

SECOND LIFE

PHP + MySQL + LSL

Universal Setup Guide

for all web-connected Second Life systems

 Camera Visit Tracker	 PayLinden Payouts	 Donate Web	 Vendor Web
Tracks avatar visits, counts entries, shows leaderboard	Queues L\$ payouts, processes via in-world object	Accepts optional donations, logs donors & amounts	Lists products, records purchases, confirms delivery

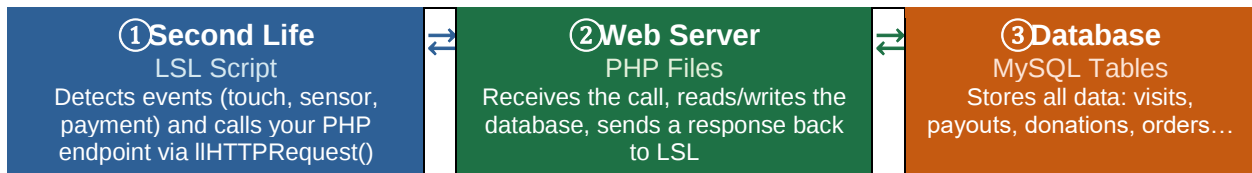
 Customer Register	
Auto-registers your item on rez so you'll always get future updates.	

... and more systems in the future

Version 2026 · JKD Style Project
[JKD-style](#)

How Every System Works

All JKD Style systems share the same three-layer architecture. Once you understand the pattern, setting up any new system takes only minutes. The only things that change between systems are the **PHP file names** and the **database table name**.

**KEY**

config.php is shared by ALL systems. Set it up once — every system uses the same database credentials file.

What You Will Need

Before starting any system, make sure you have access to all of the following:

What you need	Where to find it
Web hosting with PHP & MySQL	Your hosting provider (e.g. Wedos, Forpsi). Look for PHP version 7.4+ and MySQL / MariaDB.
FTP access to your web server	Hosting control panel → FTP accounts. You need: server address, username, password.
MySQL database credentials	phpMyAdmin or the Databases section in your hosting control panel.
Second Life account	secondlife.com — needed to place and edit in-world objects with scripts.
System files (PHP + LSL + SQL)	Provided separately for each system. Always includes config.php, one or more .php files, one .lsl file, and a .sql or sql.txt file.

PART 1 — One-Time Setup (Do This Once)

Everything in this section is done only once, even if you install multiple systems. If you have already completed these steps for a previous system, skip directly to Part 2.

Step 1 · Edit config.php

Every system ships with the same **config.php** file. Open it in any text editor (Notepad, VS Code, Notepad++) and fill in your four database credentials:

```
<?php
$hostname = 'wm36.hosting.net'; // MySQL server address from your hosting provider
$user     = 'a23381_sl';       // database username
$pass     = 'YOUR_PASSWORD';  // database password
$dbase    = 'd23381_sl';       // database name
?>
```



Never share config.php publicly or commit it to a public git repository. It contains your database password.

You will find all four values in your hosting control panel under Databases or MySQL. The server address is often listed as 'hostname' or 'MySQL host'.

Step 2 · Upload config.php via FTP

Use a free FTP client such as **FileZilla** (filezilla-project.org) to upload config.php to your web server.

1. Open FileZilla and connect using your FTP credentials.
2. Navigate to (or create) the target folder — e.g. /public_html/scripts/
3. Drag config.php from your computer into that folder.
4. That's it. Every system installed in the same folder will automatically use this file.



The folder name is up to you. Common choices: /sl/, /scripts/, /secondlife/. Just be consistent — use the same folder for all your systems.

PART 2 — Per-System Setup (Repeat for Each System)

Follow these steps for every new system you install. The process is always identical — only the file names and table names differ.

Step 3 · Create the Database Table

Each system needs its own table in MySQL. The required SQL command is included in the system's `sql.txt` file.

5. Log in to your hosting control panel and open phpMyAdmin.
6. In the left panel, click the name of your database.
7. Click the SQL tab at the top.
8. Open the system's `sql.txt` file, copy the entire contents, and paste it into phpMyAdmin.
9. Click Go. The table will be created.
10. Verify: the new table should appear in the left panel.



If you see an error saying 'Table already exists', the system was installed before. That is fine — skip this step.

Step 4 · Upload the PHP Files

Upload all `.php` files for this system (*except `config.php` if already uploaded*) to the same folder on your web server.

File	Upload to
<code>config.php</code>	<code>/scripts/config.php</code> (skip if already there)
<code>[system_name].php</code>	<code>/scripts/[system_name].php</code>
<code>list_[system_name].php</code> (if included)	<code>/scripts/list_[system_name].php</code>



The exact file names depend on the system. Each system's documentation lists the files included. The folder must match what you used for `config.php`.

Step 5 · Test the PHP Endpoint

Before touching Second Life, always verify that the PHP side is working.

11. Open your browser and navigate to the PHP file URL, e.g.:

```
https://www.yourdomain.com/scripts/[system_name].php
https://www.yourdomain.com/scripts/list_[system_name].php
```

12. Expected result: You should see a short text response (not a PHP error). The exact text depends on the system — it may say 'empty', '0', or a similar short status message.
13. If you see a blank white page or PHP error text: double-check `config.php` credentials and confirm the database table was created.

Step 6 · Place and Configure the LSL Script

Each system includes one `.lsl` script file that goes into an in-world Second Life object.

14. In Second Life, rez a prim (or use an existing object) where you want the system to run.
15. Right-click the object → Edit → open the Content tab.
16. Click New Script, delete all default content, and paste the full contents of the `.lsl` file.
17. Find the `baseURL` line near the top of the script and update it to your PHP file URL:

```
string baseURL = "https://www.yourdomain.com/scripts/[system_name].php";
```

18. Click Save. The script will compile and start running.
19. Touch the object to trigger an immediate test call, or wait for the built-in timer.



The URL must match exactly what you tested in Step 5. Include `https://` (or `http://`) and the correct folder path.

PART 3 — System Quick Reference

Use this section as a cheat sheet when installing a specific system. For each system you will find: the files included, the SQL table, and the LSL script line to update.

□ Camera / Visit Tracker

Automatically detects avatars entering sensor range, counts their visits, and records timestamps. A public web page shows a ranked leaderboard.

File	Type	Location
config.php	PHP – shared	Web server /scripts/
statistika.php	PHP – endpoint	Web server /scripts/
list_statistika.php	PHP – public page	Web server /scripts/
sql.txt	SQL – run once	phpMyAdmin only
webcam.lsl	LSL – sensor	SL object content
http.lsl	LSL – HTTP sender	SL object content

Database table created by sql.txt:

```
CREATE TABLE IF NOT EXISTS statistika (
  id          INT AUTO_INCREMENT PRIMARY KEY,
  avatar      VARCHAR(255) NOT NULL,
  visit       INT NOT NULL DEFAULT 1,
  created_at  TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  UNIQUE (avatar),
  INDEX (visit)
);
```

Line to update in http.lsl:

```
string baseURL = "https://www.yourdomain.com/scripts/statistika.php";
```



Both LSL scripts (webcam.lsl and http.lsl) must be in the SAME object. They communicate via lLMessageLinked — webcam detects avatars, http sends data to the web.

□ PayLinden — Automated Payouts

Queues Linden Dollar payouts in the database. An in-world object with DEBIT permission periodically polls the PHP endpoint and pays out any pending amounts.

File	Type	Location
config.php	PHP – shared	Web server /scripts/
paylinden.php	PHP – endpoint	Web server /scripts/
admin_prehled.php	PHP – admin panel	Web server /scripts/
uloz_jmeno.php	PHP – save name avatar	Web server /scripts/
pass.php	PHP – generate pass	Web server /scripts/
SQL_scheme	SQL – run once	phpMyAdmin only

File	Type	Location
payLinden.lsl	LSL – payout script	SL object content

Database tables created by SQL_scheme:

```
CREATE TABLE payout (
  id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  uuid    VARCHAR(36) NOT NULL UNIQUE,
  castka  INT UNSIGNED NOT NULL DEFAULT 0,
  created DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE IF NOT EXISTS avatar_names (
  uuid          VARCHAR(36) NOT NULL PRIMARY KEY,
  display_name  VARCHAR(100) NOT NULL,
  updated_at    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE payout_log (
  id      INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  uuid    VARCHAR(36) NOT NULL,
  castka  INT UNSIGNED NOT NULL,
  vyplaceno_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

Line to update in **paylinden.lsl**:

```
string baseURL = "https://www.yourdomain.com/scripts/paylinden.php";
string nameURL = "https://www.yourdomain.com/scripts/uloz_jmeno.php";
```



The object running payLinden.lsl must have DEBIT permission granted. Without it, llGiveMoney() will fail silently.

□ Donate Web

Allows visitors to make optional L\$ donations. Records the donor UUID, avatar name, amount, and timestamp. An admin page shows donation totals and top donors.

File	Type	Location
config.php	PHP – shared	Web server /scripts/
donate.php	PHP – endpoint	Web server /scripts/
list_donate.php	PHP – public/admin page	Web server /scripts/
SQL_scheme	SQL – run once	phpMyAdmin only
donate.lsl	LSL – donation handler	SL object content

Database table created by SQL_scheme:

```
CREATE TABLE `donate` (
  `id`      INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `mDarce`  VARCHAR(63) NOT NULL,
  `amount`  INT UNSIGNED NOT NULL,
  `uuid`    VARCHAR(36) NULL,
  `created_at` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  UNIQUE KEY `uq_uuid` (`uuid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Line to update in **donate.lsl**:

```
string baseURL = "https://www.yourdomain.com/scripts/donate.php";
```

□ Sell Web — Product Sales

Lists products with prices. When a visitor pays the in-world object the correct amount, the purchase is recorded and a delivery confirmation is sent via IM.

File	Type	Location
config.php	PHP - shared	Web server /scripts/
vendor.php	PHP - endpoint	Web server /scripts/
list_vendor.php	PHP - sales overview	Web server /scripts/
SQL_scheme	SQL - run once	phpMyAdmin only
!Settings	LSL - note settings	
vendor.lsl	LSL - shop script	SL object content
http.lsl	LSL - HTTP sender	

Database table created by **sql_scheme**:

```
CREATE TABLE IF NOT EXISTS vendor (
  id INT AUTO_INCREMENT PRIMARY KEY,
  product VARCHAR(255) NOT NULL,
  count INT NOT NULL DEFAULT 0,
  amount DECIMAL(10,2) NOT NULL DEFAULT 0,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  UNIQUE (product),
  INDEX (amount)
);
```

Line to update in **http.lsl**:

```
string baseURL = "https://www.yourdomain.com/scripts/vendor.php";
```

□ Customer Register

Registers avatars as customers when they touch or interact with the object. Stores UUID, name, and registration date. Useful for loyalty programs, membership lists, or event sign-ups.

File	Type	Location
config.php	PHP – shared	Web server /scripts/
registrace.php	PHP – endpoint	Web server /scripts/
registrace_vypis.php	PHP – member list page	Web server /scripts/
SQL_scheme	SQL – run once	phpMyAdmin only
registrace.lsl	LSL – registration script	SL object content

Database table created by **sql_scheme**:

```
CREATE TABLE IF NOT EXISTS `customer` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `customer_key` VARCHAR(255) NOT NULL, -- UUID avatar ze Second Life  
  `first_name` VARCHAR(100) NOT NULL,  
  `last_name` VARCHAR(100) NOT NULL,  
  `my_product` VARCHAR(100) NOT NULL,  
  `registered_at` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `unique_customer_product` (`customer_key`, `my_product`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Line to update in **on_rez_register.lsl** and **touch_to_register.lsl**:

```
string PHP_url = "https://server.com/scripts/customer_register.php"; //← update  
string PHP_owner_key = "12345678-1234-1234-1234-123456789abc"; //← update  
string my_product = "Produkt_A"; // ← update
```

Troubleshooting

The same issues appear across all systems. Check this table before asking for help.

Symptom	Likely cause & fix
Nothing happens when I touch the object	Check that the LSL script compiled without errors (look in the Script Editor for red error lines). Also verify the URL in baseURL is correct and reachable.
Object turns red / unexpected colour	This usually means the PHP endpoint returned an unexpected response. Open the URL in a browser to see what it actually returns.
PHP shows a blank page or error	Wrong credentials in config.php, or the database table does not exist. Re-check Step 1 and Step 3.
'Table already exists' error in phpMyAdmin	The table was created during a previous install. This is not an error — skip the SQL step.
Data not appearing on the web page	No events have been triggered yet (no avatar detected, no payment, etc.). Touch the object, then refresh the page.
LSL script says 'Function not supported'	The object may need permissions. For PayLinden: DEBIT permission must be granted. Touch the object and accept the permission dialog.
HTTP response comes back empty	The PHP file path in baseURL may have a typo, or the file was not uploaded. Test the URL in a browser first.
payout table shows amount but nothing is paid	The in-world payout object must be online and owned by the account that has L\$. Check the object is rezzed in-world, not just in inventory.

Complete File Reference

Every file used across all systems, with its role and destination.

File name	System	Purpose
config.php	ALL	Database credentials. Shared by every system. Edit once, never again.
statistika.php	Camera	Receives avatar name from LSL, upserts visit count in DB.
statistika_vypis.php	Camera	Public leaderboard page — shows all visitors ranked by visit count.
vypiaceni_banku.php	PayLinden	Returns next pending payout UUID+amount, zeroes the record.
admin_prehled.php	PayLinden	Admin dashboard — totals, pending queue, payout history.
donate.php	Donate	Records incoming donation with UUID, avatar name, and amount.
donate_vypis.php	Donate	Public/admin page showing donation totals and donor list.
prodej.php	Sell	Logs purchase: product, buyer UUID, avatar name, amount paid.
prodej_vypis.php	Sell	Sales overview — all transactions ordered by date.
registrace.php	Customer Reg.	Registers new customer UUID and avatar name (ignores duplicates).
registrace_vypis.php	Customer Reg.	Member list — all registered avatars with registration date.
webcam.lsl	Camera	Sensor script: detects nearby avatars, sends name to http.lsl.
http.lsl	Camera	HTTP script: sends avatar name to PHP, displays welcome message.
payLinden.lsl	PayLinden	Polls PHP every 90 seconds, pays out pending L\$ via lIGiveMoney().
donate.lsl	Donate	Handles money_event, posts donation to PHP endpoint.
prodej.lsl	Sell	Handles payment, matches amount to product, records sale.
registrace.lsl	Customer Reg.	On touch: sends avatar UUID+name to PHP for registration.
sql.txt (each system)	Each system	SQL to create that system's table. Run once in phpMyAdmin.

i

Future systems follow the same pattern. Add a new .php endpoint, a new .lsl script with the correct baseUrl, and a new sql.txt table. config.php never needs to change.