Group Report: College Event Website

COP 4710, Spring 2021

Group 62: Michael Laager

Table Of Contents:

Project Description

The goal of this project is to create a website that allows students from different universities to create and host events for their communities. These events can be private, public, or specific to specific RSOs that are created through the application. There are three tiers of users on the website: normal users, who can join/create rsos and join events available to them; admins, who can create events for RSOs that they are the admin of; and super-admins, who control the university.

Though the requirements for this project stated for users to be able to edit their comments and to give a rating to their university, unfortunately those are not present in this final product. Also, passwords are encrypted to ensure user privacy is unaffected.

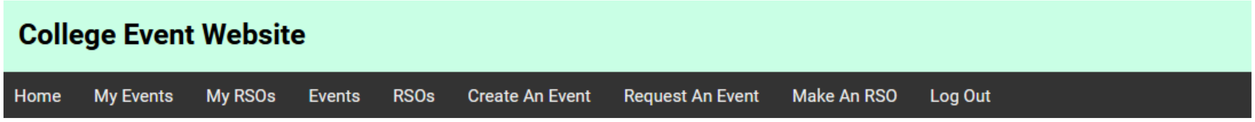
Technical Specifications

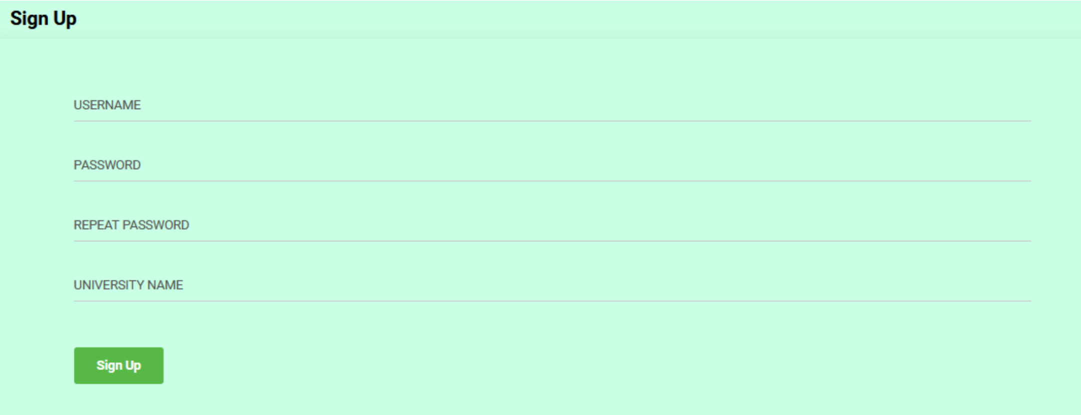
The web-based application was created using a MySQLi backend with InnoDB to support functional dependencies. PHP is used for server-user connections, and the front-end was created using a combination of HTML, CSS, and JS.

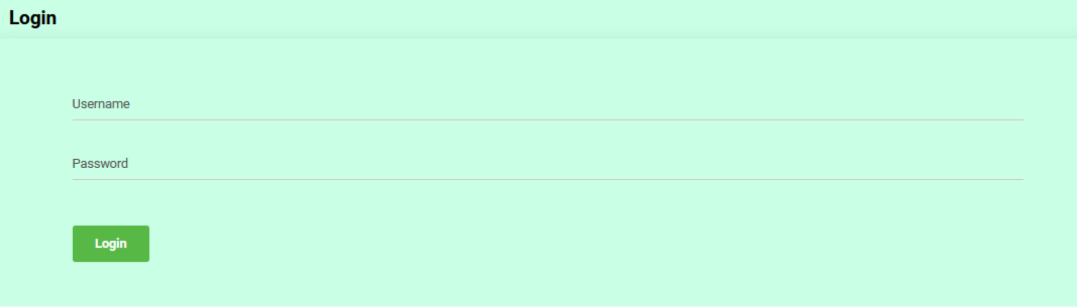
The project was created and tested locally using WAMP, which is a Windows based application that automatically integrates Apache, MySQLi, and PHP for easy and quick development. Firefox and Microsoft Edge were both used to validate front-end visuals.

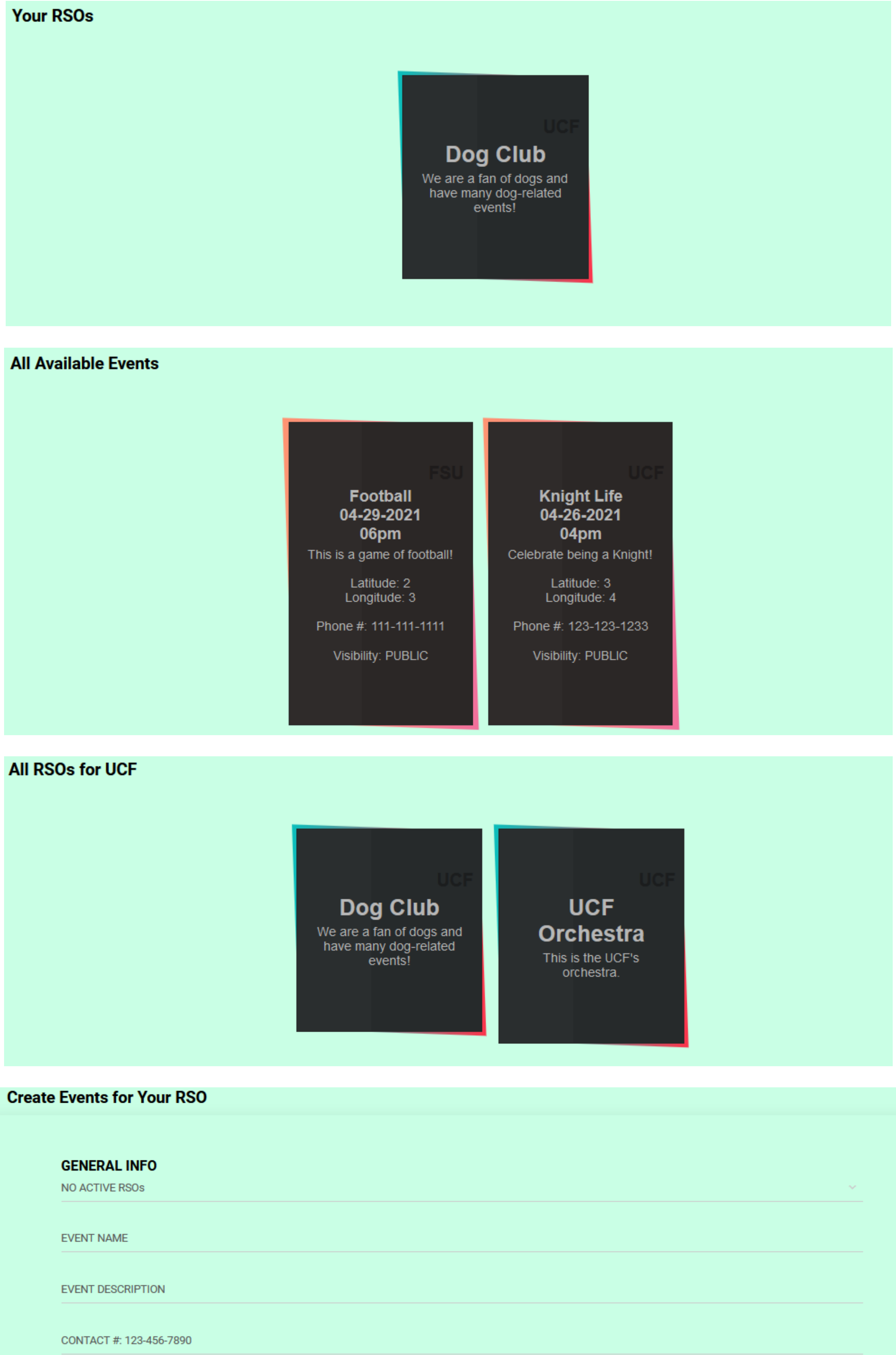
GUI

Here are screenshots from most of the available pages on the website, following an example of the header for the pages.









Your RSOs

UCF

Dog Club

We are a fan of dogs and have many dog-related events!

All Available Events

FSU

Football
04-29-2021
06pm

This is a game of football!

Latitude: 2
Longitude: 3

Phone #: 111-111-1111

Visibility: PUBLIC

UCF

Knight Life
04-26-2021
04pm

Celebrate being a Knight!

Latitude: 3
Longitude: 4

Phone #: 123-123-1233

Visibility: PUBLIC

All RSOs for UCF

UCF

Dog Club

We are a fan of dogs and have many dog-related events!

UCF

UCF Orchestra

This is the UCF's orchestra.

Create Events for Your RSO

GENERAL INFO

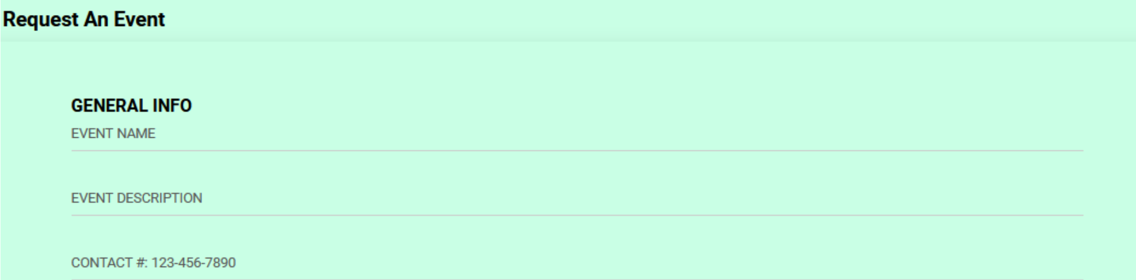
NO ACTIVE RSOs

EVENT NAME

EVENT DESCRIPTION

CONTACT #: 123-456-7890

All Pending Events for Your University



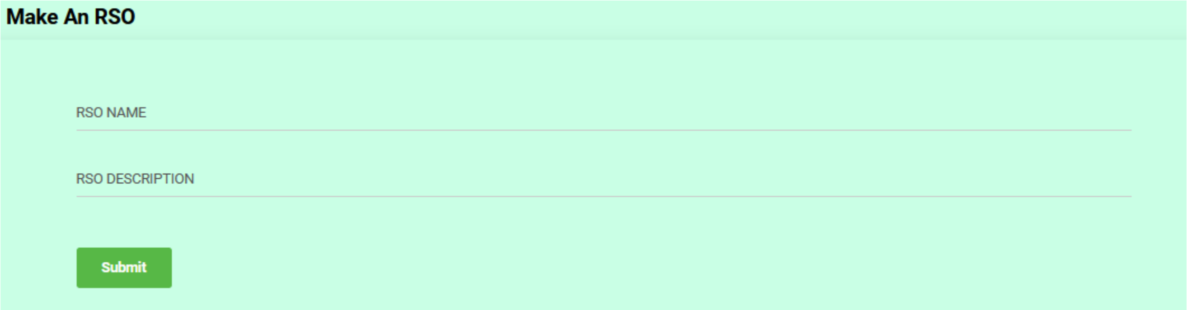
Request An Event

GENERAL INFO

EVENT NAME

EVENT DESCRIPTION

CONTACT #: 123-456-7890

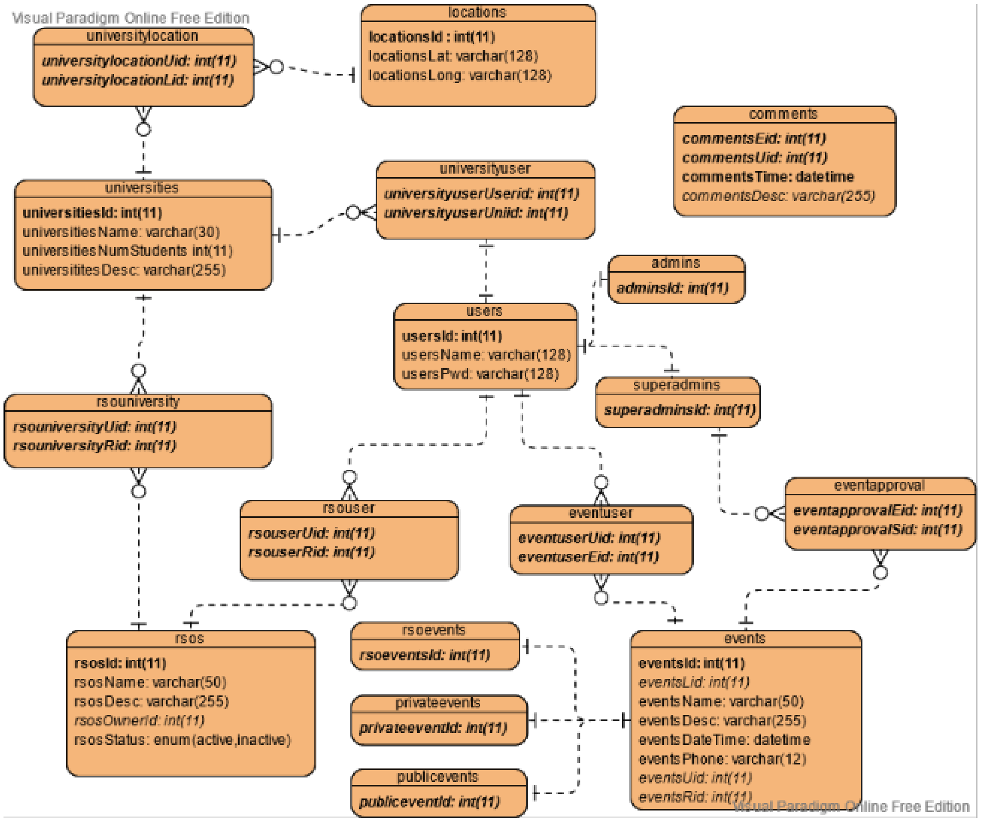


Make An RSO

RSO NAME

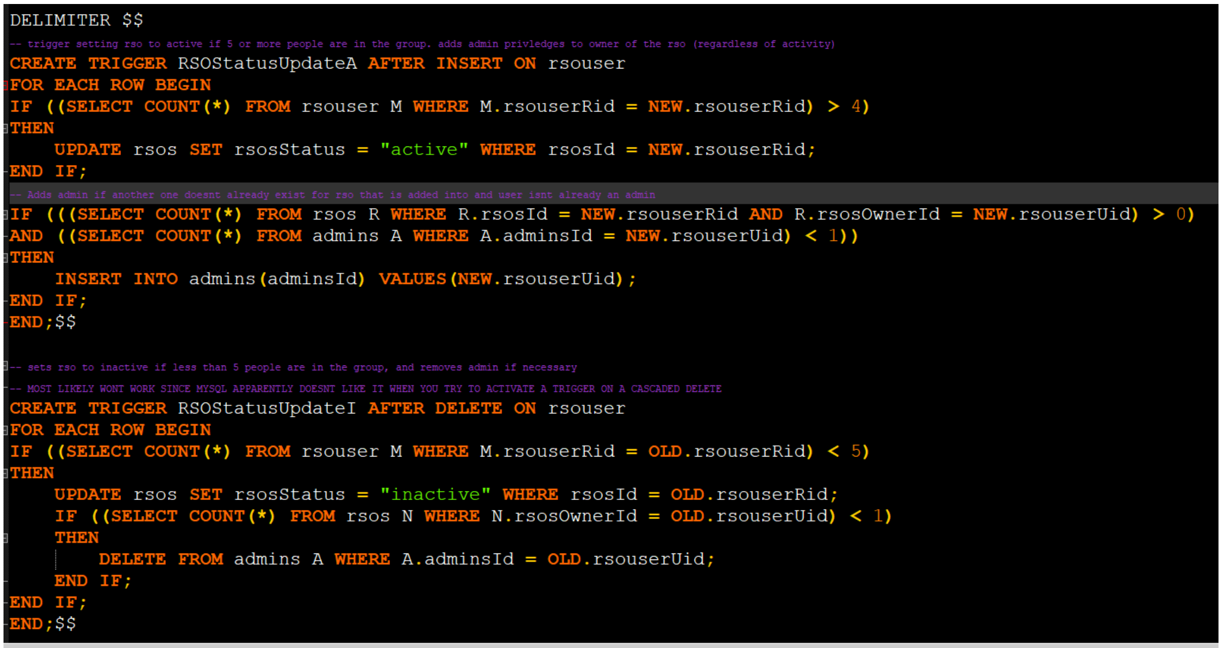
RSO DESCRIPTION

ER Diagram


Visual Paradigm Online Free Edition
universitylocation
universitylocationUid: int(11)
universitylocationLid: int(11)
locations
locationsId : int(11)
locationsLat: varchar(128)
locationsLong: varchar(128)
comments
commentsEid: int(11)
commentsUid: int(11)
commentsTime: datetime
commentsDesc: varchar(255)
universities
universitiesId: int(11)
universitiesName: varchar(30)
universitiesNumStudents int(11)
universititesDesc: varchar(255)
universityuser
universityuserUserid: int(11)
universityuserUniid: int(11)
admins
adminsId: int(11)
users
usersId: int(11)
usersName: varchar(128)
usersPwd: varchar(128)
superadmins
superadminsId: int(11)
rsouniversity
rsouniversityUid: int(11)
rsouniversityRid: int(11)
eventapproval
eventapprovalEid: int(11)
eventapprovalSid: int(11)
rsouser
rsouserUid: int(11)
rsouserRid: int(11)
eventuser
eventuserUid: int(11)
eventuserEid: int(11)
rsos
rsosId: int(11)
rsosName: varchar(50)
rsosDesc: varchar(255)
rsosOwnerId: int(11)
rsosStatus: enum(active,inactive)
rsoevents
rsoeventsId: int(11)
privateevents
privateeventId: int(11)
publicevents
publiceventId: int(11)
events
eventsId: int(11)
eventsLid: int(11)
eventsName: varchar(50)
eventsDesc: varchar(255)
eventsDateTime: datetime
eventsPhone: varchar(12)
eventsUid: int(11)
eventsRid: int(11)
Visual Paradigm Online Free Edition

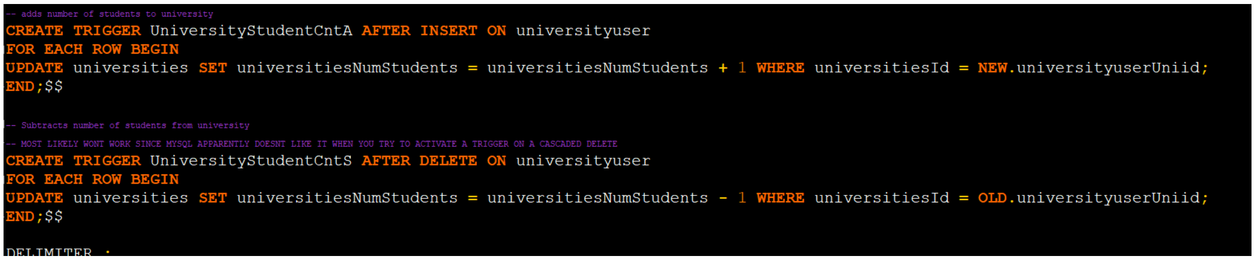
Triggers

Updating RSOs on insert or takeaway:



```
DELIMITER $$
-- trigger setting rso to active if 5 or more people are in the group. adds admin privledges to owner of the rso (regardless of activity)
CREATE TRIGGER RSOStatusUpdateA AFTER INSERT ON rsouser
FOR EACH ROW BEGIN
IF ((SELECT COUNT(*) FROM rsouser M WHERE M.rsouserRid = NEW.rsouserRid) > 4)
THEN
    UPDATE rsos SET rsosStatus = "active" WHERE rsosId = NEW.rsouserRid;
END IF;
-- Adds admin if another one doesnt already exist for rso that is added into and user isnt already an admin
IF (((SELECT COUNT(*) FROM rsos R WHERE R.rsosId = NEW.rsouserRid AND R.rsosOwnerId = NEW.rsouserUid) > 0)
AND ((SELECT COUNT(*) FROM admins A WHERE A.adminsId = NEW.rsouserUid) < 1))
THEN
    INSERT INTO admins(adminsId) VALUES(NEW.rsouserUid);
END IF;
END;$$

-- sets rso to inactive if less than 5 people are in the group, and removes admin if necessary
-- MOST LIKELY WONT WORK SINCE MYSQL APPARENTLY DOESNT LIKE IT WHEN YOU TRY TO ACTIVATE A TRIGGER ON A CASCADED DELETE
CREATE TRIGGER RSOStatusUpdateI AFTER DELETE ON rsouser
FOR EACH ROW BEGIN
IF ((SELECT COUNT(*) FROM rsouser M WHERE M.rsouserRid = OLD.rsouserRid) < 5)
THEN
    UPDATE rsos SET rsosStatus = "inactive" WHERE rsosId = OLD.rsouserRid;
    IF ((SELECT COUNT(*) FROM rsos N WHERE N.rsosOwnerId = OLD.rsouserUid) < 1)
    THEN
        DELETE FROM admins A WHERE A.adminsId = OLD.rsouserUid;
    END IF;
END IF;
END;$$
```

Updating university count on insert/removal:



```
-- adds number of students to university
CREATE TRIGGER UniversityStudentCntA AFTER INSERT ON universityuser
FOR EACH ROW BEGIN
UPDATE universities SET universitiesNumStudents = universitiesNumStudents + 1 WHERE universitiesId = NEW.universityuserUniid;
END;$$

-- Subtracts number of students from university
-- MOST LIKELY WONT WORK SINCE MYSQL APPARENTLY DOESNT LIKE IT WHEN YOU TRY TO ACTIVATE A TRIGGER ON A CASCADED DELETE
CREATE TRIGGER UniversityStudentCntS AFTER DELETE ON universityuser
FOR EACH ROW BEGIN
UPDATE universities SET universitiesNumStudents = universitiesNumStudents - 1 WHERE universitiesId = OLD.universityuserUniid;
END;$$

DELIMITER ;
```

Sample Data

Locations are given in the format: Location = (locationLat, locationLong)

- Universities
  - UCF – Location = (1,2); Description = “This is UCF’s Description!”
  - FSU – Location = (2,3); Description = “This is FSU’s Description!”
- Students
  - UCF
    - michael – UCF superadmin
    - 7 other students (“javi”,”ben”,”josh”,”nick”,”jordan”)
  - FSU
    - fsusuper – FSU superadmin
    - 2 other students (“lisa”,”edward”)
- RSO
  - UCF
    - Name: “Dog Club”; Desc “We are a fan of dogs and have many dog-related events”; Admin: “javi”; Users: (“javi”,”ben”,”nick”,”jordan”)
- Events
  - FSU
    - Name “Football”; Desc “This is a game of football!”; Location (2,3); Phone: 111-111-1111; Date/Time: 4-29-2021 @ 6pm; Visibility: Public; Users: (“lisa”, “michael”)
- Comments
  - Football
    - Commenter: lisa; Comment = “I hope FSU wins!”
    - Commenter: michael; Comment = “I hope UCF wins!”

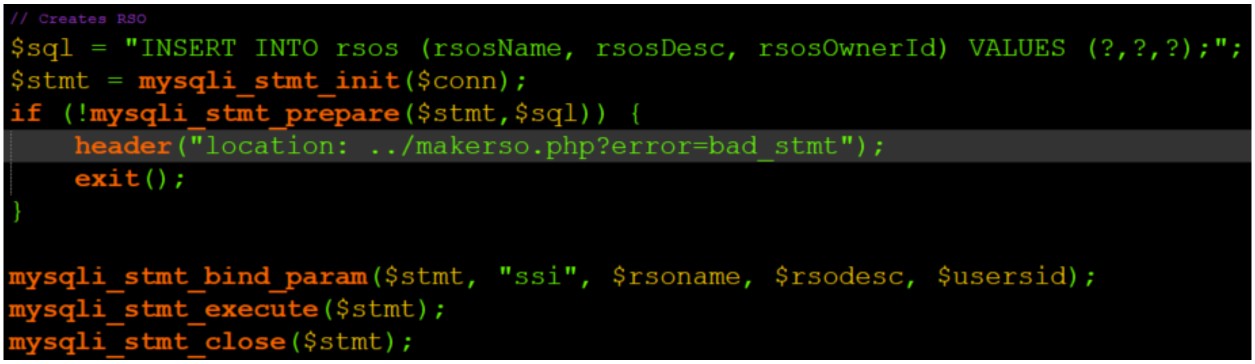
SQL and Front-End Examples

For this section, both the web-application and the SQL statements are given to show the functionality of the project. SQL statements are given as the implemented PHP queries.

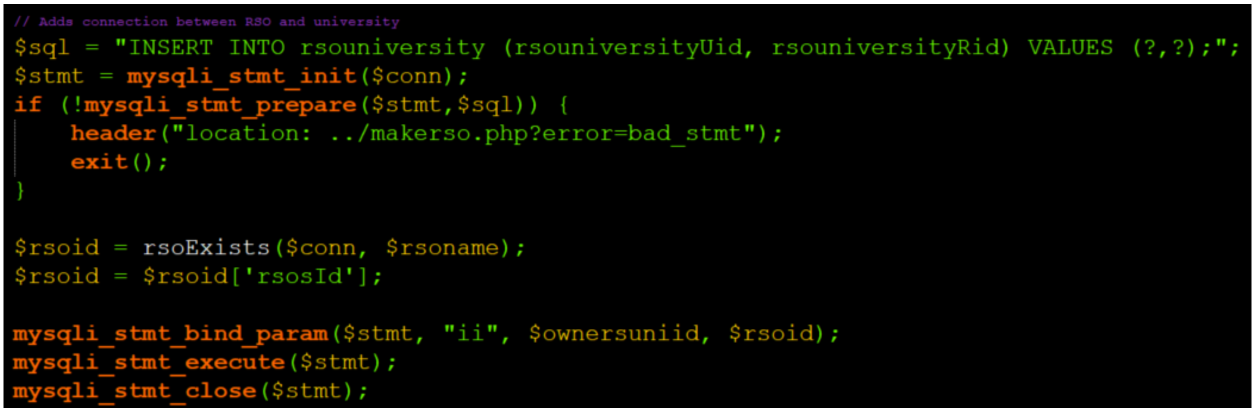
New RSO

Creating new RSO “UCF Orchestra” as user from UCF.

SQL:

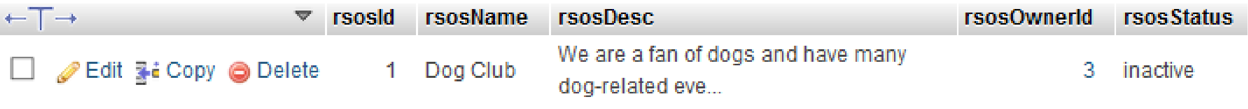


```
// Creates RSO
$sql = "INSERT INTO rsos (rsosName, rsosDesc, rsosOwnerId) VALUES (?,?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    header("location: ../makerso.php?error=bad_stmt");
    exit();
}

mysqli_stmt_bind_param($stmt, "ssi", $rsoname, $rsodesc, $usersid);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

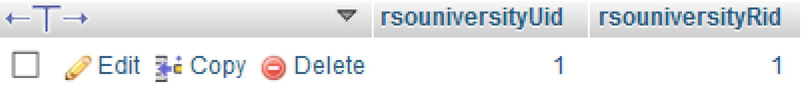


```
// Adds connection between RSO and university
$sql = "INSERT INTO rsouniversity (rsouniversityUid, rsouniversityRid) VALUES (?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    header("location: ../makerso.php?error=bad_stmt");
    exit();
}

$rsoid = rsoExists($conn, $rsoname);
$rsoid = $rsoid['rsosId'];

mysqli_stmt_bind_param($stmt, "ii", $ownersuniid, $rsoid);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

Database Before:

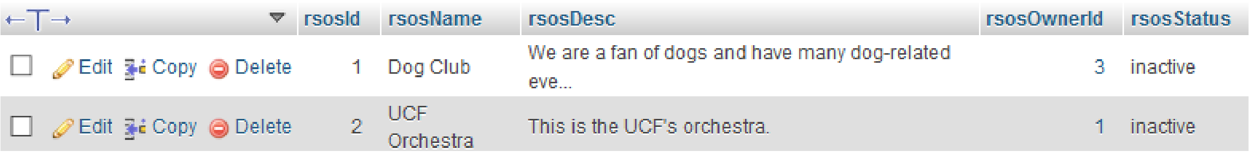


←T→	rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□ Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	inactive

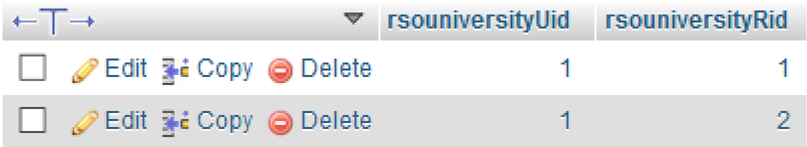


←T→	rsouniversityUid	rsouniversityRid
□ Edit Copy Delete	1	1

Database After:

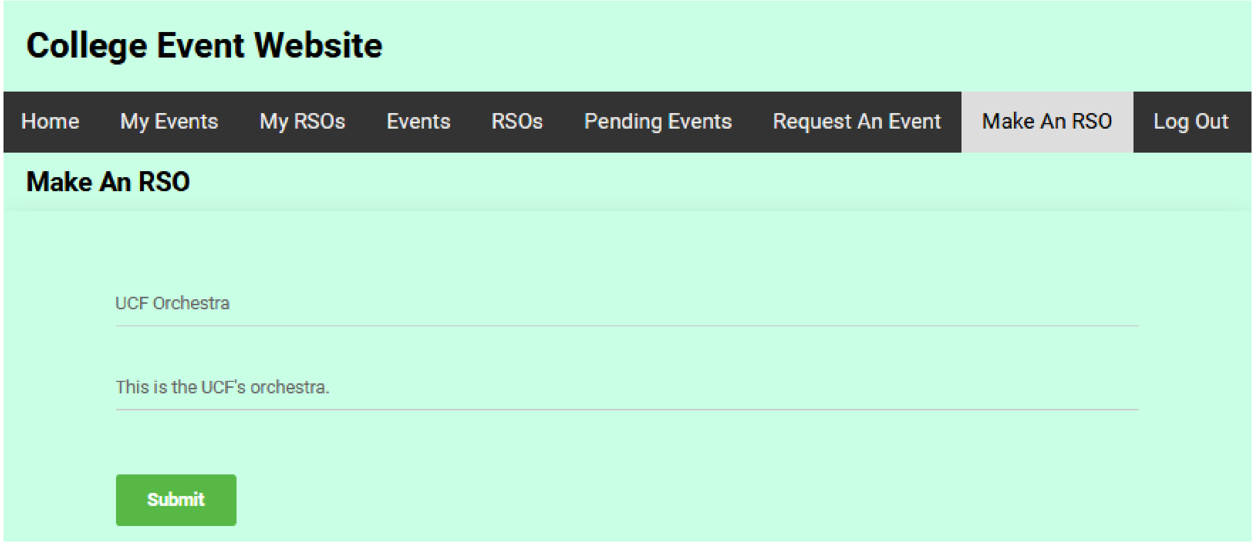


←T→ ▾	rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□ Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	inactive
□ Edit Copy Delete	2	UCF Orchestra	This is the UCF's orchestra.	1	inactive



←T→ ▾	rsouniversityUid	rsouniversityRid
□ Edit Copy Delete	1	1
□ Edit Copy Delete	1	2

Front-End Before:



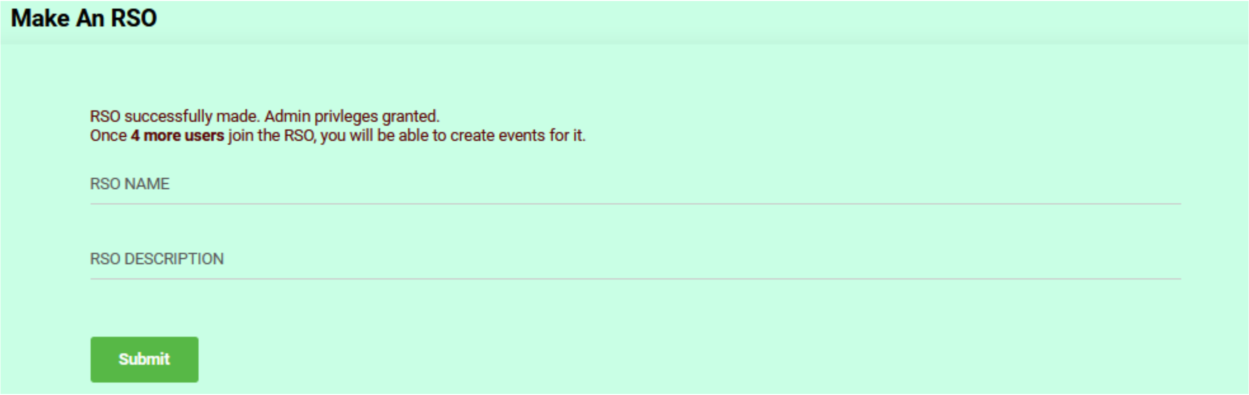
College Event Website

Make An RSO

UCF Orchestra

This is the UCF's orchestra.

Front-End After:



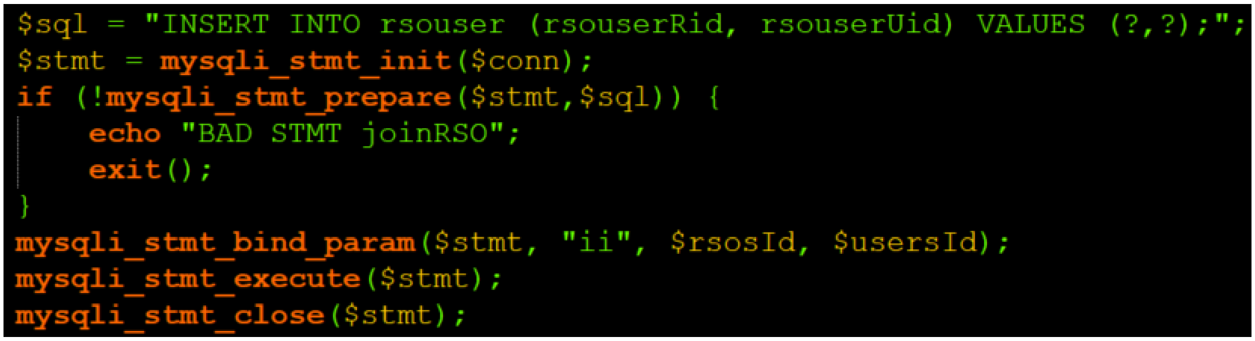
Make An RSO

RSO successfully made. Admin privleges granted.
Once **4 more users** join the RSO, you will be able to create events for it.

New Student in RSO

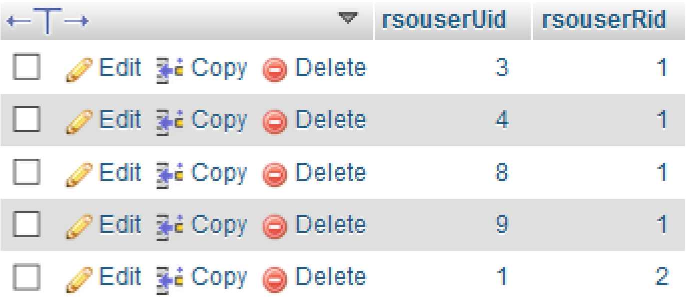
Adding student "nick" as user to "UCF Orchestra".

SQL:



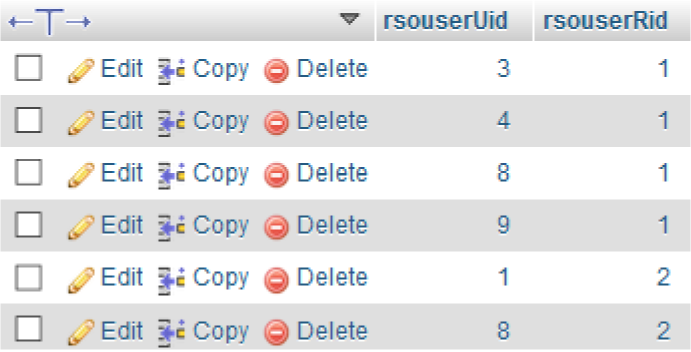
```
$sql = "INSERT INTO rsouser (rsouserRid, rsouserUid) VALUES (?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo "BAD STMT joinRSO";
    exit();
}
mysqli_stmt_bind_param($stmt, "ii", $rsosId, $usersId);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

Database Before:



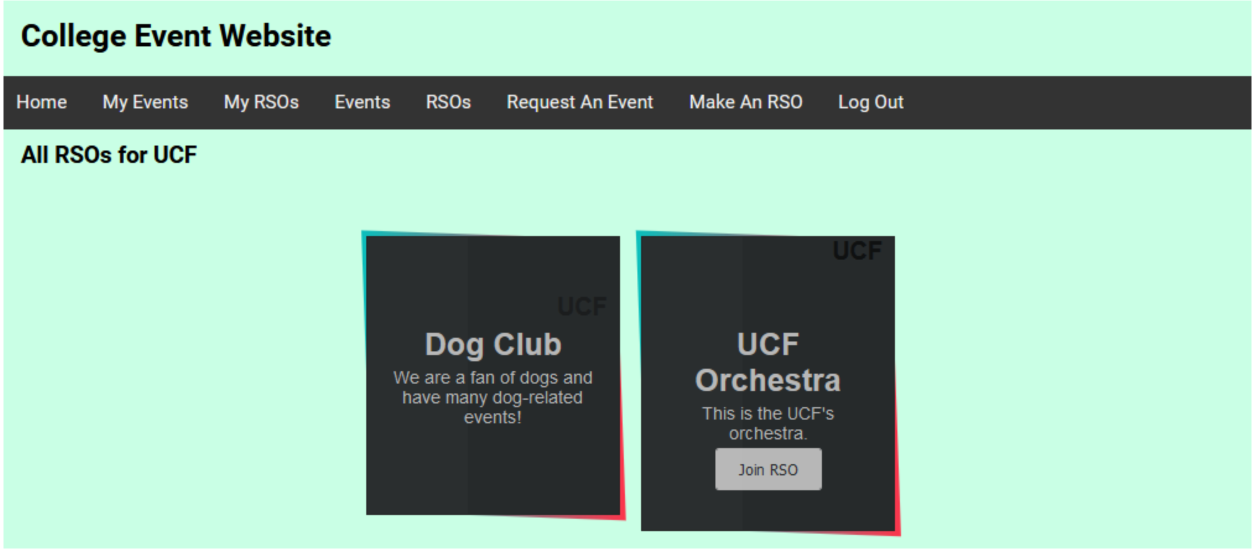
←T→	▼	rsouserUid	rsouserRid
□	Edit Copy Delete	3	1
□	Edit Copy Delete	4	1
□	Edit Copy Delete	8	1
□	Edit Copy Delete	9	1
□	Edit Copy Delete	1	2

Database After:



←T→	▼	rsouserUid	rsouserRid
□	Edit Copy Delete	3	1
□	Edit Copy Delete	4	1
□	Edit Copy Delete	8	1
□	Edit Copy Delete	9	1
□	Edit Copy Delete	1	2
□	Edit Copy Delete	8	2

Front-End Before:



College Event Website

All RSOs for UCF

UCF

Dog Club

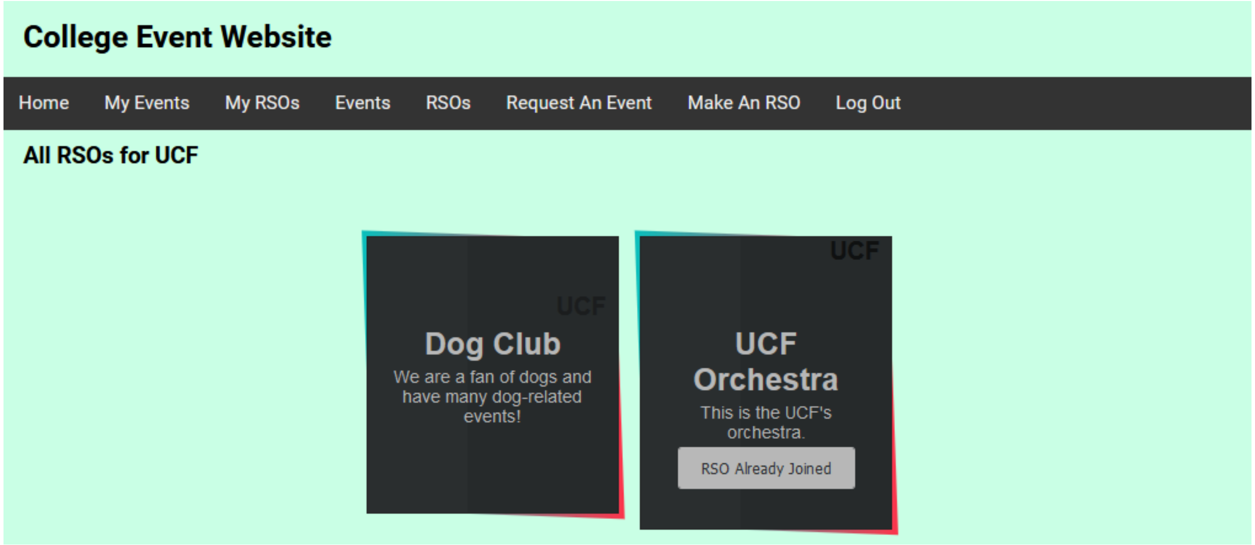
We are a fan of dogs and have many dog-related events!

UCF

UCF Orchestra

This is the UCF's orchestra.

Front-End After:



College Event Website

All RSOs for UCF

UCF

Dog Club

We are a fan of dogs and have many dog-related events!

UCF

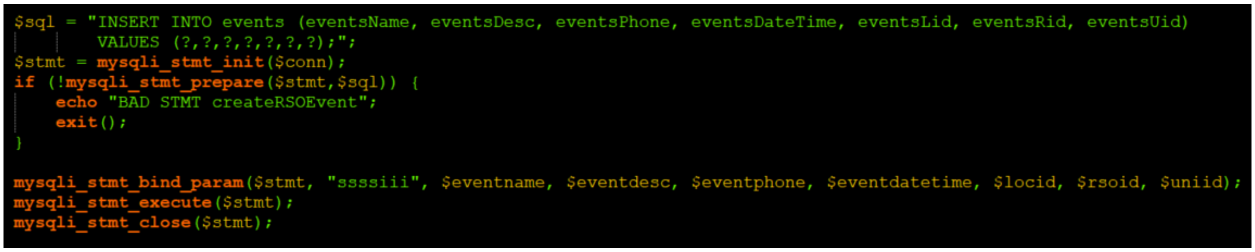
UCF Orchestra

This is the UCF's orchestra.

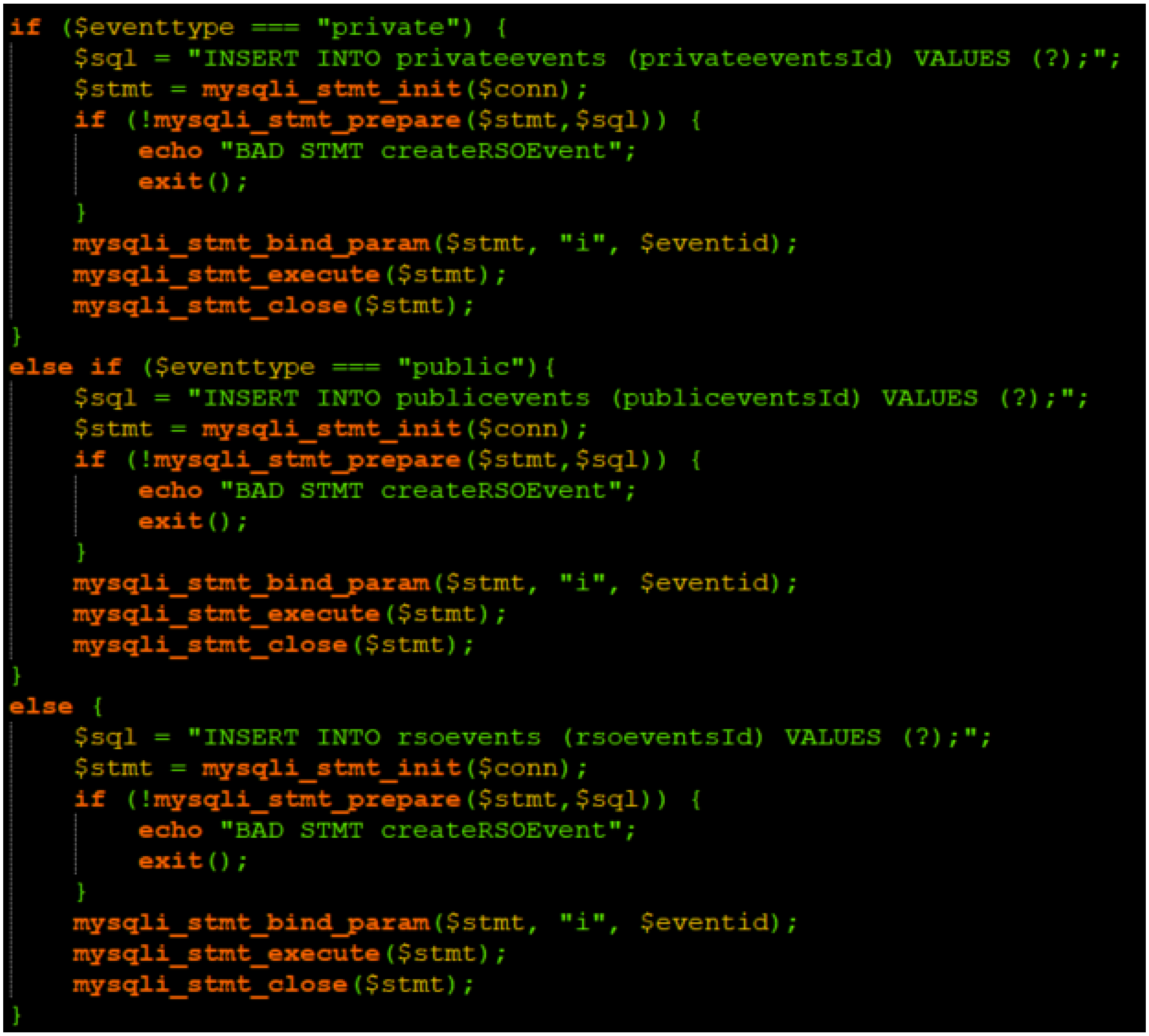
Requesting New Event

Requesting new public Event “Knight Life” as “jordan”.

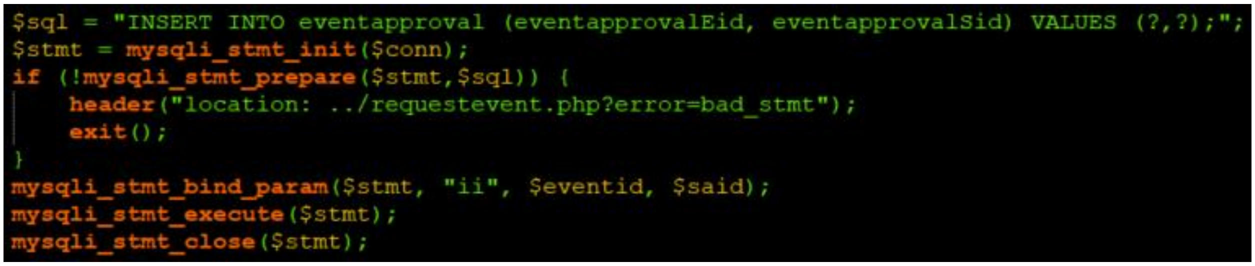
SQL:



```
$sql = "INSERT INTO events (eventsName, eventsDesc, eventsPhone, eventsDateTime, eventsLid, eventsRid, eventsUid)
        VALUES (?,?,?,?,?,?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo "BAD STMT createRSOEvent";
    exit();
}

mysqli_stmt_bind_param($stmt, "ssssiii", $eventname, $eventdesc, $eventphone, $eventdatetime, $locid, $rsoid, $uniid);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

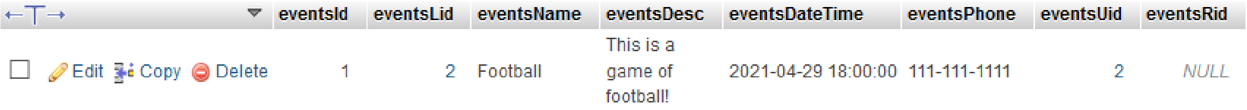


```
if ($eventtype === "private") {
    $sql = "INSERT INTO privateevents (privateeventsId) VALUES (?);";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt,$sql)) {
        echo "BAD STMT createRSOEvent";
        exit();
    }
    mysqli_stmt_bind_param($stmt, "i", $eventid);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
}
else if ($eventtype === "public"){
    $sql = "INSERT INTO publicevents (publiceventsId) VALUES (?);";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt,$sql)) {
        echo "BAD STMT createRSOEvent";
        exit();
    }
    mysqli_stmt_bind_param($stmt, "i", $eventid);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
}
else {
    $sql = "INSERT INTO rsoevents (rsoeventsId) VALUES (?);";
    $stmt = mysqli_stmt_init($conn);
    if (!mysqli_stmt_prepare($stmt,$sql)) {
        echo "BAD STMT createRSOEvent";
        exit();
    }
    mysqli_stmt_bind_param($stmt, "i", $eventid);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_close($stmt);
}
```

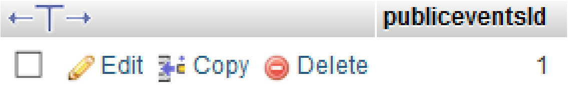


```
$sql = "INSERT INTO eventapproval (eventapprovalEid, eventapprovalSid) VALUES (?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    header("location: ../requestevent.php?error=bad_stmt");
    exit();
}
mysqli_stmt_bind_param($stmt, "ii", $eventid, $said);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

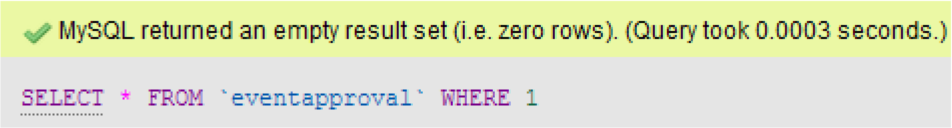
Database Before:



←T→	eventsId	eventsLid	eventsName	eventsDesc	eventsDateTime	eventsPhone	eventsUid	eventsRid
□ Edit Copy Delete	1	2	Football	This is a game of football!	2021-04-29 18:00:00	111-111-1111	2	NULL



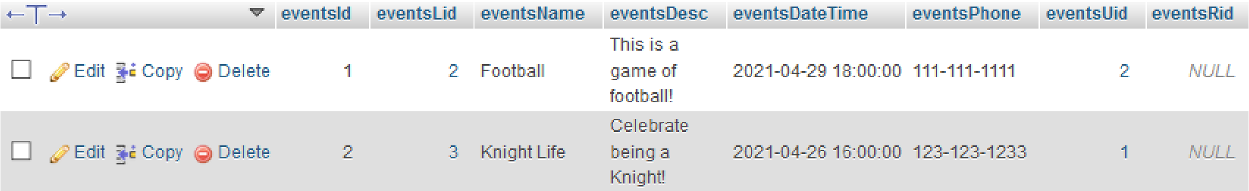
←T→	publiceventsId
□ Edit Copy Delete	1



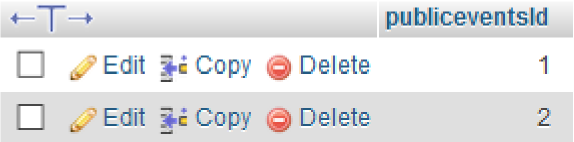
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

```
SELECT * FROM `eventapproval` WHERE 1
```

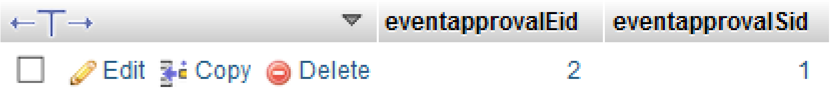
Database After:



←T→	eventsId	eventsLid	eventsName	eventsDesc	eventsDateTime	eventsPhone	eventsUid	eventsRid
□ Edit Copy Delete	1	2	Football	This is a game of football!	2021-04-29 18:00:00	111-111-1111	2	NULL
□ Edit Copy Delete	2	3	Knight Life	Celebrate being a Knight!	2021-04-26 16:00:00	123-123-1233	1	NULL

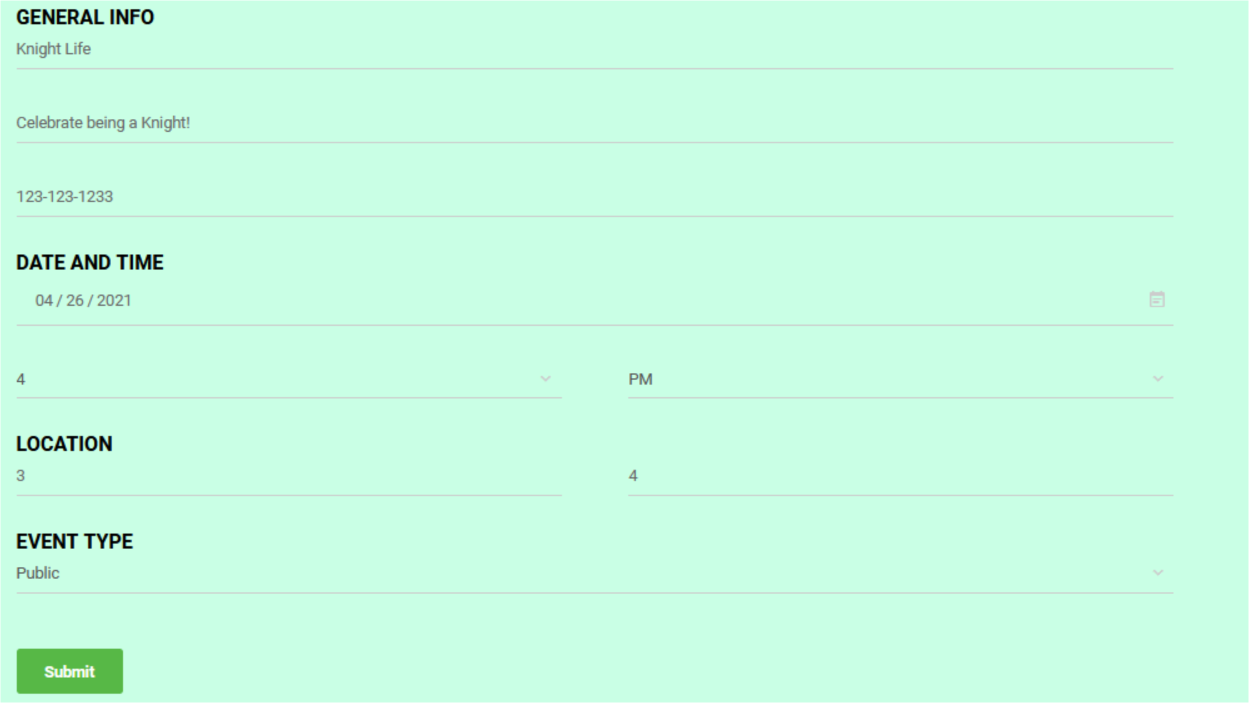


←T→	publiceventsId
□ Edit Copy Delete	1
□ Edit Copy Delete	2



←T→	eventapprovalEid	eventapprovalSid
□ Edit Copy Delete	2	1

Front-End Before:



GENERAL INFO

Knight Life

Celebrate being a Knight!

123-123-1233

DATE AND TIME

04 / 26 / 2021

4

PM

LOCATION

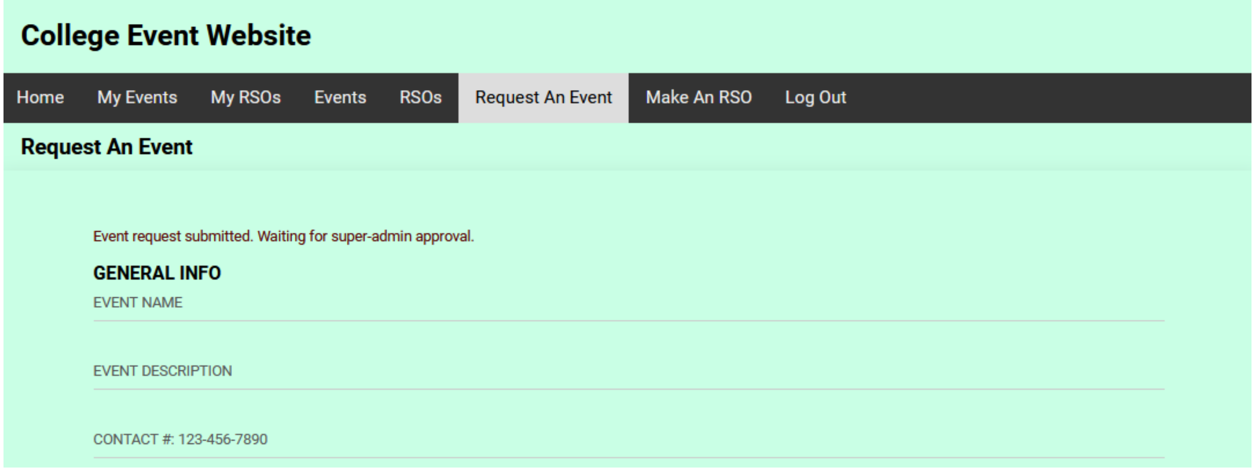
3

4

EVENT TYPE

Public

Front-End After:



College Event Website

Request An Event

Event request submitted. Waiting for super-admin approval.

GENERAL INFO

EVENT NAME

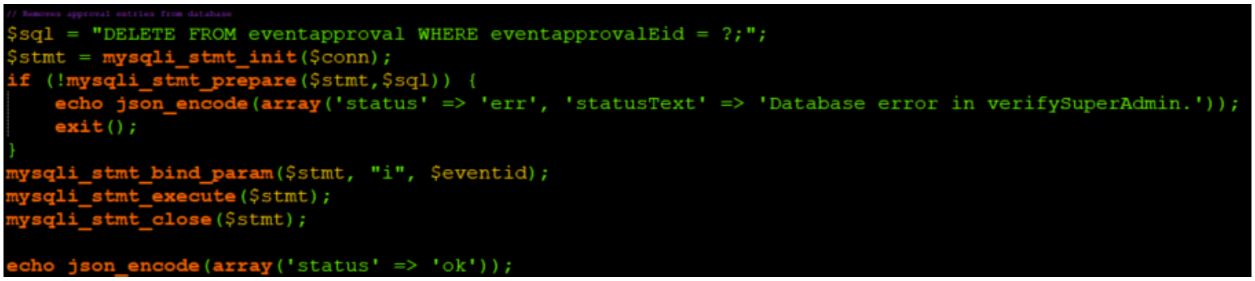
EVENT DESCRIPTION

CONTACT #: 123-456-7890

Accepting New Event

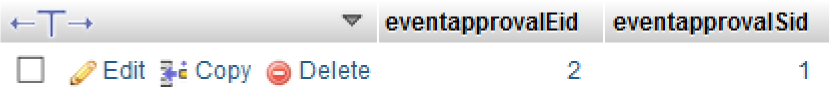
Accepting "Knight Life" as "michael".

SQL:



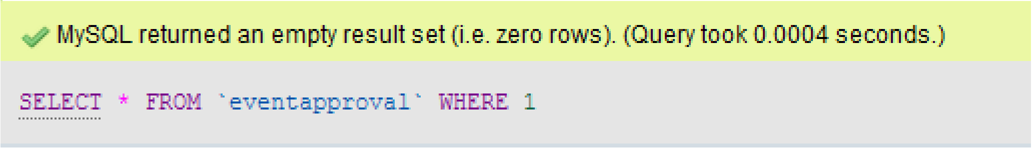
```
// Removes approval entries from database
$sql = "DELETE FROM eventapproval WHERE eventapprovalEid = ?;";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo json_encode(array('status' => 'err', 'statusText' => 'Database error in verifySuperAdmin.'));
    exit();
}
mysqli_stmt_bind_param($stmt, "i", $eventid);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);

echo json_encode(array('status' => 'ok'));
```

Database Before:



←T→	eventapprovalEid	eventapprovalSid
□ Edit Copy Delete	2	1

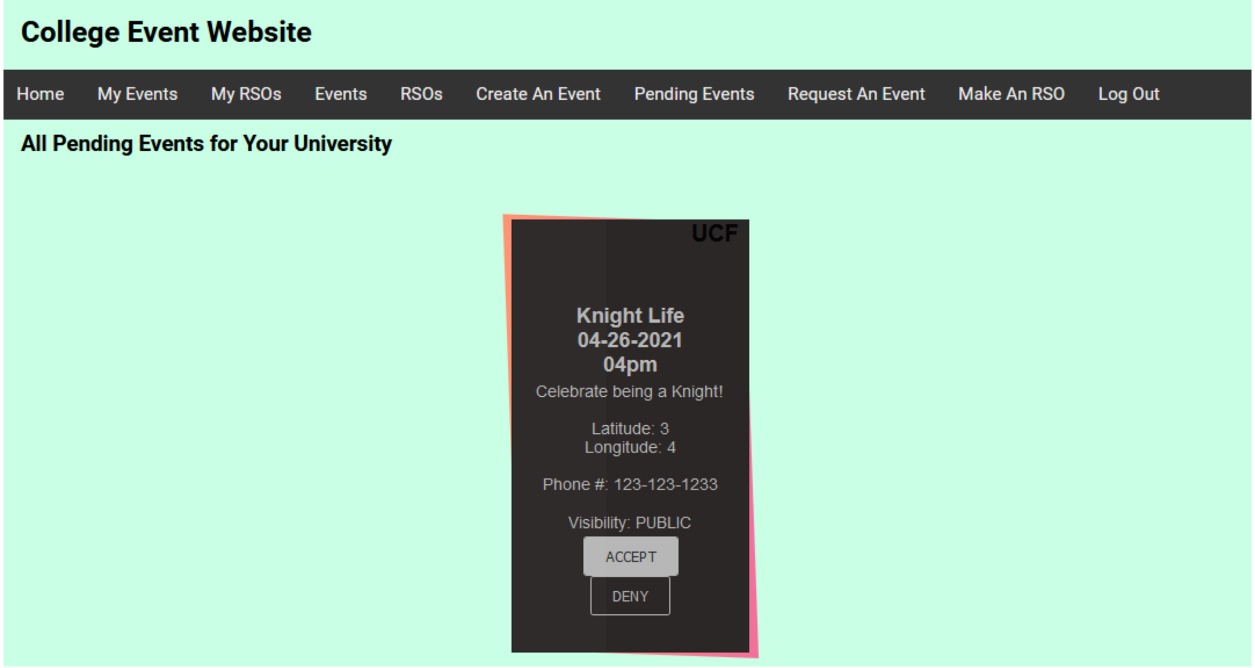
Database After:



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
SELECT * FROM `eventapproval` WHERE 1
```

Front-End Before:



All Pending Events for Your University

UCF

Knight Life
04-26-2021
04pm

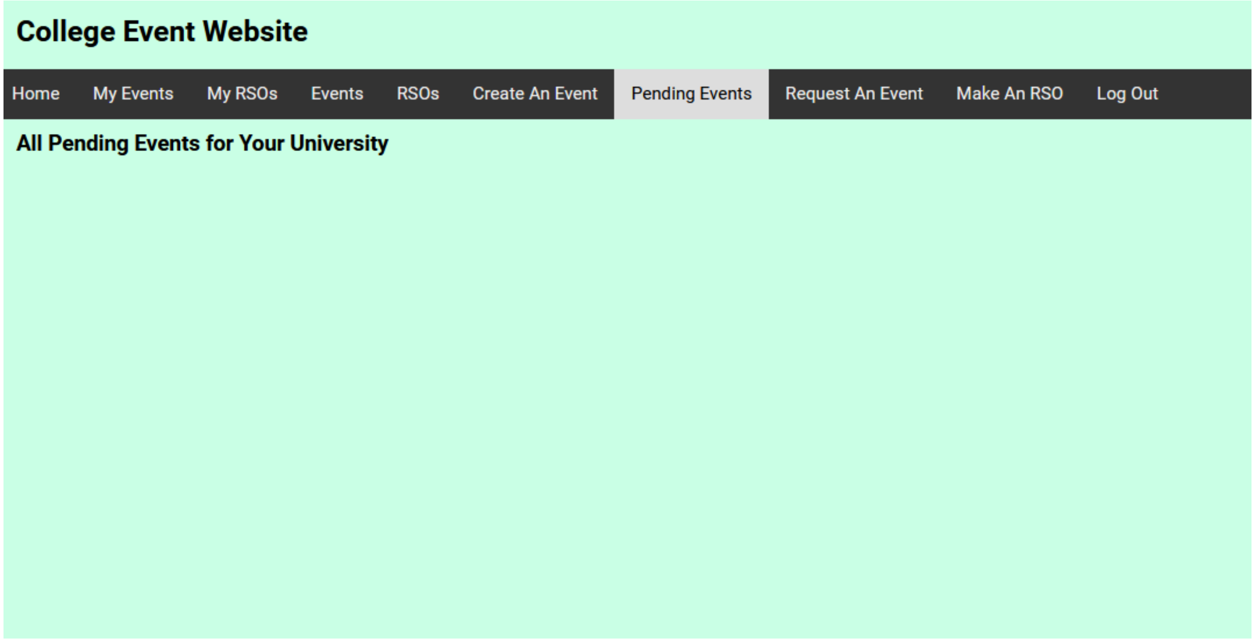
Celebrate being a Knight!

Latitude: 3
Longitude: 4

Phone #: 123-123-1233

Visibility: PUBLIC

Front-End After:



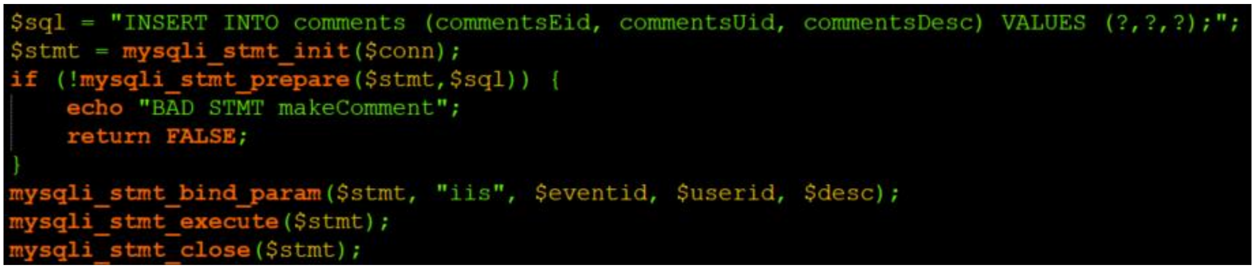
College Event Website

All Pending Events for Your University

New Comment

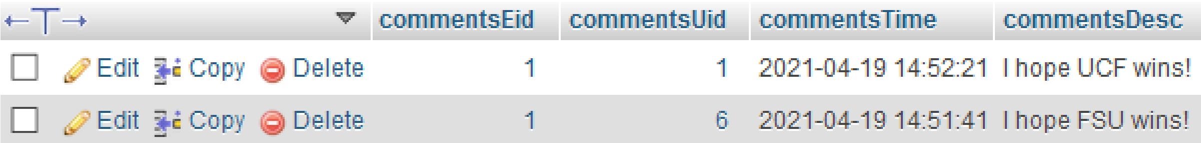
Create new comment on “Knight Life” as “jordan”.

SQL:



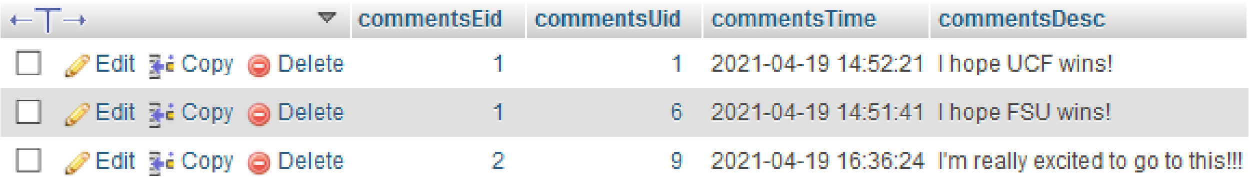
```
$sql = "INSERT INTO comments (commentsEid, commentsUid, commentsDesc) VALUES (?,?,?);";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo "BAD STMT makeComment";
    return FALSE;
}
mysqli_stmt_bind_param($stmt, "iis", $eventid, $userid, $desc);
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
```

Database Before:



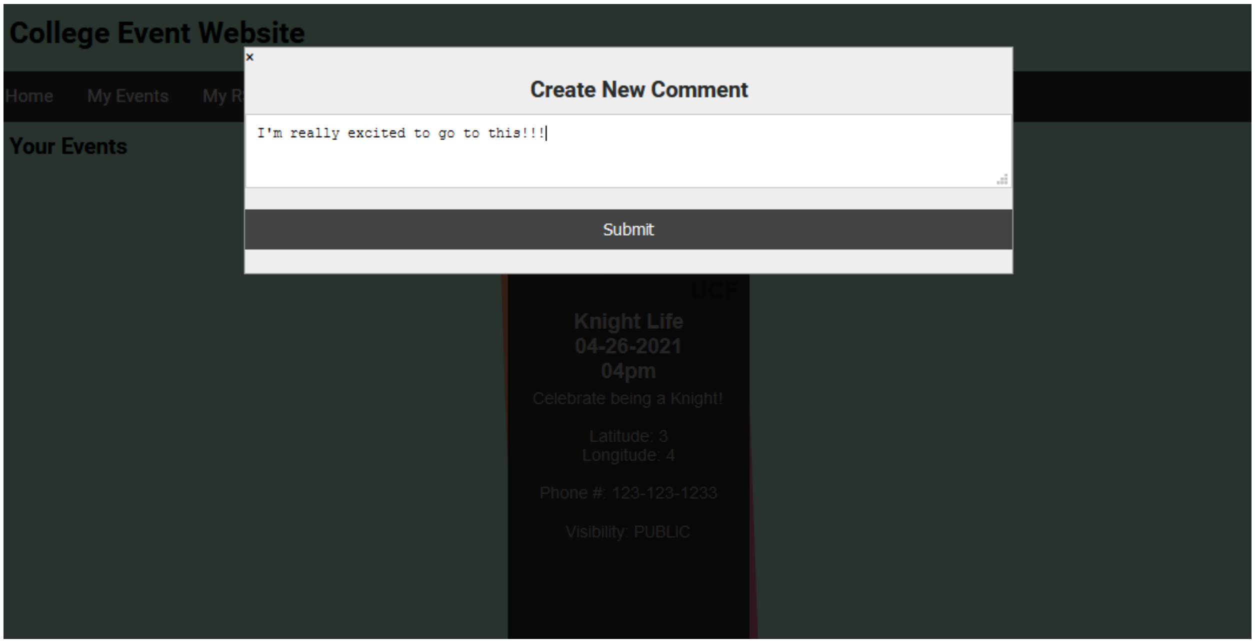
←T→		commentsEid	commentsUid	commentsTime	commentsDesc
□	Edit Copy Delete	1	1	2021-04-19 14:52:21	I hope UCF wins!
□	Edit Copy Delete	1	6	2021-04-19 14:51:41	I hope FSU wins!

Database After:



←T→		commentsEid	commentsUid	commentsTime	commentsDesc
□	Edit Copy Delete	1	1	2021-04-19 14:52:21	I hope UCF wins!
□	Edit Copy Delete	1	6	2021-04-19 14:51:41	I hope FSU wins!
□	Edit Copy Delete	2	9	2021-04-19 16:36:24	I'm really excited to go to this!!!

Front-End Before:



Create New Comment

I'm really excited to go to this!!!

UCF

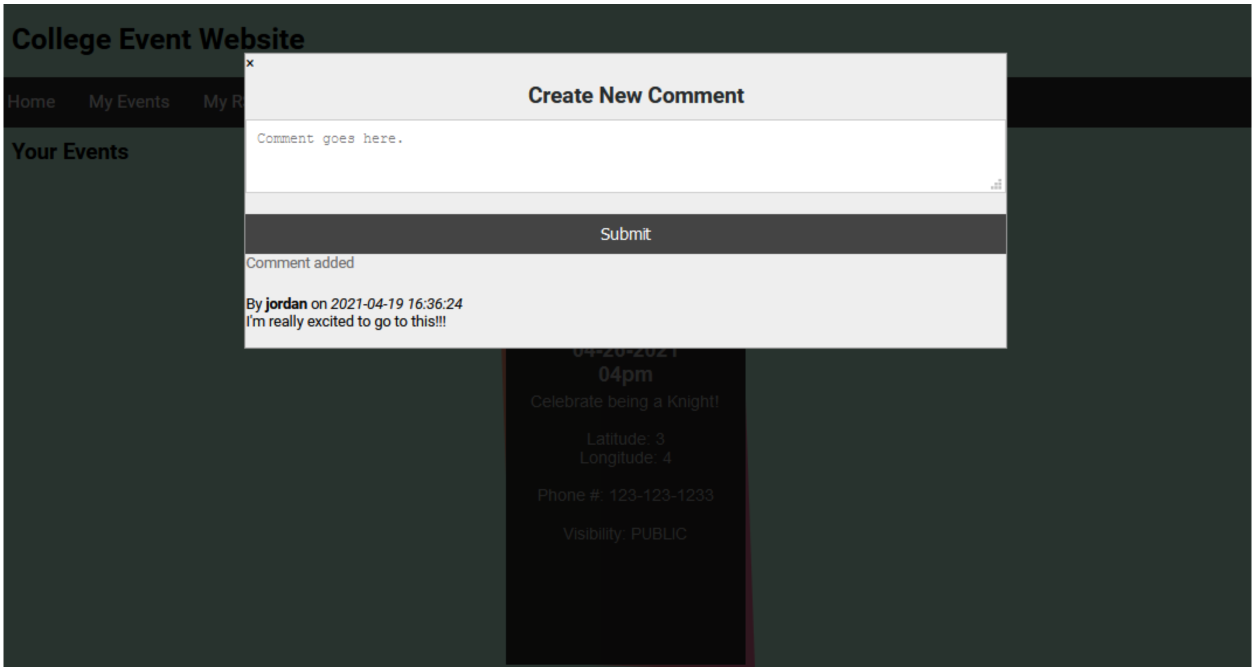
Knight Life
04-26-2021
04pm
Celebrate being a Knight!

Latitude: 3
Longitude: 4

Phone #: 123-123-1233

Visibility: PUBLIC

Front-End After:



Comment added

By **jordan** on *2021-04-19 16:36:24*
I'm really excited to go to this!!!

04-26-2021
04pm
Celebrate being a Knight!

Latitude: 3
Longitude: 4

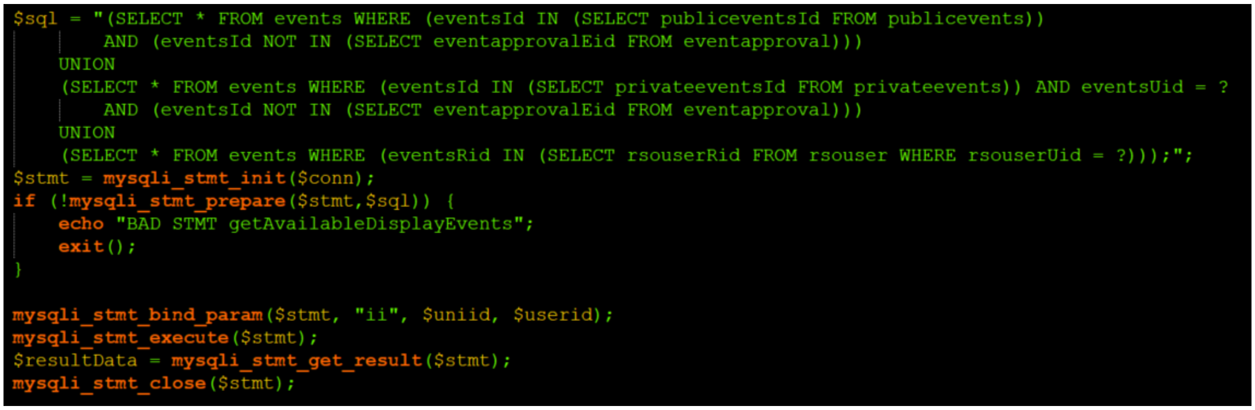
Phone #: 123-123-1233

Visibility: PUBLIC

View All Available Events

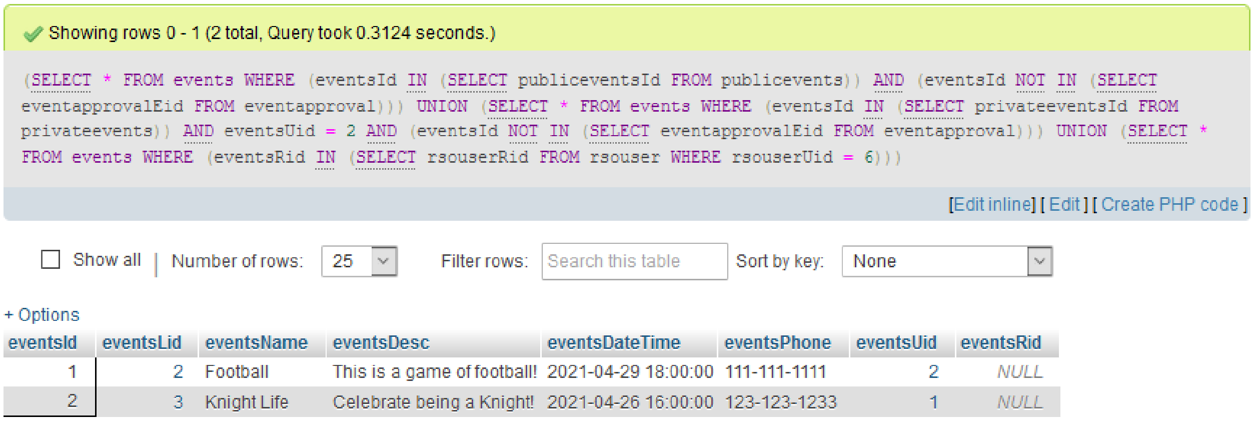
This will get all events that are available to a user. In this case, the user is "lisa".

SQL:



```
$sql = "(SELECT * FROM events WHERE (eventsId IN (SELECT publiceventsId FROM publicevents))
        AND (eventsId NOT IN (SELECT eventapprovalEid FROM eventapproval)))
    UNION
    (SELECT * FROM events WHERE (eventsId IN (SELECT privateeventsId FROM privateevents)) AND eventsUid = ?
        AND (eventsId NOT IN (SELECT eventapprovalEid FROM eventapproval)))
    UNION
    (SELECT * FROM events WHERE (eventsRid IN (SELECT rsouserRid FROM rsouser WHERE rsouserUid = ?)));";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo "BAD STMT getAvailableDisplayEvents";
    exit();
}

mysqli_stmt_bind_param($stmt, "ii", $uniid, $userid);
mysqli_stmt_execute($stmt);
$resultData = mysqli_stmt_get_result($stmt);
mysqli_stmt_close($stmt);
```

Database Result:



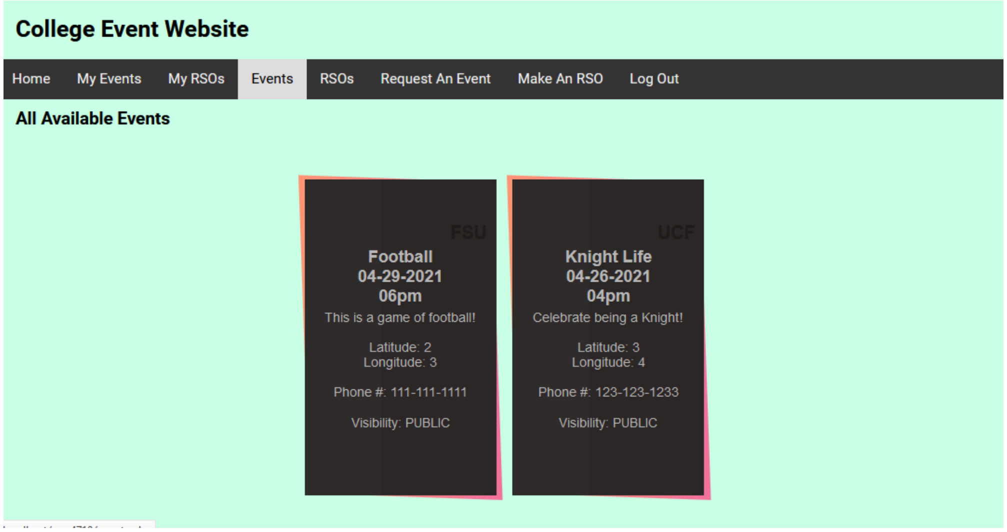
✓ Showing rows 0 - 1 (2 total, Query took 0.3124 seconds.)

```
(SELECT * FROM events WHERE (eventsId IN (SELECT publiceventsId FROM publicevents)) AND (eventsId NOT IN (SELECT
eventapprovalEid FROM eventapproval))) UNION (SELECT * FROM events WHERE (eventsId IN (SELECT privateeventsId FROM
privateevents)) AND eventsUid = 2 AND (eventsId NOT IN (SELECT eventapprovalEid FROM eventapproval))) UNION (SELECT *
FROM events WHERE (eventsRid IN (SELECT rsouserRid FROM rsouser WHERE rsouserUid = 6)))
```

☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

eventsId	eventsLid	eventsName	eventsDesc	eventsDateTime	eventsPhone	eventsUid	eventsRid
1	2	Football	This is a game of football!	2021-04-29 18:00:00	111-111-1111	2	NULL
2	3	Knight Life	Celebrate being a Knight!	2021-04-26 16:00:00	123-123-1233	1	NULL

Front-End Result:



College Event Website

All Available Events

FSU

Football
04-29-2021
06pm
This is a game of football!

Latitude: 2
Longitude: 3

Phone #: 111-111-1111

Visibility: PUBLIC

UCF

Knight Life
04-26-2021
04pm
Celebrate being a Knight!

Latitude: 3
Longitude: 4

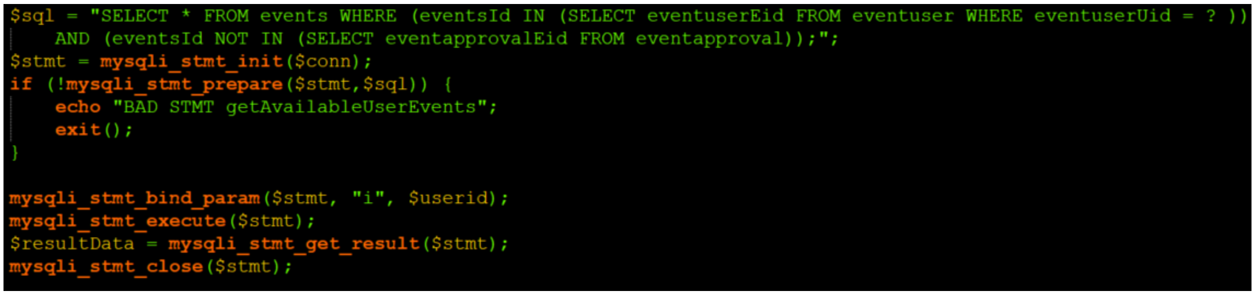
Phone #: 123-123-1233

Visibility: PUBLIC

View All User Events

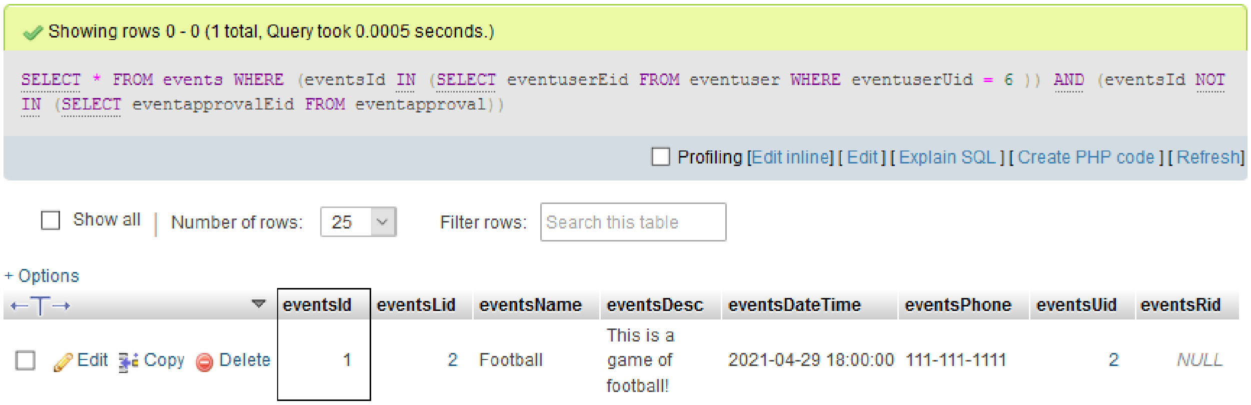
This will get all events a user is in. In this case, the user is "lisa".

SQL:



```
$sql = "SELECT * FROM events WHERE (eventsId IN (SELECT eventuserEid FROM eventuser WHERE eventuserUid = ? ))
    AND (eventsId NOT IN (SELECT eventapprovalEid FROM eventapproval));";
$stmt = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($stmt,$sql)) {
    echo "BAD STMT getAvailableUserEvents";
    exit();
}

mysqli_stmt_bind_param($stmt, "i", $userid);
mysqli_stmt_execute($stmt);
$resultData = mysqli_stmt_get_result($stmt);
mysqli_stmt_close($stmt);
```

Database Result:



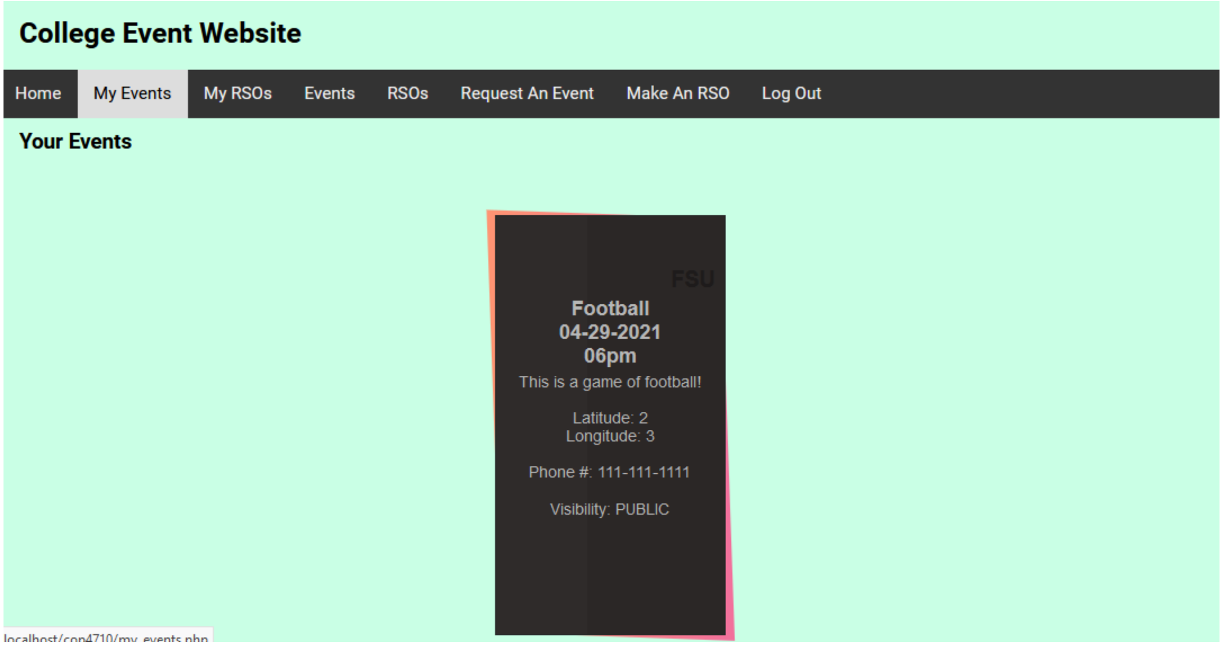
✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM events WHERE (eventsId IN (SELECT eventuserEid FROM eventuser WHERE eventuserUid = 6 )) AND (eventsId NOT IN (SELECT eventapprovalEid FROM eventapproval))
```

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

←T→	eventsId	eventsLid	eventsName	eventsDesc	eventsDateTime	eventsPhone	eventsUid	eventsRid
□ Edit Copy Delete	1	2	Football	This is a game of football!	2021-04-29 18:00:00	111-111-1111	2	NULL

Front-End Result:



Football
04-29-2021
06pm
This is a game of football!

Latitude: 2
Longitude: 3

Phone #: 111-111-1111

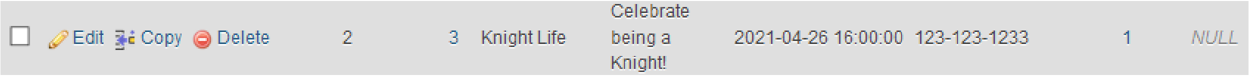
Visibility: PUBLIC

Constraint Enforcement

For this section, front-end warnings are shown for when the following actions take place.

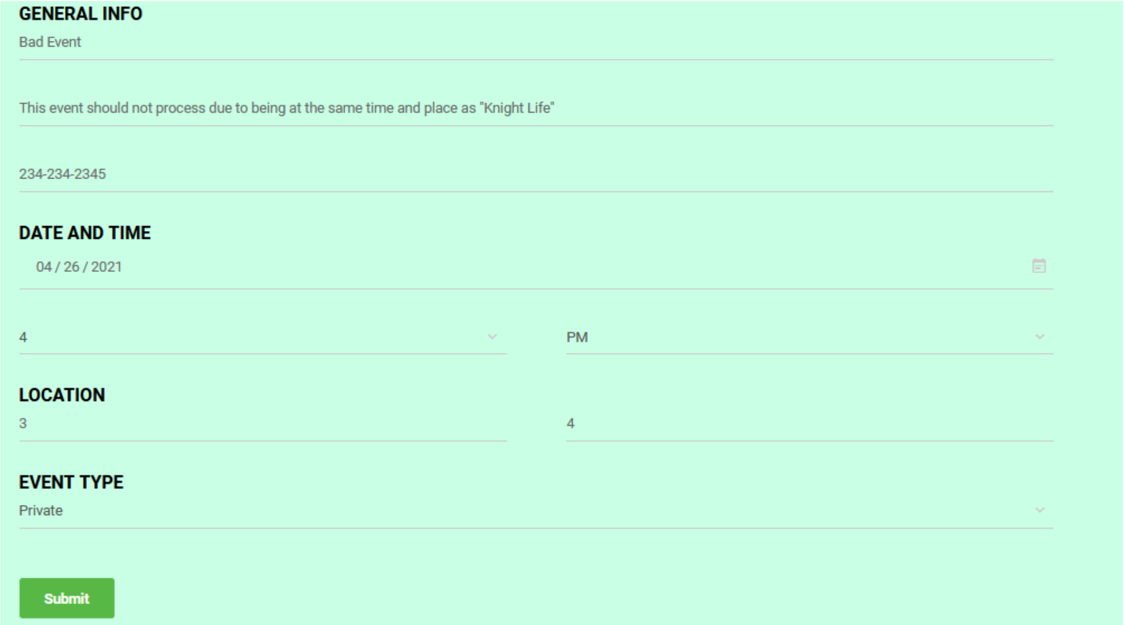
Overlapping Events

Original Event:



□	Edit	Copy	Delete	2	3	Knight Life	Celebrate being a Knight!	2021-04-26 16:00:00	123-123-1233	1	NULL

Attempted New Event:



GENERAL INFO

Bad Event

This event should not process due to being at the same time and place as "Knight Life"

234-234-2345

DATE AND TIME

04 / 26 / 2021

4

PM

LOCATION

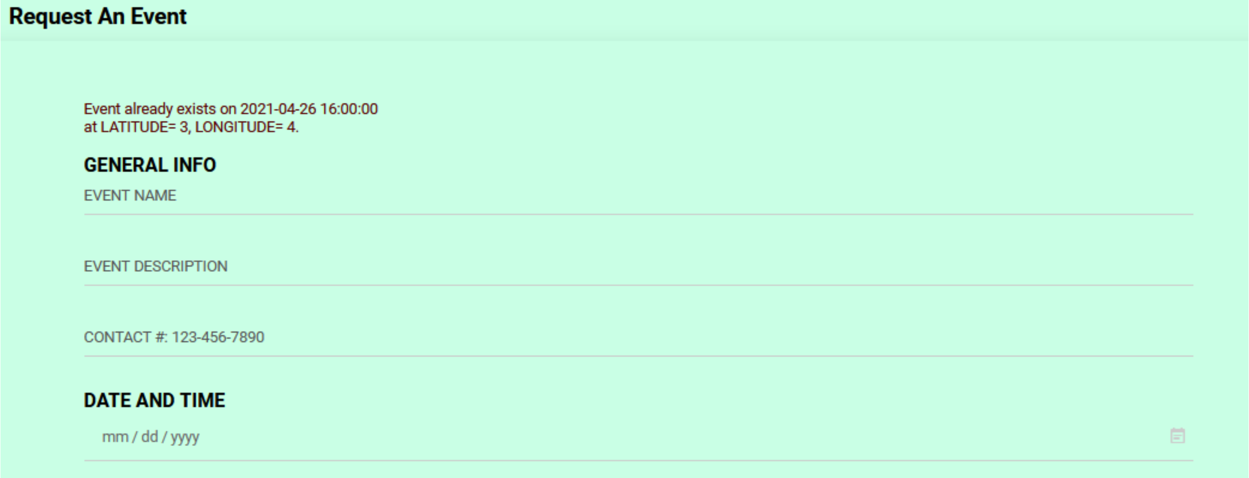
3

4

EVENT TYPE

Private

Result:



Request An Event

Event already exists on 2021-04-26 16:00:00
at LATITUDE= 3, LONGITUDE= 4.

GENERAL INFO

EVENT NAME

EVENT DESCRIPTION

CONTACT #: 123-456-7890

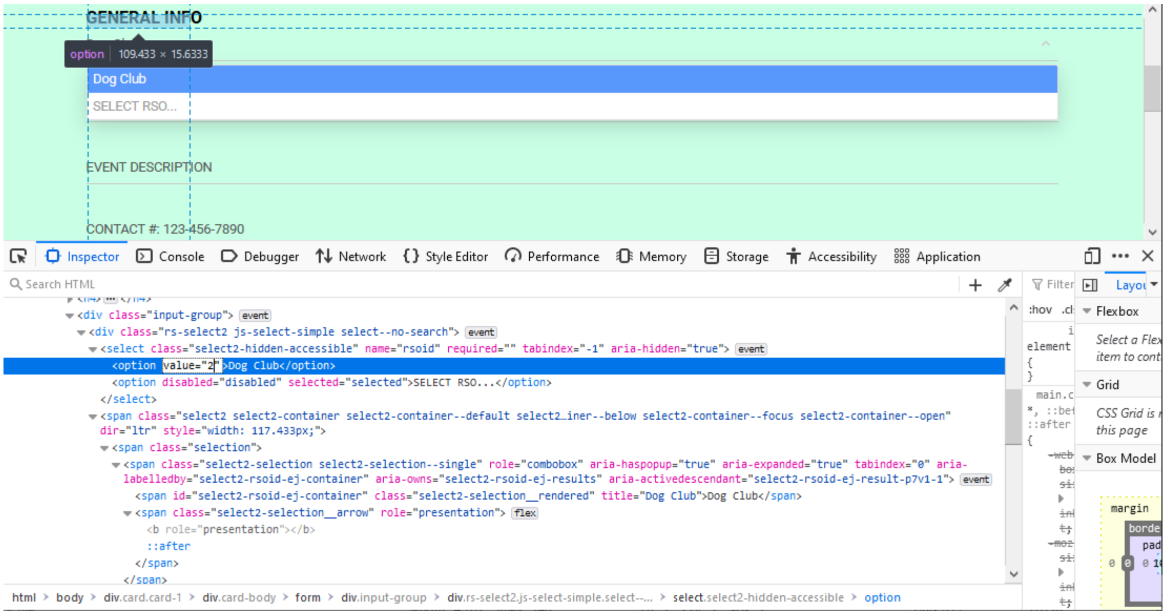
DATE AND TIME

mm / dd / yyyy

User (Other Than Appropriate Admin) Creating Event for RSO

If a user was using the site properly, this should not happen. However, if a malicious user finds an event's id and try to create an event for it, then the following will happen. Let's say "javi" was a malicious user attempting to create an event for "UCF Orchestra". (In case of confusion, at this point "Dog Club" is already active, where activation can be seen below.)

Malicious Viewer View:



GENERAL INFO

Dog Club

SELECT RSO...

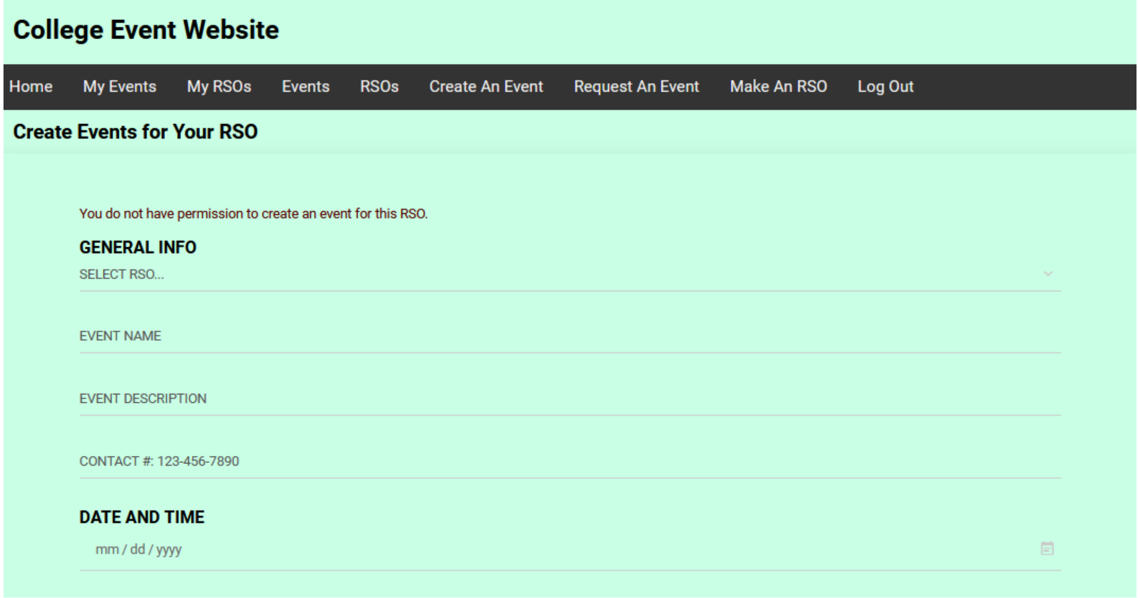
EVENT DESCRIPTION

CONTACT #: 123-456-7890

```
<div class="input-group"> event
  <div class="rs-select2 js-select-simple select--no-search"> event
    <select class="select2-hidden-accessible" name="rsoid" required="" tabindex="-1" aria-hidden="true"> event
      <option value="2">Dog Club</option>
      <option disabled="disabled" selected="selected">SELECT RSO...</option>
    </select>
    <span class="select2 select2-container select2-container--default select2…iner--below select2-container--focus select2-container--open"
    dir="ltr" style="width: 117.433px;">
      <span class="selection">
        <span class="select2-selection select2-selection--single" role="combobox" aria-haspopup="true" aria-expanded="true" tabindex="0" aria-
        labelledby="select2-rsoid-ej-container" aria-owns="select2-rsoid-ej-results" aria-activedescendant="select2-rsoid-ej-result-p7v1-1"> event
          <span id="select2-rsoid-ej-container" class="select2-selection__rendered" title="Dog Club">Dog Club</span>
          <span class="select2-selection__arrow" role="presentation"> flex
            <b role="presentation"></b>
            ::after
          </span>
        </span>
```

html > body > div.card.card-1 > div.card-body > form > div.input-group > div.rs-select2.js-select-simple.select--... > select.select2-hidden-accessible > option

Result:



Create Events for Your RSO

You do not have permission to create an event for this RSO.

GENERAL INFO

SELECT RSO...

EVENT NAME

EVENT DESCRIPTION

CONTACT #: 123-456-7890

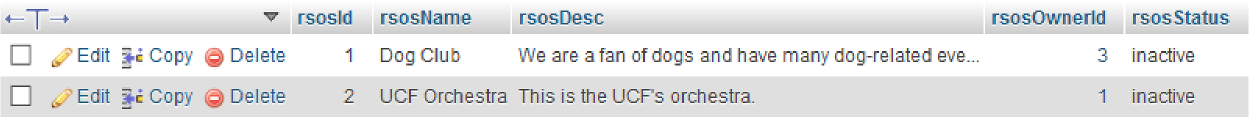
DATE AND TIME

mm / dd / yyyy

RSO Becoming Active

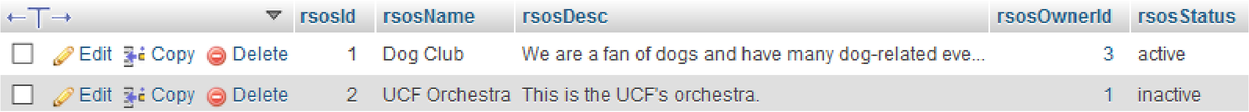
Though no active warning shows up, once the 5th user joins the RSO, the admin will have the option available to them for picking the RSO in “Create An Event”. In this case, “josh” will join “Dog Club”, thus activating it.

Database View Before:



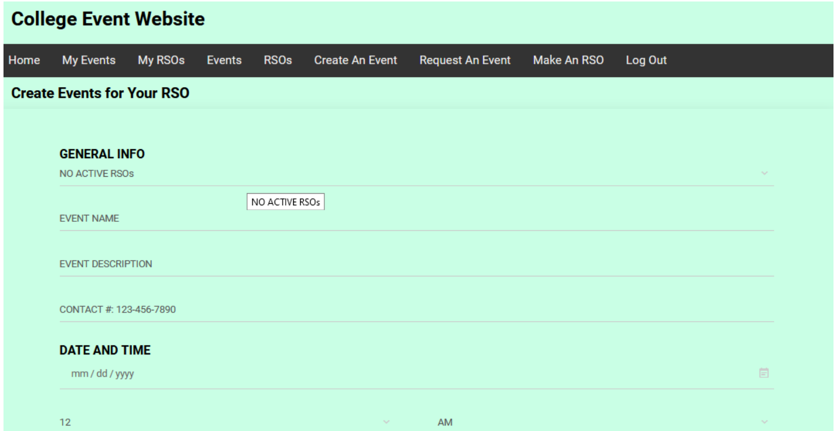
←T→		rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□	Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	inactive
□	Edit Copy Delete	2	UCF Orchestra	This is the UCF's orchestra.	1	inactive

Database View After:



←T→		rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□	Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	active
□	Edit Copy Delete	2	UCF Orchestra	This is the UCF's orchestra.	1	inactive

Admin View Before:



Create Events for Your RSO

GENERAL INFO

NO ACTIVE RSOs

NO ACTIVE RSOs

EVENT NAME

EVENT DESCRIPTION

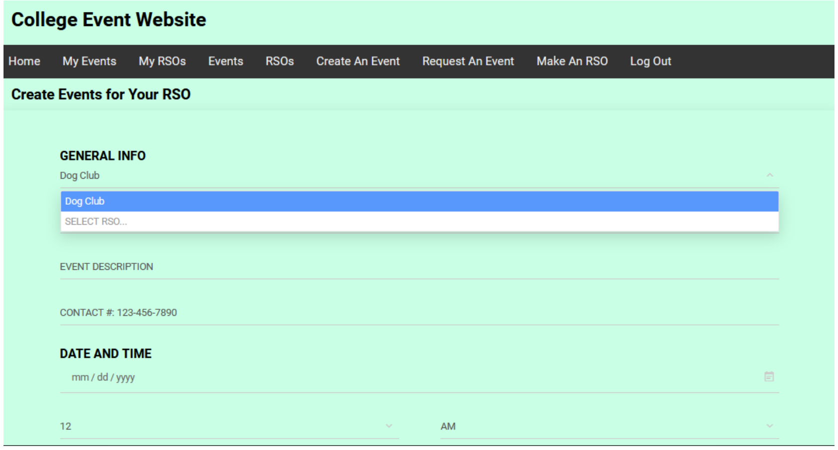
CONTACT #: 123-456-7890

DATE AND TIME

mm / dd / yyyy

12 AM

Admin View After:



Create Events for Your RSO

GENERAL INFO

Dog Club

Dog Club

SELECT RSO...

EVENT DESCRIPTION

CONTACT #: 123-456-7890

DATE AND TIME

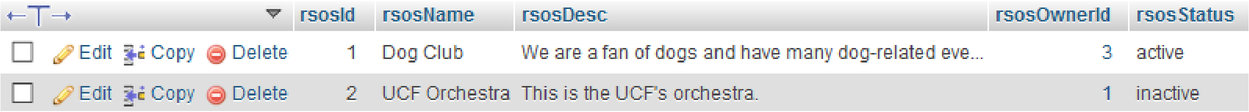
mm / dd / yyyy

12 AM

RSO Becoming Inactive

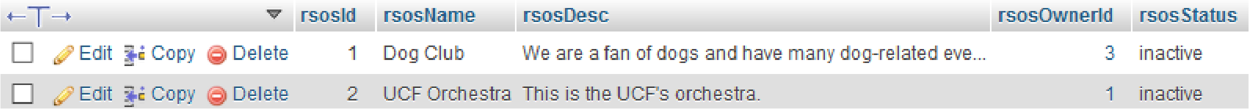
Opposite to before, if there becomes 4 or less people, the effects of being able to create events is reversed. Here, “josh” will leave “Dog Club”.

Database View Before:



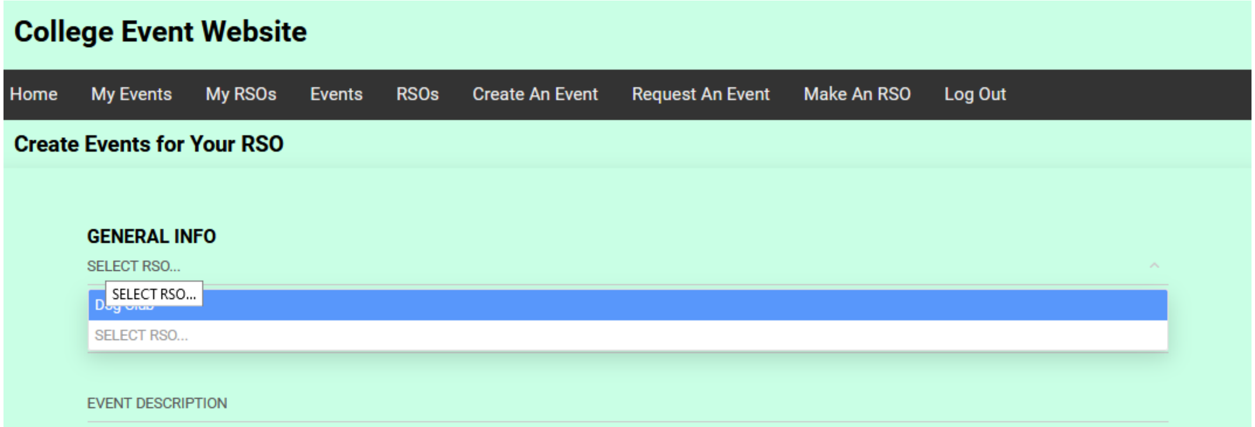
←T→ ▼	rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□ Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	active
□ Edit Copy Delete	2	UCF Orchestra	This is the UCF's orchestra.	1	inactive

Database View After:



←T→ ▼	rsosId	rsosName	rsosDesc	rsosOwnerId	rsosStatus
□ Edit Copy Delete	1	Dog Club	We are a fan of dogs and have many dog-related eve...	3	inactive
□ Edit Copy Delete	2	UCF Orchestra	This is the UCF's orchestra.	1	inactive

Admin View Before:



College Event Website

Create Events for Your RSO

GENERAL INFO

SELECT RSO...

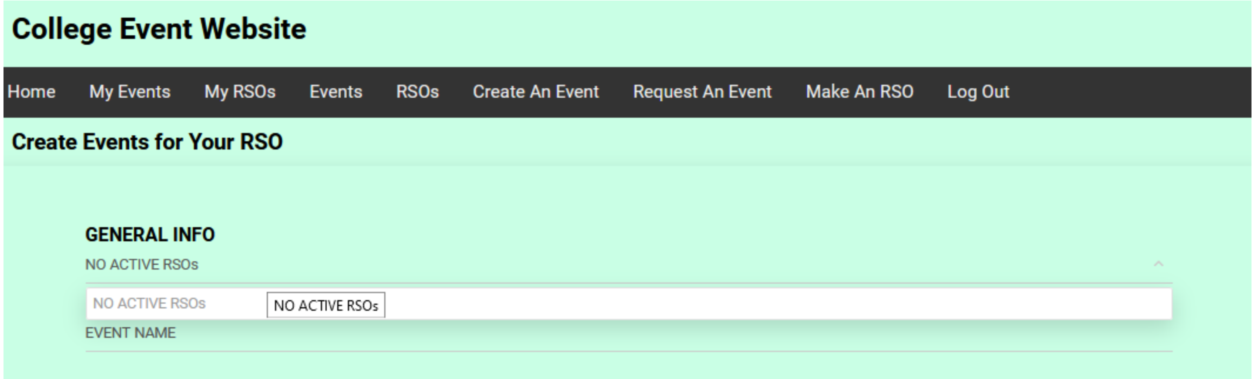
SELECT RSO...

Dog Club

SELECT RSO...

EVENT DESCRIPTION

Admin View After:



College Event Website

Create Events for Your RSO

GENERAL INFO

NO ACTIVE RSOs

NO ACTIVE RSOs

NO ACTIVE RSOs

EVENT NAME

Conclusion

The database, on simple queries, appears to give results back quickly (<0.001s). The most complicated query, the one that returns all events that are visible to a user, takes around 0.3s, which is relatively slow. Overall, this is an efficient database.

In terms of security, there are some holes that exist within the implementation that could allow for some malicious activity, but user privacy (their passwords) are encrypted using PHP's standard encryption. This means that user credentials are secure.

This project was difficult for me, as it is the first real website that I have developed in 4years. Even then, the websites that I created before were only static and had no real client-server interactions, so that increased the complexity further. I also have only had a small amount of database experience in the form of creating a bot that used MySQL for a couple of items. Overall, my expertise did not lie within this topic, so it felt daunting to do.

This does mean I learned a lot, however. I have much more experience with both JS and PHP now, as well as a knowledge of website/database security and user management. I learned how to style pages easier with CSS and understand better how web-based applications work in general. I am extremely satisfied with the work that I put in this project.