

A S PATEL CATTLE FARM

Laravel 10 Cattle Farm, Dairy & Agriculture Investment Platform

HOSTINGER INSTALLATION + PUTINCODE MARKETPLACE GUIDE

Upload • Database • .env • Laravel setup • Storage • Admin • SEO • SMTP • Security • Marketplace fields

Project stack verified from the supplied source:

Laravel 10.10+, PHP 8.1+, MySQL/MariaDB, Composer dependencies included, Vite frontend tooling, responsive web application.

Marketplace Documentation - v1.0

1. Project Overview

A S PATEL Cattle Farm is a Laravel-based cattle farm, dairy, agriculture and investment-oriented website/platform. The supplied database contains site branding, admin controls, user registration, content pages, SEO data, project/investment content, blogs, contact information, notifications and eCommerce-related configuration.

The current project branding uses the title **A S PATEL Cattle Farm**, INR currency, Asia/Kolkata timezone and an admin prefix of **admin**. The website database also includes SEO-ready pages and agriculture/cattle-farm investment content.

The source package is a Laravel 10 application and requires PHP 8.1 or newer.

2. Server / Software Requirements

Component	Requirement
PHP	8.1+ (required by composer.json)
Framework	Laravel 10.10+
Database	MySQL / MariaDB
Web server	Apache / LiteSpeed on Hostinger; mod_rewrite / .htaccess support
PHP extensions	OpenSSL, PDO MySQL, mbstring, tokenizer, XML, Ctype, JSON, BCMath, fileinfo, cURL, GD/Imagick recommended
Composer	Not always required at runtime because vendor is included, but recommended for maintenance
Node/Vite	Only required if rebuilding frontend assets
HTTPS	Strongly recommended

3. Hostinger Installation - Step by Step

Step 1 - Create or select the domain

Open Hostinger hPanel, select the website/domain, enable SSL and confirm that the domain resolves to the hosting account.

Step 2 - Select PHP version

In hPanel PHP Configuration, select PHP 8.1 or a newer compatible 8.x release. PHP 8.1 is the minimum declared by this project.

Step 3 - Upload project files

Open File Manager and upload the project ZIP into the website's **public_html**. Extract the ZIP. The Laravel project files such as **app**, **bootstrap**, **config**, **routes**, **storage**, **vendor**, **artisan**, **composer.json** and **index.php** must be located at the expected application root.

Do not leave the application one extra folder deep unless you intentionally configure the domain document root for that folder.

Step 4 - Create database

In hPanel go to Databases -> Management and create a MySQL database plus database user. Copy the exact Hostinger database name, username and password.

Step 5 - Import SQL

Open phpMyAdmin, select the new database and import the supplied SQL file **u631869004_aspatel.sql**. Wait for the import to complete without errors.

Step 6 - Create .env

Copy **.env.example** to **.env** and replace all demo/local values with the buyer's own production values. Never publish third-party API secrets from an old environment.

Key	Production example / action
APP_NAME	A S PATEL Cattle Farm
APP_ENV	production
APP_KEY	Generate with <code>php artisan key:generate</code>
APP_DEBUG	false
APP_URL	<code>https://yourdomain.com</code>
APP_TIMEZONE	Asia/Kolkata
DB_CONNECTION	mysql
DB_HOST	127.0.0.1 or localhost
DB_PORT	3306
DB_DATABASE	Hostinger database name
DB_USERNAME	Hostinger database user
DB_PASSWORD	Hostinger database password

Step 7 - Generate application key

If Hostinger Terminal/SSH is available, run **php artisan key:generate**. This creates the Laravel APP_KEY required for encryption and sessions.

Step 8 - Set writable permissions

Laravel requires **storage** and **bootstrap/cache** to be writable. Hostinger normally handles ownership automatically; use safe shared-hosting permissions such as 755 for directories and 644 for files where appropriate.

Step 9 - Storage link

If the application stores public uploads on Laravel's public disk, run **php artisan storage:link**. If a storage link is already present and working, do not recreate it unnecessarily.

Step 10 - Clear and optimize cache

After changing .env, run: **php artisan optimize:clear**. Once the application works, production optimization can use **php artisan config:cache** and **php artisan route:cache** where compatible.

Step 11 - Cron / queue

The project is configured with a database queue in the example environment and includes notification/automation-related settings. If queued emails or scheduled jobs are used, configure Hostinger Cron Jobs and/or a queue worker according to the enabled modules.

Step 12 - Test the website

Test the homepage, registration/login, project pages, contact/enquiry, admin login, images, email, payment/investment flows, user dashboard, mobile layout and HTTPS redirects.

4. Admin and Initial Production Setup

The database contains an admin account and the system uses an admin route prefix. For a marketplace/buyer installation, the buyer should immediately change the admin email, password, phone and profile information after installation.

Review the Admin settings for site title, logo, favicon, colors/theme, currency, registration, notifications, email verification, reCAPTCHA, analytics, cookie settings, SEO, eCommerce and any investment/deposit/withdrawal settings before enabling live users.

5. Email / SMTP

Configure the buyer's own SMTP mailbox in .env or through the system settings. Never ship or reuse old SMTP passwords. Recommended production setup: valid domain mailbox, authenticated SMTP, TLS/SSL, real FROM address and successful test email.

If Hostinger Email is used, enter the SMTP details shown in the buyer's Hostinger Email configuration panel. Do not enable OTP or email verification until a test message succeeds.

6. Security Before Marketplace Distribution

Important: The supplied development package contains an .env.example with sample/API-style values and also includes service credential files. Before selling the source ZIP, remove all real private keys, tokens, passwords, Firebase service credentials, OAuth secrets and stored runtime sessions.

Also clear **storage/framework/sessions**, old logs and cache files. Keep only safe placeholders in .env.example. Buyers should add their own third-party credentials.

Set APP_DEBUG=false in production, enable HTTPS, use strong admin passwords and protect backups/database exports from public access.

7. Recommended PutInCode Listing

Field	Recommended value
Product name	A S PATEL Cattle Farm - Laravel Cattle Farm, Dairy & Agriculture Investment Platform
Technology	Laravel 10, PHP 8.1+, MySQL/MariaDB, Bootstrap, JavaScript, Vite
Version	1.0
Support days	180
Demo URL	https://aspatelcattlefarm.com/
Documentation URL	Leave blank when PDF documentation is uploaded
Source ZIP	Upload a sanitized marketplace source package - not the raw production backup

8. PutInCode Short Description

A professional Laravel 10 cattle farm, dairy and agriculture investment platform with user accounts, investment/project presentation, admin management, SEO controls, blogs, notifications, responsive design and configurable eCommerce features.

9. PutInCode Full Description

A S PATEL Cattle Farm is a complete Laravel-based cattle farm, dairy, agriculture and investment platform designed for businesses that want to present farm projects, cattle and dairy opportunities, build investor/user accounts and manage content from a centralized administration panel.

The platform combines a professional public website with user registration, investment-focused project content, configurable farm and dairy information, responsive pages, blog/content tools, SEO metadata, notification settings, analytics options and an extensive admin configuration system.

Key capabilities include: cattle-farm and agriculture investment presentation, project content, user accounts, admin dashboard, SEO page management, blog system, responsive frontend, configurable registration, email/SMS/push notification options, cookie controls, analytics hooks, currency configuration, eCommerce switch, commission/bonus

settings and multiple integrations available through the Laravel application.

The project is suitable for cattle farms, dairy businesses, agriculture ventures, farm-investment platforms, livestock businesses and related rural/agri-business websites that need a scalable Laravel foundation.

The buyer receives the Laravel source code, database SQL and step-by-step Hostinger installation documentation. Third-party services such as payment gateways, SMS, social login, email providers or cloud storage require the buyer's own credentials and may require separate provider accounts.

10. Media Upload Order

Main cover image: 01_ASPatel_Cattle_Farm_Main_Cover.png

Gallery screenshots: 02 Professional Homepage, 03 Cattle Listing, 04 Cattle Details, 05 Enquiry/Contact, 06 Admin Dashboard, 07 Fully Responsive, 08 Global Livestock Delivery.

Description cards: Wide Range of Cattle Breeds, Powerful Admin Panel, Enquiry & Customer System, Mobile Friendly & Responsive, SEO & Growth Ready.

11. Five Description Cards - Layout and Text

#	Title	Layout	Description
1	Wide Range of Cattle Breeds	Image left	Showcase cows, bulls, calves, dairy cattle and breeding stock through organized farm and livestock
2	Powerful Admin Panel	Image right	Manage the website from one centralized admin area. Control content, pages, blogs, users, settings
3	Enquiry & Customer System	Image left	Create a professional connection between the farm and interested users. Registration, account feat
4	Mobile Friendly & Responsive	Image right	Deliver a clean experience across desktop, tablet and mobile devices. Responsive layouts make im
5	SEO & Growth Ready	Image left	Use configurable page titles, meta keywords, descriptions, OG information, robots controls and ima

12. Troubleshooting

Problem	Check / Fix
HTTP 500	Set PHP 8.1+, verify .env, permissions, APP_KEY and Hostinger error logs; run php artisan optimize:clear.
Database error	Verify DB_HOST, DB_DATABASE, DB_USERNAME, DB_PASSWORD and confirm SQL import finished successfully.
Blank/broken CSS	Check APP_URL, HTTPS, asset paths and whether compiled Vite assets are present.
Images missing	Check storage/public paths, storage:link and file permissions.
Admin route not found	Confirm application routes loaded and use the configured admin prefix.
Email not sending	Use buyer's own SMTP account, correct port/encryption and valid FROM address.
Queue jobs stuck	Run/configure a queue worker or change the queue strategy for the hosting plan.
Old data appears	Use a clean/sanitized database for marketplace buyers and clear sessions/cache.