To implement login history tracking for users in Laravel, you need to modify your setup to log user details like name, role, email, login\_time, and logout\_time into a separate table. Here’s how you can adjust your LoginController, add a LoginHistory model, and create the necessary migration for the login\_histories table.

**1. Create Migration for login\_histories Table**

First, create a migration for the login\_histories table:

bash

Copy code

php artisan make:migration create\_login\_histories\_table --create=login\_histories

In the generated migration file, define the schema for the table:

php

Copy code

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

/\*\*

\* Run the migrations.

\*/

public function up(): void

{

Schema::create('login\_histories', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('email');

$table->string('role');

$table->enum('status', ['active', 'inactive']); // Track user status

$table->timestamp('login\_time')->nullable();

$table->timestamp('logout\_time')->nullable();

$table->timestamps(); // created\_at and updated\_at fields

});

}

/\*\*

\* Reverse the migrations.

\*/

public function down(): void

{

Schema::dropIfExists('login\_histories');

}

};

Run the migration:

bash

Copy code

php artisan migrate

**2. Create the LoginHistory Model**

Create a model to interact with the login\_histories table:

bash

Copy code

php artisan make:model LoginHistory

In the generated LoginHistory.php file, define the fillable fields:

php

Copy code

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class LoginHistory extends Model

{

use HasFactory;

protected $fillable = [

'name',

'email',

'role',

'status',

'login\_time',

'logout\_time',

];

}

**3. Update LoginController for Login Tracking**

Next, modify the LoginController to log user information when they successfully log in.

In your authenticate() method, after a successful login (Auth::attempt()), add logic to record the login information in the login\_histories table:

php

Copy code

use App\Models\LoginHistory;

use Illuminate\Support\Facades\Auth;

// This will authenticate the User

public function authenticate(Request $request) {

$validator = $request->validate([

'email'=>'required|email',

'password'=>'required',

]);

if ($validator) {

if (Auth::attempt(['email' => $request->email, 'password' => $request->password])) {

// Get the authenticated user

$user = Auth::user();

// Log the login event in login\_histories table

LoginHistory::create([

'name' => $user->name,

'email' => $user->email,

'role' => $user->role,

'status' => 'active',

'login\_time' => now(), // Record the current login time

]);

return redirect()->route('account.dashboard')->with('success', 'User Logged in Successfully');

} else {

return redirect()->route('account.login')->with('error', 'Either email or password is incorrect');

}

} else {

return redirect()->route('account.login')->withInput()->withErrors($validator);

}

}

**4. Track Logout Events**

Update the logout() method to mark the user as inactive and record the logout time.

Modify the logout() method like this:

php

Copy code

use App\Models\LoginHistory;

use Illuminate\Support\Facades\Auth;

public function logout(Request $request) {

$user = Auth::user();

// Update the login history to mark the user as inactive and log logout time

LoginHistory::where('email', $user->email)

->where('status', 'active') // Find the active session

->latest() // Get the most recent login record

->first()

->update([

'status' => 'inactive',

'logout\_time' => now(), // Record the current logout time

]);

Auth::logout();

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

return redirect()->route('account.login')->with('success', 'User logout Successfully');

}

**Admin**

<?php

namespace App\Http\Controllers\admin;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\LoginHistory; // Import the LoginHistory model

class LoginController extends Controller

{

public function index() {

return view('admin.login');

}

// Authenticate the Admin

public function authenticate(Request $request) {

$validator = $request->validate([

'email' => 'required|email',

'password' => 'required',

]);

if ($validator) {

if (Auth::guard('admin')->attempt(['email' => $request->email, 'password' => $request->password])) {

// Check if user is not an admin

if (Auth::guard('admin')->user()->role != 'admin') {

Auth::guard('admin')->logout();

return redirect()->route('admin.login')->with('error', 'You are not authorized to view the admin dashboard');

}

// Log the admin login to login\_histories table

$admin = Auth::guard('admin')->user();

LoginHistory::create([

'name' => $admin->name,

'email' => $admin->email,

'role' => $admin->role, // Fetch the role dynamically

'status' => 'active',

'login\_time' => now(), // Log the current login time

]);

return redirect()->route('admin.dashboard')->with('success', 'Admin Logged in Successfully');

} else {

return redirect()->route('admin.login')->with('error', 'Either email or password is incorrect');

}

} else {

return redirect()->route('admin.login')->withInput()->withErrors($validator);

}

}

// Admin logout

public function logout(Request $request) {

$admin = Auth::guard('admin')->user();

// Update the login\_histories table to set the status as inactive on logout

LoginHistory::where('email', $admin->email)

->where('status', 'active') // Find the active session

->latest() // Get the most recent login record

->first()

->update([

'status' => 'inactive',

'logout\_time' => now(), // Log the current logout time

]);

// Perform the logout

Auth::guard('admin')->logout();

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

return redirect()->route('admin.login')->with('success', 'Admin logout Successfully');

}

}