Josh Dillan E.

Project Report: College Event Website

Table of Contents

Section 1: Project Description

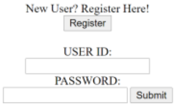
This project is a web application that displays information for events from multiple universities that users can view. Users can register an account with a name, password, email, phone number, and their university they are currently attending. Users are then alerted that their account has been created with a UID, which they need to remember in order to log in with their password.

Events can be public, private, or associated with a registered student organization (RSO). Events also have a name, description, date, start time, end time, location, university that is hosting it, contact phone and email address, and possibly a RSO that is hosting it. Public events are visible to all users, while private events are only visible to users that are attending the university that is hosting it. An RSO event is only visible to users that are members of that RSO. An RSO has an admin user, who manages requests to join the RSO. A normal user can request to join or create a new RSO with 4 other users. If the request to create an RSO is approved by a super admin, one of the 5 members of this new RSO is made an admin. Normal users can also view RSOs they are members of and choose to leave them if desired. Normal users can additionally add/edit/delete their own comments and leave a rating for an event. An RSO is considered active if there are at least 5 members, otherwise it is considered inactive. A super admin can create a university profile in the web application. Only admins can create events. Additionally, to create a RSO event, the admin must be a member of that RSO.

Section 2: GUI/Development Information

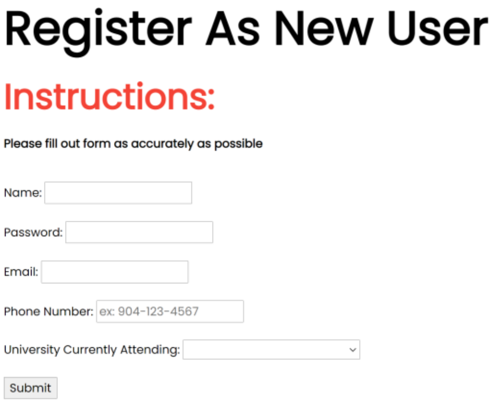
The platform used to develop the web application was Visual Studio code. The programming and markup languages used were Javascript, HTML, and SQL. The Database Management System used was Microsoft SQL Server Management Studio. The following screenshots are the user interfaces in the web application with labels.

Figure 1: Log in



New User? Register Here!

Figure 2: User Registration



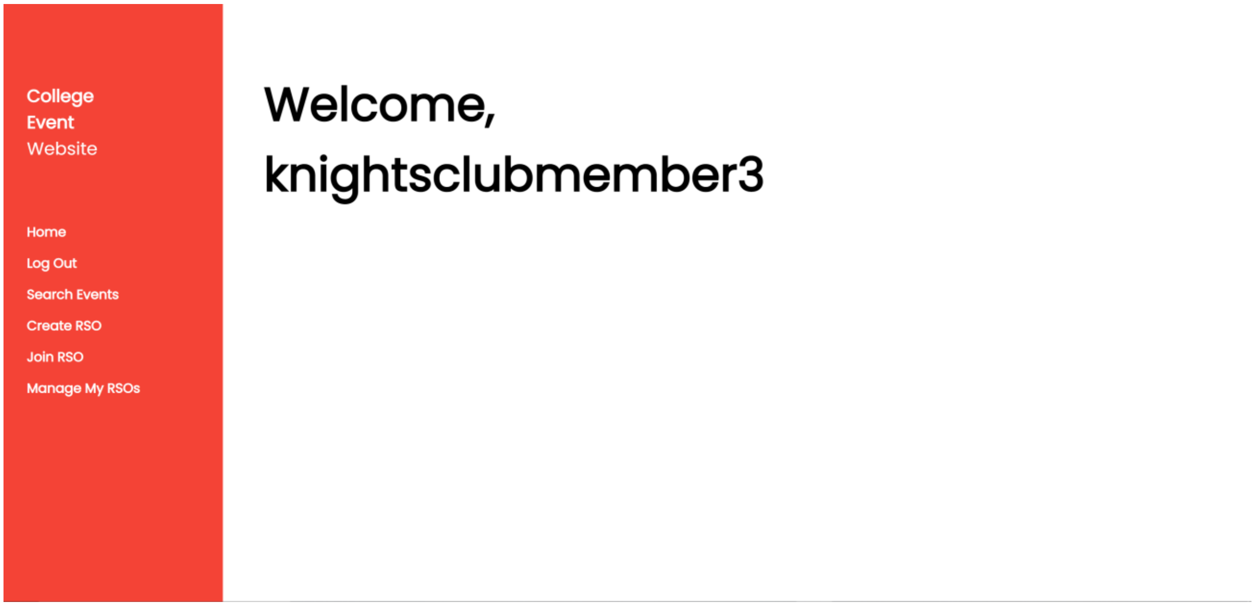
Register As New User

Instructions:

Please fill out form as accurately as possible

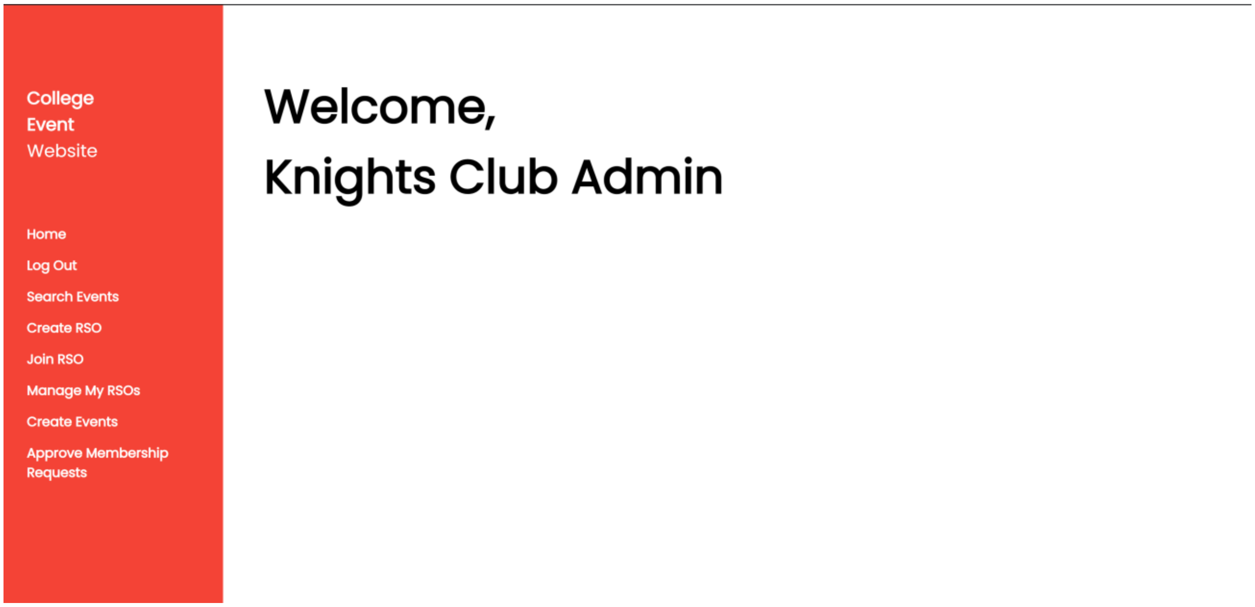
Name:
Password:
Email:
Phone Number: ex: 904-123-4567
University Currently Attending:
Submit

Figure 3: Normal User Home Screen



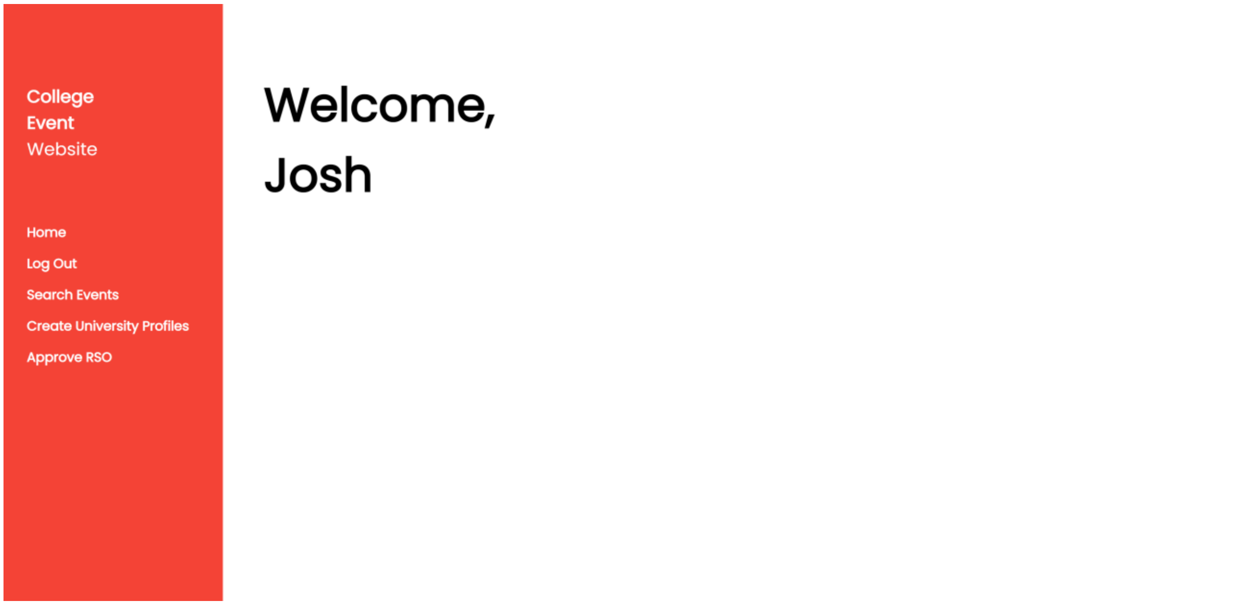
Welcome,
knightsclubmember3

Figure 4: Admin User Home Screen



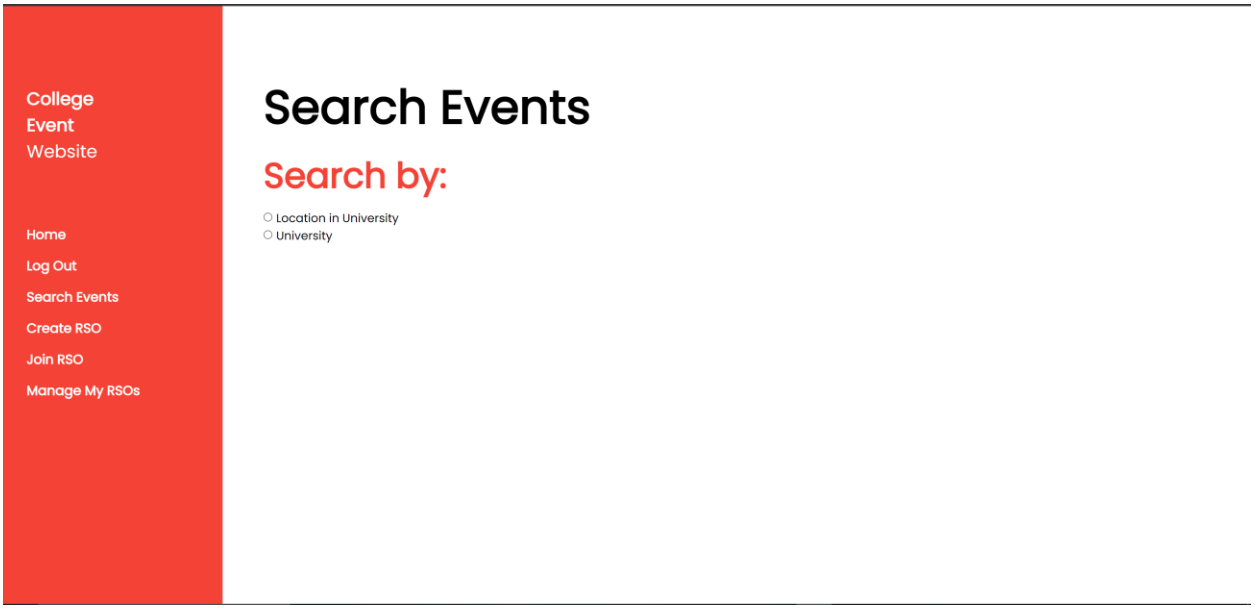
Welcome,
Knights Club Admin

Figure 5: Super Admin Home Screen



Welcome,
Josh

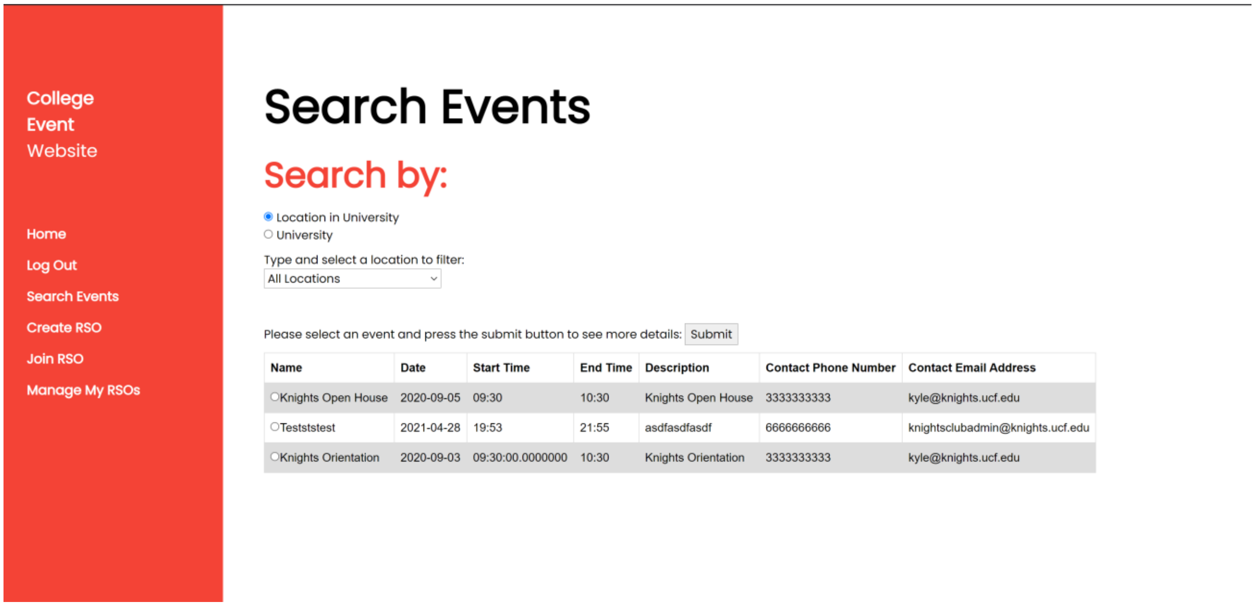
Figure 6 and 7: Search Events



Search Events

Search by:

○ Location in University
○ University



Search Events

Search by:

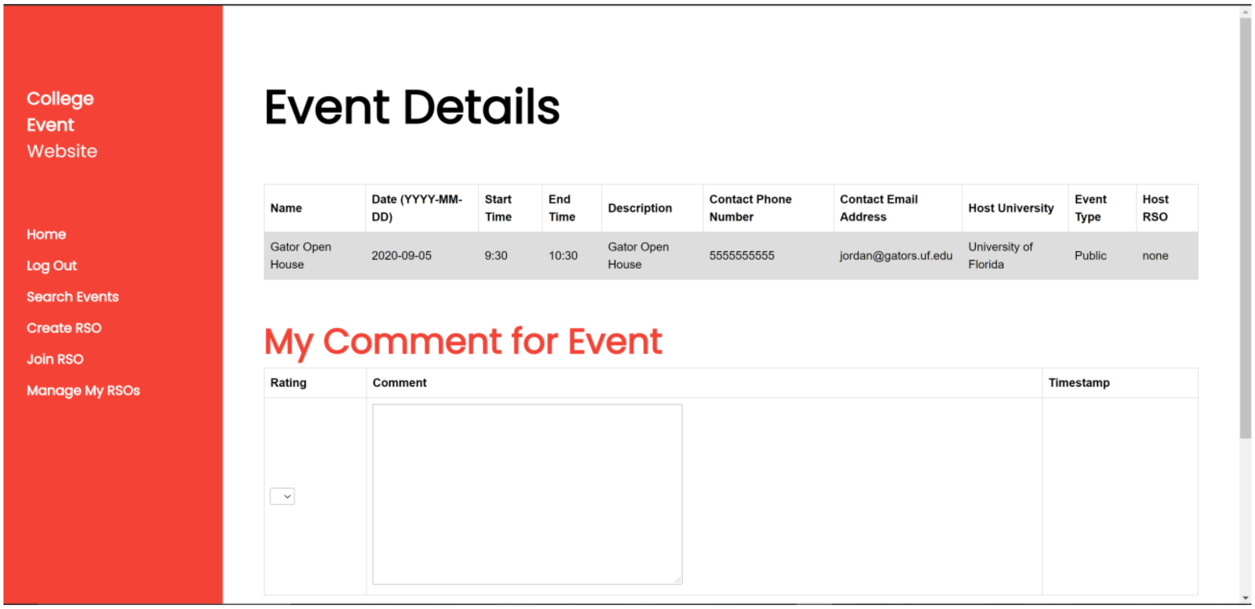
● Location in University
○ University

Type and select a location to filter: All Locations

Please select an event and press the submit button to see more details: Submit

Name	Date	Start Time	End Time	Description	Contact Phone Number	Contact Email Address
○Knights Open House	2020-09-05	09:30	10:30	Knights Open House	3333333333	kyle@knights.ucf.edu
○Teststestest	2021-04-28	19:53	21:55	asdfasdfasdf	6666666666	knightsclubadmin@knights.ucf.edu
○Knights Orientation	2020-09-03	09:30:00.0000000	10:30	Knights Orientation	3333333333	kyle@knights.ucf.edu

Figure 8 and 9: Event Information and Comments

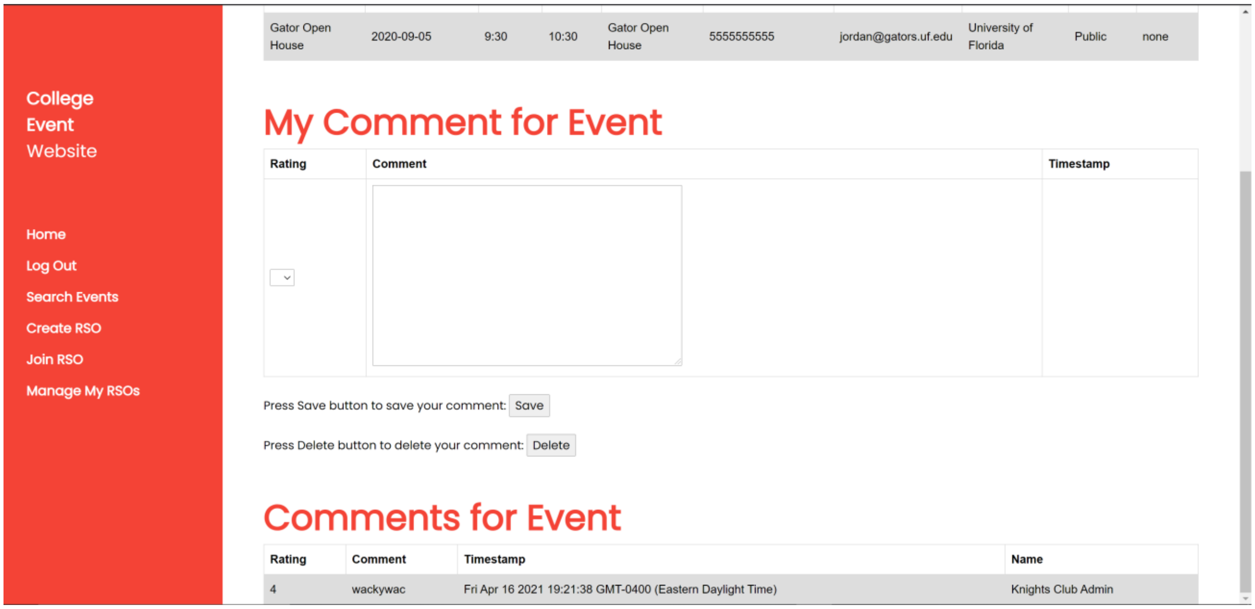


Event Details

Name	Date (YYYY-MM-DD)	Start Time	End Time	Description	Contact Phone Number	Contact Email Address	Host University	Event Type	Host RSO
Gator Open House	2020-09-05	9:30	10:30	Gator Open House	5555555555	jordan@gators.uf.edu	University of Florida	Public	none

My Comment for Event

Rating	Comment	Timestamp



Gator Open House	2020-09-05	9:30	10:30	Gator Open House	5555555555	jordan@gators.uf.edu	University of Florida	Public	none

My Comment for Event

Rating	Comment	Timestamp

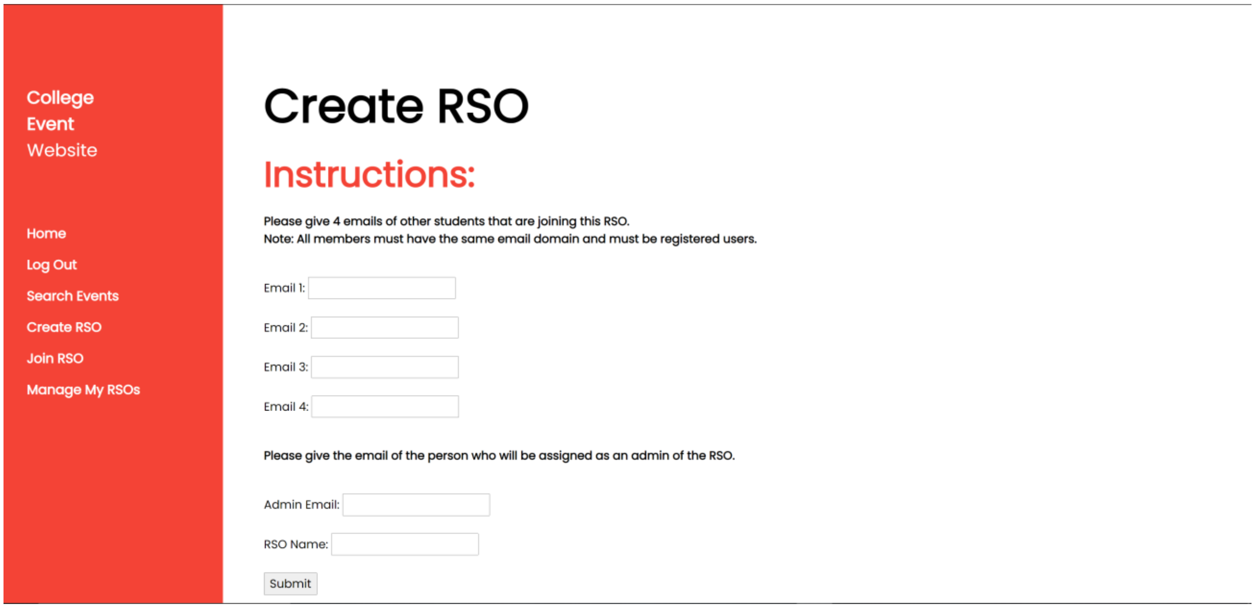
Press Save button to save your comment: Save

Press Delete button to delete your comment: Delete

Comments for Event

Rating	Comment	Timestamp	Name
4	wackywac	Fri Apr 16 2021 19:21:38 GMT-0400 (Eastern Daylight Time)	Knights Club Admin

Figure 10: Create RSO



Create RSO

Instructions:

Please give 4 emails of other students that are joining this RSO.
Note: All members must have the same email domain and must be registered users.

Email 1:

Email 2:

Email 3:

Email 4:

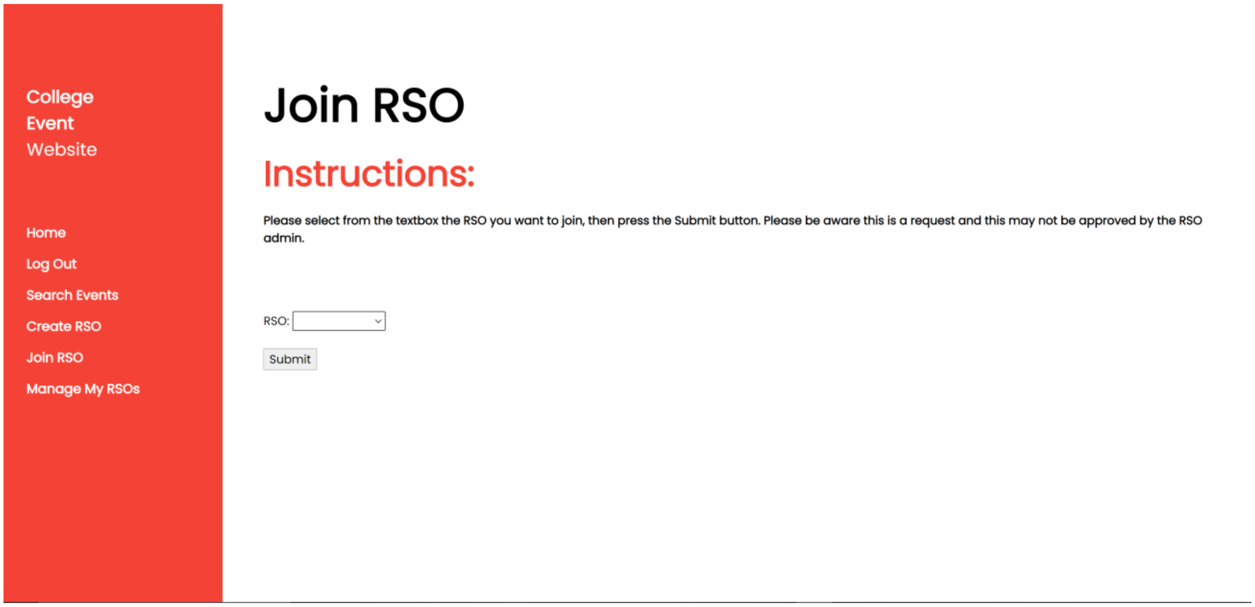
Please give the email of the person who will be assigned as an admin of the RSO.

Admin Email:

RSO Name:

Submit

Figure 11: Join RSO

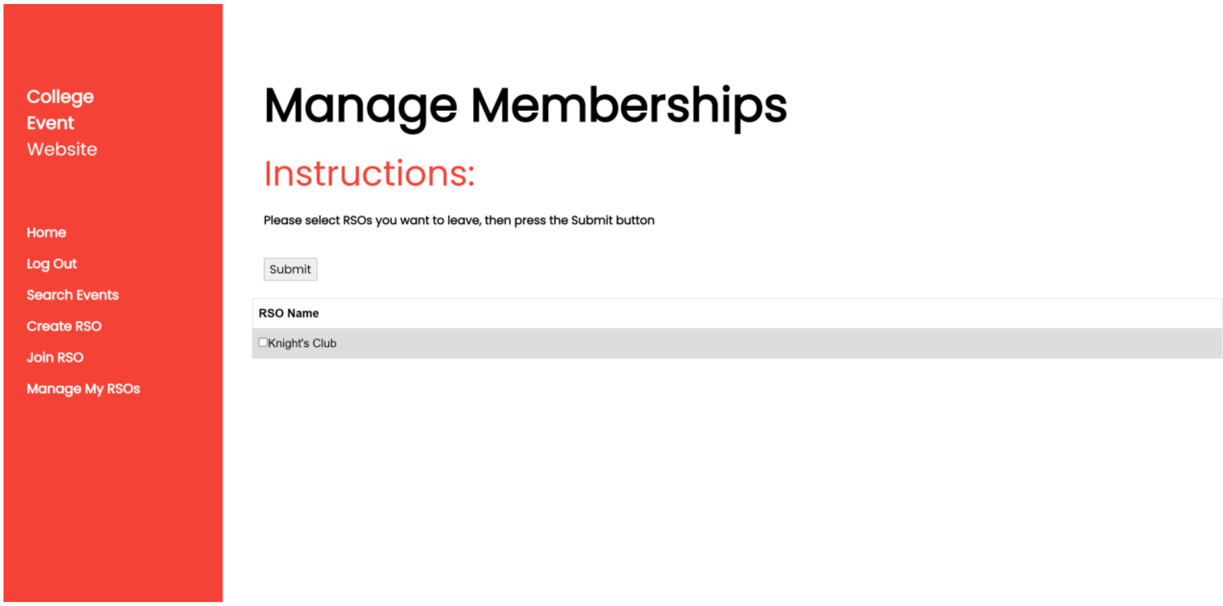


Join RSO

Instructions:

Please select from the textbox the RSO you want to join, then press the Submit button. Please be aware this is a request and this may not be approved by the RSO admin.

Figure 12: Manage my RSOs



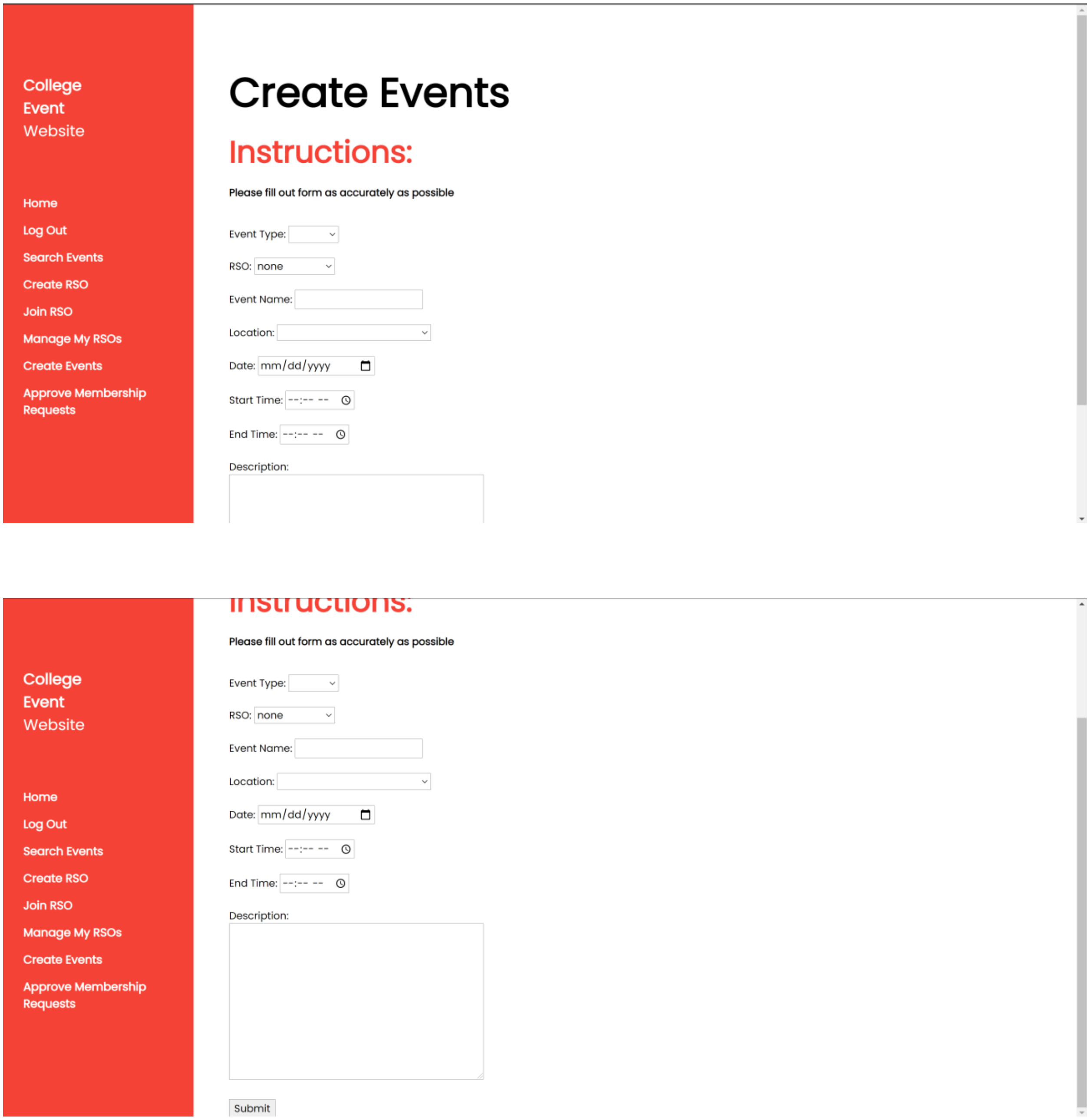
Manage Memberships

Instructions:

Please select RSOs you want to leave, then press the Submit button

RSO Name
□Knight's Club

Figure 13 and 14: Create Events



Create Events

Instructions:

Please fill out form as accurately as possible

Event Type:

RSO: none

Event Name:

Location:

Date: mm/dd/yyyy

Start Time: --:-- --

End Time: --:-- --

Description:

Instructions:

Please fill out form as accurately as possible

Event Type:

RSO: none

Event Name:

Location:

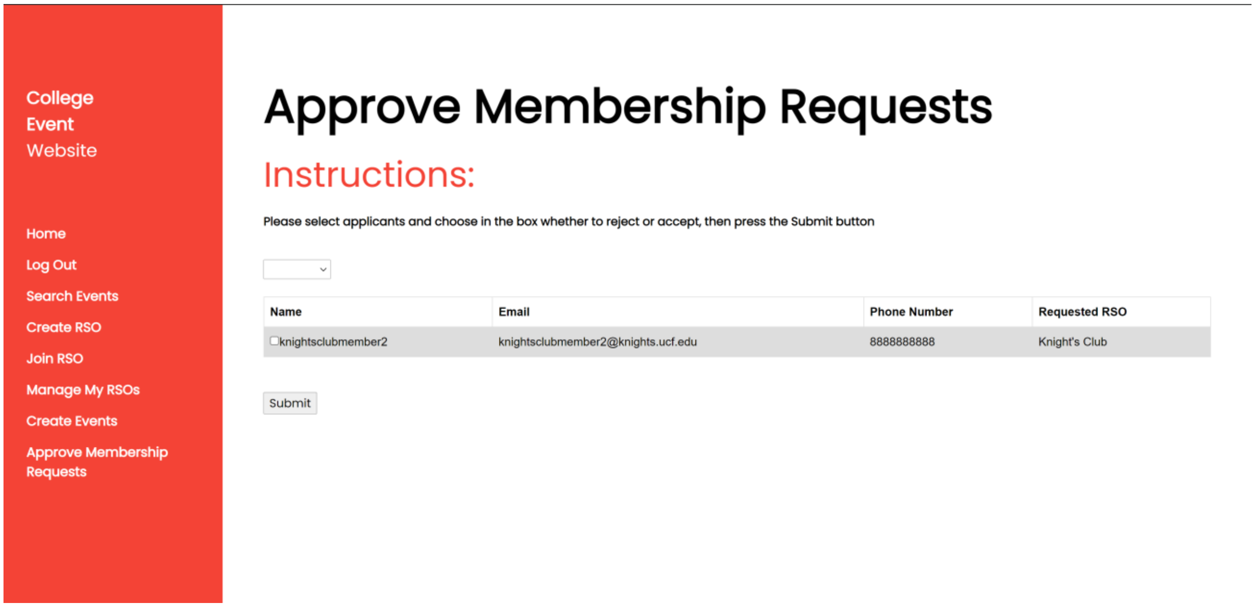
Date: mm/dd/yyyy

Start Time: --:-- --

End Time: --:-- --

Description:

Figure 15: Approve Membership Requests



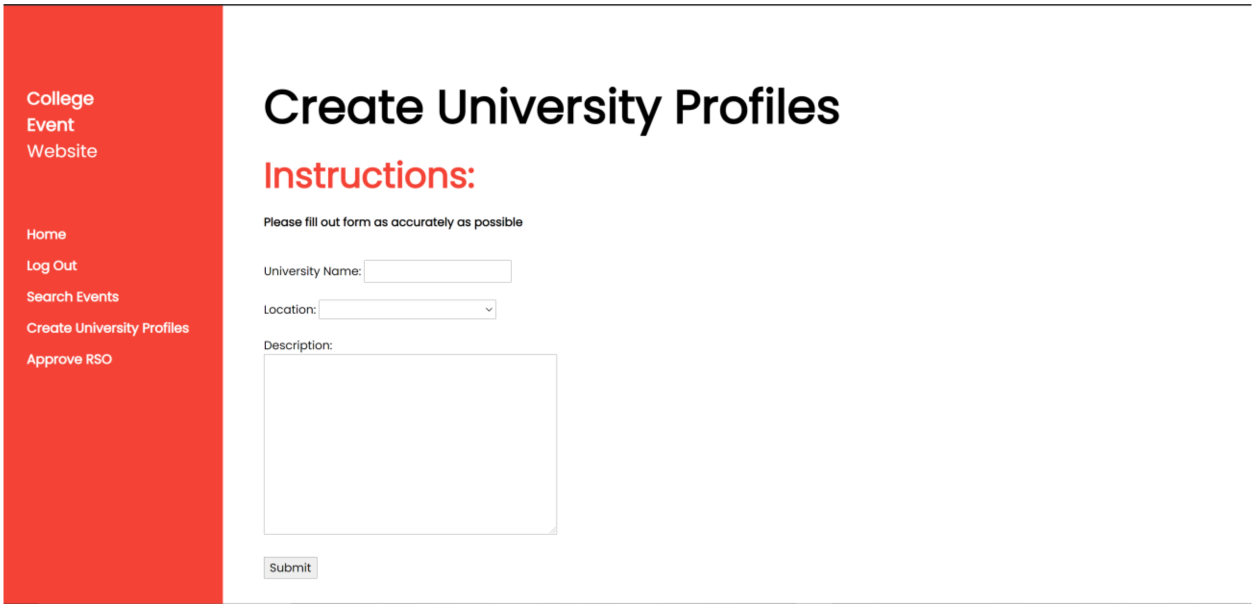
Approve Membership Requests

Instructions:

Please select applicants and choose in the box whether to reject or accept, then press the Submit button

Name	Email	Phone Number	Requested RSO
□knightsclubmember2	knightsclubmember2@knights.ucf.edu	8888888888	Knight's Club

Figure 16: Create University Profiles



Create University Profiles

Instructions:

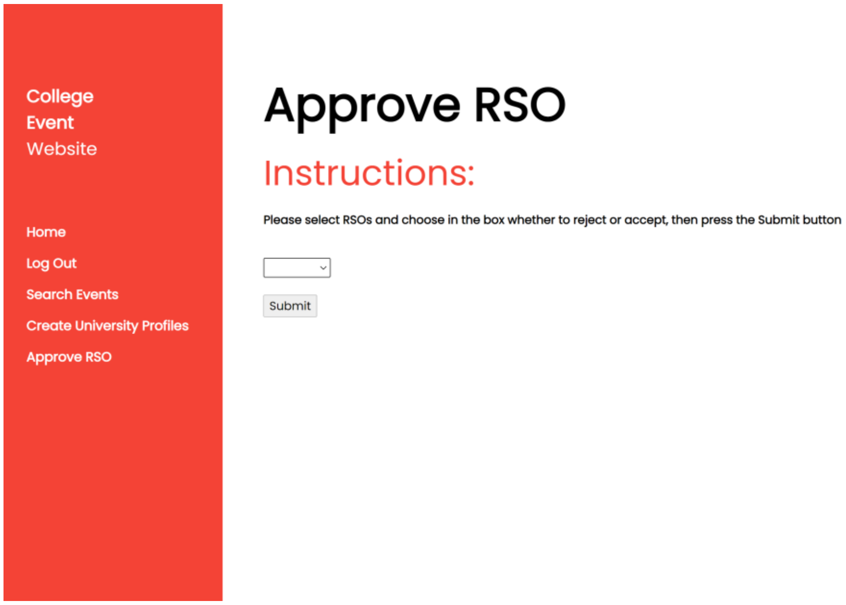
Please fill out form as accurately as possible

University Name:

Location:

Description:

Figure 17: Approve RSO

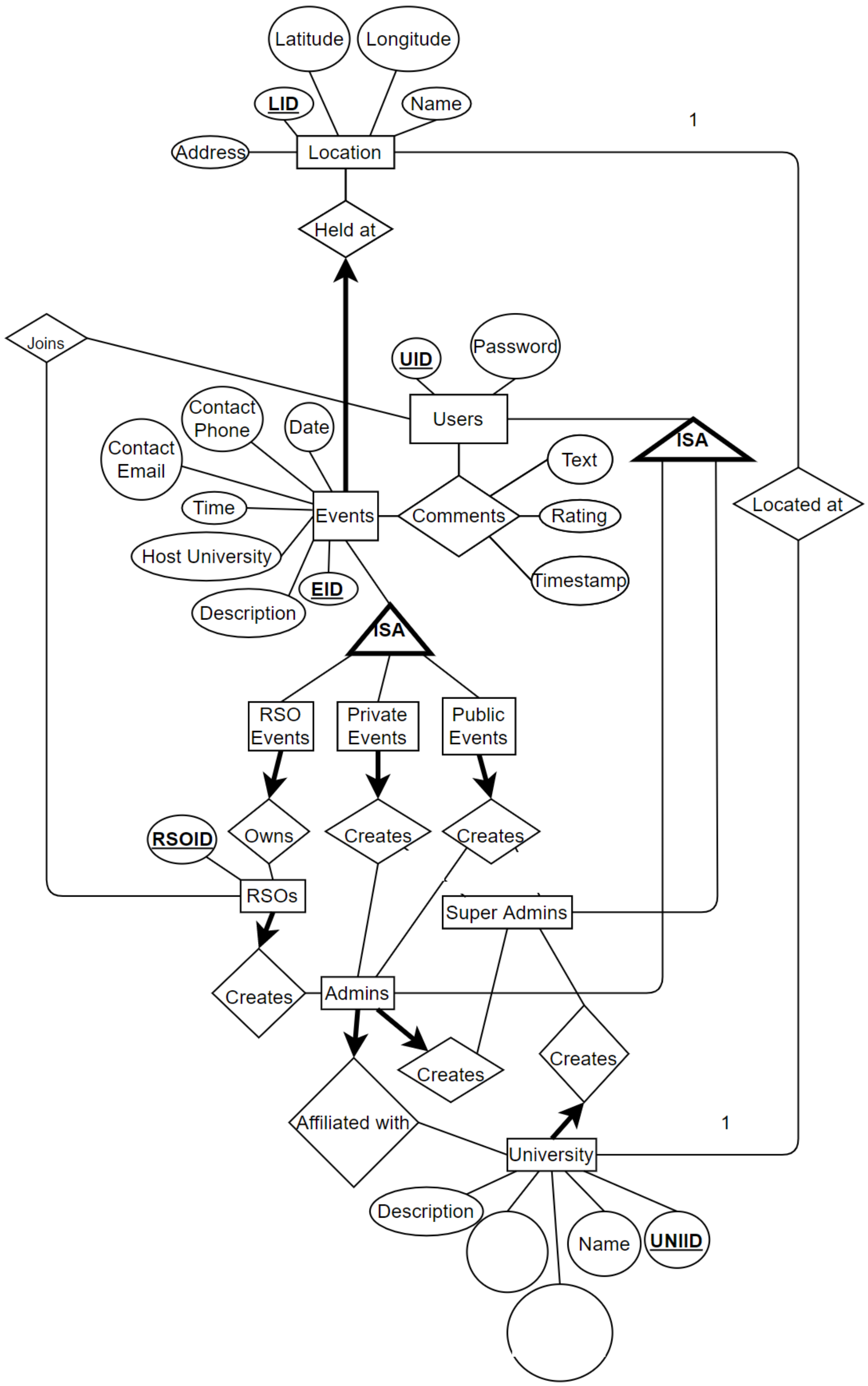


Approve RSO

Instructions:

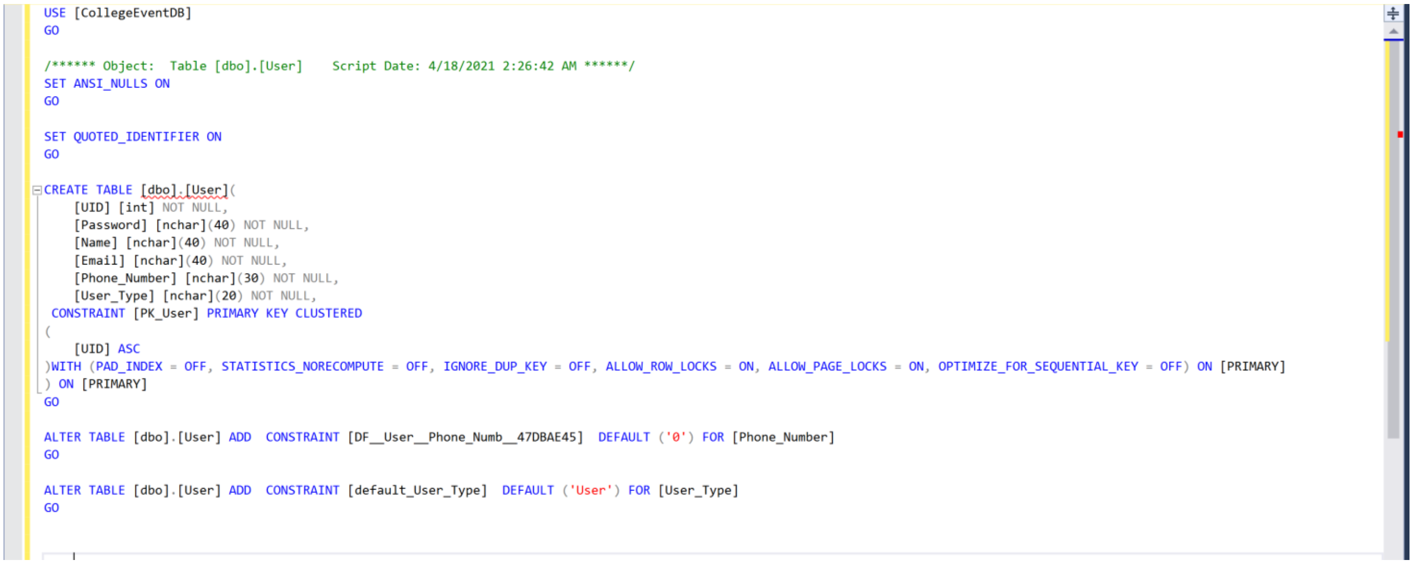
Please select RSOs and choose in the box whether to reject or accept, then press the Submit button

Section 3: ER-Model



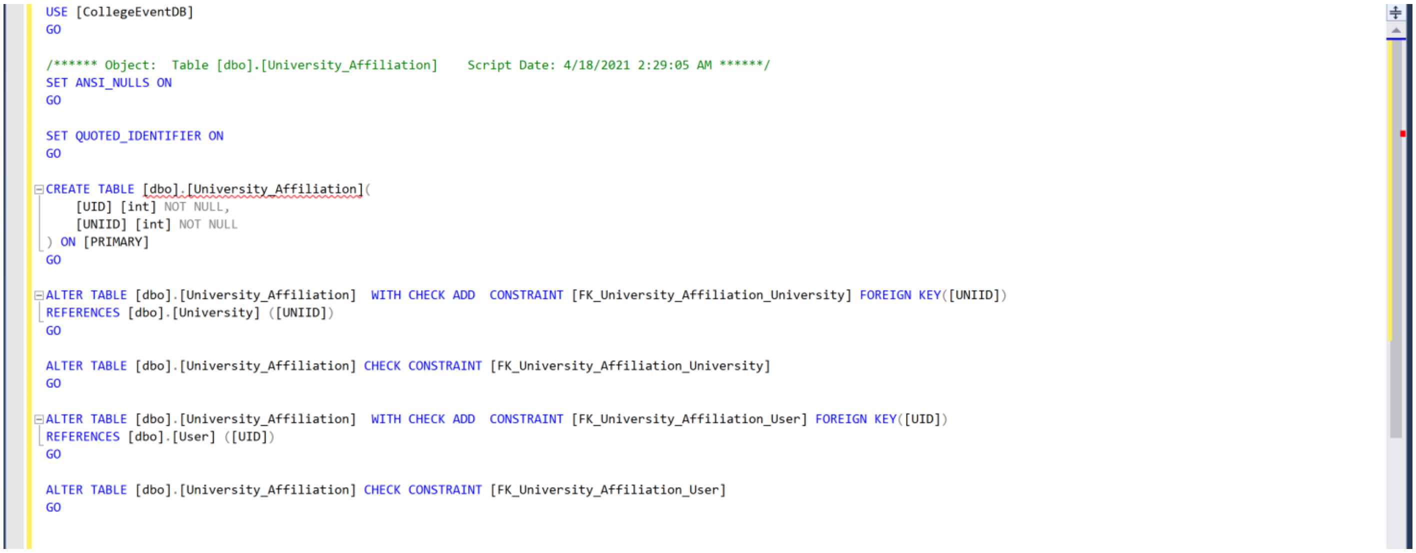
Section 4: Relational Data Model

Figure 1: User Table



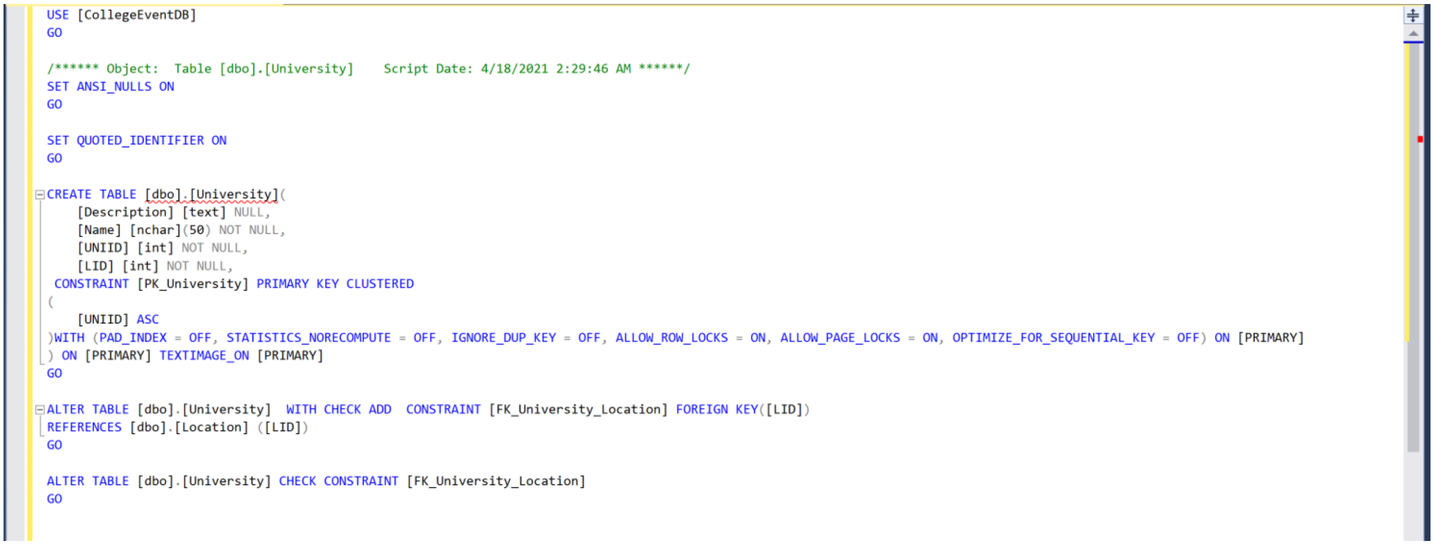
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[User]    Script Date: 4/18/2021 2:26:42 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[User](
    [UID] [int] NOT NULL,
    [Password] [nchar](40) NOT NULL,
    [Name] [nchar](40) NOT NULL,
    [Email] [nchar](40) NOT NULL,
    [Phone_Number] [nchar](30) NOT NULL,
    [User_Type] [nchar](20) NOT NULL,
 CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
(
    [UID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[User] ADD  CONSTRAINT [DF__User__Phone_Numb__47DBAE45]  DEFAULT ('0') FOR [Phone_Number]
GO

ALTER TABLE [dbo].[User] ADD  CONSTRAINT [default_User_Type]  DEFAULT ('User') FOR [User_Type]
GO
```

Figure 2: University_Affliliation Table



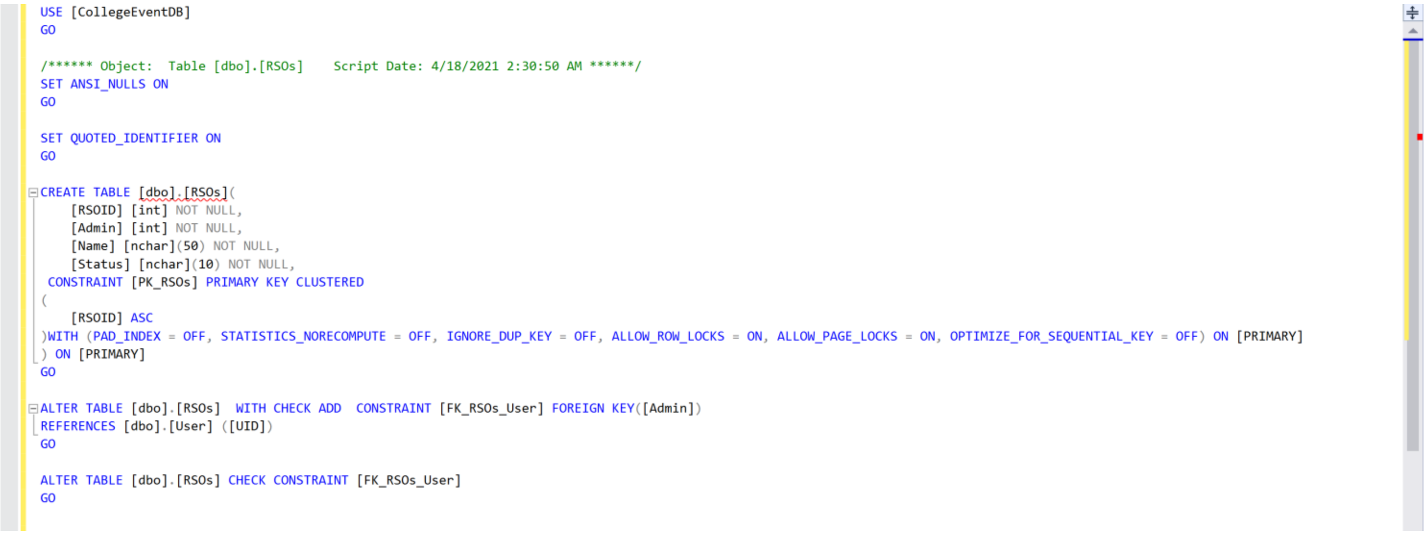
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[University_Affiliation]    Script Date: 4/18/2021 2:29:05 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[University_Affiliation](
    [UID] [int] NOT NULL,
    [UNIID] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[University_Affiliation]  WITH CHECK ADD  CONSTRAINT [FK_University_Affiliation_University] FOREIGN KEY([UNIID])
REFERENCES [dbo].[University] ([UNIID])
GO

ALTER TABLE [dbo].[University_Affiliation] CHECK CONSTRAINT [FK_University_Affiliation_University]
GO

ALTER TABLE [dbo].[University_Affiliation]  WITH CHECK ADD  CONSTRAINT [FK_University_Affiliation_User] FOREIGN KEY([UID])
REFERENCES [dbo].[User] ([UID])
GO

ALTER TABLE [dbo].[University_Affiliation] CHECK CONSTRAINT [FK_University_Affiliation_User]
GO
```

Figure 3: University Table



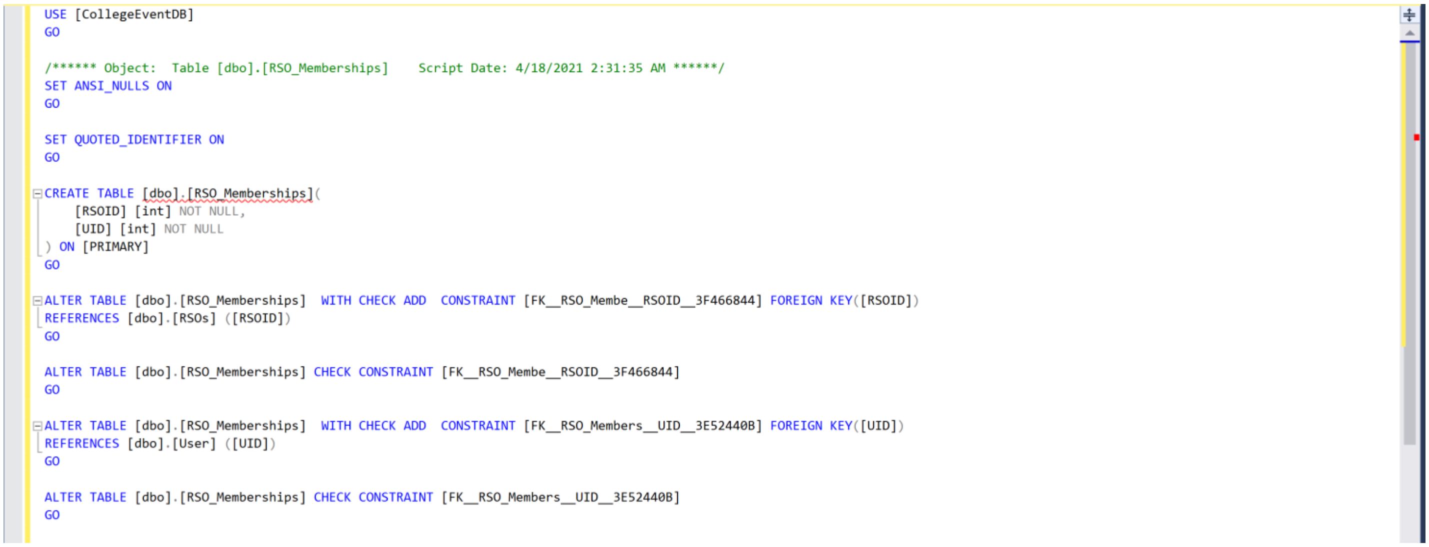
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[University]    Script Date: 4/18/2021 2:29:46 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[University](
    [Description] [text] NULL,
    [Name] [nchar](50) NOT NULL,
    [UNIID] [int] NOT NULL,
    [LID] [int] NOT NULL,
 CONSTRAINT [PK_University] PRIMARY KEY CLUSTERED
(
    [UNIID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[University]  WITH CHECK ADD  CONSTRAINT [FK_University_Location] FOREIGN KEY([LID])
REFERENCES [dbo].[Location] ([LID])
GO

ALTER TABLE [dbo].[University] CHECK CONSTRAINT [FK_University_Location]
GO
```

Figure 4: RSOs Table



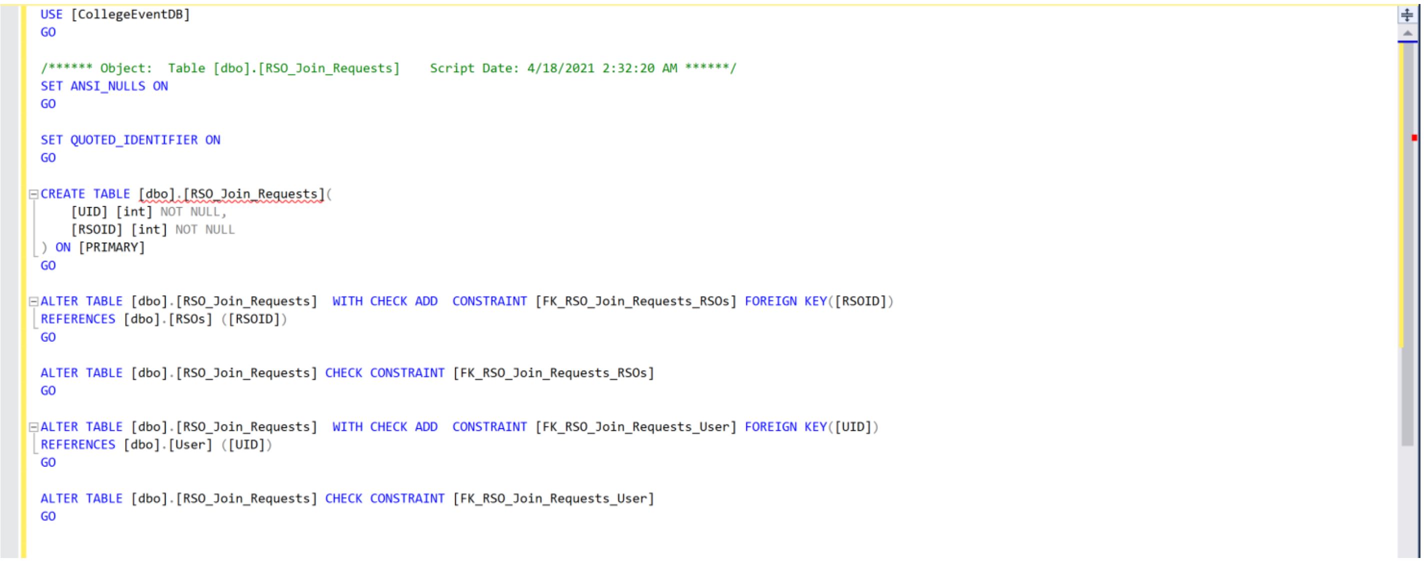
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[RSOs]    Script Date: 4/18/2021 2:30:50 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RSOs](
    [RSOID] [int] NOT NULL,
    [Admin] [int] NOT NULL,
    [Name] [nchar](50) NOT NULL,
    [Status] [nchar](10) NOT NULL,
 CONSTRAINT [PK_RSOs] PRIMARY KEY CLUSTERED
(
    [RSOID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[RSOs]  WITH CHECK ADD  CONSTRAINT [FK_RSOs_User] FOREIGN KEY([Admin])
REFERENCES [dbo].[User] ([UID])
GO

ALTER TABLE [dbo].[RSOs] CHECK CONSTRAINT [FK_RSOs_User]
GO
```

Figure 5: RSO_Memberships Table



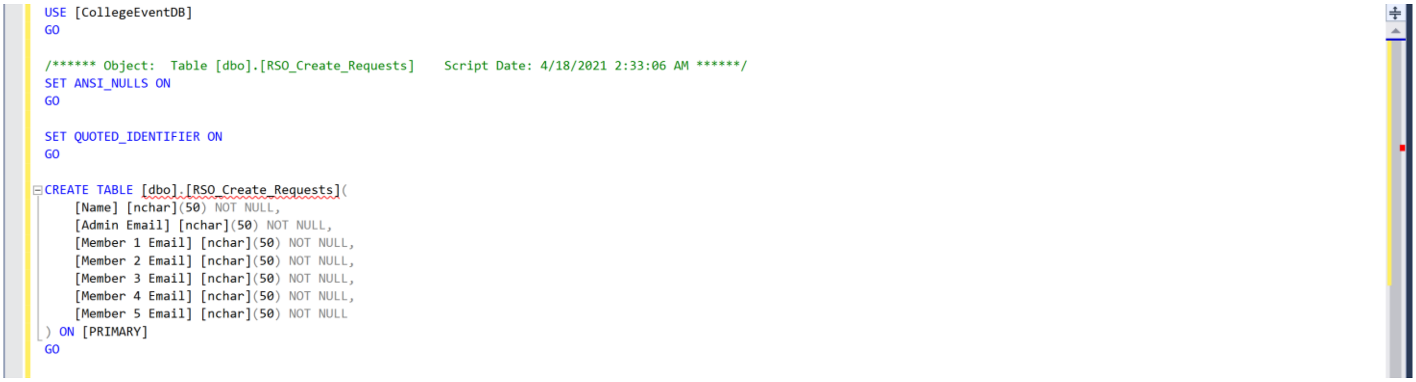
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[RSO_Memberships]    Script Date: 4/18/2021 2:31:35 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RSO_Memberships](
    [RSOID] [int] NOT NULL,
    [UID] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[RSO_Memberships]  WITH CHECK ADD  CONSTRAINT [FK__RSO_Membe__RSOID__3F466844] FOREIGN KEY([RSOID])
REFERENCES [dbo].[RSOs] ([RSOID])
GO

ALTER TABLE [dbo].[RSO_Memberships] CHECK CONSTRAINT [FK__RSO_Membe__RSOID__3F466844]
GO

ALTER TABLE [dbo].[RSO_Memberships]  WITH CHECK ADD  CONSTRAINT [FK__RSO_Members__UID__3E52440B] FOREIGN KEY([UID])
REFERENCES [dbo].[User] ([UID])
GO

ALTER TABLE [dbo].[RSO_Memberships] CHECK CONSTRAINT [FK__RSO_Members__UID__3E52440B]
GO
```

Figure 6: RSO_Join_Requests Table



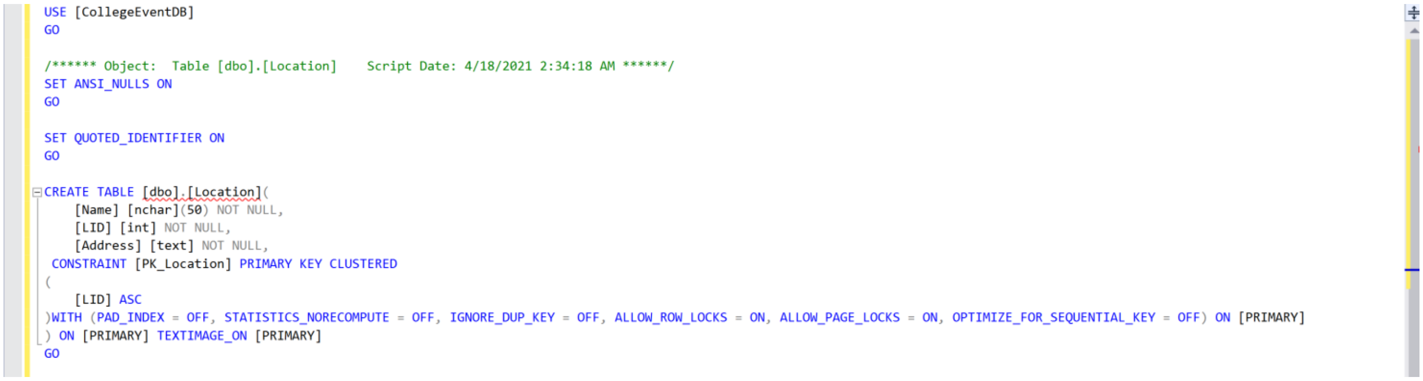
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[RSO_Join_Requests]    Script Date: 4/18/2021 2:32:20 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RSO_Join_Requests](
    [UID] [int] NOT NULL,
    [RSOID] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[RSO_Join_Requests]  WITH CHECK ADD  CONSTRAINT [FK_RSO_Join_Requests_RSOs] FOREIGN KEY([RSOID])
REFERENCES [dbo].[RSOs] ([RSOID])
GO

ALTER TABLE [dbo].[RSO_Join_Requests] CHECK CONSTRAINT [FK_RSO_Join_Requests_RSOs]
GO

ALTER TABLE [dbo].[RSO_Join_Requests]  WITH CHECK ADD  CONSTRAINT [FK_RSO_Join_Requests_User] FOREIGN KEY([UID])
REFERENCES [dbo].[User] ([UID])
GO

ALTER TABLE [dbo].[RSO_Join_Requests] CHECK CONSTRAINT [FK_RSO_Join_Requests_User]
GO
```

Figure 7: RSO_Create_Requests Table



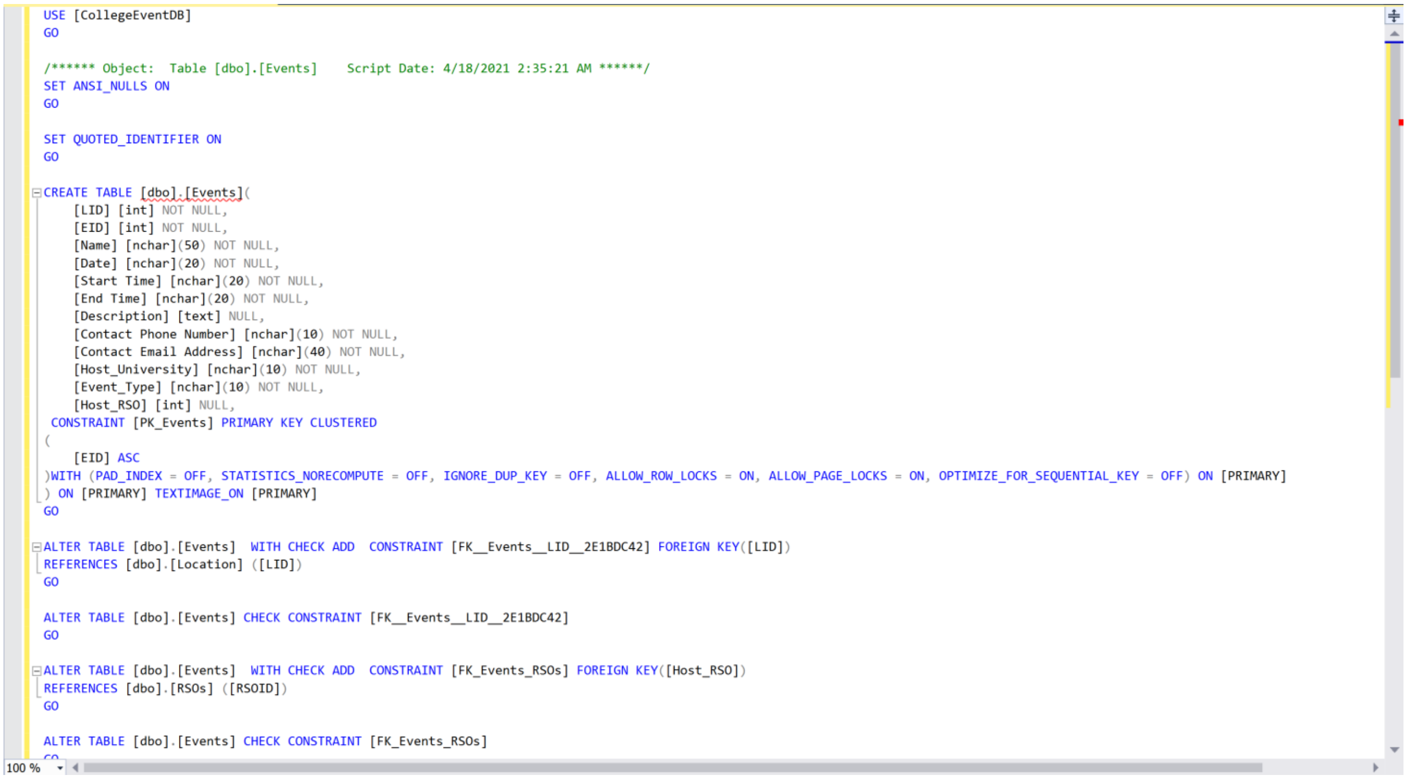
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[RSO_Create_Requests]    Script Date: 4/18/2021 2:33:06 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[RSO_Create_Requests](
    [Name] [nchar](50) NOT NULL,
    [Admin Email] [nchar](50) NOT NULL,
    [Member 1 Email] [nchar](50) NOT NULL,
    [Member 2 Email] [nchar](50) NOT NULL,
    [Member 3 Email] [nchar](50) NOT NULL,
    [Member 4 Email] [nchar](50) NOT NULL,
    [Member 5 Email] [nchar](50) NOT NULL
) ON [PRIMARY]
GO
```

Figure 8: Location Table



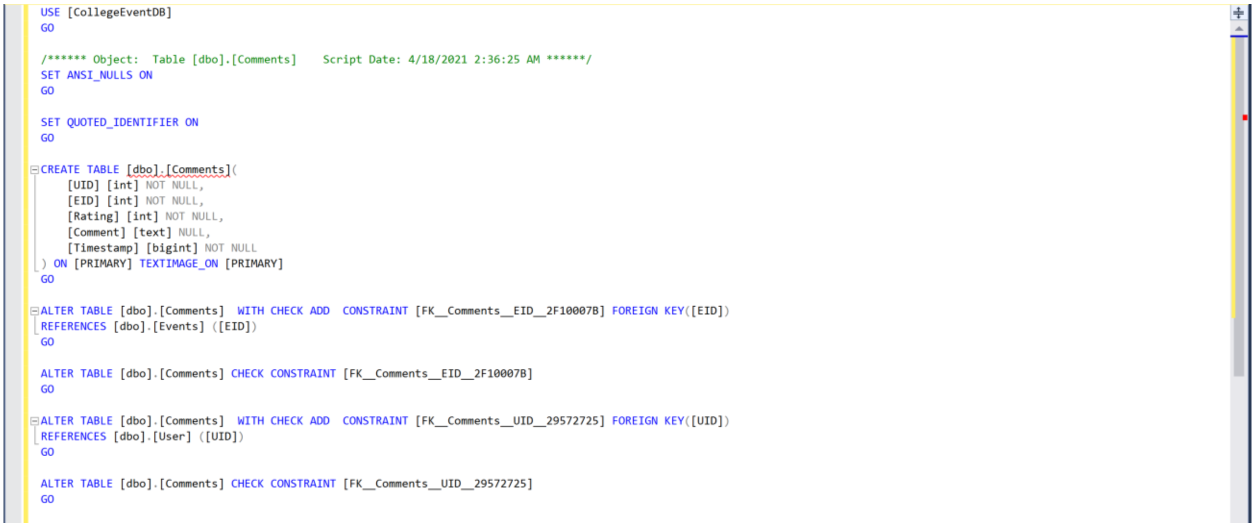
```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[Location]    Script Date: 4/18/2021 2:34:18 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Location](
    [Name] [nchar](50) NOT NULL,
    [LID] [int] NOT NULL,
    [Address] [text] NOT NULL,
 CONSTRAINT [PK_Location] PRIMARY KEY CLUSTERED
(
    [LID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Figure 9: Events Table



```
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[Events]    Script Date: 4/18/2021 2:35:21 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Events](
    [LID] [int] NOT NULL,
    [EID] [int] NOT NULL,
    [Name] [nchar](50) NOT NULL,
    [Date] [nchar](20) NOT NULL,
    [Start Time] [nchar](20) NOT NULL,
    [End Time] [nchar](20) NOT NULL,
    [Description] [text] NULL,
    [Contact Phone Number] [nchar](10) NOT NULL,
    [Contact Email Address] [nchar](40) NOT NULL,
    [Host_University] [nchar](10) NOT NULL,
    [Event_Type] [nchar](10) NOT NULL,
    [Host_RSO] [int] NULL,
 CONSTRAINT [PK_Events] PRIMARY KEY CLUSTERED
(
    [EID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Events]  WITH CHECK ADD  CONSTRAINT [FK__Events__LID__2E1BDC42] FOREIGN KEY([LID])
REFERENCES [dbo].[Location] ([LID])
GO

ALTER TABLE [dbo].[Events] CHECK CONSTRAINT [FK__Events__LID__2E1BDC42]
GO

ALTER TABLE [dbo].[Events]  WITH CHECK ADD  CONSTRAINT [FK_Events_RSOs] FOREIGN KEY([Host_RSO])
REFERENCES [dbo].[RSOs] ([RSOID])
GO

ALTER TABLE [dbo].[Events] CHECK CONSTRAINT [FK_Events_RSOs]
GO
```

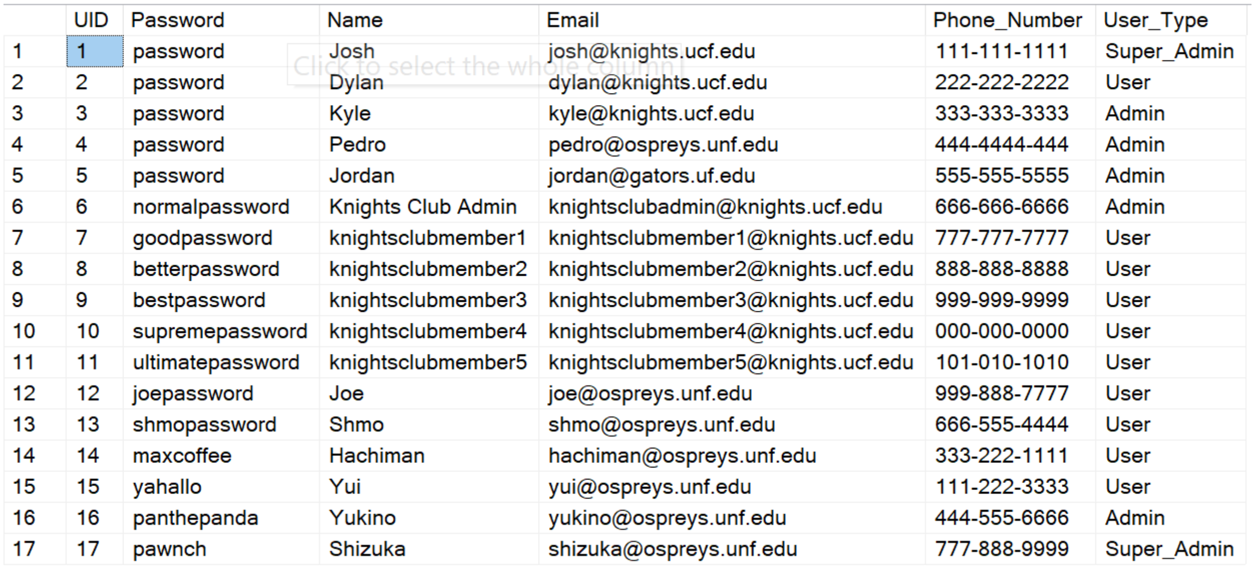
Figure 10: Comments Table



```sql
USE [CollegeEventDB]
GO

/****** Object:  Table [dbo].[Comments]    Script Date: 4/18/2021 2:36:25 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Comments](
    [UID] [int] NOT NULL,
    [EID] [int] NOT NULL,
    [Rating] [int] NOT NULL,
    [Comment] [text] NULL,
    [Timestamp] [bigint] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Comments]  WITH CHECK ADD  CONSTRAINT [FK__Comments__EID__2F10007B] FOREIGN KEY([EID])
REFERENCES [dbo].[Events] ([EID])
GO

ALTER TABLE [dbo].[Comments] CHECK CONSTRAINT [FK__Comments__EID__2F10007B]
GO

ALTER TABLE [dbo].[Comments]  WITH CHECK ADD  CONSTRAINT [FK__Comments__UID__29572725] FOREIGN KEY([UID])
REFERENCES [dbo].[User] ([UID])
GO

ALTER TABLE [dbo].[Comments] CHECK CONSTRAINT [FK__Comments__UID__29572725]
GO
```

Section 5: Sample Data for Relational Data Tables

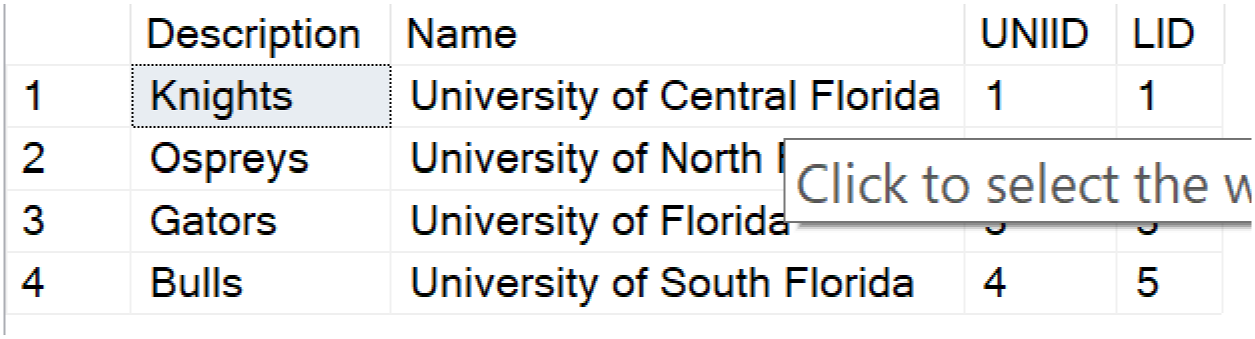
Figure 1: User Data



Click to select the whole column

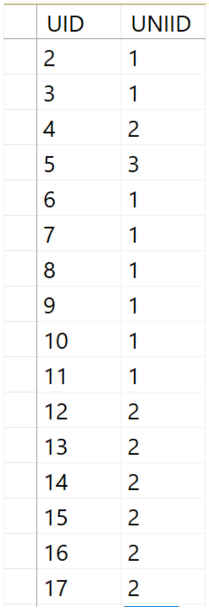
	UID	Password	Name	Email	Phone_Number	User_Type
1	1	password	Josh	josh@knights.ucf.edu	111-111-1111	Super_Admin
2	2	password	Dylan	dylan@knights.ucf.edu	222-222-2222	User
3	3	password	Kyle	kyle@knights.ucf.edu	333-333-3333	Admin
4	4	password	Pedro	pedro@ospreys.unf.edu	444-4444-444	Admin
5	5	password	Jordan	jordan@gators.uf.edu	555-555-5555	Admin
6	6	normalpassword	Knights Club Admin	knightsclubadmin@knights.ucf.edu	666-666-6666	Admin
7	7	goodpassword	knightsclubmember1	knightsclubmember1@knights.ucf.edu	777-777-7777	User
8	8	betterpassword	knightsclubmember2	knightsclubmember2@knights.ucf.edu	888-888-8888	User
9	9	bestpassword	knightsclubmember3	knightsclubmember3@knights.ucf.edu	999-999-9999	User
10	10	supremepassword	knightsclubmember4	knightsclubmember4@knights.ucf.edu	000-000-0000	User
11	11	ultimatepassword	knightsclubmember5	knightsclubmember5@knights.ucf.edu	101-010-1010	User
12	12	joepassword	Joe	joe@ospreys.unf.edu	999-888-7777	User
13	13	shmopassword	Shmo	shmo@ospreys.unf.edu	666-555-4444	User
14	14	maxcoffee	Hachiman	hachiman@ospreys.unf.edu	333-222-1111	User
15	15	yahallo	Yui	yui@ospreys.unf.edu	111-222-3333	User
16	16	panthepanda	Yukino	yukino@ospreys.unf.edu	444-555-6666	Admin
17	17	pawnch	Shizuka	shizuka@ospreys.unf.edu	777-888-9999	Super_Admin

Figure 2: University Data



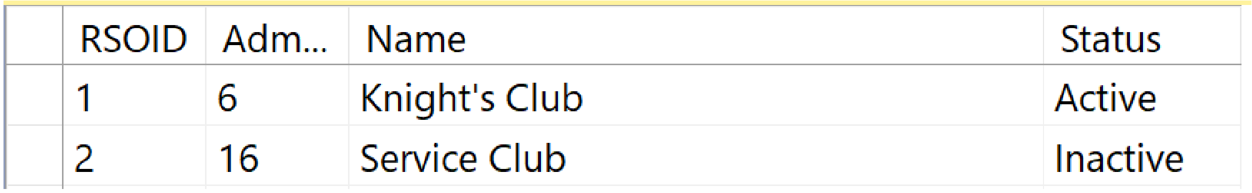
	Description	Name	UNIID	LID
1	Knights	University of Central Florida	1	1
2	Ospreys	University of North F		
3	Gators	University of Florida	3	3
4	Bulls	University of South Florida	4	5

Figure 3: University_Affliliation Data



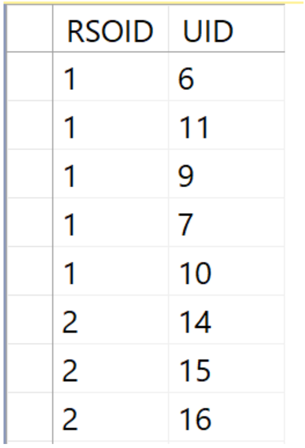
	UID	UNIID
	2	1
	3	1
	4	2
	5	3
	6	1
	7	1
	8	1
	9	1
	10	1
	11	1
	12	2
	13	2
	14	2
	15	2
	16	2
	17	2

Figure 4: RSOs Data



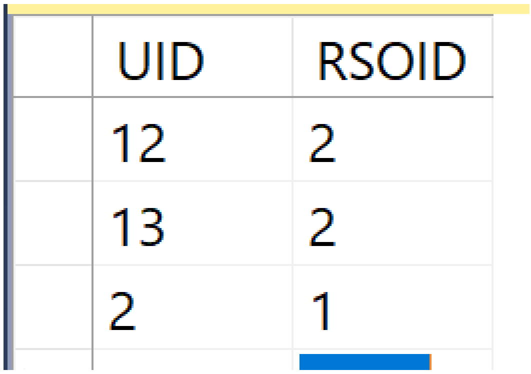
	RSOID	Adm...	Name	Status
	1	6	Knight's Club	Active
	2	16	Service Club	Inactive

Figure 5: RSO_Memberships Data



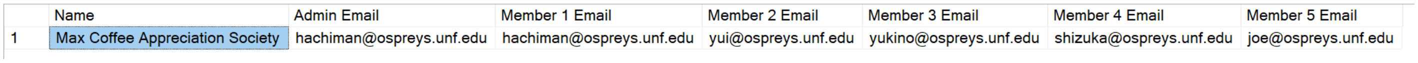
RSOID	UID
1	6
1	11
1	9
1	7
1	10
2	14
2	15
2	16

Figure 6: RSO_Join_Requests



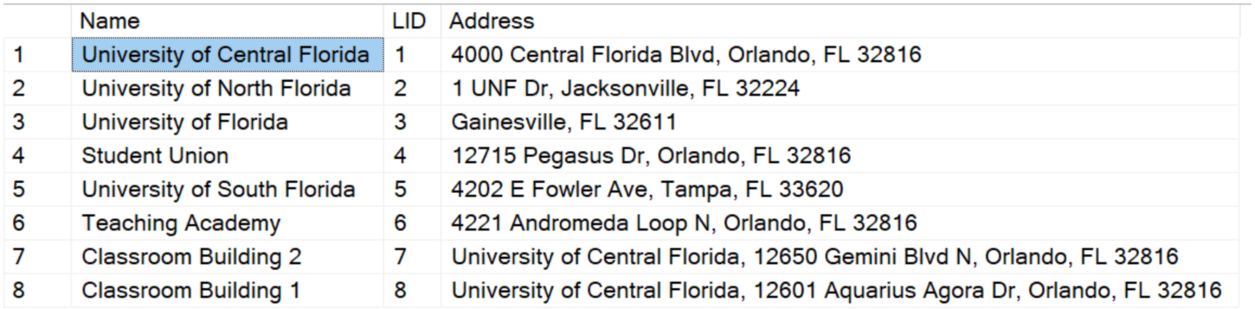
UID	RSOID
12	2
13	2
2	1

Figure 7: RSO_Create_Requests



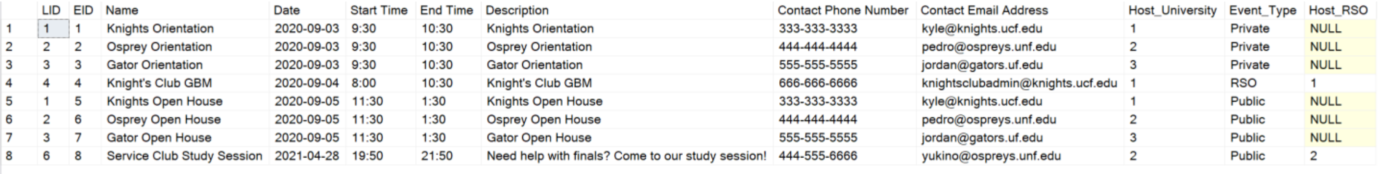
	Name	Admin Email	Member 1 Email	Member 2 Email	Member 3 Email	Member 4 Email	Member 5 Email
1	Max Coffee Appreciation Society	hachiman@ospreys.unf.edu	hachiman@ospreys.unf.edu	yui@ospreys.unf.edu	yukino@ospreys.unf.edu	shizuka@ospreys.unf.edu	joe@ospreys.unf.edu

Figure 8: Location Data



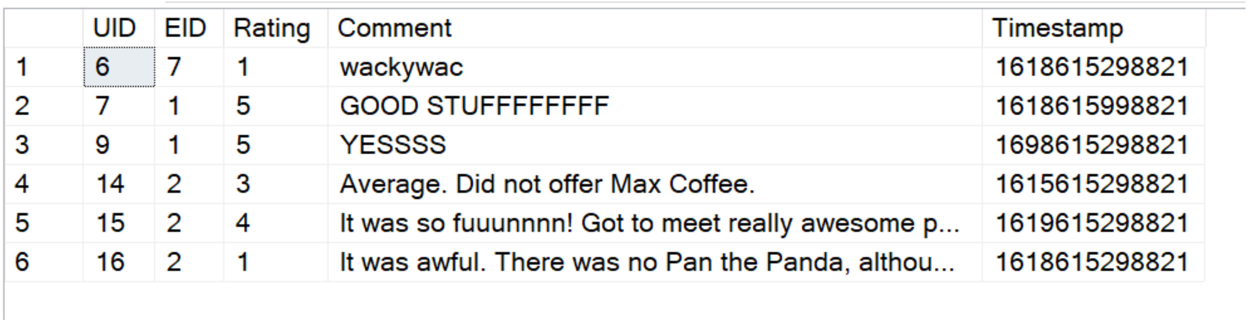
	Name	LID	Address
1	University of Central Florida	1	4000 Central Florida Blvd, Orlando, FL 32816
2	University of North Florida	2	1 UNF Dr, Jacksonville, FL 32224
3	University of Florida	3	Gainesville, FL 32611
4	Student Union	4	12715 Pegasus Dr, Orlando, FL 32816
5	University of South Florida	5	4202 E Fowler Ave, Tampa, FL 33620
6	Teaching Academy	6	4221 Andromeda Loop N, Orlando, FL 32816
7	Classroom Building 2	7	University of Central Florida, 12650 Gemini Blvd N, Orlando, FL 32816
8	Classroom Building 1	8	University of Central Florida, 12601 Aquarius Agora Dr, Orlando, FL 32816

Figure 9:Events Data



	LID	EID	Name	Date	Start Time	End Time	Description	Contact Phone Number	Contact Email Address	Host_University	Event_Type	Host_RSO
1	1	1	Knights Orientation	2020-09-03	9:30	10:30	Knights Orientation	333-333-3333	kyle@knights.ucf.edu	1	Private	NULL
2	2	2	Osprey Orientation	2020-09-03	9:30	10:30	Osprey Orientation	444-444-4444	pedro@ospreys.unf.edu	2	Private	NULL
3	3	3	Gator Orientation	2020-09-03	9:30	10:30	Gator Orientation	555-555-5555	jordan@gators.uf.edu	3	Private	NULL
4	4	4	Knight's Club GBM	2020-09-04	8:00	10:30	Knight's Club GBM	666-666-6666	knightsclubadmin@knights.ucf.edu	1	RSO	1
5	1	5	Knights Open House	2020-09-05	11:30	1:30	Knights Open House	333-333-3333	kyle@knights.ucf.edu	1	Public	NULL
6	2	6	Osprey Open House	2020-09-05	11:30	1:30	Osprey Open House	444-444-4444	pedro@ospreys.unf.edu	2	Public	NULL
7	3	7	Gator Open House	2020-09-05	11:30	1:30	Gator Open House	555-555-5555	jordan@gators.uf.edu	3	Public	NULL
8	6	8	Service Club Study Session	2021-04-28	19:50	21:50	Need help with finals? Come to our study session!	444-555-6666	yukino@ospreys.unf.edu	2	Public	2

Figure 10: Comments Data



	UID	EID	Rating	Comment	Timestamp
1	6	7	1	wackywac	1618615298821
2	7	1	5	GOOD STUFFFFFFFF	1618615998821
3	9	1	5	YESSSS	1698615298821
4	14	2	3	Average. Did not offer Max Coffee.	1615615298821
5	15	2	4	It was so fuuunnnn! Got to meet really awesome p...	1619615298821
6	16	2	1	It was awful. There was no Pan the Panda, althou...	1618615298821

Section 6: SQL Examples

1. SQL statement to insert new RSO (part of the processing of the 'Create RSO')

```
queryString = 'Select * from [dbo].[User] Where ';

        for (var i = 0; i < allEmails.length; i++){
            queryString = queryString.concat('Email = \'' + allEmails[i] + '\'');
            if (i < allEmails.length - 1)
                queryString = queryString.concat(' OR ');
        }
```

Where allEmails is an array of 5 emails of the members of a potential RSO

2. SQL statement to insert a new student to an existing RSO (part of processing of the 'Join RSO' form)

```
queryString = 'Insert into [dbo].[RSO_Memberships] values(' + inputRSOID + ', ' + inputUID + ')';
```

Where inputRSOID is the RSOID of the desired RSO and inputUID is the UID of the user wishing to join the RSO.

3.SQL statement to insert a new event (part of the processing of the ‘Create Event’ form)

```

queryString = 'Insert into [dbo].[Events] values(' + LID + ', ' + newID + ', \''
+ eventName + '\', \'' + date + '\', \'' + startTime + '\', \''
                    + endTime + '\', \'' + description + '\', \'' + phone + '\',
\'' + email + '\', ' + hostUniversity + ', \'' + eventType + '\', ' + hostRSO + '
)';
```

Where LID is the LID of a location an event is held at, newID is the EID of the new event, eventName is the name of the event, date is the date of the event, startTime is the start time of the event, endTime is the end time of the event, description is the description of the event, phone is the contact phone number of the event, email is the contact email of the event, hostUniversity is the UNIID of the university hosting the event, eventType is the type of event, and hostRSO is the RSOID of the RSO hosting the event.

4. SQL statement to insert/update a (new) comment (part of the processing of the ‘Create/Add/Modify Comment’ form)

Insert:

```
queryString = 'Insert into [dbo].[Comments] values(' + inputUID + ', ' + inputEID
 + ', ' + rating + ', \'' + comment +'\', ' + timestamp +')';
```

Where inputUID is the UID of the user wishing to join the RSO, inputEID is the EID of the event being commented on, rating is the user rating on the event, comment is the user comment text on the event, and timestamp is the timestamp of the comment.

Delete:

```
queryString = 'Delete from [dbo].[Comments] Where EID = ' + inputEID + ' AND UID
= ' + inputUID;
```

Updating a comment is the SQL statement for delete followed by the one for Insert.

5.Several SQL queries to display events—public, private, and RSO-- (part of the processing of the ‘View Event’ request by a user with a specific role)

Get Private events:

```
sqlQuery = 'Select * From [dbo].[Events] Where Host_University = ' + UNIID + ' AN
D Event_Type = \'Private\'';
```

Where UNIID is the UNIID of the user’s university

Get Public events:

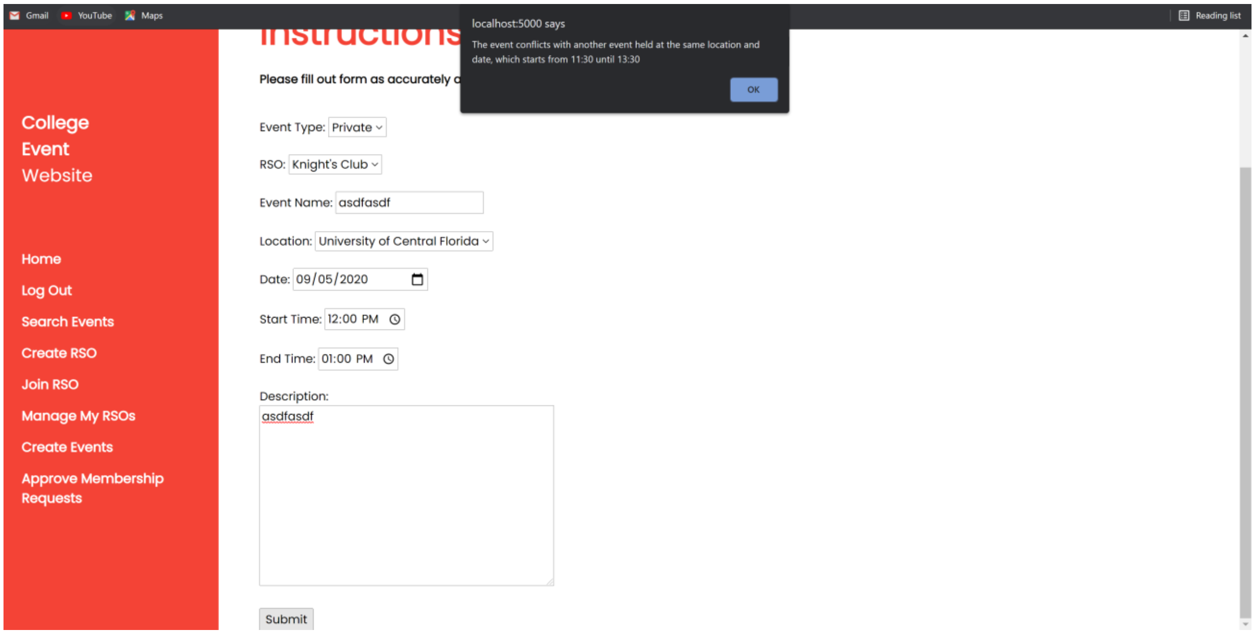
```
sqlQuery = 'Select * From [dbo].[Events] Where Event_Type = \'Public\'';
```

Get RSO events:

```
sqlQuery = 'Select * From [dbo].[RSO_Memberships] Where UID = ' + userData.recordset[0].UID;
        RSOMembershipData = await SQLquery.query(sqlQuery);
        RSOMembershipData = JSON.parse(RSOMembershipData);
        //console.log(RSOMembershipData);
        sqlQuery = 'Select * From [dbo].[Events] Where Host_University = ' + UNIID + ' AND Event_Type = \'RSO\'';
        if (RSOMembershipData.recordset.length > 0)
            sqlQuery = sqlQuery.concat(' AND (');
        for (var i = 0; i < RSOMembershipData.recordset.length; i++){
            sqlQuery = sqlQuery.concat(' Host_RSO = ' + RSOMembershipData.recordset[i].RSOID);
            if (i < RSOMembershipData.recordset.length - 1)
                sqlQuery = sqlQuery.concat(' OR ');
            if (i == RSOMembershipData.recordset.length - 1)
                sqlQuery = sqlQuery.concat(')');
        }

        // if person has no RSO memberships, this query returns empty
        if (RSOMembershipData.recordset.length == 0)
            sqlQuery = sqlQuery.concat( ' AND EID = 0');
```

Section 7: Constraints

1. New event held at the same location and has an overlapping duration with an existing event: error.



localhost:5000 says

The event conflicts with another event held at the same location and date, which starts from 11:30 until 13:30

Instructions

Please fill out form as accurately c

Event Type: Private

RSO: Knight's Club

Event Name: asdfasdf

Location: University of Central Florida

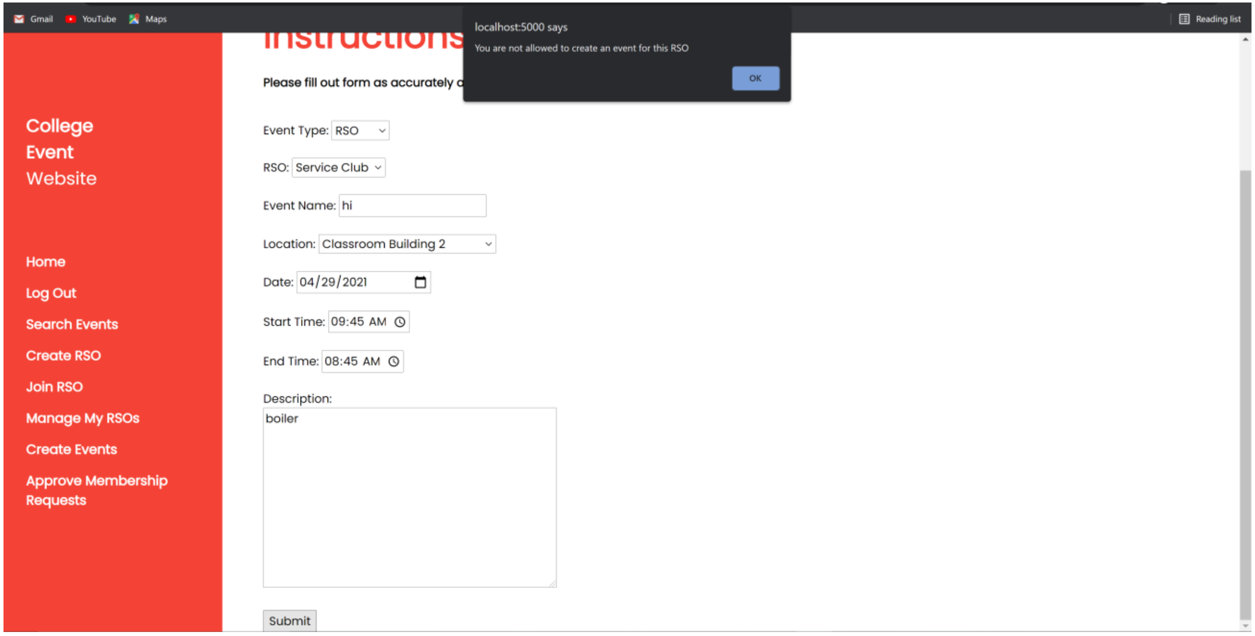
Date: 09/05/2020

Start Time: 12:00 PM

End Time: 01:00 PM

Description:
asdfasdf

2. An admin attempts to create an event for an RSO that they are not a part of: error.



localhost:5000 says

You are not allowed to create an event for this RSO

Instructions

Please fill out form as accurately c

Event Type: RSO

RSO: Service Club

Event Name: hi

Location: Classroom Building 2

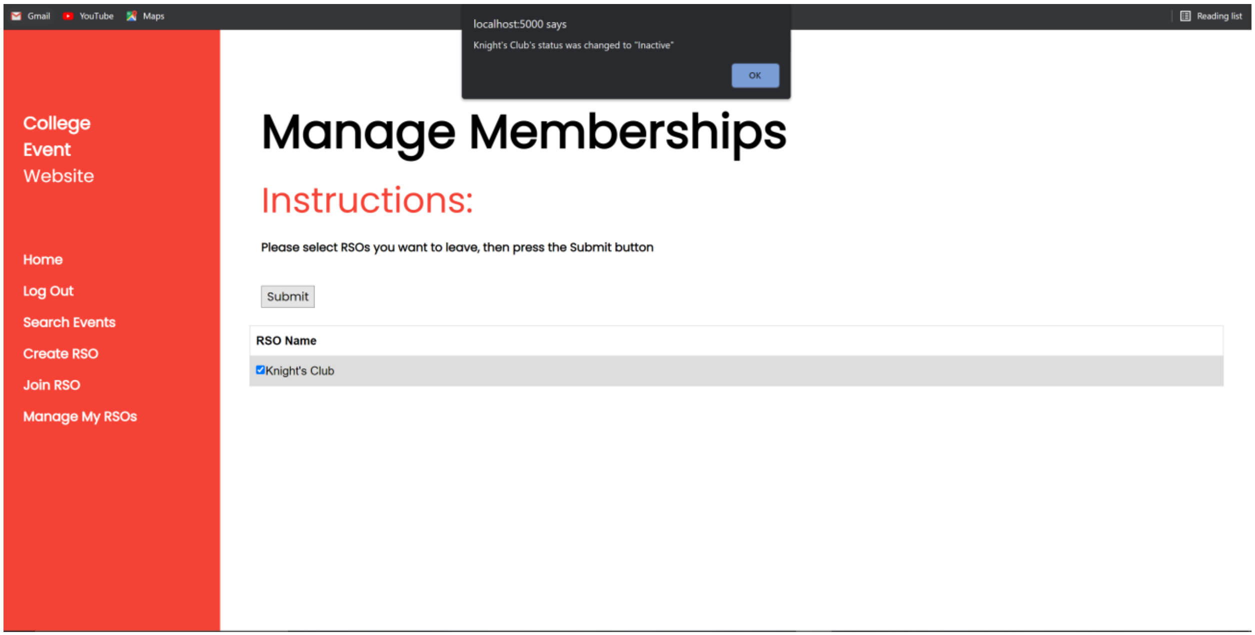
Date: 04/29/2021

Start Time: 09:45 AM

End Time: 08:45 AM

Description:
boiler

3. Status of club changes between active and inactive.



localhost:5000 says

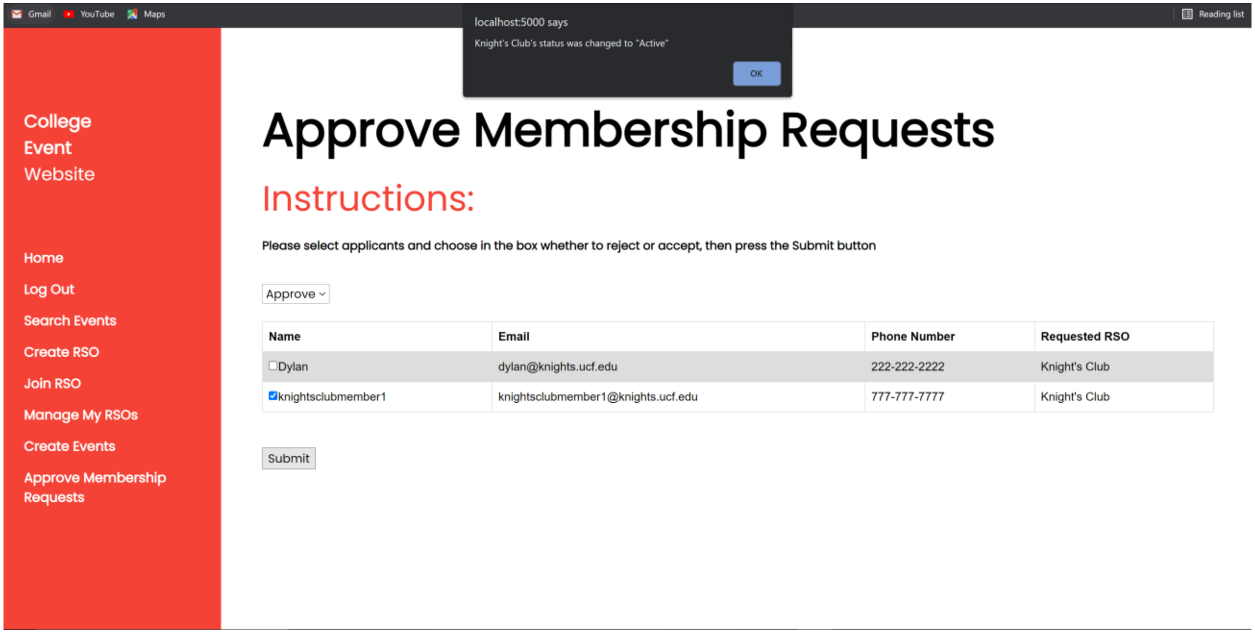
Knight's Club's status was changed to "Inactive"

Manage Memberships

Instructions:

Please select RSOs you want to leave, then press the Submit button

RSO Name
☑Knight's Club



localhost:5000 says

Knight's Club's status was changed to "Active"

Approve Membership Requests

Instructions:

Please select applicants and choose in the box whether to reject or accept, then press the Submit button

Approve

Name	Email	Phone Number	Requested RSO
□Dylan	dylan@knights.ucf.edu	222-222-2222	Knight's Club
☑knightsclubmember1	knightsclubmember1@knights.ucf.edu	777-777-7777	Knight's Club

Section 8: Conclusion

Since there were not much data stored in the tables, the query speeds were fast, so the default searching implementation that MS SQL Server offered was used. Since all IDs used in the tables that needed it were unique and of small magnitude, search indexing could be based on that.

Some desired future implementation would be to load data in xml from external sources to populate tables with data, like the Location table, as I had to manually input the data. Additionally, there were not many options for the kinds of data one could input into event information, such as pictures. I would want to add additionally functionality where videos, pictures, and other media could be uploaded. There were many problems encountered, and some of which are still present. For example, it was not mentioned until now that the web application cannot handle some special character inputs, like apostrophes, due to the use of the ejs template for html. Additionally, the data type nchar(some number) sent data with trailing spaces, which made results from select statements inaccurate to some degree. This was easily fixed with Javascript's .trim() method. Also, the choice of how a web server was going to be hosted was difficult as well, since I had no prior experience. I had to decide between using Node.js or Internet Information Services Manager to handle this, which I ended up choosing the former since it was nicely integrated into Visual Studio code. Also, this was the first time I have coded in Javascript and HTML, so concepts like asynchronous functions were new to me, which would later be needed in the development. This web application was by no means advanced; however, if I had other people to work with that had expertise in specific areas like xml and web hosting, the website could have had more advanced features.