Dynamic Task Widget

Estimated Time: 4–6 hours
Goal: Analyze the candidate's capacity to create a tiny full-stack
feature by: Database -> Laravel API -> Blade + AJAX

Section 1: Back End (Laravel)

1) Create a tasks table
Create a migration with the following columns:

- title (string)
- is_completed (boolean, default = false)
- user_id (foreign key)

2) Task Model

Build a model called **Task** related as follows:

```
public function user() {
   return $this->belongsTo(User::class);
}
```

3) Routes plus Controller
Build a three-method Task Controller:

GET / tasks
Only the logged-in user's tasks are to be returned.

POST /tasks
Make a new task (title is needed).

PUT /tasks/ {task}
Switch the task status—completed or not.

All paths have to be protected using authentication middleware.

Part 2: Front End (Blade + AJAX)

Inside dashboard.blade.php:
Include a straightforward widget called: **"My Tasks"**

The widget ought to have:
A to-do checklist
A basic format to create a new task:

- o Input: title
- o Button: Add

Required AJAX interactions:

On page load: Perform AJAX **GET /tasks** and display the tasks.

When adding a task: Send AJAX **POST /tasks**, then add the task to the list ***without reloading***.
When clicking a checkbox: Send AJAX **PUT /tasks/{task}`** to update the task status, and visually update the task (e.g., **grey + line-through** when completed).

Basic Requirements

- Use Bootstrap or Tailwind (your choice)
- No page reload
- Clean and simple code
- Submit the project on **GitHub**
  Required Demo Video (1–3 minutes) Upload a short MP4 video inside the GitHub repo showing:
  1. Running the project locally (php artisan serve + dashboard page)
  2. Viewing the “My Tasks” widget
  3. Adding a new task using the form (AJAX POST)
  4. Seeing the new task appear instantly (no page reload)
  5. Toggling task completion using the checkbox (AJAX PUT)
  6.The visual update (e.g., grey text + line-through when completed)

Thank you