

Project Update Guide (Latest Version)

This document describes **step-by-step instructions** to update any project to the **latest architecture, helpers, security features, and storage system**. Please follow the steps **carefully and in order** to avoid breaking changes.

1. File Updates

✓ Update the Following Files

1. `public/.htaccess`
2. `.htaccess` (root)
3. `public/index.php`

Replace these files **exactly** with the versions from the example (latest) project.

✗ Delete the Following File

1. `index.php` (root level)

2. Helper Function Updates

Update or replace the following helper functions with the **latest logic** from the example project:

1. `files_asset_path`
2. `get_image`
3. `upload_files_from_path_dynamic`
4. `delete_file`
5. `get_files_path`
6. `get_files_from_fileholder`
7. `delete_files_from_fileholder`
8. `upload_files_from_path_static`
9. `upload_file`
10. `language_file_exists`
11. `get_first_file_from_dir`
12. `saveImageAndGetUrl` (only if exists)
13. `get_file_link`
14. `files_asset_path_basename`
15. `get_asset_url` (add this function if not exists)

⚠ Make sure **old helper logic is completely replaced**, not partially merged.

3. Method Update

Admin Language Import

Update the following methods using the example project logic:

- `Admin\LanguageController -> import()`
 - `Admin\LanguageController -> download()`
-

4. Google reCAPTCHA Integration

4.1 Add Validation Rule

Add Google reCAPTCHA validation logic inside:

- `App\Providers\AppServiceProvider -> boot()`

Add the validation rule **only if it does not already exist**.

4.2 Extension Constant

Add the following constant (if missing):

File: `App\Constants\ExtensionConst`

```
const GOOGLE_RECAPTCHA_SLUG = 'google-recaptcha';
```

4.3 Extension Provider

Add the following provider **if not exists**:

- `App\Providers\Admin\ExtensionProvider`

Copy the full file from the example project.

4.4 Google reCAPTCHA Blade Component

Create the component using Artisan:

```
php artisan make:component Security\\GoogleRecaptchaField
```

This will generate: - Component class - View file

 **Replace both files' contents** with the versions from the example project.

4.5 Validation Message

Add or replace the validation message in:

- lang/en/validation.php

Key: g_recaptcha_verify

Value must exactly match the example project.

4.6 Seeder Update

Update the ExtensionSeeder :

- Add Google reCAPTCHA credentials
- Ensure status = false

 Credentials Source: **Google Doc → "Testing Credentials Google Doc Link → Xmin Recaptcha"**

5. Rate Limiter Feature

5.1 Middleware

1. Copy RateLimiter middleware from the example project
2. Paste it into the same path in your project

5.2 Kernel Registration

Register middleware in:

- App\\Http\\Kernel.php

Add it under the web middleware group.

6. File Handling Updates

6.1 FileController Methods

Update the following methods in:

- `app/Http/Controllers/FileController.php`

Methods to update:

1. `storeFile`
2. `removeFile`

Ensure logic matches the example project and supports the new storage architecture.

6.2 Global File Download URL Update

Update URL logic defined in:

- `routes/global.php`

Previous URL format:

```
{base_url}/file/download/{path_source}/{name}
```

Update **all codes related to this URL**, including: - Route definition - Controller logic - API responses - Frontend references

Refer to the example project for the updated implementation.

7. Storage Settings Feature

6.1 Controller

Add: - `StorageSettingsController`

Path: `App\Http\Controllers\Admin`

6.2 Routes

Add storage-related routes into:

- `routes/admin.php`

Search by controller name in the example project and copy the full route group.

6.3 Views

Copy the entire folder:

```
resources/views/admin/sections/storage-settings
```

Paste it into the same location in your project.

6.4 Service Provider

Copy:

- `App\Providers\StorageServiceProvider`

Register it in:

- `config/app.php` → `providers` array
-

6.5 Admin Navigation Menu

Add **Storage Settings** menu item in the admin side navigation blade.

Search with keyword “**Storage Settings**” in the example project.

6.6 Permissions

Add storage-related permissions in:

- `config/system-role-permissions.php`

Search with “**Storage Settings**” in the example project.

6.7 Database Migration (Storage Settings)

Add the required migration file for storage configuration:

- Copy the migration file from the example project:

```
2025_12_08_055732_add_columns_to_basic_settings_table.php
```

- Paste it into the `database/migrations` directory of your update project
- Ensure the `storage_config` column is populated via `BasicSettingsSeeder.php`
- Refer to the example project for the exact seeder logic

 Do not manually modify database records. Always rely on the seeder for consistency.

7. Installer Update

1. Copy and replace **installer files** from the example project
2. Copy installer assets from:

```
app/resources/installer/src/assets
```

Paste into:

```
public/installer
```

1. Update installer master blade asset links to **match exactly** with the example project
-

8. Asset & Storage Architecture Update

8.1 Asset Function Rules

- `asset()` → **CSS / JS only**
- `get_image()` → **Images & uploaded files**

Remove `public/` prefix from **all** `asset()` usages.

8.2 Storage Migration

1. **Copy (do not move)** all files & asset folders from:

```
public/
```

To:

```
storage/app/public/
```

2. Delete:

3. Image / document folders from `public`

4. JS / CSS / fonts folders from `storage/app/public`

8.3 Storage Linking

Run the following command:

```
php artisan storage:link
```

9. Composer Update

Update `composer.json`:

Required Changes

```
"php": "^8.1.0",  
"laravel/framework": "^10.10",  
"league/flysystem-aws-s3-v3": "^3.30"
```

Then run:

```
composer update
```

10. API Asset Base URL Update

Background

Previously, API responses were using the following function to generate asset base URLs:

```
url("/")
```

This approach is no longer recommended for asset handling.

Required Change

For **all API-related asset URLs**, replace usage of:

```
url("/")
```

With:

```
get_asset_url()
```

Scope

- Applies to **API responses only**
 - Affects image URLs, file URLs, and any asset-related base paths returned from APIs
 - Ensures consistency with the new **storage-based asset architecture**
-

Notes

- `get_asset_url()` must already be defined in helpers
 - Do **not** use `asset()` or `url()` inside API responses
 - This change is mandatory for proper S3 / local storage compatibility
-



Update Reference

For updating the project, please review and follow this file:

- `database/seeder/Update/UpdateFeatureSeeder.php`

This seeder controls feature updates and ensures update consistency across projects.

11. Credentials & Resources



Google Docs:
R1TuxFEjsl8BwHMuWGgjUYooSIUTIWYQi4/edit

<https://docs.google.com/document/d/14y7zUPjb->

Credential Sections

1. **Google reCAPTCHA**
2. Search section: **"Google Recaptcha (Xmin)"**
3. **S3 Storage Credentials**
4. Search using **project name (Crypinvest)**
5. Each project has **unique credentials**
6. If not found, contact your **team leader**



Final Notes

- Always compare with the **example project** before committing
- Clear config & cache after update:

```
php artisan optimize:clear
```



Your project is now fully updated to the latest standard.