

Laravel Topics & Features Need for Development



Table of Contents

| SL | Chapter | Sub Chapter/ Description | Status | Page |
|----|---|--|---------|------|
| 01 | Backend - Authentication (Custom) | a. Registration | Done | |
| | | b. Login and logout | Done | |
| | | c. Profile info update | Done | |
| | | d. Profile password change | Done | |
| | | e. Forget and reset password | Done | |
| | | f. Prevent browser back button after logout | Done | |
| | | g. Prevent Sweet alert after pressing back button | pending | |
| 02 | Backend - Multi authentication (Custom) | a. Setup | Done | |
| | | b. Multi Registration | Done | |
| | | c. Guard Configuration | Done | |
| | | d. Login and Logout with guard | Done | |
| | | e. Multi Dashboard with Middleware protection | Done | |
| | | f. Profile info update | | |
| | | g. Profile password change | | |
| | | h. Forget and reset password | | |
| | | i. Prevent browser back button after logout | | |
| 03 | Mailing | Sending Gmail to users | Done | |
| 04 | Localization | Bangla - English language support | | |
| 05 | Image Show | Image show with instant upload by jquery | Done | |
| 06 | Multi Image CRUD | Multiple image CRUD with other contents | | |
| 07 | Multi Dependent Dropdown | Multi dependent dropdown select with ajax data fetch and select2 | Done | |
| 08 | Multiple Select 2 options | Just like multiple tags select | Done | |

| | | | | |
|----|---------------------------------|---|------|--|
| 09 | Google Recaptcha | Someone need to ensure that he is robot or not | | |
| 10 | Google Authentication Sign In | Someone need to google authentication for login or comment | | |
| 11 | QR Generate | | | |
| 12 | PDF Generate | | | |
| 13 | Sweer Alert & Toaster Message | | Done | |
| 14 | Comment and Replay Option | | | |
| 15 | Find and Facts | Find and Facts measuring previous month | | |
| 15 | Global Search | Global search option with desire keywords. | | |
| 16 | SMS Gateway | SMS sending to users mobile | | |
| 17 | Payment Gateway | Payment gateway integration like bkaash, visa, paypal, strip etc | | |
| 18 | Notification | Real time notification send to admin or users dashboard. | | |
| 19 | Passport Authentication for API | A. Installation and Setup | Done | |
| | | B. Some SS That I might need | Done | |
| 20 | Spatie Role Management | Role management for users and set permissions into the features with Spatie package | | |
| 21 | How to create a Helper Function | | Done | |
| 22 | Summernote or Ck Editor use | | | |
| 23 | Datatable use with button | | Done | |
| 24 | Laravel Task Scheduling | | | |
| 25 | Yajra Data Table Complete CRUD | | Done | |

Chapter - 01

Backend - Authentication (Custom)

A. Registration

First part of authentication is registration. You need to register a user to manage your admin panel. There are many laravel authentication packages (Jetstream, Breez, Laravel UI) to do authentication tasks but we want to authenticate our system by custom coding. For custom Registration:

Step: 1

Install Laravel with all the necessary packages, connect with the database, and migrate the tables.

Step: 2

You need to add a user in the users table. So we will do that with seeder. In the command line write:

```
php artisan make:seeder AdminUserSeeder
```

Then go to database / seeders / AdminUserSeeder.php and write this code:

```
User::create([
    'name' => 'your name',
    'email' => 'yourmail@gmail.com',
    'password' => Hash::make(your password),
]);
```

Step: 3

Now go to database / seeders / DatabaseSeeder.php and write this code:

```
$this->call([
    AdminUserSeeder::class,
]);
```

Step: 4

Now write this command in terminal

```
php artisan db:seed
```

Then the seeding will be successful and a data will store in the users table. With this you can login

A. Login and Logout

Step: 1

Create a blade file first and write the necessary codes. Here i am creating a blade file in resources / views / admin / login / login.blade.php

```
<form action="{{ route('post.login') }}" method="POST">
    @csrf
    <div class="form-floating mb-20px">
        <input type="email" name="email" class="form-control fs-13px h-45px border-0" placeholder="Email Address" id="emailAddress" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600 fs-13px">Email Address</label>
    </div>
    <div class="form-floating mb-20px">
        <input type="password" name="password" class="form-control fs-13px h-45px border-0" placeholder="Password" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600 fs-13px">Password</label>
    </div>
    <div class="form-check mb-20px">
        <input class="form-check-input border-0" type="checkbox" value="1" id="rememberMe" />
        <label class="form-check-label fs-13px text-gray-500" for="rememberMe">Remember Me</label>
    </div>
    <a href="{{ route('forget.password.get') }}" class="mx-3 float-end">Forget Password?</a>
</div>
<div class="mb-20px">
    <button type="submit" class="btn btn-outline-danger d-block w-100 h-45px btn-lg">Log In</button>
</div>
<div class="text-blue-500">
    <a href="">Forget Password?</a>
</div>
</form>
```

Step: 2

Then You need to create routes. Go to routes / web.php

```
Route::get('/admin-login', [AdminController::class, 'index'])->name('admin.login');
Route::post('/post-login', [AdminController::class, 'login'])->name('post.login');
```

Step: 3

Create a Controller called AdminController. For that goto terminal and write
php artisan make:controller Admin/AdminController --resource

Step: 4

Go to the controller app/Http/Controllers/Admin/AdminController and write this code:

```
<?php
```

```
namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use RealRashid\SweetAlert\Facades\Alert;

class AdminController extends Controller
{
    public function index()
    {
        if(Auth::check()){
            return redirect()->route('dashboard');
        }
        else{
            return view('admin.login.login');
        }
    }

    public function create()
    {
        //
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email:rfc,dns',
            'password' => 'required',
        ]);

        $credentials = $request->only('email', 'password');
        if(Auth::attempt($credentials)){
            Alert::success('Welcome!', 'Login Successfully..');
            return redirect()->route('dashboard');
        }
        else{
            Alert::warning('Login Failed!', 'Credentials Does not Matched..');
            return redirect('/admin-login');
        }
    }
}
```

Logout

Step: 1

Create a Controller called AdminController. For that goto terminal and write
 php artisan make:controller Admin/AdminController --resource

And write this code:

```
public function logout(Request $request){
    Auth::logout();
    Alert::success('Logout Successfully Done..');
    return redirect('/admin-login');
}
```

Step: 2

Write this route in web.php

```
Route::post('/admin-logout', [AdminController::class, 'logout'])->name('admin.logout');
```

Step: 3

Write this code in your blade file:

```
<a href="" onclick="event.preventDefault(); document.getElementById('admin-logout').submit();"
class="dropdown-item">Log Out</a>
<form action="{{ route('admin.logout') }}" method="POST" id="admin-logout">
    @csrf
</form>
```


B. Profile Info Update

Step: 1

In the Blade file, resources / views / admin/ layout / master

```
<a href="{{ route('profile', Auth::user()->id) }}" class="dropdown-item">Edit Profile</a>
```

Step: 2

Create routes in web.php

```
// Profile Update
Route::get('/profile/{id}', [ProfileController::class, 'index'])->name('profile');
Route::post('/profile/update/{id}', [ProfileController::class,
'update'])->name('profile.update');
```

Step: 3

Create Controllers named ProfileController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\File;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;
use RealRashid\SweetAlert\Facades\Alert;

class ProfileController extends Controller
{

    public function index($id)
    {
```

```

    $profile = User::findOrFail($id);
    return view('admin.profile.edit', compact('profile'));
}

```

```

public function update(Request $request, $id)
{
    $profile = User::findOrFail($id);

    $request->validate([
        'image' => 'nullable|mimes:jpeg,png,jpg,gif,svg|max:2048',
        'name' => 'required|min:3|max:50',
        'email' => 'required|email:rfc,dns|email|unique:users,email,'.$request->user()->id,
        'mobile' => 'required|min:11|max:11|regex:/^(01)[0-9]{9}/',
    ]);

    try {
        DB::beginTransaction();

        // IMAGE SECTION
        $image = $request->file('image');
        if($image != null){
            if($profile->image != null){
                unlink(public_path($profile->image));
            }
            $name = Str::random(5).'.'.$image->getClientOriginalExtension();
            $destinationPath = public_path('/uploads/profile');
            if (!File::exists($destinationPath)) {
                File::makeDirectory($destinationPath, 0777, true);
            }
            Image::make($image->getRealPath())->resize(200, 200, function ($constraint)
            {
                $constraint->aspectRatio();
            })->save($destinationPath.'/'.$name);
            $destinationPath = 'uploads/profile/';
            $profile->image = $destinationPath.$name;
        }

        // IMAGE SECTION
    }
}

```

```
$profile->name = $request->name;
$profile->email = $request->email;
$profile->mobile = $request->mobile;
$profile->save();

DB::commit();
Alert::success('Admin Profile Updated Successfully..');
return back();
}
catch(\Exception $e){
    DB::rollBack();
    Alert::error($e->getMessage());
    return redirect()->back();
}
}
}
```

C. Profile Password Change

Step: 1

In the Blade file, resources / views / admin/ layout / master

```
<a href="{{ route('password', Auth::user()->id) }}" class="dropdown-item">Change Password</a>
```

Step: 2

Create routes in web.php

```
// Password Update
Route::get('/password/{id}', [ProfileController::class, 'passwordIndex'])->name('password');
Route::post('/password/update/{id}', [ProfileController::class, 'passwordUpdate'])->name('password.update');
```

Step: 3

Create Controllers named ProfileController

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\File;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;
use RealRashid\SweetAlert\Facades\Alert;

class ProfileController extends Controller
{
```

```

public function passwordIndex($id){
    $profile = User::findOrFail($id);
    return view('admin.profile.password', compact('profile'));
}

```

```

public function passwordUpdate(Request $request, $id){

    $profile = User::findOrFail($id);

    $request->validate([
        'current_password' => 'required|min:8|max:25',
        'password' => 'required|min:8|max:25|confirmed',
        'password_confirmation' => 'required|min:8|max:25',
    ], [
        'password.required' => 'New password field is required',
        'password.min' => 'New password should at least 8 words long'
    ]);

    try{
        if(Hash::check($request->current_password, $profile->password)){
            $profile->password = bcrypt($request->password);
            $profile->save();
            Auth::logout();
            return redirect()->route('admin.login');
        }
        else {
            Alert::warning("Your Current Password Do not Matched!!");
            return redirect()->back();
        }
    }
    catch (Exception $ex){
        Alert::error($ex->getMessage());
        return redirect()->back();
    }
}

```

Step: 4

Create a blade file password.blade.php in resources / views / admin / profile / password.blade.php

```
@extends('admin.layout.master')
```

```
@section('title')
```

```
Laravel Features | Password Update
```

```
@endsection
```

```
@section('admin-content')
```

```
<style>
```

```
input[type="number"]::-webkit-outer-spin-button,
```

```
input[type="number"]::-webkit-inner-spin-button {
```

```
-webkit-appearance: none;
```

```
margin: 0;
```

```
}
```

```
input[type="number"] {
```

```
-moz-appearance: textfield;
```

```
}
```

```
</style>
```

```
<div class="profile">
```

```
<div class="profile-header">
```

```
<div class="profile-header-cover"></div>
```

```
<div class="profile-header-content">
```

```
<div class="profile-header-img">
```

```

```

```
</div>
```

```
<div class="profile-header-info">
```

```

        <h4 class="mt-0 mb-1">{{ $profile->name }}</h4>
        <p class="mb-2">Back End Developer</p>
        <a href="#" class="btn btn-xs btn-yellow">Edit Profile</a>
    </div>

```

```

</div>

```

```

        <ul class="profile-header-tab nav nav-tabs">
            <li class="nav-item"><a href="#profile-post" class="nav-link active"
data-bs-toggle="tab"></a></li>
            <li class="nav-item"><a href="#profile-about" class="nav-link"
data-bs-toggle="tab"></a></li>
            <li class="nav-item"><a href="#profile-photos" class="nav-link"
data-bs-toggle="tab"></a></li>
            <li class="nav-item"><a href="#profile-videos" class="nav-link"
data-bs-toggle="tab"></a></li>
            <li class="nav-item"><a href="#profile-friends" class="nav-link"
data-bs-toggle="tab"></a></li>
        </ul>

```

```

</div>

```

```

<div class="profile-content">

```

```

    <div class="tab-content p-0">
        <div class="col-8 mx-auto">
            <form action="{{ route('password.update', $profile->id) }}" method="POST" >
                @csrf
                <div class="form-floating mb-20px">
                    <input type="password" name="current_password" id="currentPassField"
class="form-control fs-13px h-45px border-0" placeholder="Your Current Password" />
                    <label for="currentPassField" class="d-flex align-items-center
text-gray-600 fs-13px">Current Password</label>
                </div>

```

```

                <div class="form-floating mb-20px">
                    <input type="password" name="password" id="newPass"
class="form-control fs-13px h-45px border-0" placeholder="Your New Password" />

```

```

        <label for="newPass" class="d-flex align-items-center text-gray-600
fs-13px">New Password</label>
    </div>

```

```

    <div class="form-floating mb-20px">
        <input type="password" name="password_confirmation"
id="password_confirmation" class="form-control fs-13px h-45px border-0"
placeholder="Your Current Password" />
        <label for="password_confirmation" class="d-flex align-items-center
text-gray-600 fs-13px">Re-Type Password</label>
    </div>

```

```

    <div class="mb-20px">
        <button type="submit" class="btn btn-primary d-block w-100 h-45px
btn-lg">Update Password</button>
    </div>
    <div class="text-gray-500">

```

```

    </div>
</form>
</div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

@endsection

```


D. Forget and Reset Password

Reference Link:

<https://www.itsolutionstuff.com/post/laravel-custom-forgot-reset-password-exampleexample.html>

Step: 1

First you need a migration table. Here i am using 'password_reset' which is already there when you migrate you tables for the first time. The migration file look like:

```
public function up()
{
    Schema::create('password_resets', function (Blueprint $table) {
        $table->string('email')->index();
        $table->string('token');
        $table->timestamp('created_at')->nullable();
    });
}
```

```
public function down()
{
    Schema::dropIfExists('password_resets');
}
```

Then migrate the table.(php artisan migrate)

Step: 2

Create routes in web.php

```
// Forget & Reset Password Route
Route::get('forget-password', [ForgotPasswordController::class,
'showForgotPasswordForm']->name('forget.password.get'));
Route::post('forget-password', [ForgotPasswordController::class,
'submitForgotPasswordForm']->name('forget.password.post'));
Route::get('reset-password/{token}', [ForgotPasswordController::class,
'showResetPasswordForm']->name('reset.password.get'));
Route::post('reset-password', [ForgotPasswordController::class,
'submitResetPasswordForm']->name('reset.password.post'));
```

Step: 3

Create Controller named ForgotPasswordController and write this codes:

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Carbon;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Mail;
use Illuminate\Support\Str;

class ForgetPasswordController extends Controller
{

    public function showForgetPasswordForm()
    {
        return view('admin.forget-password.forgetPassword');
    }

    public function submitForgetPasswordForm(Request $request)
    {
        $request->validate([
            'email' => 'required|email|exists:users',
        ], [
            'email.exists' => 'This E-mail Do Not Matched Our Records',
        ]);

        $token = Str::random(64);

        DB::table('password_resets')->insert([
            'email' => $request->email,
            'token' => $token,
            'created_at' => Carbon::now()
        ]);

        Mail::send('admin.forget-password.email.forgetPassword', ['token' => $token],
function($message) use($request){
    $message->to($request->email);

```

```

        $message->subject('Reset Password');
    });

    return back()->with('message', 'We have e-mailed your password reset link!');
}

public function showResetPasswordForm($token) {
    return view('admin.forget-password.forgetPasswordLink', ['token' => $token]);
}

public function submitResetPasswordForm(Request $request)
{
    $request->validate([
        'email' => 'required|email|exists:users',
        'password' => 'required|string|min:8|max:25|confirmed',
        'password_confirmation' => '|min:8|max:25'
    ],
    [
        'email.exists' => 'This E-mail Do Not Matched Our Records',
        'password.required' => 'New password field is required',
        'password.min' => 'New password should at least 8 words long'
    ]);

    $updatePassword = DB::table('password_resets')
        ->where([
            'email' => $request->email,
            'token' => $request->token
        ])
        ->first();

    if(!$updatePassword){
        return back()->withInput()->with('error', 'Invalid token!');
    }

    $user = User::where('email', $request->email)
        ->update(['password' => Hash::make($request->password)]);

    DB::table('password_resets')->where(['email'=> $request->email])->delete();

```

```
return redirect('/admin-login')->with('message', 'Your password has been
changed!');
}
```

```
}
```

Step: 4

Configure the email in .env file

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=465
MAIL_USERNAME=shykot1198@gmail.com
MAIL_PASSWORD=ctqvdey]***** [You need to provide here gmail's app
password not your gamil password ]
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS="shykot1198@gmail.com"
MAIL_FROM_NAME="{APP_NAME}"
```

Step: 5

Create blade files

First create resources / views / admin / forget-password / forgetPassword.blade.php

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="utf-8" />
  <title>Laravel Features | Forget Password</title>
  <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"
name="viewport" />
  <meta content="" name="description" />
  <meta content="" name="author" />
```

```
{{-- Favicon Icon --}}
<link rel="icon" type="image/png" sizes="32x32" href="{{asset('/')}}uploads/logo/laravel.png">
```

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet" />
<link href="{{ asset('/')}}backend/assets/css/vendor.min.css" rel="stylesheet" />
<link href="{{ asset('/')}}backend/assets/css/default/app.min.css" rel="stylesheet" />
```

```
<link href="{{ asset('/')}}backend/assets/css/font-awesome.min.css" rel="stylesheet" />
```

```

</head>
<body class='pace-top'>

<div id="loader" class="app-loader">
  <span class="spinner"></span>
</div>

<div id="app" class="app">

  <main class="login-form my-5">
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-md-8">
          <div class="card">
            <h3 class="card-header text-dark-700">Reset Password</h3>
            <div class="card-body">

              @if (Session::has('message'))
                <div class="alert alert-success" style="font-size: 14px;" role="alert">
                  {{ Session::get('message') }}
                </div>
              @endif

              <blockquote class="text-warning-600" style="font-weight: bold;">Enter the valid email
address that you want to receive the reset link. Remember: The Email you are providing must have to
registered in our system.</blockquote>

              <form action="{{ route('forget.password.post') }}" method="POST">
                @csrf
                <div class="form-group row pt-3">
                  <label for="email_address" class="col-md-4 col-form-label text-xl-start">E-Mail
Address</label>
                  <div class="col-md-6">
                    <input type="email" id="email_address" class="form-control mb-2"
placeholder="Enter Your Email " name="email" required autofocus>
                    @if ($errors->has('email'))
                      <span class="text-danger-600 pt-2">{{ $errors->first('email') }}</span>
                    @endif
                  </div>
                </div>
                <div class="col-md-6 offset-md-4 ">
                  <button type="submit" class="btn btn-primary mt-2">
                    Send Password Reset Link
                  </button>
                </div>
              </form>

```

```

        </div>
    </div>
</div>
</div>
</div>
</main>

```

```

<a href="javascript:;" class="btn btn-icon btn-circle btn-success btn-scroll-to-top"
data-toggle="scroll-to-top"><i class="fa fa-angle-up"></i></a>

```

```

</div>

```

```

<script src="{{ asset('/') }}backend/assets/js/vendor.min.js"
type="a0616a96a228706fff8d35bc-text/javascript"></script>
<script src="{{ asset('/') }}backend/assets/js/app.min.js"
type="a0616a96a228706fff8d35bc-text/javascript"></script>

```

```

@include('sweetalert::alert', ['cdn' => "https://cdn.jsdelivr.net/npm/sweetalert2@9"])

```

```

<script src="{{ asset('/') }}backend/assets/js/demo/login-v2.demo.js"
type="a0616a96a228706fff8d35bc-text/javascript"></script>

```

```

<script type="a0616a96a228706fff8d35bc-text/javascript">
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','../../../../../www.google-analytics.com/analytics.js','ga');

```

```

    ga('create', 'UA-53034621-1', 'auto');
    ga('send', 'pageview');

```

```

</script>
<script src="{{ asset('/') }}backend/assets/js/rocket-loader.min.js"
data-cf-settings="a0616a96a228706fff8d35bc-j49" defer=""></script><script defer
src="https://static.cloudflareinsights.com/beacon.min.js/v652eace1692a40cfa3763df669d7439c16390797
17194"
integrity="sha512-Gi7xpJR8tSkrpF7aordPZQIw2DLtzUIZcumS8dMQjwDHEnw9I7ZLYiOj/6tZStRBGtGgN
6ceN6cMH8z7etPGLw=="
data-cf-beacon="{\"rayId\":\"714575485f7778b7\", \"version\":\"2021.12.0\", \"r\":1, \"token\":\"4db8c6ef997743fda032
d4f73cfeff63\", \"si\":100}" crossorigin="anonymous"></script>
</body>

```

```

</html>

```

Secondly, in resources / views / admin / forget-password / forgetPasswordLink.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Laravel Features | Reset Password</title>
  <meta content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"
name="viewport" />
  <meta content="" name="description" />
  <meta content="" name="author" />

  {{-- Favicon Icon --}}
  <link rel="icon" type="image/png" sizes="32x32" href="{{asset('/')}}uploads/logo/laravel.png">

  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet" />
  <link href="{{ asset('/') }}backend/assets/css/vendor.min.css" rel="stylesheet" />
  <link href="{{ asset('/') }}backend/assets/css/default/app.min.css" rel="stylesheet" />

  <link href="{{ asset('/') }}backend/assets/css/font-awesome.min.css" rel="stylesheet" />

</head>
<body class="pace-top">

<div id="loader" class="app-loader">
  <span class="spinner"></span>
</div>

<div id="app" class="app">

  <main class="login-form my-5">
    <div class="container">
      <div class="row justify-content-center">
        <div class="col-md-8">
          <div class="card">
            <h3 class="card-header">Update Password</h3>
            <div class="card-body">

              <form action="{{ route('reset.password.post') }}" method="POST">
                @csrf
                <input type="hidden" name="token" value="{{ $token }}">
```

```

        <div class="form-group row my-2">
            <label for="email_address" class="col-md-4 col-form-label text-md-right">E-Mail
Address</label>
            <div class="col-md-6">
                <input type="text" id="email_address" class="form-control" name="email"
required autofocus>
                @if ($errors->has('email'))
                    <span class="text-danger">{{ $errors->first('email') }}</span>
                @endif
            </div>
        </div>

        <div class="form-group row my-2">
            <label for="password" class="col-md-4 col-form-label
text-md-right">Password</label>
            <div class="col-md-6">
                <input type="password" id="password" class="form-control" name="password"
required autofocus>
                @if ($errors->has('password'))
                    <span class="text-danger">{{ $errors->first('password') }}</span>
                @endif
            </div>
        </div>

        <div class="form-group row my-2">
            <label for="password-confirm" class="col-md-4 col-form-label
text-md-right">Confirm Password</label>
            <div class="col-md-6">
                <input type="password" id="password-confirm" class="form-control"
name="password_confirmation" required autofocus>
                @if ($errors->has('password_confirmation'))
                    <span class="text-danger">{{ $errors->first('password_confirmation') }}</span>
                @endif
            </div>
        </div>

        <div class="col-md-6 offset-md-4">
            <button type="submit" class="btn btn-primary">
                Reset Password
            </button>
        </div>
    </form>

</div>
</div>
</div>
</div>
</div>
</main>

```



```
<a href="javascript:;" class="btn btn-icon btn-circle btn-success btn-scroll-to-top" data-toggle="scroll-to-top"><i class="fa fa-angle-up"></i></a>
```

```
</div>
```

```
<script src="{{ asset('/') }}backend/assets/js/vendor.min.js" type="a0616a96a228706fff8d35bc-text/javascript"></script>
<script src="{{ asset('/') }}backend/assets/js/app.min.js" type="a0616a96a228706fff8d35bc-text/javascript"></script>
```

```
@include('sweetalert::alert', ['cdn' => "https://cdn.jsdelivr.net/npm/sweetalert2@9"])
```

```
<script src="{{ asset('/') }}backend/assets/js/demo/login-v2.demo.js" type="a0616a96a228706fff8d35bc-text/javascript"></script>
```

```
<script type="a0616a96a228706fff8d35bc-text/javascript">
(function(i,s,o,g,r,a,m){i["GoogleAnalyticsObject"]=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date(),a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','../..../www.google-analytics.com/analytics.js','ga');
```

```
ga('create', 'UA-53034621-1', 'auto');
ga('send', 'pageview');
```

```
</script>
<script src="{{ asset('/') }}backend/assets/js/rocket-loader.min.js" data-cf-settings="a0616a96a228706fff8d35bc-|49" defer=""></script><script defer
src="https://static.cloudflareinsights.com/beacon.min.js/v652eace1692a40cfa3763df669d7439c16390797
17194"
integrity="sha512-Gi7xpJR8tSkrpF7aordPZQIw2DLtzUIZcumS8dMQjwDHEnw9I7ZLyOj/6tZStRBGtGgN
6ceN6cMH8z7etPGlw=="
data-cf-beacon="{\"rayId\":\"714575485f7778b7\", \"version\":\"2021.12.0\", \"r\":1, \"token\":\"4db8c6ef997743fda032
d4f73cfeff63\", \"si\":100}" crossorigin="anonymous"></script>
</body>
```

```
</html>
```

Thirdly, in resources / views / admin / forget-password / email / forgetPassword.blade.php

```
<h1>Forget Password Email</h1>
```

You can reset password from bellow link:

```
<a href="{{ route('reset.password.get', $token) }}">Reset Password</a>
```

E. Prevent Password Back Button After Logout

Reference Link: <https://www.w3adda.com/blog/laravel-prevent-browser-back-button-after-logout>

Step: 1

Create a middleware

```
1 php artisan make:middleware PreventBackHistory
```

Step: 2

Open up **PreventBackHistory.php** file in **app/Http/Middleware** folder and replace codes with the following codes below:

app/Http/Middleware/PreventBackHistory.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class PreventBackHistory {
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next) {
        $response = $next($request);

        return $response->header('Cache-Control','nocache, no-store, max-age=0,
must-revalidate')
            ->header('Pragma','no-cache')
            ->header('Expires','Sat, 26 Jul 1997 05:00:00 GMT');
    }
}
```

Step: 3

Register the middleware in `app/Http/Kernel.php`

```
protected $routeMiddleware = [
    ....

    'prevent-back-history' => \App\Http\Middleware\PreventBackHistory::class,
];
```

Step: 4

Add this middleware in your route in `web.php`

```
Route::prefix('admin')->middleware(['auth', 'prevent-back-history'])->group(
(function() {
    .....
    .....
}));
```


Chapter - 02

Backend - Multi Authentication (Custom)

Reference Link: <https://www.youtube.com/watch?v=XxCqj7JDBxg>

A. Setup

Step: 1

There we want to multi authentication of users and admin table. So User model and migration file will be by default. You need to create a Admin model and migration table. For that write this command:

```
php artisan make:model Admin -m
```

Step: 2

Then you need to write this code in your admin migration table:

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
    /**
```

```
     * Run the migrations.
```

```
     *
```

```
     * @return void
```

```
     */
```

```
    public function up()
```

```
    {
```

```
        Schema::create('admins', function (Blueprint $table) {
```

```
            $table->id();
```

```
            $table->string('name');
```

```
            $table->string('email')->unique();
```

```
            $table->timestamp('email_verified_at')->nullable();
```

```
            $table->string('password');
```

```
            $table->string('mobile')->nullable();
```

```
            $table->text('image')->nullable();
```

```
            $table->rememberToken();
```

```

        $table->timestamps();
    }
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('admins');
}
}

```

Step: 3

Then you need to write this code in your admin Model:

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class Admin extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $guard = 'admin';

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [

```

```

        'name',
        'email',
        'password',
    ],

    /**
     * The attributes that should be hidden for serialization.
     */
    /**
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     */
    /**
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

Step: 4

Then migrate the tables

```
php artisan migrate
```

B. Registration

Step: 1

For Registration we need to add data 2 of this table users and admins. We will do this process by seeding. So we need to create 2 seeder files. Write:

```
php artisan make:seeder UserSeeder
php artisan make:seeder AdminSeeder
```

The go to database / seeders you will find the 2 seeding files.

Write these codes in AdminSeeder.php

```
public function run()
{
    Admin::create([
        'name' => 'Super Admin',
        'email' => 'shykot1198@gmail.com',
        'password' => Hash::make('12345678'),
    ]);
}
```

Write these codes in UserSeeder.php

```
public function run()
{
    User::create([
        'name' => 'Shourav Hasan',
        'email' => 'shouravhasan03@gmail.com',
        'password' => Hash::make('aaaaaaaa'),
    ]);
}
```

Step: 2

Now go to database / seeders / DatabaseSeeder.php and write this code:

```
$this->call([
    AdminSeeder::class,
    UserSeeder::class,
]);
```

Step: 3

Now Seed this data in the terminal:

```
php artisan db:seed
```


Then the seeding will be successful and data will store in the users table and admins table. With this you can login.

C. Guard Configuration

Step: 1

Now we need to configure our guards. Go to config / auth.php

Add Guards,

```
'guards' => [  
    'web' => [  
        'driver' => 'session',  
        'provider' => 'users',  
    ],  
    'admin' => [  
        'driver' => 'session',  
        'provider' => 'admins',  
    ],  
],
```

Add Providers,

```
'providers' => [  
    'users' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\User::class,  
    ],  
    'admins' => [  
        'driver' => 'eloquent',  
        'model' => App\Models\Admin::class,  
    ],  
],
```

D. Login / Logout with Guard and Middleware Protection

Login for ADMIN

Step: 1

Create a blade file first and write the necessary codes. Here i am creating a blade file in resources / views / admin / login / login.blade.php

```
<form action="{{ route('admin.login.store') }}" method="POST">
    @csrf
    <div class="form-floating mb-20px">
        <input type="email" name="email" class="form-control fs-13px h-45px
border-0" placeholder="Email Address" id="emailAddress" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600
fs-13px">Email Address</label>
    </div>
    <div class="form-floating mb-20px">
        <input type="password" name="password" class="form-control fs-13px
h-45px border-0" placeholder="Password" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600
fs-13px">Password</label>
    </div>
    <div class="form-check mb-20px">
        <input class="form-check-input border-0" type="checkbox" value="1"
id="rememberMe" />
        <label class="form-check-label fs-13px text-gray-500" for="rememberMe">
            Remember Me
        </label>
        <a href="" class="mx-3 float-end">Forget Password?</a>
    </div>
    <div class="mb-20px">
        <button type="submit" class="btn btn-outline-danger d-block w-100 h-45px
btn-lg">Log In</button>
    </div>
    <div class="text-blue-500">
        <a href="">Forget Password?</a>
    </div>
</form>
```

Step: 2

Then You need to create routes. Go to routes / web.php

```
Route::get('/admin-login', [AdminLoginController::class, 'index'])->name('admin.login.form');
Route::post('/admin-login/store', [AdminLoginController::class, 'login'])->name('admin.login.store');
```

Step: 3

Create a Controller called AdminLoginController. For that goto terminal and write
php artisan make:controller Admin/AdminLoginController --resource

Step: 4

Go to the controller app/Http/Controllers/Admin/AdminLoginController and write this code:

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use RealRashid\SweetAlert\Facades\Alert;

class AdminLoginController extends Controller
{
    public function index()
    {
        if(Auth::guard('admin')->check()){
            return redirect()->route('admin.dashboard');
        }
        else{
            return view('admin.login.login');
        }
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email:rfc,dns',
            'password' => 'required',
        ]);
    }
}
```

```

    }
}

$credentials = $request->only('email' , 'password');
if(Auth::guard('admin')->attempt($credentials)){
    Alert::success('Welcome!', 'Login Successfully..');
    return redirect()->route('admin.dashboard');
}
else{
    Alert::warning('Login Failed!', 'Credentials Does not Matched..');
    return redirect('/admin-login');
}
}
}

```

Logout for ADMIN

Step: 1

Go to same controller and write this code:

```

public function logout(Request $request){
    Auth::guard('admin')->logout();
    Alert::success('Logout Successfully Done..');
    return redirect('/admin-login');
}
}

```

Step: 2

Write this route in web.php

```
Route::post('/admin-logout', [AdminLoginController::class, 'logout']->name('admin.logout');
```

Step: 3

Write this code in your blade file:

```

<a href="" onclick="event.preventDefault(); document.getElementById('admin-logout').submit();"
class="dropdown-item">Log Out</a>
<form action="{{ route('admin.logout') }}" method="POST" id="admin-logout">
    @csrf
</form>

```

Login for USER

Step: 1

Create a blade file first and write the necessary codes. Here i am creating a blade file in resources / views / users / login / login.blade.php

```
<form action="{{ route('user.login.store') }}" method="POST">
    @csrf
    <div class="form-floating mb-20px">
        <input type="email" name="email" class="form-control fs-13px h-45px
border-0" placeholder="Email Address" id="emailAddress" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600
fs-13px">Email Address</label>
    </div>
    <div class="form-floating mb-20px">
        <input type="password" name="password" class="form-control fs-13px
h-45px border-0" placeholder="Password" />
        <label for="emailAddress" class="d-flex align-items-center text-gray-600
fs-13px">Password</label>
    </div>
    <div class="form-check mb-20px">
        <input class="form-check-input border-0" type="checkbox" value="1"
id="rememberMe" />
        <label class="form-check-label fs-13px text-gray-500" for="rememberMe">
            Remember Me
        </label>
        <a href="" class="mx-3 float-end">Forget Password?</a>
    </div>
    <div class="mb-20px">
        <button type="submit" class="btn btn-outline-danger d-block w-100 h-45px
btn-lg">Log In</button>
    </div>
    <div class="text-blue-500">
        <a href="">Forget Password?</a>
    </div>
</form>
```

Step: 2

Then You need to create routes. Go to routes / web.php

```
Route::get('/login', [UserController::class, 'index'])->name('user.login.form');
Route::post('/login-store', [UserController::class, 'login'])->name('user.login.store');
```

Step: 3

Create a Controller called UserController. For that goto terminal and write
php artisan make:controller User/UserLoginController --resource

Step: 4

Go to the controller app/Http/Controllers/User/UserLoginController and write this code:

```
<?php

namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use RealRashid\SweetAlert\Facades\Alert;

class UserLoginController extends Controller
{

    public function index()
    {
        if(Auth::check()){
            return redirect()->route('user.dashboard');
        }
        else{
            return view('users.login.login');
        }
    }

    public function login(Request $request)
    {
        $request->validate([
            'email' => 'required|email:rfc,dns',
            'password' => 'required',
        ]);
    }
}
```

```

    }
    $credentials = $request->only('email', 'password');
    if(Auth::attempt($credentials)){
        Alert::success('Welcome!', 'Login Successfully..');
        return redirect()->route('user.dashboard');
    }
    else{
        Alert::warning('Login Failed!', 'Credentials Does not Matched..');
        return redirect('/login');
    }
}
}

```

Logout for USER

Step: 1

Go to same controller and write this code:

```

public function logout(Request $request){
    Auth::logout();
    Alert::success('Logout Successfully Done..');
    return redirect('/login');
}

```

Step: 2

Write this route in web.php

```
Route::post('/user-logout', [UserController::class, 'logout']->name('user.logout');
```

Step: 3

Write this code in your blade file:

```

<a href="" onclick="event.preventDefault(); document.getElementById('admin-logout').submit();"
class="dropdown-item">Log Out</a>
<form action="{{ route('user.logout') }}" method="POST" id="admin-logout">
    @csrf
</form>

```

E. Multi Dashboard with Middleware Protection

Now, we need two dashboards for User and Admin to do their own things. We also need to protect 2 dashboards, so that they never conflict with each other and so need to secure these dashboards in such manners that Users never access Admin's dashboard and Admin never access User's dashboard. We will do it by using middleware. Here we will create 2 middlewares. In that middleware we will call our routes.

Step: 1

Create a middleware. Here I am creating a middleware named AdminAuthenticate for Admin's dashboard protection and other will be by default in Laravel called Authenticate.php. Write this command:

```
php artisan make:middleware AdminAuthenticate
```

Go to app / Http / Middleware and you will find 2 middlewares along with others.

We will work with 2 of these: a) AdminAuthenticate.php - For Admin protection

b) Authenticate.php - For user protection and it will be there by default when you install Laravel

Step: 2 (Configure the middleware)

Firstly, go to Authenticate.php which we are using for User protection. And write this code:

```
protected function redirectTo($request)
{
    if (! $request->expectsJson()) {
        return route('user.login.form');
    }
}
```

Then, go to AdminAuthenticate.php which we created and use for Admin protection and write this code:

```
public function handle(Request $request, Closure $next)
{
    if(Auth::guard('admin')->check()){
        return $next($request);
    }
    else{
        return redirect()->route('admin.login.form');
    }
}
```

So our middleware configuration is done and now we need to register these 2 middlewares.

Step: 3 (Register the middlewares)

Go to, app / Http / Kernel.php here you will find Authenticate Middleware which is registered by default names as auth.

After that write this:

```
'admin' => \App\Http\Middleware\AdminAuthenticate::class,
```

We registered AdminAuthenticate middleware as admin named.

Step: 4 (Define Routes)

Now go to web.php define routes with middleware for both admin and users like that.

```
Route::prefix('user')->middleware(['auth'])->group(function(){  
    // Dashboard  
    Route::get('/dashboard',[UserDashboardController::class, 'index'])->name('user.dashboard');  
});
```

```
Route::prefix('admin')->middleware(['admin'])->group(function(){  
    // Dashboard  
    Route::get('/dashboard',[AdminDashboardController::class, 'index'])->name('admin.dashboard');  
});
```

Chapter - 03

Mailing - Sending Gmail to Users

There are many ways to send a mail using gmail in laravel. I am going to show one of mine.

Step: 1 (Configure SMTP SERVER)

Go to .env file and configure in mail section

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=465
MAIL_USERNAME=shykot1198@gmail.com
MAIL_PASSWORD=ctq*****
MAIL_ENCRYPTION=ssl
MAIL_FROM_ADDRESS="shykot1198@gmail.com"
MAIL_FROM_NAME="${APP_NAME}"
```

In your MAIL_PASSWORD enter your gmail's app password. For generating a app password of your Gmail. Go to your google account. You need to enable 2 step verification then you can generate a app password. If you are unable to do that search in google or youtube.

Step: 2 (Create model and migration files)

If you want to store the mail informations that you sending to a user, then you need a model and migration file to store that. so here i created this things:

Migration:

```
Schema::create('send_gmails', function (Blueprint $table) {
    $table->id();
    $table->string('to');
    $table->string('subject');
    $table->longText('body');
    $table->timestamps();
});
```

Model:

Php artisan make:model SendGmail -m

Step: 3 (Create a controller and write this codes)

Php artisan make:controller Admin/SendingGmailController

Write these codes:

```
public function index(){
    return view('admin.send-gmail.index');
}

public function store(Request $request){
    $request->validate([
        'to' => 'required|email:rfc,dns',
        'subject' => 'required|min:5|max:255',
        'body' => 'required| min:5'
    ]);

    try{
        DB::beginTransaction();

        $email = new SendGmail();
        $email->to = $request->to;
        $email->subject = $request->subject;
        $email->body = $request->body;
        $email->save();

        Mail::send('admin.send-gmail.body', ['body' => $request->body], function($message)
        use($request){
            $message->to($request->to);
            $message->subject($request->subject);
        });

        DB::commit();
        Alert::success('Email Has Sent Successfully..');
        return back();
    }
    catch (\Exception $ex){
        DB::rollBack();
        Alert::error($ex->getMessage());
        return redirect()->back();
    }
}
```

Step: 4 (Create Routes)

```
// Sending Gmail
Route::get('/email', [SendingGmailController::class, 'index']->name('send.gmail'));
Route::post('/email/store', [SendingGmailController::class, 'store']->name('store.gmail'));
```

Step: 5 (Go to your admin.send-gmail.body blade file)

Design and configure your blade file.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title></title>
</head>
<body>
  <div class="login-container">

    <div class="login-header">
      <div class="brand">
        <div class="my-2">
          {!! $body !!}
        </div>

        <small style="color: grey; text-align: center;">Develop By Shykot Hasan
Shourav</small>
      </div>

    </div>
  </div>
</body>
</html>
```

Chapter - 05

Image Show - With Instant Upload by JQuery

This chapter means when i upload an image it will show instantly without submitting it. In simple word Image Preview. So lets go

HTML CODE:

```
<div class="form-floating mb-20px">
  <div class="controls">
    
  </div>
</div>
<div class="form-floating mb-20px">
  <input type="file" accept="image/*" name="image" class="form-control fs-13px h-45px
border-0" placeholder="Your Image" id="image" />
  <label for="image" class="d-flex align-items-center text-gray-600 fs-13px">Image</label>
</div>
```

SCRIPT CODE:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
integrity="sha512-894YE6QWD5159HgZOGReFYm4dnWc1Qt5NtYSaNcOP+u1T9qYdvdihz0P
PSiiqn/+3e7Jo4EaG7TubfWGUrMQ==" crossorigin="anonymous"
referrerpolicy="no-referrer"></script>

<script type="text/javascript">
  $(document).ready(function(){
    $('#image').change(function(e){
      var reader = new FileReader();
      reader.onload = function(e){
        $('#showImage').attr('src',e.target.result);
      }
      reader.readAsDataURL(e.target.files['0']);
    });
  });
</script>
```


Chapter - 07

Multi Dependent Dropdown Selectbox with ajax data fetch

For Reference you goto laravel features system, goto Dependent Selectbox > multidependent example..

This is like you will select first category, then your selected category related subcategory will appear. Then you will select subcategory, then your selected subcategory related sub subcategory will appear.

The form is titled "Create Multi Dependent SelectBox". It contains three main sections for selection:

- Select Category:** A dropdown menu. In the second screenshot, "Fashion" is selected.
- Select SubCategory:** A dropdown menu. In the second screenshot, it shows "-- Select SubCategory --" with a list of options: "Men's Fashion" and "Women's Fashion". In the third screenshot, "Men's Fashion" is selected.
- Select Sub SubCategory:** A dropdown menu. In the third screenshot, it shows "-- Select Sub SubCategory --" with a list of options: "Men's T-Shirt".

Below these sections are input fields for "Name" (with a placeholder "Enter Name") and "Tags". A "Status" dropdown is also present, with "Active" selected.

Like This.

For that you first need a full setup, model, migrations, controller and crud for category, subcategory and sub subcategory. Then need another model, controller and migration file for multidependent example crud and functionality.

Step: 1 (Create a Migration file)

Create a migration file, write this code and migrate it.

```
Schema::create('dependent_select_boxes', function (Blueprint $table) {
    $table->id();
    $table->integer('category_id');
    $table->integer('subcategory_id');
```



```

    $table->integer('sub_subcategory_id');
    $table->string('name');
    $table->string('tags');
    $table->tinyInteger('status')->default(1);
    $table->timestamps();
});

```

Step: 2 (Model and write this codes)

```

protected $fillable = [
    'name',
    'status',
    'tags',
    'category_id',
    'subcategory_id',
    'sub_subcategory_id',
];

public function category(){
    return $this->belongsTo(Category::class, 'category_id', 'id');
}

public function subCategory(){
    return $this->belongsTo(SubCategory::class, 'subcategory_id', 'id');
}

public function subSubCategory(){
    return $this->belongsTo(SubSubCategory::class, 'sub_subcategory_id', 'id');
}

```

Step: 3 (Create a controller and write this code)

```

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Category;
use App\Models\DependentSelectBox;
use App\Models\SubCategory;
use App\Models\SubSubCategory;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use RealRashid\SweetAlert\Facades\Alert;

class MultiDependentExample extends Controller
{

    public function index()
    {
        $categories = Category::where('status',1)->get();
    }
}

```

```

    $selectboxes = DependentSelectBox::with('category', 'subCategory',
'subSubCategory')->latest()->get();
    return view('admin.multi-dependent.example', compact('categories', 'selectboxes'));
}

public function subSubcategoryFetch($subcategory_id){
    $subSubcategoryFetch = SubSubCategory::where('subcategory_id',$subcategory_id)->get();
    return json_encode($subSubcategoryFetch);
}

public function store(Request $request)
{
    $request->validate([
        'category_id' => 'required',
        'subcategory_id' => 'required',
        'sub_subcategory_id' => 'required',
        'name' => 'required|min:3|max:50',
        'tags' => 'required',
        'status' => 'required',
    ], [
        'category_id.required' => 'Select a Category Please!',
        'subcategory_id.required' => 'Select a Sub Category Please!',
        'sub_subcategory_id.required' => 'Select a Sub SubCategory Please!',
        'name.required' => 'Give a Name',
        'tags.required' => 'Give One or Multiple Tags',
        'status.required' => 'Select Status',
    ]);

    try{
        DB::beginTransaction();

        $selectbox = new DependentSelectBox();
        $selectbox->category_id = $request->category_id;
        $selectbox->subcategory_id = $request->subcategory_id;
        $selectbox->sub_subcategory_id = $request->sub_subcategory_id;
        $selectbox->name = $request->name;
        $selectbox->tags = json_encode($request->tags);
        $selectbox->status = $request->status;
        $selectbox->save();
        DB::commit();
        Alert::success('Dependent Selectbox Created Successfully..');
        return redirect()->route('all.selectbox');
    }
    catch(\Exception $e){
        DB::rollBack();
        Alert::error($e->getMessage());
        return redirect()->back();
    }
}

```

```

public function edit($id)
{
    $selectbox = DependentSelectBox::with('category', 'subCategory',
'subSubCategory')->findOrFail($id);
    $selectboxes = DependentSelectBox::with('category', 'subCategory',
'subSubCategory')->latest()->get();

    $categories = Category::where('status',1)->get();
    $subcategories =
SubCategory::where('status',1)->where('category_id',$selectbox->category_id)->get();
    $subSubCategories =
SubSubCategory::where('status',1)->where('subcategory_id',$selectbox->subcategory_id)->get();

    return view('admin.multi-dependent.example', compact('categories','subcategories',
'subSubCategories','selectboxes','selectbox'));
}

public function update(Request $request, $id)
{
    $request->validate([
        'category_id' => 'required',
        'subcategory_id' => 'required',
        'sub_subcategory_id' => 'required',
        'name' => 'required|min:3|max:50',
        'tags' => 'required',
        'status' => 'required',
    ], [
        'category_id.required' => 'Select a Category Please!',
        'subcategory_id.required' => 'Select a Sub Category Please!',
        'sub_subcategory_id.required' => 'Select a Sub SubCategory Please!',
        'name.required' => 'Give a Name',
        'tags.required' => 'Give One or Multiple Tags',
        'status.required' => 'Select Status',
    ]);
    $selectbox = DependentSelectBox::with('category', 'subCategory',
'subSubCategory')->findOrFail($id);
    try{
        DB::beginTransaction();

        $selectbox->category_id = $request->category_id;
        $selectbox->subcategory_id = $request->subcategory_id;
        $selectbox->sub_subcategory_id = $request->sub_subcategory_id;
        $selectbox->name = $request->name;
        $selectbox->tags = json_encode($request->tags);
        $selectbox->status = $request->status;
        $selectbox->save();
        DB::commit();
        Alert::success('Dependent Selectbox Updated Successfully..');
    }
}

```

```

        return redirect()->route('all.selectbox');
    }
    catch(\Exception $e){
        DB::rollBack();
        Alert::error($e->getMessage());
        return redirect()->back();
    }
}

public function destroy($id)
{
    $selectbox = DependentSelectBox::findOrFail($id);
    $selectbox->delete();
    Alert::success('Dependent Selectbox Deleted Successfully..');
    return redirect()->route('all.selectbox');
}
}

```

Step: 4 (goto web.php and create these routes)

```

// Multiple Dependent Selectbox Example
Route::get('/example', [MultiDependentExample::class, 'index']->name('all.selectbox'));
Route::post('/example/store', [MultiDependentExample::class, 'store']->name('store.selectbox'));
Route::get('/example/edit/{id}', [MultiDependentExample::class, 'edit']->name('edit.selectbox'));
Route::post('/example/update/{id}', [MultiDependentExample::class, 'update']->name('update.selectbox'));
Route::get('/example/delete/{id}', [MultiDependentExample::class, 'destroy']->name('destroy.selectbox'));
Route::get('/sub-subcategory-fetch/{subcategory_id}', [MultiDependentExample::class, 'subSubcategoryFetch']->name('sub.subcategory.fetch'));

```

Step: 5 (Create a blade file [admin.multi-dependent.example])

```

@extends('admin.layout.master')

@section('title')
    Multi Dependent | Example
@endsection

@section('extra-css')

    <!--Data Table required files -->
    <link href="{{ asset('/') }}backend/assets/plugins/datatables.net-bs5/css/dataTables.bootstrap5.min.css"
    rel="stylesheet" />
    <link href="{{ asset('/') }}plugins/datatables.net-responsive-bs5/css/responsive.bootstrap5.min.css"
    rel="stylesheet" />
    <link href="{{
    asset('/') }}backend/assets/plugins/datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css"
    rel="stylesheet" />

```

```

<link href="{{
asset('/') }}backend/assets/plugins/datatables.net-colreorder-bs5/css/colReorder.bootstrap5.min.css"
rel="stylesheet" />
<link href="{{
asset('/') }}backend/assets/plugins/datatables.net-keytable-bs5/css/keyTable.bootstrap5.min.css"
rel="stylesheet" />
<link href="{{
asset('/') }}backend/assets/plugins/datatables.net-rowreorder-bs5/css/rowReorder.bootstrap5.min.css"
rel="stylesheet" />
<link href="{{ asset('/') }}backend/assets/plugins/datatables.net-select-bs5/css/select.bootstrap5.min.css"
rel="stylesheet" />

```

```

<!-- Select2 css-->
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css" rel="stylesheet" />

```

```
@endsection
```

```

@section('admin-content')
<ol class="breadcrumb float-xl-end">
<li class="breadcrumb-item"><a href="">Home</a></li>
<li class="breadcrumb-item">Multi Dependent</li>
<li class="breadcrumb-item active">Example</li>
</ol>

```

```

<div class="row my-5 ">
<div class="mx-auto p-4" style="border: 1px solid #801419; border-radius: 4px;">
<!-- EDIT / UPDATE SECTION -->
@if(Request::routeIs('edit.selectbox'))
<h1 class="page-header ">Edit Multi Dependent SelectBox </h1>
<form action="{{ route('update.selectbox',$selectbox->id) }}" method="POST">
@csrf
<div class="row mb-15px">
<div class="col-md-4">
<label class="form-label col-form-label ">Select Category</label>
<div class="">
<select name="category_id" id="" class="form-control mb-5px
js-example-basic-single">
<option value="" disabled selected>-- Select Category --</option>
@foreach($categories as $category)
<option value="{{ $category->id }}" {{ $category->id == $selectbox->category_id
? 'selected' : '' }}>{{ $category->category_name }}</option>
@endforeach
</select>
@if ($errors->has('category_id'))
<span class="text-red-500">{{ $errors->first('category_id') }}</span>
@endif
</div>
</div>

```

```

        <div class="col-md-4">
            <label class="form-label col-form-label ">Select SubCategory</label>
            <div class="">
                <select name="subcategory_id" id="" class="form-control mb-5px
js-example-basic-single">
                    @foreach($subcategories as $subcategory)
                        <option value="{{ $subcategory->id }}" {{ $subcategory->id ==
$selectbox->subcategory_id ? 'selected' : " " }} >{{ $subcategory->subcategory_name }}</option>
                    @endforeach
                </select>
                @if ($errors->has('subcategory_id'))
                    <span class="text-red-500">{{ $errors->first('subcategory_id') }}</span>
                @endif
            </div>
        </div>
        <div class="col-md-4">
            <label class="form-label col-form-label ">Select Sub SubCategory</label>
            <div class="">
                <select name="sub_subcategory_id" id="" class="form-control mb-5px
js-example-basic-single">
                    @foreach($subSubCategories as $subSubCategory)
                        <option value="{{ $subSubCategory->id }}" {{ $subSubCategory->id ==
$selectbox->sub_subcategory_id ? 'selected' : " " }} >{{ $subSubCategory->sub_subcategory_name
}}</option>
                    @endforeach
                </select>
                @if ($errors->has('sub_subcategory_id'))
                    <span class="text-red-500">{{ $errors->first('sub_subcategory_id') }}</span>
                @endif
            </div>
        </div>
    </div>
    <div class="row mb-15px">
        <div class="col-md-4">
            <label class="form-label col-form-label ">Name</label>
            <div class="">
                <input type="text" name="name" class="form-control mb-5px" placeholder="Enter
Name" value="{{ $selectbox->name }}" />
                @if ($errors->has('name'))
                    <span class="text-red-500">{{ $errors->first('name') }}</span>
                @endif
            </div>
        </div>
        <div class="col-md-4">
            <label class="form-label col-form-label ">Tags</label>
            <div class="">
                <select name="tags[]" multiple="multiple" id="tags" class="js-example-basic-multiple
form-control mb-5px" style="width: 100%; height:100%" >
                    @php

```



```

        <span class="text-red-500">{{ $errors->first('category_id') }}</span>
    @endif
</div>
</div>
<div class="col-md-4">
    <label class="form-label col-form-label">Select SubCategory</label>
    <div class="">
        <select name="subcategory_id" id="" class="form-control mb-5px
js-example-basic-single">
            {{-- <option value="" disabled selected>-- Select SubCategory
--</option>--}}

        </select>
        @if ($errors->has('subcategory_id'))
            <span class="text-red-500">{{ $errors->first('subcategory_id') }}</span>
        @endif
    </div>
</div>
<div class="col-md-4">
    <label class="form-label col-form-label">Select Sub SubCategory</label>
    <div class="">
        <select name="sub_subcategory_id" id="" class="form-control mb-5px
js-example-basic-single">
            {{-- <option value="" disabled selected>-- Select Sub
SubCategory --</option>--}}

        </select>
        @if ($errors->has('sub_subcategory_id'))
            <span class="text-red-500">{{ $errors->first('sub_subcategory_id') }}</span>
        @endif
    </div>
</div>
</div>

<div class="row mb-15px">
    <div class="col-md-4">
        <label class="form-label col-form-label">Name</label>
        <div class="">
            <input type="text" name="name" class="form-control mb-5px" placeholder="Enter
Name" />
            @if ($errors->has('name'))
                <span class="text-red-500">{{ $errors->first('name') }}</span>
            @endif
        </div>
    </div>
    <div class="col-md-4">
        <label class="form-label col-form-label">Tags</label>
        <div class="">

```



```

        <select name="tags[]" multiple="multiple" id="tags" class="js-example-basic-multiple
form-control mb-5px" style="width: 100%; height:100% ">

        </select>
        @if ($errors->has('tags'))
            <span class="text-red-500">{{ $errors->first('tags') }}</span>
        @endif
        {{-- <input name="tags[]" multiple="multiple" required id="tags"
class="js-example-basic-multiple form-control mb-5px" style="width: 100%; height:100% " />--}}

    </div>
</div>
<div class="col-md-4">
    <label class="form-label col-form-label ">Status</label>
    <div class="">
        <select name="status" id="" class="form-control mb-5px">
            <option value="1">Active</option>
            <option value="0">Deactive</option>
        </select>
        @if ($errors->has('status'))
            <span class="text-red-500">{{ $errors->first('status') }}</span>
        @endif
    </div>
</div>
</div>
<div class="row mb-15px">
    <label class="form-label col-form-label "></label>
    <div class="">
        <button type="submit" class="btn btn-info d-block w-100 h-45px btn-lg">Create</button>
    </div>
</div>
</form>
@endif
</div>
</div>

<!-- Table Section -->
<div class="row">
    <table id="data-table-combine" class="table table-striped table-bordered align-middle"
style="font-size: 14px;">
        <thead>
            <tr>
                <th>SL</th>
                <th>Category</th>
                <th>Sub Category</th>
                <th>Sub SubCategory</th>
                <th>Name</th>
                <th>Tags</th>
                <th>Status</th>
            </tr>
        </thead>
    </table>
</div>

```

```

        <th >Action</th>
    </tr>
</thead>
<tbody>
    @foreach($selectboxs as $item)
    <tr>
        <td>{{ $loop->iteration }}</td>
        <td>{{ $item->category->category_name }}</td>
        <td>{{ $item->subCategory->subcategory_name }}</td>
        <td>{{ $item->subSubCategory->sub_subcategory_name }}</td>
        <td>{{ $item->name }}</td>
        <td>
            {{ implode(', ', json_decode($item->tags)) }}
        </td>
        <td>{{ $item->status == 1 ? 'Active' : 'Deactive' }}</td>
        <td><a href="{{ route('edit.selectbox',$item->id) }}" class="btn btn-sm btn-primary " title="Edit this
data">
            Edit
        </a>
        <a href="{{ route('destroy.selectbox',$item->id) }}" type="button" class="btn btn-danger btn-sm
deleteConfirm" id="delete" title="Delete this data">
            Delete
        </a></td>
    </tr>
    @endforeach
</tbody>
</table>
</div>

@endsection

@section('extra-js')
    <script src="https://code.jquery.com/jquery-3.6.2.min.js"
integrity="sha256-2krYZKh//PcchRtd+H+VyyQoZ/e3EcrkxhM8ycwASPA="
crossorigin="anonymous"></script>

    <!--Data Table required files -->
    <script src="{{ asset('/') }}backend/assets/plugins/datatables.net/js/jquery.dataTables.min.js"></script>
    <script src="{{
asset('/') }}backend/assets/plugins/datatables.net-bs5/js/dataTables.bootstrap5.min.js"></script>
    <script src="{{
asset('/') }}backend/assets/plugins/datatables.net-responsive/js/dataTables.responsive.min.js"></script>
    <script src="{{
asset('/') }}backend/assets/plugins/datatables.net-responsive-bs5/js/responsive.bootstrap5.min.js"></script>
    <script src="{{
asset('/') }}backend/assets/plugins/datatables.net-colreorder/js/dataTables.colReorder.min.js"></script>

```

```

<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-colreorder-bs5/js/colReorder.bootstrap5.min.js"></script>
</script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-keytable/js/dataTables.keyTable.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-keytable-bs5/js/keyTable.bootstrap5.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-rowreorder/js/dataTables.rowReorder.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-rowreorder-bs5/js/rowReorder.bootstrap5.min.js"></script>
</script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-select/js/dataTables.select.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-select-bs5/js/select.bootstrap5.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/dataTables.buttons.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons-bs5/js/buttons.bootstrap5.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.colVis.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.flash.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.html5.min.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.print.min.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/pdfmake/build/pdfmake.min.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/pdfmake/build/vfs_fonts.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/jszip/dist/jszip.min.js"></script>

```

```

<!--Select2 JS-->
<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>

```

```

<!-- SCRIPT FOR DEPENDENT SELECT BOX -->
<script type="text/javascript">
    $(document).ready(function () {
        // FOR CATEGORY
        $('#select[name="category_id"]').on('change', function () {
            var category_id = $(this).val();
            if (category_id) {
                $.ajax({
                    url: "{{ url('admin/multi-dependent-selectbox/subcategory-by-category/') }}" + category_id,
                    type: "GET",
                    dataType: "json",
                    success: function (data) {
                        // console.log(data);
                    }
                });
            }
        });
    });

```

```

        $('#select[name="sub_subcategory_id"]').html("");
        $('#select[name="subcategory_id"]').empty();
        $('#select[name="subcategory_id"]').append('<option value="" disabled selected>-- Select SubCategory --</option>');
        $.each(data, function (keys, value) {
            $('#select[name="subcategory_id"]').append('<option value="" + value.id + "">' +
value.subcategory_name + '</option>');
        });
    }
    });
} else {
    $('#select[name="subcategory_id"]').empty();
}

});

// FOR SUBCATEGORY
$('#select[name="subcategory_id"]').on('change', function () {
    var subcategory_id = $(this).val();
    if (subcategory_id) {
        $.ajax({
            url: "{{ url('admin/multi-dependent-selectbox/sub-subcategory-fetch') }}" + subcategory_id,
            type: "GET",
            dataType: "json",
            success: function (data) {
                // console.log(data);

                $('#select[name="sub_subcategory_id"]').empty();
                $('#select[name="sub_subcategory_id"]').append('<option value="" disabled selected>-- Select Sub SubCategory --</option>');
                $.each(data, function (keys, value) {
                    $('#select[name="sub_subcategory_id"]').append('<option value="" + value.id + "">' +
value.sub_subcategory_name + '</option>');
                });
            }
        });
    } else {
        $('#select[name="sub_subcategory_id"]').empty();
    }
});

// SELECT 2
$('#js-example-basic-single').select2();
$('#js-example-basic-multiple').select2({
    tags : true,
    theme : "classic"
});

});

```

```
</script>
```

```
<!-- SCRIPT FOR DATA TABLE -->
```

```
<script>
```

```
var options = {
  dom: '<"dataTables_wrapper dt-bootstrap"<"row"<"col-xl-7 d-block d-sm-flex d-xl-block
justify-content-center"<"d-block d-lg-inline-flex me-0 me-md-3"><"d-block d-lg-inline-flex"><"col-xl-5
d-flex d-xl-block justify-content-center"><"row"<"col-md-5"><"col-md-7">>>',
  buttons: [
    { extend: 'copy', className: 'btn-sm' },
    { extend: 'csv', className: 'btn-sm' },
    { extend: 'excel', className: 'btn-sm' },
    { extend: 'pdf', className: 'btn-sm' },
    { extend: 'print', className: 'btn-sm' }
  ],
  responsive: true,
  colReorder: true,
  keys: true,
  rowReorder: true,
  select: true
};
```

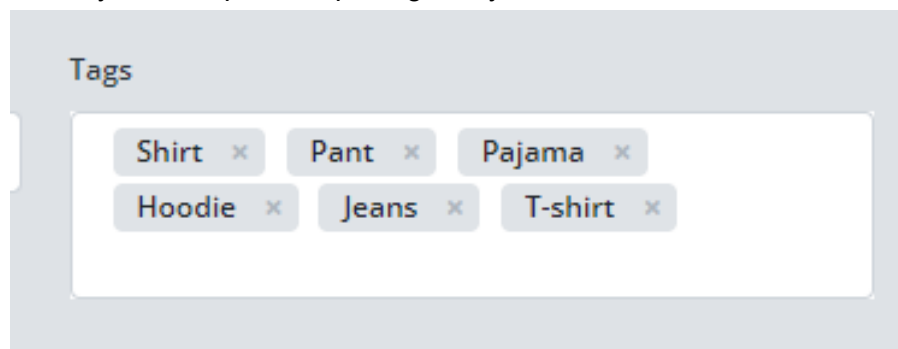
```
if ($(window).width() <= 767) {
  options.rowReorder = false;
  options.colReorder = false;
}
$('#data-table-combine').DataTable(options);
</script>
```

```
@endsection
```

Chapter - 08

Multiple Select 2 Options

Its like you will input multiple tags for your site, Like this:



You need jquery select2 for this. Call this required files in your blade file

```
<!-- Select2 css-->
<link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css" rel="stylesheet" />
```

```
<script src="https://code.jquery.com/jquery-3.6.2.min.js"
integrity="sha256-2krYZKh//PcchRtd+H+VyyQoZ/e3EcrkxhM8ycwASPA="
crossorigin="anonymous"></script>
```

```
<!--Select2 JS-->
<script src="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/js/select2.min.js"></script>
```

```
// SELECT 2
$('js-example-basic-multiple').select2({
  tags : true,
  theme : "classic"
});
```

When you will do for create functionality,

your tag will like that,

```
<select name="tags[]" multiple="multiple" id="tags" class="js-example-basic-multiple form-control mb-5px" style="width: 100%; height:100% ">
```

Your pass the data in json format

```
$selectbox->tags = json_encode($request->tags);
```

When you will do for EDIT/UPDATE functionality,

your tag will like that,

```
<select name="tags[]" multiple="multiple" id="tags" class="js-example-basic-multiple form-control mb-5px" style="width: 100%; height: 100%">
  @php
    $tags = json_decode($selectbox->tags);
  @endphp
  @foreach($tags as $tag)
    <option value="{{ $tag }}" selected>{{ $tag }}</option>
  @endforeach
</select>
```

Your pass the data in json format

```
$selectbox->tags = json_encode($request->tags);
```


Chapter - 13

Sweet Alert & Toaster Message

For Using Sweet alert and toaster message both in laravel you can use a package from realrashid

Reference Link:

<https://realrashid.github.io/sweet-alert/install>

To get started with SweetAlert2, use Composer to add the package to your project's dependencies:

```
composer require realrashid/sweet-alert
```

After installing the sweet-alert package, register the service provider

```
RealRashid\SweetAlert\SweetAlertServiceProvider::class,
```

in your `config/app.php` configuration file:

```
'providers' => [ /* * Package Service Providers... */

    RealRashid\SweetAlert\SweetAlertServiceProvider::class,

],
```

Also, add the `Alert` facade to the `aliases` array in your `app` configuration file:

```
'Alert' => RealRashid\SweetAlert\Facades\Alert::class,
```

Then publish it,

```
php artisan sweetalert:publish
```

After installing the package, add this scripts in your master blade,

```
{{--Sweet Alert--}}
@include('sweetalert::alert', ['cdn' => "https://cdn.jsdelivr.net/npm/sweetalert2@9"])
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
```

And This is for delete confirmation,

```
<script type="text/javascript">
$(document).on('click', '#delete', function(e){
    e.preventDefault();
    var link = $(this).attr("href");
    Swal.fire({
        title: 'Are you sure?',
        text: "This Record will be Permanently Deleted!",
        icon: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: 'Yes, Delete it!'
    }).then((result) => {
        if (result.isConfirmed) {
            window.location.href = link
            // Swal.fire(
            //     'Deleted!',
            //     'Your Data has been deleted.',
            //     'success'
            // )
        }
    })
});
```

Also for Toaster message, Write this codes in your master file anywhere (I recommend on top)

```
@if ($errors->any())
@foreach ($errors->all() as $error)
    @php
        Alert::toast($error, 'error')
    @endphp
@endforeach
@endif
```


Chapter - 19

Laravel Passport Authentication

Reference Link:

https://www.youtube.com/watch?v=6DZZNiUxkh8&list=PLjsp2uDzfb33uFpMsT_P_t6aw87-t6t26&index=12

https://www.youtube.com/watch?v=ZrC313McYg&list=PLjsp2uDzfb33uFpMsT_P_t6aw87-t6t26&index=13

https://www.youtube.com/watch?v=-EsCdosF7Tw&list=PLjsp2uDzfb33uFpMsT_P_t6aw87-t6t26&index=14

Installation and Setup

Step: 1 (Install Passport)

composer require laravel/passport

If time out error comes then run the bellow command **[If Timeour error comes]**

COMPOSER_MEMORY_LIMIT=-1 composer require laravel/passport

Step: 2 (Migration)

php artisan migrate

Step: 3 (Key Generate)

php artisan passport:install

Step: 4 (Change in User Model)

use Laravel\Passport\HasApiTokens;
use HasApiTokens, HasFactory, Notifiable;

Step: 5 (Update AuthServiceProvider) Update App\Providers\AuthServiceProvider

use Laravel\Passport\Passport;

In boot function add,

Passport::routes();

Note: In Laravel 9 We don't need to follow step 5

Step: 6 (Update config/auth.php)

```

        'guards' => [
            'web' => [
                'driver' => 'session',
                'provider' => 'users',
            ],

            'api' => [
                'driver' => 'passport',
                'provider' => 'users',
            ],
        ],
    ],

```

Step: 7 (Create Controller and write methods.)

Here I am creating a method named UserApiController and write method for User Register and Login.

```

public function registerUserWithPassport (Request $request){
    $data = $request->all();
    $rules = [
        'name' => 'required',
        'email' => 'required|email:rfc,dns|email|unique:users',
        'password' => 'required|min:8|max:25',
    ];
    $customMessage = [
        'name.required' => 'Name Should Required',
        'email.required' => 'Email Should Required',
        'email.unique' => 'Email should be unique',
        'password.required' => 'Password Should Required',
    ];
    $validator = Validator::make($data, $rules, $customMessage);
    if($validator->fails()){
        return response()->json($validator->errors(),422); // 422 er mane kono validation
        error paile kaj korbe
    }

    $user = new User();
    $user->name = $request->name;
    $user->email = $request->email;

```

```

$user->password = $request->password;
$user->access_token = $user->createToken('bearer_token')->accessToken;
$user->save();
$message = 'User Registered with Passport.';
return response()->json([
    'message' => $message,
    'access_token' => $user->access_token
],201); // 201 means created
}

```

```

public function loginUserWithPassport(Request $request){
    $data = $request->all();

```

```

    $rules = [
        'email' => 'required|email:rfc,dns|email|exists:users',
        'password' => 'required|min:8|max:25',
    ];
    $customMessage = [
        'email.required' => 'Email Should Required',
        'email.exists' => 'Email does not Exist',
        'password.required' => 'Password Should Required',
    ];

```

```

    $validator = Validator::make($data,$rules,$customMessage);

```

```

    if($validator->fails()){
        return response()->json($validator->errors(),422); // 422 er mane kono validation
        error paile kaj korbe
    }

```

```

    if(Auth::attempt(['email' => $data['email'], 'password' => $data['password']])){
        $user = User::where('email',$data['email']->first();
        $access_token = $user->createToken($data['email']->accessToken;
        User::where('email',$data['email']->update([
            'access_token' => $access_token
        ]);
        $message = 'User Registered Successfully.';
        return response()->json([

```

```

        'message' => $message,
        'access_token' => $access_token
    ],201); // 201 means created
    }
    else{
        $message = 'Login Failed! Email or Password Invalid..';
        return response()->json(['message' => $message],422); // 422 means error
    }
}
}

```

Step: 8 (Create Routes) routes > api.php

Here I am creating a method named UserApiController and write method for User Register and Login.

```

// PASSPORT AUTHENTICATION - Register User
Route::post('/register-user-with-passport', [UserApiController::class, 'registerUserWithPassport']);

// PASSPORT AUTHENTICATION - Login User
Route::post('/login-user-with-passport', [UserApiController::class, 'loginUserWithPassport']);

```

Step: 9 (Secure Api with Authentication token like this) routes > api.php

Here I am creating a method named UserApiController and write method for User Register and Login.

```

Route::middleware(['auth:api'])->group(function(){

    // GET USER WITH AUTHENTICATION
    Route::get('/auth-users/{id?}', [UserApiController::class, 'showUserByAuth']);

    // STORE USER WITH AUTHENTICATION
    Route::post('/auth-user-store/', [UserApiController::class, 'storeUserByAuth']);

});

```

Then go to postman and check these Url. Remember we will get a access token while login.. Add the beareer token in header when you check an API

Some Screen Shorts of API Codes You may need

Drive Link:

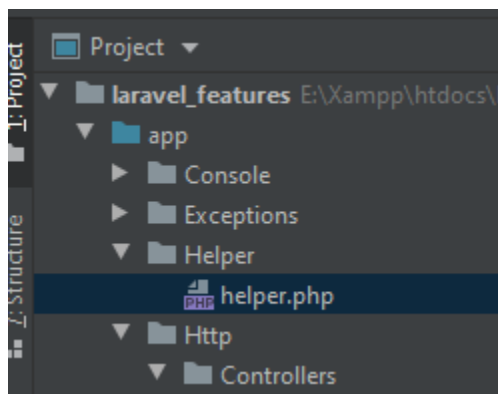
https://drive.google.com/drive/folders/10kQ6gzq4rGPcsn1v4Lg6wBOCqFkqs8qd?usp=share_link

Chapter - 21

How to Create Helper Function

Step 1: create a helper file

Go to app/ directory create a folder named Helper. In the helper Folder create a php file called helper.php



Step 2: Write the in the helper function

```
<?php
use App\Models\SendGmail;

function mail_count(){
    $mail = SendGmail::count();
    return $mail;
}
```

Step 3: Include it in our composer.json

```
"files": [
    "app/Helper/helper.php"
]
```

```
25     "autoload": {  
26         "psr-4": {  
27             "App\\": "app/",  
28             "Database\\Factories\\": "database/factories/",  
29             "Database\\Seeders\\": "database/seeders/"  
30         },  
31         "files": [  
32             "app/Helper/helper.php"  
33         ]  
34     },  
35     "autoload-dev": {
```

Step 4: Update composer

composer update

Step 4: Write this way in blade file,

```
{{ mail_count() }}
```

That is all, we can now call our function anywhere in our app

Chapter - 23

Datatable with Button

For Using Datatable you need to implement some of css and some of js files,

If You wanna implement it through CDN then, first of all goto this link: <https://cdn.datatables.net/>
And select your required styling options. I choose bootstrap5. Then paste this css and js files into your blade files.

```
<!--Data Table & Buttons required CSS files [CDN] -->
<link href="https://cdn.datatables.net/1.13.1/css/dataTables.bootstrap5.min.css" rel="stylesheet">

<link href="https://cdn.datatables.net/buttons/2.3.3/css/buttons.bootstrap5.min.css" rel="stylesheet">
<link href="https://cdn.datatables.net/colreorder/1.6.1/css/colReorder.bootstrap5.min.css"
rel="stylesheet">

<script src="https://code.jquery.com/jquery-3.6.2.min.js"
integrity="sha256-2krYZKh/PcchRtd+H+VyyQoZ/e3EcrkxhM8ycwASPA="
crossorigin="anonymous"></script>

<!--Data Table & Buttons required JS files [CDN] -->
<script type="text/javascript" src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
<script type="text/javascript"
src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></script>

<script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.3/js/dataTables.buttons.min.js"></script>
<script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.3/js/buttons.bootstrap5.min.js"></script>
<script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.colVis.min.js"></script>
<script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.html5.min.js"></script>
<script type="text/javascript" src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.print.min.js"></script>
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>

<!-- SCRIPT FOR DATA TABLE BUTTON FUNCTIONALITY -->
<script>
  var options = {
    dom: '<"dataTables_wrapper dt-bootstrap"><"row"><"col-xl-7 d-block d-sm-flex d-xl-block
justify-content-center"><"d-block d-lg-inline-flex me-0 me-md-3"><"d-block d-lg-inline-flex"><"col-xl-5
d-flex d-xl-block justify-content-center"><"row"><"col-md-5"><"col-md-7"><"p>>>';
```

```

        buttons: [
            { extend: 'copy', className: 'btn-sm' },
            { extend: 'csv', className: 'btn-sm' },
            { extend: 'excel', className: 'btn-sm' },
            { extend: 'pdf', className: 'btn-sm' },
            { extend: 'print', className: 'btn-sm' }
        ],
        responsive: true,
        colReorder: true,
        keys: true,
        rowReorder: true,
        select: true
    };

    if ($(window).width() <= 767) {
        options.rowReorder = false;
        options.colReorder = false;
    }
    $('#data-table-combine').DataTable(options);
</script>

```

Then call the id in your table tag,

```

<table id="data-table-combine" class="table table-striped table-bordered align-middle" style="font-size:
14px;">
.....
.....
</table>

```

If You do not want to use CDN then you can follow [Laravel Features System](#).

Chapter - 25

Yajra Datatable with Complete CRUD

When you have so many data in a table like 20-50k or more. Then the traditional datatable will not be efficient for you. Then you need yajra datatable because it fetch data from server side then the load will be reduce enough.

For yajra you need same process, you need a table and migrate it, create models and controller and also for big amount of data use faker & factory. After all done install yajra,

Reference Link:

<https://www.itsolutionstuff.com/post/laravel-9-yajra-datatables-example-tutorialexample.html>

<https://www.itsolutionstuff.com/post/laravel-yajra-datatables-export-to-excel-csv-button-example-example.html>

<https://yajrabox.com/docs/laravel-datatables/8.0/installation>

Step - 1 (Install yajra with buttons)

```
composer require yajra/laravel-datatables-oracle
composer require yajra/laravel-datatables-buttons
```

Step - 2 (change in config>app.php)

```
.....
'providers' => [
    ....
    Yajra\DataTables\DataTablesServiceProvider::class,
    Yajra\DataTables\ButtonsServiceProvider::class,
]
'aliases' => [
    ....
    'DataTables' =>
        Yajra\DataTables\Facades\DataTables::class,
]
.....
```

Now you have to run bellow command to get configuration file, so let's run bellow command:

```
php artisan vendor:publish --tag=datatables-buttons
```

Step - 3(Create a Controller and write these codes)

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\YajraDataTable;
use Illuminate\Http\Request;
use DataTables;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\File;
use Illuminate\Support\Facades\View;
use Illuminate\Support\Str;
use Intervention\Image\Facades\Image;
use RealRashid\SweetAlert\Facades\Alert;

class YajraDataTableController extends Controller
{
    public function index(Request $request){
        if($request->ajax()){
            $data = YajraDataTable::select('id','name','email','phone')->latest()->get();
            return Datatables::of($data)
                ->addIndexColumn()
                ->addColumn('action', function($data){
                    $btn = '<button type="button" data-bs-toggle="modal"
data-bs-target="#exampleModal" name="edit" value="'. $data->id.'" class="mx-2 edit btn
btn-primary btn-sm">Edit</button>';
                    $btn .= '<button type="button" data-bs-toggle="modal"
data-bs-target="#exampleModalDelete" name="delete" value="'. $data->id.'"
class="mx-2 delete btn btn-danger btn-sm">Delete</button>';
                    return $btn;
                })
                ->rawColumns(['action'])
                ->make(true);
        }
        return view('admin.yajra.index');
    }
}
```

```

public function store(Request $request){
    $request->validate([
        'image' => 'nullable|mimes:jpeg,png,jpg,gif,svg|max:2048',
        'name' => 'required|min:3|max:50',
        'email' => 'required|email:rfc,dns|email|unique:yajra_data_tables,email',
        // 'phone' => 'required|min:11|max:11|regex:/^(01)[0-9]{9}/',
        'phone' => 'required|min:11|max:11',
        'address' => 'required|min:3|max:255',
        'description' => 'required|min:3|max:255',
    ]);

    try {
        DB::beginTransaction();

        // IMAGE SECTION
        $image = $request->file('image');
        if($image != null){

            $name = Str::random(5).'.'.$image->getClientOriginalExtension();
            $destinationPath = public_path('/uploads/yajra-image');
            if (!File::exists($destinationPath)) {
                File::makeDirectory($destinationPath, 0777, true);
            }
            Image::make($image->getRealPath())->resize(200, 200, function ($constraint)
            {
                $constraint->aspectRatio();
            })->save($destinationPath.'/'.$name);
            $destinationPath = 'uploads/yajra-image/';
        }

        // IMAGE SECTION

        $yajra = new YajraDataTable();
        $yajra->name = $request->name;
        $yajra->email = $request->email;
        $yajra->phone = $request->phone;
        $yajra->address = $request->address;
        $yajra->description = $request->description;
        if($request->image != null){
            $yajra->image = $destinationPath.$name;
        }
    }
}

```



```

        $yajra->save();

        DB::commit();
        Alert::success('Yajra Data Created Successfully..');
        return back();
    }
    catch(\Exception $e){
        DB::rollBack();
        Alert::error($e->getMessage());
        return redirect()->back();
    }
}

public function edit($id){
    $yajra = YajraDataTable::findOrFail($id);
    // return view('admin.yajra.edit', compact('yajra'));
    // return $yajra;
    $view = View::make('admin.yajra.edit', compact('yajra'))->render();
    return response()->json(['html' => $view]);
}

public function update(Request $request, $id){
    $yajra = YajraDataTable::findOrFail($id);

    $request->validate([
        'image' => 'nullable|mimes:jpeg,png,jpg,gif,svg|max:2048',
        'name' => 'required|min:3|max:50',
        'email' => 'required|email|unique:yajra_data_tables,email,'.$yajra->id,
        'phone' => 'required|min:10|max:11',
        'address' => 'required|min:3|max:255',
        'description' => 'required|min:3|max:255',
    ]);

    try {
        DB::beginTransaction();

        // IMAGE SECTION
        $image = $request->file('image');
        if($image != null){
            if($yajra->image != null){

```

```

        unlink(public_path($yajra->image));
    }
    $name = Str::random(5).'.'.$image->getClientOriginalExtension();
    $destinationPath = public_path('/uploads/yajra-image');
    if (!File::exists($destinationPath)) {
        File::makeDirectory($destinationPath, 0777, true);
    }
    Image::make($image->getRealPath())->resize(200, 200, function ($constraint)
    {
        $constraint->aspectRatio();
    })->save($destinationPath.'/'.$name);
    $destinationPath = 'uploads/yajra-image/';
    $yajra->image = $destinationPath.$name;
}

// IMAGE SECTION

$yajra->name = $request->name;
$yajra->email = $request->email;
$yajra->phone = $request->phone;
$yajra->address = $request->address;
$yajra->description = $request->description;
$yajra->save();
DB::commit();
Alert::success('Yajra Data Updated Successfully..');
return back();
}

catch(\Exception $e){
    DB::rollBack();
    Alert::error($e->getMessage());
    return redirect()->back();
}
}

public function destroy($id){
    $yajra = YajraDataTable::findOrFail($id);
    $view = View::make('admin.yajra.delete', compact('yajra'))->render();
    return response()->json(['html' => $view]);
}

public function destroyConfirm(Request $request, $id){

```

```

    $yajra = YajraDataTable::findOrFail($id);
    $yajra->delete();
    Alert::success('Yajra Data Deleted Successfully..');
    return redirect()->back();
}
}

```

Step - 4 (Goto web.php and create these routes)

```

// YAJRA DATATABLE -- [BUTTON]
Route::get('/yajra', [YajraDataTableController::class, 'index'])->name('yajra.datatable');
Route::post('/yajra/store', [YajraDataTableController::class, 'store'])->name('store.yajra');
Route::get('/yajra/edit/{id}', [YajraDataTableController::class, 'edit'])->name('edit.yajra');
Route::post('/yajra/update/{id}', [YajraDataTableController::class, 'update'])->name('update.yajra');
Route::get('/yajra/destroy/{id}', [YajraDataTableController::class, 'destroy'])->name('destroy.yajra');
Route::post('/yajra/destroy-confirm/{id}', [YajraDataTableController::class, 'destroyConfirm'])->name('destroy.confirm');

```

Step - 5 (Create a blade file my case admin.yajra.index)

```
@extends('admin.layout.master')
```

```
@section('title')
```

```
Yajra Datatable | CRUD
```

```
@endsection
```

```
@section('extra-css')
```

```
<meta name="csrf-token" content="{{ csrf_token() }}">
```

```
<!--Data Table required files -->
```

```
<link href="{{
```

```
asset('/')}}backend/assets/plugins/datatables.net-bs5/css/dataTables.bootstrap5.min.css"
rel="stylesheet" />
```

```
<link href="{{
```

```
asset('/')}}plugins/datatables.net-responsive-bs5/css/responsive.bootstrap5.min.css"
rel="stylesheet" />
```

```

<link href="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons-bs5/css/buttons.bootstrap5.min.css"
rel="stylesheet" />

<!--Data Table Buttons required files -->

<link href="{{
asset('/')}}backend/assets/plugins/datatables.net-colreorder-bs5/css/colReorder.bootstrap5.min.cs
s" rel="stylesheet" />

<link href="{{
asset('/')}}backend/assets/plugins/datatables.net-keytable-bs5/css/keyTable.bootstrap5.min.css"
rel="stylesheet" />

<link href="{{
asset('/')}}backend/assets/plugins/datatables.net-rowreorder-bs5/css/rowReorder.bootstrap5.min.
css" rel="stylesheet" />

<link href="{{
asset('/')}}backend/assets/plugins/datatables.net-select-bs5/css/select.bootstrap5.min.css"
rel="stylesheet" />

```

```

<!--Data Table & Buttons required CSS files [CDN] -->
{{-- <link href="https://cdn.datatables.net/1.13.1/css/dataTables.bootstrap5.min.css"
rel="stylesheet">--}}

```

```

{{-- <link href="https://cdn.datatables.net/buttons/2.3.3/css/buttons.bootstrap5.min.css"
rel="stylesheet">--}}

```

```

{{-- <link href="https://cdn.datatables.net/colreorder/1.6.1/css/colReorder.bootstrap5.min.css"
rel="stylesheet">--}}

```

@endsection

@section('admin-content')

```

<ol class="breadcrumb float-xl-end">
    <li class="breadcrumb-item"><a href="">Home</a></li>
    <li class="breadcrumb-item active">Yajra Data Table</li>
</ol>

```

```

<div class="row my-5 ">
    <div class="mx-auto p-4" style="border: 1px solid #801419; border-radius: 4px;">

```

```

<!-- ADD / CREATE SECTION -->

<form action="{{ route('store.yajra') }}" method="POST" enctype="multipart/form-data" >
    <h1 class="page-header ">Create a Yajra Data </h1>

    @csrf

```

```

    <div class="row mb-15px">
      <div class="col-md-4">
        <label class="form-label col-form-label ">Name</label>
        <div class="">
          <input type="text" name="name" class="form-control mb-5px"
placeholder="Enter Name" />
        </div>
      </div>
      <div class="col-md-4">
        <label class="form-label col-form-label ">Email</label>
        <div class="">
          <input type="email" name="email" class="form-control mb-5px"
placeholder="Enter email" />
        </div>
      </div>
      <div class="col-md-4">
        <label class="form-label col-form-label ">Phone</label>
        <div class="">
          <input type="number" name="phone" class="form-control mb-5px"
placeholder="Enter Phone" />
        </div>
      </div>
    </div>

    <div class="row mb-15px">
      <div class="col-md-4">
        <label class="form-label col-form-label ">Description</label>
        <div class="">
          <input type="text" name="description" class="form-control mb-5px"
placeholder="Enter Description" />
        </div>
      </div>
      <div class="col-md-4">
        <label class="form-label col-form-label ">Address</label>
        <div class="">
          <input type="text" name="address" class="form-control mb-5px"
placeholder="Enter Address" />
        </div>
      </div>
    </div>

```

```

        <label class="form-label col-form-label ">Image</label>
        <div class="">
            <input type="file" name="image" accept="image/*" class="form-control
mb-5px" placeholder="Enter Name" />
        </div>
    </div>

</div>

<div class="row mb-15px">
    <label class="form-label col-form-label "></label>
    <div class="">
        <button type="submit" class="btn btn-info d-block w-100 h-45px
btn-lg">Create</button>
    </div>
</div>
</form>
</div>
</div>

<!-- Table Section -->
<div class="row">
    <table id="user_datatable" class="user_datatable table table-striped table-bordered
align-middle" style="font-size: 14px;">
        <thead>
            <tr>
                <th>SL</th>
                <th>Name</th>
                <th>Email</th>
                <th>Phone</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
        </tbody>
    </table>
</div>

<!-- Edit Modal -->

```

```

<div class="modal fade " id="exampleModal" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true" >
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Edit Yajra Data</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div id="modal_data"></div>
      </div>
    </div>
  </div>
</div>

<!-- Delete Modal -->
<div class="modal fade " id="exampleModalDelete" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true" >
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Delete This Yajra Data</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div id="delete_modal_data"></div>
      </div>
    </div>
  </div>
</div>

@endsection

@section('extra-js')
  <script src="https://code.jquery.com/jquery-3.6.2.min.js"
integrity="sha256-2krYZKh//PcchRtd+H+VyyQoZ/e3EcrkxhM8ycwASPA="
crossorigin="anonymous"></script>

```

```

<!--Data Table required files -->

```

```

<script src="{{ asset('/') }}vendor/datatables/buttons.server-side.js"></script>

<script src="{{
asset('/') }}backend/assets/plugins/datatables.net/js/jquery.dataTables.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-bs5/js/dataTables.bootstrap5.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-responsive/js/dataTables.responsive.min.js"></s
cript>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-responsive-bs5/js/responsive.bootstrap5.min.js"
></script>

<!--Data Table Buttons required files -->
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-colreorder/js/dataTables.colReorder.min.js"></sc
ript>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-colreorder-bs5/js/colReorder.bootstrap5.min.js">
</script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-keytable/js/dataTables.keyTable.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-keytable-bs5/js/keyTable.bootstrap5.min.js"></sc
ript>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-rowreorder/js/dataTables.rowReorder.min.js"></s
cript>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-rowreorder-bs5/js/rowReorder.bootstrap5.min.js"
></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-select/js/dataTables.select.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-select-bs5/js/select.bootstrap5.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-buttons/js/dataTables.buttons.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-buttons-bs5/js/buttons.bootstrap5.min.js"></scri
pt>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-buttons/js/buttons.colVis.min.js"></script>
<script src="{{
asset('/') }}backend/assets/plugins/datatables.net-buttons/js/buttons.flash.min.js"></script>

```



```

<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.html5.min.js"></script>
<script src="{{
asset('/')}}backend/assets/plugins/datatables.net-buttons/js/buttons.print.min.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/pdfmake/build/pdfmake.min.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/pdfmake/build/vfs_fonts.js"></script>
<script src="{{ asset('/')}}backend/assets/plugins/jszip/dist/jszip.min.js"></script>

```

```

<!--Data Table & Buttons required JS files [CDN] -->
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></script>--}}

```

```

{{-- <script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.3/js/dataTables.buttons.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.3/js/buttons.bootstrap5.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.colVis.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.html5.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdn.datatables.net/buttons/2.3.2/js/buttons.print.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jszip/3.1.3/jszip.min.js"></script>--}}
{{-- <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.53/pdfmake.min.js"></script>--}}

```

```

<script type="text/javascript">
// YAJRA DATATABLE FUNCTIONALITY + BUTTON
$(function () {
var table = $('#user_datatable').DataTable({
processing: true,
serverSide: true,
responsive: true,
autoWidth: false,
ajax: "{{ route('yajra.datatable') }}",
columns: [
{ data: 'DT_RowIndex', name: 'DT_RowIndex', orderable: false, searchable: false },
{ data: 'name', name: 'name' },
{ data: 'email', name: 'email' },

```

```

        {data: 'phone', name: 'phone'},
        {data: 'action', name: 'action', orderable: false, searchable: false},
    ],
    dom: 'BIfrtip', // Bftrtip, IBftrtip -- other options
    buttons: [
        { extend: 'copy', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        { extend: 'excel', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        { extend: 'csv', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        { extend: 'pdf', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        { extend: 'print', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        { extend: 'colvis', exportOptions: { modifier: { page: 'all', search: 'none' } } },
        // 'copy', 'csv', 'excel', 'pdf', 'print'
    ],
    "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, 'All'] ],
    });
});

// EDIT FUNCTIONALITY STARTS
$(document).on('click', '.edit', function(event){
    event.preventDefault();
    var id = $(this).val();
    // alert(id);
    $.ajax({
        url: "{{ url('admin/yajra/edit/') }}" + id,
        type: "GET",
        dataType: "json",
        success: function (data) {
            // console.log(data);
            $("#modal_data").html(data.html);
        },
        error: function(data){
            var errors = data.responseJSON;
            console.log(errors);
        }
    });
});

// EDIT FUNCTIONALITY ENDS

// DELETE FUNCTIONALITY STARTS
$(document).on('click', '.delete', function(event){

```

```

    event.preventDefault();
    var id = $(this).val();
    // alert(id);
    $.ajax({
        url: "{{ url('admin/yajra/destroy/') }}" + id,
        type: "GET",
        dataType: "json",
        success: function (data) {
            // console.log(data);
            $("#delete_modal_data").html(data.html);
        },
        error: function(data){
            var errors = data.responseJSON;
            console.log(errors);
        }
    });
    // DELETE FUNCTIONALITY ENDS

```

```
</script>
```

```
@endsection
```

Step - 6 (Create another blade file my case admin.yajra.edit)

```

<!-- EDIT / UPDATE SECTION -->
<form action="{{ route('update.yajra', $yajra->id) }}" method="POST"
enctype="multipart/form-data" >
    @csrf
    <div class="row mb-15px">
        <div class="col-md-6">
            <label class="form-label col-form-label ">Name</label>
            <div class="">
                <input type="text" name="name" class="form-control mb-5px" placeholder="Enter
Name" value="{{ $yajra->name }}" />
                @if ($errors->has('name'))
                    <span class="text-red-500">{{ $errors->first('name') }}</span>
                @endif
            </div>

```

```

    </div>
    <div class="col-md-6">
        <label class="form-label col-form-label ">Email</label>
        <div class="">
            <input type="email" name="email" class="form-control mb-5px" placeholder="Enter
email" value="{{ $yajra->email }}" />
            @if ($errors->has('email'))
                <span class="text-red-500">{{ $errors->first('email') }}</span>
            @endif
        </div>
    </div>
</div>

```

```

<div class="row mb-15px">
    <div class="col-md-6">
        <label class="form-label col-form-label ">Phone</label>
        <div class="">
            <input type="text" name="phone" class="form-control mb-5px" placeholder="Enter
Phone" value="{{ $yajra->phone }}" />
            @if ($errors->has('phone'))
                <span class="text-red-500">{{ $errors->first('phone') }}</span>
            @endif
        </div>
    </div>
    <div class="col-md-6">
        <label class="form-label col-form-label ">Description</label>
        <div class="">
            <input type="text" name="description" class="form-control mb-5px"
placeholder="Enter Description" value="{{ $yajra->description }}" />
            @if ($errors->has('description'))
                <span class="text-red-500">{{ $errors->first('description') }}</span>
            @endif
        </div>
    </div>
</div>

```

```

<div class="row mb-15px">
    <div class="col-md-6">
        <label class="form-label col-form-label ">Address</label>
        <div class="">

```

```

        <input type="text" name="address" class="form-control mb-5px" placeholder="Enter
Address" value="{{ $yajra->address }}" />
        @if ($errors->has('address'))
            <span class="text-red-500">{{ $errors->first('address') }}</span>
        @endif
    </div>
</div>

<div class="col-md-6">
    <label class="form-label col-form-label">Image</label>
    <div class="">
        
        <input type="file" name="image" accept="image/*" class="form-control mb-5px" />
        @if ($errors->has('image'))
            <span class="text-red-500">{{ $errors->first('image') }}</span>
        @endif
    </div>
</div>

</div>

<div class="modal-footer">
    <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>
    <button type="submit" class="btn btn-primary">Update</button>
</div>
</form>

```

Step - 6 (Create another blade file my case admin.yajra.delete)

```

<!-- Delete SECTION -->
<form action="{{ route('destroy.confirm', $yajra->id) }}" method="POST"
enctype="multipart/form-data" >
    @csrf
    <div class="row mb-15px">
        <div class="col-md-6">
            <label class="form-label col-form-label">Name</label>
            <div class="">
                <input type="text" name="name" class="form-control mb-5px" placeholder="Enter
Name" value="{{ $yajra->name }}" />

```

```

    </div>
  </div>
  <div class="col-md-6">
    <label class="form-label col-form-label ">Email</label>
    <div class="">
      <input type="email" name="email" class="form-control mb-5px" placeholder="Enter
email" value="{{ $yajra->email }}" />
      @if ($errors->has('email'))
        <span class="text-red-500">{{ $errors->first('email') }}</span>
      @endif
    </div>
  </div>
</div>

<div class="modal-footer">
  <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>
  <button type="submit" class="btn btn-danger">Delete</button>
</div>
</form>

```