# Enterprise Console Development Environment Setup

This document defines the authoritative workstation provisioning process for the Enterprise Console project.

## 1. Hardware & OS Preparation

1. Windows 10/11 64-bit updated.
2. Enable virtualization (for WSL2/Docker future use).
3. Admin-capable local user.

## 2. Core System Prerequisites

1. Install Microsoft Visual C++ Redistributables (latest x64 bundle).
2. Install Git for Windows (use credential manager, checkout CRLF/commit LF).
3. (Optional) Windows Terminal from Store.
4. Configure global git identity:

* `git config --global user.name "YOUR NAME"`
* `git config --global user.email you@example.com`
* `git config --global init.defaultBranch main`

## 3. PHP & Stack (Laragon)

1. Install Laragon (choose PHP 8.3 build aligning with composer platform).
2. Ensure extensions: openssl, mbstring, tokenizer, xml, ctype, json, fileinfo, curl, pdo\_sqlite (and pdo\_mysql if needed), opcache (optional).
3. Confirm `php -v` in new PowerShell session.

## 4. Node.js Toolchain

1. Install Node.js LTS (18.x or 20.x) from nodejs.org.
2. Verify `node -v`, `npm -v`.

## 5. Global CLI Tools

1. Composer (bundled with Laragon) `composer -V`.
2. Laravel Installer (optional): `composer global require laravel/installer`.
3. GitHub CLI: `winget install --id GitHub.cli`.
4. (Optional) ncu: `npm install -g npm-check-updates`.

## 6. VS Code Installation

1. Install VS Code 64-bit.
2. Enable Settings Sync (GitHub/Microsoft sign-in).
3. Key settings (User Settings JSON):

```

{

"editor.formatOnSave": true,

"files.eol": "\n",

"php.validate.executablePath": "C:\\laragon\\bin\\php\\php-8.3.x\\php.exe",

"terminal.integrated.defaultProfile.windows": "PowerShell"

}

```

1. (Optional) Set default formatter for JS/TS when adding ESLint/Prettier.

## 7. VS Code Extensions

1. PHP Intelephense (bmewburn.vscode-intelephense-client)
2. Laravel Artisan (ryannaddy.laravel-artisan)
3. Laravel Blade Formatter (shufo.vscode-blade-formatter)
4. Blade Snippets (onecentlin.laravel-blade)
5. Laravel Extra Intellisense (amiralizadeh9480.laravel-extra-intellisense)
6. Laravel Pint (open-sourcing.laravel-pint)
7. Tailwind CSS IntelliSense (bradlc.vscode-tailwindcss)
8. PostCSS Language Support (csstools.postcss)
9. GitLens (eamodio.gitlens)

10) GitHub PR (GitHub.vscode-pull-request-github)

11) DotENV (mikestead.dotenv)

12) Error Lens (usernamehw.errorlens)

13) Path Intellisense (christian-kohler.path-intellisense)

14) Markdown All in One (yzhang.markdown-all-in-one)

15) Optional: PHP Namespace Resolver, Code Spell Checker, Indent-Rainbow

## 8. Clone & Project Bootstrap

1. `cd C:\laragon\www`
2. `git clone <repo-url> enterprise-console`
3. `cd enterprise-console`
4. `copy .env.example .env`
5. `composer install`
6. `npm install`
7. `php artisan key:generate`
8. Ensure SQLite file: `if not exist database\database.sqlite type nul > database\database.sqlite`
9. Migrate (if needed): `php artisan migrate`

10) (Optional) `php artisan db:seed`

11) Symlink storage: `php artisan storage:link`

## 9. Development Scripts

1. Start concurrent dev: `composer run dev` (serve, queue:listen, pail, vite).
2. Quality full: `composer run check-all` (pint + phpstan + tests).
3. Static analysis: `composer run stan`.
4. Tests: `php artisan test` (Pest + PHPUnit).
5. Build assets: `npm run build`.

## 10. Frontend Pipeline

1. Tailwind config: `tailwind.config.js`.
2. Vite dev server: `npm run dev`.
3. Production build: `npm run build` -> outputs to `public/build`.
4. Alpine.js + axios loaded via `resources/js` entrypoints.

## 11. Auth & Security Components

1. Sanctum for SPA/token auth (set SANCTUM\_STATEFUL\_DOMAINS if SPA).
2. Socialite providers in `config/services.php` (add keys to `.env`).
3. Spatie Permission: publish/migrate if not yet: `php artisan vendor:publish --provider="Spatie\Permission\PermissionServiceProvider"`.

## 12. Observability

1. Pail for real-time logs (already in dev script).
2. Telescope (local only): `php artisan telescope:install && php artisan migrate`.
3. Restrict Telescope via gate or environment.

## 13. Quality Automation

1. Code style: `vendor\bin\pint` or `composer run lint`.
2. Pre-commit hook example (`.git/hooks/pre-commit`):

```

#!/usr/bin/env bash

composer run check-all || exit 1

```

1. Incrementally raise phpstan level (currently 6).

## 14. Environment Variables (Core)

1. APP\_KEY (generated)
2. APP\_ENV=local, APP\_DEBUG=true
3. DB\_CONNECTION=sqlite (or mysql credentials)
4. QUEUE\_CONNECTION=database (or redis)
5. CACHE\_DRIVER=file (or redis)
6. MAIL\_\*, SANCTUM\_\*, SESSION\_\*

## 15. Optional: Sail / Docker

1. `php artisan sail:install` (select services) then `./vendor/bin/sail up`.
2. Remove Laragon overlap if adopting containers.

## 16. Verification Checklist

1. `php artisan about` OK
2. `composer run check-all` passes
3. `npm run build` succeeds
4. App loads at http://localhost:8000 or Laragon vhost
5. Queue processing works
6. Permissions tables present

## 17. Maintenance

1. Monthly: `composer outdated`, `npm outdated`.
2. Run pint before PR.
3. Add CI workflow (future) replicating dev scripts.

## 18. Regenerating Documentation

1. Update this file as needed.
2. Convert to DOCX using Python helper (see below).

## 19. Matrix Summary

| Category | Item | Purpose | Install | Verify | Notes |

|----------|------|---------|---------|--------|-------|

| OS | Windows 10/11 | Host | Standard | winver | Keep updated |

| Version Control | Git | SCM | winget | git --version | CRLF on checkout |

| Shell | Windows Terminal | Improved CLI | Store | wt -v | Optional |

| PHP | PHP 8.3 | Runtime | Laragon | php -v | Matches platform |

| Web Stack | Laragon | Env mgr | Installer | (tray) | Apache/MySQL |

| Composer | Composer 2 | PHP deps | Bundled | composer -V | Global vendor/bin |

| Node | Node LTS | Frontend | nodejs.org | node -v | Vite/Tailwind |

| Global | Laravel Installer | Scaffolding | composer global | laravel -V | Optional |

| Global | GitHub CLI | PR mgmt | winget | gh --version | Auth login |

| PHP Dep | laravel/framework | Core | composer install | artisan --version | ^12 |

| PHP Dep | sanctum | Auth | composer install | about | SPA tokens |

| PHP Dep | socialite | OAuth | composer install | about | Configure providers |

| PHP Dep | spatie/permission | RBAC | composer install | migrate | Publish config |

| Dev Dep | larastan | Static analysis | composer install | composer run stan | level 6 |

| Dev Dep | pint | Style | composer install | vendor/bin/pint -V | formatOnSave |

| Dev Dep | pest/phpunit | Tests | composer install | artisan test | Unified |

| Dev Dep | telescope | Debug | composer install | artisan telescope:install | Local only |

| Node Dep | vite | Bundler | npm install | npx vite -v | Dev server |

| Node Dep | tailwindcss | CSS | npm install | npx tailwindcss -v | Utility classes |

| Node Dep | alpinejs | JS micro | npm install | (bundle) | Lightweight |

| Node Dep | axios | HTTP | npm install | (bundle) | API calls |

| IDE | VS Code | Editor | Installer | code -v | Sync settings |

| Ext | Intelephense | PHP LS | code --install-extension | (intellisense) | |

| Ext | Blade tools | Templates | code --install-extension | (format) | Formatter + snippets |

| Ext | Tailwind IntelliSense | Classes | code --install-extension | (intellisense) | |

| Ext | GitLens | Git insight | code --install-extension | (sidebar) | |

| Ext | GitHub PR | PR mgmt | code --install-extension | (panel) | |

| Ext | Pint | Style | code --install-extension | (command) | Optional |

| Ext | Error Lens | Inline issues | code --install-extension | (render) | |

| Tooling | Dev script | Concurrency | composer run dev | (processes) | server/queue/logs/vite |

| QA | check-all | CI parity | composer run check-all | (all pass) | pint+phpstan+tests |

## 20. DOCX Generation Helper

A Python script `scripts/md\_to\_docx.py` exists but targets `docs/modules.md`. For this file either:

1. Temporarily copy to `docs/modules.md` then run script; or
2. Duplicate script to target this file.

A dedicated script will be added to generate a DOCX for this environment spec.