**Laravel and React Full-Stack Guide**

**1. Setting Up Laravel**

**Installing Laravel**

**To create a new Laravel project, run:**

**composer create-project --prefer-dist laravel/laravel myproject**

**Navigate into your project:**

**cd myproject**

**Start the development server:**

**php artisan serve  
  
Install NPM packages:**

**npm install**

**Run NPM:  
npm run dev**

**Switching to MySQL Database**

**Edit .env file:**

**DB\_CONNECTION=mysql**

**DB\_HOST=127.0.0.1**

**DB\_PORT=3306**

**DB\_DATABASE=mydatabase**

**DB\_USERNAME=root**

**DB\_PASSWORD=yourpassword**

**Run migration to apply database changes:**

**php artisan migrate**

**Publish config.php:  
php artisan config: publish**

**composer require laravel/breeze --dev**

**php artisan breeze:install**

**composer require laravel/sanctum**

**composer require fruitcake/laravel-cors --with-all-dependencies**

**2. Laravel API Development**

**Creating a Model, Controller, and Migration**

**php artisan make:model Movie -mcr**

**This creates:**

* **app/Models/Movie.php (Model)**
* **database/migrations/xxxx\_xx\_xx\_create\_movies\_table.php (Migration)**
* **app/Http/Controllers/MovieController.php (Controller)**

**Define table fields in the migration:**

**public function up()**

**{**

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

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

**$table->string('title');**

**$table->text('description');**

**$table->timestamps();**

**});**

**}**

**Run migrations:**

**php artisan migrate**

**Create Factory for populating fake data:**

**php artisan make:factory StudentFactory --model=Student**

**Example:  
<?php**

**namespace Database\Factories;**

**use Illuminate\Database\Eloquent\Factories\Factory;**

**class StudentFactory extends Factory**

**{**

**public function definition(): array**

**{**

**return [**

**'student\_id' => $this->faker->unique()->uuid,**

**'firstname' => $this->faker->firstName,**

**'lastname' => $this->faker->lastName,**

**'age' => $this->faker->numberBetween(6, 18),**

**'address' => $this->faker->address,**

**'school' => $this->faker->company . ' School',**

**'grade' => 'Grade ' . $this->faker->numberBetween(1, 12),**

**'parent\_name' => $this->faker->name,**

**'email' => $this->faker->unique()->safeEmail,**

**'phoneNumber' => $this->faker->phoneNumber,**

**'proof' => 'proofs/sample.pdf', // Or leave empty if not needed for test**

**'date' => $this->faker->date(),**

**];**

**}**

**}**

**Create a Seeder (optional but recommended):**

**php artisan make:seeder StudentSeeder**

**Then edit database/seeders/StudentSeeder.php:**

**use App\Models\Student;**

**public function run(): void**

**{**

**Student::factory()->count(50)->create(); // creates 50 fake students**

**}**

**Run the seeder:**

**Add it to DatabaseSeeder.php:**

**public function run(): void**

**{**

**$this->call(StudentSeeder::class);**

**}**

**php artisan db:seed**

**Add migration file:  
php artisan make:migration create\_students\_table**

**Defining Routes**

**Edit routes/api.php:**

**use App\Http\Controllers\MovieController;**

**use Illuminate\Support\Facades\Route;**

**Route::apiResource('movies', MovieController::class);**

**View all registered routes:**

**php artisan route:list**

**3. Authentication with Laravel Sanctum**

**Install Sanctum**

**composer require laravel/sanctum**

**Publish and run the migration:**

**php artisan vendor:publish --provider="Laravel\Sanctum\SanctumServiceProvider"**

**php artisan migrate**

**Add Sanctum middleware in app/Http/Kernel.php:**

**'api' => [**

**\Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,**

**'throttle:api',**

**\Illuminate\Routing\Middleware\SubstituteBindings::class,**

**],**

**Creating Authentication Routes**

**Edit routes/api.php:**

**use App\Http\Controllers\AuthController;**

**Route::post('register', [AuthController::class, 'register']);**

**Route::post('login', [AuthController::class, 'login']);**

**Route::middleware('auth:sanctum')->post('logout', [AuthController::class, 'logout']);**

**Handling CORS (Cross-Origin Requests)**

**Edit app/Http/Middleware/HandleCors.php and update:**

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

**{**

**return $next($request)**

**->header('Access-Control-Allow-Origin', '\*')**

**->header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS')**

**->header('Access-Control-Allow-Headers', 'Content-Type, Authorization');**

**}**

**To view routes:  
php artisan route:list**

**4. React Frontend Setup**

**Create React App**

**npx create-react-app myapp**

**cd myapp**

**npm start**

**Fetching Data from Laravel API**

**Install Axios:**

**npm install axios**

**Example API call:**

**import axios from 'axios';**

**const fetchMovies = async () => {**

**const response = await axios.get('http://127.0.0.1:8000/api/movies');**

**console.log(response.data);**

**};**

**Handling Authentication in React**

**Example login function:**

**const login = async (email, password) => {**

**const response = await axios.post('http://127.0.0.1:8000/api/login', { email, password });**

**localStorage.setItem('token', response.data.token);**

**};**

**Attach token to requests:**

**axios.defaults.headers.common['Authorization'] = `Bearer ${localStorage.getItem('token')}`;**

**5. Pagination**

**Modify API call in Laravel:**

**public function index()**

**{**

**return Movie::where('user\_id', Auth::id())->paginate(5);**

**}**

**Frontend pagination:**

**const fetchMovies = async (page) => {**

**const response = await axios.get(`http://127.0.0.1:8000/api/movies?page=${page}`);**

**setMovies(response.data.data);**

**setCurrentPage(response.data.current\_page);**

**setLastPage(response.data.last\_page);**

**};**

**6. Factories and Seeders**

**Creating a Factory**

**php artisan make:factory MovieFactory --model=Movie**

**Edit database/factories/MovieFactory.php:**

**public function definition()**

**{**

**return [**

**'title' => $this->faker->sentence,**

**'description' => $this->faker->paragraph,**

**];**

**}**

**Run factory:**

**php artisan tinker**

**Movie::factory()->count(10)->create();**

**7. Deploying Laravel and React**

**Deploying Laravel**

* **Use services like Heroku, DigitalOcean, or Laravel Forge.**
* **Run migrations in production:**

**php artisan migrate --force**

**Deploying React**

**npm run build**

**Host with Vercel, Netlify, or Firebase Hosting.**

**8. Helpful Documentation**

* **[Laravel Docs](https://laravel.com/docs)**
* **[React Docs](https://reactjs.org/docs/getting-started.html)**
* **[Sanctum Authentication](https://laravel.com/docs/10.x/sanctum)**
* **[Axios for React](https://axios-http.com/docs/intro)**

**This document should cover everything you need for a complete Laravel + React integration! 🚀**