

Clipboard Snippet Manager

A local-first clipboard snippet manager with searchable history, a pinned categorized snippet library and one-click copy — delivered built-to-order with full source, report, PPT and viva kit.

01. Title

Clipboard Snippet Manager.

02. Introduction / Background

Everyone has a graveyard of useful text: the SSH command for the lab server, the perfect follow-up email, a regex that took an hour to write, the link to the report template. These live scattered across sticky notes, old chats and memory, and vanish exactly when needed. Clipboard managers solve the history half (what did I copy ten minutes ago); snippet libraries solve the reuse half (the things I copy every week). This project builds both in one desktop application: a rolling clipboard history with timestamps plus a permanent, categorized, searchable, pinnable snippet library with one-click copy-back. The data model is deliberately local-first — the most useful snippets are exactly the ones that should never sync to a cloud.

03. Problem Statement

Useful text fragments are scattered and get lost; existing tools either keep only a short unsearchable history or sync everything to a cloud account. Users need a private, organized, searchable home for the snippets they reuse daily.

04. Objectives

- Maintain a pinned snippet library sorted pinned-first with visual markers.
- Organize into Code, Commands, Links, Email and Text categories with counts.
- Provide instant live search over titles and content.
- Keep a rolling clipboard history with timestamps, separate from the library.
- Copy any snippet back to the OS clipboard in one click — all storage local-only.

05. Proposed Methodology

Snippets are modeled as {title, category, content, pinned, timestamp} objects in local storage. The library renders as cards sorted pinned-first; the search box applies live case-insensitive substring filtering over title+content. Copy uses the Clipboard API with an execCommand fallback and shows a confirmation toast. Add/edit share one dialog; pinning toggles a flag that re-sorts the view. Category counts derive from the dataset on every render.

06. System Architecture / Working

Header (branding + global search), category sidebar (filters + counts + explainer), card grid (snippet cards with pin, copy, edit), modal dialog (add/edit form), toast system (copy confirmation). The Electron shell adds native menus and persistence; the web demo runs the same logic in-page.

07. Hardware / Software Requirements

- Desktop/laptop running Windows, macOS or Linux.
- A modern browser engine or the Electron shell; no runtime dependencies.
- No network connection required — local-first by design.

08. Expected Outcomes

- A working library app with instant search, pinning and history demonstrable on realistic snippet data.
- A personal snippet workflow (add → pin → search → copy) completable in seconds.
- Complete source, user manual, report, PPT and viva Q&A.;

09. Applications

- Developers reusing commands, snippets and regexes.
- Writers and students reusing email templates and citations.
- Support staff reusing approved response texts.
- Classroom demonstration of local-first software design.

10. Future Scope

- OS-wide clipboard monitoring via platform native hooks.
- Fuzzy search.
- Encrypted sync across devices.