**API PHP**

**Step 1: Install Laravel and Set Up Passport:**

composer create-project --prefer-dist laravel/laravel student-api

cd student-api

composer require laravel/passport

php artisan migrate

php artisan passport:install

* **Add Passport to config/auth.php:**

'guards' => [

'api' => [

'driver' => 'passport',

'provider' => 'users',

],

],

* **Register Passport in App\Providers\AuthServiceProvider.php:**

use Laravel\Passport\Passport;

public function boot() {

$this->registerPolicies();

Passport::routes();

}

* **Step 2: Create Student Model and Migration:**

Command to create model:--- php artisan make:model Student -m

**database/migrations/create\_students\_table.php**

public function up() {

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

$table->id();

$table->string('name');

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

$table->string('phone');

$table->string('course');

$table->timestamps();

});

}

Command to migrate:--- php artisan migrate

* **Step 3: Create API Controller:**

Command to create controller:-- php artisan make:controller Api\StudentController

**app/Http/Controllers/Api/StudentController.php**

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use App\Models\Student;

use Illuminate\Support\Facades\Validator;

class StudentController extends Controller {

* Create Student Record API (POST):

public function store(Request $request) {

$validator = Validator::make($request->all(), [

'name' => 'required|string|max:255',

'email' => 'required|email|unique:students',

'phone' => 'required|string',

'course' => 'required|string',

]);

if ($validator->fails()) {

return response()->json(['error' => $validator->errors()], 400);

}

$student = Student::create($request->all());

return response()->json(['message' => 'Student Created Successfully', 'data' => $student], 201);

}

* 2. Fetch All Students API (GET):

public function index() {

return response()->json(Student::all(), 200);

}

* 3. Fetch Single Student API (GET):

public function show($id) {

$student = Student::find($id);

if (!$student) {

return response()->json(['error' => 'Student Not Found'], 404);

}

return response()->json($student, 200);

}

* 4. Update Student API (PUT):

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

$student = Student::find($id);

if (!$student) {

return response()->json(['error' => 'Student Not Found'], 404);

}

$student->update($request->all());

return response()->json(['message' => 'Student Updated Successfully', 'data' => $student], 200);

}

* 5. Delete Student API (DELETE):

public function destroy($id) {

$student = Student::find($id);

if (!$student) {

return response()->json(['error' => 'Student Not Found'], 404);

}

$student->delete();

return response()->json(['message' => 'Student Deleted Successfully'], 200);

}

}

* **Step 4: Create API Routes**

**routes/api.php**

use App\Http\Controllers\Api\StudentController;

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

Route::post('/students', [StudentController::class, 'store']);

Route::get('/students', [StudentController::class, 'index']);

Route::get('/students/{id}', [StudentController::class, 'show']);

Route::put('/students/{id}', [StudentController::class, 'update']);

Route::delete('/students/{id}', [StudentController::class, 'destroy']);

});