Steps:

* Install laravel
* Create datatabse
* Setup datatabse to laravel .env file
* Step 2: need full users system
* Create page table with model: php artisan make:model Page –m
* Install spatie permission for laravel: <https://spatie.be/docs/laravel-permission/v4/installation-laravel>
* Command: composer require spatie/laravel-permission
* manually add the service provider in your config/app.php file:
* Add Spatie\Permission\PermissionServiceProvider::class, inside providers array
* Publish migrations and config to:
* Command: php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"
* Add the Spatie\Permission\Traits\HasRoles trait to your User model(s):
* use Spatie\Permission\Traits\HasRoles; at the top uses
* use HasRoles; inside User class
* Reference: <https://spatie.be/docs/laravel-permission/v4/basic-usage/basic-usage>
* Migrate tables: php artisan migrate
* Create default data seeder
* php artisan make:seeder DefaultDataSeeder
* truncate basic tables
* Create Pages table data
* Create Permissions table data
* Create Roles table data & give permission to roles
* Create Users table data & give role to users
* Add default data seeder link to DatabaseSeeder.php
* Run command: php artisan db:seed
* Install laravel/ui package for default authentication system
* Package document: <https://github.com/laravel/ui>
* Once the laravel/ui package has been installed, you may install the frontend scaffolding using the ui Artisan command:
* Command: php artisan ui bootstrap –auth

Template Mastering:

* Create frontend and backend folder inside public/dist for asset files
* Create frontend and backend folder inside views for frontend and backend blade files
* Create frontend.blade.php and backend.blade.php master file inside layouts folder created by laravel/ui package
* Go to App\Http\Controllers\Auth\LoginController.php
* Make username as username by putting following function inside LoginController

public function **username**()

    {

        return 'username';

    }

* Doc: <https://laraveldaily.com/auth-login-with-username-instead-of-email/>
* For validation error changed use following function inside LoginController.php

public function **validateLogin**(**Request** $request)

    {

        $request->**validate**([

            'username' => 'required|string',

            'password' => 'required|string',

        ],[

            'username.required' => 'দয়া করে ইউজার নেম লিখুন।' ,

            'password.required' => 'দয়া করে পাসওয়ার্ড লিখুন।'

        ]);

    }

* Disable register route on web.php
* Create two route group with prefix (url prefix)and as (name) and create route for student and admin
* On LoginController inside App\Http\Controllers\Auth disable line protected $redirectTo = RouteServiceProvider::HOME;
* Create a function redirectTo for redirecting user dashboard based on their role

*/\*\**

*\* Redirectiing to user dashboard based on role.*

*\**

*\* @return***route**

*\*/*

    public function **redirectTo**() {

        if(**auth**()->**user**()->**hasrole**('student'))

        {

            return 'myaccount/dashboard';

        }else{

            return 'systemcontrol/dashboard';

        }

    }

* Go to App\Http\Middleware\RedirectIfAuthenticated middleware and apply logic for redirection if user authenticated.

*//APPLY HERE*

                if(**auth**()->**user**()->**hasrole**('student'))

                {

                    return **redirect**('myaccount/dashboard');

                }else{

                    return **redirect**('systemcontrol/dashboard');

                }

* Create isAdmin middleware to check if user is allowed to visit admin area.
* Command: php artisan make:middleware isAdmin
* Find newly created middleware to : App\Http\Middleware
* Give it logic to check user role and if fail abourt to code 401
* Register newly created middleware to kernel.php inside App\Http on line $routeMiddleware
* Put middleware on web.php route group for admin