Installation Notes:

This guide will help you install the Malawi Gradebook System on a Ubuntu Linux machine

LAMP

To install Apache:
Run:
```
sudo apt update
sudo apt install apache2
```

It will ask if you want to allocate additional disk space. Select Y.

If a message saying Pending kernel upgrade appears, press OK

Type in IP address in web browser, If the default apache page appears, then success.

To install MariaDB:
Run:
```
sudo apt install mariadb-client mariadb-server
```

It will ask if you want to allocate additional disk space. Select Y.
If a message saying Pending kernel upgrade appears, press OK

To install PHP:

Run:
```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install php
```

You can check version with:
```
php -version
```

CodeIgniter

Install PHP extensions:
Codeigniter requires some php extensions to be installed.
These include:

- Intl
- Mbstring

- Json

Run:
```
sudo apt-get install -y php-intl
sudo apt-get install php-mbstring
```

And Json should be enabled by default

Add CodeIgniter Project to server:
Run:
```
cd /var/www/html
```

to navigate to the /html directory.

Initialize the folder as a git repository:

```
sudo git init
```

Pull from the UP MalawiBravo git repository:
```
sudo git pull
https://github.com/upcs/cs341-spring-2023-malawi_bravo.git
```

Restart the Apache server:
```
sudo service apache2 restart
```

Delete the default apache page contained in the html folder:
```
cd var/www/html
sudo rm index.html
```

At this point, you should have a 404 not found error

Update project base URL:

Open Codeigniter/app/Config/App.php

Change the $baseURL to your machine's specific IP Address:
```
public string $baseURL = 'http://exampleIPAddress/';
```

Restart the Apache server:
```
sudo service apache2 restart
```

Change write access to Writable folder

```
cd var/www/html/CodeIgniter
```

```
sudo chmod -R 755 writable/
sudo chown -R www-data:www-data writable/
```

Enable mod_rewrite

Run:
```
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Open this file: /etc/apache2/sites-available/000-default.conf
Add these lines to the file:
```
<VirtualHost *:80>
    <Directory /var/www/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    . . .
</VirtualHost>
```

Change the Document Root in all these files to include CodeIgniter:

```
DocumentRoot /var/www/html/ -> DocumentRoot /var/www/html/CodeIgniter
```

Restart Apache:

```
sudo systemctl restart apache2
```

Database Setup

Install phpMyAdmin

```
sudo apt-get install phpmyadmin
```

Choose apache 2

Create Root User for MariaDB

To check version, run:

```
mysql --version
```

Stop the MySQL server:

```
sudo systemctl stop mariadb
```

Start the database without loading the grant tables or enabling networking:

```
sudo mysqld_safe --skip-grant-tables --skip-networking  &
```

Open mysql

```
mysql -u root
```

Let's tell the database server to reload the grant tables by issuing the FLUSH PRIVILEGES command

```
FLUSH PRIVILEGES;
```

Run this:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

Or this:

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('new_password');
```

depending on your mysql version, and replace new_password with your password.

You should receive a message like this:

```
Output
Query OK, 0 rows affected (0.00 sec)
```

Kill the MariaDB process

```
sudo kill `/var/run/mariadb/mariadb.pid`
```

*If this command doesn't work, run:

```
ps aux | grep mysql
```

```
Which should output something like this:
user@hostname:~$ ps aux | grep mysql
mysql     1681  0.1 17.7 684476 88716 ?        Ssl  10:07   0:04
/usr/sbin/mysqld
luke      5294  0.0  0.1  14232   944 pts/0    S+   10:50   0:00 grep
--color=auto mysql
user@hostname:~$
```

Then run:

```
sudo kill 1681
```

Now we want to restart the server, so run:

```
sudo systemctl start mariadb
```

Now you can log in with root access through this command:

```
mysql -u root -p
```

Connecting CodeIgniter to the Database

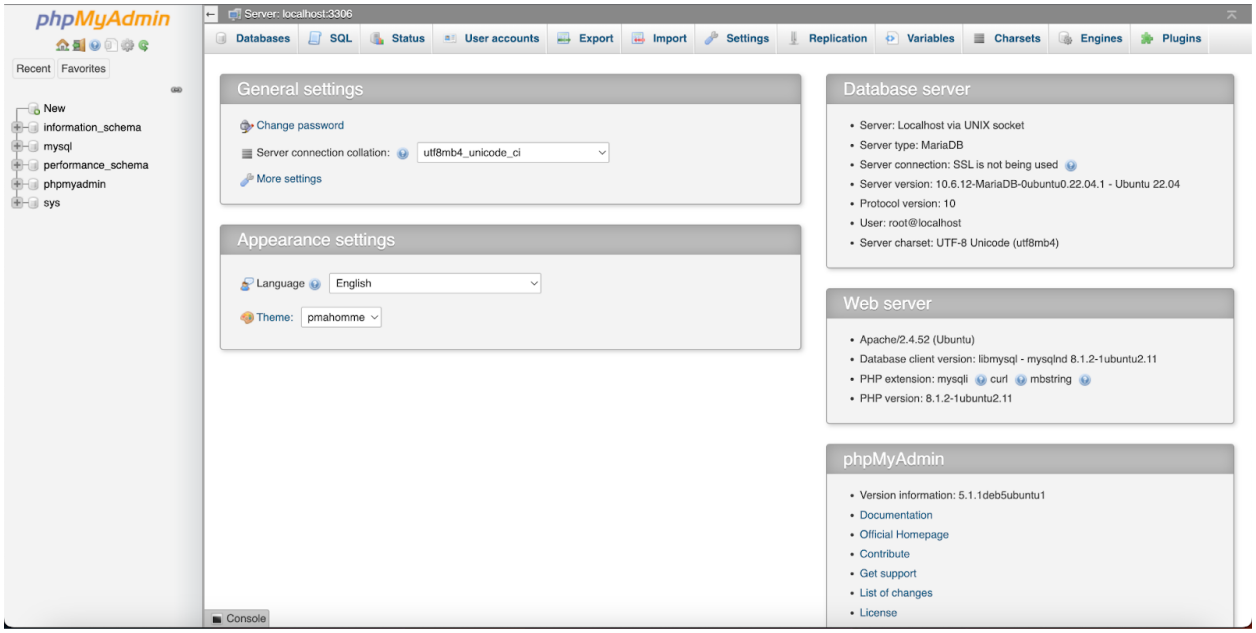
Open the .env file under the /Codeigniter directory
This contains all information need to to connect to MariaDB

Make sure that the login credentials are correct

Formatting the Database

Head back to phpmyadmin through the website
Log in with root as the user and the newly created password

It should look like this:



Click on the import tab in the top navbar

There is a file called ‘localhost.sql’ in the var/www/html directory
Choose that file to import to the database.

Click Go in the bottom right of the screen.

This should import all the data successfully

Now the server is up and running. Great Job!!

To log in to the dashboard, use these credentials:
Username: test3@gmail.com
Password: test3

User Guide:

This guide will explain the features of the software

User Authentication

This project restricts access to the database by requiring a user to log in with a valid username and password. New users can be created once a user has been logged in, and all the users information is stored in the database

Add Students/Classes

Students and Classes can be added to the database by filling out the forms

Print Report Cards

Upon Clicking this button, a list of students will appear. If one is selected, all their classes and grads will be shown in the database.

Edit Grades

All classes will be shown, and a single class can be selected. If the list is too long, a search feature has been provided to search by class name. Upon selecting a class, all students enrolled in the class can have their grades edited simultaneously.

Assign Students

First a class is chosen, then students from a list can be selected to be added to a class.

Add New Account

New Accounts can be added to the database to allow users to log in. At some point this functionality may be extended to allow users with limited access such as parents to view information in the database.

Direct Database Access

Administrators can access the database directly though the phpMyAdmin tool. It has its own authentication system which has the same credentials as the database. From here, more complex operations on the database, such as removing data or modifying the database.