Technical Manual: Cybersecurity Training 2.0

By: Robert Hand (Team Leader), Valeria Gamboa, Matthew Sexton, Brandy Jordan

Table Of Contents

[**Introduction**](#_wbleqqmdj5jk) **3**

[**System Components**](#_mujwk4na665h) **3**

[**Software**](#_5i0opjku24ka) **3**

[Development](#_ushfjeidr6dp) 3

[Deployment](#_oiow742gbzyb) 4

[**Github**](#_km7ysubeioky) **4**

[**XAMPP Installation**](#_sl2kkrlscg85) **4**

[**Netbeans Setup**](#_gl2vjtg78bhq) **4**

[**Database Setup**](#_5fmfmy2i92jf) **4**

[**Certificate Feature Config. Instructions**](#_a9qf839d99qy) **5**

[Folder to hold certificates](#_ckk3271zpohs) 5

[PHPMailer config instructions](#_yf98mq7i4b39) 5

[XAMPP config instructions](#_y1mkr3xaou6g) 5

[PDF Generation config information](#_qgknt7qk643l) 6

[Email server credentials](#_8nnv0vh41gra) 7

[Important Note](#_6v36mcv7ibsc) 7

[**Unity**](#_bphconmce7e5) **7**

[**Errors/Possible Problems**](#_qgyg8toziw2d) **7**

# Introduction

While the system is rather simple from the standpoint of a user, installing and working on the system is an entirely separate matter. This manual will start with a general overview of the technology used in the system. A significant portion of this manual will go into describing how to set up this project so that it can be worked on from another computer. After that will come the specific technical descriptions for the project.

# System Components

The system will follow a typical client server architecture. The client-side of the system will run in a user’s browser. The server side will consist of three layers. The first layer will be a web server that houses the content for the client-side of the system. It will have the webpage content as well as most of the logic for the video games that will be played when using the system. The second layer “below” the first will consist of the business layer logic, it will update the system based on a user’s actions. The third layer of the server will have a database that stores user information, as well as other information useful to the system.

# Software

## Development

The development of this system will have many components. The server will be developed using the XAMPP web server solution stack package. The web server will be an Apache server. The business layer logic of the server will be written in PHP, the PHP code will use the MySQLi relational database driver in its interactions with the database. The database will be a MariaDB database, and it will be managed using phpMyAdmin.

The client side of the system will be a website written in HTML and CSS, with JavaScript to handle any computations that the client needs to do. There will be a WebGL game implemented in the client as well. The game will be developed in the Unity game engine, which cross-compiles the game into WebGL javascript. Other than Unity, the rest of the system will be developed in the NetBeans IDE and use GitHub for version control.

## Deployment

The deployment of the system will be a website that will be hosted by a server. This server will most likely use the Linux operating system.

# Github

The github for this project is Phase-4, it is public and hosted by rjh36. It contains everything, the up to date documentation of the system, configuration scripts, test questions and answers, and all of the source code.

# XAMPP Installation

To be able to run this project someone will need to download and install XAMPP. The absolute minimum requirements for this project are the attributes Apache, MySQL, PHP, and phpMyAdmin. A best practice would be to install xampp in the “C:” directory. Starting the xampp-control.exe file as an administrator and setting up the XAMPP control panel would be a best practice.

# Netbeans Setup

1. If you haven’t already, download netbeans version 8.2 (or up).
2. Open Netbeans and create a new project, select the option to make a PHP application with existing sources.
3. For the sources folder, add the github repository folder that contains the sources, in this case “Sources”. Name the project whatever you want.
4. Select copy files from source folder to another destination, then set the destination as the folder in htdocs that matches up with the project URL (the url can be whatever you like too). Lastly, select copy files on project open.

# Database Setup

The database is setup using three scripts that are then imported into phpmyadmin. The scripts are in the config folder in the github directory. The steps to take are:

1. Open up phpMyAdmin on localhost.
2. **The next three steps must be completed in the proper order.(These files are located in the sources folder**
3. The first file to import into the database is install\_database.sql
4. The second file to import into the database is Q\_and\_A\_example.sql
5. The third file to import into the database is final\_Q\_and\_A.sql
6. After that, the database will be properly set up

# Certificate Feature Config. Instructions

The configuration to get the certificate feature working is quite complicated.

## Folder to hold certificates

In the folder for the project that resides in the htdocs folder of XAMPP (the project URL, or htdocs\ProjectName\), create a folder named “Certificates”. This is the directory in which the generated certificates will be stored.

## PHPMailer config instructions

PHPMailer is an open source library built to make mailing functions in PHP easier. The PHP function mail() is the only php support for email. PHPMailer adds support for attachments and generally just makes implementing email features in a system not be an absolute nightmare. However, it can still be rather tricky to get working properly.

1. Firstly, composer - a dependency manager for PHP - needs to be installed and integrated into XAMPP in order to make the management of PHPMailer easier.
2. [how to install PHPMailer on Windows and Linux](https://alexwebdevelop.com/phpmailer-tutorial/#how-to-install-phpmailer) Here is the tutorial I used to install and configure PHPMailer. For installation purposes, just go to the section labeled “Installing Composer and PHPMailer on Windows (if you use XAMPP, WAMP etc.)” that resides under the link and follow those instructions.

## XAMPP config instructions

The next set of steps involves messing around with the files managing XAMPP, in particular some of the .ini files. This will let PHPMailer functions work with XAMPP and smtp. (Found on stack overflow).

1. The first file to modify is xampp/sendmail/sendmail.ini. Basically, you go through the file and change values so that they look like this:

**[sendmail]  
  
smtp\_server=smtp.gmail.com  
smtp\_port=587  
error\_logfile=error.log  
debug\_logfile=debug.log  
auth\_username=phase4CyberSec2@gmail.com  
auth\_password=77malpHAS23  
force\_sender=**[**phase4CyberSec2@gmail.com**](mailto:phase4CyberSec2@gmail.com)

1. The next is xampp/php/php.ini. Modify the mail function section of the file to look like this:

**[mail function]  
; For Win32 only.  
; http://php.net/smtp  
;SMTP = localhost  
; http://php.net/smtp-port  
;smtp\_port = 25  
  
; For Win32 only.  
; http://php.net/sendmail-from  
;sendmail\_from = postmaster@localhost  
  
; For Unix only. You may supply arguments as well (default: "sendmail -t -i").  
; http://php.net/sendmail-path  
sendmail\_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"  
  
; Force the addition of the specified parameters to be passed as extra parameters  
; to the sendmail binary. These parameters will always replace the value of  
; the 5th parameter to mail(), even in safe mode.  
;mail.force\_extra\_parameters =  
  
; Add X-PHP-Originating-Script: that will include uid of the script followed by the filename  
mail.add\_x\_header = Off  
  
; Log all mail() calls including the full path of the script, line #, to address and headers  
;mail.log = "C:\xampp\apache\logs\php\_mail.log"**

## PDF Generation config information

To generate the PDF files for the certificates I used [FPDF](http://www.fpdf.org/). To use, follow these steps:

* Download the zipped folder fpdf181.
* Extract the contents of fpdf181 to a folder of the same name (ex:Sources\fpdf181\fpdf.php) in the htdocs section of XAMPP in the same directory as the rest of the source code for the system.

## Email server credentials

I took the liberty of setting up an email server. It’s just a gmail account. Here is the username and password.

* **Username:** [phase4CyberSec2@gmail.com](mailto:phase4CyberSec2@gmail.com)
* **Password:** 77malpHAS23

I also lightened security on the account to allow less secure applications to access the account.

## Important Note

Once you change the .ini files, be sure to restart XAMPP by opening the control panel and then stop and restart all of the processes.

# Unity

The games are built as Web GL in unity and are located in the github files you already downloaded. The html files have the embedded link to the unity game. In order to get the project running you do not need to do anything and can skip to to the next section. If you want to add or make changes you must load the source code unity project file and rebuild the project. Then manually update the specific unity code folder corresponding to the quiz you are changing.

First you must download and install the latest version of unity from:

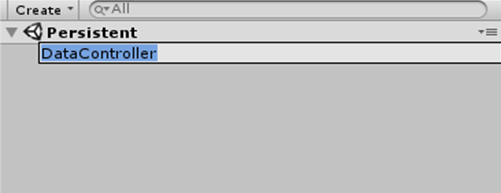
<https://unity3d.com/get-unity/download>

Following the set up instructions you must ensure you check the “web GL” feature so the games can be built and used in html code. Once installed you can open the project. Opening the project will display the project as a whole. Modification is as follows

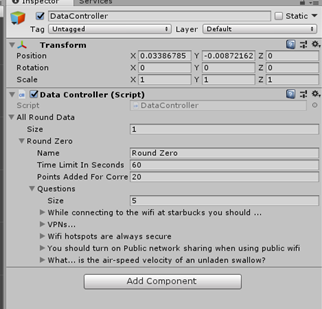
**Changing the quiz questions only:**

To change the quiz game questions, go to the scenes folder, then select persistent then select Data Controller.





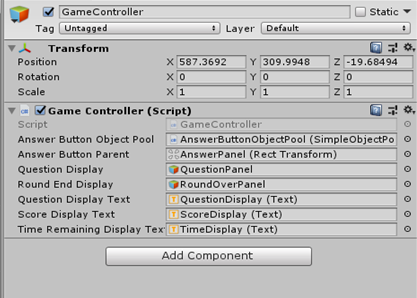
This will show the following:



From here you can resize the quiz to customize questions and time.

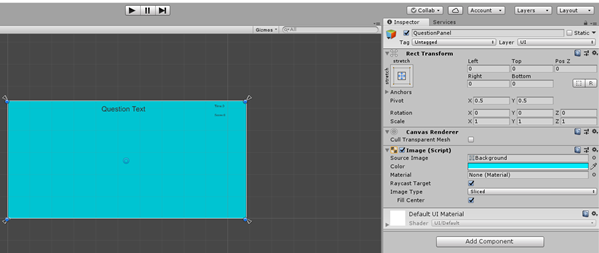
**Overall Guide to unity and how the code runs:**

The Menu Screen displays what you see at the start and is what you return to at the end of the game. Once you click the start game it loads the game and displays the Question Panel. The round over Panel remains hidden until the game is out of questions. The UI is displayed the whole time the game is running. All code that actually makes this all work is in the Scripts folder. Actions are linked between the scripts and certain button clicks through the Game controller (pic below).



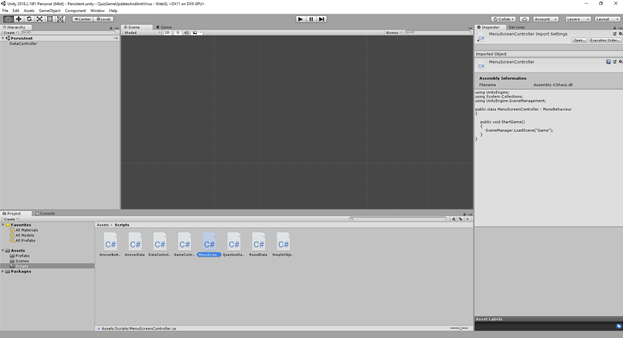
The Prefab allows you to recreate instances of the same object(in our case the answer button) and should not be changed unless you want to break the whole game! The Persistent Scene is what stores the game data and lasts through run time.

Things like title and color is easily controlled in the specific scenes.(as shown)



The Scripts folder holds all the code that actually makes the unity interface and c# code work together.

Note: The SimpleObjectPool is a general garbage collection class from unity and should be left alone! This will ensure garbage collection does not happen while the game is running!



General Disclaimer: Without a general understanding of unity and c# this guide could be hard to follow as linking scripts is very important to get the game running correctly. Adding text to certain scenes and changing general simple things like color and questions is easy while trying to add more to the quiz would require more coding.

# Errors/Possible Problems

This system has many errors and flaws, this section defines what errors and problems with the code that we know of.

1. The config file for the database currently resides in the htdocs folder, therefore the host, username, password, and database fields are all vulnerable.
2. The system does not check if the user’s entered email is a valid address.
3. The system does not do entropy checks on the password entered by the user.
4. Using the previous/next buttons after you get to the game page can cause the game to bug out in weird, unforeseen ways.
5. The case where a user passes the test twice has not been fully tested, so errors may occur in that case.
6. The case where a user attempts to send an email without a certificate already created has not been tested, so errors may occur in that case.