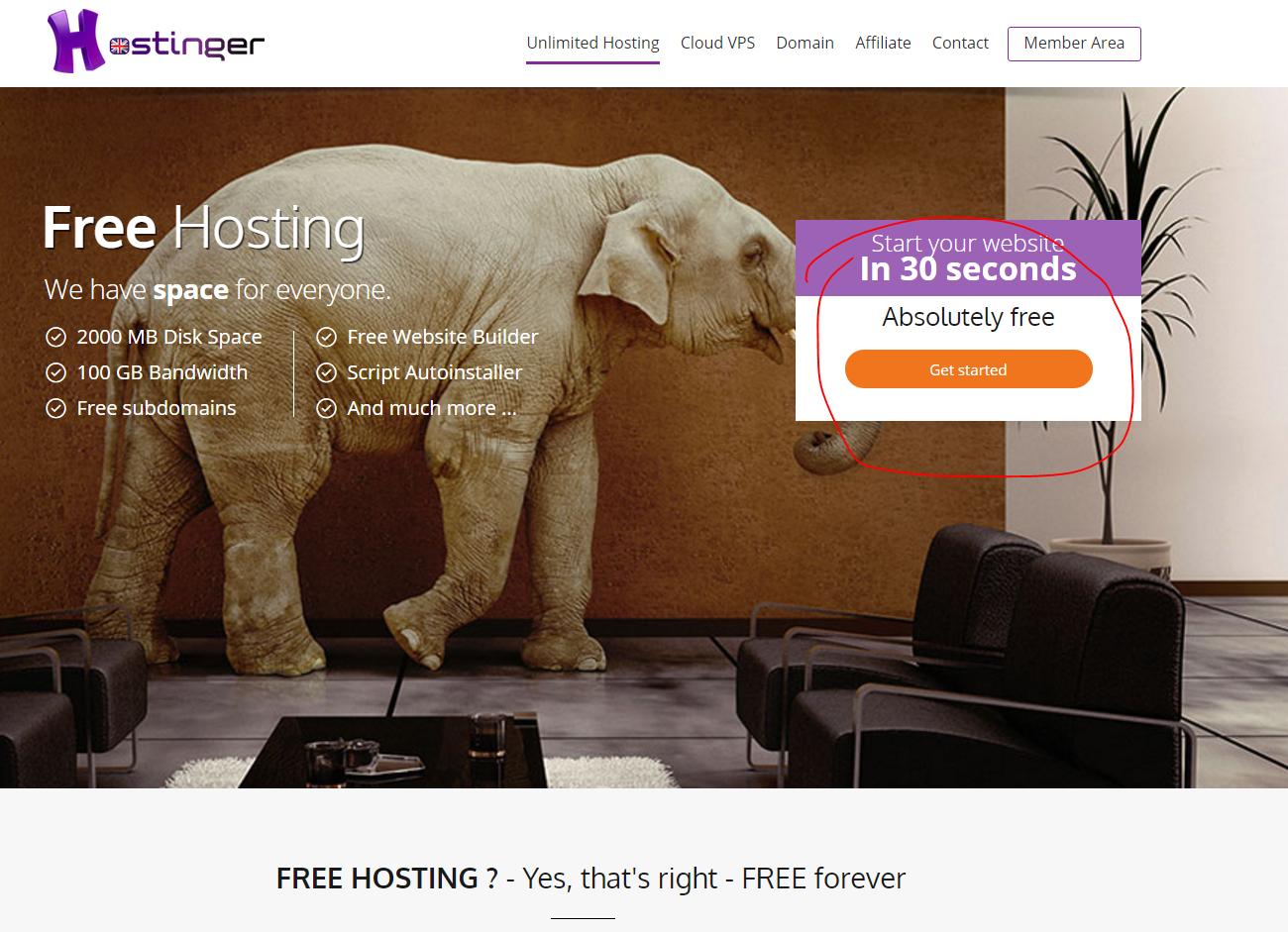
# Database Setup

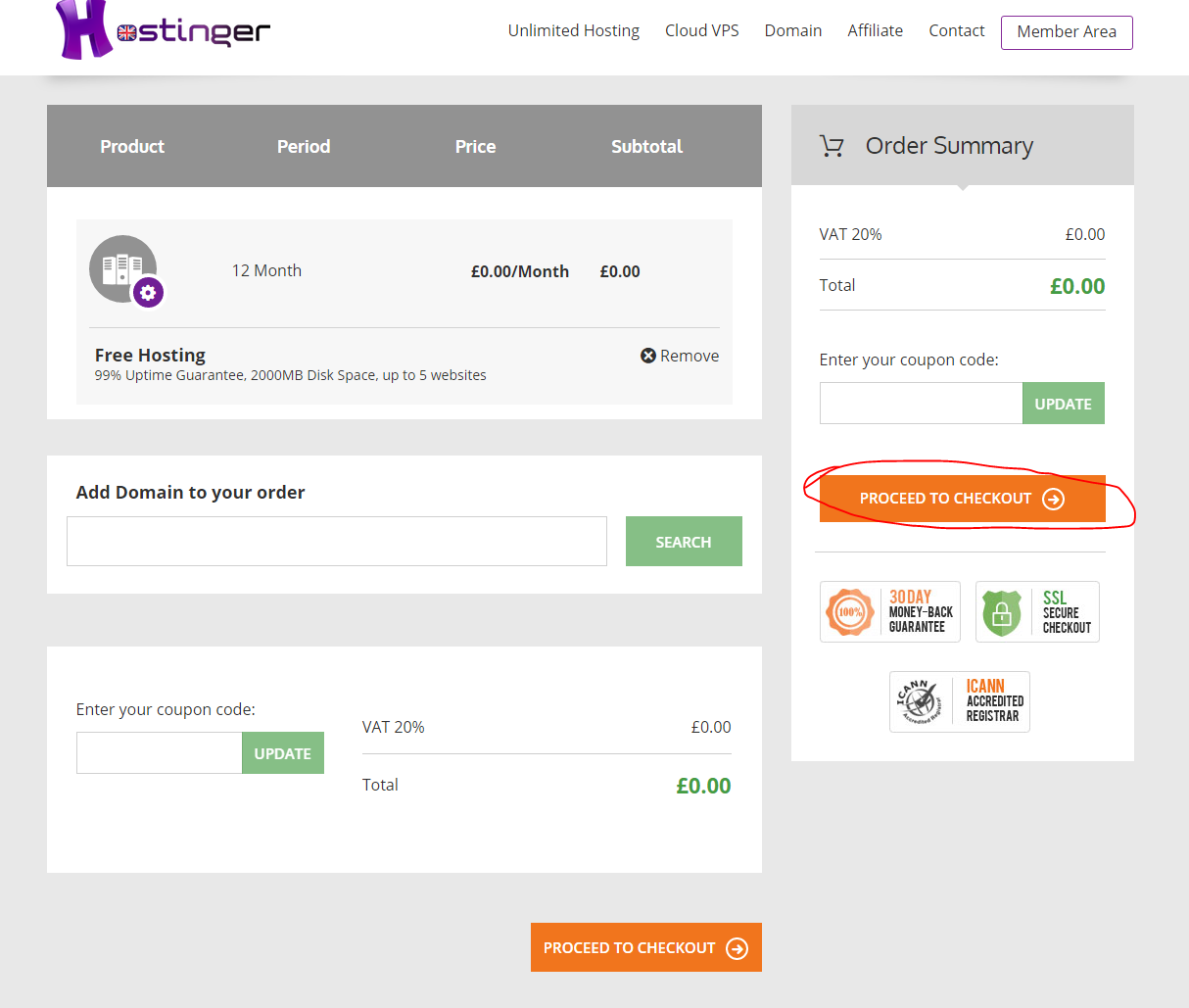
## Initial setup

Go to http://www.hostinger.co.uk/free-hosting

And click the “Get Started” button



Click “proceed to check out”

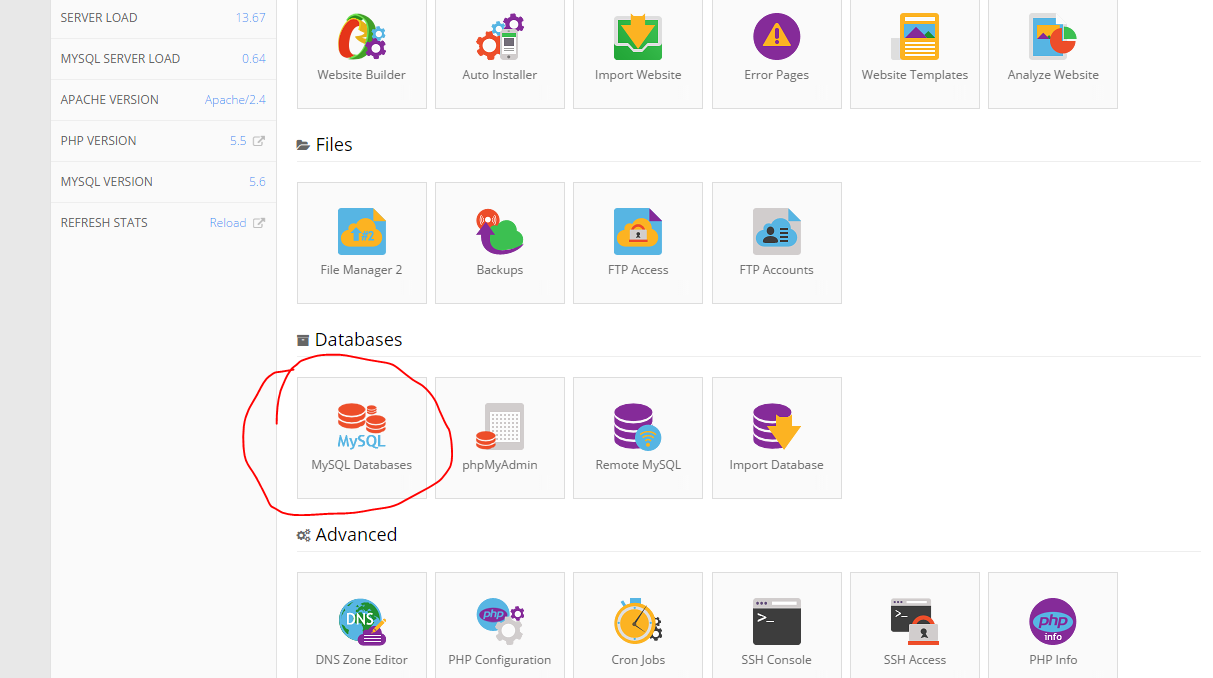


Then Sign up for an account, you can use your uni email if you want.

After signing up, check your inbox for an email asking you to confirm your account, this may take a few minuets

After confirming your account, log in.

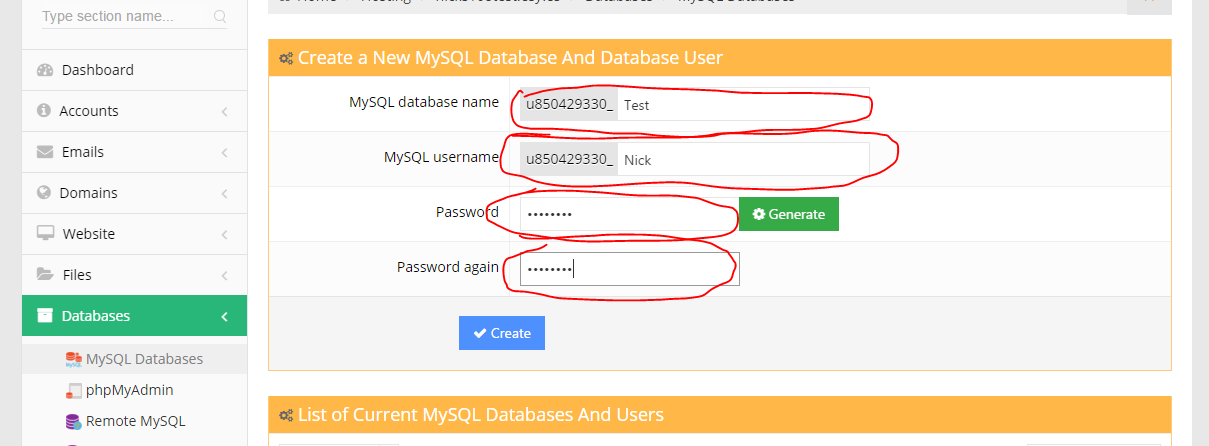
On the main page, scroll down until you see a “MySQL Database” button. Click it



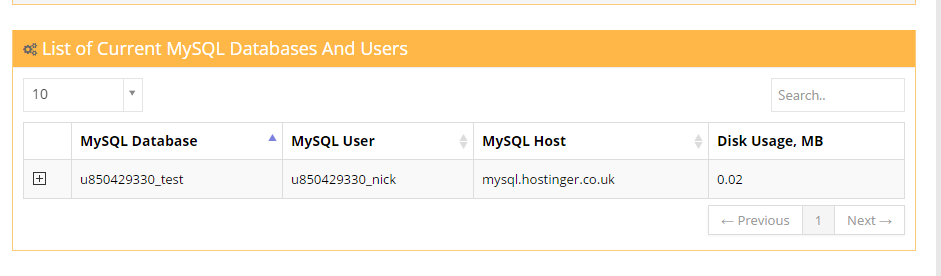
On the next screen, enter a name for your database,

Enter a user name

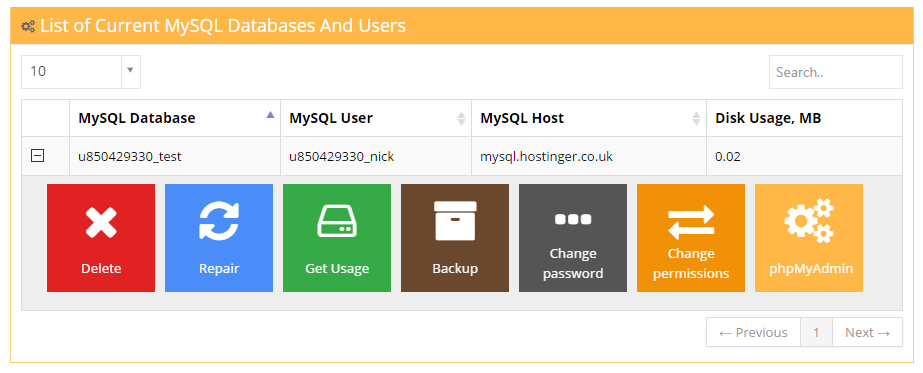
And a password



Hit the “Create” button and wait 20 seconds.



Click on the database and expend the panel

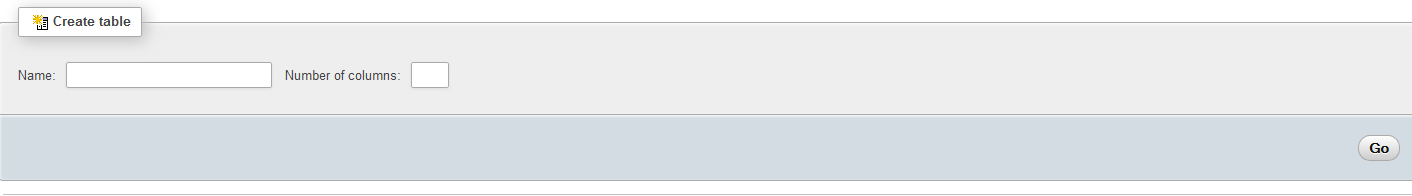


And click on the “phpMyAdmin” button

## The Database

Now we need to create some tables to hold our data.

On the new screen, enter “Users” in to the “name” text box and set the number of columns to 3 and click “go”.



On the next screen you will need to fill in the data for 3 columns

Column 1

Name = ID

Type = int

Index = PRIMARY

A\_I = true (auto incrementing)

Column 2

Name = EmailAddress

Type = Varchar

Length = 200

Column 3

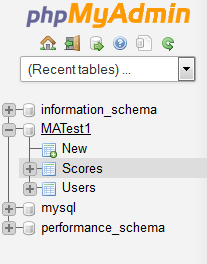
Name = Password

Type = varchar

Length = 200

And press the “save” button

New create another table (expand the tree on the left and under you database, click “new”)



Now fill in the 4 columns with:

Column 1

Name = ID

Type = int

Index = PRIMARY

A\_I = true (auto incrementing)

Column 2

Name = UserID

Type = int

Column 3

Name = Score

Type = int

Column 4

Name = DateOfScore

Type = dateTime

So now we have 2 tables. One is for holding our users data and the other is for holding scores for that user.

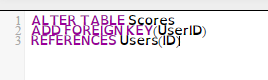
But they are not yet related to each other. As I’m sure you can remember from intro to databases, data tables need to be linked in order to ensure you maintain the accuracy of the data within the database.

So let’s do that. We are going to link the users table to the scores table (notice how I have used plurals for my table names, that because they hold more than one) in a one to many relation

Scores

Users

Use the following SQL to link you tables



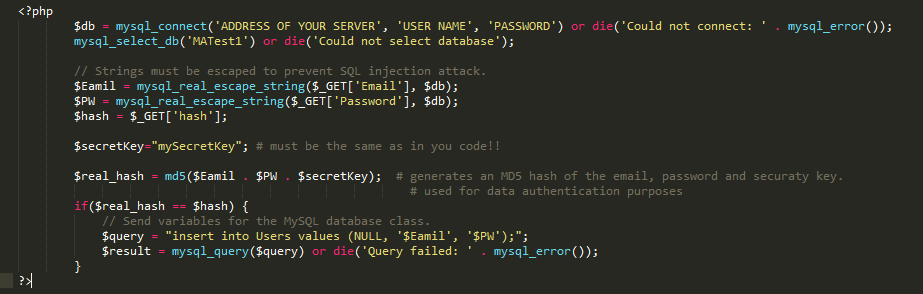
PHP pages

Ok, now that we have our database set up and are ready to go, we need to construct our php pages to read and write data between our games and our database.

Let’s start with writing to the users table

Open your text editor of choice and make a new file (BE SURE TO SAVE IT AS A .php FILE!!! Something like AddUser.php).

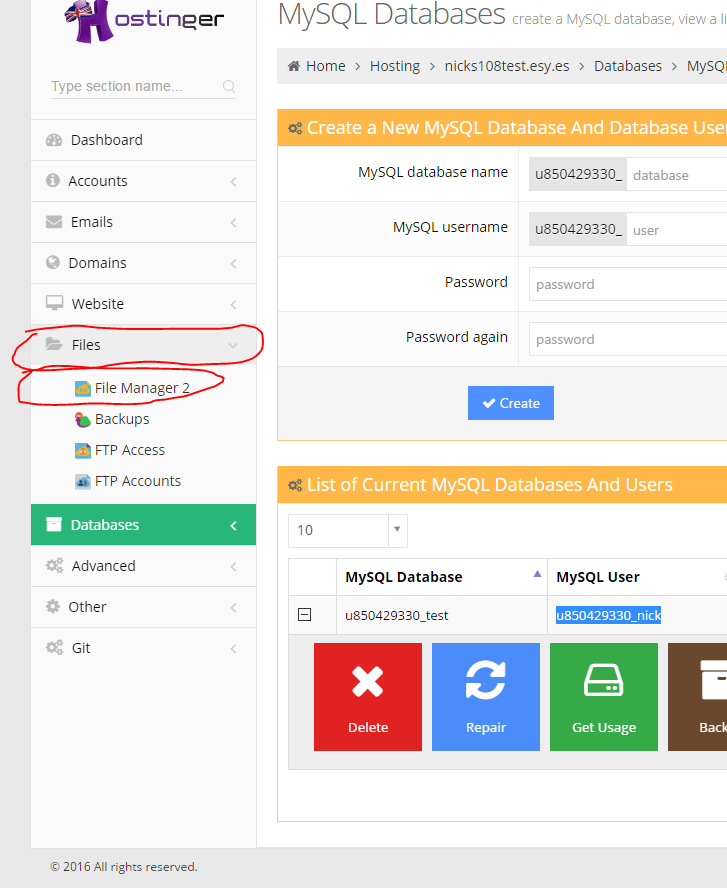
Write the following code in to the file, make sure you change the server address to the local host address, user name and password to the user name and password you use to log on the phpmyadmin, at the top.



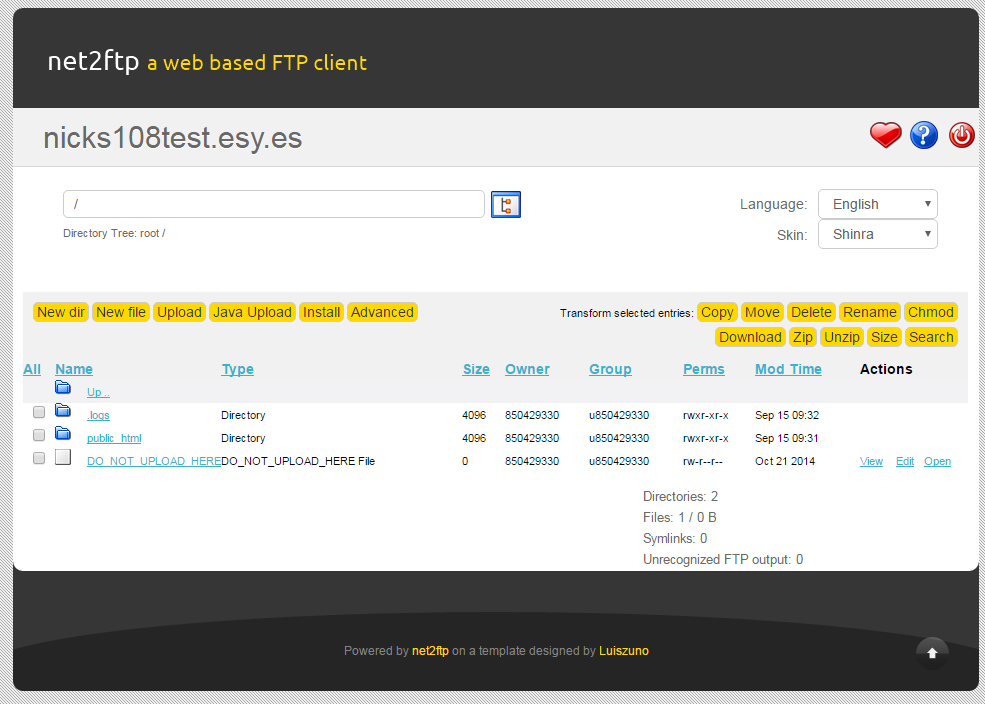
For test purposes, we will add ||true to the hash comparison if statement (just so that we don’t need to write any code to generate a hash, we’ll remove this in a min)

And save the file.

Now you need to upload the file to the server. Go back to the server and open the “Files” tab on the left panel then click “File Manager 2”.



A new tab will open and in it you will see the following



Click on the “public\_html” directory

Now click the “upload” button and select you file to upload, press the green tick when you are done.

Now let’s test the php file.

Open up a browser and type

The address of you server/the folder you just made/the name of you php file

It should look something like: http://nicks108test.esy.es/AddUser.php

At the end of this add ?Email=test Eamil &Password=pwtest &hash=qwer

Did it work?

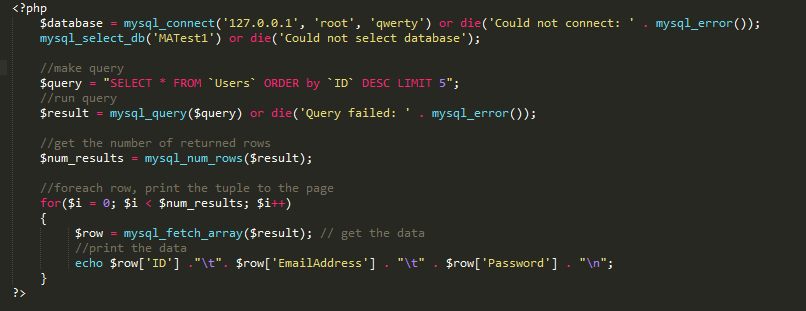
Open up the table in phpMyAdmin, navigate to the database and table and find out

Now go back to the php file and remove the ||true from the if statement. Save and upload the newly edited file to the server.

Now we’ll write a php file to read from the database.

Open a new text file and save it as “selectAllUsers.php”

Write the following in to the file



Now save you php file and drop it on to the server

Now, in your web browser, navigate to the new page (change the address to point to the new php page).

You should see a number (the ID of the user),”test Email” and “pwtest” (assuming you used the same test data I did.

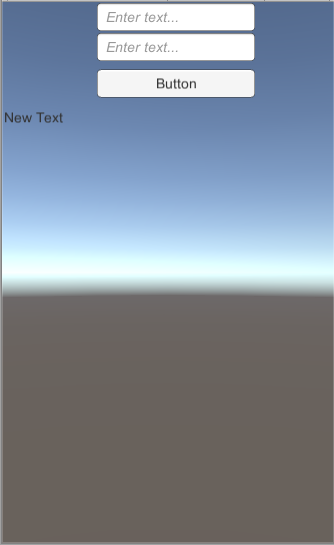
Now we need a game to continue development.

Now add the following to the top of the php files, under the “<?php” tag

error\_reporting(E\_ALL &~ E\_DEPRECATED);

## Simple game

Open a new unity project and create a new UI button, an input field and a text box. Position them so you can see the all clearly.

Now you need to create a new script called for accessing your php pages and reading the data.

Attach this script to the button

Now edit the script and delete the functions inside.

Create a private variable to hold you secretKey.

private string secretKey = "mySecretKey";

Now create a variable to hold a link to you add user URL

publicstring AddUserURL =

“http://nicks108test.esy.es/AddUser.php?”

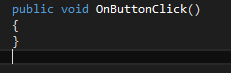
(Don’t forget the “?” at the end)

Now create a new variable to hold the link to your show all users page.

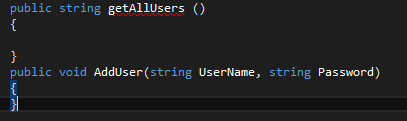
publicstring ShowAllUsersURl =

“http://nicks108test.esy.es/ShowAllUsers.php?”

Now create a new public function that will handle our button clicked event



Now create 2 more functions that will handle our database read and write operations.



Let’s start by getting all of the users

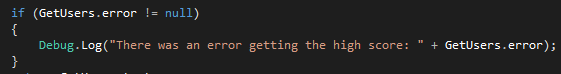
We need to contact you database via our ShowAllUsersURL. We do this with the WWW class and pass in our ShowAllUsersURL.



This will start a new thread in unity that will go to the URL we have specified and load the page in memory. But we need to wait for this thread to finish or we won’t be able to process the data, because we won’t have any to process, we’re going to make the program do nothing until the page has finished loading.



Once GetUsers has finished, we what to check it for errors and if we fins one, print the error message to the console.



And finally, we need to return the text that GetUsers loaded from the webpage.



Let’s wire this up so that on button press, we read data from the database and add it to the text UI element.

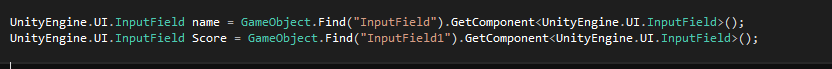
In the OnButtonClick function, get a reference to your text UI element and set its text attribute to the output of GetAllUsers.



Don’t forget to link the buttons onClick event to the OnButtonClick function. (This will break. Time for you to figure out why). Now run your code.

Now that you have fixed the issue, it’s time to write some data to the database.

Find and hold a reference for each of the input fields.

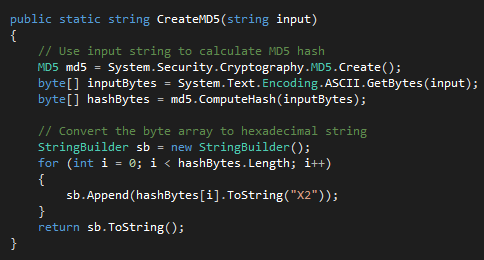


Now call AddUser in the OnButtonClick function and pass it 

The second field is blank because I’m ignoring the password for now.

Now we need to fill in our AddUser function.

First thing we need to do is create a hash so we can check the validity of the information we are passing to the database. To do this we need to include System.Security.Cryptography and System.Text and use the following function



This isn’t a cryptography lesson so don’t worry too much about what’s happening here.

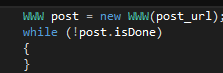
Now in our AddUsers function we can create a hash of the username, password and the secretKey



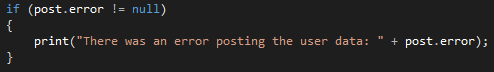
The next thing we need to do is construct the post URL



Then we need to call the WWW class and wait for it to finish posting the data to the server.



And the final thing is to catch any error that may occur.



That’s it.

Now run you code, type in a name in to the first input field and press the button. Your name should be added to the list in the text element.

Ok, now you need to work out a way to store the score (use the second input field for this) and link it with the players name

Do you think there will be an issue is 2 people called Nick entered a score? How are you going to fix that?

Now make a simple game, something like cookie clicker and once the player is board, let them submit their name and score to the database.