



OFFICIAL TECHNICAL DOCUMENTATION

# Technical, Operational & Production Handoff Guide

Directory · Evaluation · Recognition

Production Deployment & Administration Manual — a complete reference for understanding, operating, deploying, and maintaining the PLOCAL platform.

## Directory

Nationwide discovery of evaluated local businesses across all 50 U.S. states.

## Evaluation

Structured Digital, Service & Trust assessment for every activated Business.

## Recognition

Business of the Day, Month, and Year — awarded on evaluated merit.

---

Document Version: **1.0**

Issue Date: **September 2, 2026**

Classification: **Owner / Administrator Reference**

**plocal.com**

# Document Information

---

|                  |                                                                                |
|------------------|--------------------------------------------------------------------------------|
| Document Title   | PLOCAL Technical, Operational & Production Handoff Guide                       |
| Product          | PLOCAL — Directory · Evaluation · Recognition Platform                         |
| Document Version | 1.0                                                                            |
| Issue Date       | September 2, 2026                                                              |
| Audience         | Project Owner, Server Administrator, Operating Staff,<br>Maintaining Developer |
| Classification   | Internal / Owner Reference — Not for Public Distribution                       |

## How to use this guide

This document is organized as a reference manual, not a linear read. Use the Table of Contents to jump directly to the topic you need — architecture, daily operation, deployment, or troubleshooting. Every command, route, and environment variable named in this guide has been verified directly against the current PLOCAL codebase at the time of writing.

Sections marked **Owner / Infrastructure Responsibility** describe configuration and infrastructure decisions that must be supplied by the project owner or hosting administrator — they are not missing application features.

No credentials, API keys, or passwords are printed anywhere in this document. Where a secret value is referenced, a placeholder is used instead.

# Table of Contents

---

## PART I — OVERVIEW & ARCHITECTURE

|   |                       |    |
|---|-----------------------|----|
| 1 | Executive Overview    | 3  |
| 2 | System Architecture   | 5  |
| 3 | Repository Structure  | 7  |
| 4 | Backend Architecture  | 8  |
| 5 | Frontend Architecture | 10 |

## PART II — USING PLOCAL

|   |                                           |    |
|---|-------------------------------------------|----|
| 6 | Public Website Navigation                 | 11 |
| 7 | Nationwide Directory & Find Businesses    | 13 |
| 8 | Business Profiles & Contextual Navigation | 15 |
| 9 | Business Submission                       | 17 |

## PART III — PAYMENTS, EVALUATION & RECOGNITION

|    |                                        |    |
|----|----------------------------------------|----|
| 10 | Payments & Stripe                      | 19 |
| 11 | Paid Activation                        | 22 |
| 12 | Evaluation — Digital / Service / Trust | 23 |
| 13 | Public Voting                          | 24 |
| 14 | Recognition                            | 25 |

## PART IV — PORTALS & ADMINISTRATION

|    |                              |    |
|----|------------------------------|----|
| 15 | Business Portal              | 27 |
| 16 | Internal Portal & RBAC       | 29 |
| 17 | Internal Evaluation Workflow | 31 |
| 18 | Payment Operations           | 32 |
| 19 | Support Cases                | 33 |
| 20 | Blog CMS                     | 34 |
| 21 | Advertiser Management        | 35 |

## PART V — TECHNICAL FOUNDATIONS

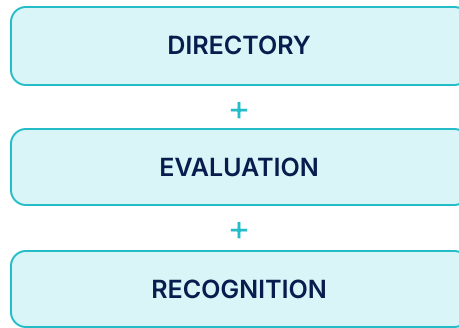
|    |                                 |    |
|----|---------------------------------|----|
| 22 | Database Architecture           | 36 |
| 23 | Authentication & Security Model | 38 |
| 24 | File Storage                    | 39 |

|                                                     |                                     |    |
|-----------------------------------------------------|-------------------------------------|----|
| 25                                                  | Email                               | 40 |
| 26                                                  | Queues & Background Jobs            | 41 |
| <b>PART VI — PRODUCTION DEPLOYMENT</b>              |                                     |    |
| 27                                                  | Production Environment Reference    | 42 |
| 28                                                  | Production Deployment               | 44 |
| 29                                                  | Production Commands Reference       | 45 |
| 30                                                  | Commands Never to Run in Production | 46 |
| 31                                                  | Database Bootstrap                  | 47 |
| 32                                                  | Stripe Production Setup             | 48 |
| 33                                                  | Google Workspace SMTP Setup         | 49 |
| 34                                                  | Web Server Configuration            | 50 |
| 35                                                  | DNS & HTTPS                         | 51 |
| 36                                                  | Backups                             | 52 |
| <b>PART VII — OPERATIONS &amp; MAINTENANCE</b>      |                                     |    |
| 37                                                  | Logs & Troubleshooting              | 53 |
| 38                                                  | Maintenance                         | 56 |
| 39                                                  | Testing                             | 57 |
| 40                                                  | Local Development Environment       | 58 |
| <b>PART VIII — QUICK REFERENCE &amp; APPENDICES</b> |                                     |    |
| 41                                                  | Owner Quick Start                   | 59 |
| 42                                                  | Administrator Daily Workflow        | 60 |
| 43                                                  | Production Launch Checklist         | 61 |
| 44                                                  | Post-Launch Checklist               | 62 |
| 45                                                  | Recommended Follow-Up Improvements  | 63 |
| 46                                                  | Glossary                            | 64 |
| 47                                                  | Quick Route Reference               | 66 |
| 48                                                  | Owner Production Responsibilities   | 67 |
| 49                                                  | Final System Status                 | 68 |

# 1. Executive Overview

---

PLOCAL is a nationwide local business evaluation platform. It is built around one core model, repeated across every U.S. state, city, and category:



PLOCAL helps **customers** find local businesses they can trust, and helps **business owners** understand how professional, credible, visible, and customer-ready their business appears online. Unlike a simple listing directory, PLOCAL does not just confirm that a business exists — it evaluates it against a structured, transparent standard, then presents the outcome publicly.

## Who PLOCAL Serves

---

### Customers

People searching for a trustworthy local provider — contractors, salons, restaurants, and other local service businesses — across any U.S. state or city.

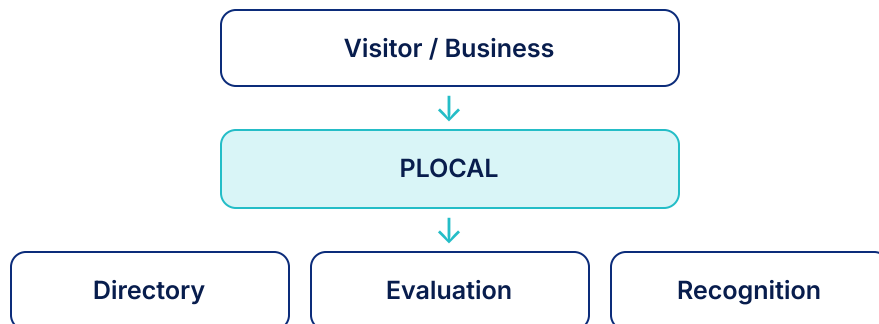
### Business Owners

Businesses that want a professional evaluation of their online presence, and the opportunity to be publicly recognized for strong performance.

## Main System Areas



## High-Level Flow

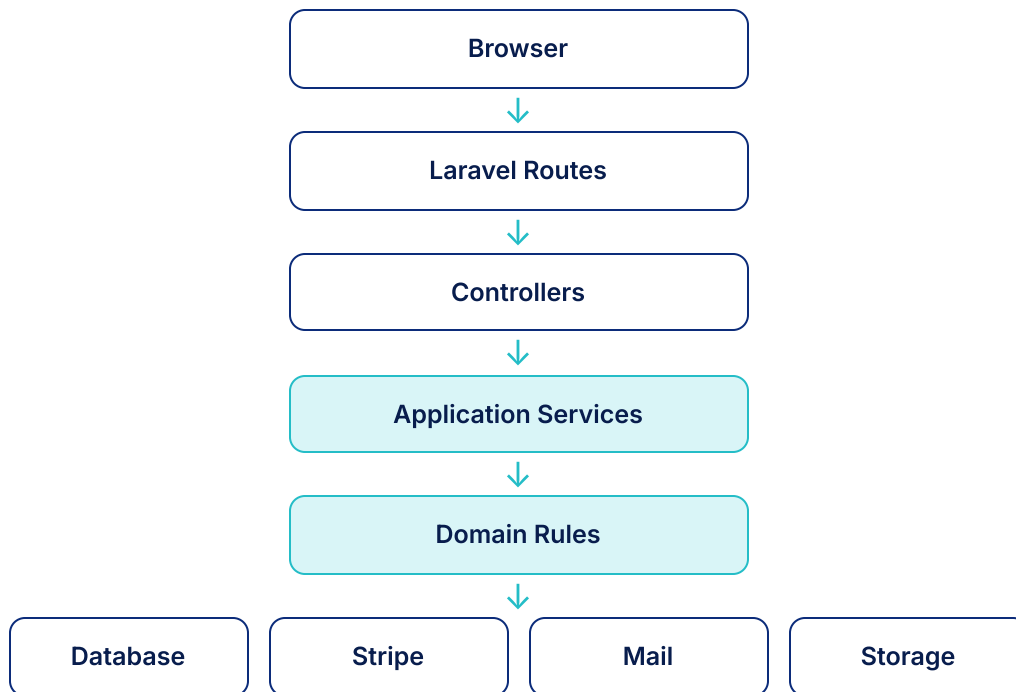


### CORE TRUST PRINCIPLE

A Business pays PLOCAL for the Evaluation service. Payment never purchases ranking, score, votes, a badge, an Award, a recommendation, or placement. This principle governs every commercial and editorial decision described in this guide.

## 2. System Architecture

PLOCAL is a single Laravel application. It is primarily server-rendered: pages are built on the server using Blade templates and returned as complete HTML, rather than relying on a separate single-page-application frontend framework. This keeps the system simple to operate, deploy, and reason about.



### Layer Responsibilities

| LAYER                       | RESPONSIBILITY                                                                                                                                                   |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Frontend</b>             | Server-rendered Blade templates, page-specific CSS/JS, and standard HTML forms. No separate frontend build is required to serve the site.                        |
| <b>Routes / Controllers</b> | Accept an HTTP request, validate input, and hand off to the correct Application Service. Controllers stay thin.                                                  |
| <b>Application Services</b> | Coordinate one specific use case end-to-end (for example, activating a Business after payment, or casting a public vote).                                        |
| <b>Domain Rules</b>         | The business rules that must always hold true regardless of caller — for example, how the PLOCAL Score is calculated, or what counts as a valid U.S. state code. |
| <b>Database</b>             | MySQL. The single source of truth for every Business, Submission, Payment, Evaluation, Vote, and Award.                                                          |

|                |                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------|
| <b>Stripe</b>  | Handles the Evaluation Checkout payment and is the sole authority on whether a payment succeeded. |
| <b>Mail</b>    | Sends invitation, notification, and transactional email through SMTP.                             |
| <b>Storage</b> | Holds private submission evidence and public content media (Blog images, advertiser logos).       |

### 3. Repository Structure

The Laravel application lives entirely inside the `backend/` directory of the repository. The directories below are the ones an owner or developer will interact with most.

| PATH                                      | PURPOSE                                                                                                                                                     |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>backend/app/Http/Controllers</code> | Route entry points, grouped by <code>Public</code> , <code>Business</code> , and <code>Internal</code> .                                                    |
| <code>backend/app/Application</code>      | One class per real use case (submitting a business, casting a vote, issuing an invitation).                                                                 |
| <code>backend/app/Domain</code>           | Pure business rules with no framework or database dependency (score calculation, state validation).                                                         |
| <code>backend/app/Infrastructure</code>   | Read-model implementations and integration-facing code.                                                                                                     |
| <code>backend/app/Console/Commands</code> | Artisan commands — production bootstrap, verification, and local developer tooling.                                                                         |
| <code>backend/resources/views</code>      | All Blade templates: <code>public/</code> , <code>business/</code> , <code>internal/</code> , and shared <code>layouts/</code> and <code>partials/</code> . |
| <code>backend/routes/web.php</code>       | Every public, Business, and Internal route.                                                                                                                 |
| <code>backend/database/migrations</code>  | The complete database schema history.                                                                                                                       |
| <code>backend/database/seeder</code>      | Reference-data seeders, including the one production-safe bootstrap seeder.                                                                                 |
| <code>backend/public</code>               | The web server document root — the only directory that should be exposed directly to the internet.                                                          |
| <code>backend/storage</code>              | Private submission evidence, logs, and framework cache — never web-exposed directly.                                                                        |
| <code>backend/tests</code>                | The automated test suite covering the majority of product behavior.                                                                                         |
| <code>assets/</code> (repository root)    | Shared CSS, JavaScript, images, and icon assets used by the public site.                                                                                    |

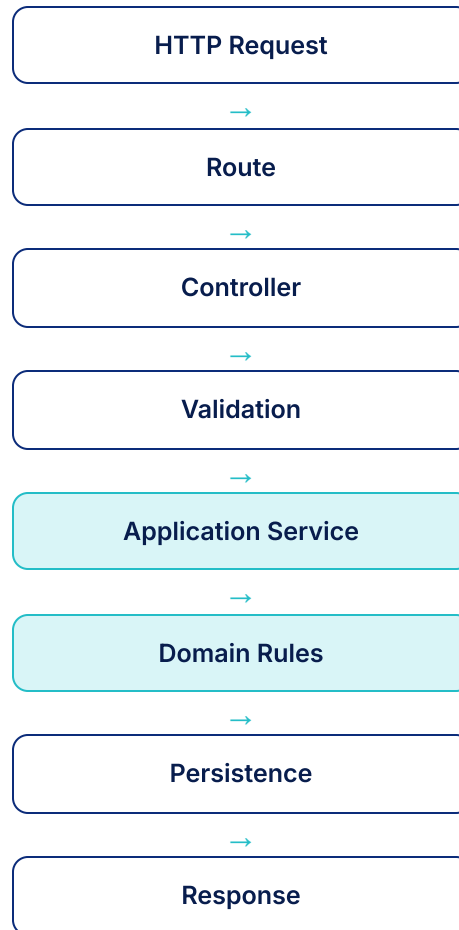
#### NOTE

This guide intentionally does not list every file — only the directories relevant to understanding, operating, or maintaining PLOCAL.

## 4. Backend Architecture

---

Every request follows the same predictable path through the backend:



### Routing

---

Routes are grouped into three clearly separated areas:

- **Public routes** — no authentication, open to any visitor.
- **Business routes** — prefixed `/business`, require a Business Portal login.
- **Internal routes** — prefixed `/internal`, require staff login and role permission.

### Validation & Authorization

---

Every state-changing request is validated before any Application Service runs. Internal routes are additionally protected by role-based permission checks (see Chapter 16 and Chapter 23), so a Reviewer account, for example, cannot perform an Administrator-only action even if it knows the correct URL.

## Background Jobs

---

Notification delivery (invitation emails, activation confirmations, support-case acknowledgements) is queued rather than sent inline, so a slow mail provider never blocks a visitor-facing request. See Chapter 26 for the current queue configuration.

## External Integrations

---

| INTEGRATION             | PURPOSE                                                          |
|-------------------------|------------------------------------------------------------------|
| Stripe                  | Evaluation Checkout payment and signed webhook confirmation.     |
| SMTP (Google Workspace) | Transactional email delivery from <code>info@plocal.com</code> . |
| MySQL                   | Primary data store for every domain object in the system.        |

## 5. Frontend Architecture

PLOCAL's frontend is server-rendered. Laravel's Blade templating engine builds each page on the server and returns finished HTML — there is no separate JavaScript framework driving page navigation, and no client-side routing to keep in sync with the backend.

### Template Organization

| AREA                       | LOCATION                                                                            |
|----------------------------|-------------------------------------------------------------------------------------|
| Shared layout & navigation | <code>resources/views/layouts</code> , <code>resources/views/public/partials</code> |
| Public pages               | <code>resources/views/public</code>                                                 |
| Business Portal pages      | <code>resources/views/business</code>                                               |
| Internal Portal pages      | <code>resources/views/internal</code>                                               |

The public site's header and footer are each a single shared partial included on every public page, so a navigation change made once is reflected everywhere automatically.

### Assets

CSS and JavaScript are organized by page and by shared concern under the repository's `assets/` directory (bootstrap and vendor libraries, a shared custom stylesheet, and page-specific stylesheets/scripts). Pages are responsive down to common mobile widths.

### Forms & Server Communication

Public forms (Submit Your Business, Contact, Support) post directly to a Laravel route. Where a form submits without a full page reload, it uses a plain JavaScript `fetch` request against the same route and displays the JSON result inline — there is no separate API layer or token-based authentication for these public forms.

## 6. Public Website Navigation

The table below covers the primary pages a visitor encounters. Internal action-only endpoints are intentionally omitted — see Chapter 47 for a short reference of the most useful routes.

| ROUTE                                                   | PURPOSE                                                                                      |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------|
| /                                                       | Home — Business of the Day, featured Recognition, and entry points into the Directory.       |
| /about.html                                             | About PLOCAL — what the platform is and why it exists.                                       |
| /contact.html                                           | General contact form.                                                                        |
| /faq.html                                               | Frequently asked questions.                                                                  |
| /support.html                                           | Support request form (account, Profile, Evaluation, billing, accessibility, Trust & Safety). |
| /trust-and-safety.html                                  | Trust & Safety overview.                                                                     |
| /protect-yourself-from-fraud.html                       | Fraud-awareness guide for customers.                                                         |
| /terms-and-conditions.html                              | Terms & Conditions.                                                                          |
| /accessibility-statement.html                           | Accessibility statement.                                                                     |
| /how-plocal-evaluation-works.html                       | Public explanation of the Evaluation model.                                                  |
| /submit-your-business                                   | Business Submission form (see Chapter 9).                                                    |
| /find-businesses                                        | Nationwide directory search (see Chapter 7).                                                 |
| /categories.html                                        | Curated category landing page.                                                               |
| /usa                                                    | National directory hub, branching into state and city hubs.                                  |
| {/state}, {/state}/{city},<br>{/state}/{city}/{service} | Geography-based discovery hierarchy.                                                         |
| /recognition                                            | Business of the Day, Month, and Year (see Chapter 14).                                       |
| /blog, /blog/{slug}                                     | Published editorial articles.                                                                |
| /pro/{businessSlug}                                     | An individual Business Profile (see Chapter 8).                                              |

## TWO DIRECTORY LAYERS

`/find-businesses` is the current, primary nationwide search surface. The geography hierarchy under `/usa` is a complementary, SEO-oriented browsing path into the same underlying evaluated Businesses — both ultimately lead to the same Business Profile pages.

## 7. Nationwide Directory & Find Businesses

PLOCAL supports all 50 U.S. states using canonical two-letter USPS state codes (for example, `CA`, `TX`, `NY`). A Submission is only accepted with a valid code from this canonical list — full state names or invalid codes are rejected at submission time. Cities are created automatically the first time a legitimately paid, activated Business reports that city — there is no need to pre-populate every city in the country in advance.

PLOCAL / FIND BUSINESSES

# Find Evaluated Local Businesses

Search PLOCAL-evaluated businesses by category, state, city, and ZIP code. Results are shown in a neutral order — never by score, votes, or awards.

Home / Find Businesses

**CATEGORY**  
All categories

**STATE**  
All states

**CITY**  
City

**ZIP CODE**  
12345

**SEARCH** [Clear filters](#)

RESULTS

*Find Businesses — nationwide search by category, state, city, and ZIP. Captured from the local development environment.*

### Search Filters

- **Category** — Construction & Home Services, Beauty & Personal Care, Restaurants & Food Businesses.
- **State** — any of the 50 canonical U.S. states.
- **City** — free-text, matched against provisioned locations.
- **ZIP** — five-digit ZIP code.

## Result Ordering

---

Results are shown in a neutral order. Placement is **not** determined by payment, votes, Awards, or Evaluation score — every eligible Business receives equal presentation weight in the result list.

## Visitor Journey

---

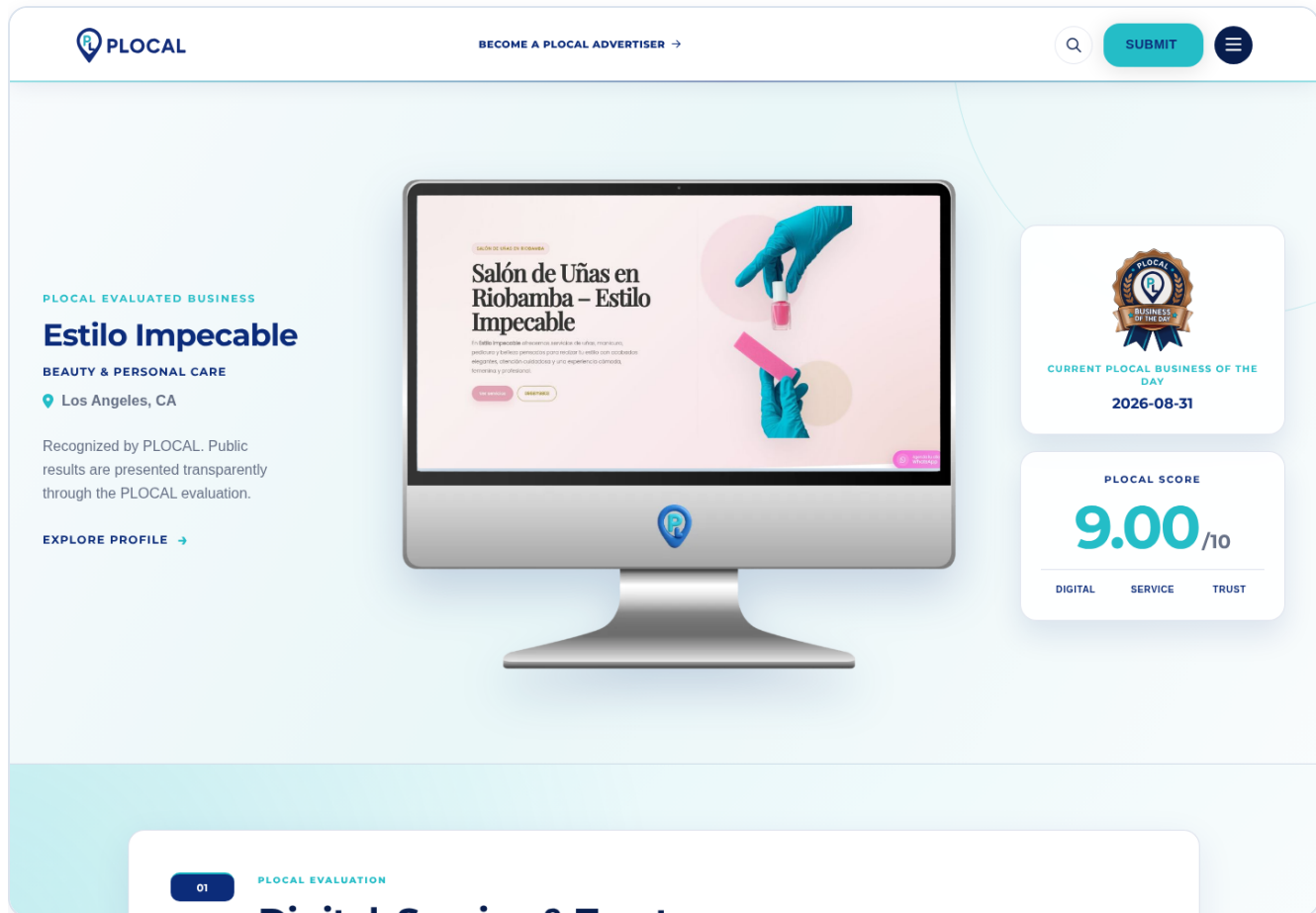
1. Open Find Businesses.
2. Select any combination of Category, State, City, or ZIP.
3. Review the neutral-order result list.
4. Open a Business Profile for full detail.
5. Return to the exact same search results using the page's Previous/Next or Back-to-Results controls (see Chapter 8).

### PUBLICATION SAFETY

Only a Business that is active, published, and discovery-eligible can appear in search results — a suspended, unpublished, or ineligible Business is never shown publicly, regardless of filters applied.

## 8. Business Profiles & Contextual Navigation

Every evaluated Business has a public profile at `/pro/{businessSlug}` .



A live Business Profile, showing category, location, website screenshot, and Digital / Service / Trust result. Captured from the local development environment.

### Profile Contents

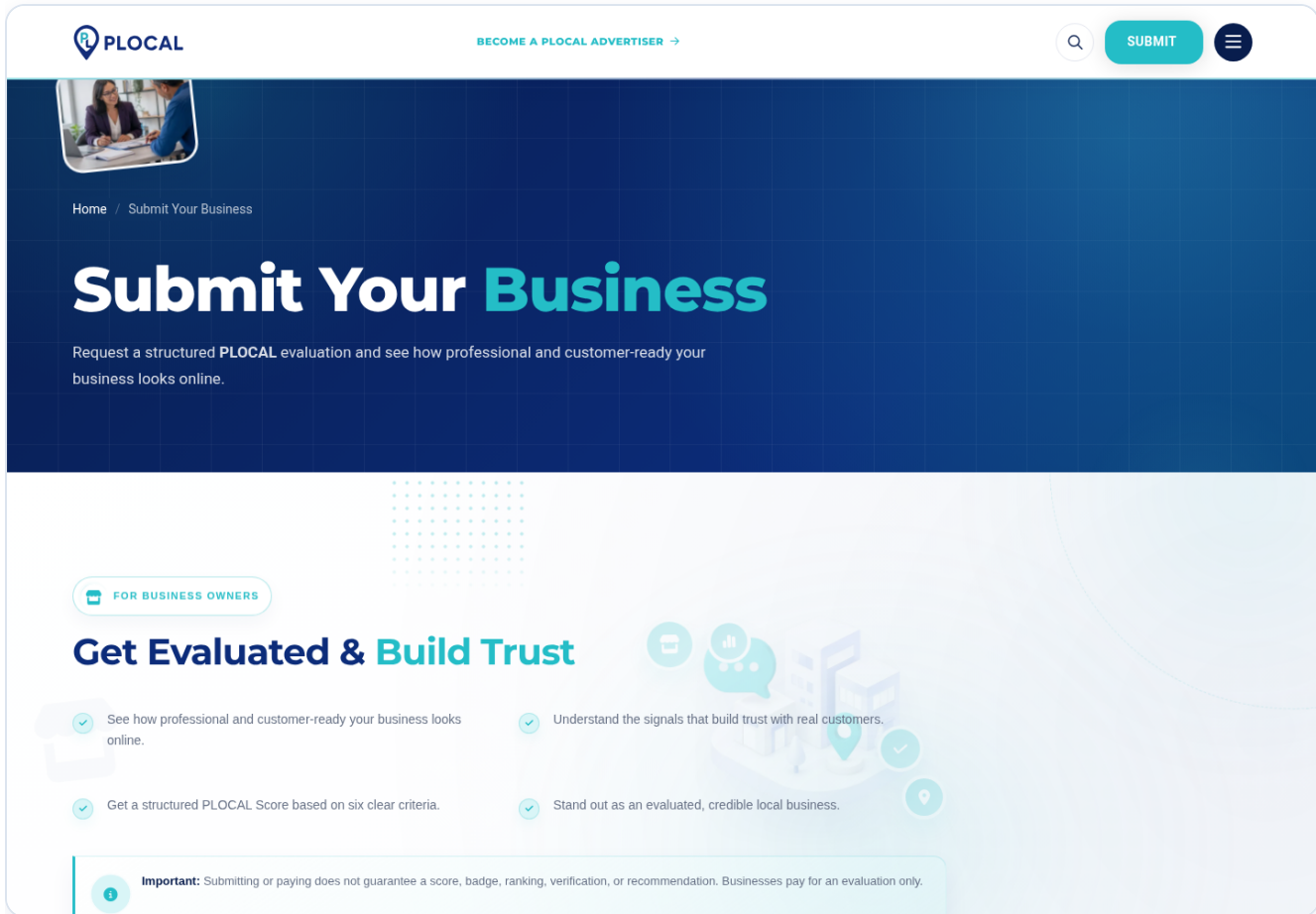
- Business identity — name, category, city and state.
- Website and submitted website screenshot.
- Evaluation result, shown in the current Digital / Service / Trust format once finalized.
- Current Recognition status, if the Business currently holds or has held an Award.
- Public Voting, where the Evaluation Window is still open for voting.
- Contextual Previous/Next navigation (below).

# Contextual Previous/Next Navigation

Profile pages remember how the visitor arrived, so browsing forward and backward stays inside the same context instead of jumping to an unrelated Business:

| ARRIVED FROM                      | PREVIOUS / NEXT BEHAVIOR                                                     |
|-----------------------------------|------------------------------------------------------------------------------|
| Newest Nominees                   | Chronological Nominee-to-Nominee navigation.                                 |
| Recognition                       | Chronological navigation within the same Day, Month, or Year level.          |
| Find Businesses                   | Navigation within the exact search-result context (same filters and page).   |
| Direct Profile visit (no context) | Falls back to Find Businesses rather than fabricating an arbitrary neighbor. |

## 9. Business Submission



**PLOCAL** [BECOME A PLOCAL ADVERTISER →](#)

Home / Submit Your Business

# Submit Your Business

Request a structured PLOCAL evaluation and see how professional and customer-ready your business looks online.

**FOR BUSINESS OWNERS**

## Get Evaluated & Build Trust

- ✓ See how professional and customer-ready your business looks online.
- ✓ Understand the signals that build trust with real customers.
- ✓ Get a structured PLOCAL Score based on six clear criteria.
- ✓ Stand out as an evaluated, credible local business.

**Important:** Submitting or paying does not guarantee a score, badge, ranking, verification, or recommendation. Businesses pay for an evaluation only.

*Submit Your Business — the public Submission form. Captured from the local development environment.*

### Information Collected

- Business name, category, city, state, and ZIP.
- Website URL, main services, and a business description.
- Contact details for the submitter.
- A website screenshot, used as evaluation evidence.

### Website Screenshot Requirement

| REQUIREMENT      | VALUE             |
|------------------|-------------------|
| Format           | JPEG              |
| Exact dimensions | 1068 × 646 pixels |

|                   |        |
|-------------------|--------|
| Maximum file size | 150 KB |
|-------------------|--------|

Verified directly against the current Submission validation rules.

## What Happens After Submission

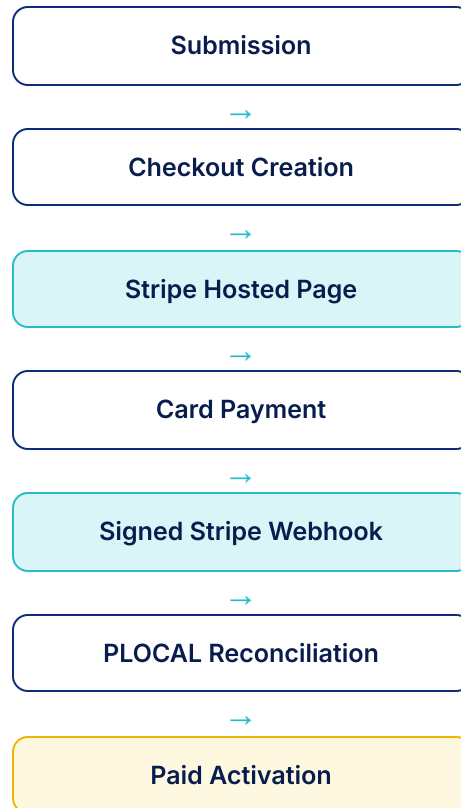
A successful Submission creates a durable record and immediately proceeds to Evaluation Checkout (Chapter 10). The Submission itself does not create a public Business Profile — publication only happens after payment is confirmed and the Business is activated (Chapter 11).

### NO PRE-PAYMENT APPROVAL STEP

The current flow does not include a "wait for approval before payment" stage, and there is no legacy Claim/Accept step exposed on the public Submission flow. Checkout follows immediately after a valid Submission.

## 10. Payments & Stripe

PLOCAL charges a fixed Evaluation service fee of **\$250 USD plus applicable taxes**, collected through Stripe's hosted Checkout page.



### PAYMENT AUTHORITY

The browser's post-checkout success page is **never** treated as proof of payment. Only Stripe's **signed server-to-server webhook event** confirms that a payment succeeded and authorizes activation. This prevents a visitor from spoofing a successful payment by manipulating the browser.

## What Payment Does and Does Not Buy

---

### Payment purchases

- The Evaluation service itself.

### Payment never purchases

- Ranking or placement
- Evaluation score
- Votes
- A badge
- An Award or Recognition
- A recommendation

## Other Payment Behavior

---

- **Automatic tax** — applicable tax is calculated and added on top of the base fee where configured.
- **Duplicate charges / refunds** — handled through Internal Payment Operations (Chapter 18), never through direct database edits.
- **Activation retry** — if activation did not complete automatically after a confirmed payment, an Administrator can retry it from Internal Payment Operations.

## Stripe: Test vs. Production

| ENVIRONMENT           | DESCRIPTION                                                    |
|-----------------------|----------------------------------------------------------------|
| Development / Sandbox | Uses Stripe test-mode keys. No real card is charged.           |
| Production / LIVE     | Uses Stripe live-mode keys. Real cards are charged real money. |

```
# .env – never commit real values
STRIPE_SECRET_KEY=
STRIPE_WEBHOOK_SECRET=
```

### Local Development

```
# Terminal 1
php artisan serve

# Terminal 2 – forwards Stripe test events to your local app
stripe listen --forward-to http://localhost:8000/webhooks/stripe
```

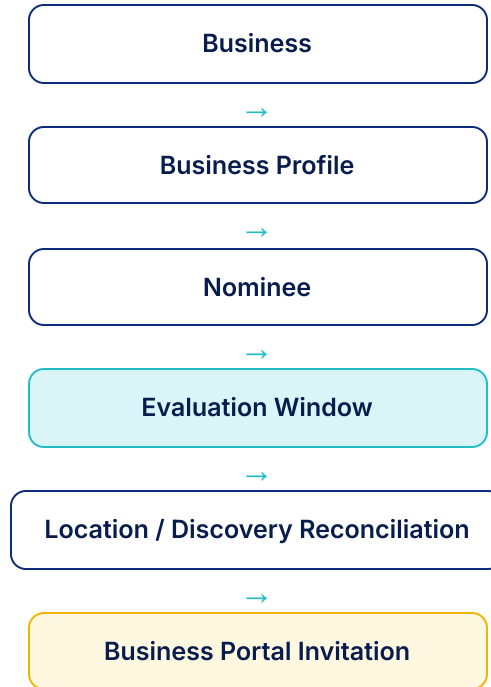
#### STRIPE CLI IS LOCAL-ONLY

The `stripe listen` command is a local development convenience. Production never uses the Stripe CLI listener — it receives webhook events directly from Stripe over HTTPS at the production domain (see Chapter 32).

## 11. Paid Activation

---

Once Stripe's signed webhook confirms payment, PLOCAL provisions everything the Business needs in a single, idempotent sequence:



**Idempotency:** replaying the same confirmed payment event does not create a duplicate Business or a duplicate Evaluation Window — activation is safe to retry.

If activation did not fully complete (for example, a transient failure between steps), an Administrator can trigger a safe retry from Internal Payment Operations rather than needing any direct database intervention (Chapter 18).

## 12. Evaluation — Digital / Service / Trust

Every activated Business is assessed across three dimensions, each scored from **0 to 10**:

Digital

Service

Trust

### PLOCAL Score

**FORMULA**

PLOCAL Score = (Digital + Service + Trust) ÷ 3, rounded exactly once to two decimal places.

### Evaluation Window

Each Evaluation runs on a fixed window of **96 elapsed hours** from the moment the Business is activated. This is wall-clock elapsed time, not 96 business hours.

### Roles Involved

| ROLE                 | RESPONSIBILITY                                                                        |
|----------------------|---------------------------------------------------------------------------------------|
| <b>Evaluator</b>     | Performs the initial Digital / Service / Trust assessment for an assigned Nominee.    |
| <b>Reviewer</b>      | Reviews and can return an Evaluator's work for changes, or complete it.               |
| <b>Administrator</b> | Holds full access, including the authority to make the Final Decision on Recognition. |

**LEGACY NOTE**

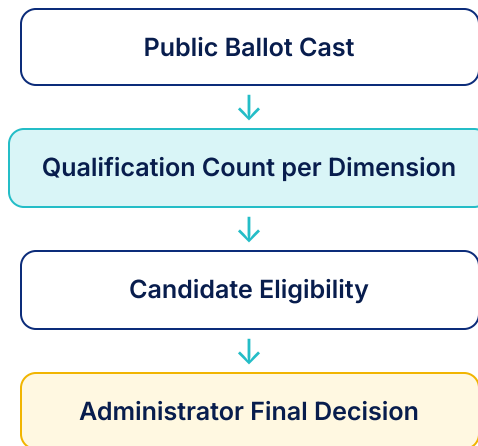
An earlier, retired 100-point "ProScore" evaluation format exists only in historical/demo data and is not the current Evaluation model. The current, active model is Digital / Service / Trust as described above.

## 13. Public Voting

While a Nominee's Evaluation Window is open, the public can cast a Vote across the same three dimensions: Digital, Service, and Trust.

### Voting Rules

- One pseudonymous device identity may cast one Ballot per Nominee, per Evaluation Window.
- Each dimension requires a target of **5 valid, distinct-device votes** to reach qualification.
- A staff Administrator can invalidate a Ballot found to be invalid, which recomputes qualification.



#### VOTES INFORM — THEY DO NOT DECIDE

Votes help establish qualification and public engagement signal. They do not directly select a winner. PLOCAL, through the Administrator Final Decision, retains final Recognition authority.

## 14. Recognition

 [BECOME A PLOCAL ADVERTISER →](#)  [SUBMIT](#) 

PLOCAL / RECOGNITION

# Official PLOCAL Award Winners

Business of the Day, Business of the Month, and Business of the Year are official PLOCAL recognition — never a score or vote leaderboard.

[Home](#) / [Recognition](#)

LATEST WINNERS

### Currently Recognized



1920x1280

BUSINESS OF THE DAY

PLOCAL NAV DEMO — Atlas



1920x1280

BUSINESS OF THE MONTH

PLOCAL NAV DEMO — Atlas



1920x1280

BUSINESS OF THE YEAR

PLOCAL NAV DEMO — Atlas

*Recognition — Business of the Day, Month, and Year. Captured from the local development environment.*

### Recognition Hierarchy



## Lifecycle

---

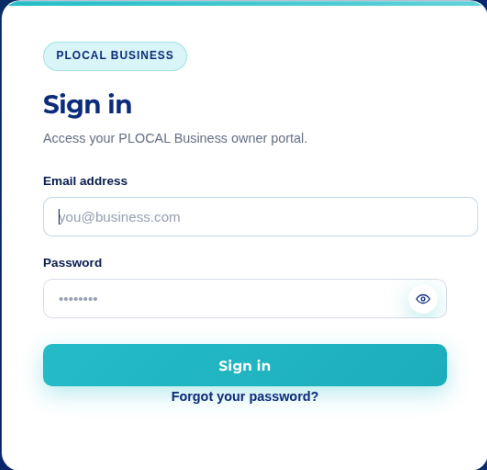
A qualifying Nominee that reaches the Evaluation and voting thresholds becomes a **Candidate**. An Administrator's **Final Decision** determines the outcome — confirmed as a Nominee, granted Honors, awarded Business of the Day, or declined. Honors and Day/Month/Year Awards are drawn from qualifying Candidates on a chronological basis.

## Historical Record

---

Past Recognition remains publicly viewable. If an Award is later revoked or replaced, that history is preserved for transparency rather than silently deleted — a revoked or inactive Award is not shown as currently valid, but the record of it having occurred is retained internally.

## 15. Business Portal



PLOCAL BUSINESS

### Sign in

Access your PLOCAL Business owner portal.

Email address

Password

Sign in

[Forgot your password?](#)

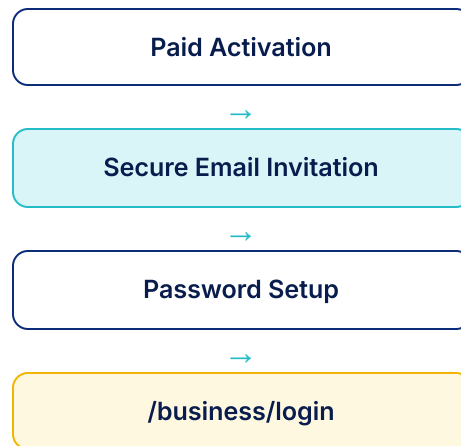
Business Portal sign-in at `/business/login` . Captured from the local development environment.

### NO PUBLIC SIGN-UP

There is no public Business registration page. A Business account only comes into existence through the invitation flow below.

## Onboarding Flow

---



## What a Business Account Can See

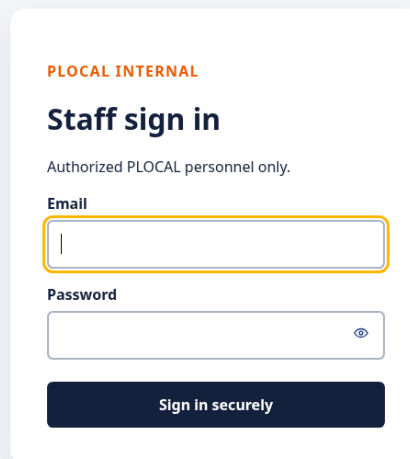
---

The dashboard is a read-only status view scoped strictly to the account's own Business:

- Business identity, category, city/state, and public Profile status and link.
- Evaluation status, and once finalized, its Digital / Service / Trust and PLOCAL Score result.
- Payment history, payment status, and refund status.
- Award and Recognition history for its own Business.

A Business account cannot view or influence another Business's data, and cannot access the Internal Portal under any circumstance — the two systems use entirely separate authentication guards.

## 16. Internal Portal & RBAC



PLOCAL INTERNAL

### Staff sign in

Authorized PLOCAL personnel only.

Email

Password

Sign in securely

Internal Portal staff sign-in at `/internal/login` . Captured from the local development environment.

### Roles

| ROLE                 | ACCESS LEVEL                                                                                    |
|----------------------|-------------------------------------------------------------------------------------------------|
| <b>Administrator</b> | Full access, including Final Decisions, Payment Operations, Users, and content management.      |
| <b>Reviewer</b>      | Reviews assigned Evaluations, can return work for changes, confirms license/verification items. |
| <b>Evaluator</b>     | Performs and submits assigned Evaluation scoring.                                               |

### Dashboard Areas

- Overview / Nominees & Evaluations

- Business Submissions
- Payment Operations
- Support Cases
- Business Duplicates
- Business of the Month / Business of the Year
- Award Corrections
- Blog
- Advertisers
- Internal Users **ADMINISTRATOR-ONLY**

Every area is gated by role permission — a Reviewer or Evaluator account attempting an Administrator-only action (such as managing Internal Users) is denied regardless of the URL used.

## 17. Internal Evaluation Workflow

---

1. Staff logs in to the Internal Portal.
2. Opens Nominees / Evaluations.
3. Opens the assigned, current Nominee.
4. Reviews the Business and submitted evidence (website, screenshot).
5. Scores the Digital dimension.
6. Scores the Service dimension.
7. Scores the Trust dimension.
8. Saves and submits the Evaluation for review.
9. Candidate eligibility is evaluated once qualification thresholds are met.
10. An Administrator makes the Final Decision.
11. The Recognition outcome, if any, appears publicly on the Business Profile and Recognition page.

### VERIFY BEFORE DOCUMENTING UI LABELS

Exact on-screen button labels may be refined over time. This workflow describes the current functional sequence rather than pixel-exact wording.

## 18. Payment Operations

---

Internal Payment Operations gives Administrators visibility and safe recovery tools over every payment:

- Authoritative payment status and history per Business.
- Refund status, including duplicate-charge handling.
- Activation recovery/retry for a payment that was confirmed but did not fully activate.
- Relevant Stripe identifiers, where displayed for reference.

### **NO MANUAL OVERRIDE AUTHORITY**

There is no "mark as paid" control that bypasses Stripe. Payment status must always remain tied to Stripe's verified, signed state — Internal Payment Operations can retry activation for an already-confirmed payment, but it cannot fabricate a payment.

## 19. Support Cases

---

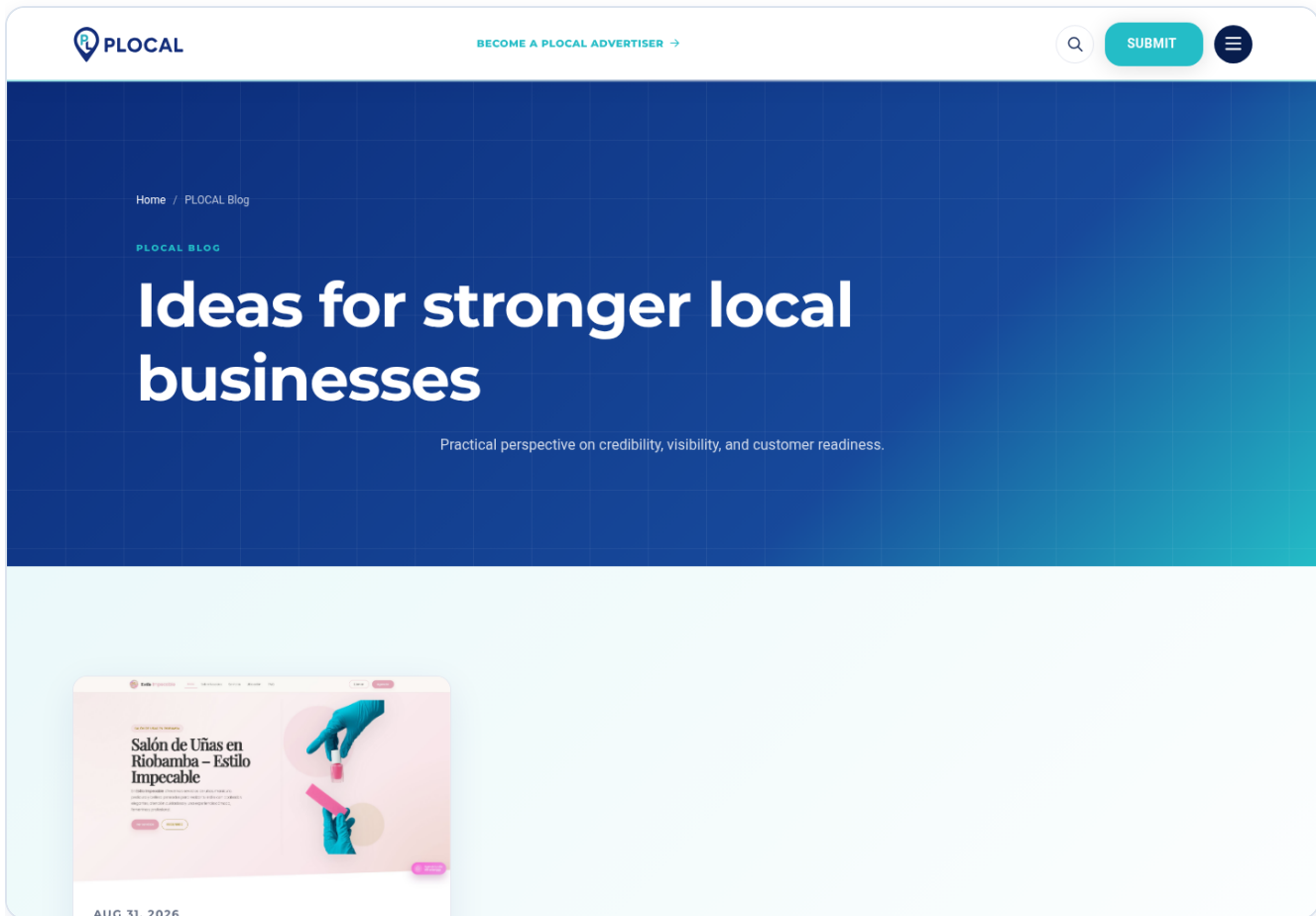
Current official contact channels:

| CHANNEL | VALUE           |
|---------|-----------------|
| Email   | info@plocal.com |
| Phone   | 888-988-3213    |

Public Contact and Support forms submit into the same internal Support Case system. Every submission is durably recorded and reviewable by Administrators from Internal Support Cases, with a status workflow (for example, received → in review → resolved).

The Business Profile's **"Report a concern"** link routes a visitor to the official Contact page, which submits into this same Support Case system — there is a single, consistent path for any public concern or fraud report, rather than a separate parallel system.

## 20. Blog CMS



Public Blog index at `/blog` . Captured from the local development environment.

### Internal Management Capabilities

- Create and edit a draft article, including featured image and alt text.
- Publish and unpublish an article.
- Feature an article for prominent homepage/blog placement.
- Delete (archive) an article.

### Public Visibility

Only published articles are ever publicly reachable at `/blog` or `/blog/{slug}` . A draft or unpublished article is never exposed on the public site.

## 21. Advertiser Management

---

### COMMERCIAL MODEL

Advertisers are recruited and contacted **outside** PLOCAL. There is currently no online advertiser self-service or payment system inside the platform — an Administrator manually adds each approved advertiser.

### What an Administrator Configures

---

- Company name and logo
- Website URL
- Active / inactive status
- Display order
- Optional visibility start/end dates

Active advertisers appear in the homepage sponsor carousel, linking only to their own `website_url`.

### STRICT SEPARATION

Advertising never affects ranking, Evaluation, Voting, or Recognition. An advertiser's logo on the homepage carries no evaluative meaning whatsoever.

## 22. Database Architecture

PLOCAL uses MySQL as its single source of truth. The conceptual relationships below matter more for day-to-day operation than the full column-level schema.



### Other Core Tables

| AREA               | HOLDS                                                          |
|--------------------|----------------------------------------------------------------|
| Internal Users     | Staff accounts and their assigned roles/permissions.           |
| States / Locations | The 50-state reference set and dynamically provisioned cities. |
| Categories         | The current product category taxonomy.                         |
| Blog Articles      | Editorial content and publication status.                      |
| Advertisers        | The homepage sponsor carousel entries.                         |

Migrations under `backend/database/migrations` are the authoritative schema history; they should be applied in order and never edited after being run against a live database.

## 23. Authentication & Security Model

### Three Separate Authentication Systems

| SYSTEM   | WHO                                     | NOTES                                                            |
|----------|-----------------------------------------|------------------------------------------------------------------|
| Public   | Any visitor                             | No authentication required for browsing pages.                   |
| Business | An activated Business's authorized user | <code>/business/login</code> , invitation-based onboarding only. |
| Internal | PLOCAL staff                            | <code>/internal/login</code> , role-based access control.        |

Keeping Business and Internal authentication completely separate ensures a compromised or misused Business account can never reach staff-only tooling, and vice versa.

### Current Security Controls

- CSRF protection on session-based forms, with a documented, narrow exception for unauthenticated public endpoints (Submission, Support, payment webhooks) that are instead protected by same-origin checks and per-route rate limiting.
- Signed URLs for the payment checkout flow.
- Stripe webhook signature verification as the sole payment authority.
- Role-based authorization (RBAC) throughout the Internal Portal.
- Rate limiting on Submission, Voting, Support Case, and payment-checkout endpoints.
- Private evidence storage, never served directly from the web root.
- Standard password hashing for Business and Internal accounts.
- Business accounts are strictly scoped to their own Business's data.

#### PRODUCTION EXPECTATIONS

Production must run with `APP_DEBUG=false` , HTTPS enabled, and secrets supplied only through environment configuration — never committed to the repository. PLOCAL does not claim a formal security certification and is not represented as immune to attack; the controls above reflect current, verified application behavior.

## 24. File Storage

| CONTENT                                      | VISIBILITY     | NOTES                                                                                            |
|----------------------------------------------|----------------|--------------------------------------------------------------------------------------------------|
| Submission evidence<br>(website screenshots) | <b>Private</b> | Served only through an authenticated/controlled Laravel route — never a direct public file path. |
| Blog cover images                            | Public         | Served through a dedicated media route.                                                          |
| Advertiser logos                             | Public         | Served through a dedicated media route.                                                          |

### PERSISTENCE ON A VPS

Uploaded files are stored on local disk under the application's storage directory. On a VPS, this directory must live on persistent storage and be included in the backup plan (Chapter 36) — a stateless or ephemeral deployment target will lose uploaded evidence and media on redeploy unless this is addressed.

## 25. Email

---

The current production design sends transactional email through **Google Workspace SMTP**, from **info@plocal.com**, using Laravel's standard Mailer.

```
# Production .env placeholders
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=info@plocal.com
MAIL_PASSWORD=<GOOGLE_WORKSPACE_SMTP_CREDENTIAL>
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=info@plocal.com
MAIL_FROM_NAME=PLOCAL
```

Never place real credentials in source control. See Chapter 33 for the Google Workspace setup procedure.

## 26. Queues & Background Jobs

---

Local development may run with `QUEUE_CONNECTION=sync`, meaning queued work executes immediately, inline, with no separate worker process. This is convenient for development but has no retry/backoff behavior.

Production should use a persistent queue driver (the database driver is supported out of the box) together with a supervised worker process:

```
php artisan queue:work \  
  --queue=default \  
  --tries=3 \  
  --backoff=60,300 \  
  --max-time=3600
```

This worker command should run continuously under a process manager — `systemd`, Supervisor, or an equivalent tool already standard on the chosen server — so it is automatically restarted if it stops. The specific supervisor is an infrastructure choice for the owner/hosting administrator; this guide does not mandate one.

## 27. Production Environment Reference

The table below lists the environment variables PLOCAL reads, verified against the current configuration. No secret values are shown — use the placeholders when configuring production.

```
# Application
APP_NAME=PLOCAL
APP_ENV=production
APP_KEY=<GENERATED_BY_ARTISAN_KEY_GENERATE>
APP_DEBUG=false
APP_URL=https://www.plocal.com

# Logging
LOG_CHANNEL=stack
LOG_STACK=single
LOG_LEVEL=error

# Database
DB_CONNECTION=mysql
DB_HOST=<PRODUCTION_DB_HOST>
DB_PORT=3306
DB_DATABASE=<PRODUCTION_DB_NAME>
DB_USERNAME=<PRODUCTION_DB_USERNAME>
DB_PASSWORD=<PRODUCTION_DB_PASSWORD>

# Session & Cache
SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=true
CACHE_STORE=database

# Queue
QUEUE_CONNECTION=database

# Mail
MAIL_MAILER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=info@plocal.com
MAIL_PASSWORD=<GOOGLE_WORKSPACE_SMTP_CREDENTIAL>
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=info@plocal.com
MAIL_FROM_NAME=PLOCAL
```

```
# Stripe
STRIPE_SECRET_KEY=<STRIPE_LIVE_SECRET_KEY>
STRIPE_WEBHOOK_SECRET=<STRIPE_LIVE_WEBHOOK_SECRET>
```

#### **ONLY WHAT PLOCAL ACTUALLY USES**

File storage defaults to the local disk driver and the queue defaults to the database driver — both already provisioned by the standard migrations. S3-compatible object storage or Redis are not required unless the owner deliberately chooses to introduce them later.

## 28. Production Deployment

---

The steps below describe a conceptual deployment to a Linux VPS. No specific hosting provider is assumed.

1. Provision a Linux server.
2. Install a compatible PHP version (PHP 8.3 or newer, per `composer.json`).
3. Install the required PHP extensions (standard Laravel/MySQL extension set).
4. Install Composer.
5. Install and configure MySQL.
6. Upload or clone the repository onto the server.
7. Configure the web server to point at `backend/public` as the document root.
8. Create and populate the production `.env` file (Chapter 27).
9. Generate the application key safely: `php artisan key:generate`.
10. Install production dependencies: `composer install --no-dev --optimize-autoloader`.
11. Set correct filesystem permissions on `storage` and `bootstrap/cache`.
12. Run migrations: `php artisan migrate --force`.
13. Run the PLOCAL production bootstrap: `php artisan plocal:bootstrap-production`.
14. Configure and start the queue worker (Chapter 26).
15. Configure SMTP (Chapter 33).
16. Configure Stripe LIVE keys and webhook (Chapter 32).
17. Configure HTTPS (Chapter 35).
18. Configure DNS (Chapter 35).
19. Confirm persistent storage for uploads (Chapter 24).
20. Confirm a backup plan is active (Chapter 36).
21. Run the production readiness check (Chapter 29).
22. Smoke-test the public flows described in Chapter 43.

## 29. Production Commands Reference

```
# Install production dependencies
composer install --no-dev --optimize-autoloader

# Apply the database schema
php artisan migrate --force

# Seed the production-safe reference data (RBAC, Evaluation policy, categories, 50
states)
php artisan plocal:bootstrap-production

# Cache configuration, routes, and views for performance
php artisan config:cache
php artisan route:cache
php artisan view:cache

# Read-only check of production configuration health
php artisan plocal:verify-production-readiness

# Start the queue worker (run under a process supervisor)
php artisan queue:work --queue=default --tries=3 --backoff=60,300 --max-time=3600
```

### RE-CACHING AFTER CONFIG CHANGES

If `.env` or route/view files change after the first deployment, re-run the `config:cache`, `route:cache`, and `view:cache` commands so the change actually takes effect.

## 30. Commands Never to Run in Production

### DESTRUCTIVE OR DEVELOPMENT-ONLY — DO NOT RUN AGAINST PRODUCTION

| COMMAND                                                                                      | WHY NOT                                                                                                      |
|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <code>migrate:fresh</code>                                                                   | Drops every table and rebuilds the schema from nothing — destroys all production data.                       |
| <code>db:wipe</code>                                                                         | Drops every table, view, and type — irreversible data loss.                                                  |
| <code>plocal:seed-contextual-nav-demo</code>                                                 | Creates local-only visual demo fixtures, not real product data.                                              |
| <code>plocal:internal-demo-evaluation:create</code> / <code>:cleanup</code>                  | Creates/removes a private local workspace demo Evaluation.                                                   |
| <code>plocal:run-professional-rating-demo</code>                                             | A synthetic demo workflow intended only for the isolated test database.                                      |
| <code>plocal:verify-dallas-demo</code> ,<br><code>plocal:verify-multi-market-demo</code>     | Verify local-only demo fixtures; meaningless (or unsafe) against production data.                            |
| <code>plocal:backfill-legacy-market-membership</code> (without <code>--dry-run</code> first) | Writes new rows for existing Businesses — always review with <code>--dry-run</code> before running for real. |

This list reflects the current Artisan command registry at the time of writing. Command names should be re-verified against `php artisan list` before each deployment, since the registry can change over time.

## 31. Database Bootstrap

---

`php artisan plocal:bootstrap-production` is the single production-safe seeding entry point. It is idempotent — running it again on an already-initialized database changes nothing.

### What It Creates

---

- Roles and permissions (RBAC)
- The active Digital / Service / Trust Evaluation policy
- The current category taxonomy
- All 50 U.S. states, active and published

### What It Deliberately Never Creates

---

- Fake Businesses
- Fake Awards
- Fake payments
- Demo Internal Users
- Demo Evaluations

#### SAFE TO RUN EARLY AND OFTEN

This command is designed to be run immediately after every fresh migration, and safely re-run later if reference data needs to be refreshed — it never touches real Business, Award, payment, or account data.

## 32. Stripe Production Setup

---

1. Sign in to the Stripe account and switch to **LIVE mode**.
2. Retrieve the LIVE secret API key and store it only in the production `.env`.
3. Create a webhook endpoint pointed at the final production domain, for example `https://www.plocal.com/webhooks/stripe` (confirm the exact current route before finalizing).
4. Retrieve the webhook's signing secret and store it as `STRIPE_WEBHOOK_SECRET`.
5. If automatic tax is used, confirm the relevant tax registrations are configured in Stripe.
6. Send a real test transaction and confirm the webhook is received and activation completes.

### THE MOST COMMON PRODUCTION PAYMENT ISSUE

A payment can succeed at Stripe while PLOCAL remains stuck "pending" if the webhook cannot reach the server — for example, if the webhook endpoint is misconfigured, the server is behind a firewall blocking Stripe's IPs, or HTTPS is not correctly terminated. Always confirm webhook delivery status inside the Stripe dashboard when investigating a stuck payment.

The Stripe CLI listener ( `stripe listen` ) is never used in production — see Chapter 10.

## 33. Google Workspace SMTP Setup

---

1. Sign in to the Google Workspace account that owns `info@plocal.com`.
2. Generate an app-specific password (or configure Workspace SMTP relay/OAuth, depending on the account's current security policy).
3. Set the Laravel `MAIL_*` variables from Chapter 25 in the production `.env`.
4. Send a real test email from the application and confirm delivery.
5. At the DNS level, confirm SPF, DKIM, and DMARC records are configured for the sending domain — this materially improves deliverability and reduces the chance of mail landing in spam.

Exact Google Workspace account settings are specific to the owner's account and are not reproduced here.

## 34. Web Server Configuration

---

The web server's document root must point at `backend/public` — never at the repository root, and never directly at `backend/` itself. Only the contents of `public` should ever be reachable directly over the web.

### Conceptual Nginx Notes

- Root: `/path/to/project/backend/public`
- Pass PHP requests to PHP-FPM.
- Route all non-file requests through `index.php` (Laravel's standard front controller pattern).

#### NO UNVERIFIED CONFIG INCLUDED

A specific, copy-paste server configuration is intentionally not included here, since an unverified configuration can be unsafe. Use Laravel's official, current deployment documentation for the exact web server directives appropriate to the chosen server software and version.

## 35. DNS & HTTPS

---

- Point the production domain's DNS at the server.
- Decide the canonical form ( `www` or non- `www` ) and redirect the other consistently.
- Install a valid HTTPS certificate and force an HTTP → HTTPS redirect.
- Set `APP_URL` to the exact canonical HTTPS domain.
- Confirm secure, HTTPS-only cookies in production.
- Confirm the Stripe webhook endpoint uses this same HTTPS domain — Stripe requires HTTPS for production webhook delivery.

This guide uses `https://www.plocal.com` as the expected canonical domain; confirm and adjust if the final configured domain differs.

## 36. Backups

---

### OWNER / INFRASTRUCTURE RESPONSIBILITY

Backups are not yet an automated part of the application itself — they are an infrastructure responsibility the owner or hosting administrator must put in place.

### Recommended Practice

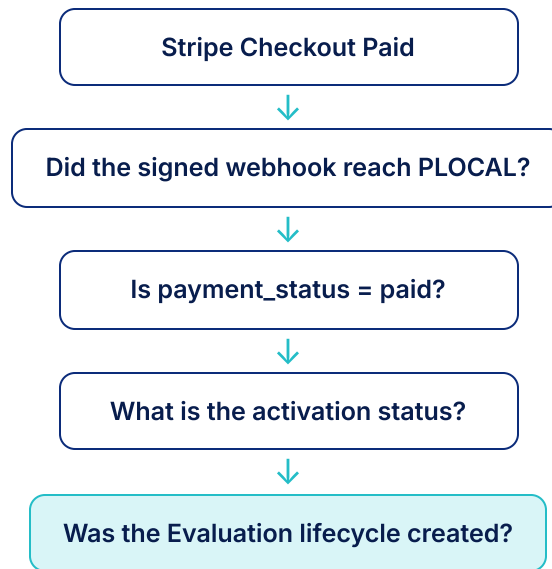
---

- Scheduled MySQL database backups, stored off-server.
- Backups of private Submission evidence storage.
- Backups of public Blog and Advertiser media.
- A defined retention period appropriate to the business's needs.
- Periodic restore testing — a backup that has never been restored is unverified.

## 37. Logs & Troubleshooting

| SYMPTOM                         | LIKELY CAUSE                             | WHERE TO CHECK                                                   | SAFE ACTION                                                                                |
|---------------------------------|------------------------------------------|------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 500 error                       | Uncaught exception                       | <code>storage/logs/laravel.log</code>                            | Read the stack trace; fix the underlying cause, never disable error visibility to hide it. |
| Database connection error       | Wrong credentials or DB unreachable      | <code>.env</code> <code>DB_*</code> values, MySQL service status | Verify credentials and that MySQL is running and reachable.                                |
| Payment stays pending           | Webhook not received                     | Stripe dashboard webhook delivery log                            | Confirm the endpoint is reachable over HTTPS; see Chapter 41.                              |
| Activation did not start        | Activation step failed after payment     | Internal Payment Operations                                      | Use Retry Activation; never edit payment rows directly.                                    |
| Email not sending               | SMTP misconfiguration                    | <code>.env</code> <code>MAIL_*</code> values, mail logs          | Send a test email; verify SMTP credentials.                                                |
| Queue not processing            | Worker not running                       | Process supervisor status, <code>failed_jobs</code> table        | Restart the worker; inspect failed jobs.                                                   |
| Image missing                   | Broken path or missing storage symlink   | Browser network tab, storage configuration                       | Confirm the file exists and the serving route is correct.                                  |
| Business not in Find Businesses | Not yet discovery-eligible               | Internal Business record status                                  | See Chapter 42 (Directory Troubleshooting).                                                |
| Permission / RBAC issue         | Account role lacks the needed permission | Internal Users role assignment                                   | Confirm the account has the correct role.                                                  |

## Payment Troubleshooting



Walk this chain top to bottom in Internal Payment Operations. If the webhook never arrived, fix delivery/HTTPS/firewall first — retrying activation cannot help until the webhook is actually received. If the webhook arrived and payment is confirmed but activation did not complete, use Retry Activation.

### NEVER EDIT PAYMENT STATUS DIRECTLY

Manually changing a payment's status in the database breaks the guarantee that payment state always reflects Stripe's verified truth. Always resolve payment issues through Internal Payment Operations.

## Directory Troubleshooting

**Symptom:** a Business was submitted and paid for, but does not appear in Find Businesses.

Check, in order:

1. Payment was actually confirmed.
2. Activation completed.
3. Business lifecycle and publication status are active/published.
4. The Business Profile itself is published.
5. The submitted state/city was correctly recognized.
6. Discovery eligibility was reached.

If a Business is stuck ineligible purely due to a geography reference-data gap, the safe reconciliation command is:

```
php artisan plocal:reconcile-discovery-eligibility --dry-run
```

Review the dry-run output first, then re-run without `--dry-run` only once the proposed change looks correct.

## 38. Maintenance

---

- Keep Laravel and PHP dependencies current, reviewing release notes before upgrading.
- Apply database migrations as they are introduced.
- Monitor the queue worker and the `failed_jobs` table.
- Monitor disk usage, particularly private evidence and media storage.
- Review logs periodically, not only when something visibly breaks.
- Periodically verify that backups completed successfully.
- Monitor Stripe webhook health from the Stripe dashboard.
- Monitor SMTP delivery for bounces or authentication failures.
- Apply security patches promptly.

### NO BLIND AUTOMATIC UPGRADES

Do not enable fully automatic, unattended dependency upgrades on a production system that handles real payments — review and test changes first.

## 39. Testing

---

PLOCAL includes an automated Laravel test suite covering the majority of product behavior, run against a fully isolated test database.

### CRITICAL SAFETY REQUIREMENT

Before any test run that writes data, positively verify which database is actually resolved — never trust a `--env=testing` flag alone to guarantee isolation. Tests must never be allowed to mutate the real development or production database.

```
# Run the full suite
php artisan test

# Run a focused subset
php artisan test --filter="SomeTestName"
```

## 40. Local Development Environment

---

```
cd backend
composer install
# configure .env for local development
# create the local database
php artisan migrate
php artisan plocal:bootstrap-production
php artisan serve
```

For local Stripe testing:

```
stripe listen --forward-to http://localhost:8000/webhooks/stripe
```

If queued work needs to be processed locally:

```
php artisan queue:work
```

Never place real production credentials in a local `.env` file.

## 41. Owner Quick Start

---

A short answer to: *"PLOCAL is already deployed — how do I actually use it?"*

### Public

- Browse PLOCAL at the home page.
- Use Find Businesses to search the directory.
- Check Recognition for current winners.
- Read the Blog for editorial content.

### Business

- Submit a business for Evaluation.
- Complete payment.
- Accept the emailed invitation.
- Log in to the Business Portal to check status.

### Administrator

- Log in to the Internal Portal.
- Review new Submissions and Payment Operations.
- Manage Evaluations and Recognition.
- Manage the Blog and Advertisers.
- Handle Support Cases and manage Internal Users.

## 42. Administrator Daily Workflow

---

### Morning

- ☐ Review new Support Cases.
- ☐ Review new Business Submissions.
- ☐ Check for payment or activation exceptions.
- ☐ Review active Evaluations in progress.
- ☐ Review current Recognition workload.

### Content

- ☐ Blog — draft, review, or publish articles.
- ☐ Advertisers — review active/inactive status and display order.

### Administration

- ☐ Manage Internal Users.
- ☐ Review Business Duplicates.
- ☐ Process any needed Award Corrections.

## 43. Production Launch Checklist

---

- ☐ VPS ready
- ☐ Production MySQL ready
- ☐ Production `.env` configured
- ☐ `APP_DEBUG=false`
- ☐ `APP_URL` correct and HTTPS
- ☐ HTTPS certificate valid
- ☐ DNS correctly pointed
- ☐ Migrations complete
- ☐ `plocal:bootstrap-production` complete
- ☐ 50 states verified
- ☐ Digital / Service / Trust policy verified active
- ☐ Stripe LIVE keys configured
- ☐ Stripe LIVE webhook configured and verified
- ☐ A real production payment tested safely
- ☐ Stripe Tax reviewed, if used
- ☐ SMTP verified with a real test email
- ☐ Queue worker running under a supervisor
- ☐ Persistent storage verified for uploads
- ☐ Backup system verified
- ☐ File permissions correct
- ☐ Home page works
- ☐ Find Businesses works
- ☐ A Business Profile works
- ☐ Submission works end-to-end
- ☐ Business login works
- ☐ Internal login works
- ☐ Blog works
- ☐ Advertisers display correctly
- ☐ robots/canonical metadata correct for production
- ☐ Logs reviewed after first deploy
- ☐ No demo/seed commands were run against production

## 44. Post-Launch Checklist

| WINDOW     | MONITOR                                                                               |
|------------|---------------------------------------------------------------------------------------|
| First hour | HTTP error rate, Stripe webhook delivery, first real payment/activation.              |
| First day  | Mail delivery, queue job processing, disk usage, database health, support inbox.      |
| First week | Search-engine indexing, backup completion, ongoing support volume, overall stability. |

Items above are operational recommendations, not application behavior claims.

## 45. Recommended Follow-Up Improvements

---

PLOCAL is prepared for handoff. The items below are genuinely non-blocking follow-ups identified during the most recent technical review — none of them prevent a production launch.

- Add a sitemap for search-engine discovery of the full directory hierarchy.
- Minor responsive polish on one directory listing layout at narrow tablet widths.
- A small number of legacy static redirects still need cleanup.
- A product decision is still pending on two informational pages referenced from older content (a badge-verification lookup page and a standalone privacy-policy page).
- Establish version control (source history) for the repository going forward, if not already in place at handoff.

## 46. Glossary

---

### **Business Submission**

The initial form a business owner completes to request a PLOCAL Evaluation.

### **Business**

The canonical record for an activated business on PLOCAL.

### **Business Profile**

The public-facing page for a Business at `/pro/{slug}`.

### **Nominee**

A Business under active Evaluation and eligible for Recognition consideration.

### **Evaluation Window**

The fixed 96-elapsed-hour period during which a Nominee is assessed and voted on.

### **Digital / Service / Trust**

The three 0–10 dimensions that make up the current Evaluation model.

### **PLOCAL Score**

$(\text{Digital} + \text{Service} + \text{Trust}) \div 3$ , rounded once.

### **Voting Ballot**

One pseudonymous-device vote cast for a Nominee within its Evaluation Window.

### **Candidate**

A Nominee that has met the qualification thresholds required for Recognition consideration.

### **Honors**

A Recognition outcome granted by Administrator Final Decision, short of a Day/Month/Year Award.

### **Business of the Day / Month / Year**

The three tiers of official PLOCAL Recognition.

### **Paid Activation**

The provisioning sequence triggered once payment is confirmed.

### **Discovery Eligibility**

Whether a Business currently qualifies to appear in Find Businesses.

### **Support Case**

A durable record created from any public Contact, Support, or concern submission.

### **Internal User**

A PLOCAL staff account (Administrator, Reviewer, or Evaluator).

**Business User**

An invited account scoped to a single activated Business.

## 47. Quick Route Reference

| AREA     | ROUTE                 |
|----------|-----------------------|
| Public   | /                     |
| Public   | /find-businesses      |
| Public   | /submit-your-business |
| Public   | /recognition          |
| Public   | /blog                 |
| Business | /business/login       |
| Internal | /internal/login       |
| Internal | /internal             |

Action-only (POST) endpoints are intentionally omitted from this reference.

## 48. Owner Production Responsibilities

The following are infrastructure and configuration responsibilities for the owner or hosting administrator — they are not missing application features.

| RESPONSIBILITY                    | OWNER ACTION                                    |
|-----------------------------------|-------------------------------------------------|
| Hosting / VPS                     | Provision and maintain the server.              |
| Domain / DNS                      | Own and configure the production domain.        |
| HTTPS                             | Obtain and renew a valid certificate.           |
| Production database credentials   | Provision and secure production MySQL access.   |
| Stripe LIVE credentials           | Provide the live secret key and webhook secret. |
| Stripe Tax registrations          | Confirm applicable tax jurisdictions, if used.  |
| Google Workspace SMTP credentials | Provide the sending mailbox credential.         |
| Backup infrastructure             | Implement scheduled, tested backups.            |
| Queue supervisor                  | Keep the queue worker running continuously.     |
| Server monitoring                 | Monitor uptime, disk, and error rates.          |

## 49. Final System Status

---

This guide reflects the PLOCAL codebase as verified at the time of writing: routes, Artisan commands, environment variables, the Digital / Service / Trust Evaluation model, the 50-state directory architecture, the Stripe payment flow, the Business Portal, the Internal Portal, the Blog, Advertisers, and the production bootstrap command were each checked directly against the current source.

### SUMMARY

The PLOCAL codebase is prepared for owner handoff and production deployment, subject to completion of the production configuration and infrastructure checklist documented in this guide (Chapters 27–36 and 43). No functionality in this document is fabricated or overstated, and no legacy or development-only tooling is represented as current production product.