

Day 6 – Public Rendering, Email Notifications & Polishing

Day 6 (8 hours). After Day 5 (Auth + Admin-lite CRUD), Day 6 focuses on **Public Content Rendering + Notifications + Polishing the DevOps flow**. This makes the training sites “client-ready”: leads flow to admin + email, pages created in admin show on the public site, and the delivery pipeline feels professional.

Outcomes (by EOD)

- Admin-created **pages** render on the public site.
- Public site loads **published pages** dynamically from DB.
- **Leads/contact submissions** trigger **email notifications** (with SES/SMTP fallback to log).
- Error handling, logging, and backups improved.
- Students deliver a **production-like semi-dynamic site**.

1. Setup & Warm-up (30 min)

- New branch: `day6-public-rendering`.
- Review Day 5 deliverables: `users`, `pages`, `leads`.
- Set up **PHPMailer** (via Composer).
- Configure `.env` for mail creds (use dummy SMTP or AWS SES sandbox).

2. Reading Pack (30 min)

- [PHP Manual – Include & Require](#) → for templating layouts.
- [PHPMailer GitHub Docs](#) → install & quick start.
- [OWASP Logging & Monitoring Cheat Sheet](#).

3. Videos (30 min)

- Traversy Media: [PHP Templating Basics](#) (~20 min).
 - [PHPMailer Tutorial](#) (~10 min).
-

4. Guided Build (3 hrs)

A) Layout System

- Create `views/layout.php` with header/footer HTML.
- Create `views/page.php` → outputs title, content_html.

In `public/index.php` add simple router:

```
$slug = $_GET['page'] ?? 'home';
$stmt = $pdo->prepare("SELECT * FROM pages WHERE slug=? AND
status='published' LIMIT 1");
$stmt->execute([$slug]);
$page = $stmt->fetch();
if (!$page) { http_response_code(404); include 'views/404.php'; exit;
}
include 'views/page.php';
```

-

B) Public Rendering of Admin Pages

- Pages created in `/admin` → published → appear at `/index.php?page=slug`.
- Navbar dynamically populated with published slugs.
- Escape all user-provided fields (`htmlspecialchars` except `content_html`).

C) Contact Notifications (Email)

- In `public/contact.php` after DB insert:
 - Use PHPMailer to send admin email:
 - Subject: "New Lead Submission"
 - Body: name/email/message
 - If mail fails, log to `/storage/logs/mail-fail.log`.

D) Logging & Error Handling

Create `app/logger.php`:

```
function log_error($msg) {  
    file_put_contents(__DIR__.'../../storage/logs/app.log', date('c')."  
    ".$msg."\n", FILE_APPEND);  
}
```

-
- Wrap DB/Email operations in `try { ... } catch {
 log_error($e->getMessage()); }.`
- In production: disable `display_errors`, log only.

E) Backup & Recovery SOP

- Add `scripts/db-backup.sh`: dumps DB to `storage/backups/db-YYYYMMDD.sql`.
- Document "restore drill" → import into MySQL test instance.

5. Independent Labs (2 hrs)

Lab 1 – Public Pages

- Create 3 pages in admin ([about](#), [services](#), [pricing](#)).
- Publish them → verify at [/index.php?page=about](#) etc.

Lab 2 – Navbar Sync

- Ensure nav auto-updates when new page is published.

Lab 3 – Contact Notification

- Submit valid form → verify:
 - DB row inserted in [leads](#).
 - Email sent (or logged).

Lab 4 – Error Simulation

- Break DB connection → submit form → error logged, safe message shown.

Lab 5 – Backup

- Run backup script → [.sql](#) file created.
- Restore DB into fresh instance → verify pages appear.

6. QA & Wrap-Up (1 hr)

- Test public + admin flows end-to-end.
- Run Lighthouse → ≥90 desktop, ≥85 mobile.
- Validate SEO (sitemap, robots, meta).
- Confirm logs show both app and mail errors.

- Push branch with commits: `layout`, `render`, `mailer`, `logging`, `backup`.
-

Acceptance Criteria (Trainer)

- Admin-created pages visible to public at slug routes.
 - Only `published` pages render; drafts/archived hidden.
 - Contact form stores lead in DB **and** sends/logs email.
 - Logs written on failure (DB/mail).
 - Backup script exists, tested by restore.
 - Git branch `day6-public-rendering` with ≥ 5 commits.
-

Deliverables (EOD)

- `views/layout.php`, `views/page.php`.
 - `public/index.php` (router).
 - `public/contact.php` updated with mailer.
 - `app/logger.php`.
 - `scripts/db-backup.sh` + sample backup file.
 - Screenshots: public pages, contact success, admin listing, error log.
-

Stretch Goals (if time left)

- Add **slug-based pretty URLs** (`.htaccess` rewrite `/about` → `index.php?page=about`).
- Add **breadcrumbs + SEO tags** from `meta_json`.
- Build **admin setting** for “site name” → reflected in public layout.

Day 6 – Evaluation Checklist (Public Rendering, Email, Logging)

Area	Criteria	Points
1) Project Setup	Branch <code>day6-public-rendering</code> created and runs in browser.	3
	<code>views/</code> , <code>scripts/</code> , and <code>storage/logs</code> folders created and organized.	2
Subtotal		5
2) Layout & Routing	<code>views/layout.php</code> and <code>views/page.php</code> created with header/footer structure.	4
	<code>public/index.php</code> uses slug-based routing (<code>?page=about</code>).	4
	404 page shows when slug not found.	2
Subtotal		10
3) Public Rendering	Pages created in admin show on public site.	5
	Only <code>status='published'</code> pages display; drafts/archived hidden.	5
	Navbar auto-populates with published pages.	5
Subtotal		15
4) Contact Form Integration	Submitting valid form inserts into <code>leads</code> DB table.	5

	Success message shown to user (safe UI, no raw errors).	3
	Email notification sent with PHPMailer OR logged if mail fails.	7
	CSRF token still validated on contact form POST.	5
Subtotal		20
5) Logging & Error Handling	<code>app/logger.php</code> writes errors to <code>storage/logs/app.log</code> .	5
	DB/mail errors logged properly (tested by forcing failure).	5
	User sees generic safe error message, not raw PHP stack trace.	5
Subtotal		15
6) Backup & Recovery	<code>scripts/db-backup.sh</code> exists and runs without error.	5
	Backup file (<code>.sql</code>) created with timestamp.	3
	Restore tested → DB data appears correctly.	7
Subtotal		15
7) QA & SEO	Lighthouse score ≥90 desktop, ≥85 mobile.	5
	Sitemap.xml and robots.txt present.	3
	Meta tags and title populated from DB/meta_json.	2
Subtotal		10
8) Git & Hygiene	≥5 meaningful commits (layout, render, mailer, logging, backup).	3
	Clear commit messages.	2
Subtotal		5



Scoring Bands

- **90–100:** Excellent – production-ready semi-dynamic site.
 - **75–89:** Good – minor polish needed (SEO/logging).
 - **60–74:** Needs improvement – fix rendering/mailer.
 - **<60:** Must redo labs with trainer help.
-



Deliverables to Collect

- `views/layout.php`, `views/page.php`.
- `public/index.php` (router with 404).
- Updated `public/contact.php` with PHPMailer.
- `app/logger.php`.
- `scripts/db-backup.sh` + sample `.sql` backup.
- Screenshots:
 - Published page on public site.
 - Contact form success + lead in DB.
 - Error log file.
 - Backup + restore proof.

Debugging Cheat Sheet

A) Common Problems & Fixes

Problem	Symptoms	Likely Cause	Quick Fix
Slug router shows blank/404	?page=about returns 404	Wrong query, status not published, missing \$_GET['page'] default	Ensure default \$slug = \$_GET['page'] ?? 'home'; query: WHERE slug=? AND status='published'; verify row exists.
Layout not applied	Header/footer missing	Not including layout wrapper	Use a layout pattern: include 'views/layout.php' and inside it include \$view;. Or in page.php include header/footer partials.
Content shows as raw HTML or double-escaped	<h2> visible as text OR no escaping	Wrong escaping policy	For title/meta → htmlspecialchars(...); for content_html (trusted from admin) → output raw or sanitize server-side once.
Navbar not updating	New pages don't appear	Navbar hard-coded	Build nav from DB: query published pages and loop; clear cache if any.
PHPMailer fails	No emails; error/timeout	Bad SMTP creds/port; TLS mismatch; SES sandbox	Log \$mail->ErrorInfo; verify host/port (587 or 465), auth, from-domain verified; in sandbox, only verified recipients work.
Mailer halts page	White screen on send	Exceptions not caught	Wrap send in try/catch; on catch → log error + show safe message.
Logs not written	app.log empty	Wrong path or no write perms	Ensure path: storage/logs/app.log;

			create dirs; set permissions; use absolute path in logger.
Backup script fails	No <code>.sql</code> file created	mysqldump not found; auth missing	Use full path to <code>mysqldump</code> ; store DB creds via env; echo errors; test command manually.
Restore doesn't reflect pages	Pages still missing	Imported wrong DB/schema	Confirm DB name; <code>SHOW TABLES</code> ; <code>SELECT COUNT(*) FROM pages</code> ; Clear ORM cache if any.
404 page never triggers	Always shows layout	Condition never reached	Check <code>\$page</code> falsey condition before include; after 404: <code>exit</code> ;
Mixed HTTP/HTTPS issues	Mail/images blocked; console warnings	Inconsistent scheme	Use site base URL config; set correct links; if on HTTPS set <code>secure</code> cookies.
Robots/sitemap issues	SEO tools can't find files	Wrong paths/perms	Place <code>sitemap.xml</code> & <code>robots.txt</code> in <code>public/</code> ; verify URLs in browser.
Slow page due to images	Lighthouse < 85 mobile	Large images; no caching	Resize/compress; serve WebP; set cache headers; lazy-load images.
Undefined index notices	PHP notices in logs	Accessing unset <code>\$_GET/\$_POST</code>	Use null coalescing: <code>\$_GET['page'] ?? 'home'</code> ; validate inputs before use.
Headers already sent	Redirects/logins fail	Output before <code>header()</code>	Remove BOM/whitespace; send headers before HTML; buffer output if needed.

B) 5-Step Diagnostic Workflow (fast)

Router sanity

```
var_dump($_GET['page'] ?? 'NULL'); // confirm slug
```

1. Run the page query and `var_dump($page)`; if null → check status/slug.

DB connectivity

```
$pdo->query("SELECT 1"); // if error, fix DSN/creds
```

- 2.

Mailer probe

```
try { $ok = $mail->send(); var_dump($ok); } catch (Throwable $e) {  
echo $e->getMessage(); }
```

3. If using SES sandbox, test with verified recipient.

Logger check

```
log_error('Test log'); // confirm line appears in storage/logs/app.log
```

- 4.

5. **Backup/restore**

Run `scripts/db-backup.sh`; confirm a timestamped `.sql` appears; import into a fresh DB and verify `pages/leads` counts.

C) Copy-Paste Snippets

1) Minimal Router (public/index.php)

```
<?php  
require __DIR__.'../app/db.php';  
$slug = $_GET['page'] ?? 'home';  
$stmt = $pdo->prepare("SELECT slug,title,content_html,meta_json FROM  
pages WHERE slug=? AND status='published' LIMIT 1");  
$stmt->execute([$slug]);  
$page = $stmt->fetch();  
if (!$page) { http_response_code(404); include  
__DIR__.'../views/404.php'; exit; }
```

```
include __DIR__.'../views/page.php';
```

2) Safe Output Policy

```
<h1><?= htmlspecialchars($page['title'], ENT_QUOTES, 'UTF-8') ?></h1>
<div class="content"><?= $page['content_html'] /* trusted from admin
*/ ?></div>
```

3) Navbar from DB

```
$nav = $pdo->query("SELECT slug,title FROM pages WHERE
status='published' ORDER BY title")->fetchAll();
foreach ($nav as $li) {
    echo '<a
href="/index.php?page='.urlencode($li['slug']).">'.htmlspecialchars($
li['title']).'</a>';
}
```

4) PHPMailer with Try/Catch + Fallback Log

```
use PHPMailer\PHPMailer\PHPMailer;
require __DIR__.'../vendor/autoload.php';
require __DIR__.'../app/logger.php';

$mail = new PHPMailer(true);
try {
    $mail->isSMTP();
    $mail->Host = $_ENV['SMTP_HOST'];
    $mail->SMTPAuth = true;
    $mail->Username = $_ENV['SMTP_USER'];
    $mail->Password = $_ENV['SMTP_PASS'];
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = 587;
    $mail->setFrom($_ENV['MAIL_FROM'], 'Site Bot');
    $mail->addAddress($_ENV['MAIL_TO']);
    $mail->Subject = 'New Lead Submission';
    $mail->Body = "Name: $name\nEmail: $email\n$message";
    $mail->send();
}
```

```
} catch (Throwable $e) {  
    log_error('MAIL_FAIL: ' . $e->getMessage());  
}
```

5) Logger

```
<?php // app/logger.php  
function log_error($msg){  
    $path = __DIR__.'../../storage/logs/app.log';  
    if (!is_dir(dirname($path))) { mkdir(dirname($path), 0775, true); }  
    file_put_contents($path, date('c').' ' . $msg.PHP_EOL, FILE_APPEND);  
}
```

6) Backup Script (bash)

```
#!/usr/bin/env bash  
set -e  
STAMP=$(date +%Y%m%d-%H%M%S)  
OUT="storage/backups/db-$STAMP.sql"  
mkdir -p storage/backups  
mysqldump -h 127.0.0.1 -u"$DB_USER" -p"$DB_PASS" "$DB_NAME" > "$OUT"  
echo "Backup -> $OUT"
```

D) Quick Acceptance Re-checks (spot test)

- `/index.php?page=about` loads with layout; 404 works.
- Navbar shows **published** pages only.
- Contact submit → DB row + email (or mail log).
- `app.log` records a forced error.
- Backup `.sql` exists and restores cleanly.