Yes. consolidated solution that eliminates this redundancy entirely

Delete the shim file entirely: `scenario_apply_freeform_paste_clipboard_run.ps1`

Use the main scenario directly: `apply_freeform_paste`

There is a related issue globally about the scenario functionality. I originally designed where there should be global scenarios and project specific scenarios. I believe PF should hold all scenarios and their scripts for a single source of truth.

I audited the scenarios based on the latest "PromptForge V2.2 Scenario Documentation", and found that the functionality is the same whether run by PF, or against the target project, with minor exceptions where running a scenario against itself, PF, may be self destructive, most notably scenario_venv_validate.

If indeed that's the case, with a bit of project awareness, we could simplify the scenario functionality to reside solely on PF, with some exception handling to not self-destruct PF.

Please review the scenarios, and update the PromptForge V2.2 Scenario Documentation, if needed.

And propose deleting or merging redundant code if it still exists.

Scenario	PF	Target Project
scenario_app_selfcheck	N/A. Is the mother of PF standards and shouldn't be able to run	Validates core PromptForge installation and project structure integrity
scenario_apply_freeform_paste_clipboard_run	CRUD and repair of PF functionality. Processes Channel-A JSON or freeform code blocks from clipboard into project files	Processes Channel-A JSON or freeform code blocks from clipboard into project files

scenario_launch_ui	I think G:\My Drive\Code\Python\PromptForge\v2\scripts\scenario_launch_ui.ps1is redundant fn was merged with **scenario_app_selfcheck.** Confirm and delete if appropriate	
scenario_venv_validate	Validates Python virtual environment setup and dependencies may be impossible to do to itself. Self destructive.	Validates Python virtual environment setup and dependencies
scenario_git_publish	Automated git workflow for committing and pushing changes	Automated git workflow for committing and pushing changes
scenario_install_global_runui	System-wide PowerShell installation enabling `runui` command from any project. Allows runui to launch PF.	System-wide PowerShell installation enabling `runui` command from any project. Allows runui to launch Target UI. UI could be a GUI or CLI if GUI doesn't exist. BTW, how will this scenario know what script is the principal UI?.
scenario_install_gui_from_sample	Seems redundant to **scenario_install_global_runui**	

scenario_runui_debug_here	Launch project UI with Python debug flags for troubleshooting	Launch project UI with Python debug flags for troubleshooting
scenario_runui_write_manifest	Purpose: Auto-generates `.pf/runui.json` manifest by detecting project type Note: This scenario is called automatically by `install_global_runui` and typically not run directly by users.	Purpose: Auto-generates `.pf/runui.json` manifest by detecting project type Note: This scenario is called automatically by `install_global_runui` and typically not run directly by users.