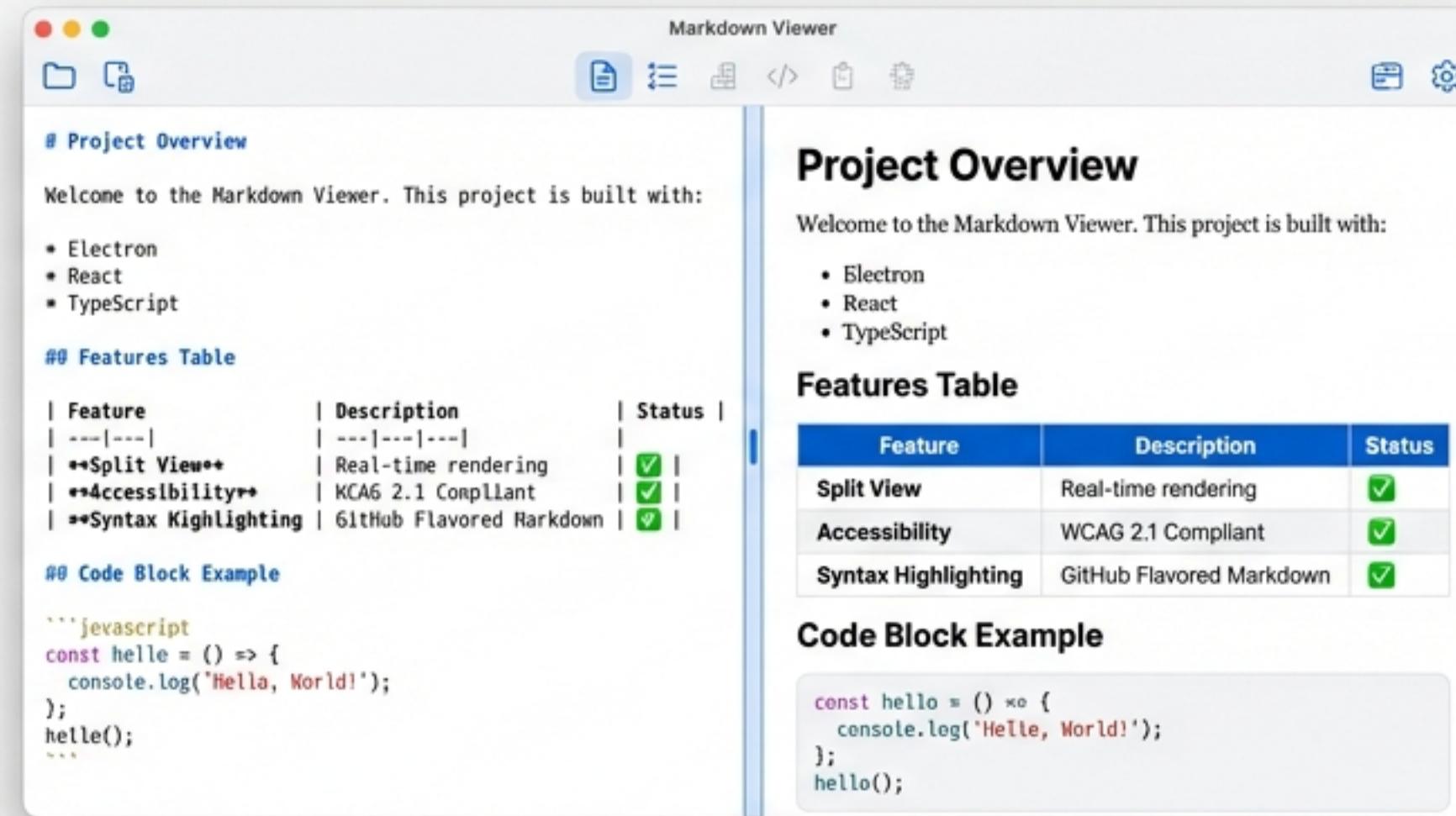


Markdown Viewer: The Evolution of a Modern Tool

A feature-rich, accessible, and performant Markdown Viewer for macOS built with Electron, React, and TypeScript.



Secure & Private

Sandboxed, 100% local, with zero telemetry.



Feature-Rich

Multi-tab editing, triple view modes, and GFM support.



Accessible

Full WCAG 2.1 compliance with complete keyboard navigation.

Day One: From Core Renderer to Multi-Window Powerhouse

November 21, 2025



v1.0.0: The Genesis

Initial release with basic Markdown rendering.

Features:

- Rendered/Raw view toggle
- Syntax highlighting



v1.1.0 - v1.2.0: Building the Cockpit

Toolbar and Status Bar are born.

Features:

- Copy, Theme Toggle, Formatting Buttons, Real-time stats (words, chars, tokens)
- Introduces Multi-Tab Support and Drag-to-Spawn Windows



v1.4.0: Critical Stabilization

Fixes major bugs: event listener accumulation, race conditions, memory leaks.

Features:

- Adds foundational UX: File → Open menu (Cmd+O) and Drag-and-Drop file opening



v2.0.0: A Major Leap

A complete overhaul of tab and window management.

Features:

- Key outcome: Robust tab dragging between windows and smart window closing logic

Solidifying the Foundation: The Migration to TypeScript

The focus shifts from rapid feature addition to creating a stable, maintainable, and type-safe codebase for future development.



Project-Wide Conversion (v2.1.0)

- The entire codebase (16 files) was migrated from JavaScript to TypeScript.
- Achieved strict type checking with zero compilation errors.

Technical Benefits

- Type-Safe IPC:** Implemented discriminated unions for secure, predictable communication between the main and renderer processes.
- Targeted Configuration:** Separate `tsconfig` files for main, preload, and renderer processes ensure precision.
- Enhanced Developer Experience:** Full IDE autocomplete and compile-time error checking.

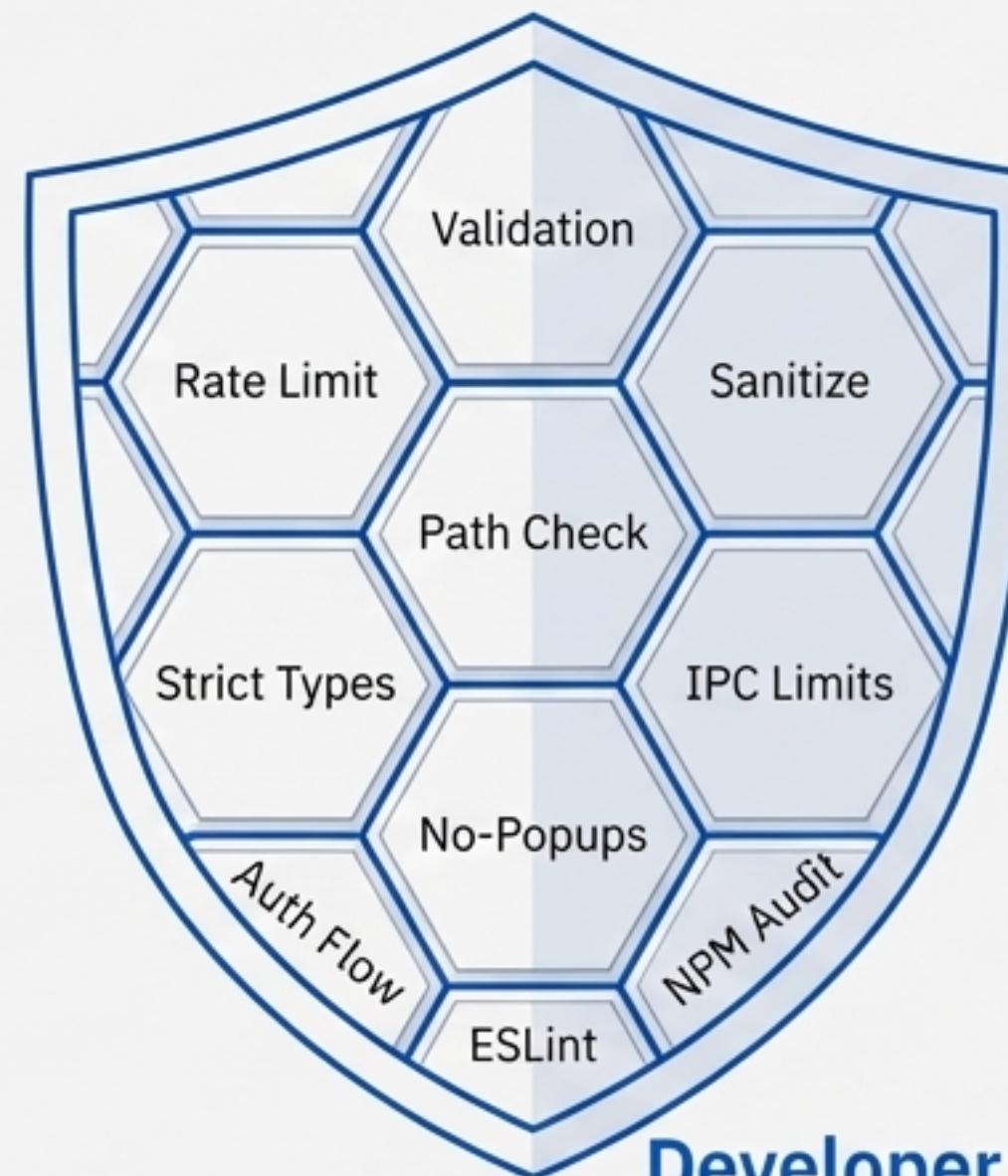
All existing functionality was preserved with zero breaking changes, demonstrating a successful and meticulous architectural upgrade.

Forging the Armor: The Major Security Release (v2.2.0)

A proactive hardening effort to build a secure-by-design application.

Threat Prevention

-  **Path Traversal Protection**
Validates all file paths to prevent unauthorized file system access.
-  **Input Validation**
Implements strict type and size checks on all IPC messages.
-  **Rate Limiting**
Sets a 100 calls/second limit on IPC handlers to prevent Denial-of-Service attacks.



Runtime Protection

- Blocked all external URL navigation and popup window creation.
- Sanitized error messages to prevent information disclosure.

Developer Safeguards

- Integrated ESLint security plugins directly into the development workflow.
- Ran `npm audit fix` to resolve known dependency vulnerabilities.



NPM Audit

Integrated ESLint security plugins directly into the development workflow.



Resource Management

Enforces limits on file size (50MB), IPC content (10MB), and concurrent windows (10) to prevent memory/resource exhaustion.

Expanding the Toolkit: Professional Output and Advanced Editing

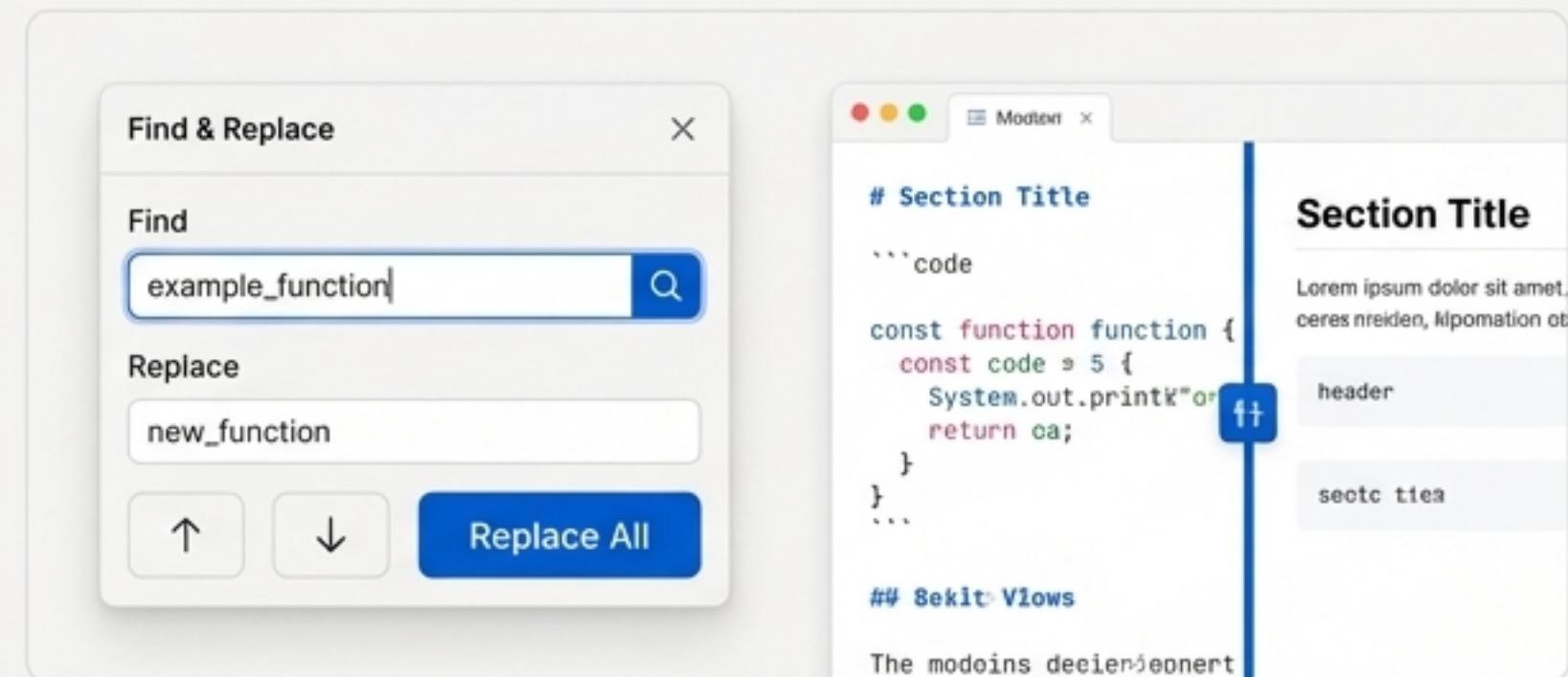
With security hardened, development accelerated to deliver power-user features.

Professional PDF Export (v2.4.0)



- ✓ Export Markdown documents to perfectly formatted PDFs.
- ✓ Preserves all formatting: headers, tables, code blocks.
- ✓ Maintains syntax highlighting for code in the final PDF.
- ✓ Securely implemented with rate limiting and content size validation.

Advanced Editing & Split View (v2.5.0)

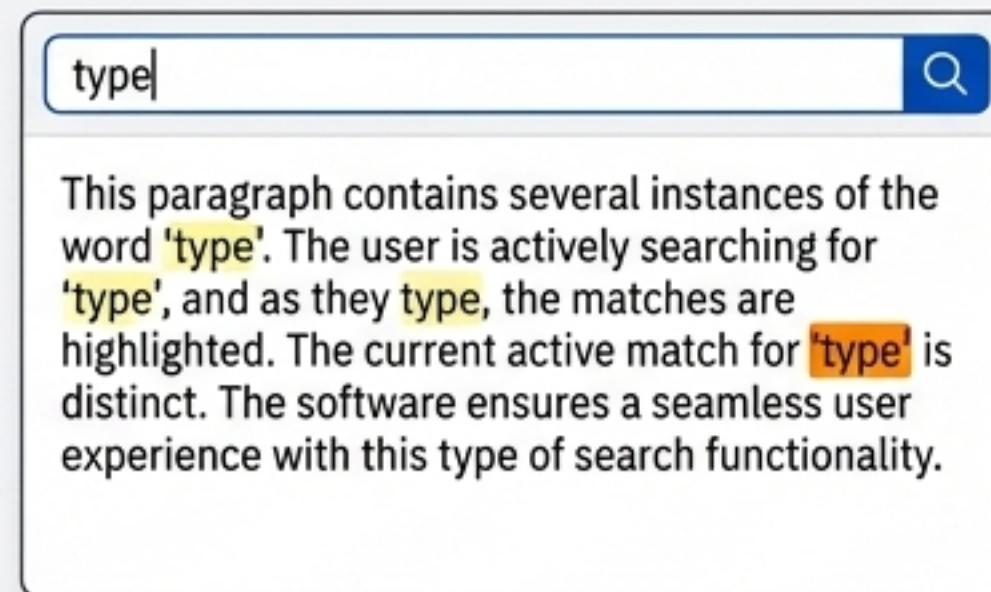


- 🔍 **Find & Replace:** A comprehensive search tool with case-sensitivity, match navigation (↑/↓), and replace-all functionality.
- .SplitContainer Mode: A new side-by-side view with a resizable divider to see the raw source and rendered preview simultaneously.
- 💾 Unified 'Save As': A single 'Cmd+S' command intelligently saves as '.md' or exports to '.pdf' based on the chosen file extension, simplifying the UI.

Polishing the Experience: From Functional to Fluid

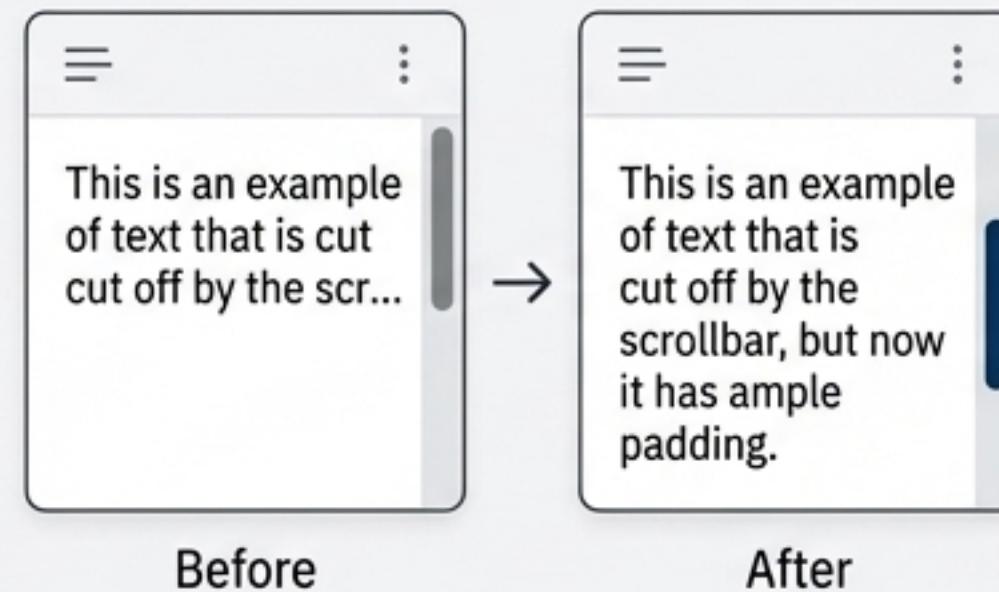
A series of updates focused on fine-tuning the user interface and interaction design.

Real-Time Incremental Search (v2.6.2)



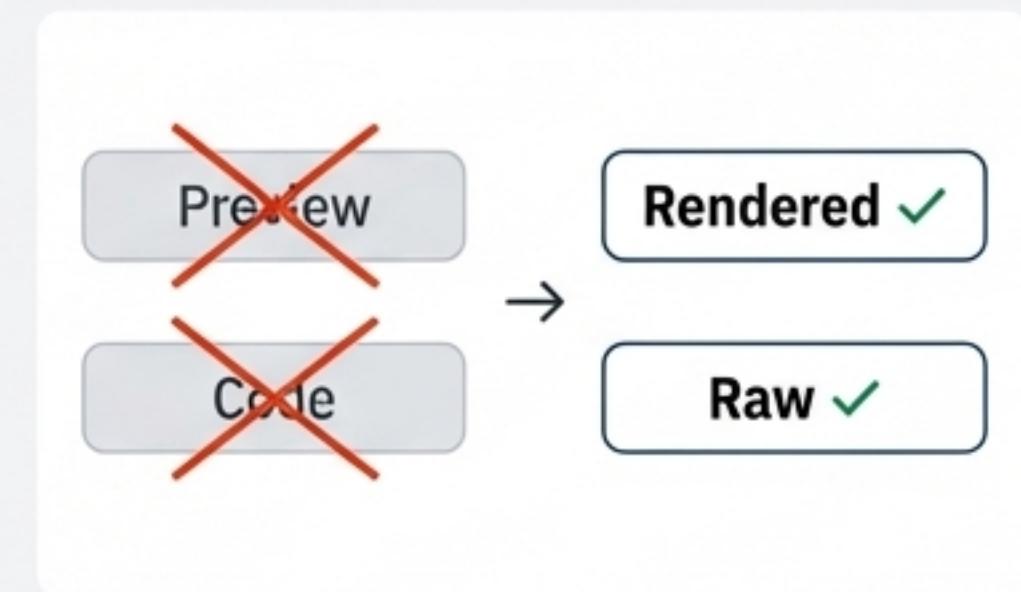
- 🔍 • Find & Replace highlights now update instantly as you type.
- ✍ • Current match is distinctly highlighted (orange) from other matches (yellow).
- 🕒 • Preserves the undo/redo stack after 'Replace All' operations.

Improved Readability & Navigation (v2.6.3)



- ⠇ • Added a visual scroll position indicator bar to the Raw view.
- [T] • Increased text padding to prevent cutoff by the scrollbar.
- ⠇ • Fixed horizontal overflow issues, ensuring long URLs and content wrap correctly.

Clarity in Language (v2.6.4)

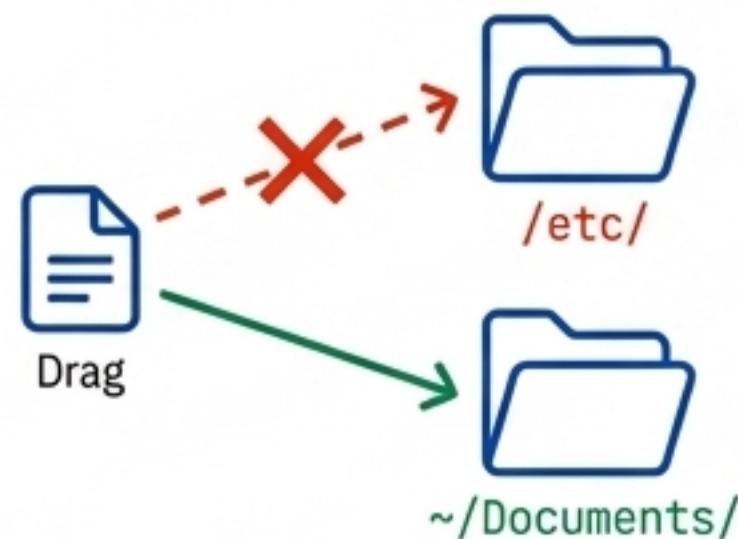


- ✓ • Renamed view modes for better user understanding:
 - 'Preview' → '**Rendered**' (processed output)
 - 'Code' → '**Raw**' (unprocessed source)
- Updated all corresponding labels, tooltips, and documentation.

Relentless Fortification: Neutralizing Critical Threats

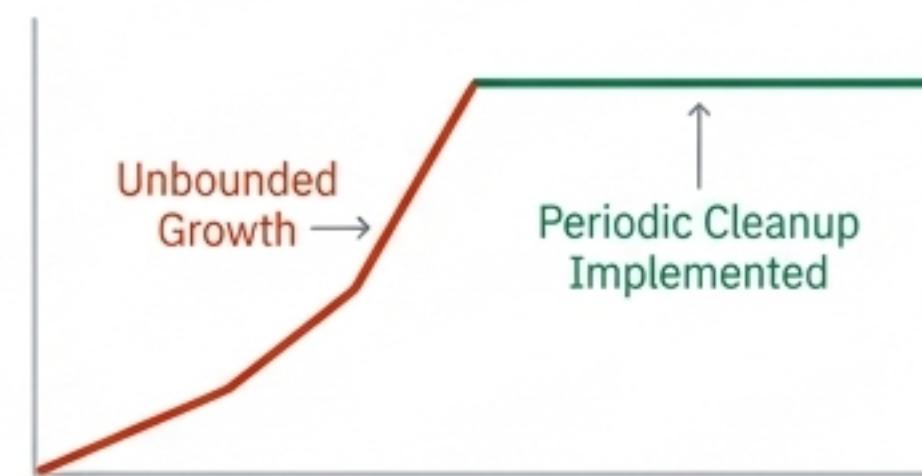
An immediate response to newly identified, high-impact security vulnerabilities.

Path Traversal in Drag-and-Drop (v2.6.4)



Fixed a critical vulnerability in file handling. Now enforces strict path validation and file extension checks (.md/.markdown) on all dropped files.

Rate Limiter Memory Leak (v2.6.7)

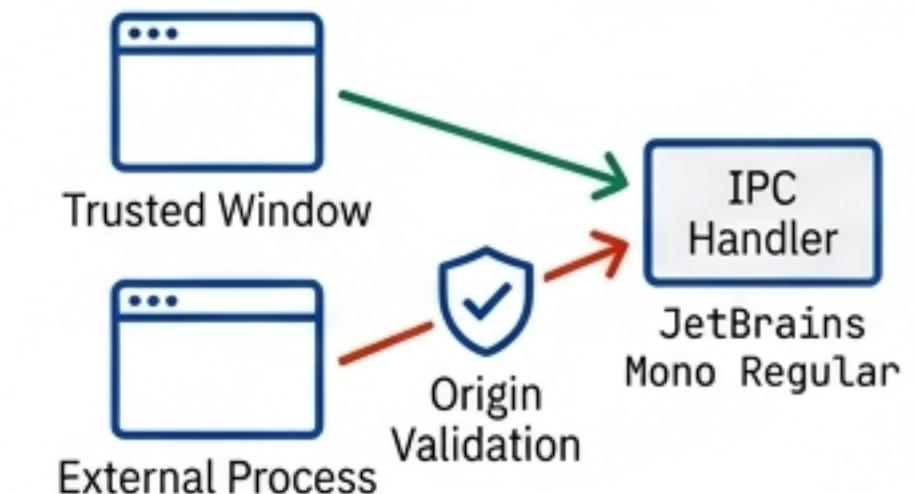


Vulnerability: CRITICAL-4, CVSS Score 7.5.

Fix: Implemented a periodic cleanup mechanism to remove stale entries, preventing unbounded memory growth and potential memory exhaustion.

Result: Risk reduced to 0.0.

IPC Origin Validation (v2.6.7)



Vulnerability: CRITICAL-5, CVSS Score 6.5.

Fix: All 8 IPC handlers now validate that the sender is a known, trusted browser window, preventing unauthorized calls from external processes.

Result: Risk reduced to 0.0.

The Final Polish: The 'Copy Code' Button

A small, highly-requested feature that perfects the developer workflow (v2.7.0).

The screenshot shows a code editor window with Python code for connecting to a database. In the bottom right corner, there is a semi-transparent button labeled 'Copy Code'. The button's state changes from a blue square with a white icon in the default state, to a darker blue square with a white hand cursor icon in the hover state, and finally to a dark blue square with a green checkmark icon in the clicked state. A tooltip 'Copied!' appears above the button in the clicked state.

```
def connect_to_database(self):
    """Establishes a connection to the database."""
    try:
        self.connection = psycopg2.connect(
            host=self.host,
            database=self.database,
            user=self.user,
            password=self.password
        )
        return True
    except Exception as e:
        print(f"Error connecting to database: {e}")
        return False
```

****Default State****

****Hover State****

****Clicked State****



Intuitive Design

Appears on hover in the bottom-right corner of all code blocks.



Theme-Aware

Semi-transparent design works perfectly across all 5 themes (Light, Dark, System, Solarized Light, Solarized Dark).



Clear Feedback

Provides an instant visual checkmark upon a successful copy action.



Secure Implementation

Uses the standard, secure Clipboard API without any complex dependencies.

The Markdown Viewer Today: A Comprehensive Feature Set



Multi-Document Management

Multi-Tab Support, Drag-to-Spawn Windows, Smart File Handling.



Viewing & Editing

Triple View Modes (Rendered, Raw, Split), Live Preview, GFM Support, Syntax Highlighting.



Text Formatting

Quick Formatting Buttons, Powerful Find & Replace, Rich Text Copy, Unified Save As (MD/PDF).



Security

Sandboxed Renderer, Context Isolation, XSS Protection (rehype-sanitize), Strict CSP, Electron Fuses, Path Traversal Protection.



Accessibility

Full ARIA Support, Complete Keyboard Navigation, Screen Reader Optimized, Enhanced Focus Indicators.



Performance

Memoized Components, Smart Re-renders, Efficient Calculations, Fast Startup.

The Road Ahead: The Future of the Viewer

The current focus is on continued security hardening, followed by major enhancements to the editor and professional feature set.

