

Functions Menu Guidelines

CONFIDENTIAL INFORMATION
Prohibited Distribution

WRITAB Docs Guidelines

Introduction

This document provides comprehensive guidelines for WRITAB Docs, the word processing component of the WRITAB web app (version 1). WRITAB Docs is built using the specified tech stack, including React for the frontend, Lexical for rich text editing, Node.js and Fastify for backend APIs, Supabase for database storage (using PostgreSQL with JSONB for document data), and Redis for real-time collaboration. All documents are saved as *.wrtb files, which can contain multiple tabs from Docs, Sheets, and Code apps. Tabs represent individual "files" within a *.wrtb container, and WRITAB Drive handles cloud storage for *.wrtb files, attachments, and other user files.

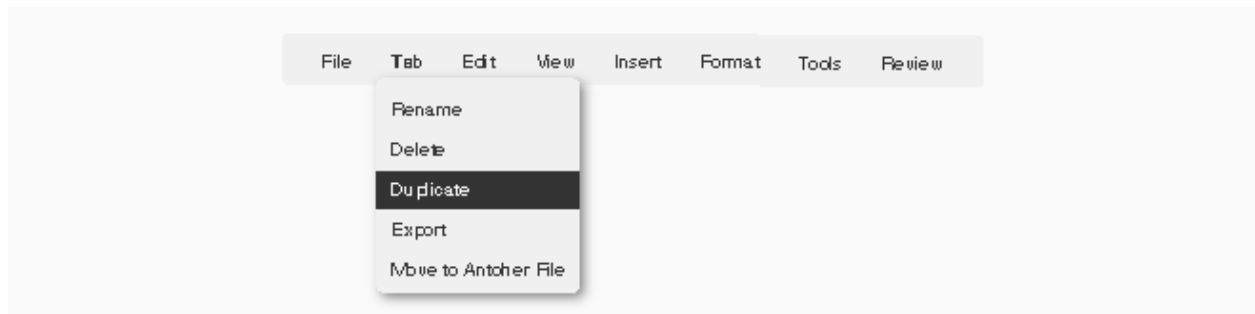
Guidelines are divided into two sections:

- **For Developers:** Technical implementation details, referencing the tech stack.
- **For Users:** Usage instructions.

The menu structure has been standardized to align with common word processors like Google Docs and Microsoft Word. Missing functions (e.g., Edit menu, Print, Word Count) have been added for completeness. Redundant items (e.g., overlapping "Show Comments" in View and Review) have been consolidated under Review for better organization. Unnecessary or repetitive sub-functions (e.g., separate Minimize/Expand Comments) have been streamlined into toggle-based options.

Menu Structure Overview

- **File:** Handles file-level operations for *.wrtb files.
- **Tab:** Tab-specific operations within a *.wrtb file.
- **Edit:** Standard editing actions.
- **View:** Controls display and modes.
- **Insert:** Adds content elements.
- **Format:** Styles text and layout.
- **Tools:** Utility functions.
- **Review:** Collaboration and change tracking.



Detailed Guidelines

File Menu

New

- **For Developers:** Creates a new *.wrtb file with an initial Docs tab. Use Redux Toolkit to initialize state, Lexical for the editor instance, and Supabase to insert a new JSONB record for the file metadata and tab content. Trigger a WebSocket (Socket.IO) event for real-time sync if collaborative.
- **For Users:** Creates a new blank document as a *.wrtb file with one Docs tab. Shortcut: Ctrl+N.

Open

- **For Developers:** Loads an existing *.wrtb file from WRITAB Drive. Query Supabase for file metadata, deserialize JSONB tab data into Lexical editor. Use Axios for API calls to Fastify backend, with Redis caching for recent files.
- **For Users:** Opens a *.wrtb file from WRITAB Drive or local upload. Displays tabs in a tabbed UI.

Save

- **For Developers:** Autosaves changes to the current *.wrtb file via Supabase upsert. Use debounced Redux actions to serialize Lexical state to JSONB. (Added for standard behavior, even with autosave.)
- **For Users:** Manually saves the current *.wrtb file. Changes are autosaved every 30 seconds.

Save As

- **For Developers:** Duplicates the current *.wrtb file with a new name in Supabase. Copy JSONB data and update metadata.
- **For Users:** Saves a copy of the current *.wrtb file with a new name in WRITAB Drive.

Print

- **For Developers:** Generates a printable view of the current tab. Use browser print API or export to PDF via backend (Node.js with a library like pdfkit if integrated; otherwise, client-side with window.print()). (Added for standardization.)
- **For Users:** Prints the current Docs tab or exports to PDF for printing.

Share

- **For Developers:** Generates a shareable link or invites users via Auth0 for authentication and Supabase row-level security. Use WebSocket for real-time collaborator updates. (Added for collaboration standard.)
- **For Users:** Shares the *.wrth file with view/edit permissions via link or email.

Close

- **For Developers:** Unloads the current *.wrth file from state, clearing Lexical editor. Ensure autosave via Redux before closing. (Added for standard file management.)
- **For Users:** Closes the current *.wrth file and returns to WRITAB Drive.
-

Tab Menu

Rename

- **For Developers:** Updates tab metadata in the *.wrth JSONB structure via Supabase. (Added for usability.)
- **For Users:** Renames the current Docs tab.

Delete

- **For Developers:** Removes the tab from the *.wrth file's JSONB array in Supabase.
- **For Users:** Deletes the current tab.

Duplicate

- **For Developers:** Clones the tab's Lexical state and inserts as a new tab in Supabase.
- **For Users:** Creates a copy of the current tab within the same *.wrth file.

Export

- **For Developers:** Converts Lexical state to formats like DOCX/PDF/HTML via backend (Node.js libraries like mammoth for DOCX). Upload to S3 if needed.
- **For Users:** Exports the tab to DOCX, PDF, TXT, or HTML.

Move to Another File

- **For Developers:** Transfers tab JSONB data between *.wrth files in Supabase. (Added for multi-tab management.)
- **For Users:** Moves the tab to a different *.wrth file in WRITAB Drive.

Edit Menu

(Added entirely for standardization, as it's essential in word processors.)

Undo

- **For Developers:** Leverages Lexical's built-in undo stack. Integrate with Redux for global state.
- **For Users:** Reverts the last action. Shortcut: Ctrl+Z.

Redo

- **For Developers:** Uses Lexical's redo stack, synced with Redux.
- **For Users:** Reapplies the last undone action. Shortcut: Ctrl+Y.

Cut

- **For Developers:** Standard browser execCommand or Lexical API for cutting selected content.
- **For Users:** Cuts selected text/image. Shortcut: Ctrl+X.

Copy

- **For Developers:** Lexical or browser copy command.
- **For Users:** Copies selected content. Shortcut: Ctrl+C.

Paste

- **For Developers:** Handles paste events in Lexical, sanitizing content.
- **For Users:** Pastes clipboard content. Shortcut: Ctrl+V.

Find

- **For Developers:** Implements search via Lexical's text traversal or browser find.
- **For Users:** Searches for text in the document. Shortcut: Ctrl+F.

Replace

- **For Developers:** Extends Find with replace logic, updating Lexical nodes.
- **For Users:** Finds and replaces text.

Select All

- **For Developers:** Selects entire document via Lexical API.
- **For Users:** Selects all content. Shortcut: Ctrl+A.
-

View Menu

Edit Mode

- **For Developers:** Sets Lexical to full edit mode, disabling suggestions.
- **For Users:** Enables direct editing.

Suggesting Mode

- **For Developers:** Activates Lexical's suggestion plugin for tracked changes.
- **For Users:** Edits suggest changes without altering original text.

Reading Mode

- **For Developers:** Renders a clean, non-editable view with Tailwind CSS for presentation. (Renamed from Presentation Reading Mode for clarity.)
- **For Users:** Displays document in read-only mode for focused reading.

Zoom In/Out

- **For Developers:** Adjusts CSS zoom or transform on the editor container. (Added for standard view controls.)
- **For Users:** Zooms the document view.

Full Screen

- **For Developers:** Uses browser fullscreen API.
- **For Users:** Expands to full screen.

Show Ruler

- **For Developers:** Toggles a custom React component for ruler display.
- **For Users:** Shows/hides page ruler.

Show Gridlines

- **For Developers:** Overlays CSS grid on the editor.
- **For Users:** Shows/hides gridlines for alignment.

Comments

- **Sub-functions:** Show/Hide Comments (toggles visibility; consolidated from original subs).
- **For Developers:** Manages comment threads via Lexical plugins, stored in Supabase JSONB.
- **For Users:** Toggles comment display.

Insert Menu

Image

- **For Developers:** Uploads to S3 via Amplify, inserts into Lexical as an image node.
- **For Users:** Inserts an image from WRITAB Drive or upload.

Link

- **For Developers:** Uses Lexical's link plugin. (Renamed from Hyperlink for standard terminology.)
- **For Users:** Adds a hyperlink to text.

Table

- **For Developers:** Inserts a table node in Lexical, possibly integrating Handsontable for advanced tables.
- **For Users:** Inserts a table.

Chart

- **For Developers:** Links to WRITAB Sheets tab or embeds simple chart via Matplotlib if backend-generated. (Added for integration.)
- **For Users:** Inserts a chart from data.

Special Characters

- **For Developers:** Opens a symbol picker, inserts via Lexical. (Renamed from Symbols.)
- **For Users:** Inserts symbols/emojis.

Horizontal Line

- **For Developers:** Inserts HR node in Lexical.
- **For Users:** Adds a horizontal rule.

Page Break

- **For Developers:** Inserts break node. (Renamed from Break.)
- **For Users:** Inserts a page break.

Header & Footer

- **For Developers:** Custom Lexical nodes for headers/footers. (Added for standard.)
- **For Users:** Adds header or footer.

Footnote

- **For Developers:** Lexical footnote plugin.
- **For Users:** Inserts a footnote.

Bookmark

- **For Developers:** Anchor node in Lexical.
- **For Users:** Adds an internal bookmark.

Equation

- **For Developers:** Integrates MathJax or similar for rendering. (Added for advanced docs.)
- **For Users:** Inserts math equations.

Format Menu

Text

- **Sub-functions:** Bold, Italic, Underline, Strikethrough, Font, Size, Color, Highlight.
- **For Developers:** Applies Lexical formatting commands.
- **For Users:** Formats selected text.

Paragraph Styles

- **Sub-functions:** Normal, Heading 1-6, Title, Subtitle.
- **For Users:** Applies predefined styles.

Alignment

- **Sub-functions:** Left, Center, Right, Justify. (Renamed from Align.)
- **For Developers:** CSS classes via Tailwind.
- **For Users:** Aligns paragraphs.

Line & Paragraph Spacing

- **For Developers:** Updates Lexical node styles.
- **For Users:** Adjusts spacing.

Lists

- **Sub-functions:** Bulleted, Numbered. (Renamed from Bullets.)
- **For Developers:** Lexical list plugins.
- **For Users:** Creates lists.

Clear Formatting

- **For Developers:** Resets Lexical node styles. (Added for standard.)

- **For Users:** Removes all formatting.

Columns

- **For Developers:** Applies multi-column CSS layout. (Added for layout options.)
- **For Users:** Sets document to multiple columns.

Tools Menu

Spelling and Grammar

- **For Developers:** Integrates a library like LanguageTool via backend API.
- **For Users:** Checks and suggests corrections.

Auto-Correct Options

- **For Developers:** Configures Lexical auto-correct rules.
- **For Users:** Customizes auto-corrections.

Set Proofing Language

- **For Developers:** Sets document language metadata in Supabase.
- **For Users:** Changes proofing language.

Translate Document

- **For Developers:** Calls external API (e.g., via gRPC) for translation, updates Lexical content.
- **For Users:** Translates the entire document.

Word Count

- **For Developers:** Traverses Lexical nodes to count words. (Added for standard.)
- **For Users:** Displays word/character count.

Voice Typing

- **For Developers:** Uses Web Speech API in browser. (Added if feasible with tech stack.)
- **For Users:** Dictates text via microphone.

Review Menu

Track Changes

- **For Developers:** Enables Lexical's change-tracking mode.
- **For Users:** Tracks all edits.

Accept Changes

- **For Developers:** Applies tracked changes to base content.
- **For Users:** Accepts suggestions.

Reject Changes

- **For Developers:** Discards tracked changes.
- **For Users:** Rejects suggestions.

Review Suggested Edits

- **For Developers:** Navigates through suggestions via Lexical.
- **For Users:** Reviews all suggestions.

Add Comment

- **For Developers:** Inserts comment node linked to selection.
- **For Users:** Adds a comment to selected text.

Edit Comment

- **For Developers:** Updates existing comment in JSONB. (Added for completeness.)
- **For Users:** Edits an existing comment.

Delete Comment

- **For Developers:** Removes comment node.
- **For Users:** Deletes a comment.

Show All Comments

- **For Developers:** Toggles sidebar with all comments (consolidated).
- **For Users:** Displays all comments in a panel.

Hide Comments

- **For Developers:** Hides comment indicators.
- **For Users:** Hides comments.

Resolve All Comments

- **For Developers:** Marks all as resolved in Supabase. (Added for standard review flow.)
- **For Users:** Resolves all open comments.

WRITAB Sheets Guidelines

Introduction

This document provides comprehensive guidelines for WRITAB Sheets, the spreadsheet component of the WRITAB web app (version 1). WRITAB Sheets is built using the specified tech stack, including React for the frontend, Handsontable for spreadsheet functionality, Node.js and Fastify for backend APIs, Supabase for database storage (using PostgreSQL with JSONB for sheet data), and Redis for real-time collaboration. All documents are saved as *.wrwb files, which can contain multiple tabs from Docs, Sheets, and Code apps. Tabs represent individual "files" within a *.wrwb container, and WRITAB Drive handles cloud storage for *.wrwb files, attachments, and other user files.

Guidelines are divided into two sections:

- **For Developers:** Technical implementation details, referencing the tech stack.
- **For Users:** Usage instructions.

The menu structure has been standardized to align with common spreadsheets like Google Sheets and Microsoft Excel. Missing functions (e.g., Edit menu, Print, Chart insertion, Formula menu) have been added for completeness. Redundant items (e.g., overlapping Filter options) have been consolidated under Data for better organization. Unnecessary or repetitive sub-functions (e.g., specific Freeze counts limited to dynamic selection) have been streamlined into more flexible options.

Menu Structure Overview

- **File:** Handles file-level operations for *.wrtb files.
- **Tab:** Tab-specific operations within a *.wrtb file.
- **Edit:** Standard editing actions.
- **View:** Controls display and layout.
- **Insert:** Adds content elements.
- **Format:** Styles cells and layout.
- **Formula:** Manages functions and calculations (added as a separate menu for standardization).
- **Data:** Data manipulation tools.
- **Tools:** Utility functions.
- **Review:** Collaboration and protection (added for collaboration standard).
- **Help:** Support resources.



Detailed Guidelines

File Menu

New

- **For Developers:** Creates a new *.wrtb file with an initial Sheets tab. Use Redux Toolkit to initialize state, Handsontable for the grid instance, and Supabase to insert a new JSONB record for file metadata and sheet data. Trigger a WebSocket (Socket.IO) event for real-time sync if collaborative.
- **For Users:** Creates a new blank spreadsheet as a *.wrtb file with one Sheets tab. Shortcut: Ctrl+N.

Open

- **For Developers:** Loads an existing *.wrtb file from WRITAB Drive. Query Supabase for file metadata, deserialize JSONB sheet data into Handsontable. Use Axios for API calls to Fastify backend, with Redis caching for recent files.
- **For Users:** Opens a *.wrtb file from WRITAB Drive or local upload. Displays tabs in a tabbed UI.

Save

- **For Developers:** Autosaves changes to the current *.wrtb file via Supabase upsert. Use debounced Redux actions to serialize Handsontable state to JSONB. (Added for standard behavior, even with autosave.)
- **For Users:** Manually saves the current *.wrtb file. Changes are autosaved every 30 seconds.

Save As

- **For Developers:** Duplicates the current *.wrtb file with a new name in Supabase. Copy JSONB data and update metadata.
- **For Users:** Saves a copy of the current *.wrtb file with a new name in WRITAB Drive.

Print

- **For Developers:** Generates a printable view of the current tab. Use browser print API or export to PDF via backend (Node.js with a library like pdfkit if integrated). (Added for standardization.)
- **For Users:** Prints the current Sheets tab or exports to PDF for printing.

Share

- **For Developers:** Generates a shareable link or invites users via Auth0 for authentication and Supabase row-level security. Use WebSocket for real-time collaborator updates. (Added for collaboration standard.)
- **For Users:** Shares the *.wrwb file with view/edit permissions via link or email.

Close

- **For Developers:** Unloads the current *.wrwb file from state, clearing Handsontable instance. Ensure autosave via Redux before closing. (Added for standard file management.)
- **For Users:** Closes the current *.wrwb file and returns to WRITAB Drive.

Tab Menu

Rename

- **For Developers:** Updates tab metadata in the *.wrwb JSONB structure via Supabase. (Added for usability.)
- **For Users:** Renames the current Sheets tab.

Delete

- **For Developers:** Removes the tab from the *.wrwb file's JSONB array in Supabase.
- **For Users:** Deletes the current tab.

Duplicate

- **For Developers:** Clones the tab's Handsontable state and inserts as a new tab in Supabase.
- **For Users:** Creates a copy of the current tab within the same *.wrwb file.

Export

- **For Developers:** Converts Handsontable data to formats like XLSX/CSV/PDF via backend (Node.js libraries like exceljs for XLSX). Upload to S3 if needed.
- **For Users:** Exports the tab to XLSX, CSV, PDF, or JSON.

Move to Another File

- **For Developers:** Transfers tab JSONB data between *.wrwb files in Supabase. (Added for multi-tab management.)
- **For Users:** Moves the tab to a different *.wrwb file in WRITAB Drive.

Edit Menu

(Added entirely for standardization, as it's essential in spreadsheets.)

Undo

- **For Developers:** Leverages Handsontable's built-in undo stack. Integrate with Redux for global state.
- **For Users:** Reverts the last action. Shortcut: Ctrl+Z.

Redo

- **For Developers:** Uses Handsontable's redo stack, synced with Redux.
- **For Users:** Reapplies the last undone action. Shortcut: Ctrl+Y.

Cut

- **For Developers:** Standard Handsontable or browser execCommand for cutting selected cells.
- **For Users:** Cuts selected content. Shortcut: Ctrl+X.

Copy

- **For Developers:** Handsontable copy command.
- **For Users:** Copies selected content. Shortcut: Ctrl+C.

Paste

- **For Developers:** Handles paste events in Handsontable, parsing content.
- **For Users:** Pastes clipboard content. Shortcut: Ctrl+V.

Find

- **For Developers:** Implements search via Handsontable's search plugin.
- **For Users:** Searches for text in the sheet. Shortcut: Ctrl+F.

Replace

- **For Developers:** Extends Find with replace logic, updating Handsontable cells.
- **For Users:** Finds and replaces text.

Select All

- **For Developers:** Selects entire sheet via Handsontable API.
- **For Users:** Selects all cells. Shortcut: Ctrl+A.

Delete

- **For Developers:** Clears selected cells or rows/columns.
- **For Users:** Deletes content from selected cells.

View Menu

Freeze

- **Sub-functions:** Rows (Up to current row), Columns (Up to current column), Unfreeze All (consolidated from specific counts for flexibility).
- **For Developers:** Uses Handsontable's freeze plugin to lock rows/columns.
- **For Users:** Freezes rows or columns to keep them visible while scrolling.

Group

- **For Developers:** Implements row/column grouping via Handsontable's nested rows/headers.
- **For Users:** Groups rows or columns for collapsible sections.

Ungroup

- **For Developers:** Removes grouping from selected rows/columns.
- **For Users:** Ungroups previously grouped sections.

Zoom In/Out

- **For Developers:** Adjusts CSS zoom or transform on the grid container. (Added for standard view controls.)
- **For Users:** Zooms the sheet view.

Full Screen

- **For Developers:** Uses browser fullscreen API.
- **For Users:** Expands to full screen.

Show Gridlines

- **For Developers:** Toggles Handsontable's gridline visibility.
- **For Users:** Shows/hides gridlines.

Hide Headers

- **For Developers:** Toggles row/column headers via Handsontable options. (Added for customization.)
- **For Users:** Hides row/column headers.

Insert Menu

Row

- **Sub-functions:** Above, Below.
- **For Developers:** Inserts rows using Handsontable's alter method.
- **For Users:** Inserts a row above or below the selected row.

Column

- **For Developers:** Inserts columns using Handsontable's alter method. (Added for completeness, as Row was present but not Column.)
- **For Users:** Inserts a column left or right of the selected column.

Cell

- **For Developers:** Shifts cells right/down via Handsontable.
- **For Users:** Inserts a new cell, shifting others.

Image

- **For Developers:** Uploads to S3 via Amplify, inserts as overlay in Handsontable (custom renderer if needed).
- **For Users:** Inserts an image from WRITAB Drive or upload.

Link

- **For Developers:** Applies hyperlink to cell via Handsontable's cell meta. (Renamed from Hyperlink for standard terminology.)
- **For Users:** Adds a hyperlink to a cell.

Chart

- **For Developers:** Inserts chart using Handsontable's chart integration or backend-generated (e.g., via Node.js with Chart.js). (Added for standard spreadsheet features.)
- **For Users:** Inserts a chart from selected data.

Function

- **For Developers:** Opens formula editor; inserts via Handsontable's formula plugin.
- **For Users:** Inserts a function into the cell.

Checkbox

- **For Developers:** Sets cell type to checkbox in Handsontable.
- **For Users:** Inserts a checkbox control.

Dropdown

- **For Developers:** Applies data validation with source list in Handsontable.
- **For Users:** Inserts a dropdown list.

Comment

- **For Developers:** Adds comment to cell meta, displayed on hover.
- **For Users:** Adds a comment to the selected cell.

Note

- **For Developers:** Similar to comment but persistent display; custom implementation.
- **For Users:** Adds a note (e.g., for longer annotations).

Sparkline

- **For Developers:** Renders mini-chart in cell via custom Handsontable renderer. (Added for advanced visuals.)
- **For Users:** Inserts a sparkline from data.

Format Menu

Number

- **Sub-functions:** General, Number, Currency, Percentage, Date, Time, Custom.
- **For Developers:** Applies number formats via Handsontable's numeric renderer.
- **For Users:** Formats cell numbers.

Text

- **Sub-functions:** Bold, Italic, Underline, Strikethrough, Font, Size, Color.
- **For Developers:** Applies text styles to cell meta.
- **For Users:** Formats cell text.

Alignment

- **Sub-functions:** Left, Center, Right, Top, Middle, Bottom. (Renamed from Align.)
- **For Developers:** Sets alignment in Handsontable options.
- **For Users:** Aligns cell content.

Wrapping

- **Sub-functions:** Wrap Text, Clip, Overflow.
- **For Developers:** Toggles wrap mode in cell styles.

- **For Users:** Controls text wrapping.

Rotation

- **Sub-functions:** 0°, 45°, 90°, -45°, -90°.
- **For Developers:** Applies CSS rotation to cell content.
- **For Users:** Rotates cell text.

Conditional Formatting

- **For Developers:** Rules-based styling via Handsontable's conditional rendering. (Added for standard.)
- **For Users:** Applies formats based on conditions.

Merge Cells

- **For Developers:** Merges selected cells using Handsontable's merge plugin. (Added for layout.)
- **For Users:** Merges cells into one.

Clear Formatting

- **For Developers:** Resets cell styles and formats. (Added for standard.)
- **For Users:** Removes all formatting.

Formula Menu

(Added as a separate menu to group calculation-related functions, standard in spreadsheets.)

Sum

- **For Developers:** Inserts SUM function via Handsontable.
- **For Users:** Sums selected range.

Average

- **For Developers:** Inserts AVERAGE function.
- **For Users:** Averages selected range.

Count

- **For Developers:** Inserts COUNT function.
- **For Users:** Counts cells in range.

Function List

- **For Developers:** Opens searchable list of all functions.
- **For Users:** Browses and inserts functions.

Named Ranges

- **For Developers:** Manages named ranges in sheet metadata stored in JSONB.
- **For Users:** Defines named ranges for formulas.

Recalculate

- **For Developers:** Forces Handsontable to recalculate formulas. (Added for manual control.)
- **For Users:** Recalculates all formulas.

Data Menu

Sort

- **Sub-functions:** Ascending, Descending, By Column.
- **For Developers:** Sorts range using Handsontable's sort plugin.
- **For Users:** Sorts selected data.

Filter

- **For Developers:** Applies auto-filter to headers via Handsontable.
- **For Users:** Adds filters to columns.

Filter Views

- **Sub-functions:** Create Filter View, Manage Filter Views (consolidated from Filter Group by View and Filter View).
- **For Developers:** Saves filter states in JSONB for reusable views.
- **For Users:** Creates temporary filter views without altering data.

Column Stats

- **For Developers:** Calculates stats (sum, avg, etc.) via backend API if complex.
- **For Users:** Shows statistics for selected column.

Data Validation

- **For Developers:** Sets validation rules on cells via Handsontable.
- **For Users:** Validates data entry (e.g., lists, numbers).

Data Cleanup

- **Sub-functions:** Remove Duplicates, Trim Whitespace.
- **For Developers:** Processes data via Handsontable or backend.
- **For Users:** Cleans up data.

Split Text to Columns

- **For Developers:** Splits cell content by delimiter into columns.
- **For Users:** Splits text into multiple columns.

Data Extraction

- **Sub-functions:** Extract Links, Emails, Numbers (clarified for specificity).
- **For Developers:** Uses regex via backend for extraction.
- **For Users:** Extracts specific data types from cells.

Pivot Table

- **For Developers:** Creates pivot via Handsontable extension or custom. (Added for advanced data analysis.)
- **For Users:** Builds a pivot table from data.

Tools Menu

Spelling and Grammar

- **For Developers:** Integrates a library like LanguageTool via backend API.
- **For Users:** Checks text in cells.

Auto-Correct Options

- **For Developers:** Configures auto-correct rules in Handsontable.
- **For Users:** Customizes auto-corrections.

Set Proofing Language

- **For Developers:** Sets sheet language metadata in Supabase.
- **For Users:** Changes proofing language.

Enable Formula Suggestions

- **For Developers:** Toggles AI-driven suggestions via gRPC to backend.
- **For Users:** Enables auto-suggest for formulas.

Enable Formula Corrections

- **For Developers:** Toggles error checking and corrections in formulas.
- **For Users:** Enables auto-corrections for formulas.

Protect Sheet

- **For Developers:** Applies protection via Handsontable's read-only modes and Supabase security. (Added for standard.)
- **For Users:** Protects sheet from edits.

Import Data

- **For Developers:** Imports from CSV/JSON via backend parsing. (Added for data entry.)
- **For Users:** Imports external data.

Review Menu

(Added for standardization, focusing on collaboration.)

Add Comment

- **For Developers:** Inserts comment on cell.
- **For Users:** Adds a comment.

Show Comments

- **For Developers:** Toggles comment display.
- **For Users:** Shows all comments.

Protect Range

- **For Developers:** Sets range-level protection.
- **For Users:** Protects specific ranges.

Track Changes

- **For Developers:** Enables change tracking via WebSocket and JSONB diffs.
- **For Users:** Tracks edits.

Accept Changes

- **For Developers:** Applies tracked changes.
- **For Users:** Accepts suggestions.

Reject Changes

- **For Developers:** Discards tracked changes.
- **For Users:** Rejects suggestions.

Help Menu

(Added for standardization.)

Help Center

- **For Developers:** Links to documentation or in-app tour.
- **For Users:** Opens WRITAB help resources.

Keyboard Shortcuts

- **For Developers:** Displays a modal with shortcuts list.
- **For Users:** Shows all shortcuts.

About

- **For Developers:** Displays version info from package.json.
- **For Users:** Shows app version and credits.

WRITAB Code Guidelines

Introduction

This document provides comprehensive guidelines for WRITAB Code, the code editor component of the WRITAB web app (version 1). WRITAB Code is built using the specified tech stack, including React for the frontend, Monaco Editor for code editing with syntax highlighting and basic completion, Node.js and Fastify for backend APIs, Supabase for database storage (using PostgreSQL with JSONB for code content and metadata), and Redis for real-time collaboration. All documents are saved as *.wrtb files, which can contain multiple tabs from Docs, Sheets, and Code apps. Tabs represent individual "files" within a *.wrtb container, and WRITAB Drive handles cloud storage for *.wrtb files, attachments, and other user files.

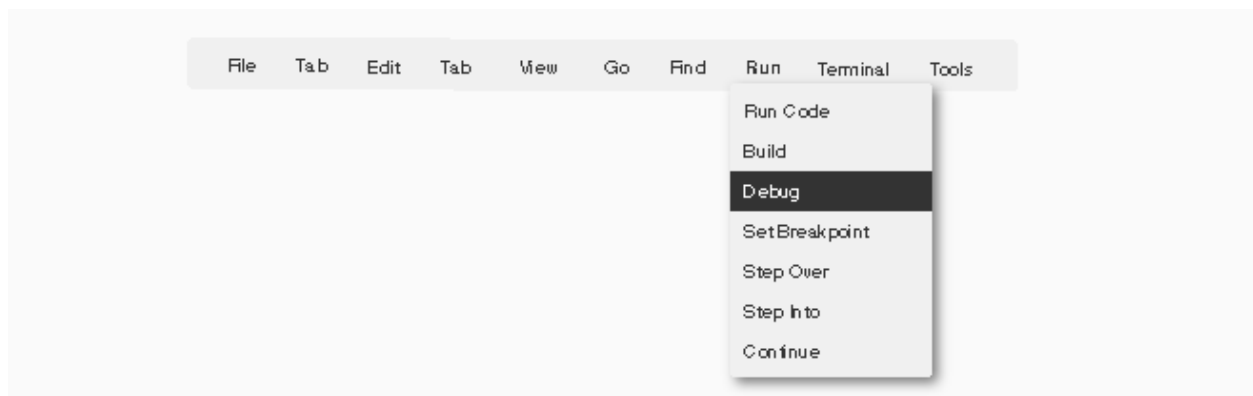
Guidelines are divided into two sections:

- **For Developers:** Technical implementation details, referencing the tech stack.
- **For Users:** Usage instructions.

The menu structure has been standardized to align with common code editors like VS Code and Monaco-based editors. Missing functions (e.g., Edit menu, Run/Debug, Terminal) have been added for completeness, drawing from standard IDE features. The "Data" menu has been removed as it is spreadsheet-specific and not relevant to code editing; no equivalent exists in standard code editors. Redundant items (e.g., overlapping Find/Replace across tabs) have been consolidated under Find for better organization. Unnecessary or repetitive sub-functions have been streamlined.

Menu Structure Overview

- **File:** Handles file-level operations for *.wrtb files.
- **Edit:** Standard editing actions.
- **Tab:** Tab-specific operations within a *.wrtb file.
- **View:** Controls display and layout.
- **Go:** Navigation tools.
- **Find:** Search and replace functions.
- **Run:** Build, compile, and execution (renamed from part of Tools for standardization).
- **Terminal:** Integrated terminal access (added for standard developer workflow).
- **Tools:** Utility functions like Command Palette.
- **Preferences:** Settings and customizations (renamed slightly for clarity).
- **Help:** Support resources.



Detailed Guidelines

File Menu

New

- **For Developers:** Creates a new *.wrtb file with an initial Code tab. Use Redux Toolkit to initialize state, Monaco Editor for the editor instance, and Supabase to insert a new JSONB record for file metadata and code content. Trigger a WebSocket (Socket.IO) event for real-time sync if collaborative.
- **For Users:** Creates a new blank code file as a *.wrtb file with one Code tab. Shortcut: Ctrl+N.

Open

- **For Developers:** Loads an existing *.wrtb file from WRITAB Drive. Query Supabase for file metadata, load code text into Monaco Editor. Use Axios for API calls to Fastify backend, with Redis caching for recent files.
- **For Users:** Opens a *.wrtb file from WRITAB Drive or local upload. Displays tabs in a tabbed UI.

Save

- **For Developers:** Autosaves changes to the current *.wrtb file via Supabase upsert. Use debounced Redux actions to serialize Monaco Editor content to JSONB. (Added for standard behavior, even with autosave.)
- **For Users:** Manually saves the current *.wrtb file. Changes are autosaved every 30 seconds.

Save As

- **For Developers:** Duplicates the current *.wrtb file with a new name in Supabase. Copy JSONB data and update metadata.
- **For Users:** Saves a copy of the current *.wrtb file with a new name in WRITAB Drive.

Print

- **For Developers:** Generates a printable view of the current tab. Use browser print API or export to PDF via backend (Node.js with a library like pdfkit if integrated). (Added for standardization.)
- **For Users:** Prints the current Code tab or exports to PDF for printing.

Share

- **For Developers:** Generates a shareable link or invites users via Auth0 for authentication and Supabase row-level security. Use WebSocket for real-time collaborator updates. (Added for collaboration standard.)
- **For Users:** Shares the *.wrtb file with view/edit permissions via link or email.

Close

- **For Developers:** Unloads the current *.wrtb file from state, clearing Monaco Editor instance. Ensure autosave via Redux before closing. (Added for standard file management.)
- **For Users:** Closes the current *.wrtb file and returns to WRITAB Drive.

Edit Menu

(Added entirely for standardization, as it's essential in code editors.)

Undo

- **For Developers:** Leverages Monaco Editor's built-in undo stack. Integrate with Redux for global state.
- **For Users:** Reverts the last action. Shortcut: Ctrl+Z.

Redo

- **For Developers:** Uses Monaco Editor's redo stack, synced with Redux.
- **For Users:** Reapplies the last undone action. Shortcut: Ctrl+Y.

Cut

- **For Developers:** Standard Monaco or browser execCommand for cutting selected text.
- **For Users:** Cuts selected text. Shortcut: Ctrl+X.

Copy

- **For Developers:** Monaco copy command.
- **For Users:** Copies selected text. Shortcut: Ctrl+C.

Paste

- **For Developers:** Handles paste events in Monaco, preserving formatting.
- **For Users:** Pastes clipboard content. Shortcut: Ctrl+V.

Select All

- **For Developers:** Selects entire editor content via Monaco API.
- **For Users:** Selects all text. Shortcut: Ctrl+A.

Format Document

- **For Developers:** Uses Monaco's formatting provider (integrate ESLint/Prettier via backend if needed). (Added for code beautification.)
- **For Users:** Formats the entire code document.

Toggle Comment

- **For Developers:** Toggles line/block comments via Monaco actions.
- **For Users:** Comments/uncomments selected lines. Shortcut: Ctrl+/.

Tab Menu

Rename

- **For Developers:** Updates tab metadata in the *.wrth JSONB structure via Supabase. (Added for usability.)
- **For Users:** Renames the current Code tab.

Delete

- **For Developers:** Removes the tab from the *.wrth file's JSONB array in Supabase.
- **For Users:** Deletes the current tab.

Duplicate

- **For Developers:** Clones the tab's Monaco content and inserts as a new tab in Supabase.
- **For Users:** Creates a copy of the current tab within the same *.wrth file.

Export

- **For Developers:** Exports Monaco content to formats like JS, PY, TXT via backend (Node.js file handling). Upload to S3 if needed.
- **For Users:** Exports the tab to common code file extensions (e.g., .js, .py, .html).

Move to Another File

- **For Developers:** Transfers tab JSONB data between *.wrth files in Supabase. (Added for multi-tab management.)
- **For Users:** Moves the tab to a different *.wrth file in WRITAB Drive.

View Menu

Sidebar

- **For Developers:** Toggles a React sidebar component (e.g., for file explorer, symbols). Use Tailwind CSS for styling.
- **For Users:** Shows/hides the sidebar (e.g., for outline, search results).

Split Editor

- **For Developers:** Splits Monaco Editor into multiple views via its diff editor or multi-editor setup.
- **For Users:** Splits the editor window for side-by-side editing.

Word Wrap

- **For Developers:** Toggles word wrap in Monaco options. (Added for standard view controls.)
- **For Users:** Enables/disables word wrapping.

Show Minimap

- **For Developers:** Toggles Monaco's minimap feature.
- **For Users:** Shows/hides code minimap.

Toggle Line Numbers

- **For Developers:** Sets line numbers visibility in Monaco.
- **For Users:** Shows/hides line numbers.

Zoom In/Out

- **For Developers:** Adjusts font size or zoom via Monaco API.
- **For Users:** Zooms the editor view.

Full Screen

- **For Developers:** Uses browser fullscreen API.
- **For Users:** Expands to full screen.

Go Menu

To File

- **For Developers:** Opens quick file picker via Monaco's quick open.
- **For Users:** Navigates to another file/tab. Shortcut: Ctrl+P.

To Line

- **For Developers:** Jumps to line number via Monaco's gotoLine action.
- **For Users:** Goes to a specific line. Shortcut: Ctrl+G.

To Symbol

- **For Developers:** Uses Monaco's symbol provider for navigation.
- **For Users:** Jumps to a symbol (e.g., function, variable).

To Definition

- **For Developers:** Implements go to definition using Monaco's language services (e.g., for TypeScript).
- **For Users:** Navigates to the definition of the selected symbol. Shortcut: F12.

Go Back

- **For Developers:** Uses Monaco's navigation history.
- **For Users:** Navigates back in history. Shortcut: Alt+Left.

Go Forward

- **For Developers:** Uses Monaco's navigation history.
- **For Users:** Navigates forward in history. Shortcut: Alt+Right.

Find Menu

Find

- **For Developers:** Opens Monaco's find widget for in-file search.
- **For Users:** Searches for text in the current tab. Shortcut: Ctrl+F.

Find in Tabs

- **For Developers:** Searches across all tabs via backend query on Supabase JSONB content.
- **For Users:** Searches across all open tabs.

Replace

- **For Developers:** Uses Monaco's replace functionality.
- **For Users:** Replaces text in the current tab. Shortcut: Ctrl+H.

Replace in Tabs

- **For Developers:** Batch replace across tabs via backend processing.

- **For Users:** Replaces text across all open tabs.

Find Next

- **For Developers:** Advances to next match in Monaco.
- **For Users:** Finds the next occurrence. Shortcut: F3.

Find Previous

- **For Developers:** Goes to previous match in Monaco.
- **For Users:** Finds the previous occurrence. Shortcut: Shift+F3.

Run Menu

(Added for standardization, incorporating Build and Compile; essential for code editors.)

Run Code

- **For Developers:** Executes code via backend (Node.js for JS, or integrate interpreters like Pyodide for Python if client-side; use gRPC for secure execution).
- **For Users:** Runs the current code tab.

Build

- **For Developers:** Triggers build process (e.g., via Node.js scripts or Docker if containerized).
- **For Users:** Builds the project.

Debug

- **For Developers:** Integrates Monaco's debugger with backend debugging (e.g., Node.js inspector).
- **For Users:** Starts debugging session. Shortcut: F5.

Set Breakpoint

- **For Developers:** Toggles breakpoints in Monaco.
- **For Users:** Adds/removes a breakpoint on the line.

Step Over

- **For Developers:** Debugger control via Monaco API.
- **For Users:** Steps over during debug. Shortcut: F10.

Step Into

- **For Developers:** Debugger control.

- **For Users:** Steps into function. Shortcut: F11.

Continue

- **For Developers:** Resumes execution.
- **For Users:** Continues from breakpoint. Shortcut: F5.

Terminal Menu

(Added for standard IDE features; integrates a terminal for commands.)

New Terminal

- **For Developers:** Embeds a web terminal (e.g., xterm.js) connected via WebSocket to backend shell.
- **For Users:** Opens a new terminal instance.

Run Command

- **For Developers:** Executes shell commands via Fastify API.
- **For Users:** Runs a command in the terminal.

Split Terminal

- **For Developers:** Splits terminal view.
- **For Users:** Splits the terminal pane.

Tools Menu

Command Palette

- **For Developers:** Opens Monaco's command palette for quick actions.
- **For Users:** Accesses all commands. Shortcut: Ctrl+Shift+P.

Problems

- **For Developers:** Displays diagnostics from Monaco (e.g., ESLint integration). (Added for error handling.)
- **For Users:** Shows problems panel.

Output

- **For Developers:** Displays build/run output in a panel.
- **For Users:** Views output from runs.

Preferences Menu

Settings

- **For Developers:** Opens JSON-based settings editor, stored in Supabase user metadata.
- **For Users:** Customizes editor settings (e.g., theme, font).

Keybindings

- **For Developers:** Allows editing keymap via Monaco API.
- **For Users:** Customizes keyboard shortcuts.

Extensions

- **For Developers:** Manages Monaco extensions (e.g., language support via backend).
- **For Users:** Installs/manages extensions.

Themes

- **For Developers:** Switches Monaco themes (light/dark). (Added for customization.)
- **For Users:** Changes editor theme.

Help Menu

Help Center

- **For Developers:** Links to documentation or in-app tour.
- **For Users:** Opens WRITAB help resources.

Keyboard Shortcuts

- **For Developers:** Displays a modal with shortcuts list.
- **For Users:** Shows all shortcuts.

About

- **For Developers:** Displays version info from package.json.
- **For Users:** Shows app version and credits.

WRITAB Drive Guidelines

Introduction

This document provides comprehensive guidelines for WRITAB Drive, the cloud storage component of the WRITAB web app (version 1). WRITAB Drive is built using the specified tech stack, including React for the frontend, Node.js and Fastify for backend APIs, Supabase for database storage (using PostgreSQL with JSONB for file metadata), AWS S3 for file storage via Amplify, and Redis for caching file lists and real-time updates. It handles storage for *.wrtb files (containing tabs from Docs, Sheets, and Code), attachments, and any other user-uploaded files, providing a unified cloud drive experience.

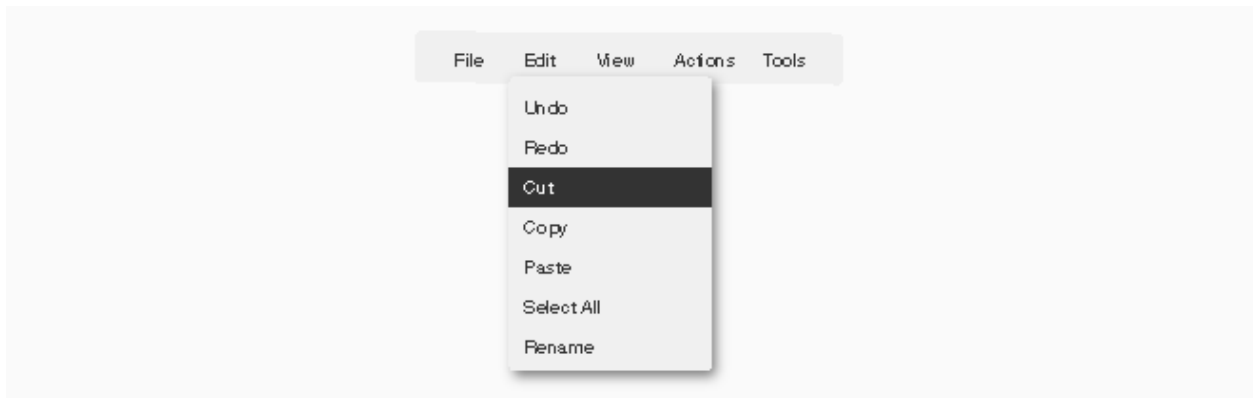
Guidelines are divided into two sections:

- **For Developers:** Technical implementation details, referencing the tech stack.
- **For Users:** Usage instructions.

The menu structure has been standardized to align with top cloud drive services like Google Drive, Dropbox, and OneDrive, while optimizing for WRITAB's *.wrtb files. Missing functions (e.g., Upload, Download, Share, Search) have been added for completeness and efficiency. Redundant items have been removed (e.g., no unnecessary overlaps in View). The Edit menu is retained for standard file operations, but streamlined.

Menu Structure Overview

- **File:** Handles creation, upload, and file operations.
- **Edit:** Standard editing actions for files/folders.
- **View:** Controls display and sorting.
- **Actions:** File-specific actions like move, delete (added as a separate menu for efficiency, common in drives).
- **Tools:** Utility functions like search (added for standard drive features).
- **Help:** Support resources.



Detailed Guidelines

File Menu

New Folder

- **For Developers:** Creates a new folder entry in Supabase metadata (JSONB for hierarchy), with optional S3 bucket prefix for storage.
- **For Users:** Creates a new folder in the current directory.

New WRITAB File

- **For Developers:** Initializes a new *.wrtb file in Supabase, linking to S3 for blob storage. Use Redux to update UI. (Added for WRITAB-specific creation.)
- **For Users:** Creates a new blank *.wrtb file.

Upload File

- **For Developers:** Handles file upload to S3 via Amplify, inserts metadata into Supabase. Use Axios for progress tracking.
- **For Users:** Uploads files from local device to WRITAB Drive.

Upload Folder

- **For Developers:** Recursively uploads folder structure to S3, mirroring in Supabase hierarchy.
- **For Users:** Uploads an entire folder.

Download

- **For Developers:** Generates presigned S3 URL or streams file via Fastify. For *.wrtb, serialize from Supabase JSONB if needed.
- **For Users:** Downloads selected file(s) or folder as ZIP.

Open

- **For Developers:** Loads file metadata from Supabase; for *.wrtb, opens in WRITAB apps (Docs/Sheets/Code) via Redux state.
- **For Users:** Opens the selected file (e.g., *.wrtb in appropriate tab).

Share

- **For Developers:** Generates shareable link with permissions via Auth0 and Supabase row-level security. Use WebSocket for real-time access updates.
- **For Users:** Shares file/folder with view/edit permissions via link or email. (Added for standard collaboration.)

Edit Menu

Undo

- **For Developers:** Reverts last action (e.g., delete/move) using Redis-cached history stack integrated with Redux.
- **For Users:** Undoes the last operation. Shortcut: Ctrl+Z.

Redo

- **For Developers:** Reapplies undone action from history stack.
- **For Users:** Redoes the last undone operation. Shortcut: Ctrl+Y.

Cut

- **For Developers:** Prepares selected items for move via clipboard state in Redux.
- **For Users:** Cuts selected file(s)/folder(s). Shortcut: Ctrl+X.

Copy

- **For Developers:** Copies metadata to clipboard state.
- **For Users:** Copies selected items. Shortcut: Ctrl+C.

Paste

- **For Developers:** Inserts copied/cut items, updating Supabase and S3 as needed.
- **For Users:** Pastes items into current directory. Shortcut: Ctrl+V.

Select All

- **For Developers:** Selects all items in current view via React state.
- **For Users:** Selects all files/folders. Shortcut: Ctrl+A.

Rename

- **For Developers:** Updates file/folder name in Supabase metadata and S3 key if applicable. (Added for standard file management.)
- **For Users:** Renames the selected item.

View Menu

List

- **For Developers:** Renders file list in table view using Tailwind CSS classes.
- **For Users:** Displays files as a detailed list (name, size, type, modified).

Grid

- **For Developers:** Switches to icon/grid layout with thumbnails (generate previews via backend if needed).
- **For Users:** Displays files as icons/grid. (Consolidated As Grid and As Columns for efficiency.)

Sort By

- **Sub-functions:** Name, Size, Type, Modified Date (Ascending/Descending).
- **For Developers:** Sorts query results from Supabase, cached in Redis for performance.
- **For Users:** Sorts the view by selected criteria.

Show Hidden Files

- **For Developers:** Toggles filter on Supabase query to include hidden items. (Added for advanced users.)
- **For Users:** Shows/hides hidden files.

Refresh

- **For Developers:** Re-queries Supabase and updates React component.
- **For Users:** Refreshes the current view. (Added for standard functionality.)

Actions Menu

(Added as a separate menu to group file operations efficiently, similar to context menus in drives.)

Move To

- **For Developers:** Updates file path in Supabase metadata and moves S3 object.
- **For Users:** Moves selected items to another folder.

Copy To

- **For Developers:** Duplicates S3 object and Supabase entry.
- **For Users:** Copies items to another location.

Delete

- **For Developers:** Soft-deletes via Supabase (move to trash table), with S3 archival.

- **For Users:** Deletes selected items (moves to Trash).

Restore

- **For Developers:** Reverses delete from trash.
- **For Users:** Restores from Trash. (Added for recoverability.)

Star/Favorite

- **For Developers:** Toggles star flag in Supabase metadata for quick access.
- **For Users:** Marks as favorite for easy retrieval. (Added for user convenience.)

Get Info

- **For Developers:** Fetches detailed metadata (size, created, permissions) from Supabase/S3.
- **For Users:** Shows file/folder properties.

Tools Menu

(Added for utility features common in drives.)

Search

- **For Developers:** Performs full-text search on Supabase metadata, with optional S3 content scan via backend.
- **For Users:** Searches for files/folders by name or content. Shortcut: Ctrl+F.

Recent Files

- **For Developers:** Queries Supabase for recently accessed items, cached in Redis.
- **For Users:** Shows recently opened files.

Trash

- **For Developers:** Views soft-deleted items in dedicated Supabase table.
- **For Users:** Opens the Trash folder.

Storage Usage

- **For Developers:** Calculates usage via S3 metrics and Supabase queries.
- **For Users:** Displays storage quota and usage. (Added for account management.)

Sync Settings

- **For Developers:** Configures offline sync using Redis and service workers. (Added for desktop/mobile integration via Electron/React Native.)
- **For Users:** Manages sync preferences.

Help Menu

Help Center

- **For Developers:** Links to documentation or in-app tour.
- **For Users:** Opens WRITAB help resources.

Keyboard Shortcuts

- **For Developers:** Displays a modal with shortcuts list.
- **For Users:** Shows all shortcuts.

About

- **For Developers:** Displays version info from package.json.
- **For Users:** Shows app version and credits.

Feedback

- **For Developers:** Submits user feedback via Fastify API to Supabase. (Added for improvement loop.)
- **For Users:** Sends feedback to developers.