**Core Functionality**

1. **Config Generator**
   * Build .cfg files through a GUI (binds, cvars, dropdowns, hotkey inputs).
2. **Import & Explain .cfg Files**
   * Read existing configs, parse them, and explain each line.
3. **Export Clean Config Files**
   * Output autoexec.cfg, practice.cfg, etc. for use in CS2.
4. **Interactive On-Screen Keyboard + Mouse**
   * Click keys/buttons to highlight their location through the on-screen display, and label them according to their in-game key binding name, allowing the user to assign commands visually.
5. **Command Documentation**
   * Built-in explanations of binds, commands, and cvars from Valve’s wiki, with additional fields for community contributions/explanations.
6. **Natural Language Input/Parsing**
   * E.g., “Make Q drop weapon” → bind "q" "drop".

**Enhancement / Quality-of-Life Goals**

1. **Conflict Detection**
   * Warn about keybind overlaps or incompatible cvars.
2. **Live Simulated Preview**
   * Show what pressing key combos would do.
3. **Steam Directory Autofinder**
   * Automatically locate the cfg/ folder.
4. **Preset & Profile Loader**
   * Load configs for s1mple, NiKo, etc. or your own profiles.
5. **Syntax Help & Tooltips**
   * Hover to explain what alias, +attack, or cheat flags do.
6. **Command Version Awareness**
   * Warn if a command is deprecated or CS2-specific.
7. **Advanced Command Search & Filters**
   * Search by type, use, or flags (bind, viewmodel, crosshair, etc.).
8. **Task-Based Wizards**
   * “I want better visibility” → shows related options.

**Long-Term & Optional Goals**

1. **Keybind Visualizer**
   * See entire keyboard layout with current bindings.
2. **Backup & Restore**
   * Snapshot configs, restore previous states.
3. **Localization**
   * Support for non-English speakers.
4. **AI Recommendations**
   * Suggest config improvements based on goals (e.g. low input lag).
5. **Plugin/Macro System**
   * Allow multi-command alias creation.
6. **Cloud Sync**
   * Optionally sync configs with GitHub or local backup folder.

**Folder Structure (src)**

A screenshot of a computer program

AI-generated content may be incorrect.

**User Flow Diagram**

1. User opens app (Tauri)
2. Chooses one of:
   * Import config
   * Build config from scratch
3. If importing:
   * File is parsed
   * Commands are explained
   * User edits, re-exports
4. If creating:
   * Selects keys/commands using GUI
   * Gets real-time preview
   * Exports .cfg file