Mejora la Plataforma Escolar con Tailwind y una Página de Inicio

Ya tenemos la base funcional de nuestro CRUD para gestionar proyectos. Ahora, es momento de **hacer que la plataforma sea atractiva visualmente** y agregar un **contexto claro** para que cualquiera que la use entienda de qué trata.

La actividad será:

1. **Integrar Tailwind CSS** para transformar el diseño básico en algo profesional y moderno.
2. Crear una página de inicio (**home**) que explique el propósito de la plataforma escolar que estamos construyendo.
3. Añadir un sistema de autenticación con Jetstream y configuraremos roles básicos:
    - Estudiantes: Podrán ver y subir sus proyectos.
    - Administradores: Tendrán acceso completo al CRUD de proyectos.

El objetivo es que la página sea clara, visualmente atractiva y ayude a los usuarios a entender la importancia de la plataforma.

Instrucciones:

1. **Integrar Tailwind CSS en el proyecto:**
    - Sigue los pasos para instalar Tailwind CSS en Laravel.
    - Aplica clases de Tailwind a los elementos de las vistas index, create y layout, mejorando la estructura visual y adaptándola a un diseño más moderno.
2. **Crear la página de inicio (Home):**
    - Crea una nueva vista llamada home.blade.php.
    - Añade un diseño atractivo que explique:
        - **Qué es la plataforma:** Una herramienta para gestionar proyectos asignados a los usuarios.
        - **Qué se puede hacer:** Crear, ver, editar y eliminar proyectos.
        - **Por qué es útil:** Facilita la organización y colaboración en el ámbito escolar.

- Usa **imágenes, colores y tipografías** para que el diseño sea más dinámico.

3. **Actualizar el sistema de rutas:**
   - Cambia la ruta raíz (/) para que dirija a la nueva página de inicio.
   - Añade un botón en la barra de navegación que permita volver a la página principal desde cualquier sección.

Agregar Jetstream con Login y Roles Básicos

✓ Instalar Jetstream:

- Ejecuta en la terminal:

```
composer require laravel/jetstream
php artisan jetstream:install livewire
npm install && npm run dev
php artisan migrate
```

✓ Añadir roles:

- Crea una migración para el atributo role en la tabla users:

```
php artisan make:migration add_role_to_users_table --table=users
```

- Define la columna en la migración:

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('role')->default('student');
    });
}
```

- Ejecuta la migración:

```
php artisan migrate
```

✓ Sembrar usuarios de ejemplo:

- Actualiza el seeder en DatabaseSeeder.php:

```
User::factory()->create([
    'name' => 'Admin User',
    'email' => 'admin@example.com',
    'password' => bcrypt('password'),
    'role' => 'admin',
]);

User::factory()->create([
    'name' => 'Student User',
    'email' => 'student@example.com',
    'password' => bcrypt('password'),
    'role' => 'student',
]);
```

- Ejecuta el seeder:

```
php artisan db:seed
```

✓ Crear middleware de roles:

- Crea el middleware:

```
php artisan make:middleware RoleMiddleware
```

- Define la lógica en RoleMiddleware.php:

```
public function handle($request, Closure $next, $role)
{
    if (auth()->check() && auth()->user()->role === $role) {
        return $next($request);
    }
    return redirect('/'); // Redirige si no tiene el rol
}
```

- Registra el middleware en Kernel.php:

```
protected $routeMiddleware = [
    'role' => \App\Http\Middleware\RoleMiddleware::class,
];
```

✓ Proteger rutas:

- Actualiza las rutas en web.php:

```
Route::middleware('role:admin')->group(function () {
    Route::resource('projects', ProjectController::class);
});

Route::middleware('role:student')->group(function () {
    Route::get('projects', [ProjectController::class, 'index'])-
>name('projects.index');
    Route::post('projects', [ProjectController::class, 'store'])-
>name('projects.store');
});
```

✓ Actualizar vistas:

- Usa condicionales en Blade para mostrar contenido según el rol:

```
@if(auth()->user()->role === 'admin')
    <a href="{{ route('projects.create') }}" class="btn btn-
primary">Create New Project</a>
@endif
```