KULLIYYAH OF INFORMATION & COMMUNICATION TECHNOLOGY

INFO 3305 WEB APPLICATION DEVELOPMENT
SECTION 2
SEMESTER 1, 2020/2021

FINAL REPORT
TITLE : IIUM COMPOUND MANAGEMENT SYSTEM

LECTURER:
MARINI BINTI OTHMAN

PREPARED BY:

NAME	MATRIC NO
MUHAMMAD HANIF BIN SALEHUDDIN	1722737
MAISARAH BINTI HAMZAH	1810328
EZZATI SHAKIRAH BINTI SAHRUDIN	1812224
MOHAMMAD FITRI AIMAN BIN MOHAMMAD AZLI	1810925
SYED MUHAMMAD AIMAN BIN SYED MOHD AZMI	1727935

TABLE OF CONTENTS

INTRODUCTION

This web application enables IIUM administrators to manage compound details effortlessly. Traditionally, all fines will be written in a receipt and were asked for students to bring them to pay at the counter. Thus, this system will help ease administrators to key in students' compound record in a system. User (i.e. Admin) can add, edit and delete students' records through the system. Moreover, this application will allow Admin to check the status of students' compounds and manage them easily.

OBJECTIVES

1. To manage students' compounds by create, edit, delete the record.
2. To store astudents' compound records in a secure database.
3. To reduce the usage of papers while filing compounds.

FEATURES AND FUNCTIONALITIES

- Register page - User/Admin to register their name, email and password to be able to access the system later.
- Login page - User/Admin to login to the system
- Homepage - Shows a previous list of students' compound record for existing User/Admin. Otherwise, it will be blank for newly registered User/Admin.
- Add New Student - Add new students' compound record
- Edit - Make changes on existing students' compound record
- Delete - Remove students' compound record from the list (i.e. student have paid the compound)

1. Controllers

There are 2 controllers created in this project. First is Home Controller; that is automatically added when we execute *php artisan ui:auth* command. It acts as an Authentication Controller. Second is the Student Controller that manages the CRUD operation. It handles adding, deleting and editing students' compound records.

2. Views

We have created a folder view named Student. Inside of Student view, there are codes to display every operation made from the Controller.

1. Create.blade.php
   It is a form for user/admin to key in students' records.
2. Index.blade.php

It shows a list/details of students' compound record with an option to Delete or Edit the record.

3. Edit.blade.php
   It is a form for users/admin to make changes to students' compound records and save them.

3. Models

We created a Students model to initialize all $fillable. A Users model has also automatically created for the login/register purposes.

4. Migrations

Used to initialize the column needed to be migrated to the database by performing *php artisan migrate*

5. Error Checking

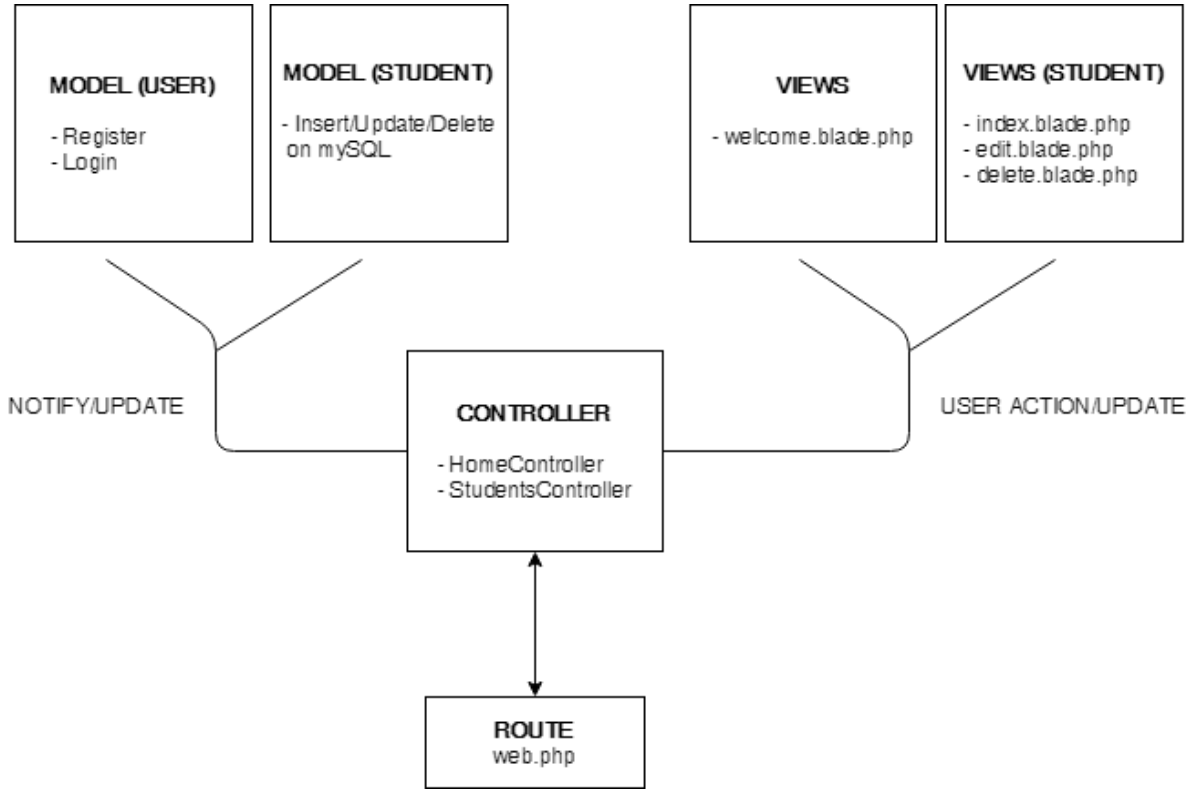
Error Checking is implemented in verifying matched email and password. When a registered user has entered the wrong password, a warning message will appear.

6. User Authentication

By running *php artisan make:auth,* the user authentication controller files have been created and implemented in this project. We also used a @csrf token that is used to verify that the authenticated user is the one actually making the requests to the application. Otherwise, they cannot proceed to enter and make changes to the system. Furthermore, the @auth and @guest directives have been used to quickly determine if the current user is authenticated or is a guest that is about to register to the system.

MVC DIAGRAM

The diagram below shows the Model-View and Control canvas for our project.



REFERENCES

1. Requests & Input. Retrieved from https://laravel.com/docs/4.2/requests
2. CSRF Token. Retrieved from https://laravel.com/docs/7.x/csrf
3. Authentication Directives. Retrieved from https://laravel.com/docs/5.7/authentication
4. Blade Templates. Retrieve from https://laravel.com/docs/8.x/blade#layouts-using-components
5. MVC (Models, Views, Controllers). Retrieved from https://blog.pusher.com/laravel-mvc-use/
6. Loop Variable. Retrieved from https://tutsforweb.com/loop-variable-foreach-blade-laravel/