

04 System Architecture

Kingdom Wife Blueprint — System Architecture

The platform stack. PHP + IONOS + MySQL · matches Sandi's locked pattern from membership-site-builder.

Stack lock (non-negotiable)

Layer	Choice	Why
Backend	PHP (no Node, no Django, no Laravel)	Matches her existing <code>bootstrap.php</code> / <code>auth.php</code> template set
Database	MySQL via phpMyAdmin	<code>db809893484</code> already provisioned on IONOS
Hosting	IONOS shared hosting (LAMP)	Already paid · DFI Training Academy on same
Frontend	Server-rendered PHP + minimal JS	Matches her static-first approach
Payments	Stripe (test → live cutover)	Industry standard · low integration cost
Email	IONOS PHP mailer OR ConvertKit	Decide based on volume; PHP-mailer for <500 sends/day

Layer	Choice	Why
Calendar	Calendly for cohort calls	Already in her tool stack
Video conf	Zoom (paid) for cohort calls; recordings to private S3 or Bunny CDN	Avoid Vimeo (PHP integration friction)
File storage	IONOS file storage for PDFs · S3 if scale demands	Most files <50MB · IONOS sufficient initially

Folder structure (under `public_html/kingdom-wife/`)

```

public_html/kingdom-wife/
├─ admin/                ← Sandi only (require_admin)
│   ├── dashboard.php
│   ├── cohort-tracking.php
│   ├── purchase-log.php
│   ├── email-queue.php
│   └─ reports/
│       └─ monthly.php
├─ includes/
│   ├── bootstrap.php    ← session start · DB connect · helpers
│   ├── auth.php         ← login · logout · require_login · require_tier
│   ├── stripe-webhook.php
│   ├── email-sender.php
│   └─ content-gate.php   ← tier-based content access checks
├─ docs/                 ← downloadable PDFs (workbook · scripture cards)
├─ database/
│   └─ schema.sql        ← canonical schema · version-tracked
├─ styles/
│   ├── kwb.css          ← Sandi Personal palette tokens
│   └─ kwb-print.css
├─ images/
├─ js/
│   └─ app.js            ← minimal · journal save · cohort RSVP
├─ content/              ← course modules · gated by tier
│   ├── module-1-inner-life/
│   │   ├── lesson-01.php
│   │   ├── lesson-02.php
│   │   └─ ...
│   ├── module-2-relational/
│   ├── module-3-authority/
│   └─ module-4-leadership/
├─ cohort/              ← Tier 3 only (require_tier 3)
└─ home.php

```

```

|   ├── recordings.php
|   └── accountability-pairs.php
|   ├── signup.php
|   ├── login.php
|   ├── logout.php
|   ├── dashboard.php           ← user-facing dashboard (their tier · their progress)
|   ├── account.php
|   ├── checkout.php           ← Stripe redirect
|   ├── thank-you.php
|   ├── index.php              ← public landing (Three Truths · sample chapter · signup CTA)
|   └── .htaccess               ← security · clean URLs · auth redirects

```

MySQL schema (extends existing `db809893484`)

-- USERS table (extend existing if shared with DFI; otherwise new)

```

CREATE TABLE kwb_users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  email VARCHAR(255) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  first_name VARCHAR(100),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  last_login TIMESTAMP NULL,
  current_tier TINYINT DEFAULT 0,           -- 0=none, 1=book, 2=course, 3=cohort
  tier_expires_at TIMESTAMP NULL,          -- for monthly tier 3
  is_admin BOOLEAN DEFAULT FALSE,
  email_verified BOOLEAN DEFAULT FALSE,
  newsletter_opt_in BOOLEAN DEFAULT TRUE,
  INDEX (email), INDEX (current_tier)
);

```

-- PURCHASES table

```

CREATE TABLE kwb_purchases (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  stripe_payment_intent VARCHAR(255) UNIQUE,
  tier_purchased TINYINT NOT NULL,
  amount_cents INT NOT NULL,
  currency CHAR(3) DEFAULT 'USD',
  status ENUM('pending', 'succeeded', 'failed', 'refunded') NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES kwb_users(id),
  INDEX (status), INDEX (created_at)
);

```

-- COHORT membership (Tier 3 only)

```

CREATE TABLE kwb_cohort_membership (
  id INT AUTO_INCREMENT PRIMARY KEY,

```

```

user_id INT NOT NULL,
cohort_number INT NOT NULL,
joined_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
left_at TIMESTAMP NULL,
accountability_partner_id INT NULL,
attended_calls INT DEFAULT 0,
total_calls_during_membership INT DEFAULT 0,
FOREIGN KEY (user_id) REFERENCES kwb_users(id),
FOREIGN KEY (accountability_partner_id) REFERENCES kwb_users(id),
INDEX (cohort_number), INDEX (left_at)
);

-- JOURNAL entries (private to user - encrypted at rest if budget allows)
CREATE TABLE kwb_journal_entries (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  prompt_id VARCHAR(100),          -- which prompt this answers
  entry_text MEDIUMTEXT,
  is_shared_with_partner BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES kwb_users(id),
  INDEX (user_id, created_at)
);

-- EMAIL queue (drives Day 0/3/7/14/30 sequence)
CREATE TABLE kwb_email_queue (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  email_template VARCHAR(50) NOT NULL,    -- e.g., 'day-0-welcome', 'day-3-intro'
  scheduled_for TIMESTAMP NOT NULL,
  sent_at TIMESTAMP NULL,
  status ENUM('pending','sent','failed','cancelled') DEFAULT 'pending',
  FOREIGN KEY (user_id) REFERENCES kwb_users(id),
  INDEX (status, scheduled_for)
);

-- EMAIL Log (after-the-fact record)
CREATE TABLE kwb_email_log (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  email_template VARCHAR(50),
  sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  opened_at TIMESTAMP NULL,
  clicked_link VARCHAR(255) NULL,
  FOREIGN KEY (user_id) REFERENCES kwb_users(id)
);

-- CONTENT progress (which Lessons a user has completed)
CREATE TABLE kwb_content_progress (
  user_id INT NOT NULL,
  content_id VARCHAR(100) NOT NULL,      -- e.g., 'module-1-lesson-3'

```

```
completed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
PRIMARY KEY (user_id, content_id),  
FOREIGN KEY (user_id) REFERENCES kwb_users(id)  
);
```

Auth flow

1. User signs up via `/signup.php` (email + password + tier choice).
 2. Password hashed with PHP `password_hash()` (bcrypt cost 12).
 3. Session cookie set via `bootstrap.php` (HttpOnly · Secure · SameSite=Lax).
 4. `auth.php::require_login()` and `auth.php::require_tier($n)` gate every protected page.
 5. Stripe checkout redirect on tier purchase. Webhook back to PHP confirms payment → `kwb_purchases` record → `kwb_users.current_tier` updated.
-

Security checklist (per `membership-site-builder` SKILL)

- ✓ HTTPS-only (IONOS auto-SSL)
 - ✓ HttpOnly + Secure + SameSite cookies
 - ✓ CSRF tokens on all forms (submit + journal)
 - ✓ Password hashed with bcrypt
 - ✓ SQL: prepared statements only (PDO with `?` placeholders)
 - ✓ `.htaccess` denies access to `/includes/` , `/database/` , `*.sql` , `*.bak`
 - ✓ Stripe webhook signature verification
 - ✓ Rate limiting on `/login.php` (5 attempts/15 min)
 - ✓ Audit log table for admin actions
 - ✓ Backup: nightly DB dump · weekly file backup · 30-day retention
-

Cross-references

- `_skills/membership-site-builder/SKILL.md` — parent stack pattern
- `_skills/kingdom-wife-membership-platform/SKILL.md` — KWB-specific spec + Stack Lock
- `_knowledge/SECURITY_PRIVACY.md` — what stays out of git

- `02_action_plan.md` — Phase 2 build sequence
- `05_automation_plan.md` — what runs without Sandi