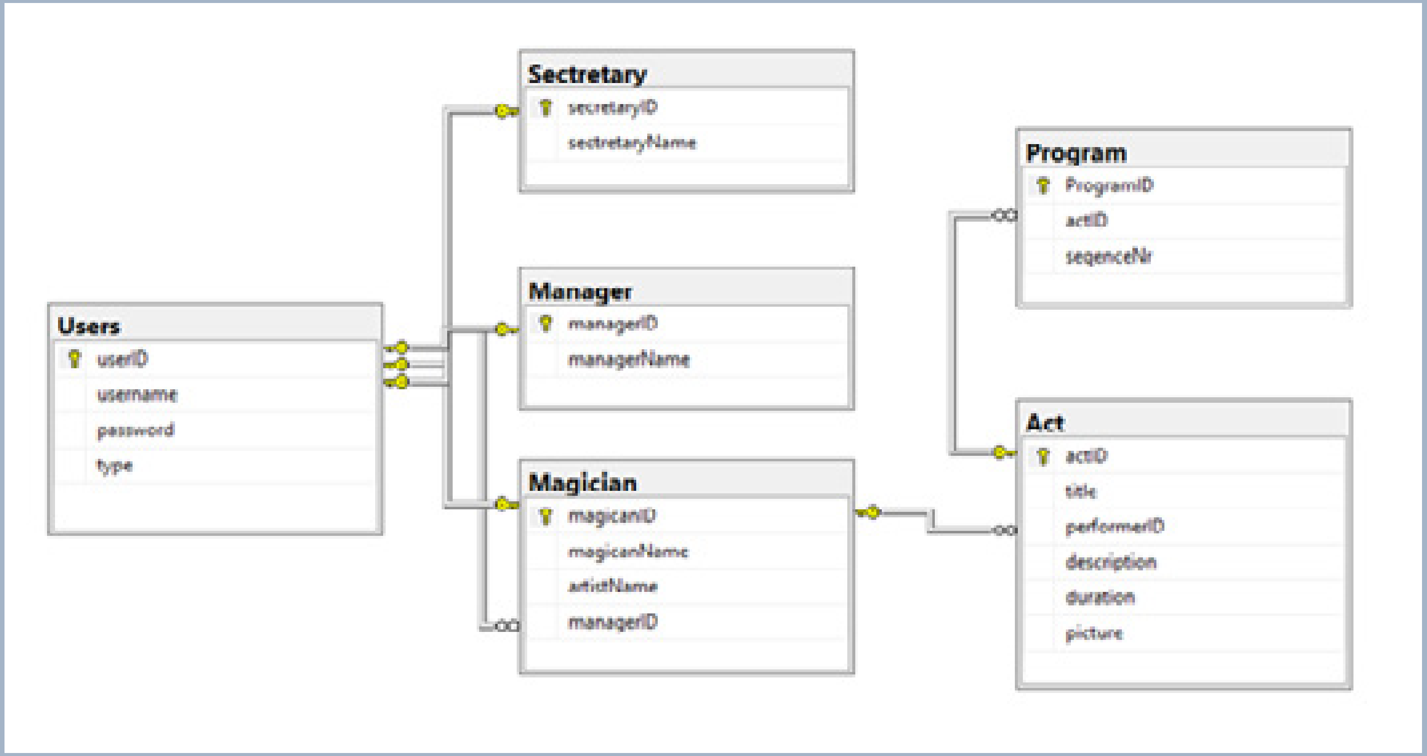


Las Vegas Show

Make a database for Las Vegas Magical Stage.
Make a corresponding web site.
Make a table Magician with id, name, artist name and password. This table must also hold a manager and one or more secretaries.
Make a table Act with title, performing magician, short description, duration and a link to a picture.
Make a table Program with id, act and sequence number.

Because I didn't thought that a magician could be a secretary or manager I made three "type" tables, Secretary, Manager and Magician.

All of them has a primary key that is foreign key from the Users table.

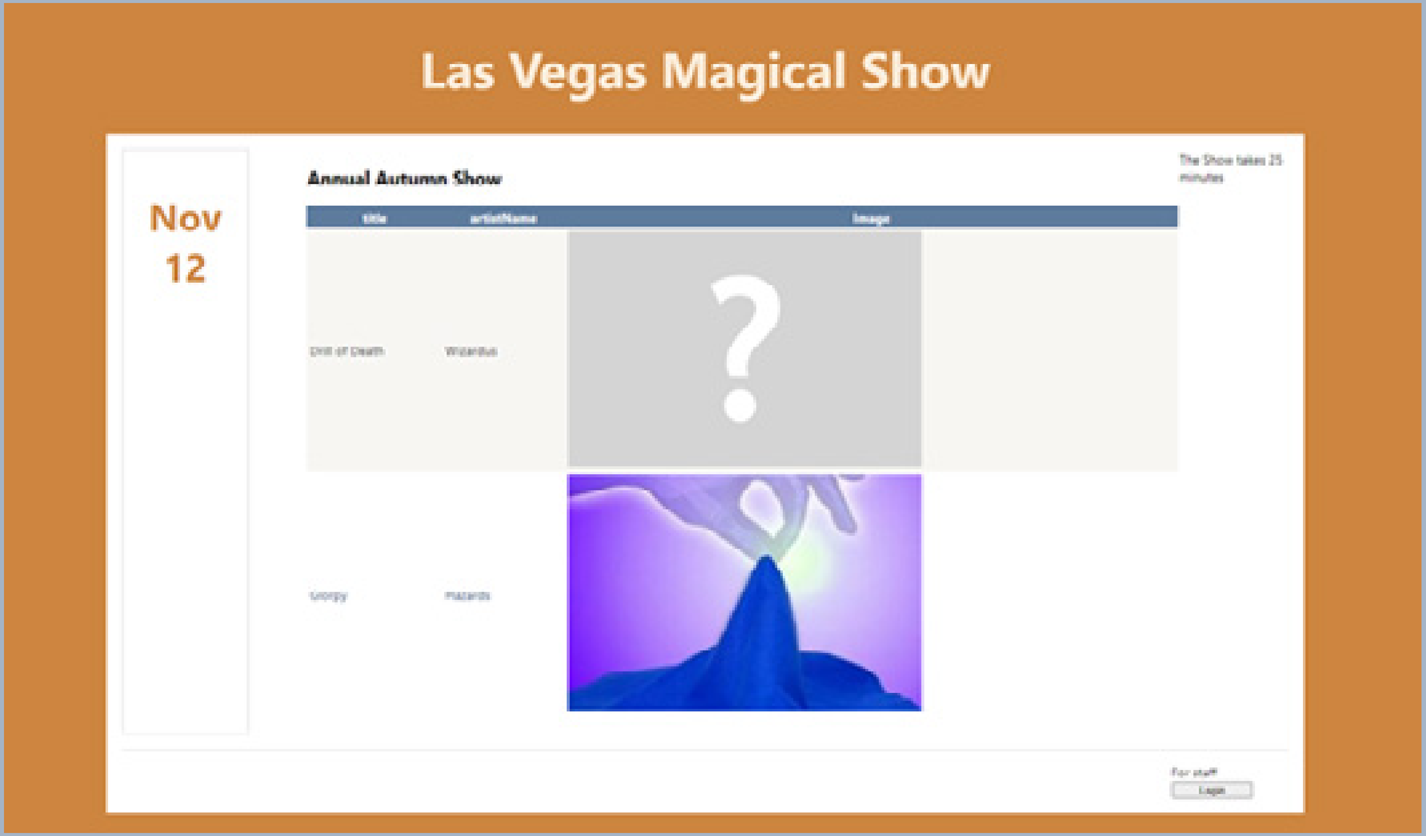
Besides that, I have made an Act table where performerID is a foreign key to magicianID. And the actID shows up in the Program table.

Las Vegas Show

The site must have a public welcome page showing the program (with pictures) plus total duration of the performance.

To show the program I chose to use a simple GridView. The last column is a ImageField to show the pictures from the link each picture fields has.

To show the duration of the show in the right top corner on the Index page; I used a SELECT SUM() sql query, which send back a table with one column with the Sum (rdr.GetValue(0)) of all durations from the acts.



Las Vegas Magical Show

Nov 12

Annual Autumn Show

The Show takes 25 minutes

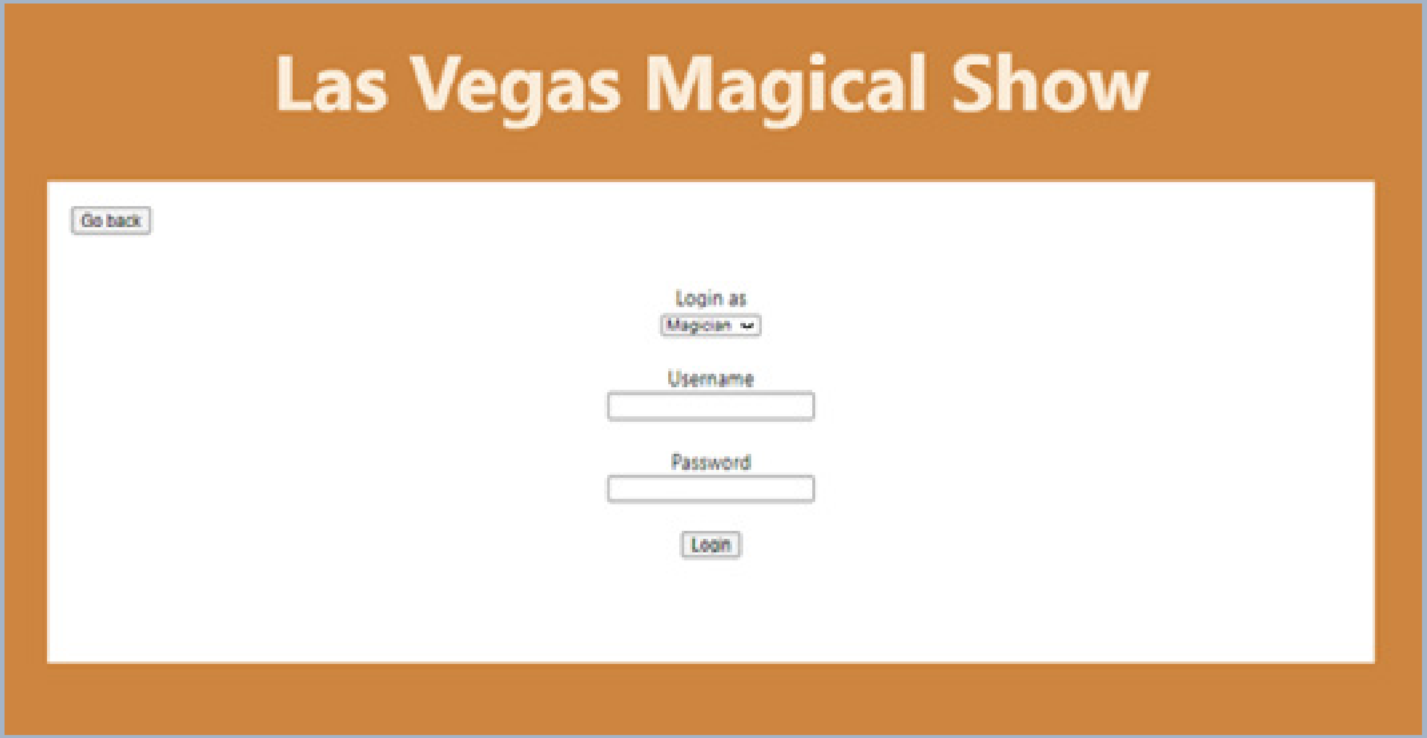
title	artistName	Image
Drill of Death	Wizardus	
[illegible]	Hazards	

For staff

Login

```
conn.Open();
    try
    {
        Sqlselection = "SELECT SUM(Duration) FROM Act INNER JOIN Program ON Act.actID
        = Program.actID";
        cmd = new SqlCommand(Sqlselection, conn);
        rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            LabelDuration.Text = "The Show takes " + rdr.GetValue(0).ToString() + "
            minutes";
         }
     }
     finally
     {
         conn.Close();
     }
```

Las Vegas Show



The site must have a simple login and logout page for all users.

To make it simple for myself, a made dropdown with the three types of staff, that sends the user to a different kind of profile page based on the selected type.

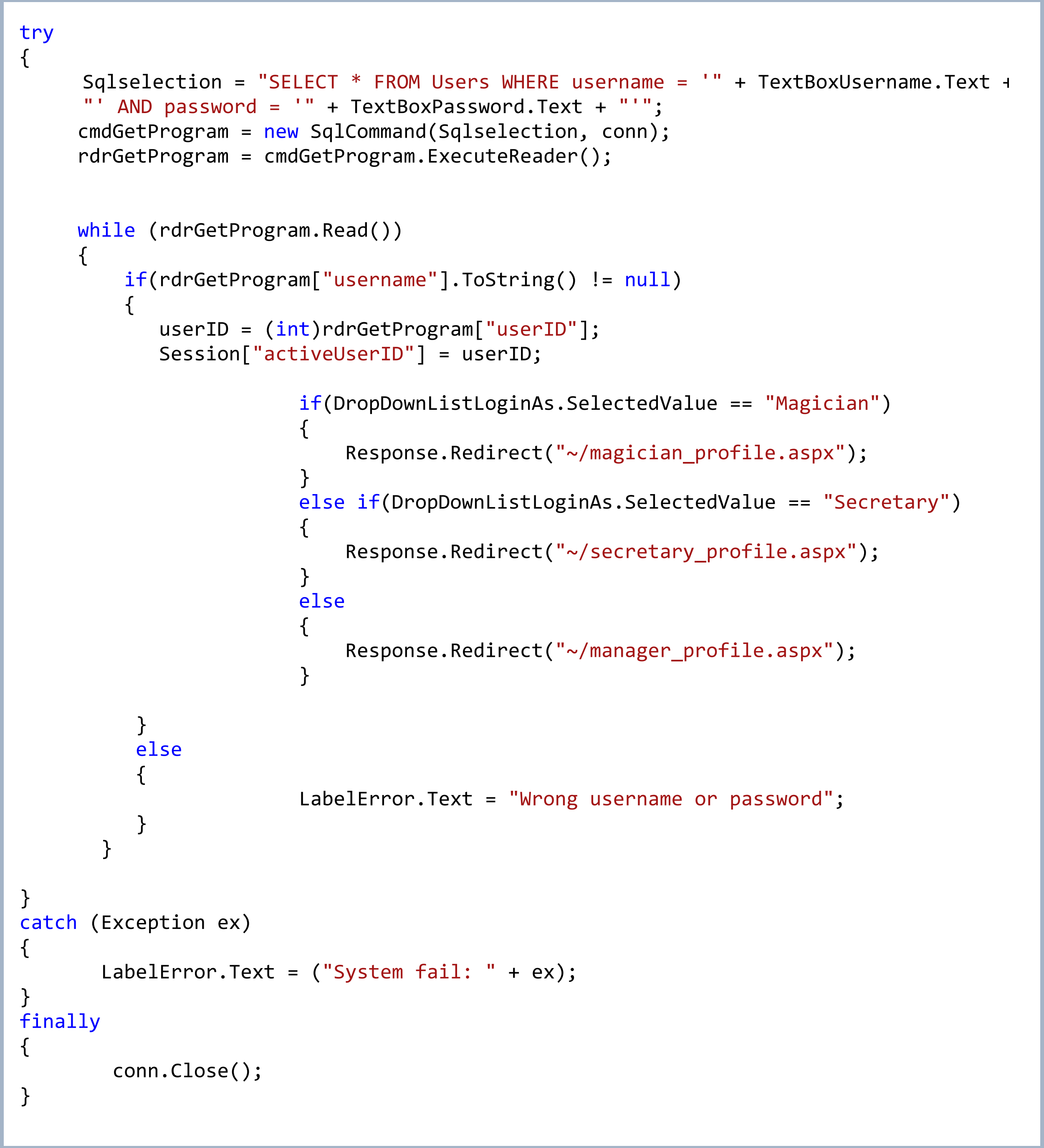
Las Vegas Show

But before that, it checks if the username and password exist in the Users table and in the same row.

First it tries to select a row by the Sqlselection string.

Secondly, it checks if the username in the result is empty, if not the username and password is correct.

At last, it sets the Session["activeUserID"] to the userID from the result before it sends the user to the profile page.



```
try
{
     Sqlselection = "SELECT * FROM Users WHERE username = '" + TextBoxUsername.Text
     "' AND password = '" + TextBoxPassword.Text + "'";
     cmdGetProgram = new SqlCommand(Sqlselection, conn);
     rdrGetProgram = cmdGetProgram.ExecuteReader();


     while (rdrGetProgram.Read())
     {
         if(rdrGetProgram["username"].ToString() != null)
         {
            userID = (int)rdrGetProgram["userID"];
            Session["activeUserID"] = userID;

                        if(DropDownListLoginAs.SelectedValue == "Magician")
                        {
                            Response.Redirect("~/magician_profile.aspx");
                        }
                        else if(DropDownListLoginAs.SelectedValue == "Secretary")
                        {
                            Response.Redirect("~/secretary_profile.aspx");
                        }
                        else
                        {
                            Response.Redirect("~/manager_profile.aspx");
                        }

          }
          else
          {
                        LabelError.Text = "Wrong username or password";

          }
       }

}
catch (Exception ex)
{
       LabelError.Text = ("System fail: " + ex);
}
finally
{
        conn.Close();
}
```

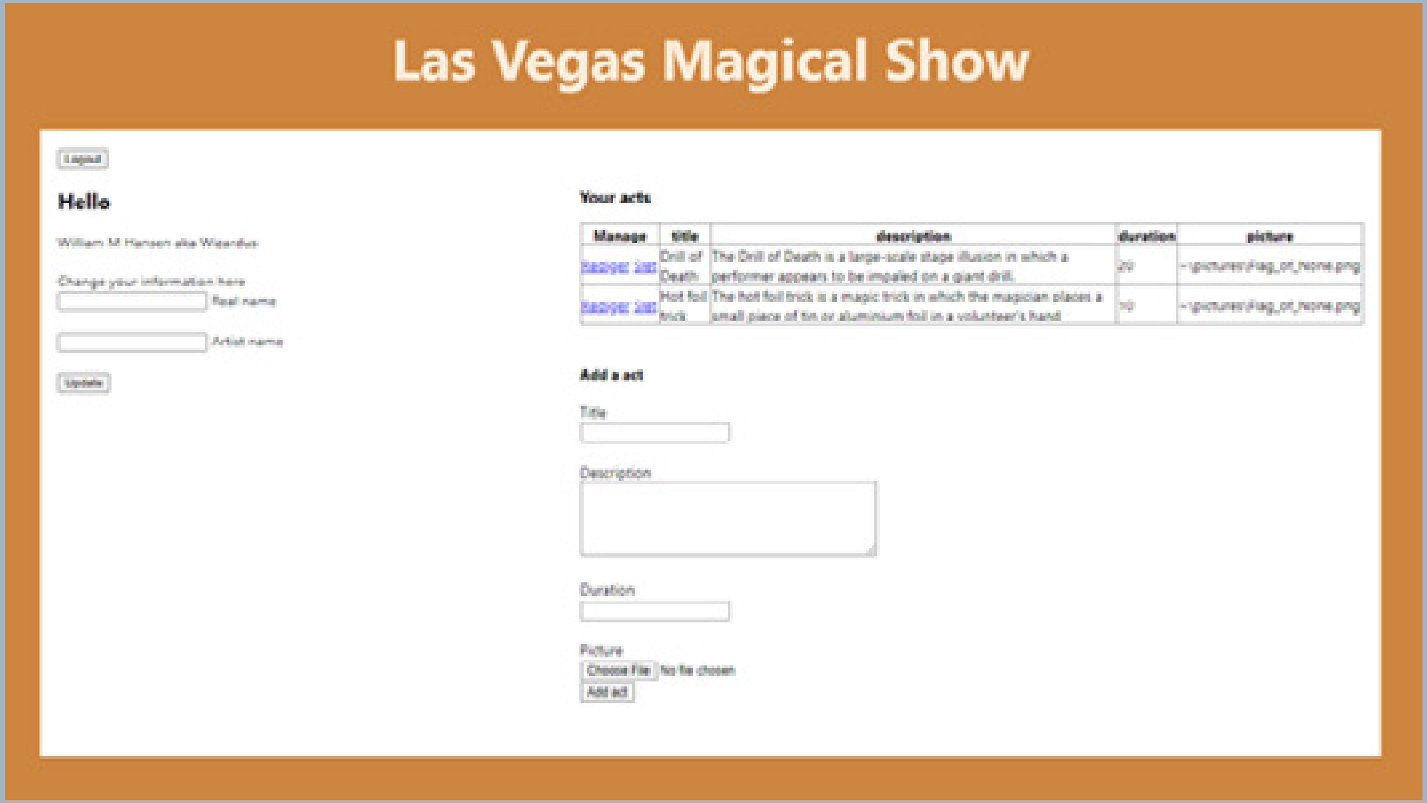
```
private void SetUpPage()
{
     if (userID != 0)
     {
                string query = "SELECT * FROM Manager WHERE managerID = @userID";
                conn.Open();
                try
                {
                    SqlCommand cmdGetPersonData = conn.CreateCommand();
                    cmdGetPersonData.CommandText = query;
                    cmdGetPersonData.Parameters.Add("@userID", SqlDbType.Int);
                    cmdGetPersonData.Parameters["@userID"].Value = userID;
                    rdr = cmdGetPersonData.ExecuteReader();

                    if (rdr.HasRows)
                    {
                        while (rdr.Read())
                        {
                            LabelName.Text = rdr.GetValue(1).ToString();
                        }
                    }
                    else
                    {
                        Response.Redirect("~/no_allowence.aspx");
                    }

                }
                catch (SqlException ex)
                {
                    LabelError.Text = "SQL error \n" + ex;
                }
                finally
                {
                    conn.Close();
                }

       }
       else
       {
                Response.Redirect("~/no_allowence.aspx");
       }
}
```

Las Vegas Show

On each profile page the SetUpPage function checks if the userID exist in the manager-, magician- or secretary table, depended on which type the the user selected in the login page.

Las Vegas Show

If a magician logs in the magician should be able to CRUD (create, read, update, delete) own acts. Try to use a trigger to add a default picture link to acts with no picture data. Make it possible to upload pictures to a folder in the project.



Las Vegas Magical Show

Logout

Hello

William M Hansen aka Wizardus

Change your information here

Real name

Artist name

Update

Your acts

Manage	title	description	duration	picture
Rediger Slet	Drill of Death	The Drill of Death is a large-scale stage illusion in which a performer appears to be impaled on a giant drill.	20	~\pictures\flag_of_none.png
Rediger Slet	Hot foil trick	The hot foil trick is a magic trick in which the magician places a small piece of tin or aluminium foil in a volunteer's hand	10	~\pictures\flag_of_none.png

Add a act

Title

Description

Duration

Picture

Choose File No file chosen

Add act

In the left side, we can see who is logged in, and the magician can change his/hers own Real name and artist name.

In the right side, the magician can see information about his/hers acts, shown in a GridView.

In the Gridview the magician can Update and Delete the acts.

Under the GridView, the magician can add a new act.

Las Vegas Show

```
if(TextBoxRealName.Text != "")
{
        try
        {
                conn.Open();
                SqlCommand cmdUdateRealName = conn.CreateCommand();
                string sqlUpdate = "UPDATE Magician SET magicanName = @newRealName WHERE
                magicanID = @userID";
                cmdUdateRealName.CommandText = sqlUpdate;
                cmdUdateRealName.Parameters.Add("@newRealName", SqlDbType.NVarChar, 50);
                cmdUdateRealName.Parameters.Add("@userID", SqlDbType.Int);

                cmdUdateRealName.Parameters["@newRealName"].Value = TextBoxRealName.Text
                cmdUdateRealName.Parameters["@userID"].Value = userID;
                cmdUdateRealName.ExecuteNonQuery();
         }
         catch (SqlException ex)
         {
                 LabelError.Text = "Sql error: \n" + ex.Message;
         }
         catch (Exception ex)
         {
                 LabelError.Text = "System error\n" + ex.Message;
         }
         finally
         {
                 TextBoxRealName.Text = "";
                 conn.Close();
          }
}
```

Example of updating the Real name in the database (userID is the same as Session["activeUserID"])

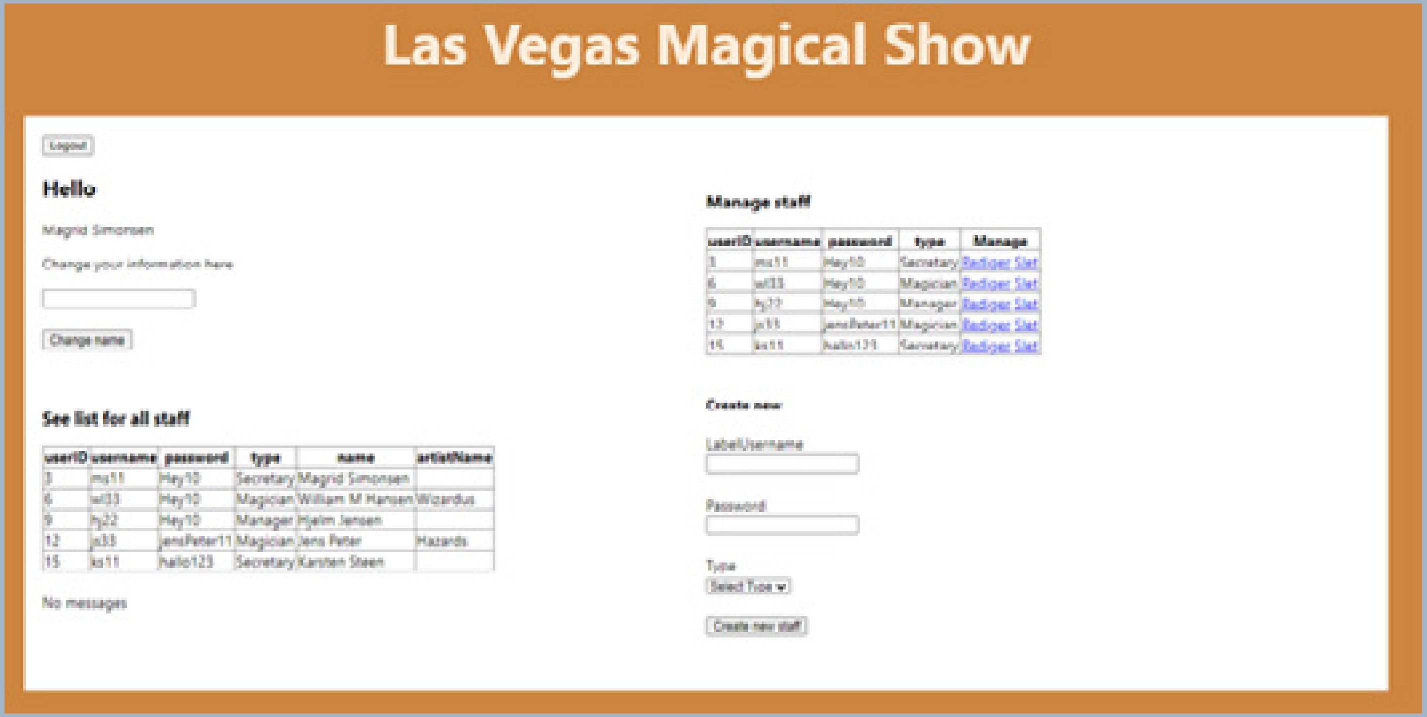
Las Vegas Show

Code behind what happens then the "Add act" button is pushed.

It checks if there is a file uploaded. If not, it will insert the default image url. If there is a file it will save the image in the "savepath" url, and make the filename that will be inserted in the table.

```
string filename;
string savePath = @"C:\Users\Sara\OneDrive -
EFIF\Webudvikling\Backend\CS\LasVegasShow\pictures\";
if (FileUploadPicture.HasFile)
{
     filename = @"~\pictures\" + FileUploadPicture.FileName;
     savePath += filename;
     FileUploadPicture.SaveAs(savePath);
} else
{
     filename = @"~\pictures\Flag_of_None.png";
     FileUploadPicture.SaveAs(savePath + "Flag_of_None.png");
}

conn.Open();
try
{
     SqlCommand cmdAddAct = conn.CreateCommand();
      string sqlins = "INSERT INTO Act (title, performerID, description, duration,
      picture) VALUES (@title, @performerID, @description, @duration, @picture)";
     cmdAddAct.CommandText = sqlins;
     cmdAddAct.Parameters.Add("@title", SqlDbType.NVarChar, 50);
     cmdAddAct.Parameters.Add("@performerID", SqlDbType.Int);
     cmdAddAct.Parameters.Add("@description", SqlDbType.NVarChar, 300);
     cmdAddAct.Parameters.Add("@duration", SqlDbType.Int);
     cmdAddAct.Parameters.Add("@picture", SqlDbType.NVarChar, 50);

     cmdAddAct.Parameters["@title"].Value = TextBoxTitle.Text;
     cmdAddAct.Parameters["@performerID"].Value = userID;
     cmdAddAct.Parameters["@description"].Value = TextBoxDescription.Text;
     cmdAddAct.Parameters["@duration"].Value = TextBoxDuration.Text;
     cmdAddAct.Parameters["@picture"].Value = filename;

     cmdAddAct.ExecuteNonQuery();
     TextBoxTitle.Text = "";
     TextBoxDescription.Text = "";
     TextBoxDuration.Text = "";
}
catch (SqlException ex)
{
      LabelMessage.Text = "SQL fail: \n" + ex.Message;
}
catch (Exception ex)
{
      LabelMessage.Text = "System error\n" + ex.Message;
}
      finally
{
      conn.Close();
      SetUpPage();
}
```

Las Vegas Show



Las Vegas Magical Show

Hello

Magrid Simonsen

Change your information here

See list for all staff

userID	username	password	type	name	artistName
3	ms11	Hey10	Secretary	Magrid Simonsen	
6	wl33	Hey10	Magician	William M Hansen	Wizardus
9	hj22	Hey10	Manager	Hjalm Jensen	
12	js33	jensPeter11	Magician	Jens Peter	Hazards
15	ks11	hallo123	Secretary	Karsten Steen	

No messages

Manage staff

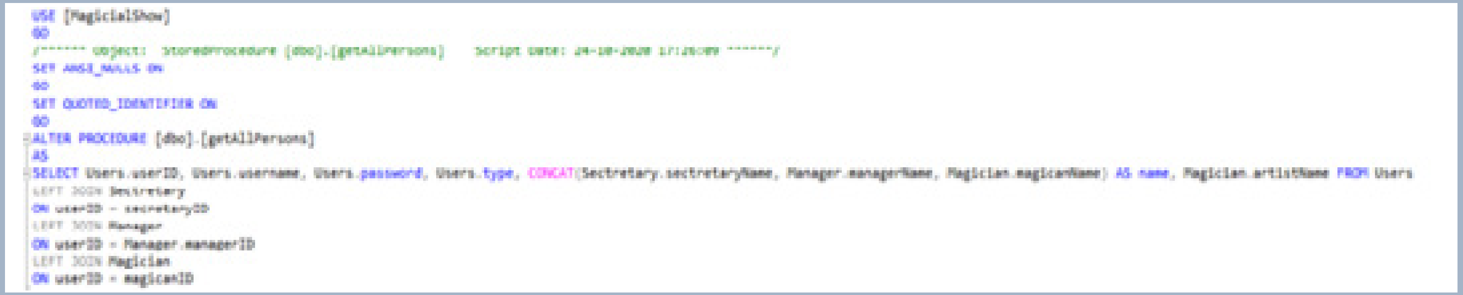
userID	username	password	type	Manage
3	ms11	Hey10	Secretary	Rediger Slet
6	wl33	Hey10	Magician	Rediger Slet
9	hj22	Hey10	Manager	Rediger Slet
12	js33	jensPeter11	Magician	Rediger Slet
15	ks11	hallo123	Secretary	Rediger Slet

Create new

LabelUsername

Password

Type



```sql
USE [MagicialShow]
GO
/****** Object:  StoredProcedure [dbo].[getAllPersons]    Script Date: 24-10-2020 17:25:09 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[getAllPersons]
AS
SELECT Users.userID, Users.username, Users.password, Users.type, CONCAT(Sectretary.sectretaryName, Manager.managerName, Magician.magicanName) AS name, Magician.artistName FROM Users
LEFT JOIN Sectretary
ON userID = secretaryID
LEFT JOIN Manager
ON userID = Manager.managerID
LEFT JOIN Magician
ON userID = magicanID
```

If a secretary logs in the secretary should be allowed to CRUD (create, read, update, delete) the persons.

The Secretary profile page is almost a copy of the magician profile page, just with access to the Users table instead of the Act.

I decided that a secretary can't change other staff's name or artist name. But that the secretary should be able to see it.

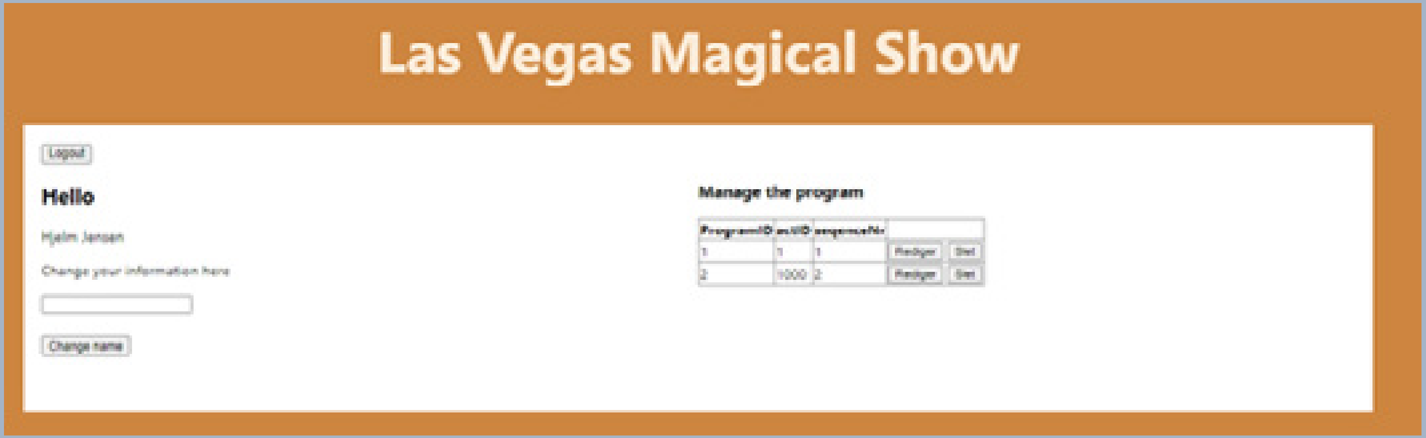
Therefore I made a GridView in the left side which use this stored procedure (getAllPersons) as the data source.

The getAllPersons procedure selects all the users and combines the three name columns into one by the CONCAT method.

Las Vegas Show

If a manager logs in the manager should be able to select acts and their sequence for the program.

The Manager profile page is like the secretary page almost a copy, besides that it's only the Program table that the managers can access.



Las Vegas Magical Show

Hello

Hjalm Jensen

Change your information here

Manage the program

ProgramID	actID	sequenceNr		
1	1	1	Rediger	Slet
2	1000	2	Rediger	Slet