Project Brief: PowerShell Workspace

Vision: Build a modern engineering workspace powered by the PowerShell SDK rather than modifying PowerShell itself. The application hosts the PowerShell engine and provides a rich UI including a terminal, command builder, sidebar, file browser, preview pane, object inspector, data visualizers, and plugin system.

Architecture:
UI → PowerShell SDK → Windows/.NET.

Core technologies: C#, .NET, WinUI 3 or WPF, Microsoft.PowerShell.SDK.

Key SDK concepts: - Runspaces: isolated PowerShell sessions (tabs).
- Parser and AST for syntax-aware editing and script generation.
- Pipeline execution with .NET objects instead of plain text.
- Metadata discovery for commands, parameters, modules, and help.
- Built-in completion engine for IntelliSense.

Core features: - Collapsible command sidebar. - Dynamic command builder from PowerShell metadata. - Visual pipeline editor. - Integrated file browser with quick path insertion. - Preview pane for TXT, Markdown, JSON, XML, CSV, HTML, PDF, DOCX, XLSX, PPTX, images, audio, and more. - Object inspector. - Pipeline inspector. - Smart renderers (tables, graphs, waveforms, JSON trees, etc.). - Extensible plugin architecture for previewers, visualizers, and engineering tools.

Guiding principles: 1. Embed PowerShell; don't modify it. 2. Separate execution from presentation. 3. Visualize objects rather than raw text. 4. Build for extensibility. 5. Preserve compatibility with existing PowerShell scripts and modules.

Long-term vision: Create a modern engineering workspace that combines the strengths of PowerShell, an IDE, File Explorer, data viewers, and visualization tools while remaining fully powered by the PowerShell engine.