Tutorial for Beginners - Setting Up A Online Scoreboard for Games made using Unity and C-Sharp

James

February 16, 2019

0 Brief Introduction

I decided to make this tutorial after teaching myself how to do this by following other tutorials and running into allot probelms with them. Though they were helpful and I was able to eventually figure things out they were assuming I had allot of knowledge I didn't and were mostly so out of date the SQL and php code they recommended wouldn't run due to deprication. I'm aiming this tutorial at complete beginners to php and SQL and webservers hopefully it will save someone a large portion of the 12+ hours of blind troubleshooting I went through.

Before we get started here's a quick overview of how we're going to create our online scoreboard that we can add/retrieve highscores from with-in our game.

1. Create a online scores table hosted on a webserver called `https://www.000webhost.com/`.

   note - you could use other webservers or setup your own we just happen to be using 000webhost as it's the one that worked for me.

2. Create two php webpages to send and retrieve scores from/to our online scores table, these will need to be hosted on the same webserver we host our scores table.

3. Create a scoreboard and a submission form in our game that will be able to send/retrieve scores to/from our php webpages via their URLs.

   note - you could use another game engine or application builder we just happen to be using unity for this part.

Ones we're done you should have a scoreboard that operates simlairly to this one `https://documentary-ores.000webhostapp.com/`
Make sure to use adobe reader as other readers may not let you copy the code correctly.

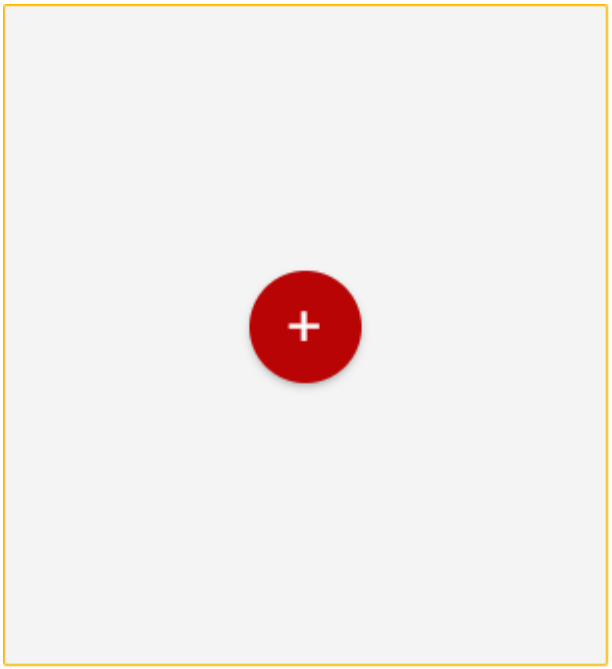
1 Creating a online scores table.

1.1 Create an account and verify your email address

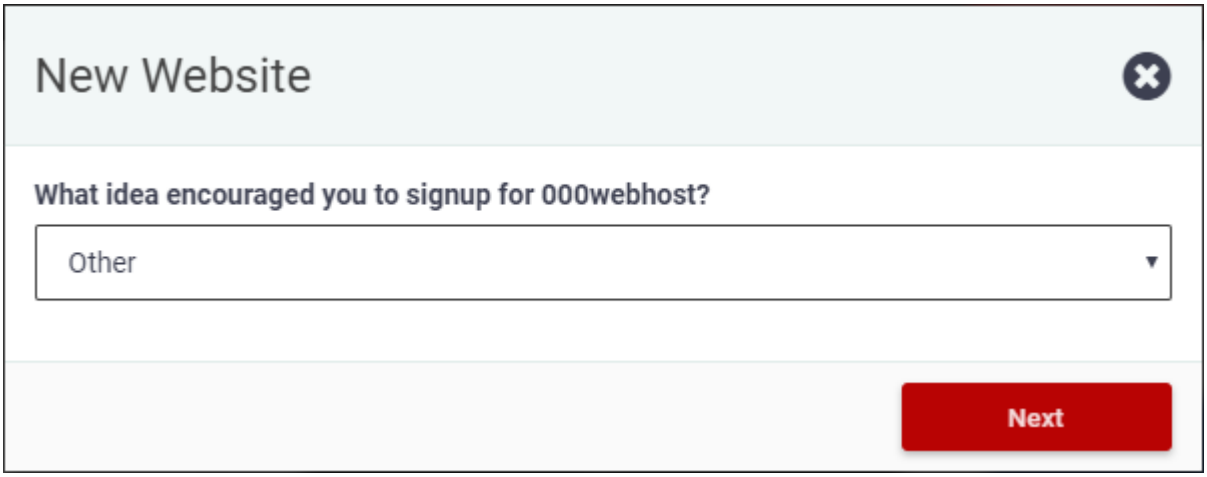
`https://www.000webhost.com/cpanel-login`

1.2 Setting up the webserver

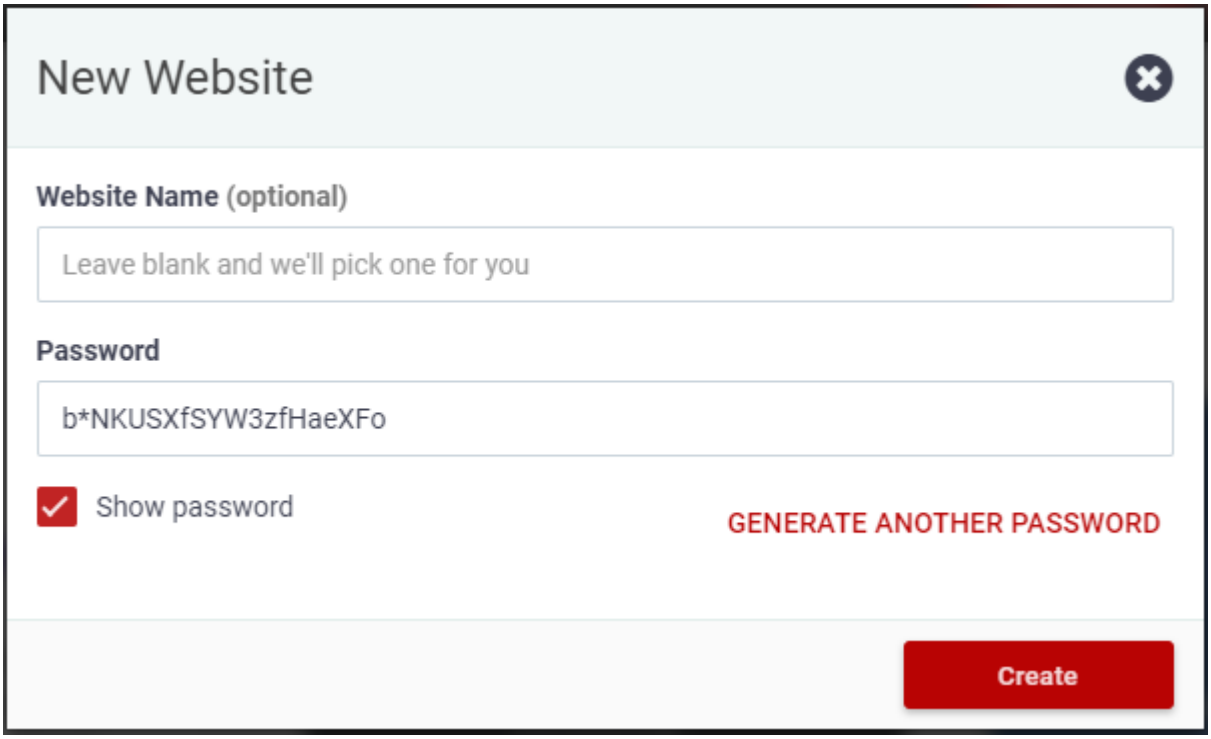
1. Ones you've verfied your email address you'll be redirected to a newuser part of their site, click on the rectangal with a + in the centre as showen below.



2. Select other in the pulldown menu in the survey box and click next.

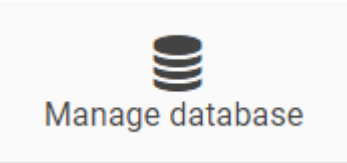


3. Type in a website name if you wish or leave it blank it doesn't matter you can take a note of your websites password if you like though for this tutorial it won't be used for anything, then click create.

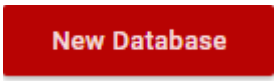


1.3 Create a database

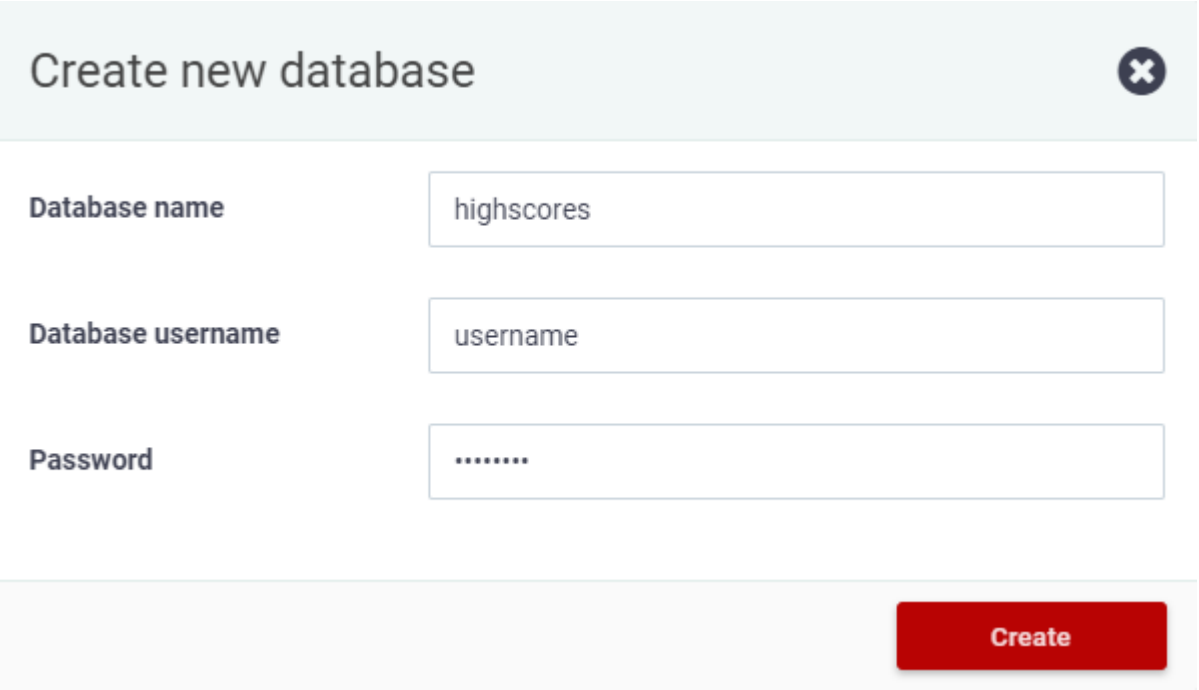
1. Ones the websites been created click on manage database.



2. Ones that's opened click on new database



3. A form should pop-up fill it in then click Create.

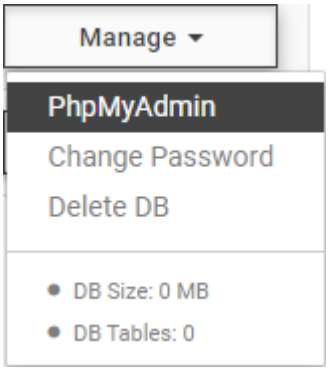


4. It should start creating your database, take note of the name of the database (written under DB Name) and the new username (written under DB User) the password you set for the database will coresspond with the new username but not the username you set in the previous step the website seems to just take that username as a suggestion.

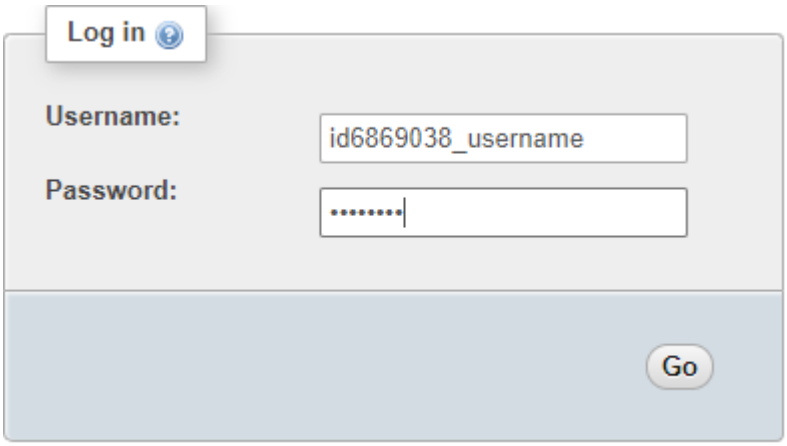
DB Name	DB User	DB Host	
id6869038_highsco...	id6869038_userna...	localhost	Creating database

1.4 Create a high-scores table in your database

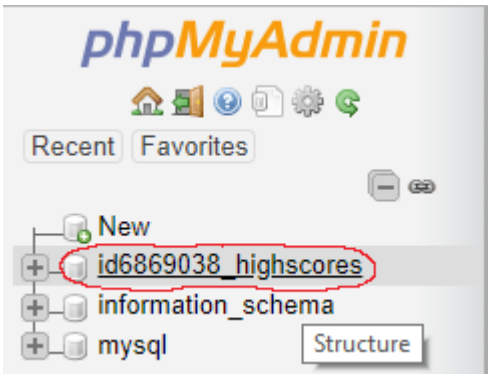
1. Ones it's finnished creating the databse (shouldn't take more than 20-60s) click on manage and then PhpMyAdmin.



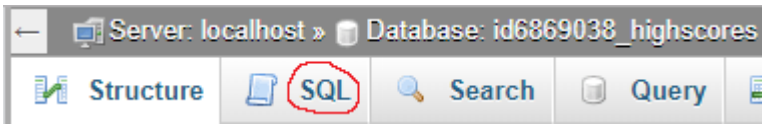
2. If it brings you to a login screen use the new username that the website set for you with the coressponding passowrd, otherwise skip this step.



3. Now click on the name of the database we just created, in this example id6869038_highscores.

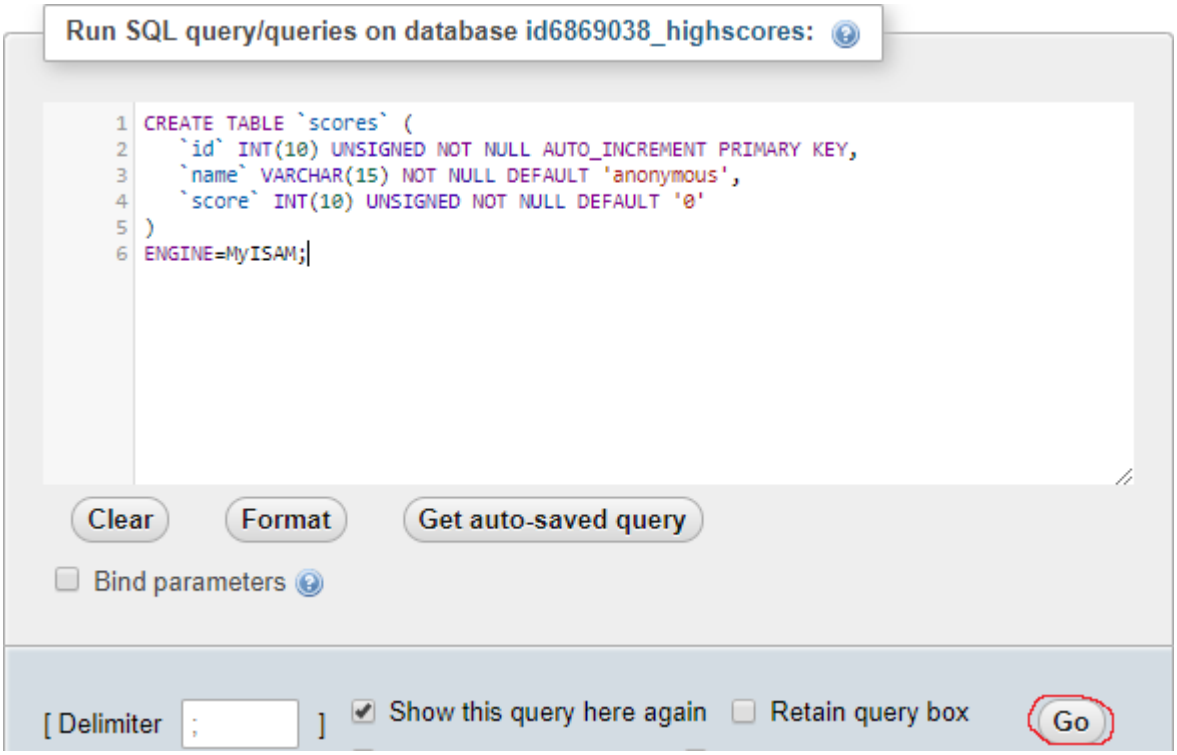


4. Ones it's loaded click on SQL.



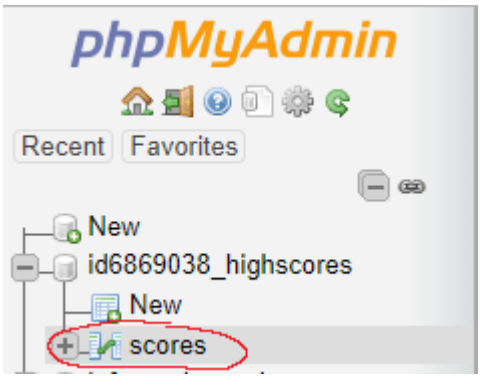
5. Add the following code to the text box as showen and click Go.

```
CREATE TABLE `scores` (
   `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
   `name` VARCHAR(15) NOT NULL DEFAULT 'anonymous',
   `score` INT(10) UNSIGNED NOT NULL DEFAULT '0'
)
ENGINE=MyISAM;
```



(This creates the highscores table)

6. If sucessful a scores option should appear below were your database is listed as showen, if you don't see a scores option click on the + sign to the left of your database in this example to the left of id6869038_highscores. Now click on the scores option.



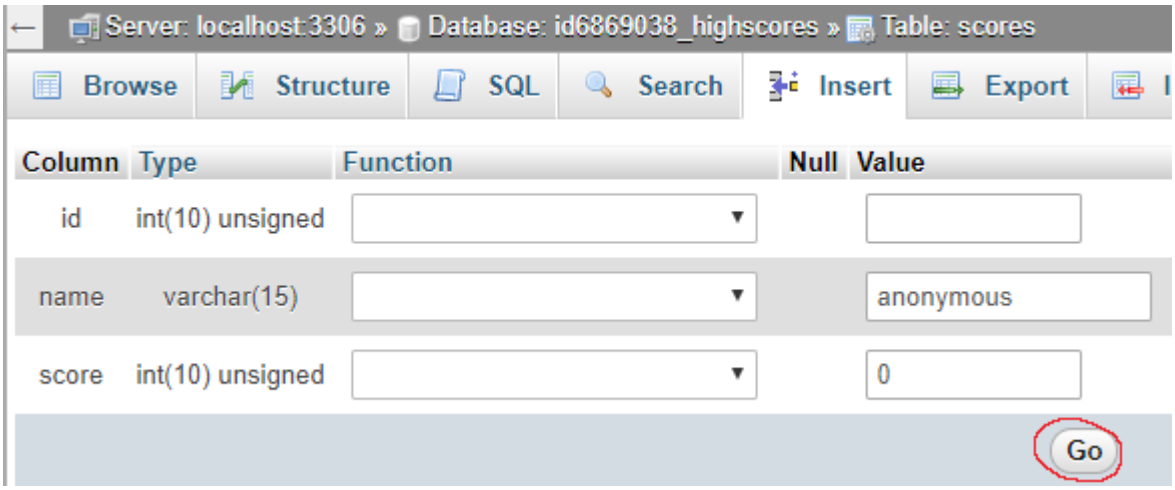
7. You should be able to see your tables currently empty we're going to add one entry to the table as this will be helpful in testing things later on.

id	name	score

1.5 Adding an entry to our table

1. Click on Insert

2. On the new page click Go as showen.



3. It should of added the entry, it redirects us to the SQL page when we add entries for some reason, if you click on Browse we can double check that the entries been added.

4. You should now see the new entry added to the table as showen below.

id	name	score
1	anonymous	0

As we add names and scores to the database their id numbers will be automatically generated counting up from 1 this will help avoid exact duplicate entries but doesn't serve any other purpose in this tutorial.

We're now ready to start creating our php files and uploading them to our website.

2 Create and upload the php files

2.1 Creating addscore.php and display.php

1. Create two files on your computer called addscore.php and display.php.

2. Open addscore.php with notepad or you favored code editor and add the following:

```
<?php
  $servername = "localhost";
  $username = "id6869038_username";
  $password = "password";
  $dbName = "id6869038_highscores";

    // Send variables for the MySQL database class.
    $conn = new mysqli($servername , $username, $password,
        $dbName);
  if  (!$conn) {
  die("Could not connect: " . mysql_error());
  }

        // Strings must be escaped to prevent SQL injection
            attack.
        $name = mysqli_real_escape_string($conn, $_GET['name'
            ]);
        $score = mysqli_real_escape_string($conn, $_GET['score
            ']);
        $hash = $_GET['hash'];

        $secretKey="3t96c5p3l61t9";
    # Change this value to match the value stored in the
        client javascript below

        $real_hash = md5($name . $score . $secretKey);
        if($real_hash == $hash) {
            // Send variables for the MySQL database class.
            $query = "insert into scores values (NULL, '$name
                ', '$score');";
            $result = mysqli_query($conn, $query) or die('
                Query failed: ' . mysql_error());
        }
?>
```

3. Edit the username[line 3] password[line 4] and databasename[line 5] within the quotation marks to match the username password and databasename we got for the database in part 1. You can also change the secretKey[line 18] just make sure to take a note of it as we'll need it in part 3.

   Save addscore.php.

4. Open display.php and add the following:

```
1  <?php
2    $servername = "localhost";
3    $username = "id6869038_username";
4    $password = "password";
5    $dbName = "id6869038_highscores";
6
7      // Send variables for the MySQL database class.
8      $conn = new mysqli($servername , $username, $password,
           $dbName);
9    if  (!$conn) {
10   die("Could␣not␣connect:␣" . mysql_error());
11   }
12
13     $query = "SELECT␣*␣FROM␣‘scores‘␣ORDER␣by␣‘score‘␣DESC␣
           LIMIT␣10";
14     $result = mysqli_query($conn ,$query) or die('Query␣
           failed:␣' . mysql_error());
15
16     $num_results = mysqli_num_rows($result);
17
18     for($i = 0; $i < $num_results; $i++)
19     {
20          $row = mysqli_fetch_array($result);
21          echo $row['name'] . "\t" . $row['score'] . "\n";
22     }
23 ?>
```

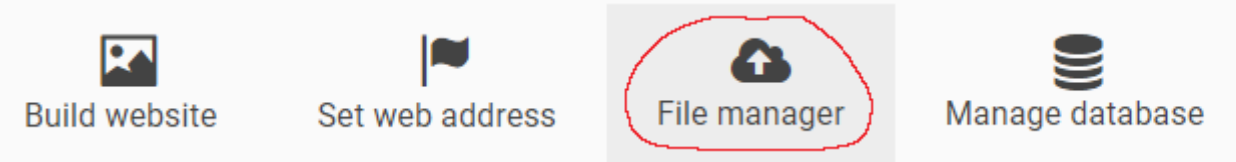
5. As with addscore.php edit the username[line 3] password[line 4] and databasename[line 5] within the quotation marks and save.

We are now ready to upload our php files to our website.

2.2 Uploading addscore.php and display.php to our website

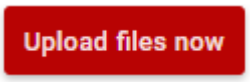
1. Log back into `https://www.000webhost.com/cpanel-login`.

2. Click on File manager as showen.



3. You should be able to see your websites url on this webpage take a note of it and add https:// to the front of it for completion, in the example below its https://documentary-ores.000webhostapp.com.

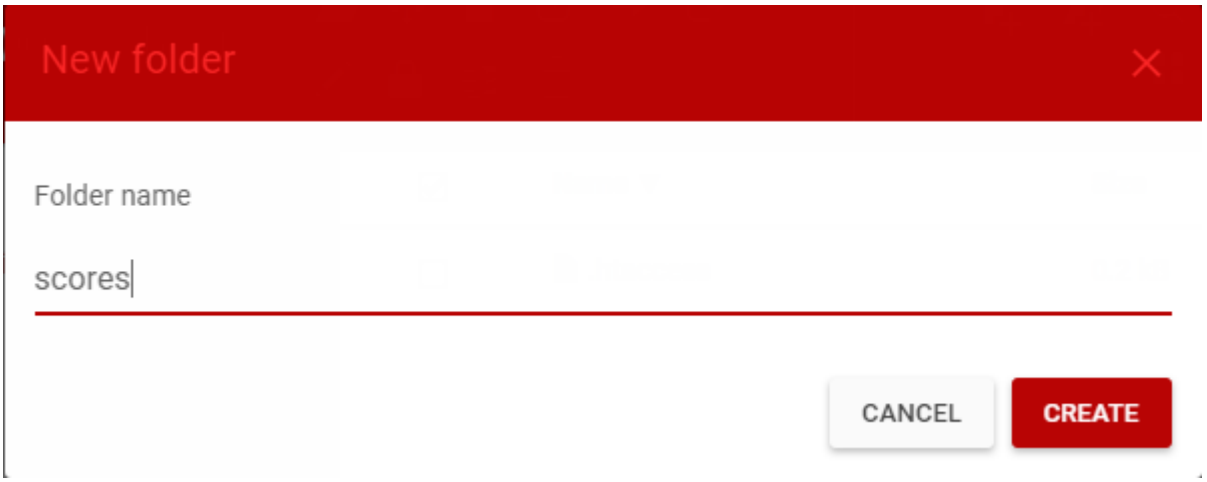
4. Click on Upload files now.



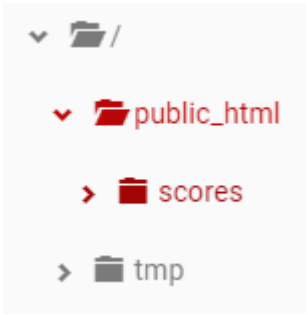
5. Make sure your in the public_html folder you'll be able to tell if it is as it'll be displayed in red and have a picture of a open folder next to it as shown below, you'll also be able to see it contents a file called ".htaccess".

6. Click on New Folder in the top right courner.

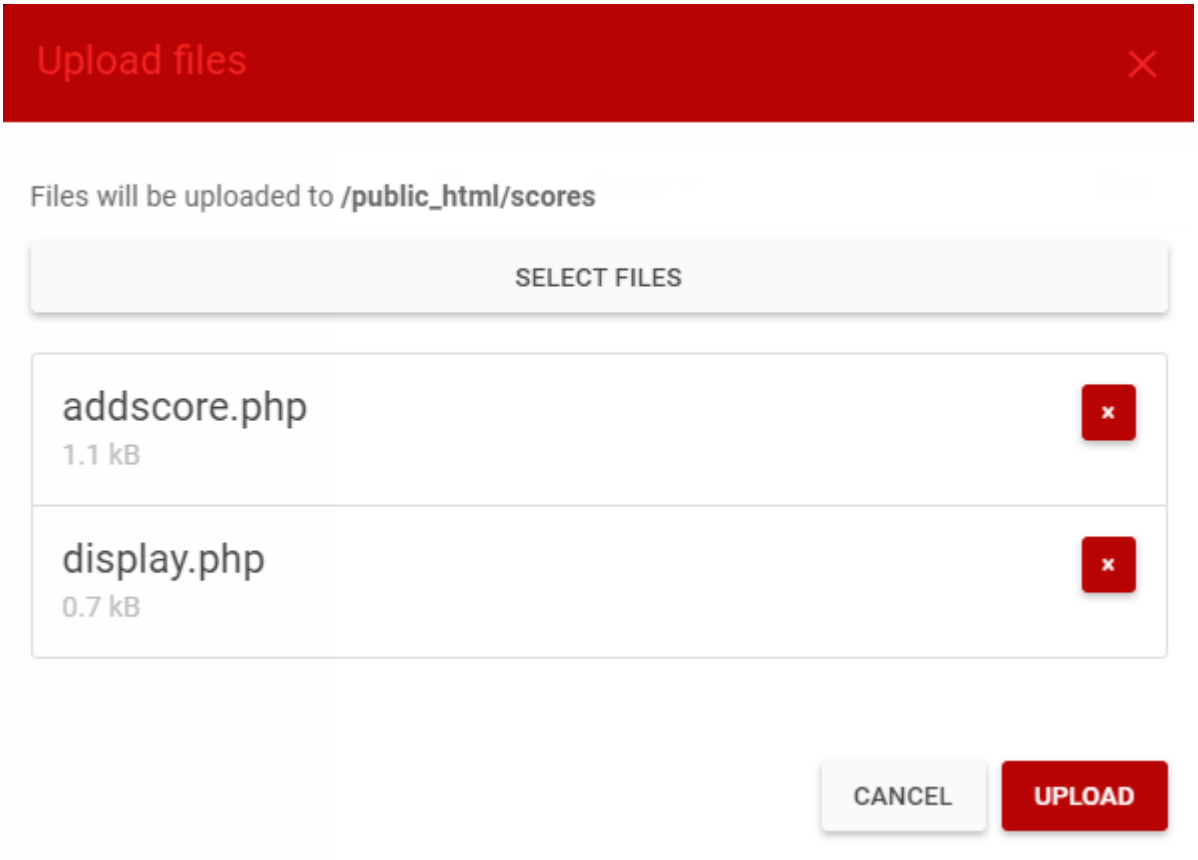
7. Type in the folder name "scores" and click CREATE.



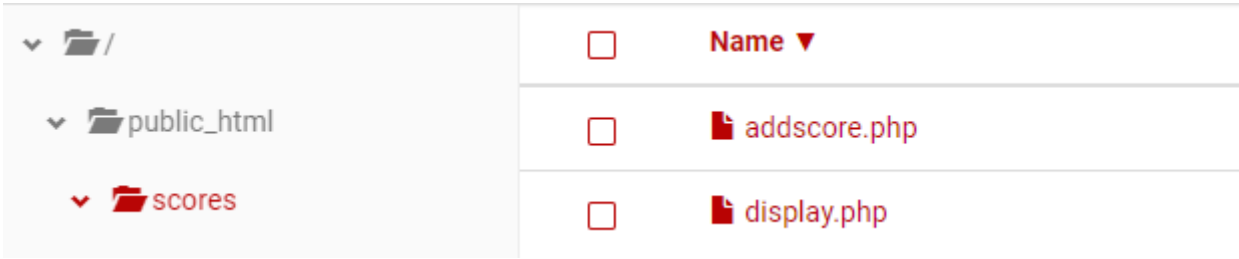
8. Click on scores on the left hand side of the screen to open it.



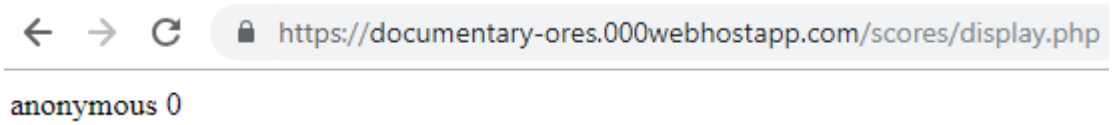
9. Drag with the cursor addscore.php and display.php we made earlier in this part from your pc to anywhere on the webpage to begin uploading them, ones this is done a box should popup as showen below, click UPLOAD.



10. Ones their uploaded they should appear in the scores folder as showen.



We can now access the top 10 scores from the database via our website. Type your websites url we noted earlier in this part with /scores/display.php added to the end if everything we've done so far has been sucessful you should see "anonymous 0" written in your browser as showen below, if not go back to 2.1 step 3 and 5 and make sure you typed your database's username, password and databasename correctly.



We are now ready to start setting up our game to send and retrieve scores from our online scores table.

3 Creating a sample game with a scoreboard and submission form with access to our online highscores table.

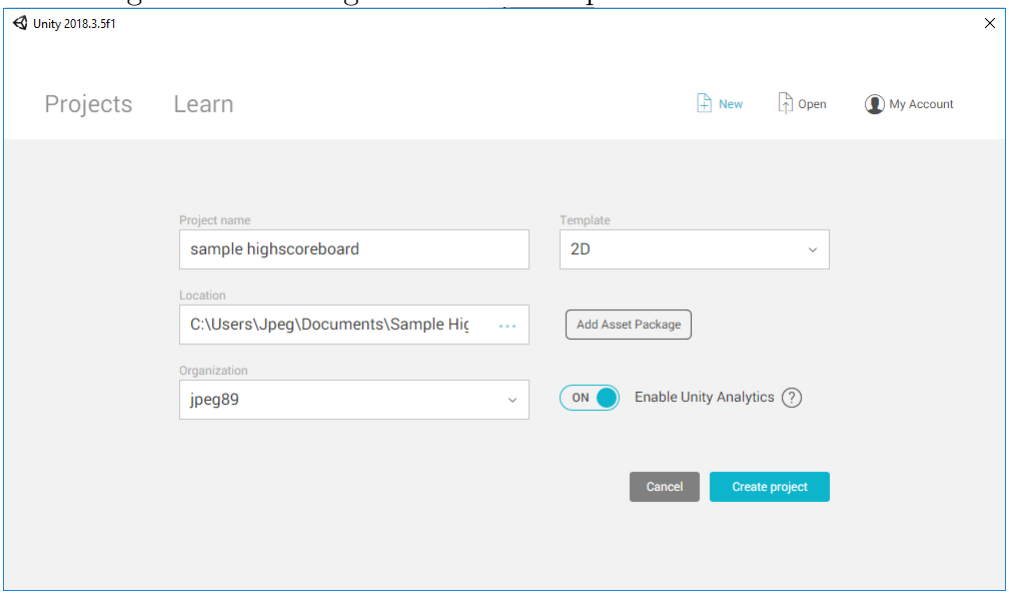
The purpose of this section is simple to illustrate how to setup a online scoreboard and submission board within unity it's going to be very bare bones for the sake of brevity/simplicity you'll probably wanna change and adapt allot of this for better asthetics for your own game but this should be a good starting point.

A small level of knowledge on how to use unity is going to be assumed as if your using this tutorial you've likely already made most of game.

3.1 Setting up our scene

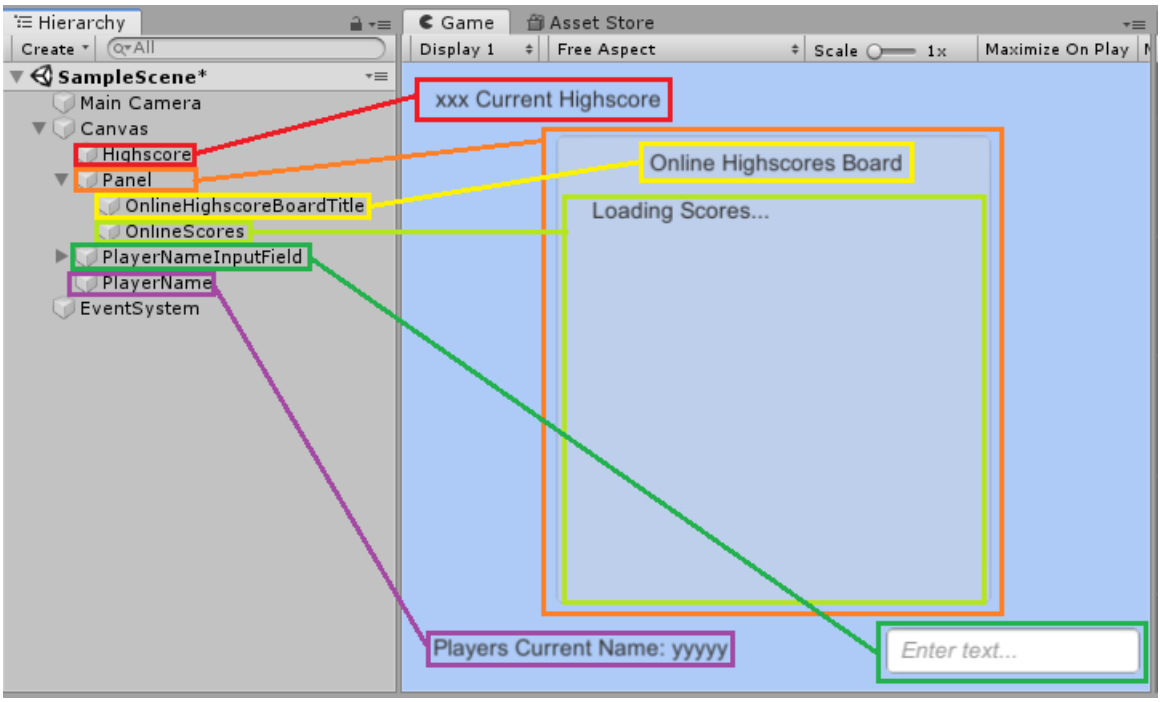
1. Assumeing you have it installed open unity, for this tutorial we're going to use version 2018.3.5f1 but any version should work.

2. Create a new unity project using the 2D Template we'll call the project something like "online highscoreboard template"



3. You can download a template scene with some gameobjects added in from here:
`https://drive.google.com/open?id=18kR1hWNWbhbCJt6ADToVXLXE9nFBtujw`
and add/open it in your project and skip to 3.2.
Alternatively you can set it up yourself. Create gameobject UI/Panel

with UI/Text objects OnlineHighscore and OnlineScores parented to it, then create two gameobjects UI/Text named Highscore and Playername parented to the same Canvas as the Panel fianlly create gameobject UI/Input field named PlayerNameInputfield also parent this to the canvas try and arranage the object in the scene so they look similar to the picture below.



3.2 Scripting our scene

We're going to create 2 empty gameobjects "DoNotDestroyScoreAndPlayernameTracker" and "ScoreboardManager" to contain the 3 script files we're also going to create and link up to the scene. You could create "DoNotDestroyScoreAndPlayernameTracker" on another scene in your game to carry over the variables of the players score and name to this scene as it'll be a GameObject that isn't destroyed between scene changes.

1. Create two empty gameobjects and name them "DoNotDestroyScoreAndPlayernameTracker" and "ScoreboardManager"

2. Add a script called "ScoreAndPlayername" to the gameobject "DoNotDestroyScoreAndPlayernameTracker".

3. Add the following c-sharp code to the script file we just created:

```
using UnityEngine;

public class ScoreAndPlayername : MonoBehaviour
{
    public int score; // the players current score
    public string playerName = "John"; // the player current
        given name

    void Start()
    {
        DontDestroyOnLoad(gameObject); // stops the game
            object being destroyed on scene changes.
        // There is no scene changes in this tutorial, the
            only reason we have this is so it could be
        // used to keep track of scores between scenes in a
            game.


        score = Random.Range(0,100); // randomly generates a
            score between 0 and 100 to represent the
        // the players score in your game
    }
}
```

4. Remember to save it - I've tried to comment the code the best I can so it's understandable basically this code is just a stand in for your game and genrates a random score for the fictitious player John.

5. Add a script to "ScoreboardManager" and call it "UIController", add the following code to the script file:

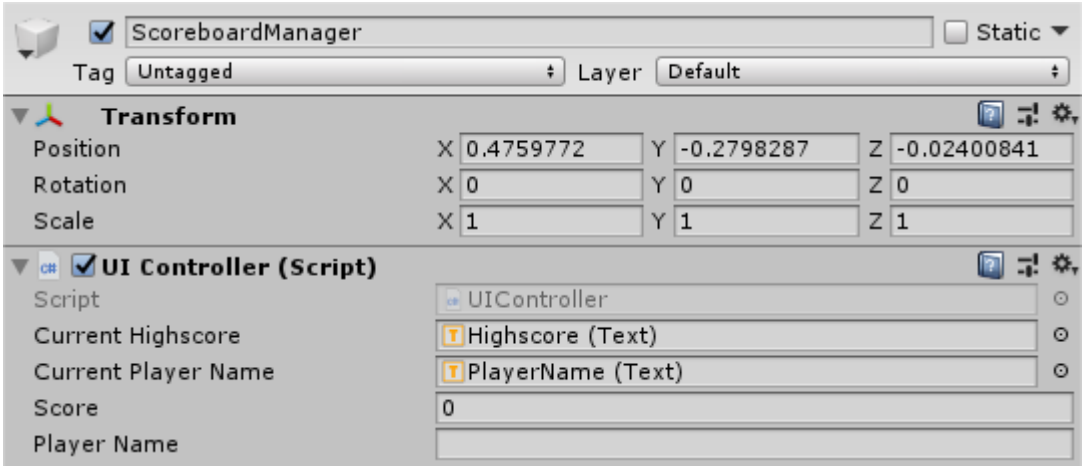
```
using UnityEngine;
using UnityEngine.UI;

public class UIController : MonoBehaviour
{
    public Text currentHighscore; //Reference "xxx␣Current␣
        Highscore" text component of that GameObject in our
        scene
    public Text currentPlayerName; //Reference "yyyyy␣Current
        ␣Name" text component of that GameObject in our scene
```

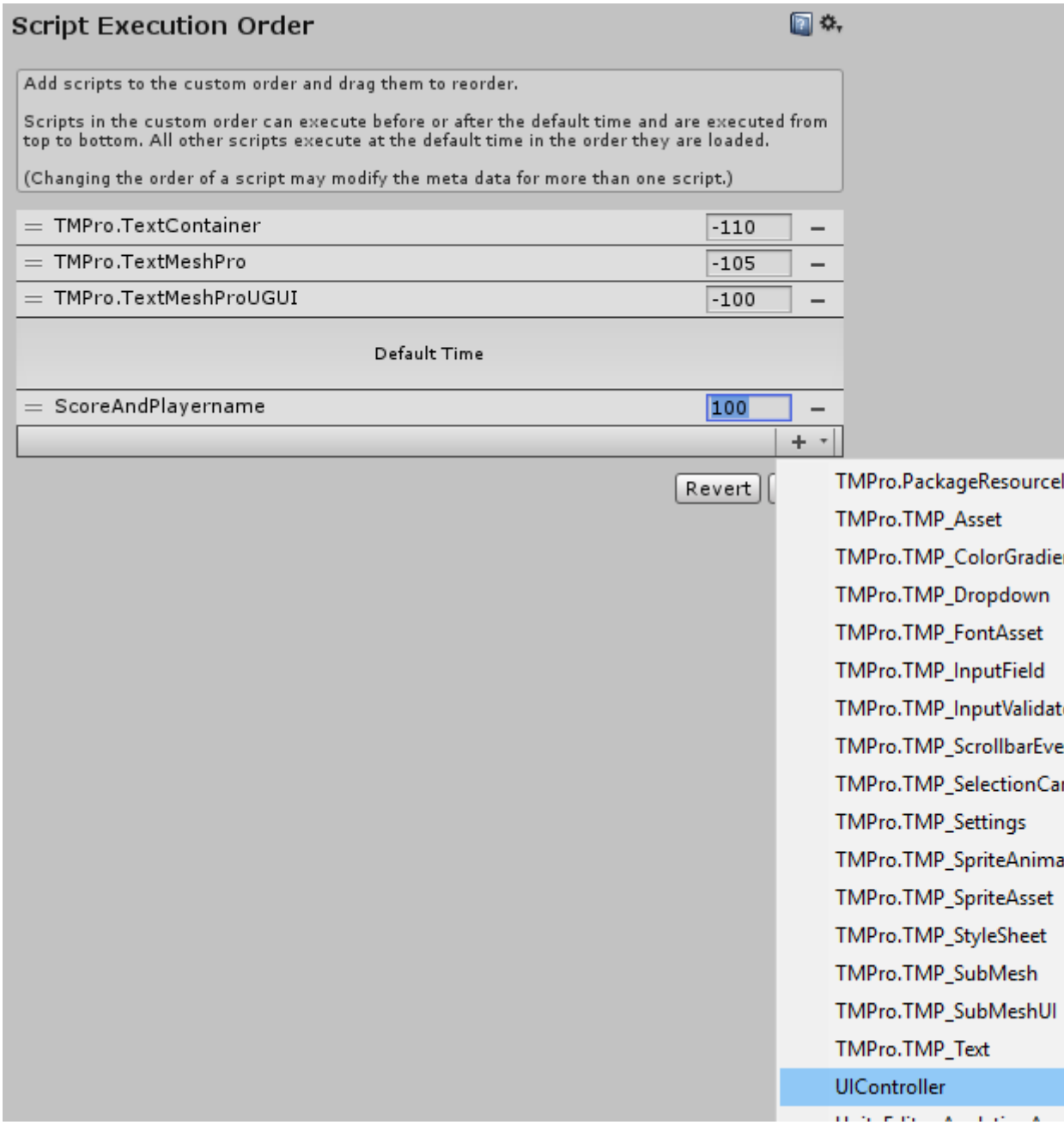
```
8     public int score;
9     public string playerName;
10
11    void Start()
12    {
13        playerName = GameObject.Find("
              DoNotDestroyScoreAndPlayernameTracker").
              GetComponent<ScoreAndPlayername>().playerName;
14        score = GameObject.Find("
              DoNotDestroyScoreAndPlayernameTracker").
              GetComponent<ScoreAndPlayername>().score;
15        // fetches the value for score and playerName from our
               DoNotDestroyObject GameObject's
              ScoreAndPlayername script
16
17
18        currentHighscore.text = score.ToString() + "␣Current␣
              Highscore";
19        // prints the score on the top left of the screen
20
21        currentPlayerName.text = "Player's␣Current␣Name:␣" +
              playerName;
22        // prints the name on the bottom left of the screen
23    }
24
25    public void OnUsernameEdit()
26    {
27        playerName = GameObject.Find("PlayerNameInputField").
              GetComponent<InputField>().text;
28        currentPlayerName.text = "Player's␣Current␣Name:␣" +
              playerName;
29        // updates the local playerName to match the userinput
               username and prints it on the bottom left of the
              screen
30
31        GameObject.Find("DoNotDestroyScoreAndPlayernameTracker
              ").GetComponent<ScoreAndPlayername>().playerName =
               playerName;
32        // updates playerName on the ScoreAndPlayername script
               not nessasairy for the tutorial but if we don't
              it it could
33        // cause problems if you change to a different scene
              and back to this one in your own game
```

```
34      }
35  }
```

6. Save the script - the code above is responsible for updateing any changes to players username and score made on this scene and displaying them in the scene it does this by replaceing the text in the "Highscore" and "PlayerName" gameobjects because of this we'll need to drag the "Highscore" and "PlayerName" objects into the vacant slots next to "Current Highscore" and "Current Player Name" on the new "UI Controller (Script)" component of the "ScoreboardManager" Object in the inspector tab in unity as showen below. At the end we'll also link this script to the PlayerNameInputField.



7. Another thing we'll need to do is fix a bug that will now be occuring due to the order in which unity executes the two script files we just made, as we want the code in the void Start() function of the UIController to execute after the code in the void Start() function of ScoreAndPlayername because one refrences the other to do this we go in unity Edit/ Project Settings.../ Code Execution Order and add "ScoreAndPlayername" and "UIController" by clicking on the "+" as showen below. You may not need to do this in your own game if the "ScoreAndPlayername" script is intialised on a previous scene but it's important for this example.



8. Set "ScoreAndPlayername" to 100 and "UIController" to 200 this will ensure that "ScoreAndPlayername" is always executed first. Make sure to click apply.

9. Now to add our last script create a script called "SubmitAndLoadScores" and add it to the "ScoreboardManager" object, this is the

script that's going to bridge the gap between our game and the online scoreboard we setup in the previous part.

10. Add the followng code to the script and edit the secretKey, addScoreURL and highScoreslistURL strings within the quotation marks[line 8, 9 and 10] to match the php URLs and secret key we setup in the previous part be sure to add the ? at the end of the addscore URL note - you'll only need to change the secretKey if you changed it in the previous part:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class SubmitAndLoadScores : MonoBehaviour
7  {
8      private string secretKey = "3t96c5p3l61t9"; // Edit this value
           and make sure it's the same as the one stored on addscore.
           php
9      string addScoreURL = "https://documentary-ores.000webhostapp.
           com/scores/addscore.php?"; //be sure to add a ? to your url
10     string highScoreslistURL = "https://documentary-ores.000
           webhostapp.com/scores/display.php";
11
12     public Text displayScores; //Reference "Loading␣Scores..." text
            component of that GameObject in our scene
13
14     int highScore;
15     string playerUsername;
16
17     void Start()
18     {
19         LoadScoresAndNamesFromDatabase(); //loads scoreboard and
               displays it ingame
20     }
21
22
23     public string Md5Sum(string strToEncrypt)
24     {
25         System.Text.UTF8Encoding ue = new System.Text.UTF8Encoding
               ();
26         byte[] bytes = ue.GetBytes(strToEncrypt);
```

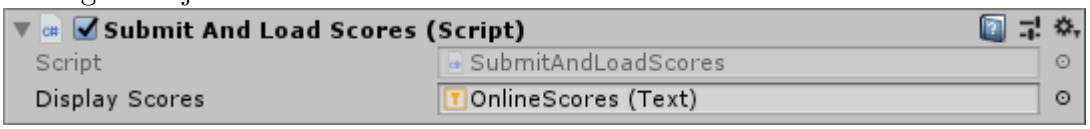
```
27
28          // encrypt bytes
29          System.Security.Cryptography.MD5CryptoServiceProvider md5 =
                new System.Security.Cryptography.
                MD5CryptoServiceProvider();
30          byte[] hashBytes = md5.ComputeHash(bytes);
31
32          // Convert the encrypted bytes back to a string (base 16)
33          string hashString = "";
34
35          for (int i = 0; i < hashBytes.Length; i++)
36          {
37              hashString += System.Convert.ToString(hashBytes[i], 16)
                    .PadLeft(2, '0');
38          }
39
40          return hashString.PadLeft(32, '0');
41      } // Hashes/encrypts the username and score using the secretKey
            for transmission over the web to
42        // our php file https://documentary-ores.000webhostapp.com/
            scores/addscore.php
43
44
45      public void LoadScoresAndNamesFromDatabase()
46      {
47          StartCoroutine(FetchScoresAndNamesFromDatabase());
48      }
49      // Fetches the scores and names from the Database
50
51
52      public void SubmitScoreAndNameToOnlineDatabase() //loads when
            user pushes enter(On End Edit (String)) after typing in
            name
53      {
54          highScore = GameObject.Find("ScoreboardManager").
                GetComponent<UIController>().score; // fetches the
                score
55          playerUsername = GameObject.Find("ScoreboardManager").
                GetComponent<UIController>().playerName; // fetches the
                 playername
56          StartCoroutine(PostScores(playerUsername, highScore)); //
                post them online
57      }
```

```
58      // Post current score and name to the Database
59
60
61      IEnumerator PostScores(string name, int score) // used to post
            scores online
62      {
63          // This connects to a server side php script that will add
                the name and score to a MySQL DB.
64          // Supply it with a string representing the players name
                and the players score.
65          string hash = Md5Sum(name + score + secretKey);
66
67
68          string post_url = addScoreURL + "name=" + WWW.EscapeURL(
                name) + "&score=" + score + "&hash=" + hash;
69          Debug.Log(post_url);
70
71          // Post the URL to the site and create a download object to
                 get the result.
72          WWW hs_post = new WWW(post_url);
73          yield return hs_post; // Wait until the download is done
74
75          if (hs_post.error != null)
76          {
77              Debug.Log("There␣was␣an␣error␣posting␣the␣high␣score:␣"
                     + hs_post.error);
78          }
79          else
80          {
81              Debug.Log("possible␣score␣upload␣sucess,␣reloading␣
                    scoreboard");
82              StartCoroutine(FetchScoresAndNamesFromDatabase()); //
                    If score upload was successful update local
                    highscoreboard so the user can see
83          }
84      }
85
86      // Get the scores from the MySQL DB to display in a GUIText.
87      IEnumerator FetchScoresAndNamesFromDatabase()
88      {
89          WWW hs_get = new WWW(highScoreslistURL);
90          yield return hs_get;
91
```

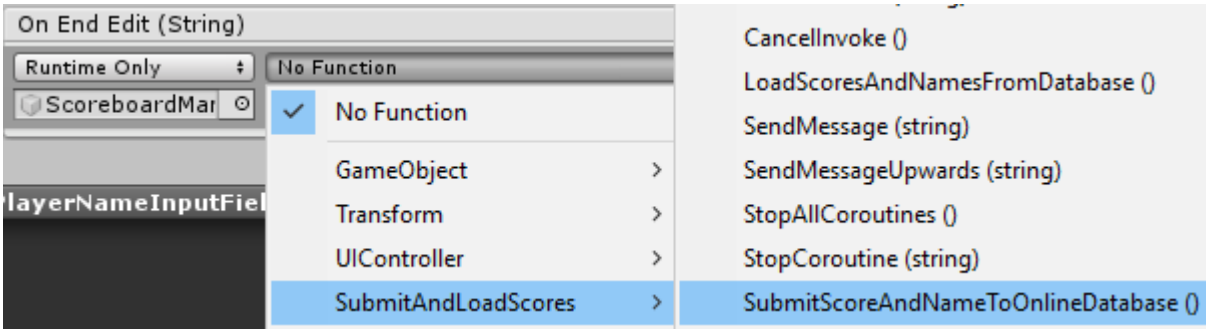
```
92          if (hs_get.error != null)
93          {
94              Debug.Log("There was an error getting the high score
                    names: " + hs_get.error);
95          }
96          else
97          {
98              displayScores.text = hs_get.text; // this is a GUIText
                    that will display the scores in game.
99          }
100     }
101
102 }
```

11. And save the script. We're almost done we just have to link a few things in our scene and we should have a working scoreboard.
note - you can ignore the yellow warning this script causes in the console it should still all work as intended

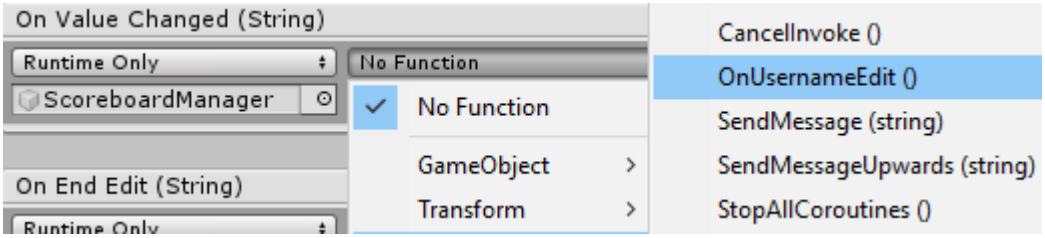
12. First we need to drag the "OnlineScores" object the one parented to the Panal object into the vacant "Display Scores" Slot on our "Submit And Load Scores (Script)" component attached to the "Scoreboard-Manager" object as showen below.



13. Now we need to add functionality to our PlayerNameInputField by linking it to some functions we setup earlier in our script files, to do this, select the PlayerNameInputField object in unity and scroll down on the inspector to "On End Edit (String)" click the plus sign then drag the "ScoreboardManager" to the "None (Object)" Slot that appeared, now click on the pull down menu that currently reads "No Function" and click on SubmitAndLoadScores/SubmitScoreAndNameToOnline-Database() as showen below.



14. Above "On End Edit (String)" where is says "On Value Changed (String)" click the + button drag the "ScoreboardManager" to the "None (Object)" Slot that appeared like before, and click on the pull down menu that currently reads "No Function" and click on UIController/OnUsernameEdit() as showen below.



That's us completed finnished hopefully when you run the game you should be able to submit scores by typing a name into the PlayerNameInputField and pushing enter this should alos cause the scoreboard on screen to update. Their a link to all the things created in this tutorial here for if your still having problems.
`https://drive.google.com/open?id=1gksHlNDfUlQZDycnfSdD8cQTYBCyveOJ`

It's worth being aware if you wish to host your unity game online with a WebGl build the scoreboard won't work unless the games hosted on a website that supports https:// protcols I'm not sure why this is but presumable it's a security feature of some of the code in "SubmitAndLoadScores" script file we made.