Laravel 10 + Breeze ব্যবহার করে **Multiple Authentication (Admin + Customer)** সেটআপ করার জন্য ধাপে ধাপে কাজ করবো।

আমরা নিম্নলিখিত বিষয়গুলো করবো:  
✅ Laravel 10 ইনস্টল  
✅ Breeze Authentication সেটআপ  
✅ Database তৈরি ও Role ফিল্ড যোগ করা  
✅ Middleware তৈরি (Admin + Customer)  
✅ Routes এবং Blade Views সেটআপ  
✅ Postman দিয়ে API টেস্ট

**📌 Step 1: Laravel 10 Project Setup**

প্রথমে Laravel 10 ইন্সটল করুন:

* composer create-project laravel/laravel:^10.0 CarRentalSystem
* cd CarRentalSystem

Breeze ইন্সটল করুন:

* composer require laravel/breeze --dev
* php artisan breeze:install

Frontend স্ক্যাফোল্ডিং তৈরি করুন (Blade ব্যবহারের জন্য):

* php artisan breeze:install blade

Dependency ইনস্টল করুন:

* npm install && npm run dev

**Step 2: Database Setup**

1️⃣ **.env ফাইলে Database কানেকশন কনফিগার করুন:**

* DB\_DATABASE=car\_rental
* DB\_USERNAME=root
* DB\_PASSWORD=

2️⃣ **মাইগ্রেশন রান করুন:**

* php artisan migrate

**📌 Step 3: User Model-এ Role যোগ করুন**

👉 **app/Models/User.php আপডেট করুন:**

namespace App\Models;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class User extends Authenticatable

{

use Notifiable;

protected $fillable = [

'name',

'email',

'password',

'role'

];

protected $hidden = [

'password',

'remember\_token',

];

public function isAdmin()

{

return $this->role === 'admin';

}

public function isCustomer()

{

return $this->role === 'customer';

}

}

**📌 Step 4: Role ফিল্ড যোগ করা (Migration)**

👉 **নতুন মাইগ্রেশন তৈরি করুন:**

php artisan make:migration add\_role\_to\_users\_table --table=users

👉 **মাইগ্রেশন ফাইল (database/migrations/YYYY\_MM\_DD\_add\_role\_to\_users\_table.php) আপডেট করুন:**

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration {

public function up()

{

Schema::table('users', function (Blueprint $table) {

$table->string('role')->default('customer'); // Default role: customer

});

}

public function down(){

Schema::table('users', function (Blueprint $table) {

$table->dropColumn('role');

});

}

};

👉 **মাইগ্রেশন রান করুন:**

* php artisan migrate

**📌 Step 5: Middleware তৈরি করা (Admin & Customer)**

👉 **Admin Middleware তৈরি করুন:**

* php artisan make:middleware AdminMiddleware

👉 **app/Http/Middleware/AdminMiddleware.php আপডেট করুন:**

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class AdminMiddleware

{

public function handle(Request $request, Closure $next)

{

if (Auth::check() && Auth::user()->isAdmin()) {

return $next($request);

}

return redirect('/')->with('error', 'Unauthorized access!');

}

}

👉 **Customer Middleware তৈরি করুন:**

* php artisan make:middleware CustomerMiddleware

👉 **app/Http/Middleware/CustomerMiddleware.php আপডেট করুন:**

namespace App\Http\Middleware;

use Closure;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class CustomerMiddleware

{

public function handle(Request $request, Closure $next)

{

if (Auth::check() && Auth::user()->isCustomer()) {

return $next($request);

}

return redirect('/')->with('error', 'Unauthorized access!');

}

}

👉 **Middleware রেজিস্ট্রেশন করুন (app/Http/Kernel.php)**

protected $routeMiddleware = [

'admin' => \App\Http\Middleware\AdminMiddleware::class,

'customer' => \App\Http\Middleware\CustomerMiddleware::class,

];

**📌 Step 6: Routes সেটআপ করা (web.php)**

👉 **routes/web.php আপডেট করুন:**

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\Auth\AuthenticatedSessionController;

use App\Http\Controllers\DashboardController;

Route::get('/', function () {

return view('welcome');

});

// Default Breeze Authentication Routes

require \_\_DIR\_\_.'/auth.php';

// Customer Dashboard Route

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

Route::get('/customer/dashboard', [DashboardController::class, 'customerDashboard'])->name('customer.dashboard');

});

// Admin Dashboard Route

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

Route::get('/admin/dashboard', [DashboardController::class, 'adminDashboard'])->name('admin.dashboard');

});

**📌 Step 7: Dashboard Controller তৈরি করা**

👉 **Dashboard Controller তৈরি করুন:**

php artisan make:controller DashboardController

👉 **app/Http/Controllers/DashboardController.php আপডেট করুন:**

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class DashboardController extends Controller

{

public function customerDashboard()

{

return view('dashboard.customer');

}

public function adminDashboard()

{

return view('dashboard.admin');

}

}

**📌 Step 8: Blade Views তৈরি করা**

👉 **Customer Dashboard View (resources/views/dashboard/customer.blade.php)**

@extends('layouts.app')

@section('content')

<div class="container">

<h1>Customer Dashboard</h1>

<p>Welcome, {{ Auth::user()->name }}</p>

</div>

@endsection

👉 **Admin Dashboard View (resources/views/dashboard/admin.blade.php)**

@extends('layouts.app')

@section('content')

<div class="container">

<h1>Admin Dashboard</h1>

<p>Welcome, {{ Auth::user()->name }}</p>

</div>

@endsection

👉 **app/Http/Controllers/Auth/AuthenticatedSessionController.php আপডেট করুন:**

public function store(LoginRequest $request): RedirectResponse

    {

        $request->authenticate();

        $request->session()->regenerate();

        $user = Auth::user();

        if($user->role == 'admin')

            return redirect()->intended('/admin/dashboard');

        elseif($user->role = 'customer')

            return redirect()->intended('/customer/dashboard');

        else

            return redirect()->intended(RouteServiceProvider::HOME);

    }

**Step 9: Postman দিয়ে টেস্ট করুন**

**১. রেজিস্ট্রেশন (Register New User)**

POST http://127.0.0.1:8000/register

Content-Type: application/json

{

"name": "Admin User",

"email": "admin@example.com",

"password": "password",

"password\_confirmation": "password",

"role": "admin"

}

**২. লগইন (Login)**

POST http://127.0.0.1:8000/login

Content-Type: application/json

{

"email": "admin@example.com",

"password": "password"

}

**৩. Customer Dashboard**

GET http://127.0.0.1:8000/customer/dashboard

Authorization: Bearer {ACCESS\_TOKEN}

**৪. Admin Dashboard**

GET http://127.0.0.1:8000/admin/dashboard

Authorization: Bearer {ACCESS\_TOKEN}

**📌 Step 10: Cache Clear & Serve**

php artisan config:clear

php artisan cache:clear

php artisan route:clear

php artisan view:clear

php artisan migrate --seed

php artisan serve

**Customer(CRUD) using user model**

* **php artisan make:controller Admin/CustomerController –resource**

**Step 1: ALL view**

**controller: Admin/CustomerController.php**

1. **flowing index method**

public function index()

    {

        $customers = User::latest()->get();

        // return view('pages.index',compact('customers'));

        return view('dashboard.admin.customers.index', compact('customers'));

    }

1. **View :** [**Dashboard/admin/customers/index.blade.php**](resources/views/dashboard/admin/customers/index.blade.php)
2. **Web route :**

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

//customer

    Route::get('/customers', [CustomerController::class, 'index'])->name('admin.customers.index');

});

1. **resources\views\layouts\admin\partials /[sidebar\_left\_panel.blade.php](resources/views/layouts/admin/partials/sidebar_left_panel.blade.php)**

<a href="{{route('admin.customers.index')}}" class="sl-menu-link @if($segment1==='customers') active show-sub @else  @endif">

            <div class="sl-menu-item">

              <i class="menu-item-icon icon ion-ios-photos-outline tx-20"></i>

              <span class="menu-item-label">Customers</span>

            </div><!-- menu-item -->

          </a><!-- sl-menu-link -->

**Step 2: Create view**

1. **flowing create method**

public function create()

    {

        return view('dashboard.admin.customers.create');

    }

1. **View** [**: Dashboard/admin/customers/create.blade.php**](resources/views/dashboard/admin/customers/create.blade.php)
2. **Web route :**

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

//customer

   Route::get('/customers/create', [CustomerController::class, 'create'])->name('admin.customers.create');

});

1. **Route add: admin.customers.index.blade.php**

<div class="col-sm-2">

                <a href="{{ route('admin.customers.create') }}" class="btn btn-success"><i class="material-icons">&#xE147;</i> <span>Add New Customer</span></a>

</div>

**Toastr setup:**

**Install : composer require yoeunes/toastr**

**usage:**

**toastr()->success('Data has been saved successfully!');**

**return redirect()->route('posts.index');**

**Step 3: Store data**

1. **flowing store method**

public function store(Request $request){

        $request->validate([

            'name' => ['required', 'string', 'max:255'],

            'email' => ['required', 'string', 'lowercase', 'email', 'max:255', 'unique:'.User::class],

            'password' => ['required'],

        ]);

        User::create([

            'name' => $request->name,

            'email' => $request->email,

            'password' => Hash::make($request->password),

            'phone' => $request->phone,

            'address' => $request->address,

        ]);

        toastr()->success('Customer Add SuccessFully !!');

        return redirect()->route('admin.customers.index');

    }

1. **Web route:**

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

//customer

   Route::post('/customers/store', [CustomerController::class, 'store'])->name('admin.customers.store');

});

1. **Route add**

<form action="{{ route('admin.customers.store') }}" method="POST">

**Step 4: Data Delete**

1. **flowing destroy method**

public function destroy(string $id)

    {

        User::where('id',$id)->delete();

        toastr()->success('Customer Delete SuccessFully !!');

        return redirect()->route('admin.customers.index');

    }

1. **Web route:**

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

//customer

   Route::delete('/customers/{id}', [CustomerController::class, 'destroy'])->name('admin.customers.destroy');

});

1. [**Modal add**](resources/views/dashboard/admin/customers/index.blade.php) **:**

**Madal delete +**

1. **Route add:**

 <form action="{{ route('admin.customers.destroy',$customer->id) }}" method="post">

**Step 4: Edit view**

1. **flowing edit method**

public function edit(string $id)

    {

        $customer = User::where('id',$id)->first();

        return view('dashboard.admin.customers.edit',['customer'=>$customer]);

    }

1. **View :** [**Dashboard/admin/customers/edit.blade.php**](resources/views/dashboard/admin/customers/edit.blade.php)
2. **Web route.**

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

//customer

   Route::get('/customers/{id}/edit', [CustomerController::class, 'edit'])->name('admin.customers.edit');

});

1. **Route add:** **Dashboard/admin/customers/index.blade.php**

 <a href="{{ route('admin.customers.edit',$customer->id) }}" class="edit"><i class="material-icons" data-toggle="tooltip" title="Edit">&#xE254;</i></a>

**Step 5: Edit Update**

1. **flowing update method**

public function update(Request $request, string $id){

        $request->validate([

            'name' => 'required',

        ]);

        User::where('id',$id)->update([

            'name' => $request->input('name'),

            'email' => $request->input('email'),

            'phone' => $request->input('phone'),

            'address' => $request->input('address'),

        ]);

        toastr()->success('Customer update SuccessFully !!');

        return redirect()->route('admin.customers.index');

    }

1. **Web route:**

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

//customer

   Route::put('/customers/{id}/update', [CustomerController::class, 'update'])->name('admin.customers.update');

});

1. **Route add: Dashboard/admin/customers/edit.blade.php**

 <form action="{{ route('admin.customers.update',$customer->id) }}" method="POST">

                        @csrf

                        @method('PUT')

**Rental (CRUD)**

**php artisan make:controller Admin/RentalController –resource**

**Step 1: ALL view**

**controller: Admin/ RentalController.php**

1. **flowing index method**

public function index(){

        return view('dashboard.admin.rentals.index');

    }

1. **View :** [**Dashboard/admin/rentals/index.blade.php**](resources/views/dashboard/admin/rentals/index.blade.php)
2. **Web route :**

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

//customer

   Route::get('/rentals', [RentalController::class, 'index'])->name('admin.rentals.index');

});

1. **resources\views\layouts\admin\partials /[sidebar\_left\_panel.blade.php](resources/views/layouts/admin/partials/sidebar_left_panel.blade.php)**

<a href="{{route('admin.rentals.index')}}" class="sl-menu-link @if($segment1==='rentals') active show-sub @else  @endif">

            <div class="sl-menu-item">

              <i class="menu-item-icon icon ion-ios-photos-outline tx-20"></i>

              <span class="menu-item-label">Rentals</span>

            </div><!-- menu-item -->

          </a><!-- sl-menu-link -->

**Step 2: Create view**

1. **flowing create method**

public function create(){

        return view('dashboard.admin.rentals.create');

    }

1. **View** [**: Dashboard/admin/rentals/create.blade.php**](resources/views/dashboard/admin/rentals/create.blade.php)
2. **Web route :**

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

   Route::get('/rentals/create', [RentalController::class, 'create'])->name('admin.rentals.create');

});

1. **Route add:** [**admin.rentals.index.blade.php**](resources/views/dashboard/admin/rentals/index.blade.php)

<div class="col-sm-2">

<a href="{{ route('admin.rentals.create') }}" class="btn btn-success"><i class="material-icons">&#xE147;</i> <span>Add New Rental</span></a>

</div>

* **php artisan make:model Rental –m**

php artisan migrate --path=database/migrations/2025\_02\_20\_160131\_create\_rentals\_table.php

* [**2025\_02\_20\_160131\_create\_rentals\_table.php**](database/migrations/2025_02_20_160131_create_rentals_table.php)
* **Models (User, Car, Rental)**

**Car model**

public function rentals(){

        return $this->hasMany(Rental::class);

    }

**User model**

public function rentals(){

        return $this->hasMany(Rental::class);

    }

**Rental model**

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Rental extends Model{

    use HasFactory;

    protected $fillable = ['user\_id', 'car\_id', 'start\_date', 'end\_date', 'total\_cost', 'status'];

    public function user(){

        return $this->belongsTo(User::class);

    }

    public function car(){

        return $this->belongsTo(Car::class);

    }

}

1. **flowing create method :update**

public function create(){

        $users = User::all();

        $cars = Car::all();

        return view('dashboard.admin.rentals.create',compact('users', 'cars'));

    }

**Step 3: Store data**

1. **flowing store method**

public function store(Request $request){

        //dd($request->all());

        $request->validate([

            'start\_date' => 'required|date',

            'end\_date' => 'required|date|after:start\_date',

            'status' => 'required|in:ongoing,completed,canceled',

        ]);

        $start = \Carbon\Carbon::createFromFormat('m/d/Y', $request->start\_date);

        $end = \Carbon\Carbon::createFromFormat('m/d/Y', $request->end\_date);

        //  $difference = $start->diffInDays($end);

         $cars = Car::find($request->car\_id);

         $total\_cost=$cars->daily\_rent\_price\*$start->diffInDays($end);

         Rental::create([

            'user\_id' => $request->user\_id,

            'car\_id' => $request->car\_id,

            'start\_date' => $start,

            'end\_date' => $end,

            'total\_cost' => $total\_cost,

            'status' => $request->status,

        ]);

        toastr()->success('Rental Add SuccessFully !!');

        return redirect()->route('admin.rentals.index');

    }

1. **Web route:**

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

 Route::post('/rentals/store', [RentalController::class, 'store'])->name('admin.rentals.store');

});

1. **Route add**

 <form action="{{ route('admin.rentals.store') }}" method="POST">

 @csrf

**Step 1: ALL view Update**

**controller: Admin/ RentalController.php**

1. **flowing index method(update)**

public function index(){

        $rentals = Rental::with(['user', 'car'])->latest()->get();

        return view('dashboard.admin.rentals.index', compact('rentals'));

    }

1. **View :** [**Dashboard/admin/rentals/index.blade.php**](file:///C:\xampp\htdocs\car-rental-system\resources\views\dashboard\admin\rentals\index.blade.php)

**Step 4: Data Delete**

1. **flowing destroy method**

public function destroy(string $id){

        Rental::where('id',$id)->delete();

        toastr()->success('Rentals Delete SuccessFully !!');

        return redirect()->route('admin.rentals.index');

    }

1. **Web route:**

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

Route::delete('/rentals/{id}', [RentalController::class, 'destroy'])->name('admin.rentals.destroy');

});

1. [**Modal add**](resources/views/dashboard/admin/rentals/index.blade.php) **:**

**Madal delete +**

1. **Route add:**

 <form action="{{ route(admin.rentals.destroy,$customer->id) }}" method="post">

**Step 4: Edit view**

1. **flowing edit method**

public function edit(string $id){

        $users = User::all();

        $cars = Car::all();

        $rental = Rental::findOrFail($id);

        return view('dashboard.admin.rentals.edit', compact('rental', 'users', 'cars'));

    }

1. **View :** [**Dashboard/admin/rentals/edit.blade.php**](resources/views/dashboard/admin/rentals/edit.blade.php)
2. **Web route.**

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

Route::get('/rentals/{id}/edit', [RentalController::class, 'edit'])->name('admin.rentals.edit');

});

1. **Route add:** **Dashboard/admin/customers/index.blade.php**

 <a href="{{ route('admin.rentals.edit',$rental->id) }}" class="edit"><i class="material-icons" data-toggle="tooltip" title="Edit">&#xE254;</i></a>

**Step 5: Edit Update**

1. **flowing update method**

 public function update(Request $request, string $id){

        //  dd($request->all());

        $start = \Carbon\Carbon::createFromFormat('Y-m-d', $request->start\_date);

        $end = \Carbon\Carbon::createFromFormat('Y-m-d', $request->end\_date);

        // Find Car & Check

        $car = Car::find($request->car\_id);

        if (!$car) {

            return back()->withErrors(['car\_id' => 'Selected car not found!']);

        }

        // Calculate Total Cost

        $total\_cost = $car->daily\_rent\_price \* $start->diffInDays($end);

        // Validation

        $request->validate([

            'start\_date' => 'required|date',

            'end\_date' => 'required|date|after:start\_date',

            'status' => 'required|in:ongoing,completed,canceled',

        ]);

        // Update Rental

        Rental::where('id', $id)->update([

            'user\_id' => $request->input('user\_id'),

            'car\_id' => $request->input('car\_id'),

            'start\_date' => $request->input('start\_date'),

            'end\_date' => $request->input('end\_date'),

            'status' => $request->input('status'),

            'total\_cost' => $total\_cost,

        ]);

        toastr()->success('Customer updated successfully!');

        return redirect()->route('admin.rentals.index');

    }

1. **Web route:**

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

Route::put('/rentals/{id}/update', [RentalController::class, 'update'])->name('admin.rentals.update');

});

1. **Route add: Dashboard/admin/customers/edit.blade.php**

Route::put('/rentals/{id}/update', [RentalController::class, 'update'])->name('admin.rentals.update');

1. **flowing index method**
2. **View**
3. **Web route.**
4. **Route add**
5. **flowing destroy method**
6. **View**
7. **Web route.**
8. **Route add**

**Image store :**

**Storage Link Create**

যদি ইমেজ **storage** ফোল্ডারে সেভ হয়, তাহলে লারাভেলের **public/storage** লিংক তৈরি করুন:

bash

CopyEdit

php artisan storage:link