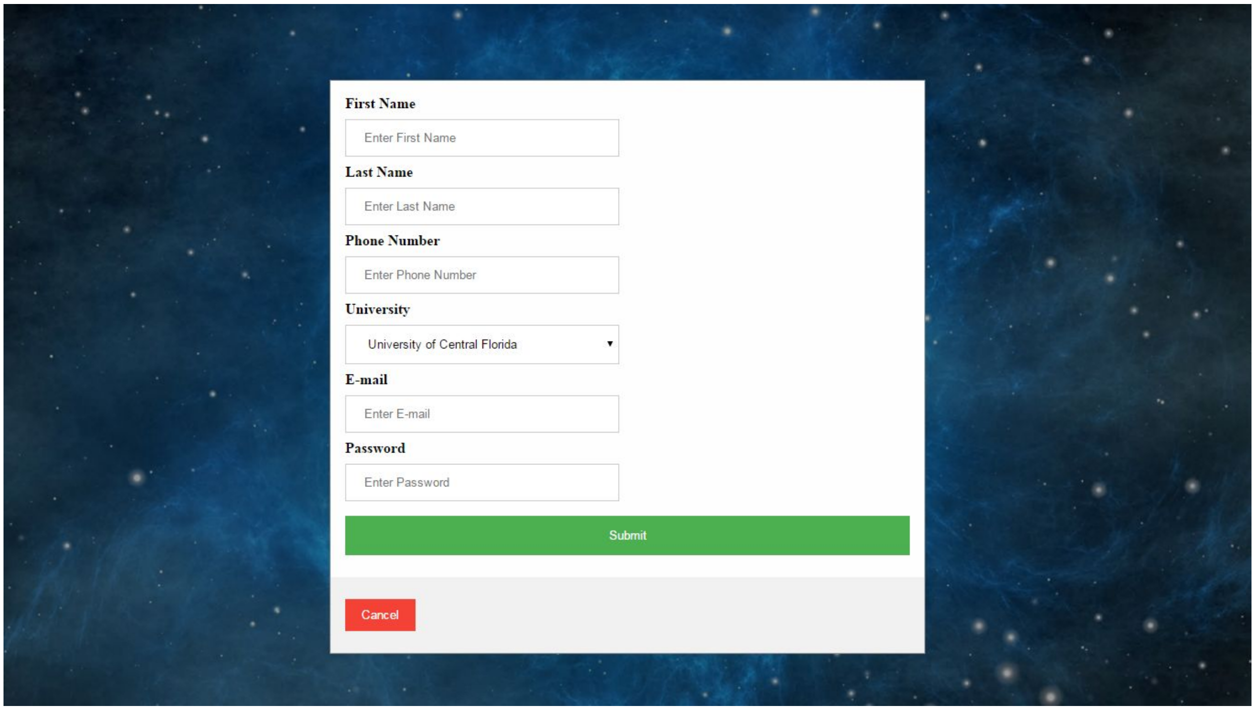
Our application is a web-based application that is hosted locally using the XAMPP stack (cross-platform, Apache, MySQL (MariaDB), PHP, and Perl to implement our PHP code and integrate with our database. The frontend website aspect of our application is written using HTML, CSS, and JavaScript. Our application uses the DBMS (Database Management System), MySQL.

Screenshots:

Login:



Register:



First Name

Enter First Name

Last Name

Enter Last Name

Phone Number

Enter Phone Number

University

University of Central Florida ▾

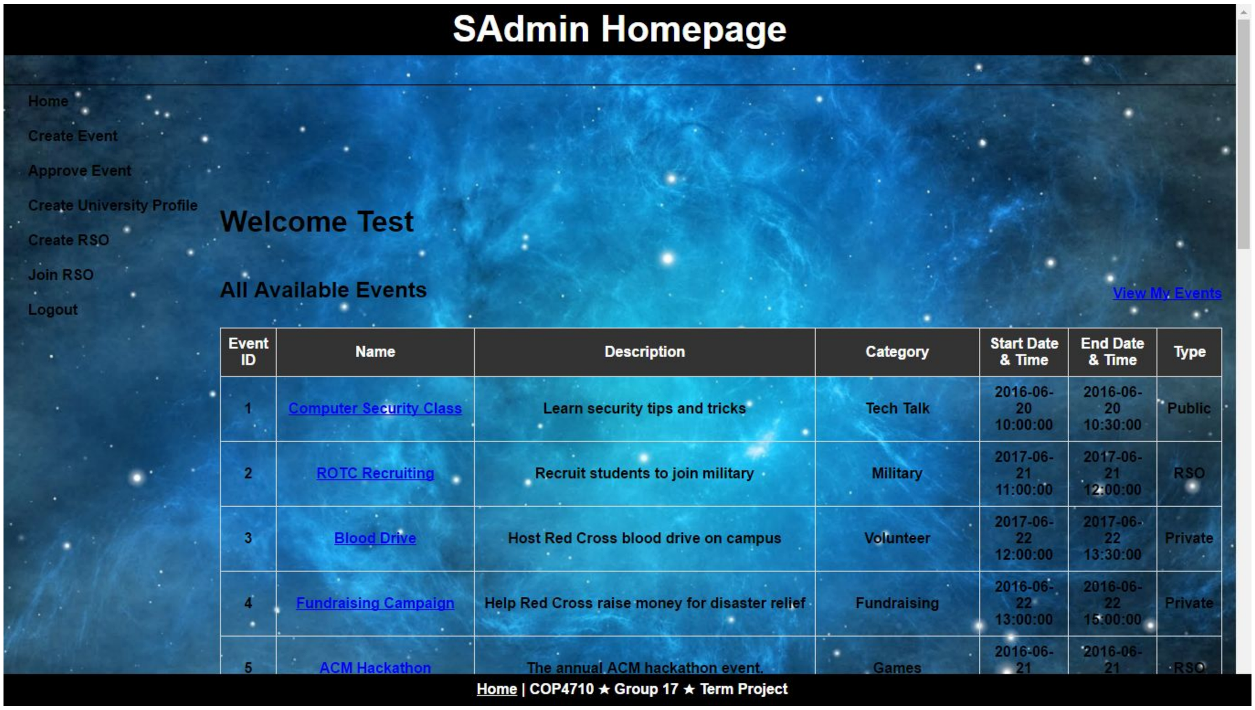
E-mail

Enter E-mail

Password

Enter Password

Homepage:



SAdmin Homepage

Welcome Test

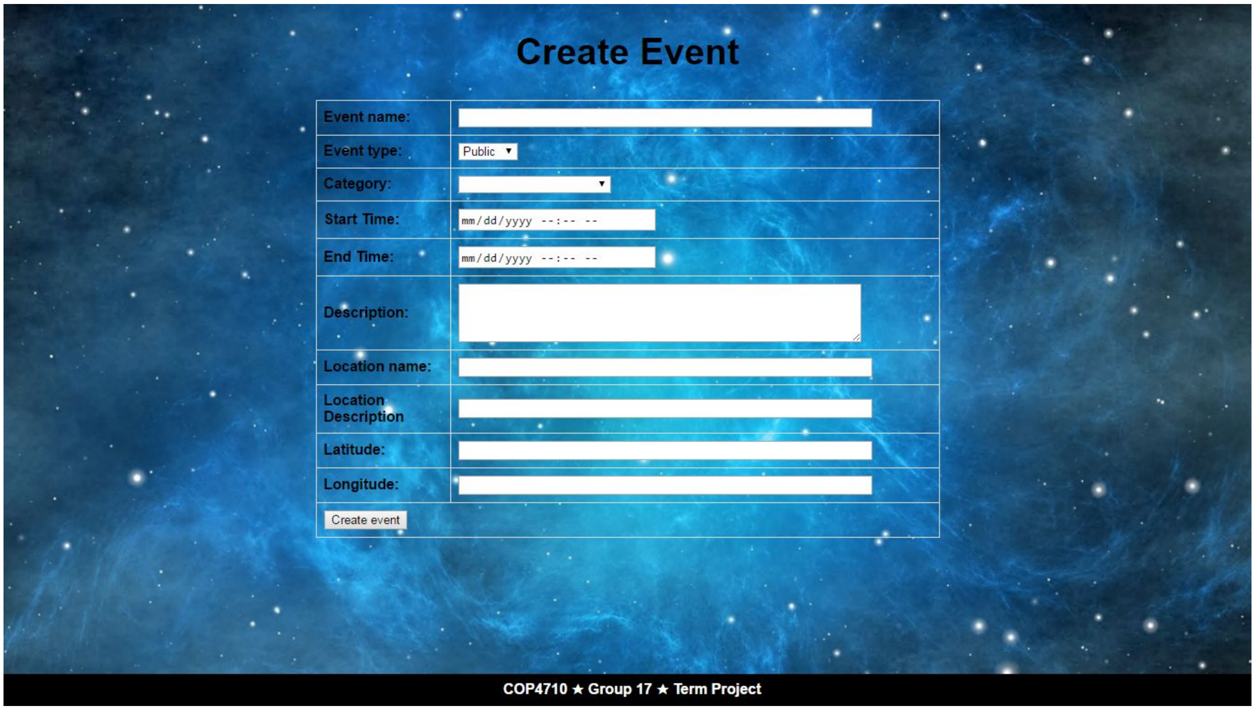
All Available Events

View My Events

Event ID	Name	Description	Category	Start Date & Time	End Date & Time	Type
1	Computer Security Class	Learn security tips and tricks	Tech Talk	2016-06-20 10:00:00	2016-06-20 10:30:00	Public
2	ROTC Recruiting	Recruit students to join military	Military	2017-06-21 11:00:00	2017-06-21 12:00:00	RSO
3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-22 12:00:00	2017-06-22 13:30:00	Private
4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2016-06-22 13:00:00	2016-06-22 15:00:00	Private
5	ACM Hackathon	The annual ACM hackathon event.	Games	2016-06-21	2016-06-21	RSO

Home | COP4710 ★ Group 17 ★ Term Project

Create Event:

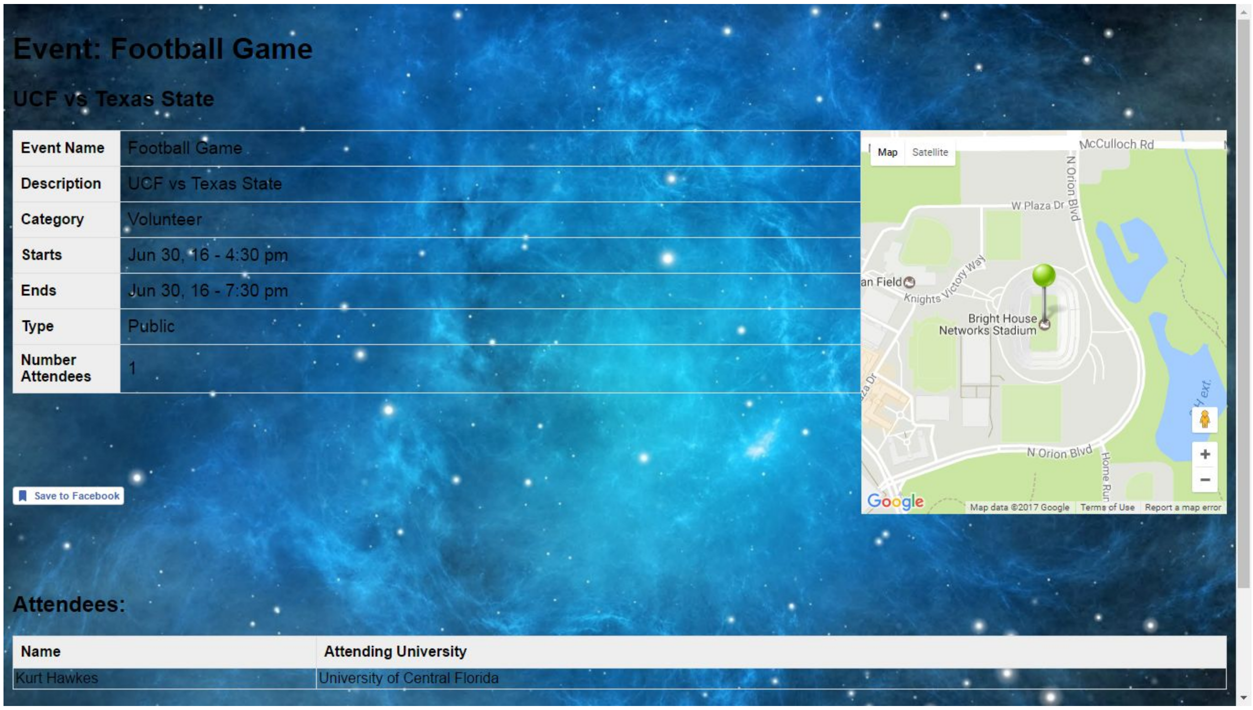


Create Event

Event name:	
Event type:	Public ▾
Category:	▾
Start Time:	mm/dd/yyyy --:-- --
End Time:	mm/dd/yyyy --:-- --
Description:	
Location name:	
Location Description	
Latitude:	
Longitude:	
Create event	

COP4710 ★ Group 17 ★ Term Project

View Event:



Event: Football Game

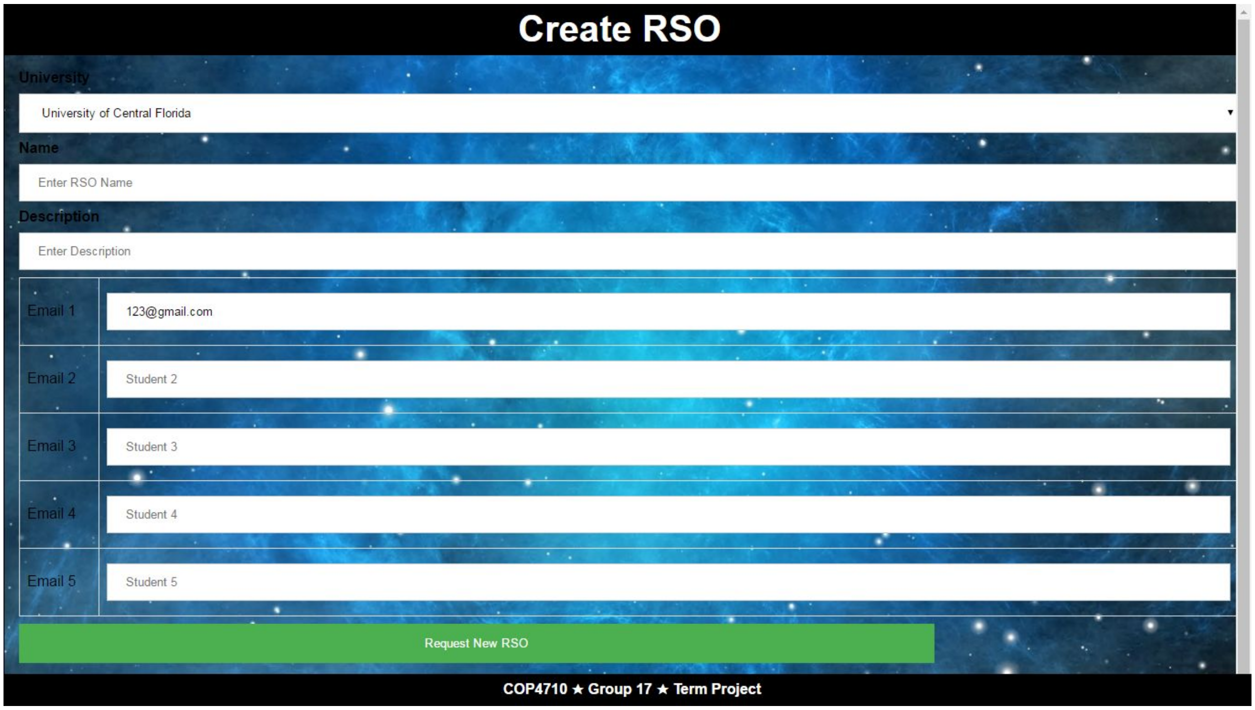
UCF vs Texas State

Event Name	Football Game
Description	UCF vs Texas State
Category	Volunteer
Starts	Jun 30, 16 - 4:30 pm
Ends	Jun 30, 16 - 7:30 pm
Type	Public
Number Attendees	1

Attendees:

Name	Attending University
Kurt Hawkes	University of Central Florida

Create RSO:



Create RSO

University

University of Central Florida ▾

Name

Enter RSO Name

Description

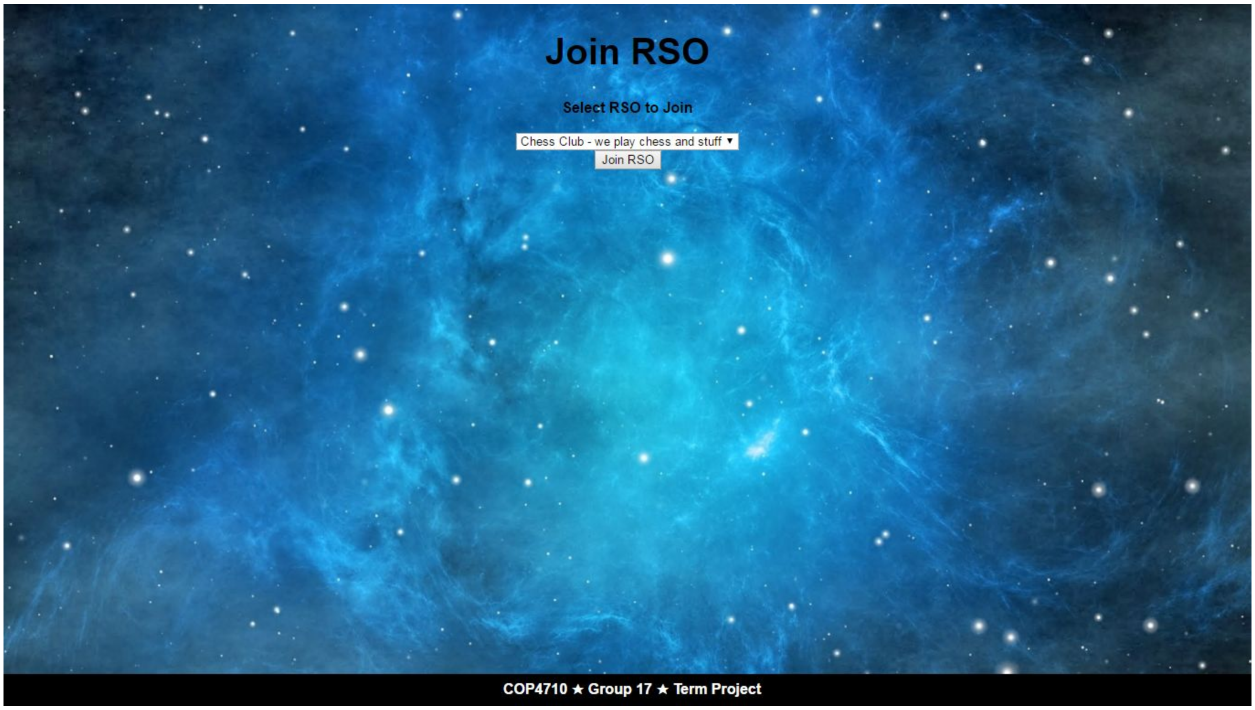
Enter Description

Email 1	123@gmail.com
Email 2	Student 2
Email 3	Student 3
Email 4	Student 4
Email 5	Student 5

Request New RSO

COP4710 ★ Group 17 ★ Term Project

Join RSO:



Join RSO

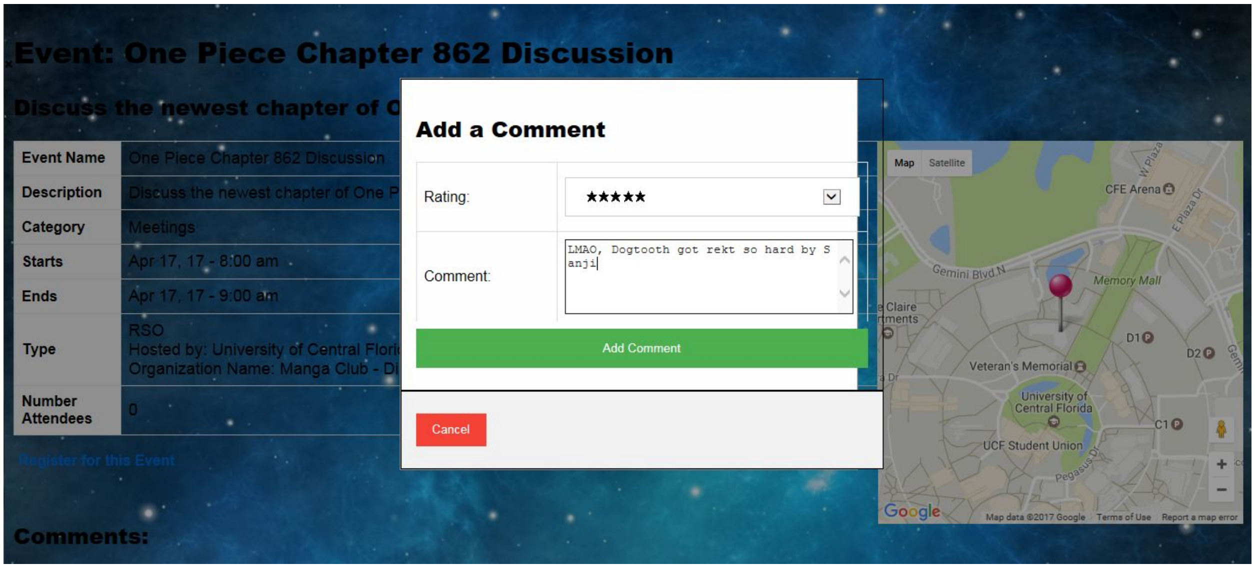
Select RSO to Join

Chess Club - we play chess and stuff ▾

Join RSO

COP4710 ★ Group 17 ★ Term Project

Add Comment:



Event: One Piece Chapter 862 Discussion

Discuss the newest chapter of O

Event Name	One Piece Chapter 862 Discussion
Description	Discuss the newest chapter of One P
Category	Meetings
Starts	Apr 17, 17 - 8:00 am
Ends	Apr 17, 17 - 9:00 am
Type	RSO Hosted by: University of Central Flori Organization Name: Manga Club - Di
Number Attendees	0

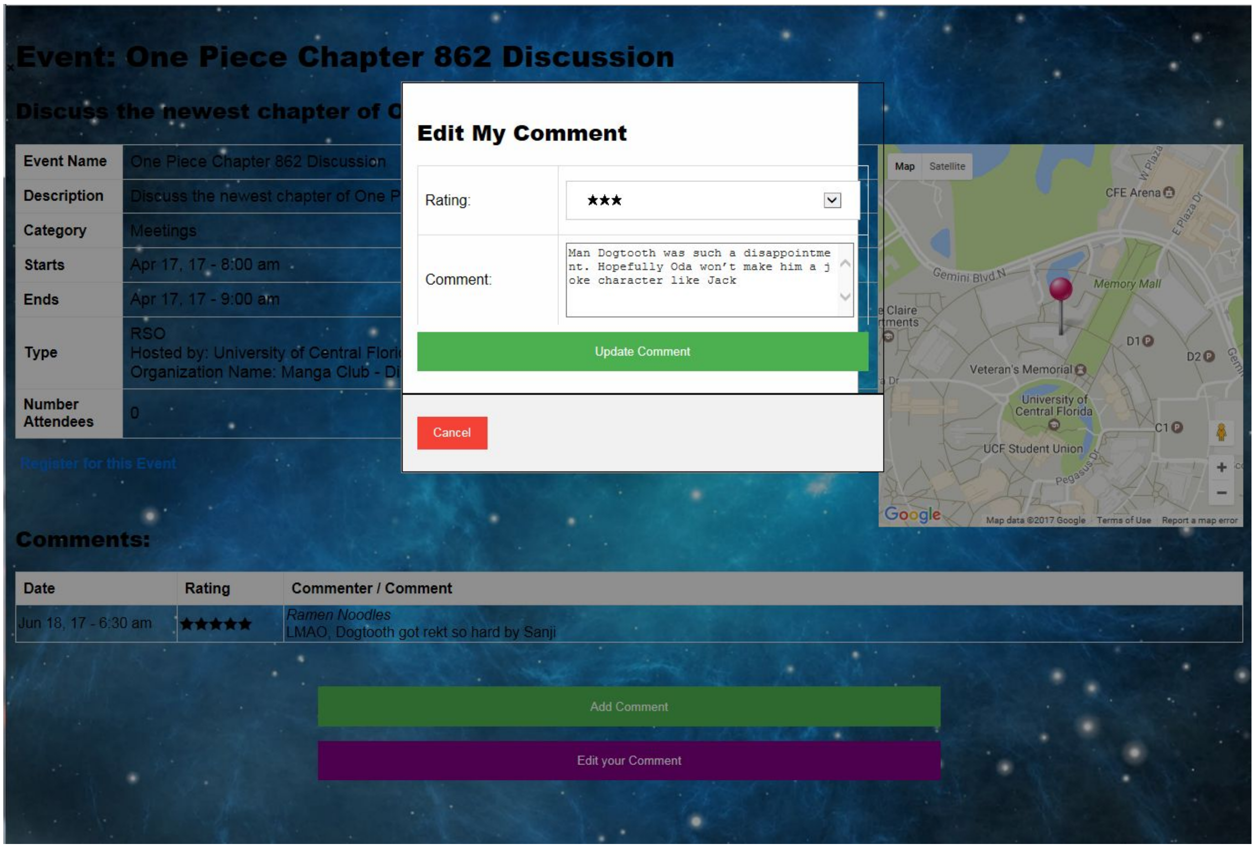
Register for this Event

Add a Comment

Rating:	★★★★★
Comment:	LMAO, Dogtooth got rekt so hard by Sanji

Comments:

Edit Comment:



Event: One Piece Chapter 862 Discussion

Discuss the newest chapter of O

Event Name	One Piece Chapter 862 Discussion
Description	Discuss the newest chapter of One P
Category	Meetings
Starts	Apr 17, 17 - 8:00 am
Ends	Apr 17, 17 - 9:00 am
Type	RSO Hosted by: University of Central Flori Organization Name: Manga Club - Di
Number Attendees	0

Register for this Event

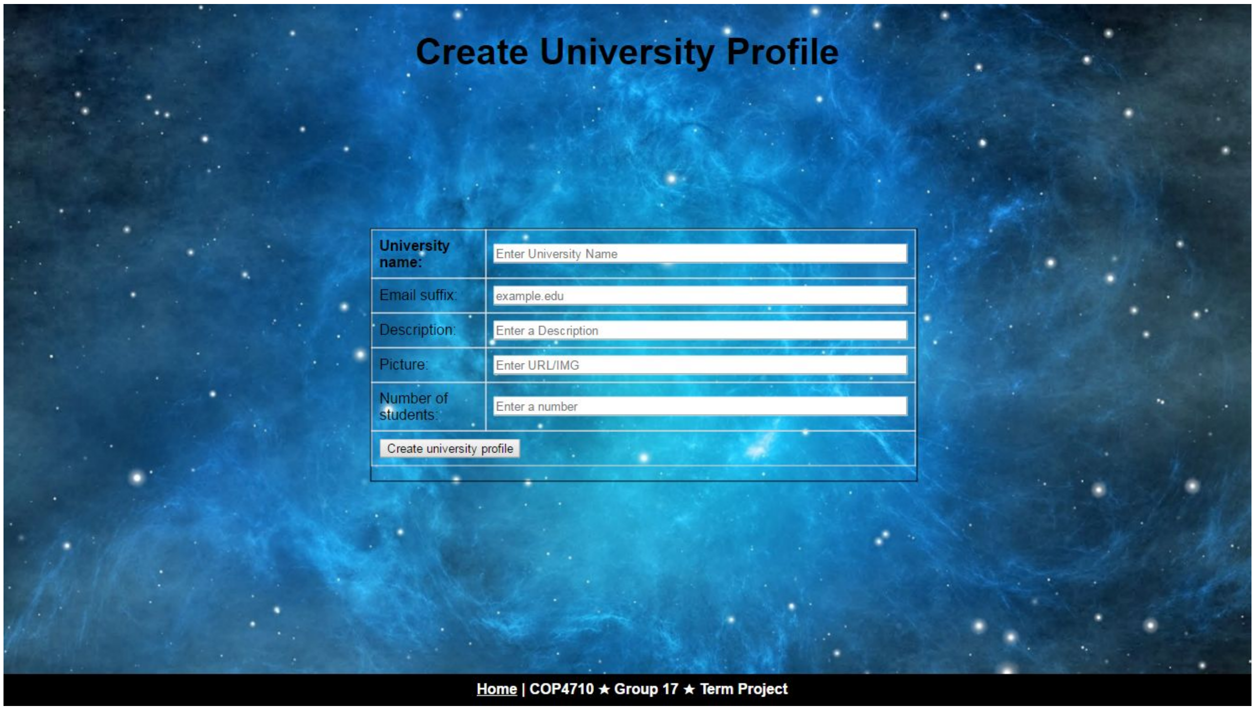
Edit My Comment

Rating:	★★★
Comment:	Man Dogtooth was such a disappointment. Hopefully Oda won't make him a joke character like Jack

Comments:

Date	Rating	Commenter / Comment
Jun 18, 17 - 6:30 am	★★★★★	*Ramen Noodles* LMAO, Dogtooth got rekt so hard by Sanji

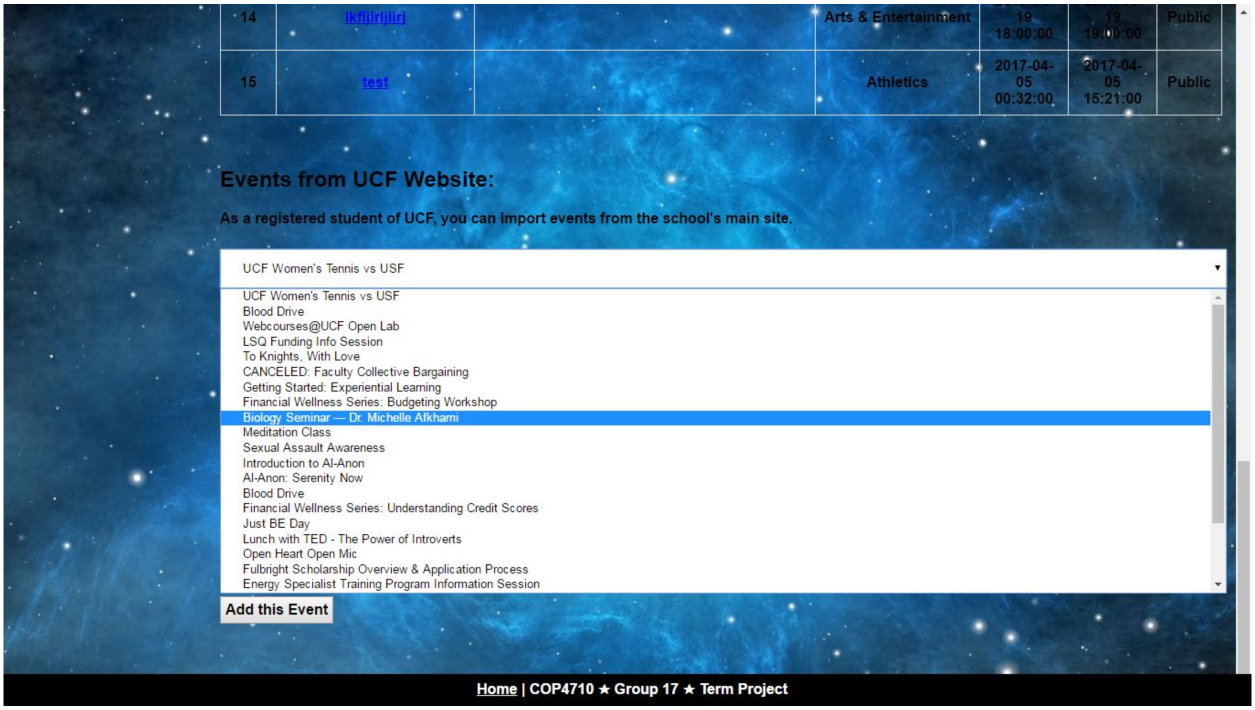
Create University Profile:



Create University Profile

University name:	Enter University Name
Email suffix:	example.edu
Description:	Enter a Description
Picture:	Enter URL/IMG
Number of students:	Enter a number
Create university profile	

Bonus Features (Events RSS Feed):



14	lkfljlrljlirj		Arts & Entertainment	19 18:00:00	19 19:00:00	Public
15	test		Athletics	2017-04-05 00:32:00	2017-04-05 15:21:00	Public

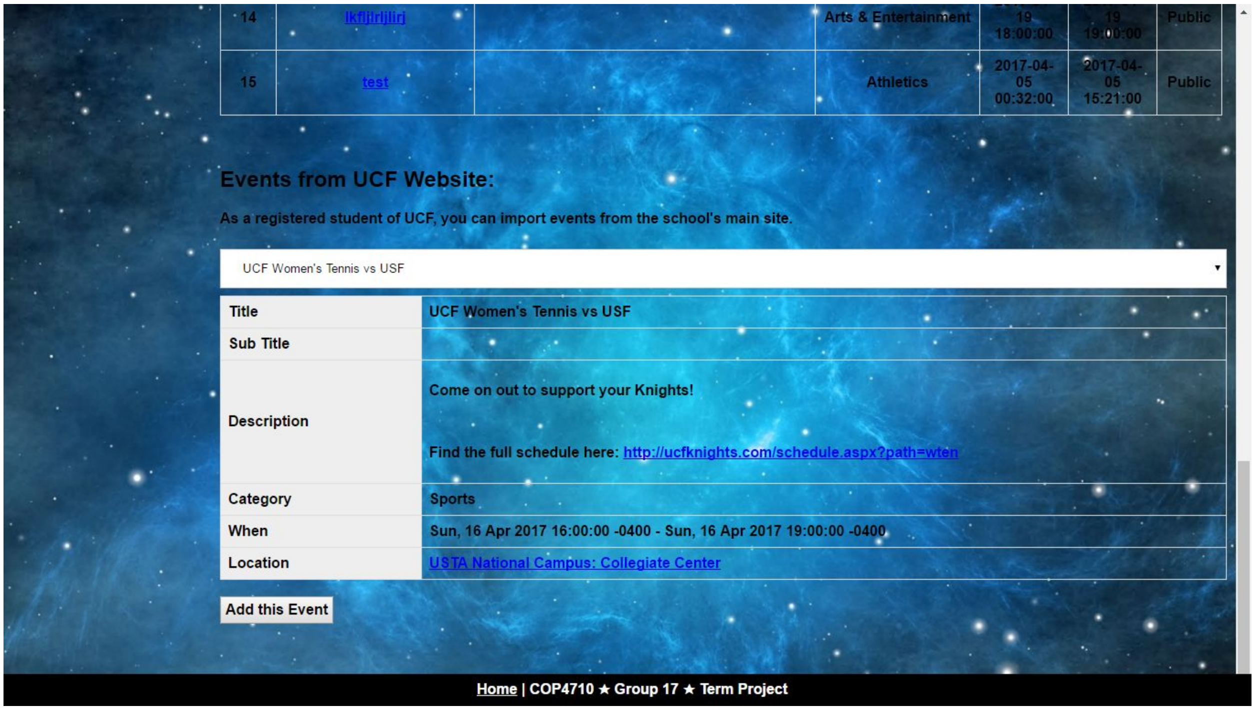
Events from UCF Website:

As a registered student of UCF, you can import events from the school's main site.

UCF Women's Tennis vs USF ▾

- UCF Women's Tennis vs USF
- Blood Drive
- Webcourses@UCF Open Lab
- LSQ Funding Info Session
- To Knights, With Love
- CANCELED: Faculty Collective Bargaining
- Getting Started: Experiential Learning
- Financial Wellness Series: Budgeting Workshop
- Biology Seminar — Dr. Michelle Afkhami
- Meditation Class
- Sexual Assault Awareness
- Introduction to Al-Anon
- Al-Anon: Serenity Now
- Blood Drive
- Financial Wellness Series: Understanding Credit Scores
- Just BE Day
- Lunch with TED - The Power of Introverts
- Open Heart Open Mic
- Fulbright Scholarship Overview & Application Process
- Energy Specialist Training Program Information Session

Home | COP4710 ★ Group 17 ★ Term Project



14	lkfljlrljlirj		Arts & Entertainment	19 18:00:00	19 19:00:00	Public
15	test		Athletics	2017-04-05 00:32:00	2017-04-05 15:21:00	Public

Events from UCF Website:

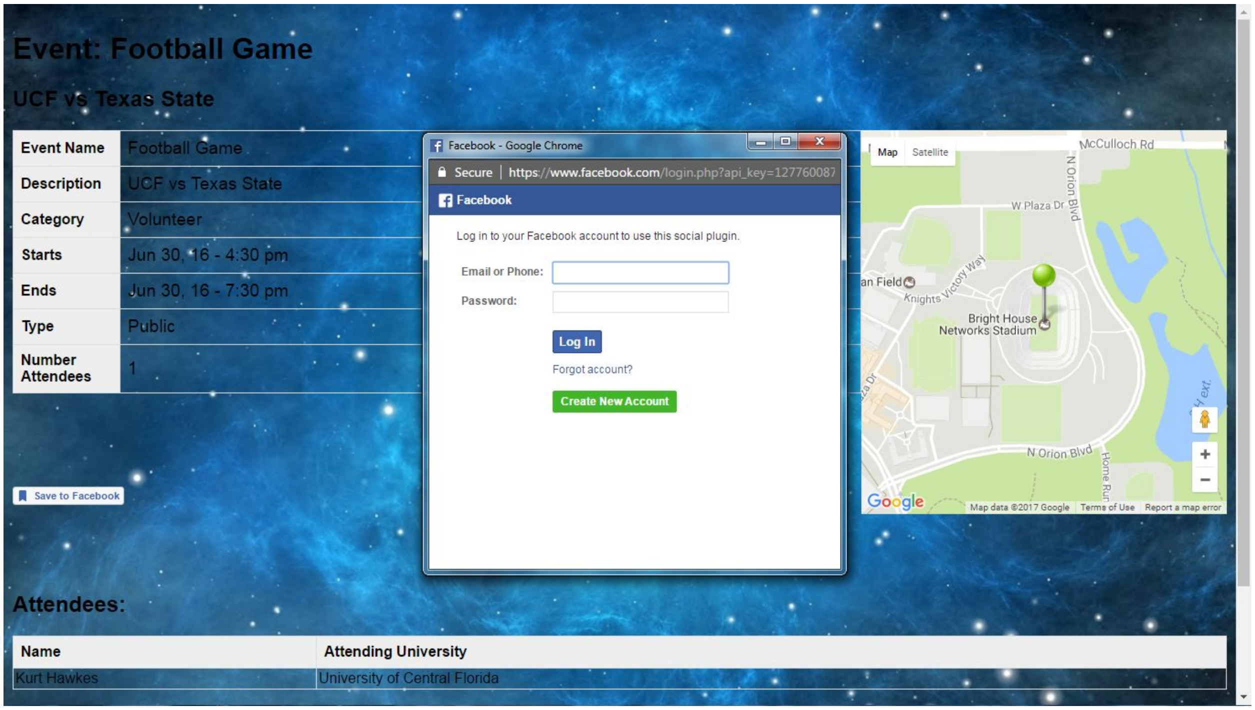
As a registered student of UCF, you can import events from the school's main site.

UCF Women's Tennis vs USF ▾

Title	UCF Women's Tennis vs USF
Sub Title	
Description	Come on out to support your Knights! Find the full schedule here: http://ucfknights.com/schedule.aspx?path=wten
Category	Sports
When	Sun, 16 Apr 2017 16:00:00 -0400 - Sun, 16 Apr 2017 19:00:00 -0400
Location	USTA National Campus: Collegiate Center

Home | COP4710 ★ Group 17 ★ Term Project

Social Media Integration (Facebook):



Event: Football Game

UCF vs Texas State

Event Name	Football Game
Description	UCF vs Texas State
Category	Volunteer
Starts	Jun 30, 16 - 4:30 pm
Ends	Jun 30, 16 - 7:30 pm
Type	Public
Number Attendees	1

Facebook - Google Chrome

Secure | https://www.facebook.com/login.php?api_key=127760087

Facebook

Log in to your Facebook account to use this social plugin.

Email or Phone:

Password:

Log In

Forgot account?

Create New Account

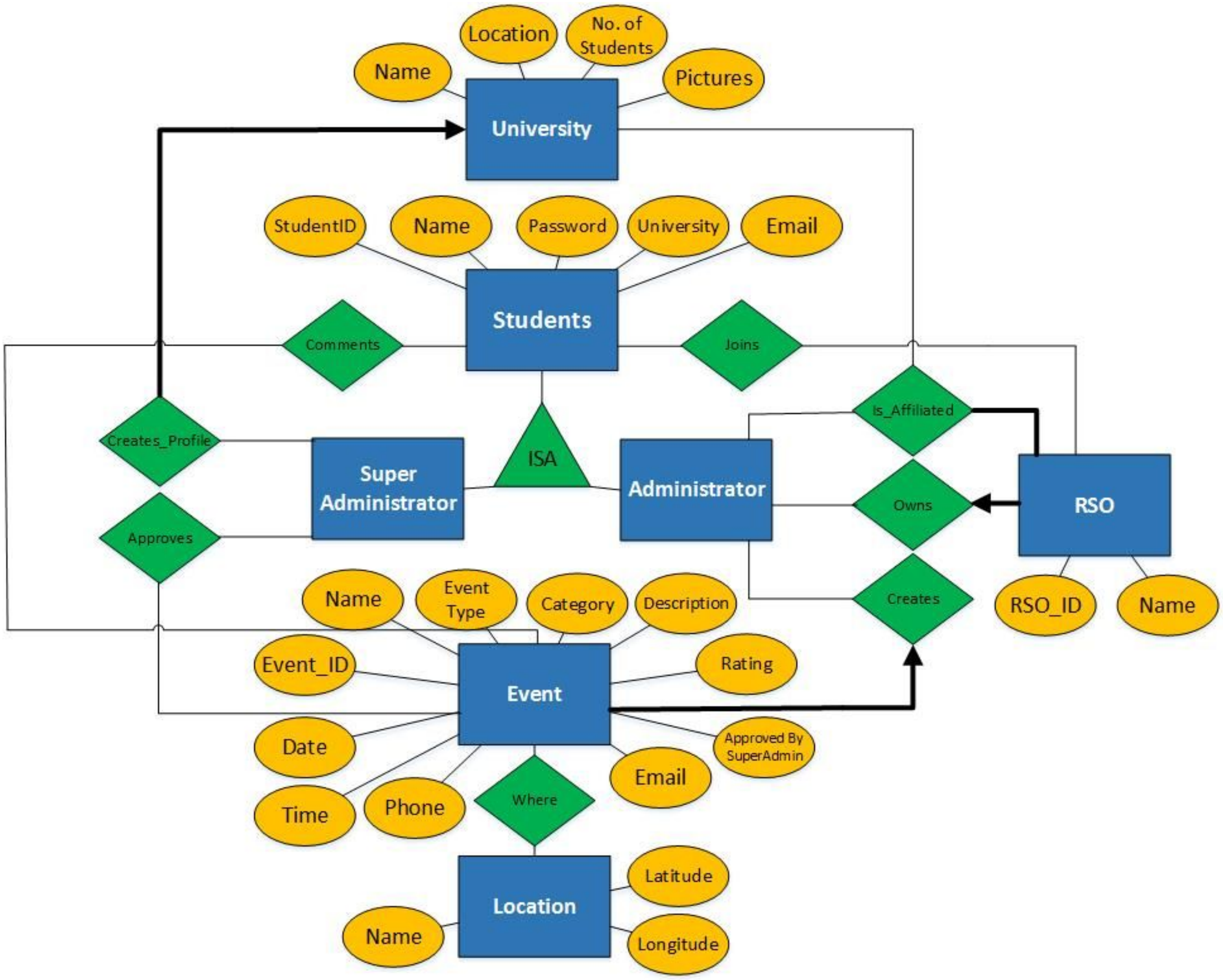
Save to Facebook

Attendees:

Name	Attending University
Kurt Hawkes	University of Central Florida

ER-Model:

ER Diagram:



The ER Diagram above is a prototype the group brainstormed together as the first step towards the project. It helped immensely when creating our relational schema for the database and we used it to keep track of what was needed and how we could enforce some constraints through the DBMS. Some features above were changed slightly or optimized through code within the application to remove redundant tables or tables and views were added to implement extra features that we were not able to enforce via code for the application.

Constraints:

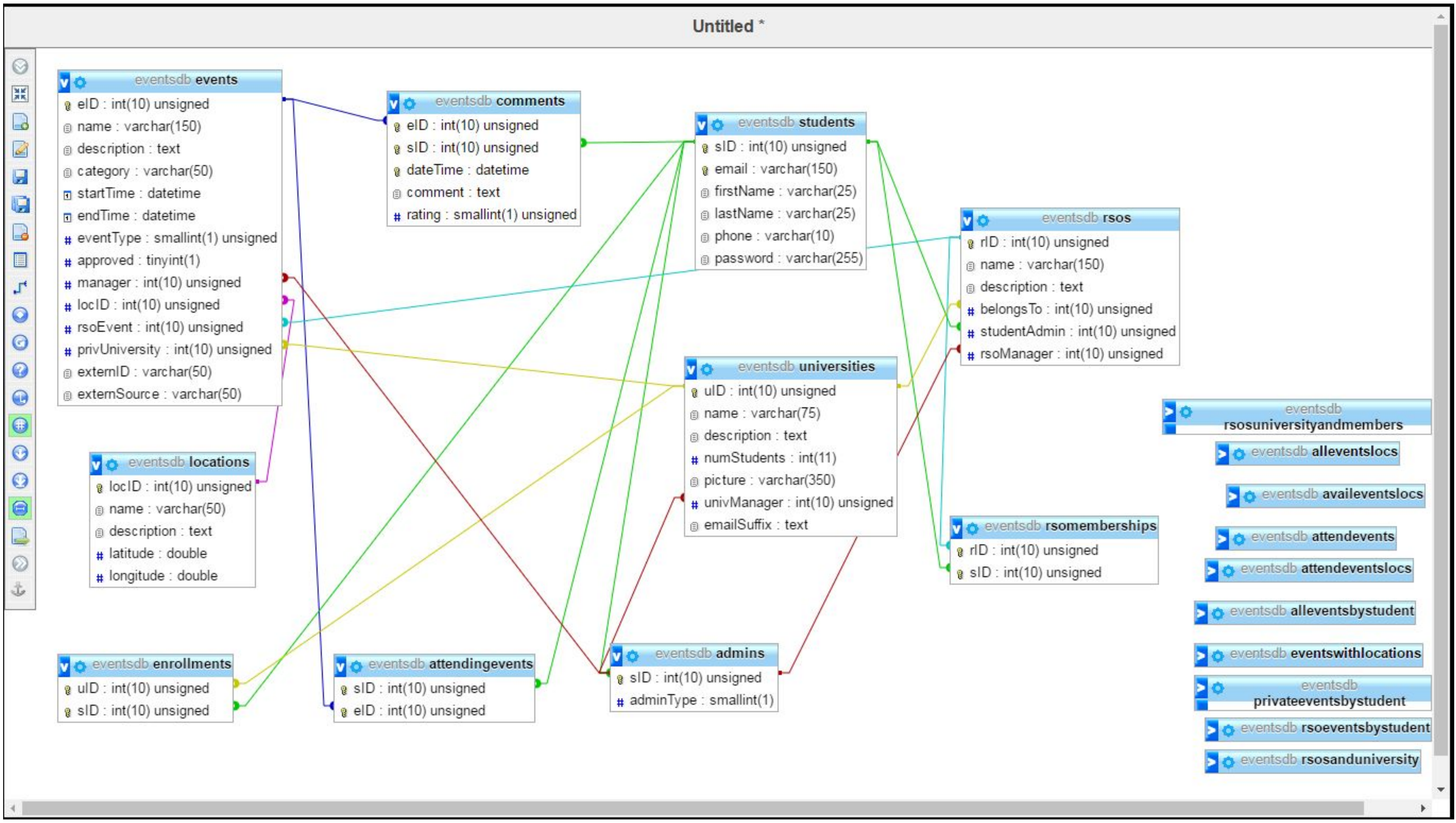
Constraints captured by the ER-model/DBMS are:

- No overlapping events, times and same location, is enforced with a trigger that checks the location ID/Name and then checks multiple statements based on start time and end time.
- User types (Students, Admins, SuperAdmins) are enforced with an ISA and identifying the type of admin a user is through an adminType attribute, queries on specific users are done through the attribute SID.
- Event types (Public, Private, RSO) are enforced with an attribute Event_Type to identify the type of event.
- Events shown have to be approved by a super admin which is checked through the attribute App_by_SA, NULL for N/A and 1 if approved.
- SuperAdmins are the only ones who can create a University Profile on the website.
- Admins or SuperAdmins are the only ones who can create Events which is enforced by checking the SID in the admin table.
- Students who create an RSO is upgraded to an Admin (of that RSO).
- Admin/Students can view all Public events, Private events (of their university), and RSO events (of their affiliated RSOs). Enforced with a view in the DBMS.
- SuperAdmins can view all events. Enforced with a view in the DBMS.
- Email attribute for all users are UNIQUE, since it is used for the login feature.
- All Users (Students, Admins, SuperAdmins) can join an RSO (of their University)

Other constraints not enforced by the DBMS, but enforced through the application's code are:

- Comments can be made by any user on an Event, but limited to 1 and users can edit their comments. The comments made are time-stamped by checking the logged in user's system time and this is stored within the database for the comments table. Super Admins can edit and delete other user's comments. These are all are enforced through code written in the application.
- Ratings are set between 0 and 5 (handled by a trigger). Users can rate an event along with their comments. Enforced via code in the App.
- Password attribute for all Users are hashed (using MD5) and salted which is handled inside the application's code.
- Creation of a new RSO requires the User creating it and 4 other students minimum (5+ students total) and all of these students must have the same e-mail suffix of that university to be accepted. This is handled through the application's code.

Relational Data Model:



Students Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Students` (
  `sID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(150) NOT NULL,
  `firstName` VARCHAR(25) NULL,
  `lastName` VARCHAR(25) NULL,
  `phone` VARCHAR(10) NULL,
  `password` VARCHAR(255) NOT NULL,
  PRIMARY KEY (`sID`),
  UNIQUE INDEX `Stud_email_UNIQUE` (`email` ASC))
```

Admin Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Admins` (
  `sID` INT UNSIGNED NOT NULL,
  `adminType` SMALLINT(1) NOT NULL,
  PRIMARY KEY (`sID`),
  CONSTRAINT `Admins_sID`
          FOREIGN KEY (`sID`)
          REFERENCES `EventsDB`.`Students` (`sID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION)
```

University Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Universities` (
  `uID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(75) NULL,
  `description` TEXT(500) NULL,
  `numStudents` INT NULL,
  `picture` VARCHAR(350) NULL,
  `aID` INT UNSIGNED NOT NULL,
  `univManager` INT UNSIGNED NOT NULL,
  `emailSuffix` TEXT(125) NOT NULL,
  PRIMARY KEY (`uID`),
  INDEX `Univ_aID_idx` (`aID` ASC),
  INDEX `Univ_sID_idx` (`univManager` ASC),
  CONSTRAINT `Univ_sID`
          FOREIGN KEY (`univManager`)
          REFERENCES `EventsDB`.`Admins` (`sID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION,
  CONSTRAINT `Univ_aID`
          FOREIGN KEY (`aID`)
          REFERENCES `EventsDB`.`Addresses` (`aID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION)
```

Locations Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Locations` (
  `locID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NULL,
  `description` TEXT(500) NULL,
  `latitude` REAL NULL,
  `longitude` REAL NULL,
  PRIMARY KEY (`locID`))
```

RSO Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`RSOs` (
  `rID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(150) NULL,
  `description` TEXT(500) NULL,
  `belongsTo` INT UNSIGNED NOT NULL,
  `studentAdmin` INT UNSIGNED NOT NULL,
  `rsoManager` INT UNSIGNED,
  PRIMARY KEY (`rID`),
  INDEX `RSOs_uID_idx` (`belongsTo` ASC),
  INDEX `RSOs_admin_sID_idx` (`rsoManager` ASC),
  INDEX `RSOs_sID_idx` (`studentAdmin` ASC),
  CONSTRAINT `RSOs_uID`
        FOREIGN KEY (`belongsTo`)
        REFERENCES `EventsDB`.`Universities` (`uID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `RSOs_admin_sID`
        FOREIGN KEY (`rsoManager`)
        REFERENCES `EventsDB`.`Admins` (`sID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `RSOs_sID`
        FOREIGN KEY (`studentAdmin`)
        REFERENCES `EventsDB`.`Students` (`sID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
```

Events Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Events` (
  `eID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(150) NOT NULL,
  `description` TEXT NULL,
  `category` VARCHAR(50) NULL,
  `startTime` DATETIME NOT NULL,
  `endTime` DATETIME NOT NULL,
  `eventType` SMALLINT(1) UNSIGNED NULL,
  `approved` TINYINT(1) NOT NULL DEFAULT 0,
  `manager` INT UNSIGNED NOT NULL,
  `locID` INT UNSIGNED NOT NULL,
  `rsoEvent` INT UNSIGNED NULL,
  `privUniversity` INT UNSIGNED NULL,
  `externID` VARCHAR(50),
  `externSource` VARCHAR(50),
  PRIMARY KEY (`eID`),
  INDEX `Events_locID_idx` (`locID` ASC),
  INDEX `Events_sID_idx` (`manager` ASC),
  INDEX `Events_rID_idx` (`rsoEvent` ASC),
  INDEX `Events_uID_idx` (`privUniversity` ASC),
  CONSTRAINT `Events_locID`
        FOREIGN KEY (`locID`)
        REFERENCES `EventsDB`.`Locations` (`locID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Events_sID`
        FOREIGN KEY (`manager`)
        REFERENCES `EventsDB`.`Admins` (`sID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Events_rID`
        FOREIGN KEY (`rsoEvent`)
        REFERENCES `EventsDB`.`RSOs` (`rID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Events_uID`
        FOREIGN KEY (`privUniversity`)
        REFERENCES `EventsDB`.`Universities` (`uID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
```

Comments Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Comments` (
  `eID` INT UNSIGNED NOT NULL,
  `sID` INT UNSIGNED NOT NULL,
  `dateTime` DATETIME NOT NULL,
  `comment` TEXT(500) NULL,
  `rating` SMALLINT(1) UNSIGNED NULL,
  INDEX `Cmnt_sID_idx` (`sID` ASC),
  INDEX `Cmnt_eID_idx` (`eID` ASC),
  PRIMARY KEY (`eID`, `sID`, `dateTime`),
  CONSTRAINT `Cmnt_sID`
        FOREIGN KEY (`sID`)
        REFERENCES `EventsDB`.`Students` (`sID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Cmnt_eID`
        FOREIGN KEY (`eID`)
        REFERENCES `EventsDB`.`Events` (`eID`)
```

```
          ON DELETE NO ACTION
          ON UPDATE NO ACTION)
```

RSOMemberships Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`RSOMemberships` (
  `rID` INT UNSIGNED NOT NULL,
  `sID` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`rID`, `sID`),
  INDEX `RSOMemb_rID_idx` (`rID` ASC),
  INDEX `RSOMemb_sID_idx` (`sID` ASC),
  CONSTRAINT `RSOMemb_sID`
          FOREIGN KEY (`sID`)
          REFERENCES `EventsDB`.`Students` (`sID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION,
  CONSTRAINT `RSOMemb_rID`
          FOREIGN KEY (`rID`)
          REFERENCES `EventsDB`.`RSOs` (`rID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION)
```

AttendingEvents Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`AttendingEvents` (
  `sID` INT UNSIGNED NOT NULL,
  `eID` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`sID`, `eID`),
  INDEX `AttdEvent_eID_idx` (`eID` ASC),
  INDEX `AttdEvent_sID_idx` (`sID` ASC),
  CONSTRAINT `AttdEvent_sID`
          FOREIGN KEY (`sID`)
          REFERENCES `EventsDB`.`Students` (`sID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION,
  CONSTRAINT `AttdEvent_eID`
          FOREIGN KEY (`eID`)
          REFERENCES `EventsDB`.`Events` (`eID`)
          ON DELETE NO ACTION
          ON UPDATE NO ACTION)
```

Enrollment Table:

```
CREATE TABLE IF NOT EXISTS `EventsDB`.`Enrollments` (
  `uID` INT UNSIGNED NOT NULL,
  `sID` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`uID`, `sID`),
  INDEX `Enroll_sID_idx` (`sID` ASC),
  INDEX `Enroll_uID_idx` (`uID` ASC),
  CONSTRAINT `Enroll_sID`
        FOREIGN KEY (`sID`)
        REFERENCES `EventsDB`.`Students` (`sID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `Enroll_uID`
        FOREIGN KEY (`uID`)
        REFERENCES `EventsDB`.`Universities` (`uID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
```

__No Overlapping Events Trigger:__

```
USE `EventsDB`$$
DROP TRIGGER IF EXISTS `EventsDB`.`Events_BEFORE_INSERT` $$
USE `EventsDB`$$
CREATE DEFINER = CURRENT_USER TRIGGER `EventsDB`.`Events_BEFORE_INSERT`
BEFORE INSERT ON `Events` FOR EACH ROW
BEGIN
        DECLARE msg VARCHAR(128);
        DECLARE existing INT;
        SET existing = (
  SELECT eID FROM Events E
   WHERE locID = NEW.locID AND (
        (E.startTime >= NEW.startTime AND E.startTime <= NEW.endTime) OR
   (E.endTime   >= NEW.startTime AND E.endTime   <= NEW.endTime) OR
   (E.startTime >= NEW.startTime AND E.endTime   <= NEW.endTime) OR
   (E.startTime <= NEW.startTime AND E.endTime   >= NEW.endTime))
        );
        IF (existing IS NOT NULL) THEN
        SET msg = 'Events Error: Event overlaps with another event for the same timeframe / location.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
        END IF;
        IF ((NEW.eventType = 2) AND (NEW.privUniversity IS NULL)) THEN
  SET msg = 'Events Error: Private events must have an associated University.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
 END IF;
        IF ((NEW.eventType = 3) AND (NEW.rsoEvent IS NULL)) THEN
  SET msg = 'Events Error: Private events must have an associated University.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
 END IF;
END$$

USE `EventsDB`$$
DROP TRIGGER IF EXISTS `EventsDB`.`Events_BEFORE_UPDATE` $$
USE `EventsDB`$$
CREATE DEFINER = CURRENT_USER TRIGGER `EventsDB`.`Events_BEFORE_UPDATE`
BEFORE UPDATE ON `Events` FOR EACH ROW
BEGIN
        DECLARE msg VARCHAR(128);
        DECLARE existing INT;
        SET existing = (
  SELECT eID FROM Events E
   WHERE locID = NEW.locID AND (
        (E.startTime >= NEW.startTime AND E.startTime <= NEW.endTime) OR
   (E.endTime   >= NEW.startTime AND E.endTime   <= NEW.endTime) OR
   (E.startTime >= NEW.startTime AND E.endTime   <= NEW.endTime) OR
   (E.startTime <= NEW.startTime AND E.endTime   >= NEW.endTime))
        );
        IF (existing IS NOT NULL) THEN
        SET msg = 'Events Error: Event overlaps with another event for the same timeframe /
location.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
        END IF;
        IF ((NEW.eventType = 2) AND (NEW.privUniversity IS NULL)) THEN
  SET msg = 'Events Error: Private events must have an associated University.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
 END IF;
        IF ((NEW.eventType = 3) AND (NEW.rsoEvent IS NULL)) THEN
  SET msg = 'Events Error: Private events must have an associated University.';
        SIGNAL SQLSTATE '45000' SET message_text = msg;
 END IF;
END$$
```

Comments/Rating Trigger:

```
USE `EventsDB`$$
DROP TRIGGER IF EXISTS `EventsDB`.`Comments_BEFORE_INSERT` $$
USE `EventsDB`$$
CREATE DEFINER = CURRENT_USER TRIGGER
`EventsDB`.`Comments_BEFORE_INSERT` BEFORE INSERT ON `Comments` FOR EACH
ROW
BEGIN
        DECLARE msg VARCHAR(128);
```

```
    IF (NEW.rating < 0 OR NEW.rating > 5) then
    SET msg = concat('Events Error: Ratings must be between 0-5 inclusive: rating=',
CAST(NEW.rating AS CHAR(5)));
    SIGNAL SQLSTATE '45000' SET message_text = msg;
    END IF;
END$$

USE `EventsDB`$$
DROP TRIGGER IF EXISTS `EventsDB`.`Comments_BEFORE_UPDATE` $$
USE `EventsDB`$$
CREATE DEFINER = CURRENT_USER TRIGGER
`EventsDB`.`Comments_BEFORE_UPDATE` BEFORE UPDATE ON `Comments` FOR EACH
ROW
BEGIN
    DECLARE msg VARCHAR(128);
    IF (NEW.rating < 0 OR NEW.rating > 5) then
    SET msg = concat('Events Error: Ratings must be between 0-5 inclusive: rating=',
CAST(NEW.rating AS CHAR(5)));
    SIGNAL SQLSTATE '45000' SET message_text = msg;
    END IF;
END$$
```

Events certain Users can see Views:

```
CREATE OR REPLACE VIEW PrivateEventsByStudent AS
    SELECT S.sID, Evt.* FROM Students S INNER JOIN Enrollments E
    ON S.sID = E.sID INNER JOIN Events Evt ON E.uID = Evt.privUniversity;

CREATE OR REPLACE VIEW RSOEventsByStudent AS
    SELECT S.sID, Evt.* FROM Students S INNER JOIN RSOMemberships R
    ON S.sID = R.sID INNER JOIN Events Evt ON R.rID = Evt.rsoEvent;

CREATE OR REPLACE VIEW AllEventsByStudent AS
    SELECT * FROM PrivateEventsByStudent UNION
    SELECT * FROM RSOEventsByStudent UNION
    SELECT 0 as sID, Evt.* FROM Events Evt WHERE eventType=1;
```

Event Location View:

```
CREATE OR REPLACE VIEW EventsWithLocations AS
    SELECT Evt.*, L.name as locName, L.description as locDescription, L.latitude,
        L.longitude
    FROM Events Evt INNER JOIN Locations L ON Evt.locID = L.locID;
```

Which Users are attending this Event View:

```
CREATE OR REPLACE VIEW AttendEvents AS
        SELECT E.*, sID
        FROM Events E
        INNER JOIN AttendingEvents A ON A.eID = E.eID;
```

Which University an RSO is affiliated with View:

```
CREATE OR REPLACE VIEW RSOsAndUniversity AS
        SELECT R.rID, R.name, R.description, R.belongsTo, R.studentAdmin, R.rsoManager,
                U.name as universityName
        FROM RSOs R
        INNER JOIN Universities U ON U.uid=R.belongsTo;
```

All members inside of an RSO View:

```
CREATE OR REPLACE VIEW RSOsUniversityAndMembers AS
        SELECT R.rID, R.name, R.description, R.belongsTo, R.studentAdmin, R.rsoManager,
                U.name as universityName, sID
        FROM RSOs R
        INNER JOIN Universities U ON U.uid=R.belongsTo
        INNER JOIN RSOMemberships M ON M.rID=R.rID;
```

Demo:

Universities Table:

```
INSERT INTO Universities (name, description, numStudents, univManager, emailSuffix)
VALUES ('University of Central Florida','Orlando Campus',60767, 1, 2, 'knights.ucf.edu');
INSERT INTO Universities (name, description, numStudents, univManager, emailSuffix)
VALUES ('University of Florida','Gainsville Campus',51725, 2, 3, 'uf.edu');
INSERT INTO Universities (name, description, numStudents, univManager, emailSuffix)
VALUES ('University of South Florida','Tampa Campus',48793, 3, 4, 'usf.edu');
INSERT INTO Universities (name, description, numStudents, univManager, emailSuffix)
VALUES ('University of Miami','Miami Campus',16774, 4, 5, 'um.edu');
```

Students Table:

```
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('admin', 'admin', '4075551212', 'admin@eventsdb.ucf.edu',
'$2y$10$1iCCLJYocWro2xa8bYKvieIUVlV.tOT8dPf.Frk3BK9GAqI/R79oS');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('UCF', 'Admin', '4075551234', 'admin@knights.ucf.edu',
'$2y$10$onfHVh1gGa/Lb6JIS5SKO.8ZvheV9Y/oHU8cpnYdy8GG0dOKWHqmi');
```

```
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('UF', 'Admin', '4075551234', 'admin@uf.edu',
'$2y$10$JdMRvA.rsnSgSR1iMHSuO.YbD16EZ0Th8lfS8CUK9CjluMHkChVHe');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('USF', 'Admin', '4075551234', 'admin@usf.edu',
'$2y$10$hhw/SIfC31Auym0GCdup1.hkKv2CD/vK1vjCJWsBOTC6pdYlo.W.S');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('UM', 'Admin', '4075551234', 'admin@um.edu',
'$2y$10$OKEh0JhnFBiUaAgo9KzhT.w.Z2RIdAyuOpbZ6TYRRrEvjkIEPy8Qm');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Daniel', 'Truong', '4078230001', 'dtruong5@knights.ucf.edu',
'$2y$10$onfHVh1gGa/Lb6JIS5SKO.8ZvheV9Y/oHU8cpnYdy8GG0dOKWHqmi');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Camilo', 'Junoy', '4078230002', 'cj@knights.ucf.edu',
'$2y$10$JdMRvA.rsnSgSR1iMHSuO.YbD16EZ0Th8lfS8CUK9CjluMHkChVHe');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Brian', 'Nguyen', '4078230003', 'bnuygen@knights.ucf.edu',
'$2y$10$hhw/SIfC31Auym0GCdup1.hkKv2CD/vK1vjCJWsBOTC6pdYlo.W.S');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('William', 'Sawran', '4078230004', 'swill@knights.ucf.edu',
'$2y$10$OKEh0JhnFBiUaAgo9KzhT.w.Z2RIdAyuOpbZ6TYRRrEvjkIEPy8Qm');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Test', 'Test', '4078230001', 'test@uf.edu',
'$2y$10$onfHVh1gGa/Lb6JIS5SKO.8ZvheV9Y/oHU8cpnYdy8GG0dOKWHqmi');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Hello', 'Goodbye', '4078230002', 'sayonara@uf.edu',
'$2y$10$JdMRvA.rsnSgSR1iMHSuO.YbD16EZ0Th8lfS8CUK9CjluMHkChVHe');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('John', 'Doe', '4078230003', 'john.doe@usf.edu',
'$2y$10$hhw/SIfC31Auym0GCdup1.hkKv2CD/vK1vjCJWsBOTC6pdYlo.W.S');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Jane', 'Doe', '4078230004', 'jane.doe@um.edu',
'$2y$10$OKEh0JhnFBiUaAgo9KzhT.w.Z2RIdAyuOpbZ6TYRRrEvjkIEPy8Qm');
INSERT INTO Students (firstName, lastName, phone, email, password) VALUES
        ('Ramen', 'Noodles', '4078230005', 'rnoodles@knights.ucf.edu',
'$2y$10$onfHVh1gGa/Lb6JIS5SKO.8ZvheV9Y/oHU8cpnYdy8GG0dOKWHqmi');
```

RSOs Table:

```
INSERT INTO RSOs (name, description, belongsTo, rsoManager, studentAdmin) VALUES
        ('Hack UCF', 'Learn cyber security', 1, 2, 6);
INSERT INTO RSOs (name, description, belongsTo, rsoManager, studentAdmin) VALUES
        ('Chess Club', 'Have fun and learn how to play Chess with others', 2, 3, 10);
```

```
INSERT INTO RSOs (name, description, belongsTo, rsoManager, studentAdmin) VALUES
        ('Anime Club', 'Watch and Discuss anime with others', 3, 4, 12);
INSERT INTO RSOs (name, description, belongsTo, rsoManager, studentAdmin) VALUES
        ('SMASH Club', 'Play Smash (Melee, Brawl, Smash 4) with other students and have fun',
4, 5, 13);
```

Events Table:

```
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID) VALUES
        ('Computer Security Class', 'Learn security tips and tricks', 'Tech Talk', '2017-06-20
10:00:00', '2017-06-20 10:30:00', 1, 2, 5);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID, rsoEvent) VALUES
        ('ROTC Recruiting', 'Recruit students to join military', 'Military', '2017-06-21 11:00:00',
'2017-06-21 12:00:00', 0, 3, 6, 2);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID, privUniversity) VALUES
        ('Blood Drive', 'Host Red Cross blood drive on campus', 'Volunteer', '2017-06-22
12:00:00', '2017-06-22 13:30:00', 2, 2, 1, 1);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID, privUniversity) VALUES
        ('Fundraising Campaign', 'Help Red Cross raise money for disaster relief', 'Fundraising',
'2017-06-22 13:00:00', '2017-06-22 15:00:00', 2, 3, 2, 2);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID, rsoEvent) VALUES
        ('ACM Hackathon', 'The annual ACM hackathon event.', 'Games', '2017-06-21 11:00:00',
'2017-06-21 12:00:00', 0, 5, 4, 4);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID) VALUES
        ('Chess Tournament', 'Friendly Chess tournament come learn or compete in high level
play', 'Games', '2017-07-01 18:00:00', '2017-07-01 20:00:00', 1, 14, 8);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID, privUniversity) VALUES
        ('Smash Tournament', 'Friendly Locals Smash Tournament', 'Volunteer', '2017-06-29
17:00:00', '2017-06-29 18:00:00', 2, 14, 7, 1);
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager,
locID) VALUES
        ('Football Game', 'UCF vs Texas State', 'Sports', '2017-06-30 16:30:00', '2017-06-30
19:30:00', 1, 14, 9);
```

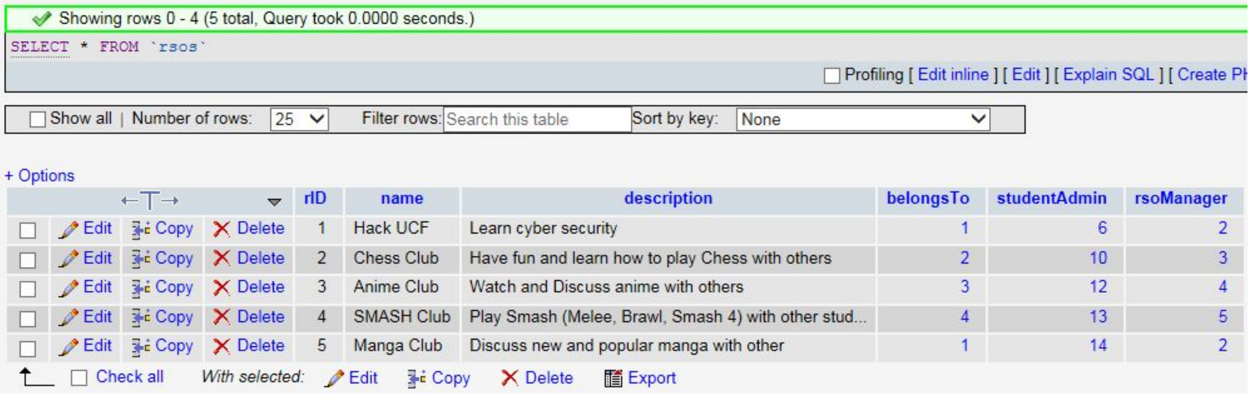
Comments Table:

```
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (1, 8, '2017-06-10
11:11:11', 'The seminar was well done. Learned interesting new security techniques.', 5);
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (1, 9, '2017-06-11
11:11:11', 'Several good security tips shared.', 4);
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (2, 11, '2017-06-12
11:11:11', 'Very informative information provided about serving in the military.', 3);
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (6, 6, '2017-06-13
11:11:11', 'Amazing Tournament, wish more people came though!', 5);
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (6, 6, '2017-06-14
11:11:11', 'This was a very poorly run event.', 1);
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (7, 7, '2017-06-15
11:11:11', 'I got destroyed by everyone, but it was fun', 3);
```

SQL Examples and Results:

Insert new RSO:

```
INSERT INTO rsos (name, description, belongsTo, studentAdmin, rsoManager)
        VALUES('Manga Club', 'Discuss new and popular manga with others', 1, 14, 2);
```



✓ Showing rows 0 - 4 (5 total, Query took 0.0000 seconds.)

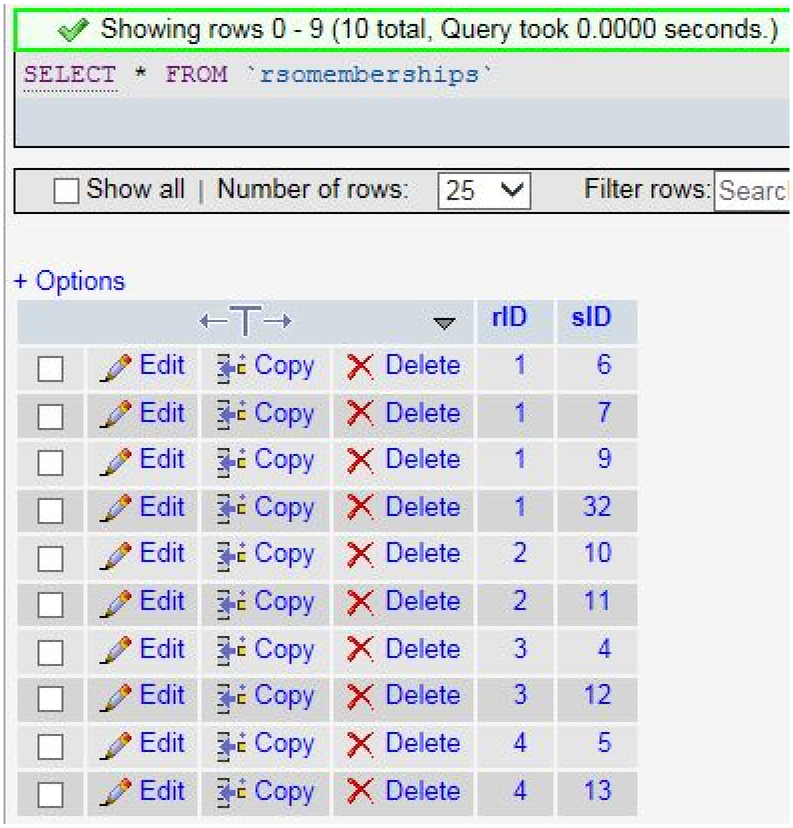
```
SELECT * FROM `rsos`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create Pl
□ Show all | Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options

				rID	name	description	belongsTo	studentAdmin	rsoManager
□	Edit	Copy	Delete	1	Hack UCF	Learn cyber security	1	6	2
□	Edit	Copy	Delete	2	Chess Club	Have fun and learn how to play Chess with others	2	10	3
□	Edit	Copy	Delete	3	Anime Club	Watch and Discuss anime with others	3	12	4
□	Edit	Copy	Delete	4	SMASH Club	Play Smash (Melee, Brawl, Smash 4) with other stud...	4	13	5
□	Edit	Copy	Delete	5	Manga Club	Discuss new and popular manga with other	1	14	2

Insert new student to existing RSO:

```
INSERT INTO rsomemberships (rID, sID) VALUES(1, 32);
```



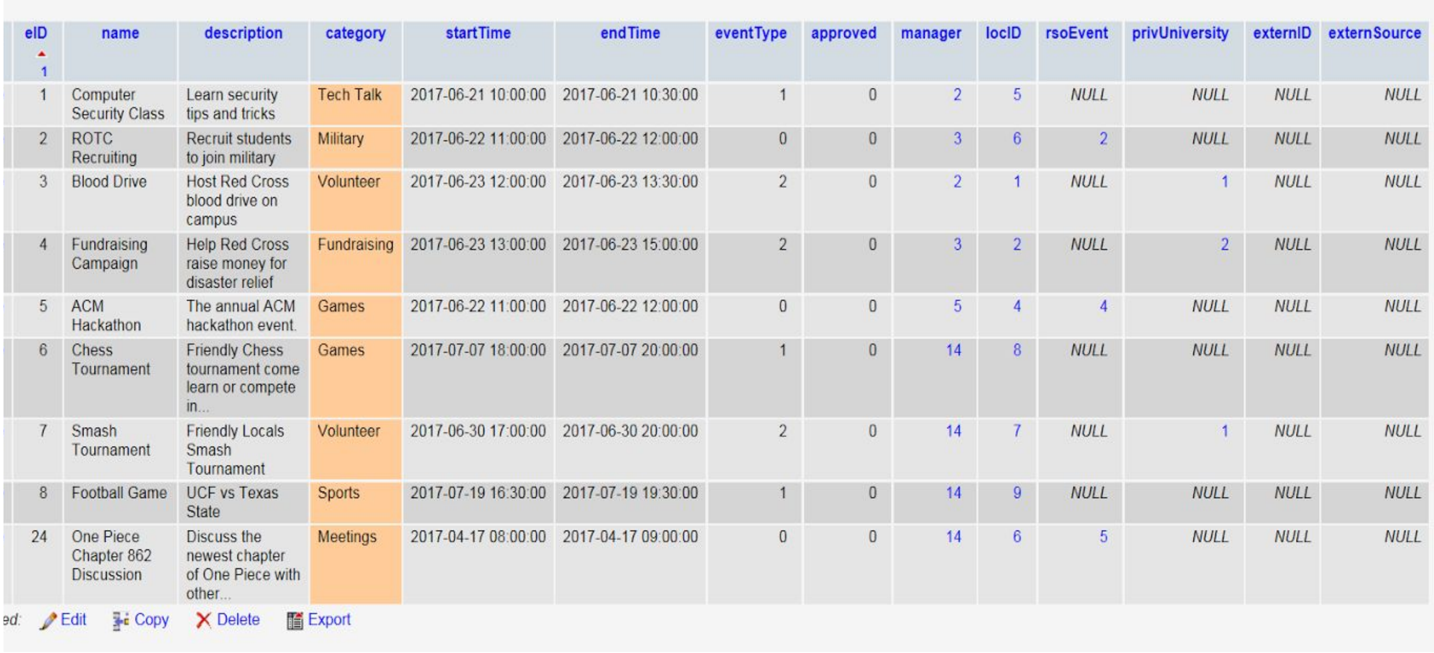
Showing rows 0 - 9 (10 total, Query took 0.0000 seconds.)

```
SELECT * FROM `rsomemberships`
```

				rID	sID
□	Edit	Copy	Delete	1	6
□	Edit	Copy	Delete	1	7
□	Edit	Copy	Delete	1	9
□	Edit	Copy	Delete	1	32
□	Edit	Copy	Delete	2	10
□	Edit	Copy	Delete	2	11
□	Edit	Copy	Delete	3	4
□	Edit	Copy	Delete	3	12
□	Edit	Copy	Delete	4	5
□	Edit	Copy	Delete	4	13

Insert new event:

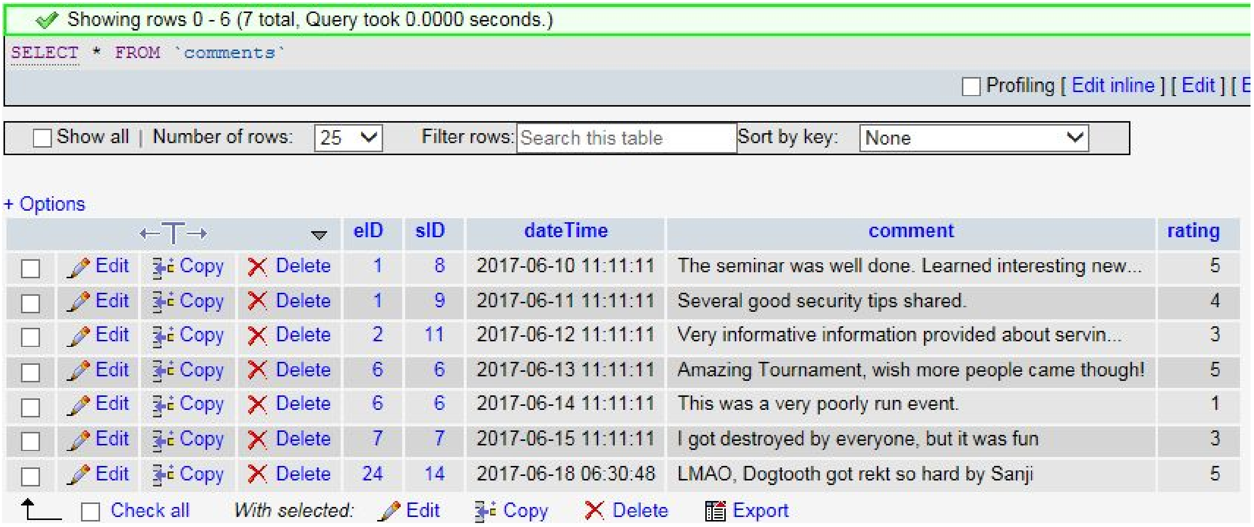
INSERT INTO Events (name, description, category, startTime, endTime, eventType, manager, locID) VALUES ('One Piece Chapter 862 Discussion', 'Discuss the newest chapter of One Piece with others', 'Meetings', '2017-04-17 08:00:00', '2017-04-17 09:00:00', 0, 14, 6);



eID ▲ 1	name	description	category	startTime	endTime	eventType	approved	manager	locID	rsoEvent	privUniversity	externID	externSource
1	Computer Security Class	Learn security tips and tricks	Tech Talk	2017-06-21 10:00:00	2017-06-21 10:30:00	1	0	2	5	NULL	NULL	NULL	NULL
2	ROTC Recruiting	Recruit students to join military	Military	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	3	6	2	NULL	NULL	NULL
3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2017-06-23 13:00:00	2017-06-23 15:00:00	2	0	3	2	NULL	2	NULL	NULL
5	ACM Hackathon	The annual ACM hackathon event.	Games	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	5	4	4	NULL	NULL	NULL
6	Chess Tournament	Friendly Chess tournament come learn or compete in...	Games	2017-07-07 18:00:00	2017-07-07 20:00:00	1	0	14	8	NULL	NULL	NULL	NULL
7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
8	Football Game	UCF vs Texas State	Sports	2017-07-19 16:30:00	2017-07-19 19:30:00	1	0	14	9	NULL	NULL	NULL	NULL
24	One Piece Chapter 862 Discussion	Discuss the newest chapter of One Piece with other...	Meetings	2017-04-17 08:00:00	2017-04-17 09:00:00	0	0	14	6	5	NULL	NULL	NULL

Insert a new comment:

INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (24, 14, '2017-06-18 06:30:48', 'LMAO, Dogtooth got rekt so hard by Sanji', 5);



✓ Showing rows 0 - 6 (7 total, Query took 0.0000 seconds.)

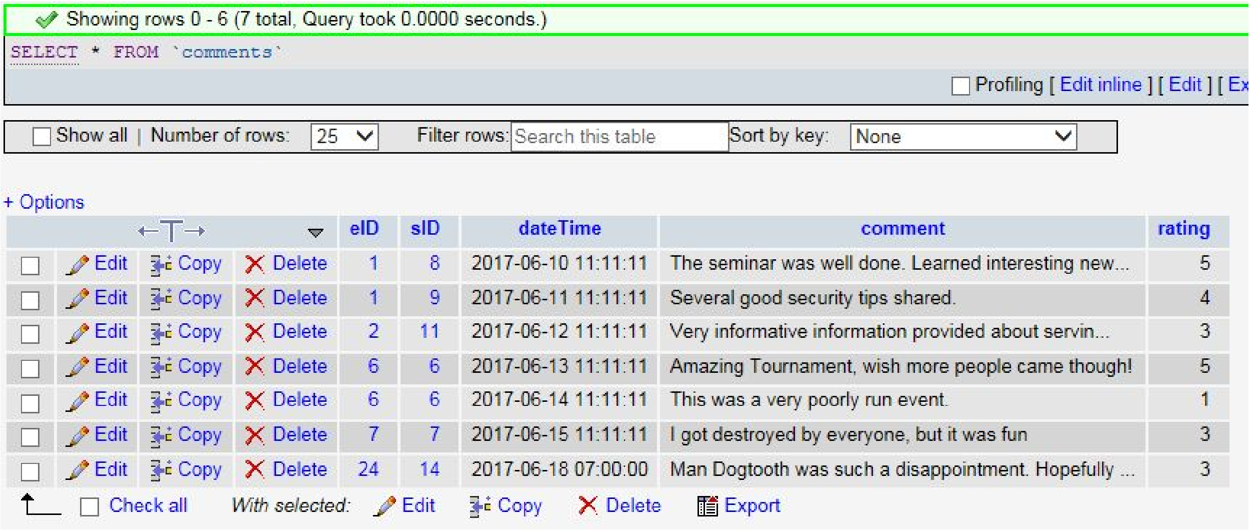
```
SELECT * FROM `comments`
```

□ Profiling [Edit inline] [Edit] [E
□ Show all | Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected: Edit Copy Delete Export

				eID	sID	dateTime	comment	rating
□	Edit	Copy	Delete	1	8	2017-06-10 11:11:11	The seminar was well done. Learned interesting new...	5
□	Edit	Copy	Delete	1	9	2017-06-11 11:11:11	Several good security tips shared.	4
□	Edit	Copy	Delete	2	11	2017-06-12 11:11:11	Very informative information provided about servin...	3
□	Edit	Copy	Delete	6	6	2017-06-13 11:11:11	Amazing Tournament, wish more people came though!	5
□	Edit	Copy	Delete	6	6	2017-06-14 11:11:11	This was a very poorly run event.	1
□	Edit	Copy	Delete	7	7	2017-06-15 11:11:11	I got destroyed by everyone, but it was fun	3
□	Edit	Copy	Delete	24	14	2017-06-18 06:30:48	LMAO, Dogtooth got rekt so hard by Sanji	5

Update a comment:

DELETE FROM Comments WHERE eID=24 AND sID=14;
INSERT INTO Comments (eID, sID, dateTime, comment, rating) VALUES (24, 14, '2017-06-18 07:00:00', 'Man Dogtooth was such a disappointment. Hopefully Oda won't make him a joke character like Jack', 3);



✓ Showing rows 0 - 6 (7 total, Query took 0.0000 seconds.)

```
SELECT * FROM `comments`
```

□ Profiling [Edit inline] [Edit] [Ex
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

	←T→		▼	eID	sID	dateTime	comment	rating
□	Edit	Copy	Delete	1	8	2017-06-10 11:11:11	The seminar was well done. Learned interesting new...	5
□	Edit	Copy	Delete	1	9	2017-06-11 11:11:11	Several good security tips shared.	4
□	Edit	Copy	Delete	2	11	2017-06-12 11:11:11	Very informative information provided about servin...	3
□	Edit	Copy	Delete	6	6	2017-06-13 11:11:11	Amazing Tournament, wish more people came though!	5
□	Edit	Copy	Delete	6	6	2017-06-14 11:11:11	This was a very poorly run event.	1
□	Edit	Copy	Delete	7	7	2017-06-15 11:11:11	I got destroyed by everyone, but it was fun	3
□	Edit	Copy	Delete	24	14	2017-06-18 07:00:00	Man Dogtooth was such a disappointment. Hopefully ...	3

Views were used to display the different Events depending on the User-type session currently logged in. These views were used in the application's code checking the $_Session['adminType'] displaying events depending on the User's privileges.

Display public events:

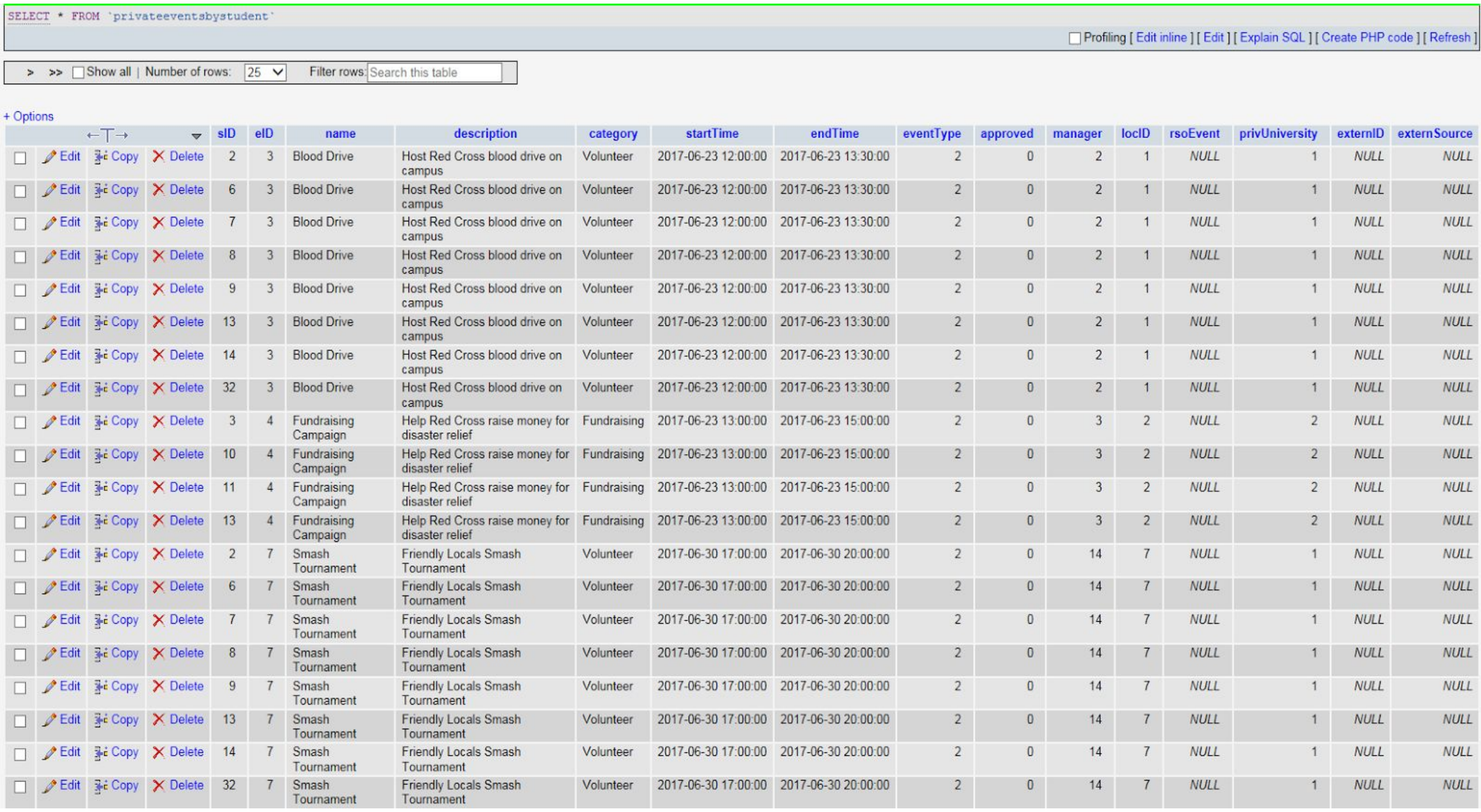
All public events are displayed by default, but here is the view used to view all events.

```
CREATE OR REPLACE VIEW AllEventsByStudent AS
        SELECT * FROM PrivateEventsByStudent UNION
        SELECT * FROM RSOEventsByStudent UNION
        SELECT 0 as sID, Evt.* FROM Events Evt WHERE eventType=1;
```

Display private events:

Private events can only be seen by those who are enrolled to that University from the enrolled table. Here is the view used for specifically private events only.

```
CREATE OR REPLACE VIEW PrivateEventsByStudent AS
 SELECT S.sID, Evt.* FROM Students S INNER JOIN Enrollments E ON S.sID = E.sID INNER
JOIN Events Evt ON E.uID = Evt.privUniversity;
```



```
SELECT * FROM `privateeventsbystudent`
```

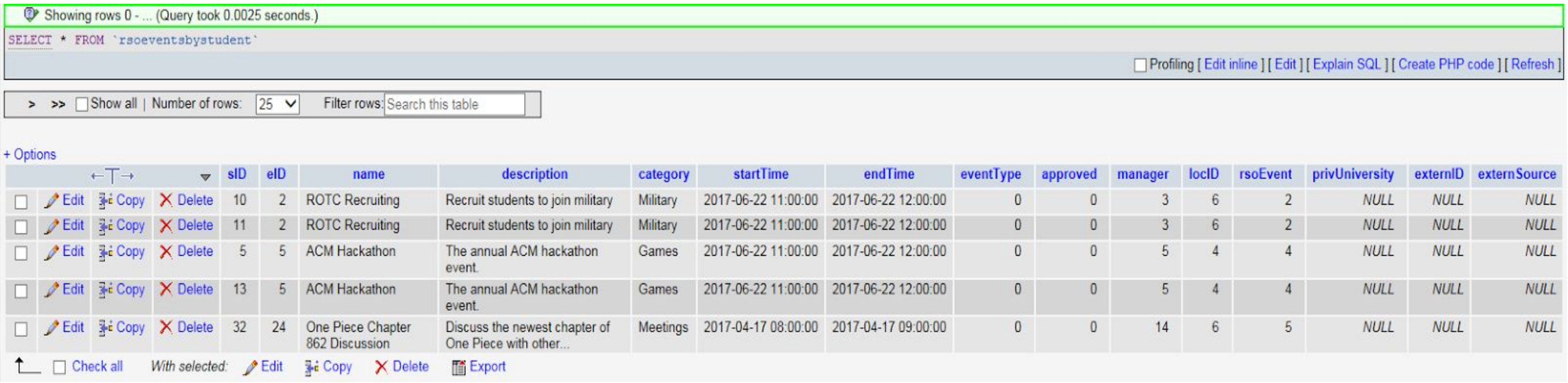
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
> >> □ Show all | Number of rows: 25 Filter rows: Search this table
+ Options

←T→				sID	eID	name	description	category	startTime	endTime	eventType	approved	manager	locID	rsoEvent	privUniversity	externID	externSource
□	Edit	Copy	Delete	2	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	6	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	7	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	8	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	9	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	13	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	14	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	32	3	Blood Drive	Host Red Cross blood drive on campus	Volunteer	2017-06-23 12:00:00	2017-06-23 13:30:00	2	0	2	1	NULL	1	NULL	NULL
□	Edit	Copy	Delete	3	4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2017-06-23 13:00:00	2017-06-23 15:00:00	2	0	3	2	NULL	2	NULL	NULL
□	Edit	Copy	Delete	10	4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2017-06-23 13:00:00	2017-06-23 15:00:00	2	0	3	2	NULL	2	NULL	NULL
□	Edit	Copy	Delete	11	4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2017-06-23 13:00:00	2017-06-23 15:00:00	2	0	3	2	NULL	2	NULL	NULL
□	Edit	Copy	Delete	13	4	Fundraising Campaign	Help Red Cross raise money for disaster relief	Fundraising	2017-06-23 13:00:00	2017-06-23 15:00:00	2	0	3	2	NULL	2	NULL	NULL
□	Edit	Copy	Delete	2	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	6	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	7	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	8	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	9	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	13	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	14	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL
□	Edit	Copy	Delete	32	7	Smash Tournament	Friendly Locals Smash Tournament	Volunteer	2017-06-30 17:00:00	2017-06-30 20:00:00	2	0	14	7	NULL	1	NULL	NULL

Display RSO events:

RSO events can only be seen by those who are in that RSO from the RSOMemberships table. Here is the view used for specifically RSOs only.

```
CREATE OR REPLACE VIEW RSOEventsByStudent AS
 SELECT S.sID, Evt.* FROM Students S INNER JOIN RSOMemberships R ON S.sID = R.sID
INNER JOIN Events Evt ON R.rID = Evt.rsoEvent;
```



Showing rows 0 - ... (Query took 0.0025 seconds.)

```
SELECT * FROM `rsoeventsbystudent`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
> >> □ Show all | Number of rows: 25 Filter rows: Search this table
+ Options

				sID	eID	name	description	category	startTime	endTime	eventType	approved	manager	locID	rsoEvent	privUniversity	externID	externSource
□	Edit	Copy	Delete	10	2	ROTC Recruiting	Recruit students to join military	Military	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	3	6	2	NULL	NULL	NULL
□	Edit	Copy	Delete	11	2	ROTC Recruiting	Recruit students to join military	Military	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	3	6	2	NULL	NULL	NULL
□	Edit	Copy	Delete	5	5	ACM Hackathon	The annual ACM hackathon event.	Games	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	5	4	4	NULL	NULL	NULL
□	Edit	Copy	Delete	13	5	ACM Hackathon	The annual ACM hackathon event.	Games	2017-06-22 11:00:00	2017-06-22 12:00:00	0	0	5	4	4	NULL	NULL	NULL
□	Edit	Copy	Delete	32	24	One Piece Chapter 862 Discussion	Discuss the newest chapter of One Piece with other...	Meetings	2017-04-17 08:00:00	2017-04-17 09:00:00	0	0	14	6	5	NULL	NULL	NULL

Conclusion:

In summary, the project integrates database design and creation with frontend and backend web design and development. The application closely models a realistic enterprise consisting of entities and relationships that enforce integrity rules and constraints. The application also usefully and beneficially serves the enterprise by providing a functional and intuitive user-database interface that users (university students) will find informative, readily-accessible, and reasonably secure.

Database performance:

Due to relatively small size of our database, the performance of our database is also relatively quick. Since our queries are not checking and comparing multiple tables at the same time, they all possess an acceptable execution type of less than 0.1 seconds. Therefore, we expect our database performance to be consistently fast when executing any queries for users.

Desired Features:

While the application does feature some social media integration through the ability to share events through users' Facebook accounts and "tweeting" about events, a desired feature we would like to include is the implementation of Twitter feeds based on the university that users are enrolled in (i.e. a student attending the University of Florida would see the University of Florida's Twitter feed while a student attending the University of Central Florida would see the University of Central Florida's Twitter feed).

Another feature we would like to include is the validation of email addresses. While the application already checks students' emails to make sure that they are associated with a specific university profile, the application does not check to see if an email is a real, working email.

Problems Encountered:

During the development of our project, we encountered multiple problems. Perhaps the largest problem to overcome was the fact that no group members had prior experience with any database and website development. Because of this, there was a large learning curve that had to be overcome for developing using languages such as HTML, CSS, and PHP along with tools such as XAMPP and phpMyAdmin.

Another problem that was encountered was the difficulty associated with organizing meeting times that were convenient for every group member. With differing school and work schedules, it was difficult to find enough time for the entire team to meet and work on the project. Having as many group members present for each meeting was crucial to the success of the project as it minimized the possibility of code conflicts between different group members working on the same parts of the website or database.