■# Alumni Portal: Complete Technical Verification and AWS Readiness Report

Generated on: 2026-02-20 00:48:00
Project path: C:/xampp/htdocs/alumni_portal

1. Executive Status

* Local web stack: Apache running from XAMPP
* Local PostgreSQL: running and accepting connections on 127.0.0.1:5432
* DB smoke check: db_test.php passes
* API syntax check: pass across all PHP files
* API runtime GET sweep: 0 runtime fatals across 77 endpoints
* AWS IaC scaffold: present (terraform/, deployment/, Dockerfile, GitHub workflow)
* Local fallback mode: configured and documented

2. Critical Fixes Applied

2.1 Legacy Auth/DB compatibility fixes

The following endpoints were upgraded from legacy patterns
(new Auth(), authenticate(), Database->connect())
to the active middleware/DB contract
(new Auth($db), validateRequest(), getConnection()):

* api/apply_job.php
* api/create_event.php
* api/get_events.php
* api/pin_post.php
* api/unpin_post.php

2.2 Functional bug fixes

* api/get_messages.php
- Fixed typo: file_get_content() -> file_get_contents()
* api/search_directory.php
- Fixed undefined role context ($user was not defined)
- Added current-user role lookup and admin privacy behavior
* models/Event.php
- Aligned model queries/inserts with actual DB schema:
- creator_user_id (not creator_id)
- start_datetime/end_datetime (not event_date/event_time)
- banner_image_url, location_type, location_address, virtual_link
- Added compatibility aliases in select output (event_date, event_time, location, banner_url)

2.3 Test tooling fixes

* test_all_features.ps1
- Fixed URL parsing bug caused by unescaped &
* Added local verification script: verify_local.ps1
- DB check
- Public endpoint check
- Full API fatal sweep
- Optional authenticated check using TEST_EMAIL/TEST_PASSWORD

3. Verification Evidence

3.1 Database and endpoint checks

* C:/xampp/php/php.exe db_test.php -> Connected to the alumni_portal successfully
* GET /api/get_active_streams.php -> 200
* GET /api/get_jobs.php -> 200
* GET /api/get_events.php -> 200

3.2 Full API sweep snapshot

Saved to: docs/verification_latest.json

Summary:
* Total APIs checked: 77
* Runtime fatal count: 0
* Status distribution without auth token:
- 200: 11
- 401: 62
- 400: 4

This distribution is expected for authenticated endpoints called without bearer tokens.

4. PostgreSQL Auto-Start (Local Fallback)

Because Windows service registration was blocked in non-elevated shell,
a user-level startup fallback was configured:

* Startup script:
C:/Users/shiro/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/start_xampp_postgres.bat
* Behavior:
checks 127.0.0.1:5432 with pg_isready and starts PostgreSQL only when down

Manual start command:

```
C:\xampp\pgsql\pgsql\bin\pg_ctl.exe -D C:\xampp\pgsql\data start -l C:\xampp\pgsql\logfile
```

5. AWS Readiness Review

5.1 Ready

* Terraform modules exist for VPC/ALB/ECS/RDS/S3/CloudFront
* Deploy scripts exist (deployment/deploy-aws.sh, deployment/migrate-to-rds.sh)
* Docker files exist (Dockerfile, docker-compose.yml)
* Environment templates exist (.env.example, .env.production)

5.2 Preflight gaps before deploy

* Ensure tools are available in PATH on deployment shell:
- terraform
- docker
- composer
* Terraform backend S3 bucket must exist or backend config must be updated:
- alumni-portal-terraform-state
* Confirm production domain and ACM certificate before enabling HTTPS listener.

6. Local-to-AWS Working Strategy

1. Keep local testing on XAMPP + PostgreSQL fallback.
2. Run local verification:
powershell -ExecutionPolicy Bypass -File .\verify_local.ps1
3. Run auth verification by setting TEST_EMAIL and TEST_PASSWORD env vars.
4. Deploy with Terraform and deployment scripts from AWS-capable shell.

7. Final Assessment

The project is now stable for local testing with PostgreSQL fallback,
and the previously crashing APIs were fixed. The repository is structurally
AWS-ready, with deployment depending on machine/tooling prerequisites and
final cloud configuration (backend bucket, domain, SSL).