

APERTURE

Photo & Digital-Product Store

Sell your photography online — your server, your rules.

Complete User Manual

Installation • Administration • Payments • Troubleshooting

Version 1.1

Table of Contents

Table of Contents	2
1. Introduction	4
2. Features	4
Storefront.....	4
Selling & payments	4
Receipts & email	5
Administration	5
Private client galleries	5
3. System Requirements	5
4. Installation	6
4.1 Copy the files.....	6
4.2 Install PHP extensions (if needed).....	6
4.3 Create the database and user	6
4.4 Enter your database credentials	6
4.5 Set folder permissions.....	7
4.6 Run the installer	7
4.7 Recommended PHP settings for large photos	7
5. Connecting PayPal & Credit-Card Payments.....	7
5.1 What you need.....	8
5.2 Create your API credentials.....	8
5.3 Enter the keys in Aperture	8
5.4 How a payment flows.....	8
6. Enabling SSL (HTTPS)	8
6.1 Prerequisites	9
6.2 Install Certbot and issue a certificate.....	9
6.3 Automatic renewal.....	9
6.4 Force HTTPS and harden cookies	9
7. Using the Site as a Customer	9

7.1 Browsing galleries	9
7.2 Adding to the cart	10
7.3 Checking out.....	10
7.4 Downloading photos and the receipt	10
7.5 Returning to an order later	10
7.6 Clients with a private gallery.....	10
8. Using the Site as an Administrator.....	10
8.1 Dashboard	10
8.2 Creating galleries.....	10
8.3 Uploading photos.....	11
8.4 Pricing, titles and cover photos.....	11
8.5 Managing orders	11
8.6 Settings reference	11
9. Client Galleries & Private Access.....	12
9.1 How gallery visibility works.....	12
9.2 Create a client account	12
9.3 Assign galleries to a client	12
9.4 Share the sign-in details	13
9.5 Manage existing clients.....	13
10. Maintenance & Best Practices	13
10.1 Back up regularly	13
10.2 Security checklist.....	13
10.3 Email deliverability.....	14
11. Troubleshooting	14
Appendix A. File & Folder Map	15

1. Introduction

Aperture is a self-hosted online store for selling photographs and other digital products. It is designed to feel like a polished, gallery-first boutique — in the spirit of services such as Pixieset — while running entirely on your own Ubuntu LAMP server, with no monthly fees, no commissions to a third party, and your customers' data under your control.

Visitors browse your galleries, view large watermarked previews, add photos to a cart, and pay securely with PayPal or any major credit or debit card. The instant payment clears, they receive an emailed PDF receipt and can download the original, full-resolution files. You manage everything — galleries, pricing, uploads and orders — from a clean administrator panel in your browser.

This manual walks you through every step: installing the software, connecting it to PayPal, securing the site with SSL, and operating it day to day both as a customer and as the administrator. A troubleshooting section at the end addresses the issues you are most likely to encounter.

At a glance: Aperture is written in plain PHP and MySQL with no external libraries to install. If your server can run a normal PHP website, it can run Aperture.

2. Features

Storefront

- **Refined design.** Elegant, gallery-first design with a hero banner, gallery grid and a full-screen photo viewer.
- **Galleries.** Organize photos into galleries (collections / categories), each with its own cover, description and public or hidden visibility.
- **Watermarked previews.** Large preview images carry a tiled, semi-transparent watermark so your work cannot be lifted before purchase.
- **Mobile friendly.** Fully responsive — looks great on phones, tablets and desktops.

Selling & payments

- **Flexible pricing.** Set a price for every photo individually, apply a default price per gallery, or bulk-apply one price to an entire gallery in a single click.
- **PayPal & cards.** A single PayPal Checkout integration accepts both PayPal balances and Visa, Mastercard, American Express and debit cards — no second payment processor required.
- **Secure capture.** Payments are captured and verified on your server before an order is ever marked paid.
- **Instant delivery.** After payment, customers download the untouched, high-resolution original through a private, tokenized link.

Receipts & email

- **PDF receipts.** A polished, itemized PDF receipt is generated automatically for every order.
- **Email delivery.** The receipt is emailed to the customer along with a link back to their downloads, and can be re-downloaded at any time.

Administration

- **Dashboard.** Dashboard with revenue, order and catalog totals at a glance.
- **Bulk upload.** Drag-and-drop uploader that accepts many images at once.
- **Order management.** Order list and detail views, with status filters and one-click receipt access.
- **Self-service settings.** Settings page to control branding, currency, accent color, watermark, payment keys and email — no code editing required.

Private client galleries

- **Client accounts.** Create password-protected logins for your clients (for example, a wedding couple).
- **Private galleries.** Mark galleries Private and assign them to specific clients; they stay hidden from the public.
- **Personal view.** A signed-in client sees only the galleries you prepared for them — a clean, personal storefront.

3. System Requirements

Aperture runs on a standard “LAMP” stack. Most shared or virtual private Ubuntu servers already meet these requirements.

Component	Requirement
Operating system	Ubuntu 20.04 LTS or newer (any modern Linux works)
Web server	Apache 2.4+ (Nginx also works with equivalent configuration)
PHP	Version 7.4 or newer; PHP 8.1+ recommended
PHP extensions	pdo_mysql, gd, curl, mbstring (all standard)
Database	MySQL 5.7+ or MariaDB 10.3+
Mail	A local mail transfer agent (Postfix/sendmail) for receipt email — optional but recommended
TLS / SSL	A domain name and an SSL certificate (free via Let’s Encrypt) for secure checkout
PayPal	A PayPal Business account and REST API credentials

Why GD and cURL? GD generates the watermarked previews and thumbnails; cURL talks to PayPal's API. Both ship with virtually every PHP install — if either is missing, install the php-gd and php-curl packages and restart Apache.

4. Installation

The whole process takes about ten minutes. You will copy the files into place, create a database, set a few permissions, and run a one-time web installer.

4.1 Copy the files

Upload or copy the entire Aperture folder into your web root. For example, to serve it from `https://yourdomain.com/store`, place it at:

```
/var/www/html/store
```

4.2 Install PHP extensions (if needed)

On a fresh Ubuntu server you can ensure everything is present with:

```
sudo apt update
sudo apt install apache2 php php-mysql php-gd php-curl php-mbstring mysql-server
sudo systemctl restart apache2
```

4.3 Create the database and user

Open the MySQL shell with `sudo mysql` and run the following, choosing your own strong password:

```
CREATE DATABASE photostore CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'photostore'@'localhost' IDENTIFIED BY 'a-strong-password';
GRANT ALL PRIVILEGES ON photostore.* TO 'photostore'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

4.4 Enter your database credentials

Edit `includes/config.php` and set the four database constants to match what you just created:

```
define('DB_HOST', 'localhost');
define('DB_NAME', 'photostore');
define('DB_USER', 'photostore');
define('DB_PASS', 'a-strong-password');
```

This is the only file you ever need to edit by hand. Everything else is configured from the admin panel.

4.5 Set folder permissions

PHP must be able to write uploaded originals and generated previews. Give the web server ownership of the two storage folders:

```
cd /var/www/html/store
sudo chown -R www-data:www-data uploads cache
sudo chmod -R 775 uploads cache
```

4.6 Run the installer

1. Open <https://yourdomain.com/store/install.php> in your browser.
2. Enter your store name and create the administrator username, email and password (at least 8 characters).
3. Click Install. The installer creates all the database tables and seeds sensible defaults.
4. When it reports success, delete `install.php` from the server (see the security note below).
5. Click through to the sign-in page and log in with the admin account you just created.

Important — delete the installer. Once setup succeeds, remove `install.php` from your server. As a safeguard the installer automatically locks itself the moment an administrator account exists, but deleting the file removes any doubt.

4.7 Recommended PHP settings for large photos

High-resolution photos are large. If uploads fail or get truncated, raise these values in your PHP configuration (typically `/etc/php/8.x/apache2/php.ini`), then restart Apache:

Setting	Suggested value	Purpose
<code>upload_max_filesize</code>	64M	Largest single file accepted
<code>post_max_size</code>	256M	Total size of one upload request
<code>max_file_uploads</code>	60	How many files per drag-and-drop batch
<code>memory_limit</code>	256M	Headroom for generating previews
<code>max_execution_time</code>	120	Time allowed to process a big batch

5. Connecting PayPal & Credit-Card Payments

Aperture uses PayPal Checkout, which presents your customers with a PayPal button and a “Debit or Credit Card” button. This single integration therefore accepts both PayPal accounts and ordinary cards — you do not need Stripe or any other processor.

5.1 What you need

- **Business account.** A PayPal Business account (free to upgrade from a Personal account).
- **API credentials.** REST API credentials — a Client ID and Secret — created in the PayPal Developer Dashboard.
- **Matching currency.** Your store currency set to a currency your PayPal account supports.

5.2 Create your API credentials

6. Sign in at <https://developer.paypal.com> with your PayPal Business login.
7. Open Apps & Credentials from the dashboard menu.
8. Toggle between Sandbox (for testing) and Live (for real money) at the top of the page.
9. Click Create App, give it a name such as “Aperture Store”, and create it.
10. Copy the Client ID and the Secret that PayPal displays.

5.3 Enter the keys in Aperture

11. In the admin, go to Settings and scroll to Payments — PayPal & Cards.
12. Choose Sandbox while testing, or Live to take real payments.
13. Paste the Client ID and the Secret, set your currency, and save.

Checkout stays disabled until valid credentials are saved, so customers can never reach a broken payment screen.

Test before you go live. In Sandbox mode, PayPal gives you fake buyer accounts (under Sandbox > Accounts) you can use to place a complete test order — including the credit-card option — without spending real money. Verify the download links and the emailed receipt, then switch to Live and re-enter your Live keys.

5.4 How a payment flows

14. The customer clicks a PayPal or card button at checkout.
15. Your server asks PayPal to create an order for the exact cart total (recomputed server-side, never trusted from the browser).
16. The customer approves the payment in the PayPal pop-up.
17. Your server captures the payment and confirms PayPal reports it as COMPLETED.
18. Only then is the order marked paid, the receipt emailed, and the downloads unlocked.

6. Enabling SSL (HTTPS)

Secure checkout requires HTTPS. Modern browsers warn visitors away from non-secure payment pages, and PayPal expects your site to be served over TLS. The easiest and free option is a Let’s Encrypt certificate issued by Certbot.

6.1 Prerequisites

- A registered domain name (e.g. yourstudio.com) pointing at your server's IP address.
- Ports 80 and 443 open in your firewall.
- Apache already serving your site over plain HTTP.

6.2 Install Certbot and issue a certificate

On Ubuntu with Apache, run:

```
sudo apt install certbot python3-certbot-apache  
sudo certbot --apache -d yourstudio.com -d www.yourstudio.com
```

Certbot asks for an email, has you agree to the terms, and — crucially — offers to redirect all HTTP traffic to HTTPS. Choose the redirect option so every visitor lands on the secure version automatically.

6.3 Automatic renewal

Let's Encrypt certificates last 90 days. Certbot installs a timer that renews them automatically; confirm it works with a dry run:

```
sudo certbot renew --dry-run
```

6.4 Force HTTPS and harden cookies

If you did not let Certbot add the redirect, add this to your HTTP virtual host so visitors are always upgraded to HTTPS:

```
<VirtualHost *:80>  
    ServerName yourstudio.com  
    Redirect permanent / https://yourstudio.com/  
</VirtualHost>
```

Cookie tip: Once HTTPS is active, Aperture's session cookies are served over the secure channel automatically. For maximum hardening you may also set `session.cookie_secure = 1` in `php.ini` so cookies are never sent over plain HTTP.

7. Using the Site as a Customer

This section describes the experience your buyers have. No account or sign-up is required — anyone can browse and purchase.

7.1 Browsing galleries

The home page presents your galleries as a grid of cover images. Selecting a gallery opens its photos. Selecting any photo opens the full-screen viewer, where the visitor can move between images and see the price, dimensions and licensing note.

7.2 Adding to the cart

- From a gallery, each photo has an Add to cart button.
- From a photo's page, the visitor can Add to cart or use Buy now to go straight to checkout.
- The cart icon in the top bar always shows the number of items.

7.3 Checking out

19. On the cart page, the visitor reviews their items and selects Checkout.
20. They enter their name and email — this is where the receipt and download links are sent.
21. They pay with the PayPal button or the Debit or Credit Card button.
22. On success they are taken to their order page.

7.4 Downloading photos and the receipt

The order page lists every purchased photo with a Download original button that delivers the full-resolution file. A Download receipt (PDF) button provides the itemized receipt. The same receipt and a link back to this page are emailed automatically.

7.5 Returning to an order later

If a customer closes the page, they can return through Find my order in the footer and enter their order number and email to regain access to their downloads. The link in their emailed receipt also brings them straight back.

7.6 Clients with a private gallery

If the photographer has given you a private login, select Client sign in at the top right and enter your username and password. Your account shows only the galleries prepared for you. See section 9 for the full client experience.

8. Using the Site as an Administrator

Sign in at /store/login.php to reach the admin panel. The dark navigation bar across the top gives you Dashboard, Galleries, Upload, Clients, Orders and Settings.

8.1 Dashboard

The dashboard summarizes your galleries, photos, paid orders and total revenue, and lists your most recent orders. If PayPal is not yet configured, a reminder appears here.

8.2 Creating galleries

23. Go to Galleries and select New gallery.

24. Give it a title, an optional description and a default price.
25. Set visibility to Public (shown to everyone), Private (only the clients you assign — see section 9), or Hidden (only you).
26. Save. You can edit or delete a gallery at any time — deleting also removes its photos and files.

8.3 Uploading photos

27. Go to Upload (or choose Upload on a specific gallery).
28. Pick the destination gallery.
29. Optionally set a price for this batch; leave it blank to use the gallery's default.
30. Drag many photos onto the drop zone, or click to browse and multi-select.
31. Select Upload photos. Each image is stored and a watermarked preview is prepared automatically.

Supported formats: JPG, PNG, GIF and WebP. Filenames are turned into friendly titles automatically, which you can refine later.

8.4 Pricing, titles and cover photos

Open a gallery's Manage photos page to:

- Edit the title and price of each photo, then Save all changes.
- Use Apply to all to set one price across every photo in the gallery.
- Set any photo as the gallery's Cover, or delete photos you no longer wish to sell.

8.5 Managing orders

The Orders page lists every order with its status. Filter by Paid, Pending or Refunded, open any order to see the customer, payment reference and items, and re-download its PDF receipt whenever needed.

8.6 Settings reference

Everything below is configured from the Settings page — no code required.

Setting	What it controls
Store name	The name shown in the header, emails and receipts
Currency	The currency for all prices and PayPal charges
Accent color	The brand color used across the storefront and receipts
Hero kicker / title / subtitle	The headline text on the home-page banner
Watermark enabled	Turns the preview watermark on or off
Watermark text	The text tiled across previews (e.g. your studio name)
Watermark opacity	How visible the watermark is (5–90%)

PayPal mode	Sandbox (testing) or Live (real payments)
PayPal Client ID / Secret	Your REST API credentials
Support email	Shown to customers for help; used as the receipt contact
From address	The address receipts are sent from

Watermark & fonts: For the most attractive watermark, install the DejaVu fonts (sudo apt install fonts-dejavu) or drop your own TrueType font at assets/fonts/wm.ttf. Saving the settings page rebuilds all previews so changes take effect immediately. Original files are never watermarked.

9. Client Galleries & Private Access

Beyond your public storefront, Aperture lets you create private galleries and share them with specific clients through their own password-protected logins. This is ideal for privately delivering a wedding, a portrait session or any commissioned shoot — the client sees only what you prepared for them, and the public never sees it at all.

9.1 How gallery visibility works

Every gallery has one of three visibility levels, chosen when you create or edit it:

Visibility	Who can see it
Public	Everyone — listed on your storefront for all visitors and guests.
Private	Only the client accounts you assign to it. Hidden from guests and the public.
Hidden	Only you, the administrator — use for drafts that are not ready to share.

The golden rule: A signed-in client sees **ONLY** the galleries you have assigned to them — not your public galleries. This keeps each client’s experience personal and focused on their own photographs.

9.2 Create a client account

32. In the admin, open Clients and select New client.
33. Enter a username, the client’s full name (for example “The Smith Wedding”), an optional email, and a password.
34. Select Create client — the account is ready immediately.

9.3 Assign galleries to a client

35. First set the gallery’s visibility to Private (Galleries → Edit).
36. Open Clients, find the client, and select Assign galleries.
37. Tick every gallery this client should see, then select Save access.

You can assign many galleries to one client, and the same gallery to many clients. Changes take effect immediately.

9.4 Share the sign-in details

At the top of the Clients page you'll find the exact sign-in web address for your store. Send your client three things: that link, their username and their password. When they sign in, their home page greets them by name and shows only their galleries — they can browse, purchase and download just like any customer.

9.5 Manage existing clients

- **Edit.** Edit updates the client's name and email, or enables/disables the account.
- **Password.** Password sets a new password — useful if a client is locked out.
- **Disable.** Disabling an account blocks sign-in without deleting anything; re-enable it any time.
- **Delete.** Delete removes the account and its gallery assignments; the galleries themselves are untouched.

Automatic upgrade: The client-accounts feature upgrades your database automatically the first time you open the admin after installing it — there is no manual SQL to run, and your existing galleries, orders and admin account are untouched.

10. Maintenance & Best Practices

10.1 Back up regularly

Two things must be backed up: the database (your catalog and orders) and the uploads folder (your original files).

```
# Database
mysqldump -u photostore -p photostore > aperture-backup.sql

# Original files
tar czf aperture-uploads.tar.gz uploads/
```

Store these backups somewhere off the server. The cache/previews folder does not need backing up — it is regenerated automatically from the originals.

10.2 Security checklist

- Delete install.php after setup.
- Serve the whole site over HTTPS.
- Use a strong, unique admin password and a strong database password.
- Give each client a unique password, and disable or delete accounts you no longer need.

- Keep Ubuntu, PHP and MySQL updated with security patches.
- Leave the supplied .htaccess files in place — they block direct web access to your originals and PHP includes.

10.3 Email deliverability

Receipts are sent with the server’s mail system. If they are not arriving, install a mail transfer agent such as Postfix and ensure your From address uses your real domain. To dramatically improve deliverability, add SPF and DKIM DNS records for your domain, or relay outgoing mail through a transactional email provider.

11. Troubleshooting

The table below covers the most common issues and their fixes.

Symptom	Likely cause & fix
Blank white page or HTTP 500	A PHP error. Check the Apache error log (/var/log/apache2/error.log). Usually a missing extension (install php-gd, php-curl) or wrong database credentials in config.php.
Installer: SQL syntax error	Resolved in the current version. If seen on an older copy, drop and recreate the empty database, then re-run install.php.
“Payments are not configured” at checkout	Add your PayPal Client ID and Secret in Settings, and make sure the secret was saved (leave it blank only to keep an existing one).
PayPal auth failed (HTTP 401)	The Client ID/Secret are wrong, or they belong to the other mode. Sandbox keys only work in Sandbox mode; Live keys only in Live.
Previews are blank or broken	The GD extension is missing or the cache folder is not writable. Install php-gd and ensure cache/ is owned by www-data and is mode 775.
Watermark looks tiny or unstyled	No TrueType font was found. Install fonts-dejavu or add assets/fonts/wm.ttf, then re-save Settings to rebuild previews.
Uploads fail or stop partway	Raise upload_max_filesize, post_max_size, max_file_uploads and memory_limit in php.ini (see section 4.7), then restart Apache.
“This download is not available”	The order is not marked paid, or the link/token is wrong. Confirm the payment captured in Orders; the buyer should use the link in their receipt email.
Receipt email never arrives	The server has no working mailer. Install Postfix, verify the From address, and check the customer’s spam folder. The receipt can always be downloaded from the order page.
Cart empties between pages	Browser cookies are blocked, or PHP sessions cannot be written. Confirm cookies are enabled and the PHP session directory is writable.
A client can’t see their gallery	Confirm the gallery’s visibility is Private, that it is ticked for that client under

Clients → Assign galleries, and that the client's account is enabled.

A signed-in client sees no galleries

No galleries are assigned to them yet, or the account is disabled. Assign at least one gallery under Clients. (Remember: signed-in clients do not see public galleries.)

Still stuck? Almost every problem leaves a clue in the Apache error log. Watching it live while you reproduce the issue (`sudo tail -f /var/log/apache2/error.log`) is the fastest path to a diagnosis.

Appendix A. File & Folder Map

Path	Purpose
<code>index.php</code>	Storefront home (gallery grid + hero)
<code>collection.php</code> / <code>photo.php</code>	Browse a gallery / view a single photo
<code>cart.php</code> / <code>checkout.php</code>	Shopping cart and checkout
<code>order.php</code> / <code>download.php</code>	Order page and tokenized original downloads
<code>preview.php</code>	Generates watermarked previews & thumbnails
<code>install.php</code>	One-time setup (delete after running)
<code>includes/</code>	Core code: config, database, PayPal, PDF, email, layout
<code>admin/</code>	Administrator panel pages
<code>admin/clients.php</code>	Manage client accounts and private-gallery assignments
<code>assets/</code>	Stylesheet, JavaScript, optional watermark font
<code>uploads/</code>	Original high-resolution files (protected)
<code>cache/previews/</code>	Generated previews (safe to delete; auto-rebuilt)

— Thank you for using Aperture —