

Group Report, University Event Webapp

COP 4710 Spring 2023

Matthew Daley

Table of Contents

PROJECT DESCRIPTION.....	4
GUI	5
PLATFORM	5
LANGUAGES	5
DBMS	5
CREATE RSO	6
CREATE EVENT	6
VIEW EVENTS	7
ER-MODEL	8
ER DIAGRAM	8
TRIGGERS	8
<i>trg_OnDeleteRSOStatus</i>	8
<i>trg_UpdateRSOStatus</i>	8
<i>trg_CreateRSOStatus</i>	9
<i>END</i>	9
<i>tr_increment_num_students</i>	9
RELATIONAL DATA MODEL	10
CREATE TABLE	10
CHECK	10
CREATE ASSERTION	20
CREATE TRIGGER.....	20
SAMPLE DATA	20
<i>Events table</i>	20
<i>University table</i>	20
<i>Users table</i>	21
<i>RSO Table</i>	21
<i>RSO_Member table</i>	22
<i>Location Table</i>	23
<i>Reactions Table</i>	23
<i>Event_Picture Table</i>	24
<i>Uni_Picture Table</i>	24
<i>Catagory Table</i>	24
SQL	25
NEW RSO	25
JOIN RSO	26
NEW EVENT	26
NEW COMMENT	26
QUERIES.....	27
CONSTRAINT ENFORCEMENT	27
CONFLICTING EVENT TIMES	27
EVENT FOR WRONG RSO	28
RSO STATUS.....	28

DEMO 29

CONCLUSION 29

 DATABASE PERFORMANCE.....29

 DESIRED FEATURES29

 PROBLEMS ENCOUNTERED29

Project description

Most universities in the country host events around campus and off campus. These events are organized by college students in most cases. Students are clustered RSOs (Registered Student Organizations) by different organizations, clubs, fraternities around campus. These events could be of different types: social, fundraising, tech talks, etc. Now, each university has a website where they post their events for the upcoming weeks. One needs to check the website to add each event to his/her calendar. These events are just official events and not all events around the university are included. Another limitation is that one has no way to track weekly events.

You are asked to implement a web-based application that solves the problems. Any student (user) may register with the application to obtain a user ID and a password. There are three user levels: super admin who creates a profile for a university (name, location, description, number of students, pictures, etc.), admin who owns an RSO and may host events, and student who uses the application to look up information about the various events.

Admin can create events with name, event category, description, time, date, location, contact phone, and contact email address. A location should be set from a map (Bing, Google, open street map) with name, latitude, longitude, etc. To populate the database, one can use feeds (e.g., RSS, XML) from events.ucf.edu. Each admin is affiliated with one university, and one or more RSOs. A user can request to create a new RSO or to join an existing one. A new RSO can be created with at least 4 other students with the same email domain (university), e.g., @knights.ucf.edu; and one of them should be assigned as an administrator.

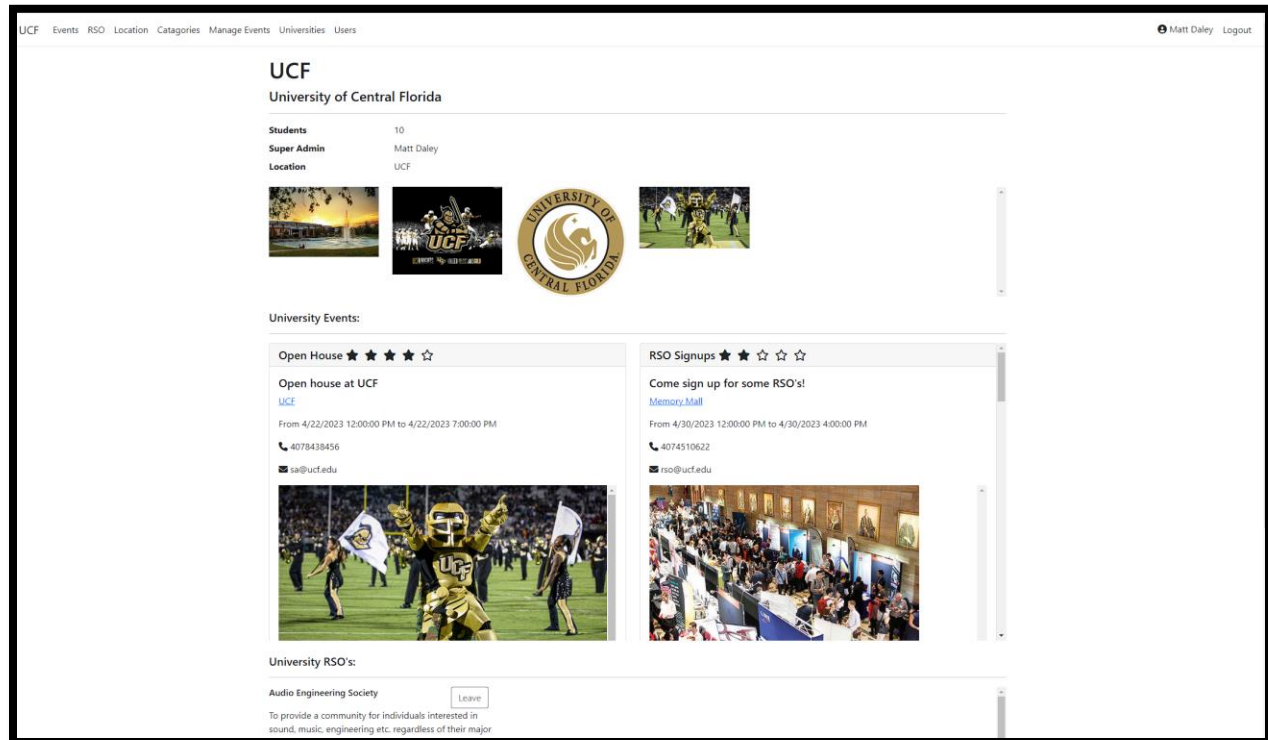
There are different types of events (social, fundraising, tech talks, etc.). Each event can be public, private, or an RSO event. Public events can be seen by everyone; private events can be seen by the students at the host university; and an RSO events can only be seen by members of the RSO. In addition, events can be created without an RSO (public events). Such events must be approved by the super admin. After an event has been published, users can add, remove, and edit their comments on the event, as well as rating the event on a scale of 1-5 (stars). The application should offer some social network integration, e.g., posting from the application to Facebook or Google.

When logged in, Student should be able to view all public events, private events at their university, and event of RSOs of which they are member. They will not be able to create events, but should be able to rate, comment and edit (update) their comments for any event.

GUI

The black boxes aren't actually in the application, just for this report

Platform



View of a Super Admin after logging in.

Languages

The languages and technologies used for this project were MS SQL Server, C# with ASP.NET Core 6.0, EF Core and dependency injection. The spoken language supported by this application is limited to just English.

DBMS

The DBMS of choice for this project is MS SQL Server

Create RSO

To create an RSO, a user must give it a name and a description then pick four other students that attend the same university as them. The application will then make sure that each student is unique and that there are at least 5 before passing it to the database.

After an invite is sent, the invitees get a notification in their navigation bar that shows they have an invitation. The invitation is below.

RSO's Pending Approval:

Example RSO

This is just an example of what an RSO invite would look like

Approve

Deny

Rso

Name

Description

User1

Matt Daley

User2

Matthew Daley

User3

Sirgio Adamaes

User4

Teacher Assistant

Create

[Back to List](#)

Create Event

Users can create events by picking a preset location from the map which is rather large at the bottom of the screen. If the user does not see their desired location on the map already, they can create a new location also using a map with the "Don't see your location?" link.

Create

Event

Name

Contact Phone

Email

Category

STEM

Visibility

Public

Description

StartTime

04/17/2023 12:00 PM

EndTime

04/17/2023 01:00 PM

Selected Location

Create

[Don't see your location?](#)

Map

View Events

My GUI for viewing elements has a twist to it. I have created a y-scrollable div for each of the different types of events: public, private (University) and RSO events. Each event has its own comment section as seen in the bottom right event "First Practice" where the logged in user can edit or delete only their own comments. Also shown are any images associated with each event as well as the rating (in stars) and any other relevant information underneath the event name.

University events are also displayed on the University landing page as seen just under the GUI heading.

Search For Events

Public Events

Water balloon fight ★★★★★☆


Visit UCF for a community water balloon fight

Memory Mall

From 4/26/2023 12:00:00 PM to 4/19/2023 1:30:00 PM

4075924262

recruiting@ucf.edu



Painting ★★★★★


Public painting class

UCF

From 4/19/2023 7:00:00 AM to 4/19/2023 8:00:00 AM

40752645225

paint@ucf.edu



University Events

Open House ★★★★★☆


Open house at UCF

UCF

From 4/22/2023 12:00:00 PM to 4/22/2023 7:00:00 PM

4078438456

sa@ucf.edu



RSO Signups ★☆☆☆☆


Come sign up for some RSO's!

Memory Mall

From 4/30/2023 12:00:00 PM to 4/30/2023 4:00:00 PM

4074510622

rso@ucf.edu



Your RSO Events

First Practice ★★★★★

First practice of the season. Bring water!

Marching Knights at Memory Mall

From 4/19/2023 5:00:00 PM to 4/19/2023 7:30:00 PM

407-158-9286

marching@knights.ucf.edu



We looked great out there today ~Sirgio Adamaes

Edit Delete

React

Chess Meeting ★★★★★

Chess practice, bring shoes

Marching Knights at Tomorrow Land

From 4/17/2023 12:00:00 PM to 4/17/2023 1:00:00 PM

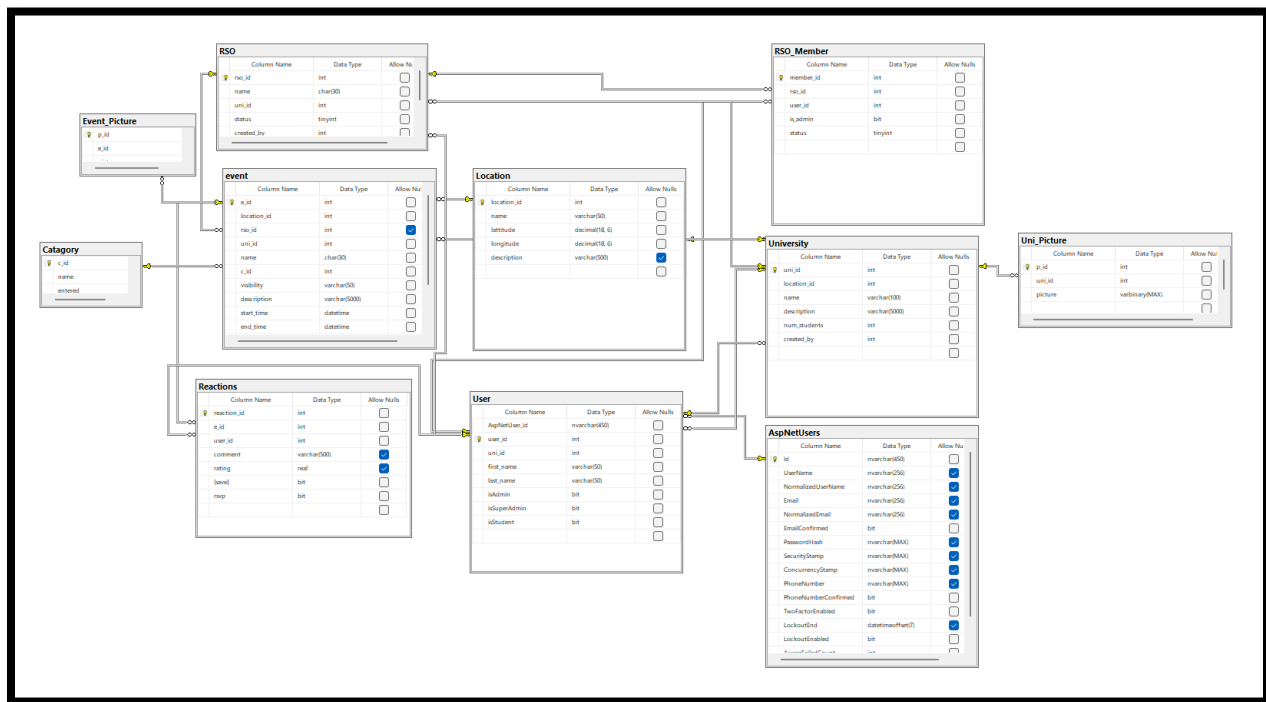
407891259

chess@ucf.edu



ER-Model

ER Diagram



Using Microsoft SQL Server Management Studio, this is my ER-Diagram.

Triggers

trg_OnDeleteRSOStatus

ON [dbo].[RSO_Member]

AFTER DELETE

AS

BEGIN

BEGIN

UPDATE RSO

SET status = 4

WHERE rso_id = (

SELECT rso_id

FROM RSO_Member

WHERE status = 1

AND rso_id = (select rso_id from DELETED)

GROUP BY rso_id

HAVING COUNT(*) < 5

);

END

END

trg_UpdateRSOStatus

ON [dbo].[RSO_Member]

AFTER UPDATE

AS

BEGIN


```

IF EXISTS (
    SELECT rso_id
    FROM inserted
    WHERE status = 1
)
BEGIN
    UPDATE RSO
    SET status = 1
    WHERE rso_id = (
        SELECT rso_id
        FROM RSO_Member
        WHERE rso_id = (select rso_id from inserted) AND status = 1
        GROUP BY rso_id
        HAVING COUNT(*) >= 5
    )
    UPDATE [User]
    SET isAdmin = 1
    WHERE user_id = (
        select created_by
        FROM RSO
        WHERE rso_id = (select rso_id from inserted)
    )
END

trg_CreateRSOStatus
ON [dbo].[RSO_Member]
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT rso_id
        FROM inserted
        WHERE status = 1
    )
    BEGIN
        UPDATE RSO
        SET status = 1
        WHERE status != 2 AND rso_id = (
            SELECT rso_id
            FROM RSO_Member
            WHERE rso_id = (select rso_id from inserted) AND status = 1
            GROUP BY rso_id
            HAVING COUNT(*) >= 5
        )
    END
END

tr_increment_num_students
ON [dbo].[User]
AFTER INSERT
AS
BEGIN
    UPDATE [dbo].[University]
    SET num_students = num_students + 1

```

```
WHERE uni_id IN (SELECT uni_id FROM inserted)
END
```

Relational Data Model

CREATE TABLE

```
USE [Project]
GO
/***** Object: Table [dbo].[User]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[User](
    [AspNetUser_id] [nvarchar](450) NOT NULL,
    [user_id] [int] IDENTITY(1000,1) NOT NULL,
    [uni_id] [int] NOT NULL,
    [first_name] [varchar](50) NOT NULL,
    [last_name] [varchar](50) NOT NULL,
    [isAdmin] [bit] NOT NULL,
    [isSuperAdmin] [bit] NOT NULL,
    [isStudent] [bit] NOT NULL,
    CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
(
    [user_id] 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
/***** Object: Table [dbo].[AspNetUsers]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUsers](
    [Id] [nvarchar](450) NOT NULL,
    [UserName] [nvarchar](256) NULL,
    [NormalizedUserName] [nvarchar](256) NULL,
    [Email] [nvarchar](256) NULL,
    [NormalizedEmail] [nvarchar](256) NULL,
    [EmailConfirmed] [bit] NOT NULL,
    [PasswordHash] [nvarchar](max) NULL,
    [SecurityStamp] [nvarchar](max) NULL,
    [ConcurrencyStamp] [nvarchar](max) NULL,
    [PhoneNumber] [nvarchar](max) NULL,
    [PhoneNumberConfirmed] [bit] NOT NULL,
    [TwoFactorEnabled] [bit] NOT NULL,
    [LockoutEnd] [datetimeoffset](7) NULL,
    [LockoutEnabled] [bit] NOT NULL,
    [AccessFailedCount] [int] NOT NULL,
    CONSTRAINT [PK_AspNetUsers] PRIMARY KEY CLUSTERED
(
    [Id] 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
/***** Object: View [dbo].[CombinedUsers]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE VIEW [dbo].[CombinedUsers]
AS
SELECT      dbo.[User].user_id, dbo.[User].uni_id, dbo.[User].first_name,
dbo.[User].last_name, dbo.[User].isAdmin, dbo.[User].isSuperAdmin, dbo.[User].isStudent,
dbo.[User].AspNetUser_id, dbo.AspNetUsers.UserName, first_name + ' ' + last_name AS
'full_name'
FROM        dbo.AspNetUsers INNER JOIN
            dbo.[User] ON dbo.AspNetUsers.Id = dbo.[User].AspNetUser_id
GO
/***** Object: Table [dbo].[Event]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Event](
    [e_id] [int] IDENTITY(1,1) NOT NULL,
    [location_id] [int] NOT NULL,
    [rso_id] [int] NULL,
    [uni_id] [int] NOT NULL,
    [name] [char](30) NOT NULL,
    [c_id] [int] NOT NULL,
    [visibility] [varchar](50) NOT NULL,
    [description] [varchar](5000) NOT NULL,
    [start_time] [datetime] NOT NULL,
    [end_time] [datetime] NOT NULL,
    [status] [tinyint] NOT NULL,
    [phone] [varchar](50) NOT NULL,
    [email] [varchar](50) NOT NULL,
    CONSTRAINT [PK_event_1] PRIMARY KEY CLUSTERED
(
    [e_id] 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
/***** Object: Table [dbo].[RSO]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[RSO](
    [rso_id] [int] IDENTITY(1,1) NOT NULL,
    [name] [char](30) NOT NULL,
    [uni_id] [int] NOT NULL,
    [status] [tinyint] NOT NULL,
    [created_by] [int] NOT NULL,
    [description] [varchar](500) NOT NULL,
    CONSTRAINT [PK_RSO] PRIMARY KEY CLUSTERED
(

```

```

        [rso_id] 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
/***** Object: Table [dbo].[RSO_Member]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[RSO_Member](
    [member_id] [int] IDENTITY(1,1) NOT NULL,
    [rso_id] [int] NOT NULL,
    [user_id] [int] NOT NULL,
    [is_admin] [bit] NOT NULL,
    [status] [tinyint] NOT NULL,
    CONSTRAINT [PK_RSO_Member] PRIMARY KEY CLUSTERED
(
    [member_id] 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
/***** Object: View [dbo].[RSOEvents]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[RSOEvents]
AS
SELECT      dbo.[User].user_id, dbo.RSO_Member.is_admin, dbo.RSO.name AS rso_name,
dbo.Event.e_id, dbo.Event.location_id, dbo.Event.name, dbo.Event.c_id,
dbo.Event.visibility, dbo.Event.description, dbo.Event.start_time,
           dbo.Event.end_time, dbo.Event.status, dbo.Event.phone,
dbo.Event.email, dbo.RSO.rso_id
FROM        dbo.RSO_Member INNER JOIN
           dbo.RSO ON dbo.RSO_Member.rso_id = dbo.RSO.rso_id INNER JOIN
           dbo.[User] ON dbo.RSO_Member.user_id = dbo.[User].user_id INNER
JOIN
           dbo.AspNetUsers ON dbo.[User].AspNetUser_id = dbo.AspNetUsers.Id
INNER JOIN
           dbo.Event ON dbo.RSO.rso_id = dbo.Event.rso_id
GO
/***** Object: Table [dbo].[__EFMigrationsHistory]    Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[__EFMigrationsHistory](
    [MigrationId] [nvarchar](150) NOT NULL,
    [ProductVersion] [nvarchar](32) NOT NULL,
    CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY CLUSTERED
(
    [MigrationId] 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
/***** Object: Table [dbo].[AspNetRoleClaims]      Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoleClaims](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [RoleId] [nvarchar](450) NOT NULL,
    [ClaimType] [nvarchar](max) NULL,
    [ClaimValue] [nvarchar](max) NULL,
    CONSTRAINT [PK_AspNetRoleClaims] PRIMARY KEY CLUSTERED
(
    [Id] 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
/***** Object: Table [dbo].[AspNetRoles]      Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoles](
    [Id] [nvarchar](450) NOT NULL,
    [Name] [nvarchar](256) NULL,
    [NormalizedName] [nvarchar](256) NULL,
    [ConcurrencyStamp] [nvarchar](max) NULL,
    CONSTRAINT [PK_AspNetRoles] PRIMARY KEY CLUSTERED
(
    [Id] 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
/***** Object: Table [dbo].[AspNetUserClaims]      Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserClaims](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UserId] [nvarchar](450) NOT NULL,
    [ClaimType] [nvarchar](max) NULL,
    [ClaimValue] [nvarchar](max) NULL,
    CONSTRAINT [PK_AspNetUserClaims] PRIMARY KEY CLUSTERED
(
    [Id] 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
/***** Object: Table [dbo].[AspNetUserLogins]    Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserLogins](
    [LoginProvider] [nvarchar](128) NOT NULL,
    [ProviderKey] [nvarchar](128) NOT NULL,
    [ProviderDisplayName] [nvarchar](max) NULL,
    [UserId] [nvarchar](450) NOT NULL,
    CONSTRAINT [PK_AspNetUserLogins] PRIMARY KEY CLUSTERED
(
    [LoginProvider] ASC,
    [ProviderKey] 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
/***** Object: Table [dbo].[AspNetUserRoles]    Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserRoles](
    [UserId] [nvarchar](450) NOT NULL,
    [RoleId] [nvarchar](450) NOT NULL,
    CONSTRAINT [PK_AspNetUserRoles] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [RoleId] 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
/***** Object: Table [dbo].[AspNetUserTokens]    Script Date: 4/18/2023 6:36:57 PM
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserTokens](
    [UserId] [nvarchar](450) NOT NULL,
    [LoginProvider] [nvarchar](128) NOT NULL,
    [Name] [nvarchar](128) NOT NULL,
    [Value] [nvarchar](max) NULL,
    CONSTRAINT [PK_AspNetUserTokens] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [LoginProvider] ASC,

```

```

        [Name] 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
/***** Object: Table [dbo].[Catagory]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Catagory](
    [c_id] [int] IDENTITY(1,1) NOT NULL,
    [name] [varchar](50) NOT NULL,
    [entered] [datetime] NOT NULL,
    CONSTRAINT [PK_Catagory] PRIMARY KEY CLUSTERED
(
    [c_id] 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
/***** Object: Table [dbo].[Event_Picture]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Event_Picture](
    [p_id] [int] IDENTITY(1,1) NOT NULL,
    [e_id] [int] NOT NULL,
    [picture] [varbinary](max) NOT NULL,
    CONSTRAINT [PK_Event_Picture] PRIMARY KEY CLUSTERED
(
    [p_id] 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
/***** Object: Table [dbo].[Location]    Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Location](
    [location_id] [int] IDENTITY(1000,1) NOT NULL,
    [name] [varchar](50) NOT NULL,
    [latitude] [decimal](18, 6) NOT NULL,
    [longitude] [decimal](18, 6) NOT NULL,
    [description] [varchar](500) NULL,
    CONSTRAINT [PK_Location] PRIMARY KEY CLUSTERED
(
    [location_id] 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
/***** Object: Table [dbo].[Reactions]      Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Reactions](
    [reaction_id] [int] IDENTITY(1,1) NOT NULL,
    [e_id] [int] NOT NULL,
    [user_id] [int] NOT NULL,
    [comment] [varchar](500) NULL,
    [rating] [real] NULL,
    [save] [bit] NOT NULL,
    [rsvp] [bit] NOT NULL,
    CONSTRAINT [PK_Reactions] PRIMARY KEY CLUSTERED
(
    [reaction_id] 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
/***** Object: Table [dbo].[Uni_Picture]      Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Uni_Picture](
    [p_id] [int] IDENTITY(1,1) NOT NULL,
    [uni_id] [int] NOT NULL,
    [picture] [varbinary](max) NOT NULL,
    CONSTRAINT [PK_Uni_Picture] PRIMARY KEY CLUSTERED
(
    [p_id] 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
/***** Object: Table [dbo].[University]      Script Date: 4/18/2023 6:36:57 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[University](
    [uni_id] [int] IDENTITY(1,1) NOT NULL,
    [location_id] [int] NOT NULL,
    [name] [varchar](100) NOT NULL,
    [description] [varchar](5000) NOT NULL,
    [num_students] [int] NOT NULL,
    [created_by] [int] NOT NULL,
    CONSTRAINT [PK_University] PRIMARY KEY CLUSTERED
(
    [uni_id] 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
SET ANSI_PADDING ON
GO
CHECK
ALTER TABLE [dbo].[Catagory] ADD CONSTRAINT [DF_Catagory_entered] DEFAULT (getdate())
FOR [entered]
GO
ALTER TABLE [dbo].[Event] ADD CONSTRAINT [DF_event_status] DEFAULT ((0)) FOR [status]
GO
ALTER TABLE [dbo].[Reactions] ADD CONSTRAINT [DF_Reactions_save] DEFAULT ((0)) FOR
[save]
GO
ALTER TABLE [dbo].[Reactions] ADD CONSTRAINT [DF_Reactions_rsvp] DEFAULT ((0)) FOR
[rsvp]
GO
ALTER TABLE [dbo].[RSO] ADD CONSTRAINT [DF_RSO_status] DEFAULT ((0)) FOR [status]
GO
ALTER TABLE [dbo].[RSO_Member] ADD CONSTRAINT [DF_RSO_Member_is_admin] DEFAULT ((0))
FOR [is_admin]
GO
ALTER TABLE [dbo].[RSO_Member] ADD CONSTRAINT [DF_RSO_Member_status] DEFAULT ((0)) FOR
[status]
GO
ALTER TABLE [dbo].[User] ADD CONSTRAINT [DF_User_isAdmin] DEFAULT ((0)) FOR [isAdmin]
GO
ALTER TABLE [dbo].[User] ADD CONSTRAINT [DF_User_isSuperAdmin] DEFAULT ((0)) FOR
[isSuperAdmin]
GO
ALTER TABLE [dbo].[User] ADD CONSTRAINT [DF_User_isStudent] DEFAULT ((1)) FOR
[isStudent]
GO
ALTER TABLE [dbo].[AspNetRoleClaims] WITH CHECK ADD CONSTRAINT
[FK_AspNetRoleClaims_AspNetRoles_RoleId] FOREIGN KEY([RoleId]
REFERENCES [dbo].[AspNetRoles] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetRoleClaims] CHECK CONSTRAINT
[FK_AspNetRoleClaims_AspNetRoles_RoleId]
GO
ALTER TABLE [dbo].[AspNetUserClaims] WITH CHECK ADD CONSTRAINT
[FK_AspNetUserClaims_AspNetUsers_UserId] FOREIGN KEY([UserId]
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT
[FK_AspNetUserClaims_AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[AspNetUserLogins] WITH CHECK ADD CONSTRAINT
[FK_AspNetUserLogins_AspNetUsers_UserId] FOREIGN KEY([UserId]
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserLogins] CHECK CONSTRAINT
[FK_AspNetUserLogins_AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_AspNetUserRoles_AspNetRoles_RoleId] FOREIGN KEY([RoleId])

```

```

REFERENCES [dbo].[AspNetRoles] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_AspNetUserRoles_AspNetRoles_RoleId]
GO
ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_AspNetUserRoles_AspNetUsers_UserId] FOREIGN KEY([UserId])
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_AspNetUserRoles_AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[AspNetUserTokens] WITH CHECK ADD CONSTRAINT
[FK_AspNetUserTokens_AspNetUsers_UserId] FOREIGN KEY([UserId]
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserTokens] CHECK CONSTRAINT
[FK_AspNetUserTokens_AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[Event] WITH CHECK ADD CONSTRAINT [FK_event_Catagory] FOREIGN
KEY([c_id])
REFERENCES [dbo].[Catagory] ([c_id])
GO
ALTER TABLE [dbo].[Event] CHECK CONSTRAINT [FK_event_Catagory]
GO
ALTER TABLE [dbo].[Event] WITH CHECK ADD CONSTRAINT [FK_event_Location] FOREIGN
KEY([location_id])
REFERENCES [dbo].[Location] ([location_id])
GO
ALTER TABLE [dbo].[Event] CHECK CONSTRAINT [FK_event_Location]
GO
ALTER TABLE [dbo].[Event] WITH CHECK ADD CONSTRAINT [FK_event_RS0] FOREIGN
KEY([rso_id])
REFERENCES [dbo].[RS0] ([rso_id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Event] CHECK CONSTRAINT [FK_event_RS0]
GO
ALTER TABLE [dbo].[Event] WITH CHECK ADD CONSTRAINT [FK_event_University] FOREIGN
KEY([uni_id])
REFERENCES [dbo].[University] ([uni_id])
GO
ALTER TABLE [dbo].[Event] CHECK CONSTRAINT [FK_event_University]
GO
ALTER TABLE [dbo].[Event_Picture] WITH CHECK ADD CONSTRAINT [FK_Event_Picture_Event]
FOREIGN KEY([e_id])
REFERENCES [dbo].[Event] ([e_id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Event_Picture] CHECK CONSTRAINT [FK_Event_Picture_Event]
GO
ALTER TABLE [dbo].[Reactions] WITH CHECK ADD CONSTRAINT [FK_Reactions_event] FOREIGN
KEY([e_id])
REFERENCES [dbo].[Event] ([e_id])
ON DELETE CASCADE

```

```

GO
ALTER TABLE [dbo].[Reactions] CHECK CONSTRAINT [FK_Reactions_event]
GO
ALTER TABLE [dbo].[Reactions] WITH CHECK ADD CONSTRAINT [FK_Reactions_User] FOREIGN
KEY([user_id])
REFERENCES [dbo].[User] ([user_id])
GO
ALTER TABLE [dbo].[Reactions] CHECK CONSTRAINT [FK_Reactions_User]
GO
ALTER TABLE [dbo].[RSO] WITH CHECK ADD CONSTRAINT [FK_RSO_University] FOREIGN
KEY([uni_id])
REFERENCES [dbo].[University] ([uni_id])
GO
ALTER TABLE [dbo].[RSO] CHECK CONSTRAINT [FK_RSO_University]
GO
ALTER TABLE [dbo].[RSO] WITH CHECK ADD CONSTRAINT [FK_RSO_User] FOREIGN
KEY([created_by])
REFERENCES [dbo].[User] ([user_id])
GO
ALTER TABLE [dbo].[RSO] CHECK CONSTRAINT [FK_RSO_User]
GO
ALTER TABLE [dbo].[RSO_Member] WITH CHECK ADD CONSTRAINT [FK_RSO_Member_RSO] FOREIGN
KEY([rso_id])
REFERENCES [dbo].[RSO] ([rso_id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[RSO_Member] CHECK CONSTRAINT [FK_RSO_Member_RSO]
GO
ALTER TABLE [dbo].[RSO_Member] WITH CHECK ADD CONSTRAINT [FK_RSO_Member_User] FOREIGN
KEY([user_id])
REFERENCES [dbo].[User] ([user_id])
GO
ALTER TABLE [dbo].[RSO_Member] CHECK CONSTRAINT [FK_RSO_Member_User]
GO
ALTER TABLE [dbo].[Uni_Picture] WITH CHECK ADD CONSTRAINT [FK_Uni_Picture_University]
FOREIGN KEY([uni_id])
REFERENCES [dbo].[University] ([uni_id])
GO
ALTER TABLE [dbo].[Uni_Picture] CHECK CONSTRAINT [FK_Uni_Picture_University]
GO
ALTER TABLE [dbo].[University] WITH CHECK ADD CONSTRAINT [FK_University_Location]
FOREIGN KEY([location_id])
REFERENCES [dbo].[Location] ([location_id])
GO
ALTER TABLE [dbo].[University] CHECK CONSTRAINT [FK_University_Location]
GO
ALTER TABLE [dbo].[University] WITH CHECK ADD CONSTRAINT [SuperAdmin] FOREIGN
KEY([created_by])
REFERENCES [dbo].[User] ([user_id])
GO
ALTER TABLE [dbo].[University] CHECK CONSTRAINT [SuperAdmin]
GO
ALTER TABLE [dbo].[User] WITH CHECK ADD CONSTRAINT [FK_User_AspNetUsers] FOREIGN
KEY([AspNetUser_id])
REFERENCES [dbo].[AspNetUsers] ([Id])
GO
ALTER TABLE [dbo].[User] CHECK CONSTRAINT [FK_User_AspNetUsers]
GO

```

```

ALTER TABLE [dbo].[User] WITH CHECK ADD CONSTRAINT [FK_User_University1] FOREIGN
KEY([uni_id])
REFERENCES [dbo].[University] ([uni_id])
GO
ALTER TABLE [dbo].[User] CHECK CONSTRAINT [FK_User_University1]
GO

```

CREATE ASSERTION

Assertions were handled in the back end of the webapp

CREATE TRIGGER

Already stated above

Sample Data

Events table

	e_id	location_id	rso_id	uni_id	name	c_id	visiblity	description	start_time	end_time	status	phone	email
1	20	1000	11	1	Movie Night	4	RSO	Come out and watch a movie with us!	2023-04-17 18:00:00.000	2023-04-17 21:00:00.000	1	407-957-2591	movie@knights.ucf.edu
2	21	1005	13	1	First Practice	5	RSO	First practice of the season. Bring water!	2023-04-19 17:00:00.000	2023-04-19 19:30:00.000	1	407-158-9286	marching@knights.ucf.edu
3	22	1000	13	1	Open House	4	University	Open house at UCF	2023-04-22 12:00:00.000	2023-04-22 19:00:00.000	1	4078438456	sa@ucf.edu
4	23	1005	7	1	RSO Signups	4	University	Come sign up for some RSO's!	2023-04-30 12:00:00.000	2023-04-30 16:00:00.000	1	4074510622	rso@ucf.edu
5	24	1005	7	1	Water balloon fight	4	Public	Visit UCF for a community water balloon fight	2023-04-26 12:00:00.000	2023-04-19 13:30:00.000	1	4075924262	recruiting@ucf.edu
6	25	1000	13	1	Painting	4	Public	Public painting class	2023-04-19 07:00:00.000	2023-04-19 08:00:00.000	1	40752645225	paint@ucf.edu
7	26	1003	13	1	Chess Competition	3	Public	Chess battle live	2023-04-27 12:00:00.000	2023-04-27 13:00:00.000	1	4075951212	this@that.com
8	27	1001	13	1	Chess Meeting	2	RSO	Chess practice, bring shoes	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	407891259	chess@ucf.edu
9	28	1006	12	1	Archery Lessons	5	University	Archery	2023-04-01 12:00:00.000	2023-04-01 13:00:00.000	1	4075952123	archery@ucf.edu
10	29	1006	9	1	Pick up Basketball	5	University	Pick up game at the courts	2023-04-06 12:00:00.000	2023-04-06 13:00:00.000	1	4075955995	bb@bb.com
11	30	1001	9	1	Science Fair	2	University	UCF Science Fair	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075955877	science@ucf.edu
12	31	1007	10	1	Golf Tryouts	5	University	Golf Tryouts	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075125998	golf@ucf.edu
13	32	1000	7	1	Disk golf	2	RSO	Disk Golf	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075912375	dg@ucf.edu
14	33	1002	7	1	Nursing Seminar	2	University	Nursing day	2023-04-20 12:00:00.000	2023-04-20 13:00:00.000	1	4075911211	nurse@ucf.edu
15	34	1004	12	1	ESports	4	RSO	Esports	2023-04-17 14:00:00.000	2023-04-17 15:00:00.000	1	4170585895	esports@ucf.edu
16	35	1004	7	1	Group Reading	4	Public	UCF Reading	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4071599568	ucf@ucf.edu
17	36	1003	7	1	Event Events	2	RSO	This is an event	2023-03-26 12:00:00.000	2023-03-26 13:00:00.000	1	4075956589	event@ucf.edu
18	37	1006	7	1	Pickleball	4	University	Pickleball!	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075959596	pickle@ucf.edu
19	39	1000	7	1	Algebra	2	University	Math	2023-05-20 12:00:00.000	2023-05-20 13:00:00.000	2	4075961238	math@ucf.edu
20	40	1007	7	1	Biking	5	RSO	Biking at the uni	2023-04-25 12:00:00.000	2023-04-25 13:00:00.000	1	4075954584	bike@ucf.edu

University table

	uni_id	location_id	name	description	num_students	created_by
1	1	1000	UCF	University of Central Florida	11	2
2	2	1003	MIT	Massachusetts Institute of Technology	6	1008

Users table

* Note: all RSO's were created by 6 users given the 6 admins

	AspNetUser_id	user_id	uni_id	first_name	last_name	isAdmin	isSuperAdmin	isStudent
1	fd2fa92f-8da9-4e35-a7ce-dc036498b2d7	2	1	Matt	Daley	1	1	1
2	5f54531b-51d9-4a07-80ec-e7ca0882c060	1001	1	Matthew	Daley	0	0	1
3	e3247f85-5fdf-493b-94ee-bb77eed15892	1002	1	b	b	0	0	1
4	a35a847c-b0fc-4f10-905f-7c74489c83a6	1003	1	c	c	0	0	1
5	ca731a25-b253-4684-a0c5-c6170ffa5e2c	1004	1	d	d	0	0	1
6	fd471596-5e63-4e52-a001-893eba44b17f	1005	1	e	e	0	0	1
7	7a47c620-3aa7-4055-ae63-d19ad3547e19	1006	1	First	Last	0	0	1
8	604f3529-a588-40e9-bf4f-9e2fcc2ca4d0	1007	1	zebra	zofran	0	0	1
9	4a1d84db-87ef-4317-9203-f53bdd16ac3a	1008	2	James	Winston	1	1	1
10	93b949b2-2240-48e2-9181-60a4e4bec391	1009	1	Sirgio	Adamaes	1	0	1
11	38490abe-0c0e-461d-8750-f054c4e8b146	1010	2	Alex	Jones	0	0	1
12	07e9b862-82e5-4d55-9bdd-e24d43218475	1011	2	Bery	Benson	0	0	1
13	2f473541-a1af-4342-a249-c3383778b15	1012	2	Cathy	Catus	0	0	1
14	ef9a8c7a-6fa4-49b4-892b-abe97590a619	1013	2	Danny	Donalds	0	0	1
15	868bddfc-b78a-49e7-94a4-cee20eeaa335	1014	1	Teacher	Assistant	1	0	1
16	712ed5da-993c-4626-b913-597703bf0d12	1015	1	fds	fdas	1	0	1
17	92a4b664-b51b-47c5-8e77-9aa97b1dfa41	1016	2	h	h	0	0	1
18	b69dd821-9bd9-4e30-9d8b-4f6884afb573	1017	1	New	User	1	0	1

RSO Table

	rso_id	name	uni_id	status	created_by	description
1	7	Aviator Knights	1	2	2	Aviator Knights is a student led organization at UC...
2	8	Audio Engineering Society	1	2	2	To provide a community for individuals interested i...
3	9	Bee Club	1	2	2	The mission of the Bee Club at UCF is to create a ...
4	10	Chess Club	1	2	2	Weekly meetings Fridays 6-9PM In the first Floor o...
5	11	Cinema and Film Appreciation	1	2	2	The purpose of the Cinema and Film Appreciation ...
6	12	First Person View Knights	1	2	2	The mission of First Person View Knights is to give...
7	13	Marching Knights	1	2	1009	UCF Marching Band
8	18	EIT	2	0	1008	Engineering IT
9	19	The MIT Crew	2	0	1008	Everything MIT
10	20	The best RSO ever	1	4	1015	Obviously
11	21	The best RSO ever	1	4	1015	Desc
12	22	1	1	4	1014	1
13	23	Example RSO	1	0	2	This is just an example of what an RSO invite wou...
14	24	TA Demo RSO	1	2	2	This is just a temp RSO
15	25	Name	1	0	1017	Desc
16	27	RSO Name	2	0	2	RSO Description
17	28	RSO Name	1	0	2	RSO Description

RSO_Member table

(Only the first 32 out of hundreds)

	member_id	rso_id	user_id	is_admin	status
1	1	7	1001	0	1
2	2	7	1002	0	1
3	3	7	1003	0	1
4	4	7	1004	0	1
5	6	8	1004	0	1
6	7	8	1005	0	1
7	8	8	1006	0	1
8	9	8	1007	0	1
9	11	9	1002	0	1
10	12	9	1003	0	1
11	13	9	1004	0	1
12	14	9	1007	0	1
13	16	10	1001	0	1
14	17	10	1002	0	1
15	18	10	1003	0	1
16	19	10	1005	0	1
17	21	11	1004	0	0
18	22	11	1005	0	1
19	23	11	1006	0	1
20	24	11	1007	0	1
21	26	12	1001	0	1
22	27	12	1002	0	1
23	28	12	1003	0	1
24	29	12	1006	0	1
25	31	8	1002	0	1
26	32	11	1002	0	1
27	33	13	1007	0	1
28	34	13	1006	0	1
29	35	13	1005	0	1
30	36	13	1004	0	1
31	37	13	1009	1	1
32	74	18	1010	0	0

Location Table

	location_id	name	latitude	longitude	description
1	1000	UCF	28.601919	-81.200417	University
2	1001	Tomorrow Land	28.905919	-82.250417	Disney
3	1002	Meriden	41.535502	-72.799673	Connecticut City
4	1003	MIT	42.359233	-71.093151	Massachusetts Institute of Technology
5	1004	John C. Hitt Library	28.600613	-81.201259	UCF Library
6	1005	Memory Mall	28.604212	-81.199080	UCF Field
7	1006	RWC	28.595712	-81.199498	UCF Recreation and Wellness Center
8	1007	Addition Financial Arena	28.607558	-81.197202	UCF Basketball
9	1008	CB1	28.603758	-81.200498	CB1 UCF
10	1009	Buena Vista	38.836850	-106.117147	Colorado

Reactions Table

	reaction_id	e_id	user_id	comment	rating	save	rsvp
1	6	20	2	NULL	5	0	0
2	7	24	2	Had a blast!	4	0	0
3	8	22	2		4	0	0
4	9	23	2	It was kind of crowded	3	0	0
5	10	28	2	Got a bullseye!	4	0	0
6	11	25	2	I love painting	5	0	0
7	12	26	2	Wasn't super exciting..	3	0	0
8	13	24	1009	NULL	5	0	0
9	14	25	1009	That looks so realistic!	5	0	0
10	15	22	1009	Most excited I've been for a house in a while	4	0	0
11	16	23	1009	Way better experience on here	1	0	0
12	17	30	1009	Science rules!	5	0	0
13	18	21	1009	We looked great out there today	5	0	0
14	19	24	1009	I got soaked!	5	0	0
15	20	27	1009	Shoes were a good call, thanks!	4	0	0
16	21	30	1015	I kinda like science!	2	0	0
17	22	30	1015		1	0	0
18	23	34	1014	Video games and sports	4	0	0

Event_Picture Table

	p_id	e_id	picture
1	14	24	0xFFD8FFE13DB94578696600004D4D002A00000008001101...
2	15	24	0xFFD8FFE000104A46494600010101000000000000FFE1002...
3	16	23	0xFFD8FFE000104A46494600010101000000000000FFE1002...
4	17	20	0x5249464684560100574542505650382078560100B044069D...
5	18	28	0xFFD8FFE000104A46494600010101004800480000FFDB004...
6	19	29	0xFFD8FFE000104A46494600010101004800480000FFDB004...
7	20	30	0x524946461AF9000057454250565038200EF900009045059...
8	21	31	0xFFD8FFE000104A46494600010101000000000000FFE1002...
9	22	32	0xFFD8FFE000104A46494600010100000100010000FFE1006...
10	23	40	0xFFD8FFE000104A46494600010201004800480000FFE11D2...
11	24	37	0xFFD8FFE000104A46494600010101012C012C0000FFDB004...
12	25	36	0xFFD8FFE000104A46494600010101000000000000FFE1002...

Uni_Picture Table

	p_id	uni_id	picture
1	1	1	0xFFD8FFE000104A46494600010101006000600000FFFE003...
2	2	1	0xFFD8FFE000104A46494600010201004800480000FFE11D2...
3	3	1	0xFFD8FFE000104A46494600010101004800480000FFDB004...
4	4	1	0xFFD8FFE000104A46494600010101004800480000FFDB004...

Category Table

	c_id	name	entered
1	2	STEM	2023-04-14 19:25:00.710
2	3	Engineering	2023-04-14 19:25:06.963
3	4	Social	2023-04-14 19:25:13.420
4	5	Sports	2023-04-17 18:00:07.363
5	6	Computer Science	2023-04-17 22:30:45.443
6	7	New Category	2023-04-18 17:35:59.823

SQL

New RSO

When a new RSO is created, it must have 5 members that were invited. These unaccepted invitations are represented by a status of 0. rso_id and member_id are auto generated when these are inserted

```

INSERT INTO RSO (name, description, uni_id, created_by) VALUES ('RSO Name', 'RSO Description', 1, 2)

select * from RSO r
WHERE r.uni_id = 1
AND r.created_by = 2
AND r.name = 'RSO Name'

```

100 %

	rso_id	name	uni_id	status	created_by	description
1	28	RSO Name	1	0	2	RSO Description

```

INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (28, 1001, 0)
INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (28, 1002, 0)
INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (28, 1003, 0)
INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (28, 1004, 0)
INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (28, 2, 1)

select * from RSO_Member r
WHERE r.rso_id = 28

```

100 %

	member_id	rso_id	user_id	is_admin	status
1	153	28	1001	0	0
2	154	28	1002	0	0
3	155	28	1003	0	0
4	156	28	1004	0	0
5	157	28	2	0	1

Join RSO

```

INSERT INTO RSO_Member(rso_id, user_id, status) VALUES (27, 2, 1)

select * from RSO_Member r
WHERE r.rso_id = 27
AND r.user_id = 2

```

100 %

Results Messages

	member_id	rso_id	user_id	is_admin	status
1	158	27	2	0	1

New Event

```

INSERT INTO Event(location_id, rso_id, uni_id, name, description, c_id, visibility, start_time, end_time, status, phone, email)
VALUES (1000, 28, 1, 'name', 'desc', 2, 'visibility', 100, 101, 1, 'phone', 'email')

select * from Event e
WHERE e.name = 'name'
AND e.description = 'desc'

```

100 %

Results Messages

	e_id	location_id	rso_id	uni_id	name	c_id	visibility	description	start_time	end_time	status	phone	email
1	50	1000	28	1	name	2	visibility	desc	1900-04-11 00:00:00.000	1900-04-12 00:00:00.000	1	phone	email

New Comment

```

INSERT INTO Reactions(e_id, user_id, comment) VALUES (50, 2, 'comment text')

select * from Reactions r
WHERE r.comment = 'comment text'

```

100 %

Results Messages

	reaction_id	e_id	user_id	comment	rating	save	rsvp
1	40	50	2	comment text	NULL	0	0

Queries

Queries to display events—public, private, and RSO-- (part of the processing of the ‘View Event.’ request by a user with a specific role). Note, I consider University events to be private and thus store it as such.

```
SELECT * FROM Event e
WHERE e.visibility = 'Public'
```

	e_id	location_id	rso_id	uni_id	name	c_id	visibility	description	start_time	end_time	status	phone	email
1	24	1005	7	1	Water balloon fight	4	Public	Visit UCF for a community water balloon fight	2023-04-26 12:00:00.000	2023-04-19 13:30:00.000	1	4075924262	recruiting@ucf.edu
2	25	1000	13	1	Painting	4	Public	Public painting class	2023-04-19 07:00:00.000	2023-04-19 08:00:00.000	1	40752645225	paint@ucf.edu
3	26	1003	13	1	Chess Competition	3	Public	Chess battle live	2023-04-27 12:00:00.000	2023-04-27 13:00:00.000	1	4075951212	this@that.com
4	35	1004	7	1	Group Reading	4	Public	UCF Reading	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4071599568	ucf@ucf.edu
5	42	1002	9	1	Public E	2	Public	This is a STEM event	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	454564569	fdsa@a.df
6	43	1000	7	1	Name	5	Public	This is public	2023-04-18 12:00:00.000	2023-04-18 13:00:00.000	1	456456456	456456@gdfsa.com

```
SELECT * FROM Event e
WHERE e.visibility = 'University'
```

	e_id	location_id	rso_id	uni_id	name	c_id	visibility	description	start_time	end_time	status	phone	email
1	22	1000	13	1	Open House	4	University	Open house at UCF	2023-04-22 12:00:00.000	2023-04-22 19:00:00.000	1	4078438456	sa@ucf.edu
2	23	1005	7	1	RSO Signups	4	University	Come sign up for some RSO's!	2023-04-30 12:00:00.000	2023-04-30 16:00:00.000	1	4074510622	rso@ucf.edu
3	28	1006	12	1	Archery Lessons	5	University	Archery	2023-04-01 12:00:00.000	2023-04-01 13:00:00.000	1	4075952123	archery@ucf.edu
4	29	1006	9	1	Pick up Basketball	5	University	Pick up game at the courts	2023-04-06 12:00:00.000	2023-04-06 13:00:00.000	1	4075955995	bb@bb.com
5	30	1001	9	1	Science Fair	2	University	UCF Science Fair	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075955877	science@ucf.edu
6	31	1007	10	1	Golf Tryouts	5	University	Golf Tryouts	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075125998	golf@ucf.edu
7	33	1002	7	1	Nursing Seminar	2	University	Nursing day	2023-04-20 12:00:00.000	2023-04-20 13:00:00.000	1	4075911211	nurse@ucf.edu
8	37	1006	7	1	Pickleball	4	University	Pickleball!	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075959596	pickle@ucf.edu
9	39	1000	7	1	Algebra	2	University	Math	2023-05-20 12:00:00.000	2023-05-20 13:00:00.000	2	4075961238	math@ucf.edu
10	48	1009	8	1	fdsafads	5	University	jkifdsa	2023-04-18 12:00:00.000	2023-04-18 13:00:00.000	1	fdsafdsa	fdsafdsa

```
SELECT * FROM Event e
WHERE e.visibility = 'RSO'
```

	e_id	location_id	rso_id	uni_id	name	c_id	visibility	description	start_time	end_time	status	phone	email
1	20	1000	11	1	Movie Night	4	RSO	Come out and watch a movie with us!	2023-04-17 18:00:00.000	2023-04-17 21:00:00.000	1	407-957-2591	movie@knights.ucf.edu
2	21	1005	13	1	First Practice	5	RSO	First practice of the season. Bring water!	2023-04-19 17:00:00.000	2023-04-19 19:30:00.000	1	407-158-9286	marching@knights.ucf.edu
3	27	1001	13	1	Chess Meeting	2	RSO	Chess practice, bring shoes	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	407891259	chess@ucf.edu
4	32	1000	7	1	Disk golf	2	RSO	Disk Golf	2023-04-17 12:00:00.000	2023-04-17 13:00:00.000	1	4075912375	dg@ucf.edu
5	34	1004	12	1	ESports	4	RSO	Esports	2023-04-17 14:00:00.000	2023-04-17 15:00:00.000	1	4170585895	esports@ucf.edu
6	36	1003	7	1	Event Events	2	RSO	This is an event	2023-03-26 12:00:00.000	2023-03-26 13:00:00.000	1	4075956589	event@ucf.edu
7	40	1007	7	1	Biking	5	RSO	Biking at the uni	2023-04-25 12:00:00.000	2023-04-25 13:00:00.000	1	4075954584	bike@ucf.edu
8	47	1001	10	1	fdsa	7	RSO	Test	2023-04-08 12:00:00.000	2023-04-08 13:00:00.000	1	fdsa	fdas

Constraint Enforcement

Conflicting Event Times

Create

Event

- Timeframe overlaps with event "Pickleball" at location "RWC". Please pick another time or try a different location

Event for Wrong RSO

The way my application operates, users don't even get the opportunity to create events for other RSO's of which they aren't an admin for. I do this by only populating the form with RSO's of which they are an admin of. It goes a step further if they are only the admin of one RSO, setting the RSO field for them and hiding it.

RSO Status

*Note: The way I structured my database, a status of 0 represents student approval pending, 1 represents student approved, pending super admin approval and 4 represents disbanded where there once was 5 members but now there is not.

```

SELECT RM.rso_id, COUNT(*) AS 'Num members', R.status
FROM RSO_Member RM
INNER JOIN RSO R ON RM.rso_id = R.rso_id
WHERE RM.rso_id = 7 AND R.rso_id = 7
GROUP BY RM.rso_id, R.status

INSERT INTO RSO_Member (rso_id, user_id, status) VALUES (7, 2, 1)

SELECT RM.rso_id, COUNT(*) AS 'Num members', R.status
FROM RSO_Member RM
INNER JOIN RSO R ON RM.rso_id = R.rso_id
WHERE RM.rso_id = 7 AND R.rso_id = 7
GROUP BY RM.rso_id, R.status

DELETE FROM RSO_Member
WHERE rso_id = 7 AND user_id = 2

SELECT RM.rso_id, COUNT(*) AS 'Num members', R.status
FROM RSO_Member RM
INNER JOIN RSO R ON RM.rso_id = R.rso_id
WHERE RM.rso_id = 7 AND R.rso_id = 7
GROUP BY RM.rso_id, R.status

```

00 %

Results Messages

	rso_id	Num members	status
1	7	4	0

	rso_id	Num members	status
1	7	5	1

	rso_id	Num members	status
1	7	4	4

Demo

I submitted two demos because I didn't realize how long my first one was and that the instructions recommended that it stay under 15 minutes. During my first demo, I found out that my rating represented by either full stars or outlines wasn't working properly. I immediately knew why and fixed it soon after and it was working correctly for the second, shorter demo. Other than that, everything else worked perfectly both times.

Conclusion

Database Performance

I am quite happy with my database performance. Using the views regularly and creating indexes for everything that I queried data on greatly improved database performance. The only thing that surprises me is the amount of time it takes to load the University landing page. I think this is due to using a couple of rather large pictures for my university though.

Desired Features

One desired feature that I have is to be able to save events. While this feature can be easily implemented and the database is already set up for it, I didn't have the time or effort to do it in the time that the project was due. Another similar feature that I would like to implement is the ability to RSVP for events. The database is also set up for this and could be easily implemented. One final thing that I might implement in the future is the automatic deletion of events after so much time has passed from the end time.

Problems encountered

One problem that I kept encountering is I had to make changes to the database on multiple occasions. I do believe that this was due to the generalness of the project and the personal interpretation that was put into it. If I were to do it again, I would scaffold out the requirements of the application before starting, similar to a SCRUM approach. With that said, I have learned almost everything about web application development during this project. I watch a lot of videos on it and learned how to develop in a C# ASP.NET Core 6.0 environment and how to separate things in MVC. If I were to start over again, I might choose to use a newer method for single person development like Blazor, though that isn't as wide used in industry. I could also work on my front end creativity and breaking outside of just using Bootstrap 5.