# ✅ Project Instructions: Dynamic Process Flow System for Laravel App with Spatie Permissions

## 📘 Project Overview:

You are assisting with the \*\*integration of a dynamic, reusable process flow module\*\* into an existing Laravel application. The app uses [Spatie Laravel Permissions](https://spatie.be/docs/laravel-permission/v5/introduction) and dynamic route-based middleware (based on [this CodeAndDeploy tutorial](https://codeanddeploy.com/blog/laravel/how-to-create-a-custom-dynamic-middleware-for-spatie-laravel-permission)).

## 🎯 Goal:

To implement a \*\*model-agnostic dynamic process flow system\*\* that supports tracking and managing step-based workflows across key models.

## 📌 Key Requirements:

1. \*\*Target Models\*\*:

- `PsipName`: Represents a public sector investment project.

- `Activity`: Represents activities under each `PsipName`.

2. \*\*Process Flow Behavior\*\*:

- Each `Activity` must have its own independent process flow.

- Each `PsipName` may have a process flow (optional, but supported).

- A `PsipName`’s process flow is marked as \*\*complete automatically\*\* only when \*\*all related `Activity` items are completed\*\*.

- Future support for `SubActivity` or other models must be considered.

3. \*\*Model-Agnostic Design\*\*:

- Use polymorphic relationships (`model\_type`, `model\_id`) in relevant tables to support any Eloquent model.

4. \*\*Database Tables\*\* (expected or to be created):

- `process\_flows`: Holds definition metadata and model association.

- `process\_steps`: Steps within a flow.

- `process\_conditions`: Logic that determines if a step can proceed.

- `process\_actions`: Actions triggered when steps complete.

- `process\_instances`: Tracks progress per model instance.

- `process\_logs`: Logs transitions and outcomes.

5. \*\*User Interaction\*\*:

- A UI element (e.g. button) must be available to manually complete the current step.

- Once clicked, the system automatically:

- Checks conditions.

- Executes actions.

- Moves to the next step.

- Optionally updates parent model flows if applicable (e.g., PsipName).

6. \*\*Permissions\*\*:

- All flow-related routes must use Spatie’s dynamic middleware (`permission`) based on route names.

- Ensure only authorized users can advance steps.

## 📎 Future-Proofing:

- The system should be designed so that any future Eloquent model (like `SubActivity`) can plug into the same flow mechanism without new logic being written.

## 🧪 Testing:

- Support both unit and feature tests for:

- Advancing a step.

- Completing flows.

- Linking model states.

- Permission-blocked access.

## 🧭 Starting Prompt for New Chat

You can paste this to start a new chat and begin implementation:

\*\*Prompt to Paste in New Chat:\*\*

**```**

I am building a Laravel 10 application that uses Spatie Laravel Permissions and a dynamic middleware pattern based on this tutorial:

https://codeanddeploy.com/blog/laravel/how-to-create-a-custom-dynamic-middleware-for-spatie-laravel-permission

Now I want to build a dynamic, model-agnostic process flow system inside this application. The system should track workflows for Eloquent models. Here’s what I want:

1. The models `PsipName` and `Activity` must be able to have process flows.

2. `Activity` must always have its own process flow.

3. `PsipName` may have one, and its flow should be marked “completed” automatically when all of its `Activity` items are completed.

4. The process flow must support other models in the future (e.g. `SubActivity`), so please design with polymorphic relationships in mind.

5. There must be a button (or UI placeholder) that allows users to manually complete a step and advance the process.

6. The flow must consist of: steps, conditions, actions, and logs.

7. Each process instance must be stored (per model), and every step must be loggable.

8. Use Spatie’s dynamic middleware (`permission`) to restrict advancing a step based on route name.

Let’s begin by designing the database schema and relationships. Please generate migrations for:

- process\_flows

- process\_steps

- process\_conditions

- process\_actions

- process\_instances

- process\_logs

Ensure these use polymorphic relationships to associate flows with any Eloquent model.

**```**