MySQL 8.4.8+ Upgrade — Code Impact Report

Repos assessed: VisoAdmin (`mvpold_db`), KaaryamAPI (`mvpold_db_one`)
Current server: MySQL 8.2.0 (WAMP, local dev) **Target:** MySQL 8.4.8+
Scope of this report: does the application *code* in either repo need to change for this upgrade to succeed. Server/ops-only items are listed separately (§3) since they gate the same cutover, but they are not code changes.
Date: 2026-08-24

Verdict

> **No application code changes are required in either repo for this upgrade.**
>
> Everything checked — Laravel DB configuration, PHP/MySQL driver compatibility, migration syntax, raw SQL, schema identifiers, column charsets, and database objects — came back clean in both codebases. The upgrade is blocked only by **two server-side configuration items** (§3), neither of which lives in application code.

1. Methodology

Checked in both repos (VisoAdmin, KaaryamAPI):

- `config/database.php` mysql connection block (charset, collation, `strict`, `engine`)
- `composer.json` / `composer.lock` for PHP version, Laravel version, doctrine/dbal version
- Installed PHP's `pdo_mysql` /mysqlnd version and protocol support
- Every `database/migrations/*.php` file for `->change()`, `renameColumn()`, explicit `ENGINE=`, and other version-sensitive DDL
- All `app`, `database`, `config`, `routes` source for deprecated/removed MySQL syntax: `GROUP BY col ASC/DESC`, `FULLTEXT`, `CHANGE MASTER TO` / `SHOW SLAVE STATUS` (renamed to REPLICA terminology in 8.4), `mysql_upgrade` invocations, `mysql_native_password` references

Checked against the live server (once reachable):

- `SELECT VERSION()`, `SHOW VARIABLES LIKE 'sql_mode'`, `SHOW VARIABLES LIKE 'default_storage_engine'`, `SHOW VARIABLES LIKE 'collation_server'`
- `mysql.user` — accounts and their auth plugins
- `information_schema.COLUMNS` — every column's charset in both databases
- `information_schema.VIEWS` / `TRIGGERS` / `ROUTINES` / `EVENTS` — for any database objects with version-sensitive syntax
- Actual table/column names in both databases cross-referenced against the identifiers MySQL 8 added as reserved words
- `D:\wamp64\bin\mysql\mysql8.2.0\my.ini`, reviewed line by line for startup variables removed in 8.4 (specifically `default_authentication_plugin`, replaced by `authentication_policy`)

2. Code impact findings — by category

#	Category	Verdict	Evidence
1	Charset / collation config	**NO CHANGE**	Both `config/database.php` explicitly set `utf8mb4` / `utf8mb4_unicode_ci`, overriding the server's own default (`utf8mb4_0900_ai_ci`) on every connection — version-independent
2	PHP driver compatibility	**NO CHANGE**	`mysqlnd 8.2.13` bundled with the PHP runtime already speaks `caching_sha2_password` and the MySQL 8.4 wire protocol natively
3	Migration DDL (`->change()`,	**NO CHANGE**	VisoAdmin: zero usages, doctrine/dbal not even installed. KaaryamAPI: 3 already-applied migrations use it — won't re-run, no bearing on the upgrade

	`renameColumn()`)		(dbal pinned at 2.13.0 — worth bumping only for future alter-column migrations)
4	Explicit `ENGINE=` in migrations	**NO CHANGE**	One raw `ENGINE=MyISAM` statement (VisoAdmin's `roles` table) — valid, version-independent SQL, not a compatibility issue (see §4)
5	Deprecated `GROUP BY col ASC/DESC`	**NO CHANGE**	Not present in either repo
6	`FULLTEXT` indexes on MyISAM	**NO CHANGE**	None exist in either repo
7	Replication commands (renamed to REPLICA terms in 8.4)	**NO CHANGE**	Not referenced anywhere in app/database/config/routes in either repo
8	`mysql_upgrade` binary (removed in 8.4)	**NO CHANGE**	Not called from either repo
9	Reserved-word collisions in schema identifiers	**NO CHANGE**	Live cross-check of every table/column name in both databases against MySQL 8's reserved-word additions — zero collisions (`events` table flagged and cleared: never a reserved word)
10	Column-level charset (legacy `utf8` / `utf8mb3`)	**NO CHANGE**	Queried every column in both databases — 100% already `utf8mb4`
11	Views / triggers / stored procedures / events	**NO CHANGE**	None exist in either database
12	Server config file (`my.ini`) startup variables	**FLAGGED — NOT CODE**	Reviewed line by line; no now-removed 8.4 variables present. This is a server file, not part of either repo — out of scope for "code impact," covered in §3

3. Non-code items that still gate the upgrade

These are **not application code** — nothing to change in either repo — but the upgrade will fail or misbehave without them:

3.1 root's auth plugin — connection will fail without this ACTION REQUIRED

Both apps connect via `DB_USERNAME=root`, and that account is currently provisioned with `mysql_native_password`, which MySQL 8.4 disables by default. Confirmed via `my.ini` that this isn't a config-file setting — it's an account-level property.

> **Action:** `ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY '<password>';` before cutover. (Or explicitly re-enable the old plugin on the new server with `--mysql-native-password=ON`, but converting the account is the cleaner fix.)

3.2 sql_mode — currently blank, a fresh install won't be ACTION REQUIRED

The current server's `sql_mode` is empty (`sql_mode=""` in `my.ini`), while a stock MySQL 8.x install defaults to `ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION`. Neither app overrides `sql_mode` at the Laravel connection level (`'strict' => false` in both). At least one existing query relies on `ONLY_FULL_GROUP_BY` being off (`VendorServicesInfo::getServices()` in KaaryamAPI).

> **Action:** carry the current `sql_mode=""` line into the new 8.4.8 install's `my.ini` rather than accepting its out-of-the-box default — unless the strict-mode-dependent queries are fixed first (separate, larger cleanup).

4. Adjacent finding — not required for this upgrade

A related but separate issue was found while investigating: `'engine' => null` in both apps' `config/database.php` means every migration falls back to the server's `default_storage_engine`, explicitly set to `MyISAM` in `my.ini`. This currently affects **54% of VisoAdmin's tables (25/46)** and **41% of KaaryamAPI's tables (18/44)** — no FK enforcement, no transaction support on those tables.

> This is **not a blocker for reaching 8.4.8** — MySQL 8.4 still fully supports MyISAM — so it's out of scope for this report's verdict. Full detail (with complete table lists) is in the companion document `mysql-storage-engine-audit.md` / `.pdf`.

5. Pre-cutover checklist

- ☐ `ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY '<password>';` (§3.1)
- ☐ Set `sql_mode=""` explicitly in the new 8.4.8 `my.ini`, or resolve the strict-mode-dependent queries first (§3.2)
- ☐ After upgrading, re-run `SELECT VERSION();`, `SHOW VARIABLES LIKE 'sql_mode'`, `SHOW VARIABLES LIKE 'default_storage_engine'` and confirm both apps still connect and run
- ☐ No code changes to schedule in either repo — §2 is a closed list, nothing pending