x_make_cli_scaffolder_x â€" Command-Deck Bootstrap Forge

This forge fabricates a disciplined CLI project skeleton in seconds. Point it at a workspace, feed it the mission brief, and it welds together a typed package, console entrypoint, and pytest harness so downstream tooling launches without guesswork.

Mission Log

- Generate a ready-to-build CLI repository with `pyproject.toml`, console script binding, and starter documentation.
- Harden the default layout with type-hinted modules, pytest scaffolding, and an opt-in MIT license stub.
- Enforce consistent naming by normalising package and script identifiers before writing to disk.
- Return a structured summary of created artefacts so orchestrators and audits can log evidence immediately.

Instrumentation

- Python 3.11 or newer.
- Optional QA net: Ruff, Black, MyPy, Pyright, pytest.

Operating Procedure

1. `python -m venv .venv`
2. `\.venv\Scripts\Activate.ps1`
3. `python -m pip install --upgrade pip`
4. `pip install -r requirements.txt` *(if you add extras to the template)*
5. `python -m x_make_cli_scaffolder_x --project-name demo_cli --target-dir C:\\workspace\\sandboxes`

The command writes the scaffold under the chosen target directory. Inspect the generated README, CLI module, and tests before committing.

Evidence Checks

Check	Command
Formatting sweep	`python -m black .`
Lint interrogation	`python -m ruff check .`
Type audit	`python -m mypy .`
Static contract scan	`python -m pyright`
Functional verification	`pytest`

Reconstitution Drill

During the monthly rebuild I target a scratch workspace, run the scaffolder twiceâ€"once with tests, once withoutâ€"and confirm both runs emit unique packages with clean QA sweeps. Any drift in template contents is reconciled in Change Control before release.

Conduct Code

Do not ship template adjustments without updating tests and documentation. Every change must prove its value in Change Control with recorded QA output.

Sole Architect's Note

The Forge carries my signature from slug normalisation to README copy. It folds seamlessly into the wider LegatusXY release cadence and keeps new modules disciplined from minute one.