

# Blind Grading System for Ethical Law School

---

## Project Commission

---

Ethical Law School would like to remove all bias between professors and student exams. In order to achieve this there needs to be a layer of abstraction between exam attempts and the student name/ID numbers that took those exams. The faculty will then grade submissions not knowing who the student was and therefore unable to channel any subconscious bias when grading. The database contains data about students, exams, and courses. The database used is MySQL and the application will be built on the Laravel framework for PHP.

## Highlights

---

- Comprised of 14 tables (BCNF)
- Allows for user sign-in and registration
- Select, Inserts and Updates handled by application

## Technology Used

---

- Docker Desktop
- MariaDB 10.3.6 (MySQL Server Variant)
- PHP 7.2.5
- Laravel PHP Framework
- Nginx Webserver
- HTML5
- CSS3

## How to set up Local Installation

---

In order to setup this application you will need PHP programming language, MySQL database and a webserver. One approach would be to install something like XAMPP on

your desktop and install this directory wherever Apache's document root has been set up. Unfortunately, you will have to do more research for this setup, since that is not the preferred way we used. Instead we utilized Docker Desktop to virtualize a linux operating system on our local machine. Here are the steps:

1. Download Docker Desktop for your Operating System and install it.
  - IMPORTANT NOTE: If installing on Windows, please make sure to uncheck or decline the use of Windows Containers. You want to use Linux containers. [Youtube Tutorial on Installing Docker Desktop for Windows](#)
2. Download the source code from this project by downloading it from its [GitHub Repository](#). Unzip it the file to a location you can recall later.
3. Once you have Docker Desktop installed and running on your machine, make a copy of the sample environment file, [.env.example](#). You should be able to use the sample to run environment file to build the containers.
4. Open a terminal window or command prompt for your operating system.
5. Change directory to the location you unzipped it to. When you are within the directory you need to run the following command: `docker-compose up --build`. This will read the [project's docker-compose.yml](#) and install all of the necessary containers on your local machine. Building may take anywhere between 10-20 minutes depending on your internet speed.
6. Once the containers are deployed and running, you will need to execute the following command to initialize the database for laravel. `docker-compose exec app php artisan migrate`.
7. At this point the application should be running and listening for connections at (<http://0.0.0.0:8090>) OR (<http://localhost:8090>).

## Populate database with dummy data

---

To populate the database with dummy data, you will need to have the docker containers successfully running and also have some MySQL client on your local machine. MySQL Workbench works well on Windows and Sequel Pro for MacOS.

1. In your preferred MySQL client make a connection to the MySQL database running within the Docker container.
  - Host: 0.0.0.0 (or localhost or 127.0.0.1)
  - Port: 33061 (or the value you set in [.env](#) for DB\_HOST\_PORT)
  - User: root
  - Password: password (or the value you set in [.env](#) for MYSQL\_ROOT\_PASSWORD)

- Database: laravel (or the value you set in [.env](#) for DB\_DATABASE)
- 2. Open [database-extract.sql](#) in a text editor and copy all of its contents.
- 3. Once connected to the database execute a query from the contents you copied from [database-extract.sql](#). The execution of this query will generate data for use.

## Password for all dummy data accounts

The password for all accounts from the dummy data is `testaccount` .