Researching Project IoT Platform using Laravel and Mobile App

' Week4 Report '

Lecturer: Mr. HEL Chanthan

Prepared by Mr. VANNAK Sovannroth

2019-2020

Outline

- Planning for First Month
- Planning for Second Month
- Creating User Login and Register
- Starting apply Admin Template for User login
- Starting apply Admin Template for Backend
- Creating User (create, edit & delete) on backend connect to database
- Creating Route of User, Category, Company, Social, and About
- Creating Controller for collections
- Creating Alert Message

❖ Planning for First Month

Activity (Start on 06.08.2020)	Thursday (Presentation day)	Friday	Monday	Tuesday	Wednesday (Meeting day)
Week 1	Installation MongoDB (Shell and Compass) and connect localhost between Shell and Compass.(Locally)	Performing CRUD (Create, Read, Update, Delete) on Mongo Shell and MongoDB Compass.	Creating MongoDB Atlas account and connect it to Mongo Shell or MongoDB Compass, and Driver (Node.js).(Cloud)	Installation Visual Studio Code and connect driver to MongoDB Atlas.(Cloud)	Connecting sample Backend to MongoDB Cluster and storing Products in the Database.
Week 2	Connecting driver to Mongo Shell.(Locally)	Starting to build sample backend connect with or without MongoDB Atlas.	Build creating, editing and deleting Products on server.	Transmitting and fetching data to/from the Database.	Creating Login and Signup. (Email and password)
Week 3	Learning template Bootstrap 4	Install wampserver and Laravel	Run Laravel server with MySQL(Database)	Choose free template Bootstrap for Front-End & Backend	Installation Laravel Module for Backend and Front-End
Week 4	Starting to build backend for project. & Creating Login and Signup.	User management: • Create new user • Reset user password • Delete and edit user	Manage Info: • Logo • Name • Contact info • Map • Social	Manage new category: • Create new category • Edit and delete exiting categories.	Manage news: • Post news • Edit news • Delete news

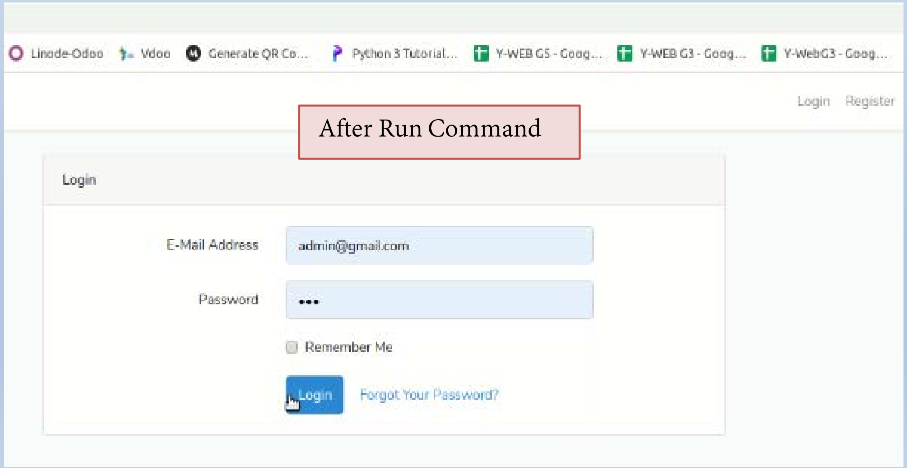
❖ Planning for Second Month

Activity (Start on 06.08.2020)	Thursday (Presentation day)	Friday	Monday	Tuesday	Wednesday (Meeting day)
Week 5	Create Dashboard	Create Dashboard	Manage Calculate: • Addition • Subtraction • Multiplication • Division	Write API & Read API	Write API & Read API
Week 6	Connect to ESP8266 or NodeMcu & fetch data from Arduino	Get data from Sensors and Calculate.	Show data on Dashboard	Connect Backend to Front-End	Front-End fetch data from Backend
Week 7	Choose Flutter or React Native for build Mobile App.	Learning Flutter or React native	Start build sample project	Create Horizontal list view & build recent products grid view.	Manage Info: • Product details • Home page Ui modification • Shopping cart & list page • User login & Register • Contact info
Week 8	Creating Users, brand and category on database & Adding products to the database.	Auto complete search for categories and brands & Uploading Images to firebase.	Login and signup screen redesign & Google sign in for IOS and Android .	Firebase Auth using provider package & Dashboard UI/UX and Database product modeling.	Load products from firebase part & loading products from database.

❖ Creating User Login and Register

- ❑ Stage1 Run migration for user login (Make sure already create database as greenhouse)
  $ php artisan migrate
- ❑ Stage2 Generate user login in Laravel 7
  $ composer require laravel/ui
  $ php artisan ui bootstrap –auth
  $ npm install && npm run dev

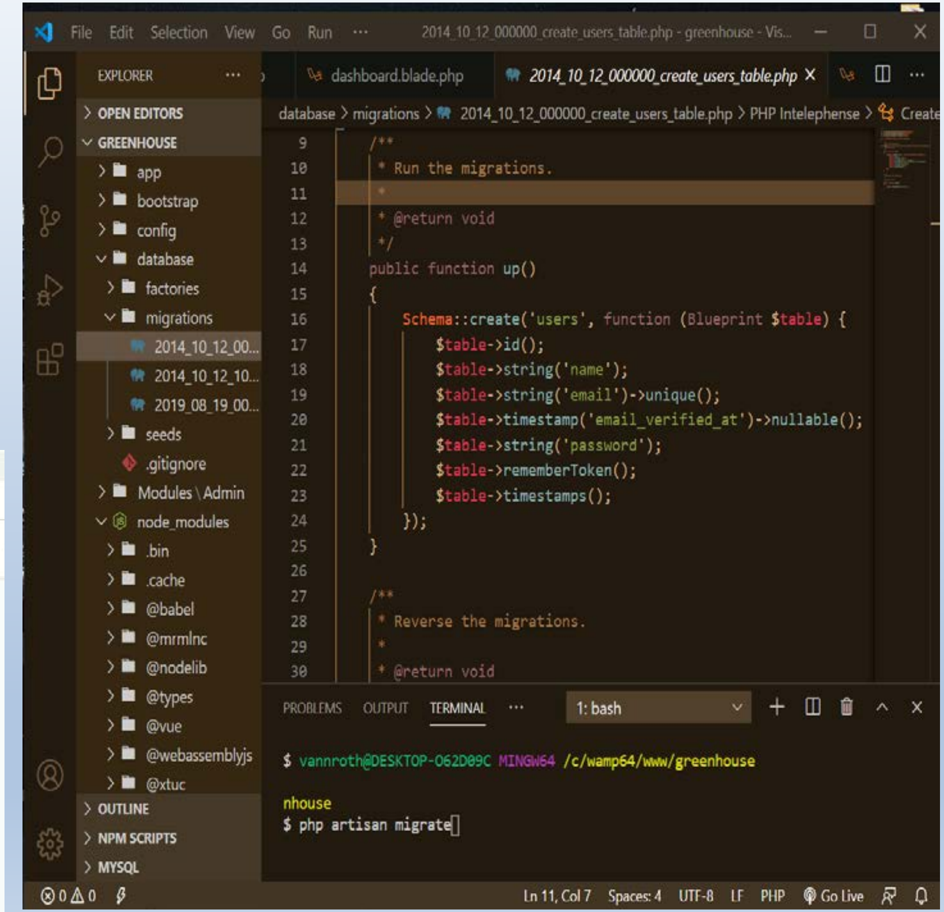
https://www.itsolutionstuff.com/post/laravel-6-auth-login-with-username-or-email-tutorialexample.html



After Run Command

Login

E-Mail Address	admin@gmail.com
Password	•••
	Remember Me



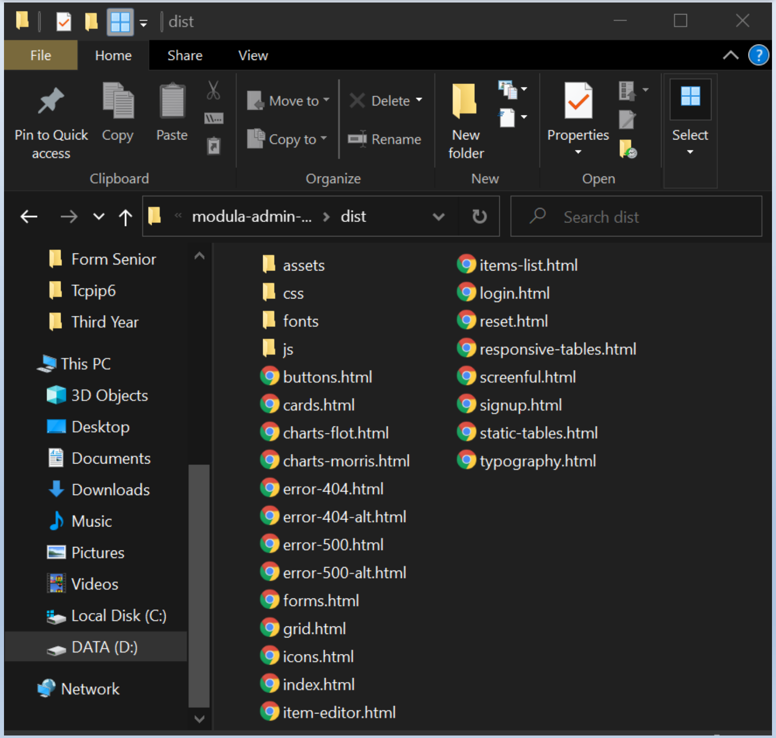
```
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
```

```
$ vannroth@DESKTOP-062D09C MINGW64 /c/wamp64/www/greenhouse

nhouse
$ php artisan migrate
```

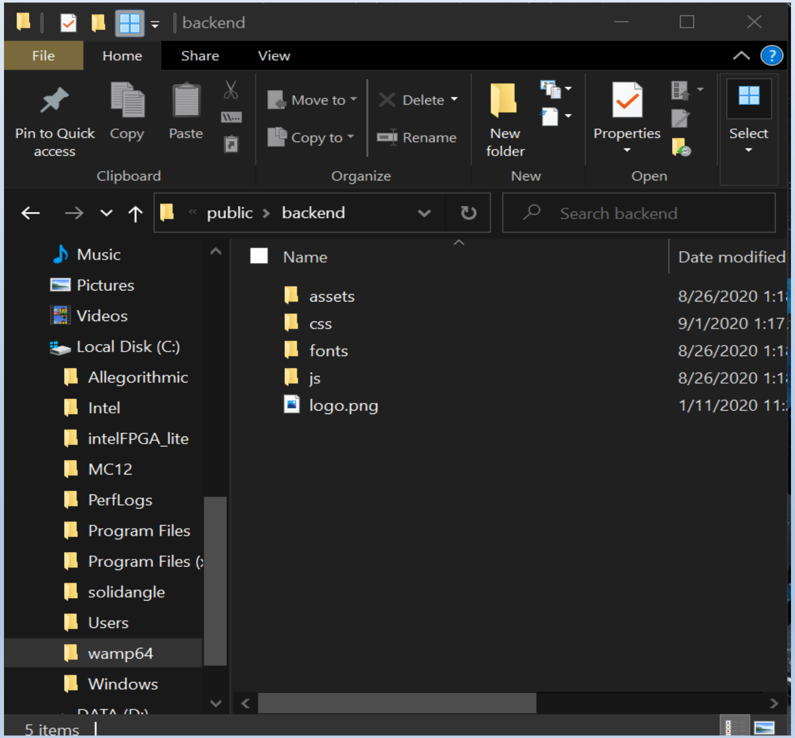
❖ Starting apply Admin Template for User login

Create new folder as backend in greenhouse\public and copy all folder in admin template and paste in folder greenhouse\public\backend.

https://github.com/Chanthan89/Smart-Irrigation/tree/master/Weekly%20Presentation/Sept-2020/VANNAK%20Sovannroth/User%20Login

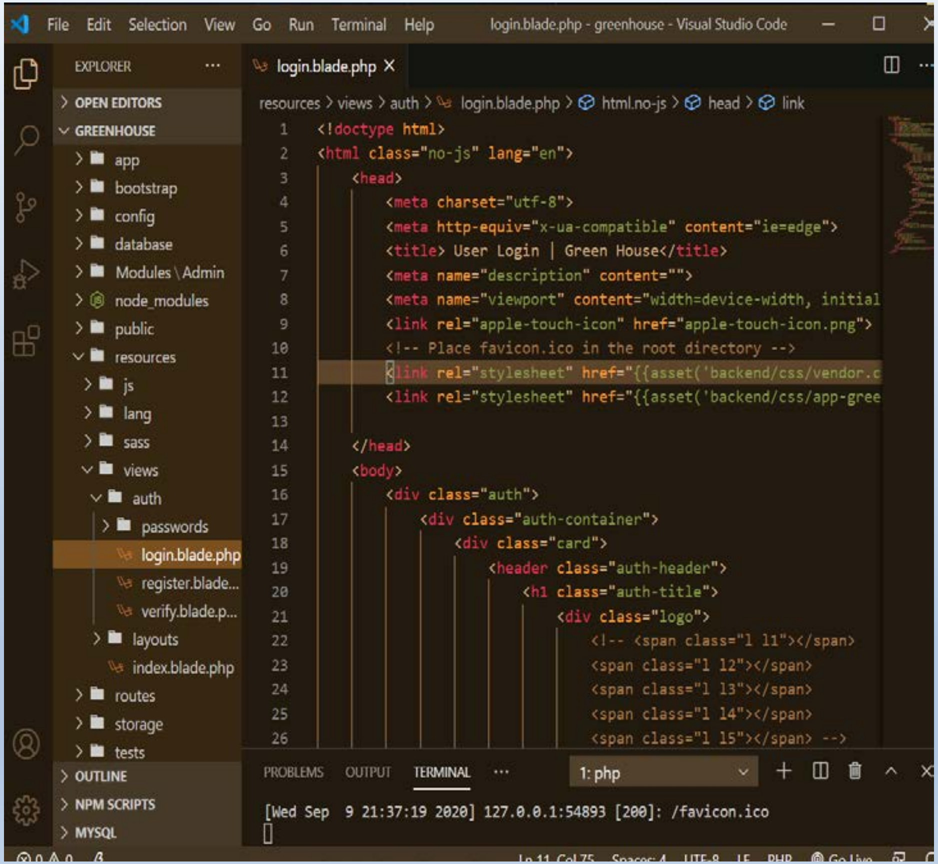


assets
css
fonts
js
buttons.html
cards.html
charts-flot.html
charts-morris.html
error-404.html
error-404-alt.html
error-500.html
error-500-alt.html
forms.html
grid.html
icons.html
index.html
item-editor.html
items-list.html
login.html
reset.html
responsive-tables.html
screenful.html
signup.html
static-tables.html
typography.html



Name	Date modified
assets	8/26/2020 1:1
css	9/1/2020 1:17
fonts	8/26/2020 1:1
js	8/26/2020 1:1
logo.png	1/11/2020 11:

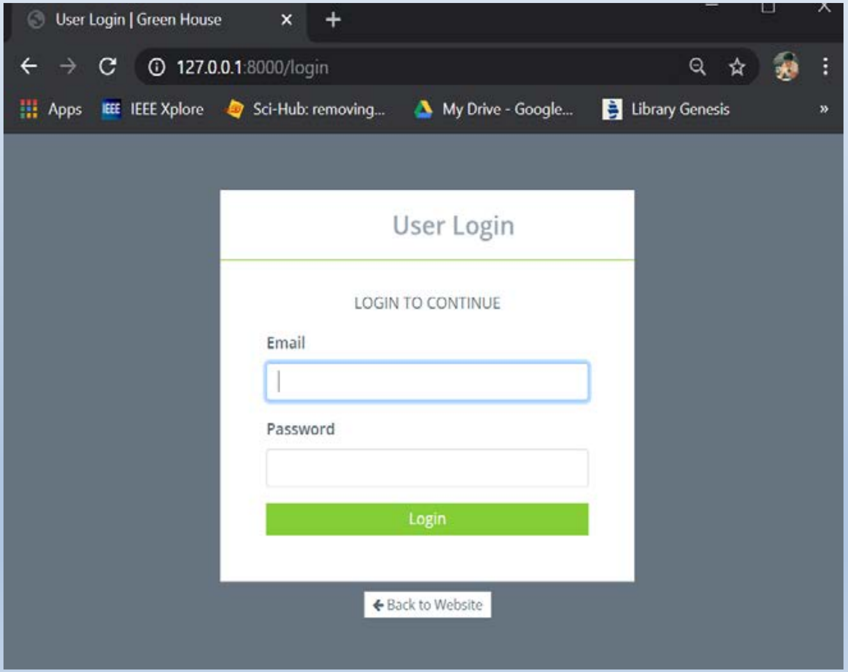
❖ Starting apply Admin Template for User login



```
<!doctype html>
<html class="no-js" lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="x-ua-compatible" content="ie=edge">
        <title> User Login | Green House</title>
        <meta name="description" content="">
        <meta name="viewport" content="width=device-width, initial
        <link rel="apple-touch-icon" href="apple-touch-icon.png">
        <!-- Place favicon.ico in the root directory -->
        <link rel="stylesheet" href="{{asset('backend/css/vendor.c
        <link rel="stylesheet" href="{{asset('backend/css/app-gree

    </head>
    <body>
        <div class="auth">
            <div class="auth-container">
                <div class="card">
                    <header class="auth-header">
                        <h1 class="auth-title">
                            <div class="logo">
                                <!-- <span class="l l1"></span>
                                <span class="l l2"></span>
                                <span class="l l3"></span>
                                <span class="l l4"></span>
                                <span class="l l5"></span> -->
```

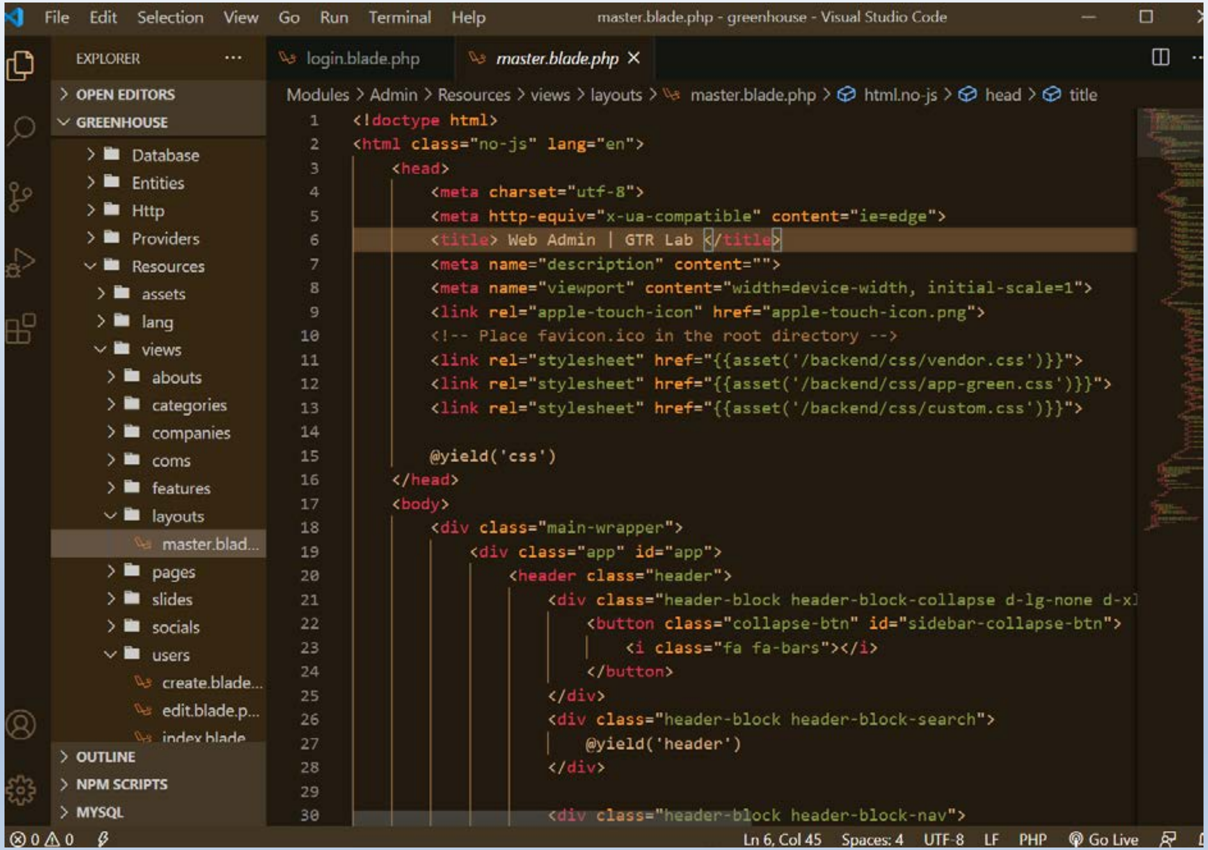
```
[Wed Sep  9 21:37:19 2020] 127.0.0.1:54893 [200]: /favicon.ico
```

Open file login.html with VS code and copy code to login.blade.php, and then edit link with old code and new code.



PAGE 06

❖ Starting apply Admin Template for Backend

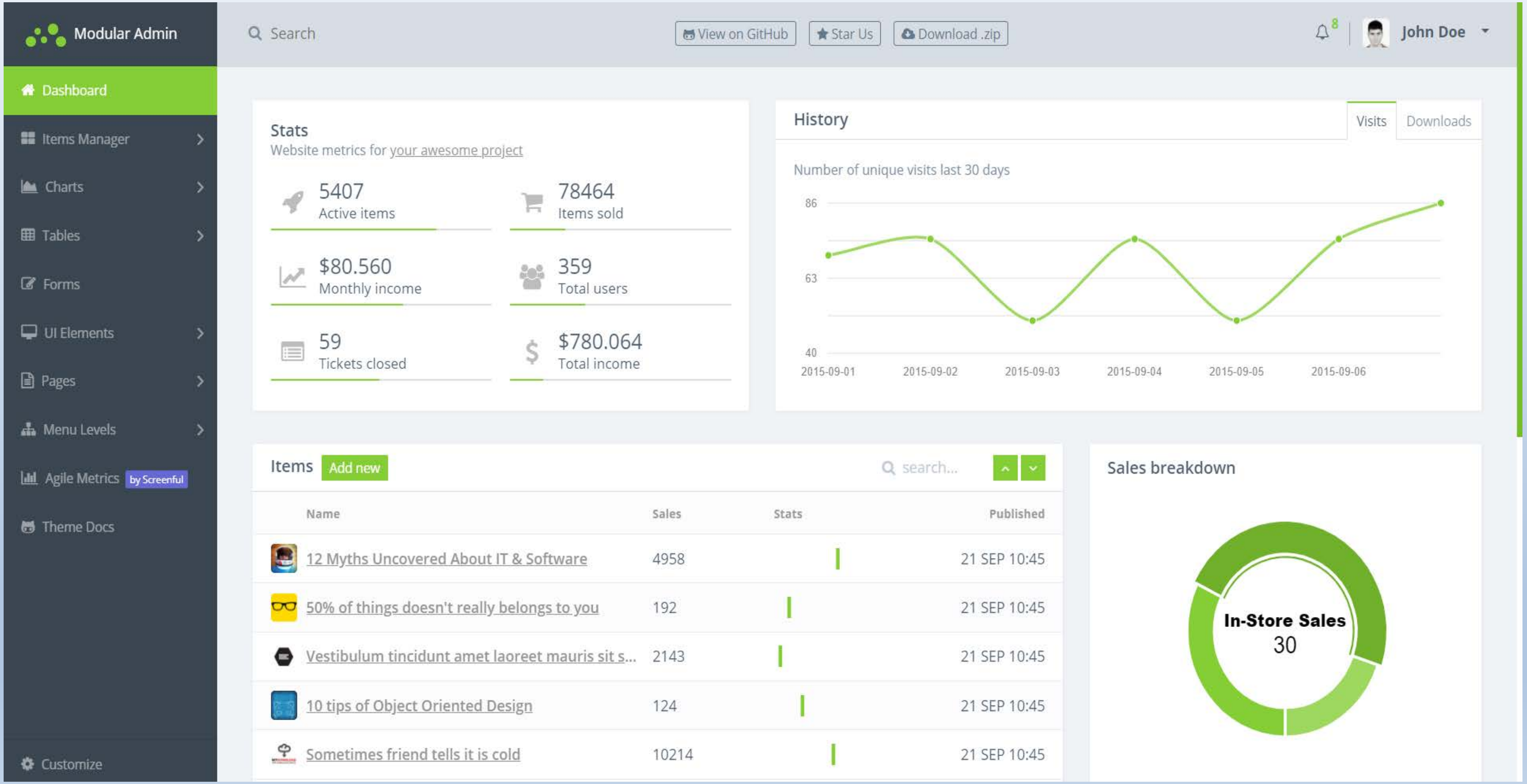


```
<!doctype html>
<html class="no-js" lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="x-ua-compatible" content="ie=edge">
        <title> Web Admin | GTR Lab </title>
        <meta name="description" content="">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel="apple-touch-icon" href="apple-touch-icon.png">
        <!-- Place favicon.ico in the root directory -->
        <link rel="stylesheet" href="{{asset('/backend/css/vendor.css')}}">
        <link rel="stylesheet" href="{{asset('/backend/css/app-green.css')}}">
        <link rel="stylesheet" href="{{asset('/backend/css/custom.css')}}">

        @yield('css')
    </head>
    <body>
        <div class="main-wrapper">
            <div class="app" id="app">
                <header class="header">
                    <div class="header-block header-block-collapse d-lg-none d-x
                        <button class="collapse-btn" id="sidebar-collapse-btn">
                            <i class="fa fa-bars"></i>
                        </button>
                    </div>
                    <div class="header-block header-block-search">
                        @yield('header')
                    </div>

                    <div class="header-block header-block-nav">
```

Open file index.html with VS code and copy code to master.blade.php in Module for backend, and then edit link with old code and new code.

https://github.com/Chanthan89/Smart-Irrigation/tree/master/Weekly%20Presentation/Sept-2020/VANNAK%20Sovannroth/Create%20Admin

❖ Starting apply Admin Template for Backend



Stats

Website metrics for your awesome project

5407 Active items

78464 Items sold

$80.560 Monthly income

359 Total users

59 Tickets closed

$780.064 Total income

History

Visits | Downloads

Number of unique visits last 30 days

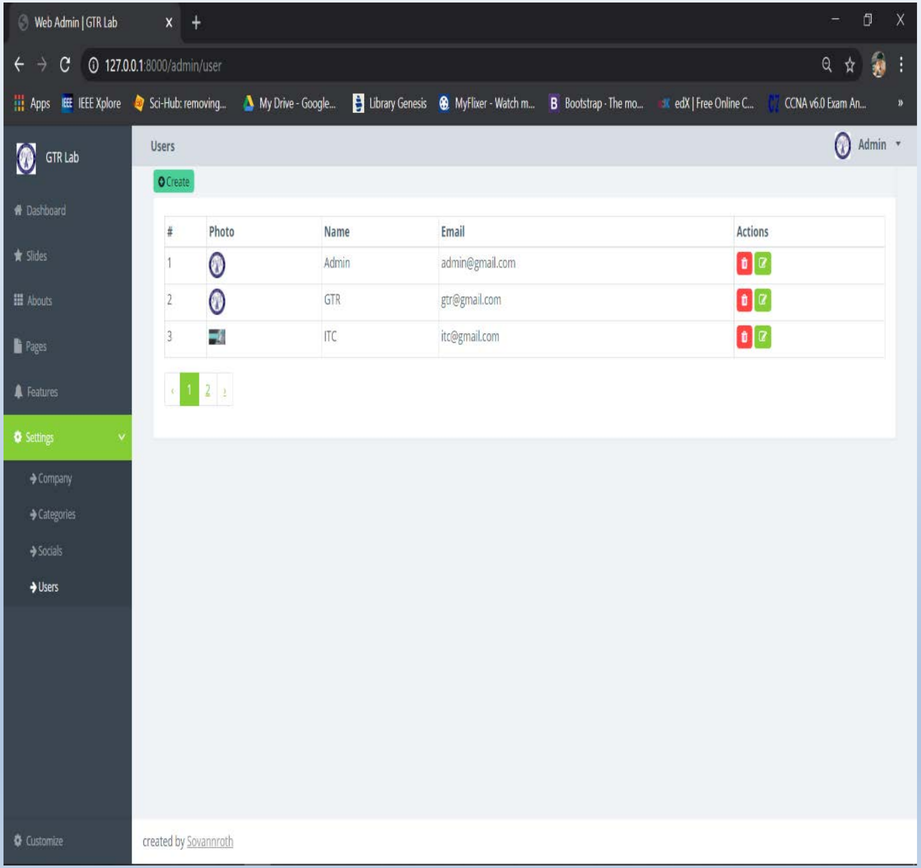
Items Add new

search...

Name	Sales	Stats	Published
12 Myths Uncovered About IT & Software	4958		21 SEP 10:45
50% of things doesn't really belongs to you	192		21 SEP 10:45
Vestibulum tincidunt amet laoreet mauris sit s...	2143		21 SEP 10:45
10 tips of Object Oriented Design	124		21 SEP 10:45
Sometimes friend tells it is cold	10214		21 SEP 10:45

Sales breakdown

❖ Creating User (create, edit & delete) on backend connect to database

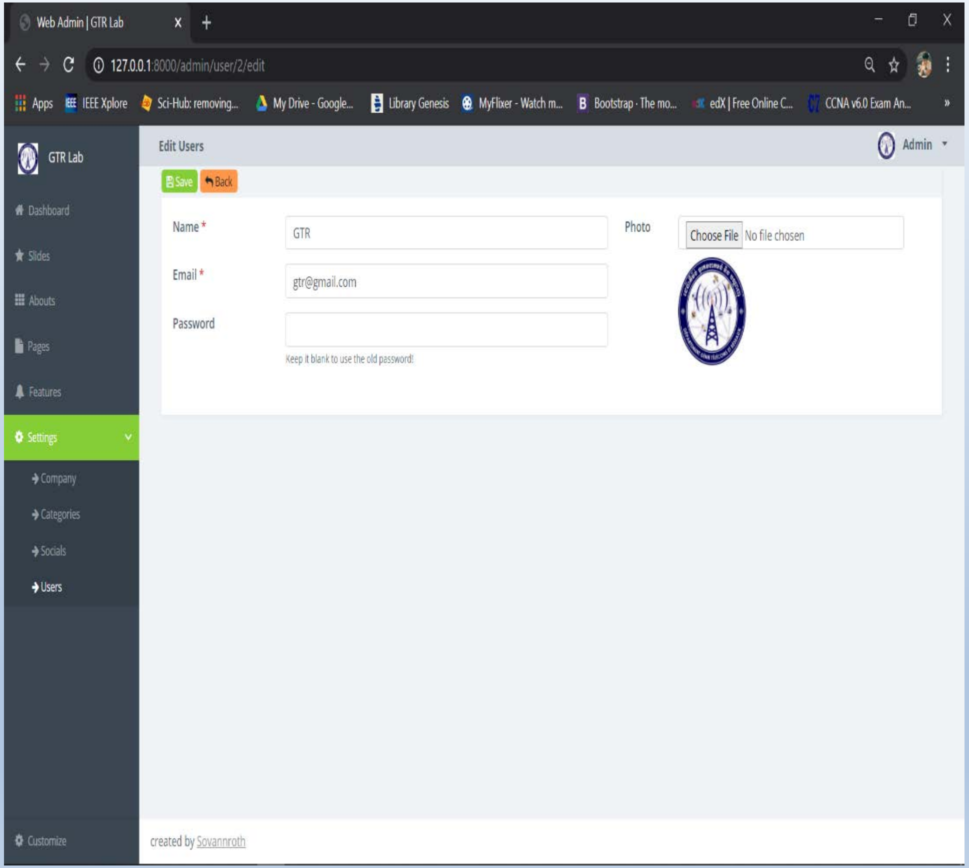


Users

#	Photo	Name	Email	Actions
1		Admin	admin@gmail.com	
2		GTR	gtr@gmail.com	
3		ITC	itc@gmail.com	

‹ 1 2 ›

created by Sovannroth



Edit Users

Name * GTR

Email * gtr@gmail.com

Password

Keep it blank to use the old password!

Photo No file chosen

created by Sovannroth

❖ Creating User (create, edit & delete) on backend connect to database

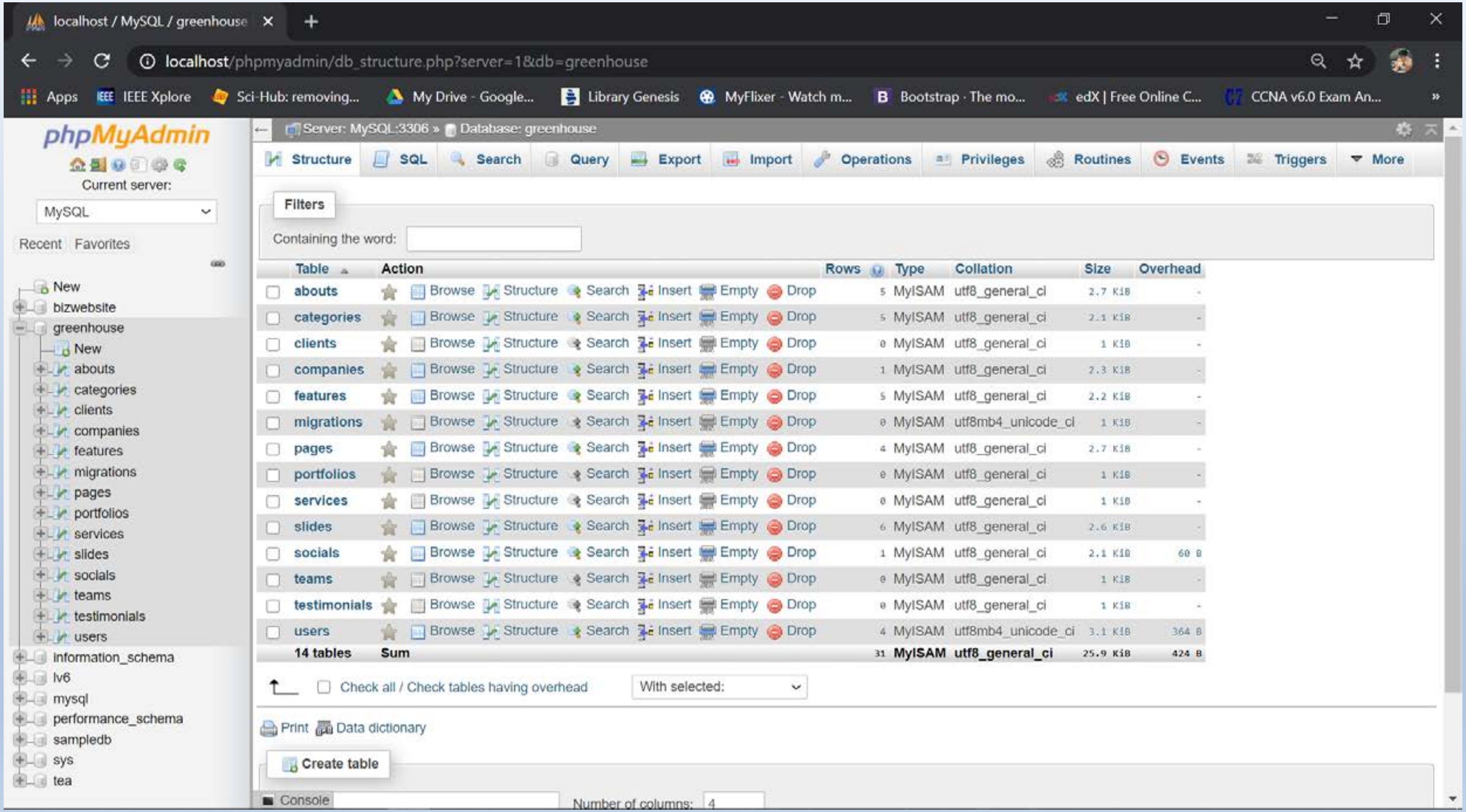


Table	Action	Rows	Type	Collation	Size	Overhead
abouts	Browse Structure Search Insert Empty Drop	5	MyISAM	utf8_general_ci	2.7 KiB	-
categories	Browse Structure Search Insert Empty Drop	5	MyISAM	utf8_general_ci	2.1 KiB	-
clients	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
companies	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	2.3 KiB	-
features	Browse Structure Search Insert Empty Drop	5	MyISAM	utf8_general_ci	2.2 KiB	-
migrations	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8mb4_unicode_ci	1 KiB	-
pages	Browse Structure Search Insert Empty Drop	4	MyISAM	utf8_general_ci	2.7 KiB	-
portfolios	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
services	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
slides	Browse Structure Search Insert Empty Drop	6	MyISAM	utf8_general_ci	2.6 KiB	-
socials	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	2.1 KiB	60 B
teams	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
testimonials	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
users	Browse Structure Search Insert Empty Drop	4	MyISAM	utf8mb4_unicode_ci	3.1 KiB	364 B
14 tables	Sum	31	MyISAM	utf8_general_ci	25.9 KiB	424 B

❖ Creating User (create, edit & delete) on backend connect to database

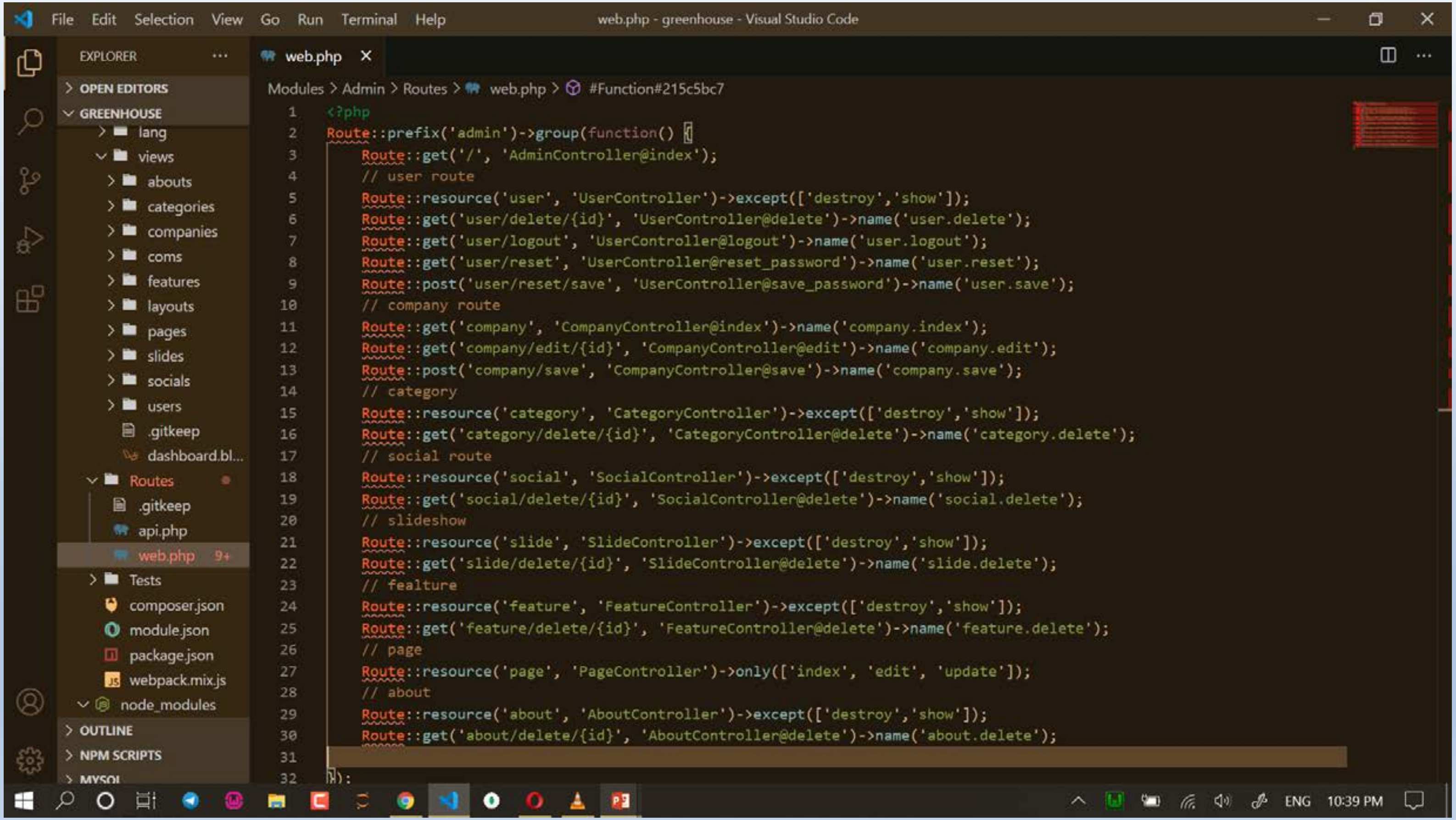
❑ Step for creating collections:

1. Route
2. Controller
3. Construct
4. Index
5. Create
6. Upload file
7. Store Data
8. Edit
9. Alert

Create User, Category, Company, Social, and About below link:

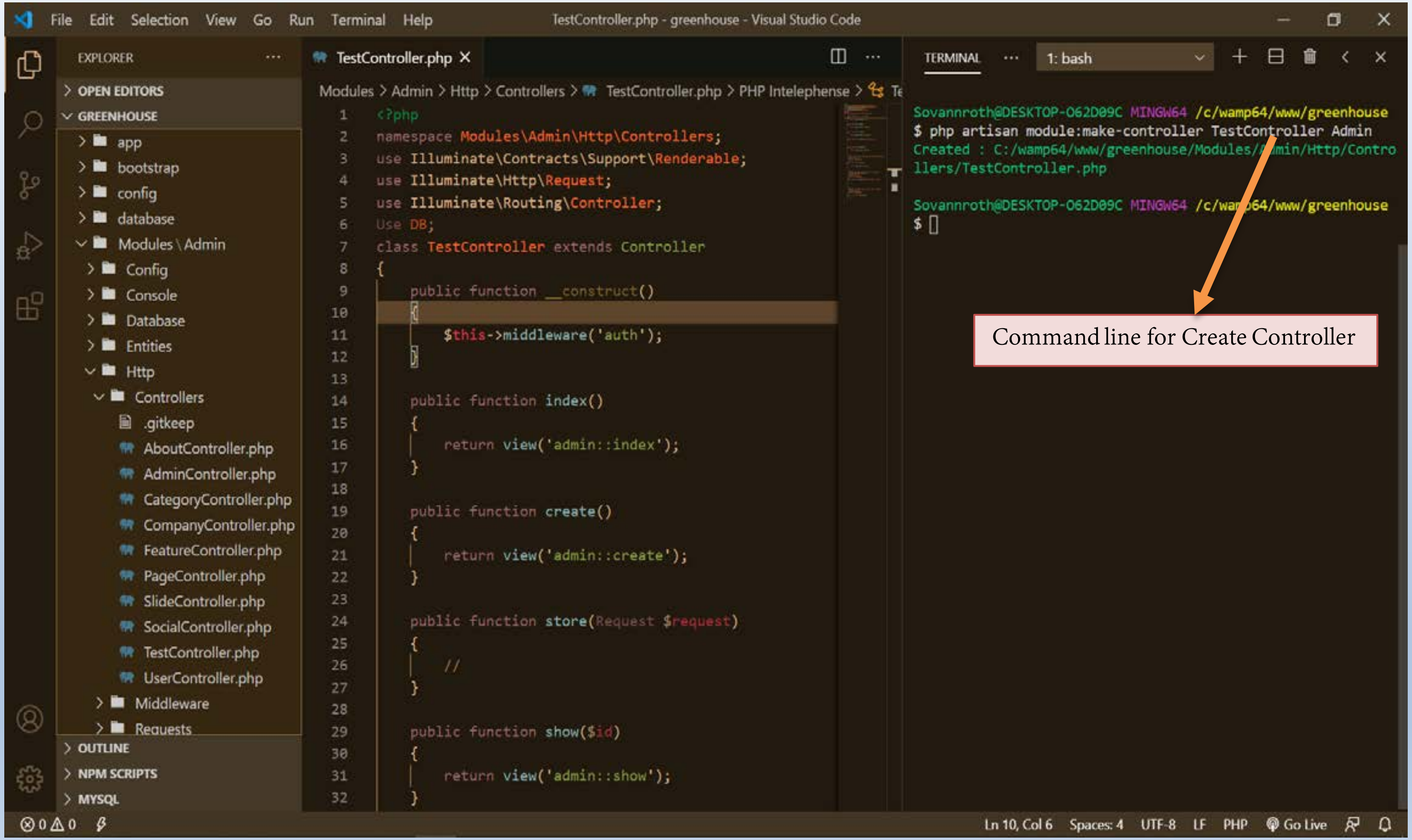
https://github.com/Chanthan89/Smart-Irrigation/tree/master/Weekly%20Presentation/Sept-2020/VANNAK%20Sovannroth

❖ Creating Route of User, Category, Company, Social, and About.



```php
<?php
Route::prefix('admin')->group(function() {
    Route::get('/', 'AdminController@index');
    // user route
    Route::resource('user', 'UserController')->except(['destroy','show']);
    Route::get('user/delete/{id}', 'UserController@delete')->name('user.delete');
    Route::get('user/logout', 'UserController@logout')->name('user.logout');
    Route::get('user/reset', 'UserController@reset_password')->name('user.reset');
    Route::post('user/reset/save', 'UserController@save_password')->name('user.save');
    // company route
    Route::get('company', 'CompanyController@index')->name('company.index');
    Route::get('company/edit/{id}', 'CompanyController@edit')->name('company.edit');
    Route::post('company/save', 'CompanyController@save')->name('company.save');
    // category
    Route::resource('category', 'CategoryController')->except(['destroy','show']);
    Route::get('category/delete/{id}', 'CategoryController@delete')->name('category.delete');
    // social route
    Route::resource('social', 'SocialController')->except(['destroy','show']);
    Route::get('social/delete/{id}', 'SocialController@delete')->name('social.delete');
    // slideshow
    Route::resource('slide', 'SlideController')->except(['destroy','show']);
    Route::get('slide/delete/{id}', 'SlideController@delete')->name('slide.delete');
    // fealture
    Route::resource('feature', 'FeatureController')->except(['destroy','show']);
    Route::get('feature/delete/{id}', 'FeatureController@delete')->name('feature.delete');
    // page
    Route::resource('page', 'PageController')->only(['index', 'edit', 'update']);
    // about
    Route::resource('about', 'AboutController')->except(['destroy','show']);
    Route::get('about/delete/{id}', 'AboutController@delete')->name('about.delete');

});
```

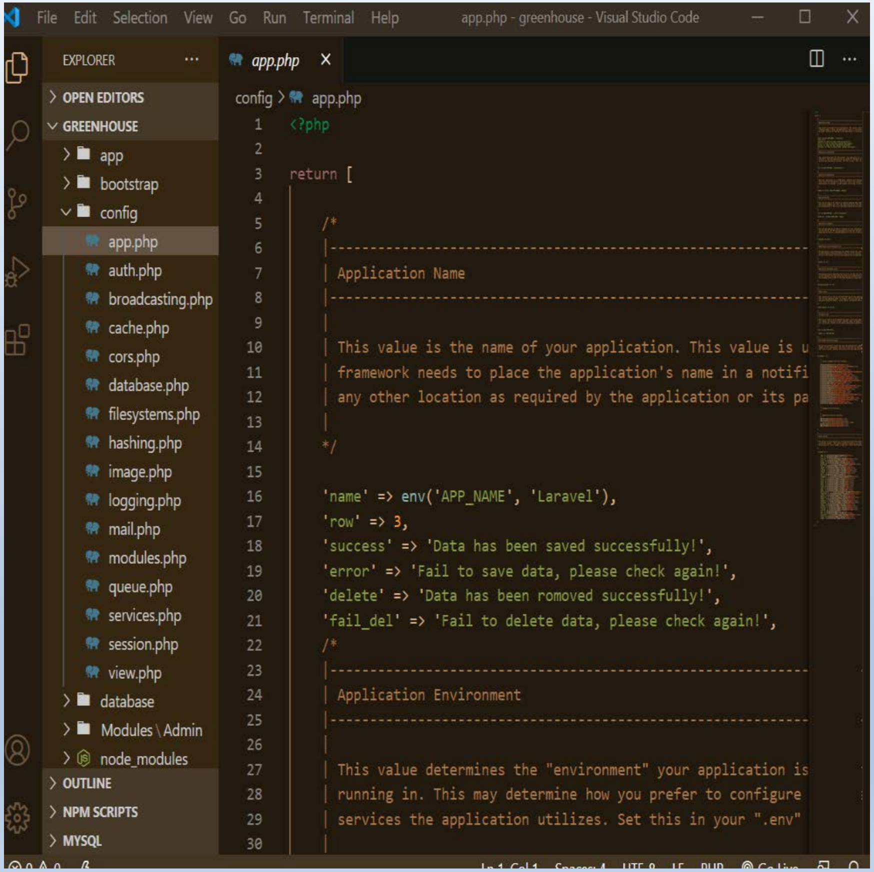
❖ Creating Controller for collections



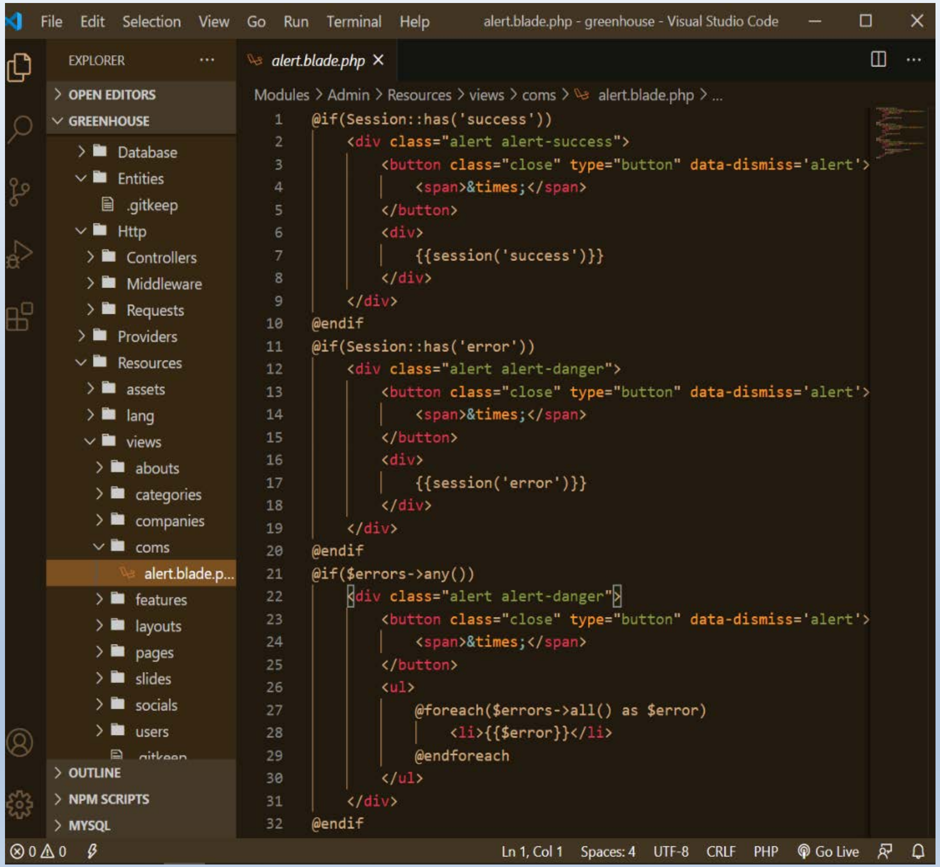
```php
<?php
namespace Modules\Admin\Http\Controllers;
use Illuminate\Contracts\Support\Renderable;
use Illuminate\Http\Request;
use Illuminate\Routing\Controller;
Use DB;
class TestController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        return view('admin::index');
    }

    public function create()
    {
        return view('admin::create');
    }

    public function store(Request $request)
    {
        //
    }

    public function show($id)
    {
        return view('admin::show');
    }
```

```
Sovannroth@DESKTOP-062D09C MINGW64 /c/wamp64/www/greenhouse
$ php artisan module:make-controller TestController Admin
Created : C:/wamp64/www/greenhouse/Modules/Admin/Http/Contro
llers/TestController.php

Sovannroth@DESKTOP-062D09C MINGW64 /c/wamp64/www/greenhouse
$
```

❖ Creating Alert Message



```
<?php

return [

    /*
    |--------------------------------------------------------------------------
    | Application Name
    |--------------------------------------------------------------------------
    |
    | This value is the name of your application. This value is u
    | framework needs to place the application's name in a notifi
    | any other location as required by the application or its pa
    |
    */

    'name' => env('APP_NAME', 'Laravel'),
    'row' => 3,
    'success' => 'Data has been saved successfully!',
    'error' => 'Fail to save data, please check again!',
    'delete' => 'Data has been romoved successfully!',
    'fail_del' => 'Fail to delete data, please check again!',
    /*
    |--------------------------------------------------------------------------
    | Application Environment
    |--------------------------------------------------------------------------
    |
    | This value determines the "environment" your application is
    | running in. This may determine how you prefer to configure
    | services the application utilizes. Set this in your ".env"
    |
```



```
@if(Session::has('success'))
    <div class="alert alert-success">
        <button class="close" type="button" data-dismiss='alert'>
            <span>&times;</span>
        </button>
        <div>
            {{session('success')}}
        </div>
    </div>
@endif
@if(Session::has('error'))
    <div class="alert alert-danger">
        <button class="close" type="button" data-dismiss='alert'>
            <span>&times;</span>
        </button>
        <div>
            {{session('error')}}
        </div>
    </div>
@endif
@if($errors->any())
    <div class="alert alert-danger">
        <button class="close" type="button" data-dismiss='alert'>
            <span>&times;</span>
        </button>
        <ul>
            @foreach($errors->all() as $error)
                <li>{{$error}}</li>
            @endforeach
        </ul>
    </div>
@endif
```

Thank You!