**DSP Workbench / TestHarness Requirements Document**

**Overview**

The DSP Workbench (TestHarness) will become a permanent, modular part of the Sower application. It will serve as an internal tool for:

* Testing internally developed audio Effects.
* Testing audio Generators (e.g., Sine waves, Square waves, etc.).
* Providing live feedback for tuning Effect parameters.
* Hosting and testing third-party VST effects in a simple environment.
* Supporting development of future UI elements based on real, working audio engines.

**Primary Goals**

* **Simple UI for Generators and Effects** selection and configuration.
* **Real-time audio playback** through selected effects.
* **Volume Control** for overall output.
* **Play/Pause/Stop controls**.
* **Extendable architecture** to add new Generators, Effects, and Plugin hosting easily.
* **Non-destructive testing** environment (won't affect project sessions).

**Core Features**

**Audio Routing**

* Integrated with JUCE's AudioDeviceManager.
* Connects to system-selected audio device (e.g., ASIO4ALL, VoiceMeeter).
* Internally managed via AudioPlayer class.

**Playback Control**

* Play: Start audio playback.
* Pause: Halt playback but maintain position.
* Stop: Halt playback and reset to start.
* Volume: Real-time adjustable gain on output.

**Generator Selection**

* ComboBox to select available Generators.
* Supported Generators (Phase 1):
  + SineGenerator
  + SquareGenerator
  + TriangleGenerator
  + NoiseGenerator (future)
* Dynamic controls based on generator:
  + Frequency (Hz) slider
  + Amplitude slider (optional)

**Effect Selection**

* ComboBox to select an Effect.
* Supported Effects (Phase 1):
  + GainEffect
  + (ReverbEffect, ChorusEffect - future)
* Dynamic controls based on effect:
  + Gain Slider (for GainEffect)
  + Other parameter controls as needed.

**Visual Feedback (Phase 2+)**

* Status label (Playing / Paused / Stopped)
* Later additions:
  + Real-time audio metering (optional)
  + Waveform visualization (optional)

**Architectural Considerations**

* **Keep TestHarness modular**: Easy to move into other applications or modules.
* **Avoid tight coupling** with Main Workbench or Project State.
* **Minimal dependencies** on other app systems.
* **Future-friendly** for Plugin (VST/AU) hosting.
* **Lightweight UI**: No heavy visualizations unless user-enabled.

**Future Enhancements**

* MIDI generator for synth effect testing.
* Automation scripting for parameter sweeps.
* Host 3rd-party VST plugins and chain with internal effects.
* Preset save/load for Generator and Effect configurations.
* Multi-voice or polyphonic generator tests.

**Constraints**

* Should run safely even when no audio device is available.
* Should not affect main project audio engine.
* Should use JUCE components and best practices.

**Document Version**: 0.1 (Draft)

(Prepared in collaboration with the Sower App Development Team 🚀)