Settings Module (multilingual)

Below are **complete, ready-to-paste** snippets to implement a multilingual Settings module with the DB structure you described (`settings` + `setting_translations`). The page lives at `view('admin.settings.settings.index')` and follows the same dashboard style as your Languages module.

---

1) Routes (add to `routes/web.php` in the Admin group)

```
use App\Http\Controllers\Admin\SettingsController;

Route::prefix('admin/settings')->name('admin.settings.')-
>middleware(['auth'])
    ->group(function () {
        // Settings main page
        Route::get('/main', [SettingsController::class, 'index'])-
>name('main');
        // Save settings
        Route::post('/main', [SettingsController::class, 'save'])-
>name('save');
    });
```

The redirect in the controller uses `route('admin.settings.main')`.

---

2) Controller: `app/Http/Controllers/Admin/SettingsController.php`

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Setting;
use App\Models\Language;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class SettingsController extends Controller
{
    /**
     * Show settings form with multilingual tabs for translatable fields.
     */
    public function index()
    {
```

```
        // Ensure the baseline keys exist (safe to call anytime)
        $this->ensureDefaultSettingsExist();

        $languages = Language::orderBy('id')->get();
        $settings  = Setting::with('translations')->get()->keyBy('key');

        return view('admin.settings.settings.index', compact('settings',
'languages'));
    }

    /**
     * Persist settings values (handles both translatable & non-
translatable).
     */
    public function save(Request $request)
    {
        $languages  = Language::all();
        $langByCode = $languages->keyBy('code');

        // Validation rules
        $rules = [
            // translatable
            'values.website_name.*' => ['nullable', 'string', 'max:255'],
            'values.description.*'  => ['nullable', 'string'],

            // non-translatable
            'values.color'          => ['nullable', 'regex:/^#?([A-Fa-f0-9]
{6}|[A-Fa-f0-9]{3})$/'],
            'values.email'          => ['nullable', 'email', 'max:255'],
            'values.phone'          => ['nullable', 'string', 'max:50'],
            'values.logo'           => ['nullable', 'image',
'mimes:jpg,jpeg,png,webp,svg', 'max:2048'],
        ];

        $validated = $request->validate($rules);

        // Fetch settings keyed by `key` for convenience
        $settings = Setting::all()->keyBy('key');

        // 1) Handle file upload (logo)
        if ($request->hasFile('values.logo')) {
            $path  = $request->file('values.logo')->store('settings',
'public');
            $value = '/storage/' . $path; // store the public URL/path in DB

            $setting = $settings['logo'] ?? Setting::firstOrCreate(['key' =>
'logo'], ['type' => 'image']);

            // replicate the same file path for all languages for simplicity
            foreach ($langByCode as $lang) {
                $setting->translations()->updateOrCreate(
```

```
                    ['language_id' => $lang->id],
                    ['value'       => $value]
                );
            }
        }

        // 2) Non-translatable (replicate across all languages)
        foreach (['color', 'email', 'phone'] as $key) {
            if ($request->filled("values.$key")) {
                $val = $request->input("values.$key");

                $setting = $settings[$key]
                    ?? Setting::firstOrCreate(
                        ['key' => $key],
                        ['type' => $key === 'color' ? 'color' : 'text']
                    );

                foreach ($langByCode as $lang) {
                    $setting->translations()->updateOrCreate(
                        ['language_id' => $lang->id],
                        ['value'       => $val]
                    );
                }
            }
        }

        // 3) Translatable keys (per-locale inputs)
        foreach (['website_name', 'description'] as $key) {
            if ($request->has("values.$key")) {
                $pairs   = $request->input("values.$key", []);
                $setting = $settings[$key]
                    ?? Setting::firstOrCreate(
                        ['key' => $key],
                        ['type' => $key === 'description' ? 'textarea' :
'text']
                    );

                foreach ($pairs as $locale => $val) {
                    if (!isset($langByCode[$locale])) {
                        continue; // ignore unknown locale codes
                    }
                    $setting->translations()->updateOrCreate(
                        ['language_id' => $langByCode[$locale]->id],
                        ['value'       => $val]
                    );
                }
            }
        }

        return redirect()
            ->route('admin.settings.main')
```

```
            ->with('success', __('dashboard.settings_saved'));
    }

    /**
     * Create baseline setting rows if missing so the view always has them.
     */
    protected function ensureDefaultSettingsExist(): void
    {
        $defaults = [
            'logo'         => 'image',
            'website_name' => 'text',
            'description'  => 'textarea',
            'color'        => 'color',
            'email'        => 'email',
            'phone'        => 'text',
        ];

        foreach ($defaults as $key => $type) {
            Setting::firstOrCreate(['key' => $key], ['type' => $type]);
        }
    }
}
```

This controller assumes you already have `Setting` and `SettingTranslation` models similar to what you attached (with `translations()` / `belongsTo` relations and a helper like `getValue($locale = null)`).

---

3) View: `resources/views/admin/settings/settings/index.blade.php`

```
@extends('layouts.backend.app')

@section('title', __('dashboard.settings'))

@section('content')
<div class="page-header">
    <h3 class="page-title">{{ __('dashboard.settings') }}</h3>
    <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
            <li class="breadcrumb-item">
                <a
href="{{ route('admin.settings.main') }}">{{ __('dashboard.settings') }}</a>
            </li>
            <li class="breadcrumb-item active" aria-
current="page">{{ __('dashboard.main_settings') }}</li>
        </ol>
    </nav>
</div>
```

```
@if(session('success'))
    <div class="alert alert-success">{{ session('success') }}</div>
@endif

<div class="col-md-12 grid-margin stretch-card">
  <div class="card">
    <div class="card-body">
      <h4 class="card-title mb-4">{{ __('dashboard.main_settings') }}</h4>

      <form class="forms-sample" action="{{ route('admin.settings.save') }}"
method="POST" enctype="multipart/form-data">
        @csrf

        {{-- Non-translatable settings
-------------------------------------}}
        <div class="row">
            {{-- Logo --}}
            <div class="col-md-6 mb-4">
                <label class="form-label">{{ __('dashboard.logo') }}</label>
                <div class="d-flex align-items-center gap-3">
                    @php($logo = optional($settings['logo'] ?? null)-
>getValue())
                    @if($logo)
                        <img src="{{ $logo }}" alt="logo" style="height:48px;
width:auto; border:1px solid #eee; padding:4px; border-radius:6px;">
                    @endif
                    <input type="file" name="values[logo]" class="form-
control" accept="image/*">
                </div>
                @error('values.logo') <div class="text-danger">{{ $message }}
</div> @enderror
            </div>

            {{-- Color --}}
            <div class="col-md-3 mb-4">
                <label class="form-label">{{ __('dashboard.main_color') }}</
label>
                <input type="color" class="form-control form-control-color"
name="values[color]"
                       value="{{ old('values.color',
optional($settings['color'] ?? null)->getValue()) }}">
                @error('values.color') <div class="text-
danger">{{ $message }}</div> @enderror
            </div>

            {{-- Email --}}
            <div class="col-md-6 mb-4">
                <label class="form-label">{{ __('dashboard.email') }}</label>
                <input type="email" name="values[email]" class="form-control"
                       value="{{ old('values.email',
```

```
optional($settings['email'] ?? null)->getValue()) }}">
                @error('values.email') <div class="text-
danger">{{ $message }}</div> @enderror
            </div>

            {{-- Phone --}}
            <div class="col-md-6 mb-4">
                <label class="form-label">{{ __('dashboard.phone') }}</label>
                <input type="text" name="values[phone]" class="form-control"
                       value="{{ old('values.phone',
optional($settings['phone'] ?? null)->getValue()) }}">
                @error('values.phone') <div class="text-
danger">{{ $message }}</div> @enderror
            </div>
        </div>

        <hr class="my-4" />

        {{-- Translatable settings (language tabs)
--------------------------}}
        <ul class="nav nav-tabs" id="settingsTabs" role="tablist">
            @foreach($languages as $i => $lang)
                <li class="nav-item" role="presentation">
                    <button class="nav-link {{ $i===0 ? 'active' : '' }}"
id="tab-{{ $lang->code }}"
                            data-bs-toggle="tab" data-bs-target="#pane-
{{ $lang->code }}" type="button"
                            role="tab" aria-controls="pane-{{ $lang->code }}"
aria-selected="{{ $i===0 ? 'true' : 'false' }}">
                        {{ $lang->native ?? strtoupper($lang->code) }}
                    </button>
                </li>
            @endforeach
        </ul>
        <div class="tab-content border border-top-0 p-3"
id="settingsTabsContent">
            @foreach($languages as $i => $lang)
                @php
                    $dir = $lang->direction ?? 'ltr';
                    $code = $lang->code;
                @endphp
                <div class="tab-pane fade {{ $i===0 ? 'show active' : '' }}"
id="pane-{{ $lang->code }}" role="tabpanel" aria-labelledby="tab-{{ $lang-
>code }}">
                    <div class="row" dir="{{ $dir }}">
                        {{-- Website Name --}}
                        <div class="col-md-6 mb-4">
                            <label class="form-
label">{{ __('dashboard.website_name') }}</label>
                            <input type="text" class="form-control"
                                   name="values[website_name][{{ $code }}]"
```

```
                                    value="{{ old("values.website_name.$code",
optional($settings['website_name'] ?? null)->translations-
>firstWhere('language_id', $lang->id)->value ?? '') }}">
                                @error("values.website_name.$code") <div
class="text-danger">{{ $message }}</div> @enderror
                            </div>

                            {{-- Description --}}
                            <div class="col-md-12 mb-4">
                                <label class="form-
label">{{ __('dashboard.description') }}</label>
                                <textarea class="form-control" rows="4"
                                          name="values[description]
[{{ $code }}]">{{ old("values.description.$code",
optional($settings['description'] ?? null)->translations-
>firstWhere('language_id', $lang->id)->value ?? '') }}</textarea>
                                @error("values.description.$code") <div
class="text-danger">{{ $message }}</div> @enderror
                            </div>
                        </div>
                    </div>
                @endforeach
            </div>

            <div class="mt-4">
                <button type="submit" class="btn btn-
success">{{ __('dashboard.save') }}</button>
                <a href="{{ url()->previous() }}" class="btn btn-
dark">{{ __('dashboard.cancel') }}</a>
            </div>
        </form>
    </div>
  </div>
</div>
@endsection
```

---

4) (Optional) Model helper — `app/Models/Setting.php`

If your `Setting` model does not already include it, add this convenience method to pull the current locale value with fallback:

```
public function getValue($locale = null)
{
    $language = null;

    if ($locale) {
        $language = \App\Models\Language::where('code', $locale)->first();
    } else {
```

```
        // Try to map the current app locale to a Language row
        $current = app()->getLocale();
        $language = \App\Models\Language::where('code', $current)->first();
    }

    if ($language && ($t = $this->translations->firstWhere('language_id',
$language->id))) {
        return $t->value;
    }

    return optional($this->translations->first())->value;
}
```

---

5) Usage notes

- **Translatable fields:** `website_name`, `description` are per-language using tabs.
- **Global fields:** `logo`, `color`, `email`, `phone` are stored once and replicated to all languages (so frontend lookups by current language keep working without special-casing).
- **Storage:** `logo` is saved in `storage/app/public/settings` and referenced via `/storage/...`. Make sure you've run `php artisan storage:link` on the server.
- **Extensibility:** To add more settings later, just create a row in `settings` with the desired `key` and `type`, add its input(s) to the Blade, and update the controller's validation block if needed.

That's it!

Paste the controller, the route definitions, and the Blade view as-is and you'll have a clean, multilingual Settings page under `admin/settings/main` rendering the view `admin.settings.settings.index`.