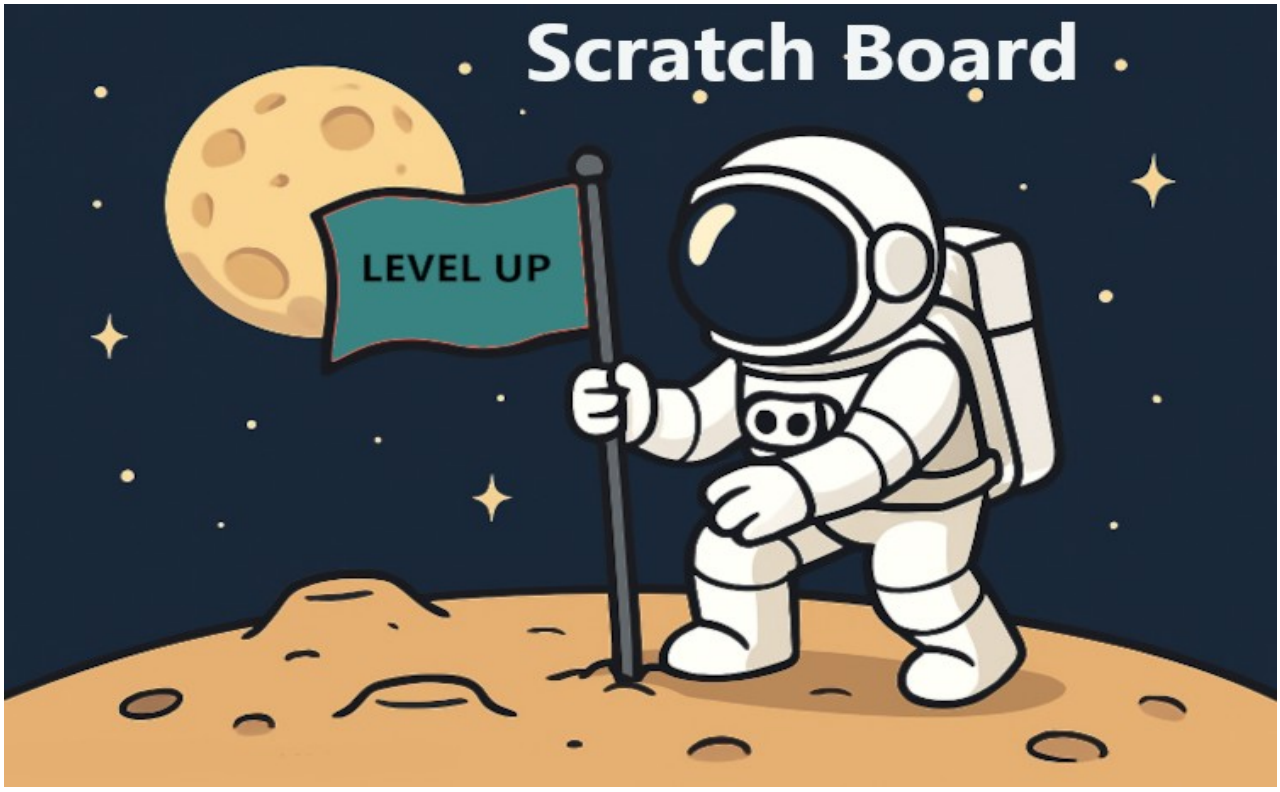


SCRATCH BOARD END-USER MANUAL

Abstract

The Scratch Board End-User Manual services as a comprehensive guide designed to help users efficiently download, install, and operate the Scratch Board application. This manual offers an in-depth overview of the software, detailing its core features, capabilities, and functionalities. It includes step-by-step instructions to ensure a smooth setup, empowering users to maximize their experience. This guide provides essential information to help you navigate and fully utilize the application's core tools and resources.

James Keenan

Go to: Scratch Board Code Repository

Table of Contents

Section I

This section provides an introduction to Scratch Board, a lightweight and versatile desktop productivity application meticulously designed to assist users in capturing, organizing, and exploring information and ideas in an efficient and seamless manner. With its intuitive interface and powerful functionality, Scratch Board makes information management both practical and enjoyable. It empowers users to stay organized and focused, particularly in a fast-paced, evolving environment, while enhancing their overall productivity.

Product Overview

Scratch Board is a lightweight desktop productivity application designed to help users capture, organize, and explore information and ideas efficiently. Combining the best features of note-taking, sticky notes, and advanced tools like MAC address querying and DOCSIS log parsing, Scratch Board offers a fun and practical environment for professional use.

Key Features

- **Scratch Notes:** Create colorful, draggable, and resizable sticky notes that auto-save, making it easy to capture quick thoughts and ideas on the fly.
- **Custom Links:** Easily add and delete quick links to helpful online resources for fast access.
- **Modem Log Parser:** Troubleshoot connection issues by parsing DOCSIS modem logs and gaining insights into network performance.
- **Execute Batch Files:** Seamlessly run Windows batch files for efficient task automation.
- **Notepad:** A fully integrated text editor, similar to Windows Notepad, with features to save, edit, and open plain-text files.
- **MAC Vendor Lookup:** Query MAC addresses to find the originating company and country, using the IEEE OUI database.
- **Password Generator:** Generate strong and secure passwords with entropy for better security.
- **Charts:** View informative charts on DOCSIS signals, fiber signals, bandwidth requirements, and more, to aid in decision-making.

User Interface & Navigation

- **Dashboard:** A modern dashboard that displays real-time statistics, graphs, and widgets to keep you informed.
- **Sidebar Navigation:** Quickly switch between different views and tools within the app for streamlined productivity.
- **Contacts Rolodex:** Maintain a digital address book by adding, editing, and deleting contacts, including their name, email, phone number, and website.
- **Note Categories:** Organize notes into useful categories such as Internet, Email, Phone, Video, Streaming, Notes, and Ideas.

Data Storage & Exporting

- **Local Database Storage:** Scratch Board uses a lightweight SQLite database to securely store user data, including notes, contacts, and categorized information.
- **Database Files:** The app creates and stores database files automatically in the `sb_data` folder within the program directory.
- **Export/Import Support:** Easily export the database to a human-readable `JSON` format for portability, or import it back into the app for further use.

Extras

- **Rich Text Formatting:** Customize your notes with rich-text formatting options, including font styles, sizes, and colors.
- **Image Embedding:** Embed images directly into notes using the provided toolbar button for enhanced visual notes and open them up in a window with a click of the right mouse button.
- **Keyboard Shortcuts:** Navigate quickly between views and tools using keyboard shortcuts, ensuring a fast and efficient workflow.

List of Tools & Libraries

This section provides an overview of the key tools and libraries used in the project. Tools are primarily responsible for building, packaging, and enhancing the functionality of the application, while libraries offer core functionalities like data parsing, system interaction, and visual enhancements.

Tool	Description
altgraph	A dependency analysis tool used by PyInstaller for analyzing Python modules and their dependencies, helping with packaging applications.
pyinstaller	A tool for converting Python programs into standalone executables, allowing distribution of applications without requiring users to install Python.
pyinstaller-hooks-contrib	A collection of additional hooks for PyInstaller to help it correctly handle special packages or custom libraries during packaging.
PyQt5-Qt5	The Qt5 library for Python, enabling the development of cross-platform graphical user interfaces (GUIs) using the Qt framework.
PyQt5_sip	A helper library for binding the Qt5 C++ libraries to Python, enabling interaction between the two.

Tool	Description
PySide6	Python bindings for Qt6, providing tools for building cross-platform GUI applications with the latest version of the Qt framework.
PySide6_Addons	Additional functionality and features for PySide6 to extend its capabilities with more advanced features and widgets.
PySide6_Essentials	Core modules for PySide6, including the essential classes and tools required for GUI development.
python-dateutil	A library for parsing and manipulating dates and times, extending Python's datetime module to handle more complex date operations.
pytz	A library for working with time zones, allowing accurate and timezone-aware datetime objects.
pywin32-ctypes	A Python package for calling into native Windows APIs using ctypes, allowing for low-level interaction with Windows system libraries.
PyYAML	A library for parsing and generating YAML (YAML Ain't Markup Language), used in configuration files and data serialization.
QDarkStyle	A dark theme for Qt-based applications, providing a consistent dark mode appearance across different platforms.
QtPy	A wrapper that provides a unified interface to both PyQt and PySide, simplifying the code for working with Qt's graphical user interface tools.
shiboken6	A tool for generating Python bindings for Qt6, enabling the creation of Python bindings for C++ libraries used in Qt-based applications.
sgmllib3k	A compatible version of the sgmllib module for Python 3, used for parsing SGML documents in Python 3.x.

Library	Description
beautifulsoup4	A library for parsing HTML and XML documents, often used for web scraping tasks.
certifi	A package providing Mozilla's root certificates for validating SSL

Library	Description
	certificates used in HTTPS requests.
charset-normalizer	A library for character encoding detection and normalization, commonly used in combination with requests.
colorama	A cross-platform library for colored terminal text, useful for creating colored output in CLI applications.
contourpy	A library for contour plotting and data visualization, useful in scientific and data analysis.
cycler	A library for creating iterable cycle objects, often used in matplotlib for controlling color cycles.
filelock	A platform-independent file locking mechanism, used to prevent concurrent access to the same resource.
fonttools	A library for manipulating OpenType and TrueType fonts, typically used for font creation or modification.
fsspec	A library for abstracting filesystem access, supporting a variety of backends such as local, cloud, or distributed file systems.
idna	A library that supports internationalized domain names (IDN) for proper DNS handling with non-ASCII characters.
kiwisolver	A fast, efficient solver for constraint-based layout problems, often used in GUI frameworks like matplotlib or Kivy.
Markdown	A Python implementation of the Markdown parser, used for converting Markdown syntax into HTML.
markdown2	A fast and complete Markdown-to-HTML converter with extensions for enhanced parsing and rendering capabilities.
packaging	A library for parsing version strings, requirements, and metadata of Python packages, helping with packaging and distribution.
pefile	A library for reading and analyzing Windows Portable Executable (PE) files, useful for malware analysis and reverse engineering.
pillow	A fork of the Python Imaging Library (PIL), providing powerful image processing capabilities like opening, editing, and saving various image formats.
pip-licenses	A tool for managing and displaying license information for Python dependencies, helping with compliance for open-source projects.

Library	Description
prettytable	A library for creating ASCII-based tables for displaying data in a well-organized format in terminal applications.
psutil	A cross-platform library for retrieving system and process information, such as CPU, memory, disk, and network usage.
Pygments	A library for syntax highlighting, often used in code editors, IDEs, or documentation to display code with colorized syntax.
pyparsing	A library for building grammars and parsers, useful for interpreting and analyzing custom data formats or languages.
regex	A modern, efficient alternative to Python's built-in re module for working with regular expressions.
requests	A popular library for sending HTTP requests, simplifying interaction with RESTful APIs and web servers.
setuptools	A Python package for building, distributing, and managing Python packages, often used in combination with pip.
sgmllib3k	A Python 3-compatible version of the sgmllib module, used for parsing SGML (Standard Generalized Markup Language) documents.
six	A compatibility library for ensuring code runs on both Python 2 and Python 3, providing uniform APIs for both versions.
soupsieve	A modernized version of the HTMLParser module used by BeautifulSoup for parsing and manipulating HTML content.
tqdm	A fast, extensible progress bar library, used to display progress in long-running loops or tasks.
typing_extensions	A backport of new features from Python's typing module, allowing access to newer type hinting features on older Python versions.
urllib3	A powerful, user-friendly HTTP client library for managing connections, pooling, and retries in HTTP communication.
wcwidth	A library for determining the width of Unicode characters, used for ensuring proper alignment and rendering in terminal or console output.

List of Development Tools

This section provides an overview of the essential development tools used throughout the project. These tools support various stages of software development, from writing and testing code to managing project dependencies and facilitating collaboration. Whether for version control, code editing, or debugging, each tool plays a vital role in enhancing productivity and ensuring high-quality software development.

Tool	Description
Python	The primary programming language used for developing the application. Python is known for its versatility to develop software quickly.
PyCharm	A popular integrated development environment (IDE) for Python. It provides powerful tools for coding, debugging, and testing.
GitHub	A platform for version control and collaborative development. GitHub hosts code repositories, enabling team collaboration and version tracking.
Git	A distributed version control system used for tracking code changes and collaboration. It integrates seamlessly with GitHub.
pytest	A testing framework for Python that makes it easy to write simple and scalable test cases, ensuring the reliability of the application.
Flake8	A tool for enforcing coding style standards in Python. It checks the code for style violations and potential errors, improving code quality.
Virtualenv	A tool for creating isolated Python environments, ensuring that dependencies are consistent across development and production environments.
pip	Python's package installer, used for managing dependencies and installing libraries from the Python Package Index (PyPI).

Installation

Installing **Scratch Board** is super easy. The application is packaged as a single native Windows ***.exe** file using PyInstaller, meaning you don't need to worry about setting up Python or dealing with complex dependencies.

It's a **native Windows executable**, so all you need to do is download the file and run it—there's no installation process, no need for an IDE, and no extra configuration required. Everything is bundled into one self-contained file for maximum convenience.

Steps for Installation

1. **Download the latest version of Scratch Board** from the official Scratch Board GitHub Repository under **Assets**.
    - Make sure you download the most recent version.
    - Drag the downloaded ScratchBoard.exe file to your desktop.
    - ***Important:** the software will create an sb_data folder – keep this folder on the desktop next to your ScratchBoard.exe file (it's your database).
2. Once the file is downloaded and dragged to your desktop, simply **double-click on the ScratchBoard.exe file** to launch the application.
    - If the blue Windows SmartScreen displays on **first run: click > more info then run anyway.**

      - otherwise -
    - The app will start immediately, with no need for additional installation steps, dependencies, or setup.
3. If you want to access it again later, just **double-click the ScratchBoard.exe file** to run it again—no installation is necessary, and **no changes are made to your system**.

This native Windows one-file bundle ensures that **Scratch Board** is ready to use right away. Whether you're troubleshooting issues, taking notes, or working with the integrated tools, you can start right away without worrying about complex configurations or system requirements.

It's a *truly portable and hassle-free solution*, designed to get you up and running in less than two minutes. Simply **download, run, and enjoy**.

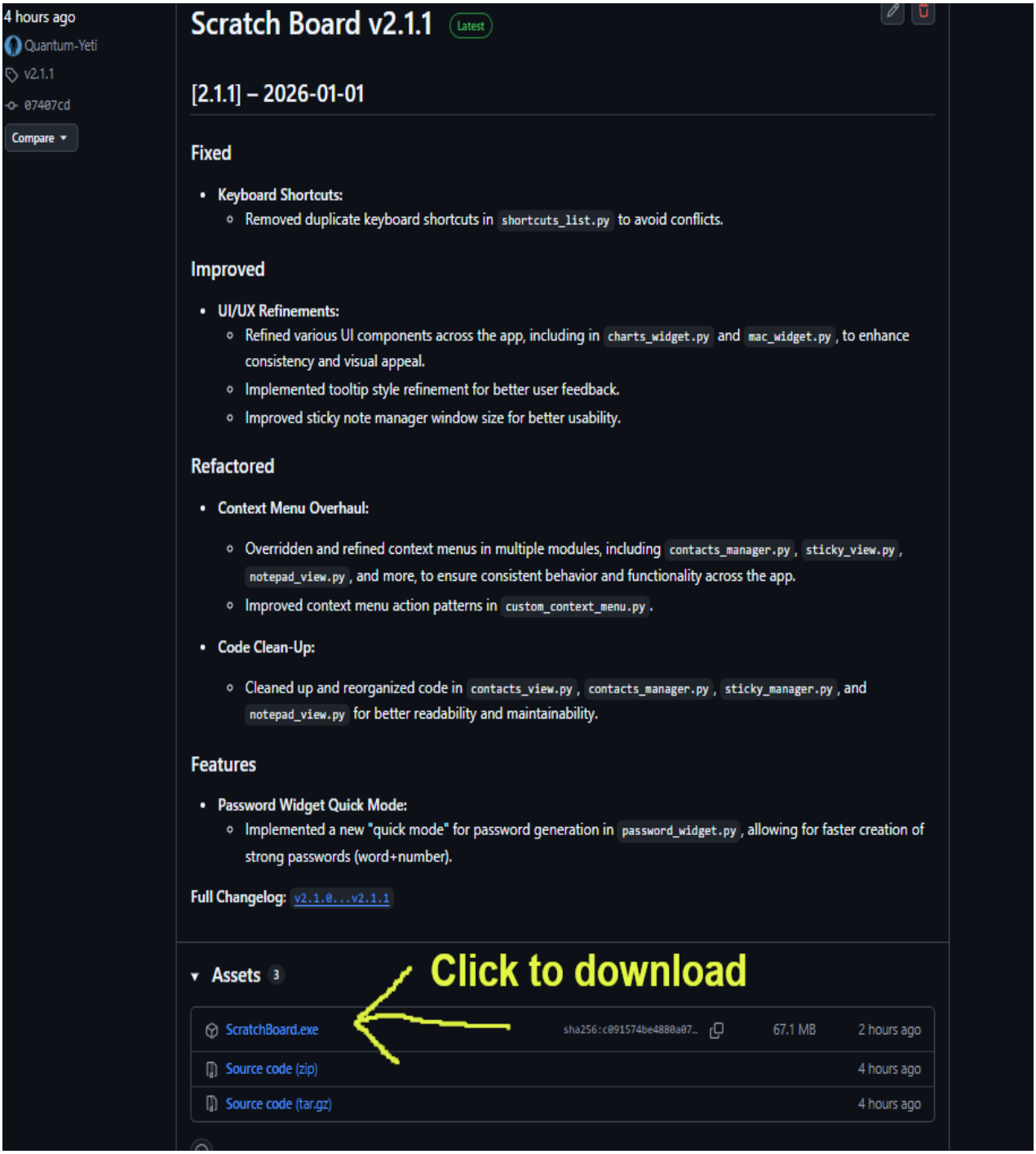


Figure 1: Download from repository.

Section II

This section provides an overview with illustrations of some of the key functionalities and essential features that are integral to the Scratch Board application. It aims to familiarize users with the core tools and capabilities available, offering insight into how each feature enhances the overall user experience and improves productivity.

Core Application Features

Dashboard

The Dashboard offers a comprehensive, user-friendly interface that provides quick access to key features and tools. It displays note statistics, a multi-line chart, and vital system information, as well as a MAC vendor lookup tool and a custom link widget. Additionally, the dashboard includes a dedicated area for quick note-taking or entering your ARM statement.

Designed with both functionality and aesthetics in mind, it serves as an intuitive, visually appealing starting point for seamless navigation and efficient use of the application.

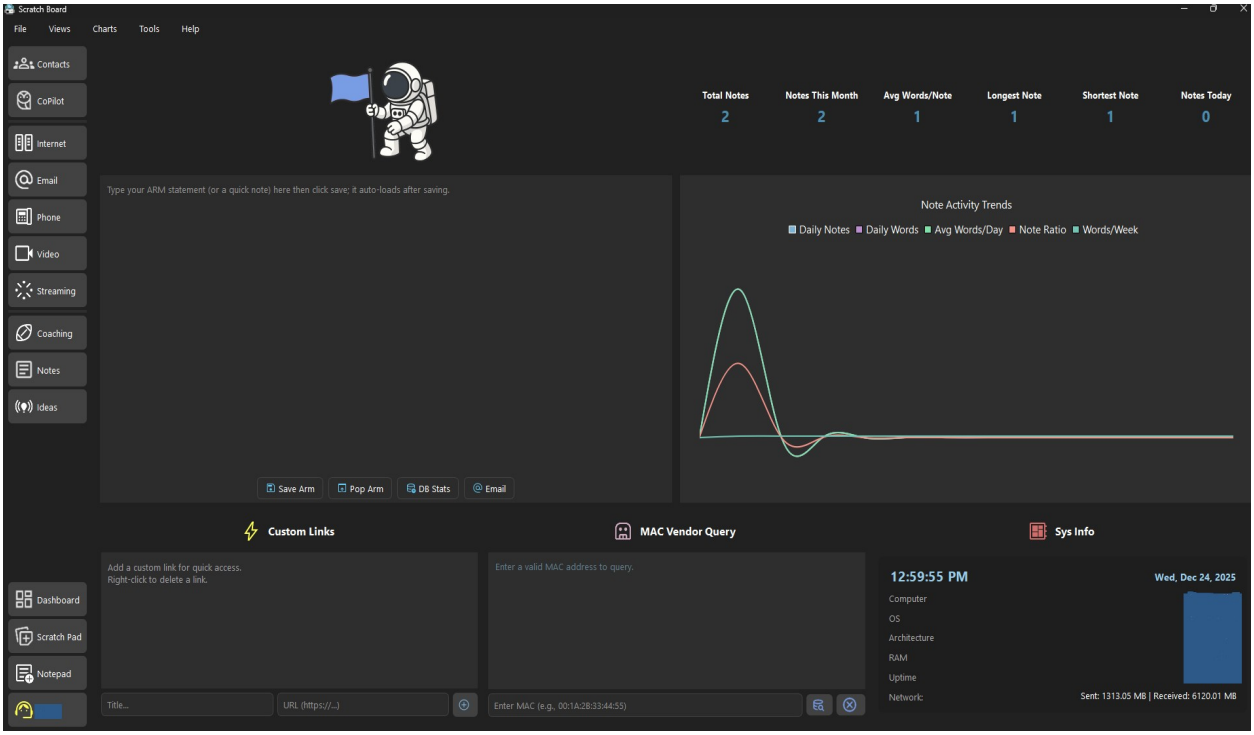


Figure 2: Dashboard View

Sticky Notes

Sticky Notes offer a versatile and dynamic way to capture quick thoughts, reminders, and important information. These colorful, draggable and resizable notes automatically save as you work, ensuring your ideas are securely stored without manual effort. Designed for ease of use, they allow for seamless organization of quick thoughts and information, offering a flexible solution for note-taking.

Whether you're brainstorming or jotting down a quick to-do list, Sticky Notes provide an intuitive and efficient way to stay organized. There is a simple sticky-note manager window with just two options: add a sticky-note or show all of the sticky notes. Each sticky note can be closed, deleted, or have a background color change.

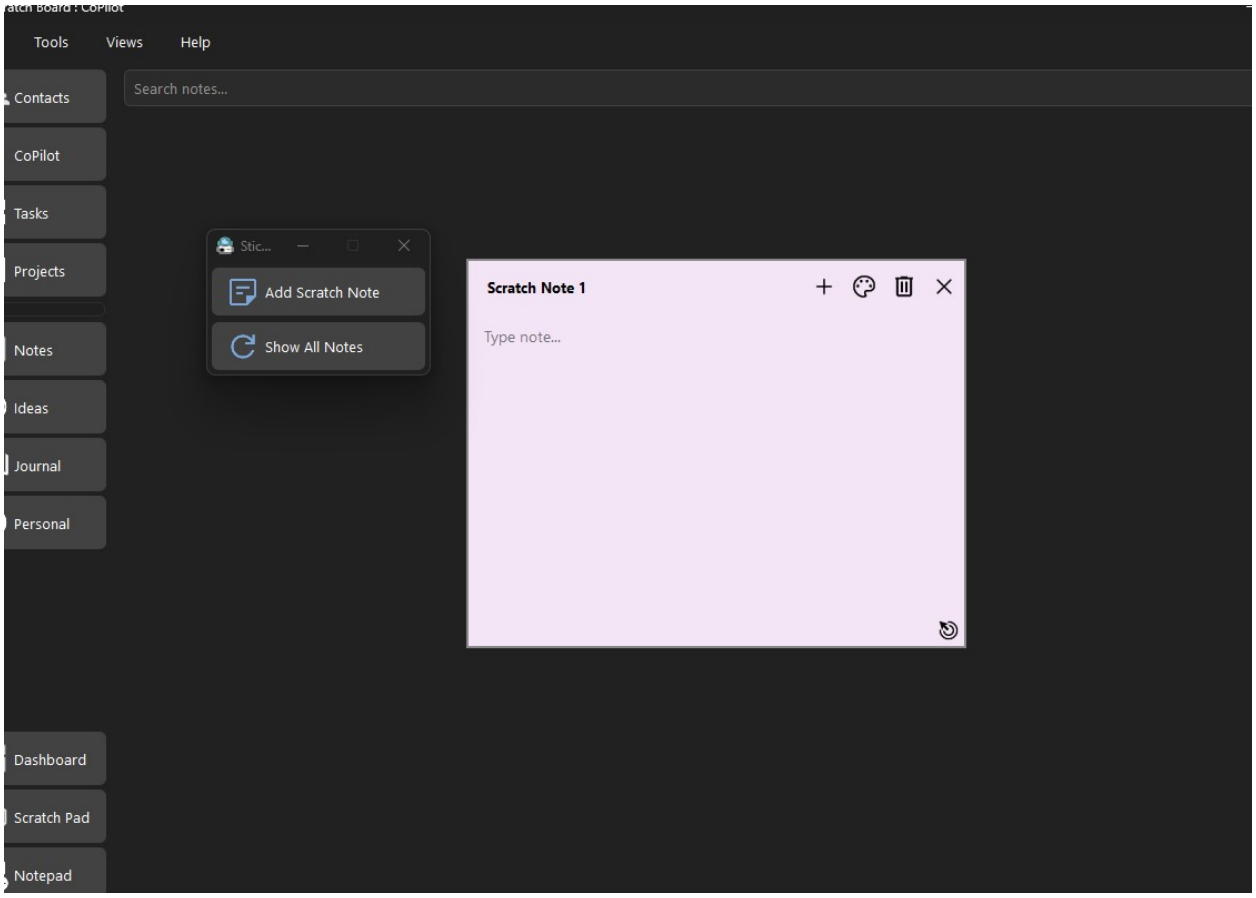


Figure 3: Sticky Note View

Notepad

Notepad provides a simple yet powerful text-editing experience within the application. It offers all the essential features of a standard notepad, including the ability to create, edit, save, and open plain-text files. Fully integrated into the app, the Notepad Widget ensures quick access to your notes without leaving the main interface.

Whether for jotting down ideas, drafting text, or organizing thoughts, this lightweight tool is designed for maximum convenience and seamless productivity, making it easy to stay focused and efficient.

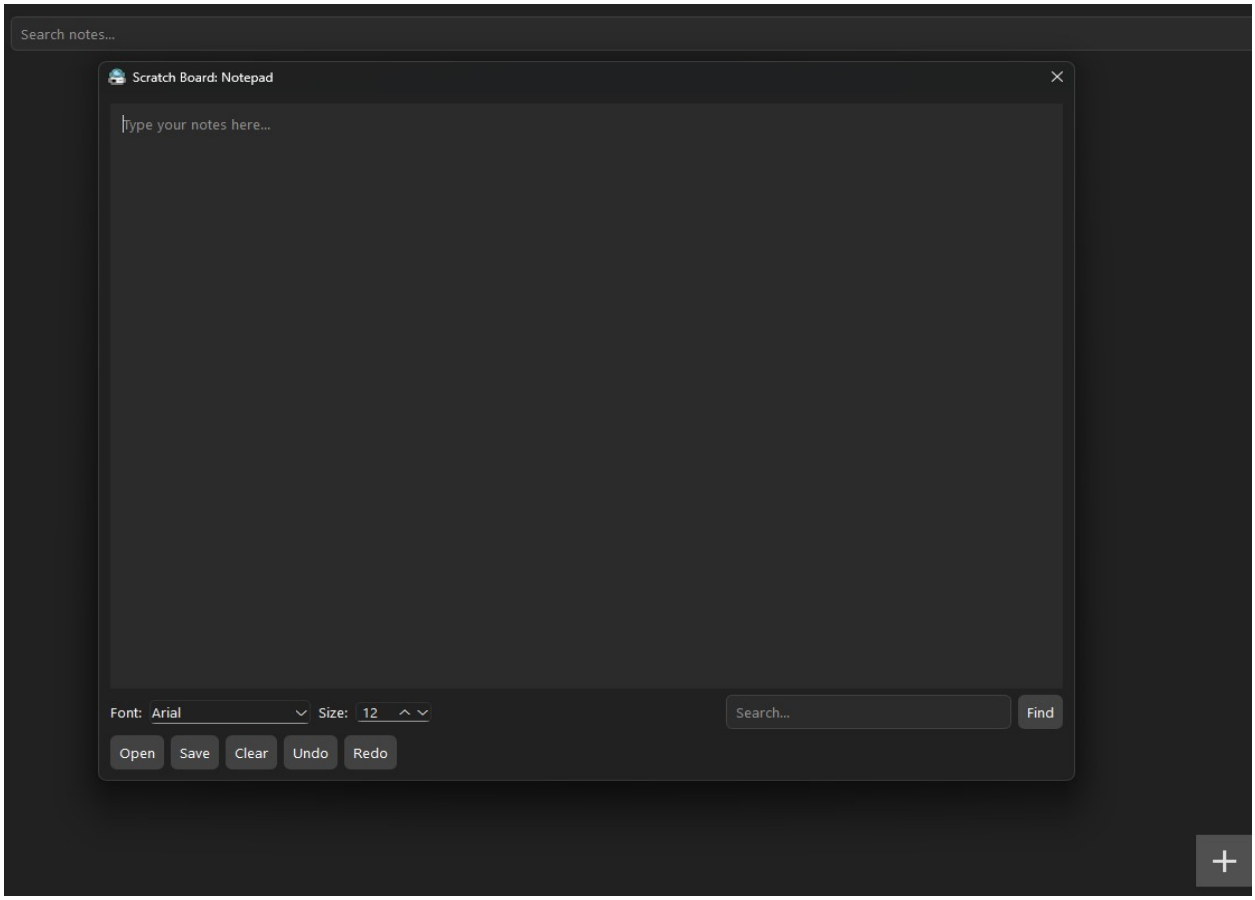


Figure 4: Notepad View

Contacts

Contacts View provides a streamlined, organized interface for managing professional contacts. It allows users to easily add, edit, and delete contact information, including names, email addresses, phone numbers, and websites. With an intuitive layout and simple navigation, the Contacts View ensures that important contact details are always at your fingertips, making it easier to stay connected and organized.

Whether for managing contacts for departments, supervisors, or managers this feature enhances efficiency and accessibility within the application.

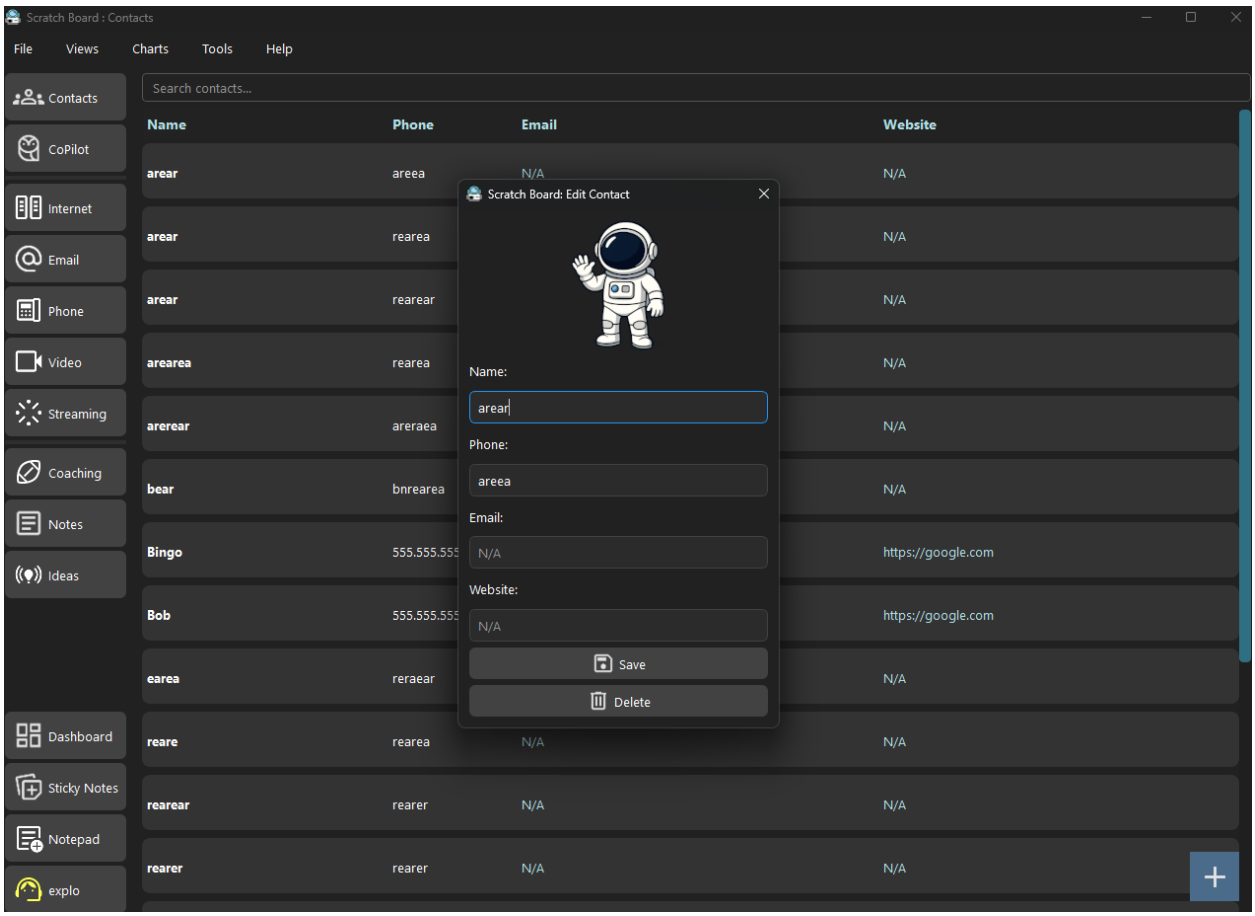


Figure 5: Contacts View

Password Generator

The password generator is a convenient tool for generating strong, randomized passwords. With three modes to choose from and customizable settings for length, complexity, and entropy, it allows users to create unique passwords tailored to their needs. The tool simplifies the process of safeguarding online accounts by providing quick and secure password generation. Whether you're creating a quick, one-time password or generating a highly complex one per customer request, the password generator ensures you can meet the demands of today's security needs.

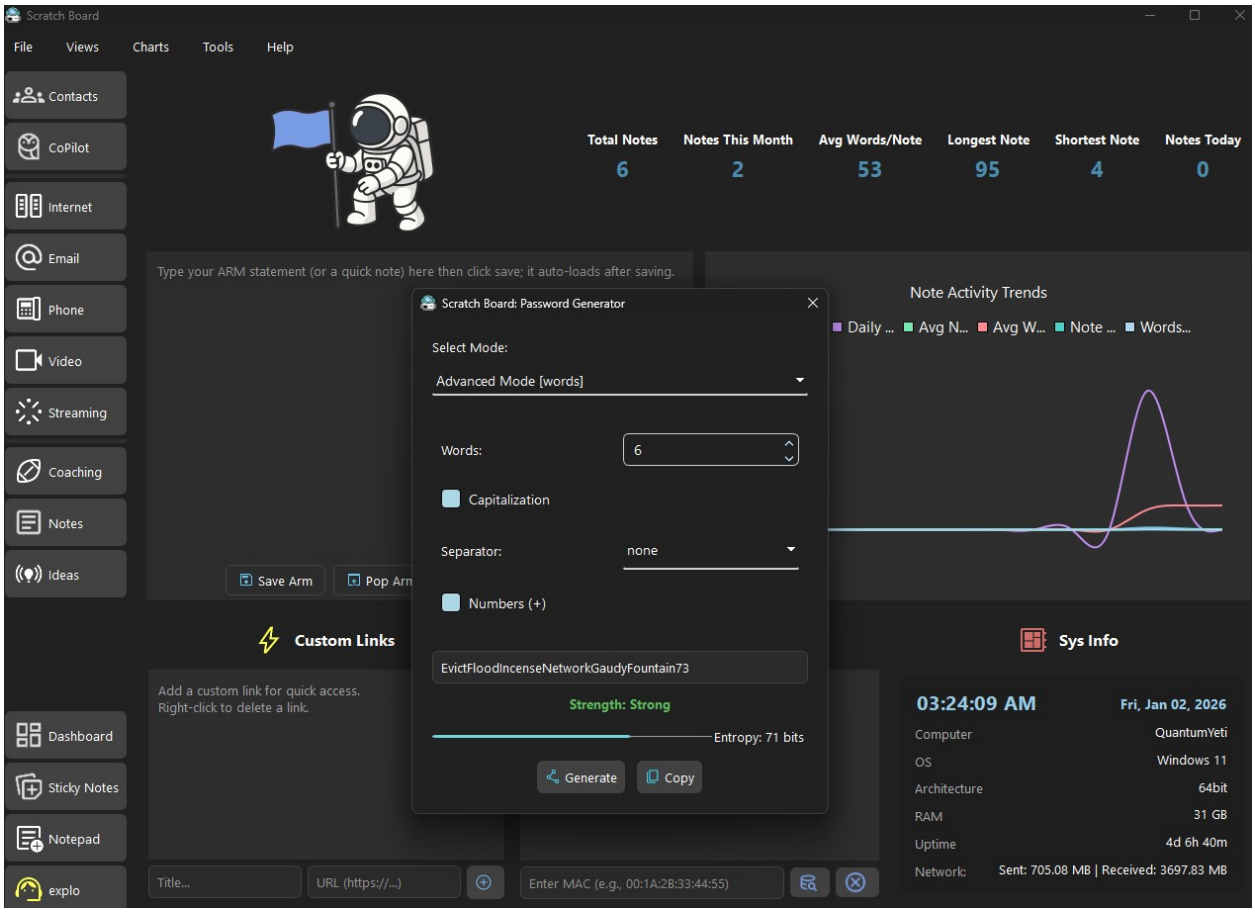
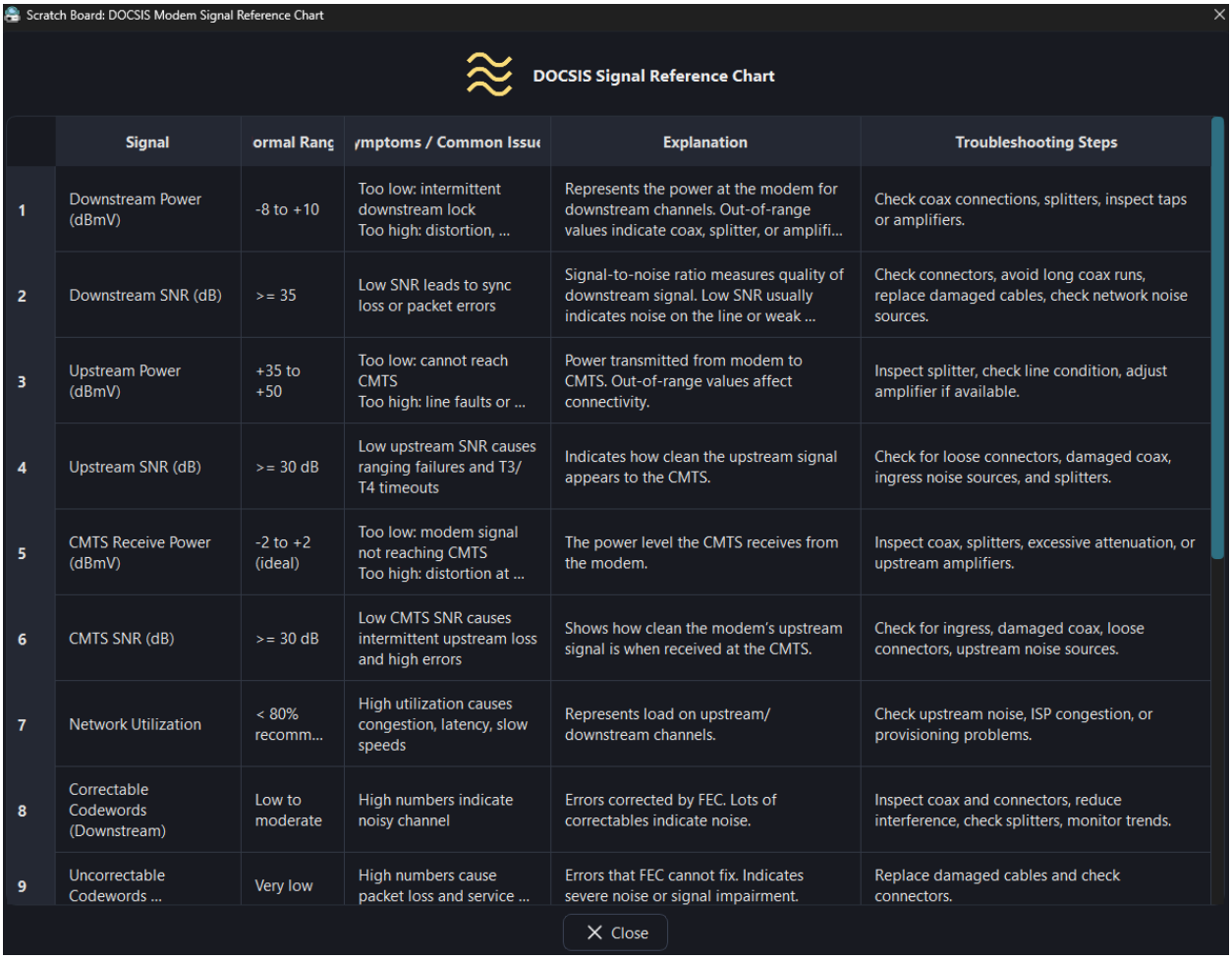


Figure 6: Password Generator View

Information Charts

The information charts found within the charts menu provide dynamic, data-driven visualizations to help you quickly understand and analyze key information. Whether it's DOCSIS signal statistics, fiber signal metrics, or bandwidth requirements, the widget displays complex data in easy-to-read charts, making technical details more accessible.

Designed for both simplicity and depth, the Info Chart Widget allows users to interact with various data sets and gain insights at a glance, making it an invaluable tool for both troubleshooting and decision-making.



	Signal	ormal Rang	ymptoms / Common Issue	Explanation	Troubleshooting Steps
1	Downstream Power (dBmV)	-8 to +10	Too low: intermittent downstream lock Too high: distortion, ...	Represents the power at the modem for downstream channels. Out-of-range values indicate coax, splitter, or amplifi...	Check coax connections, splitters, inspect taps or amplifiers.
2	Downstream SNR (dB)	>= 35	Low SNR leads to sync loss or packet errors	Signal-to-noise ratio measures quality of downstream signal. Low SNR usually indicates noise on the line or weak ...	Check connectors, avoid long coax runs, replace damaged cables, check network noise sources.
3	Upstream Power (dBmV)	+35 to +50	Too low: cannot reach CMTS Too high: line faults or ...	Power transmitted from modem to CMTS. Out-of-range values affect connectivity.	Inspect splitter, check line condition, adjust amplifier if available.
4	Upstream SNR (dB)	>= 30 dB	Low upstream SNR causes ranging failures and T3/T4 timeouts	Indicates how clean the upstream signal appears to the CMTS.	Check for loose connectors, damaged coax, ingress noise sources, and splitters.
5	CMTS Receive Power (dBmV)	-2 to +2 (ideal)	Too low: modem signal not reaching CMTS Too high: distortion at ...	The power level the CMTS receives from the modem.	Inspect coax, splitters, excessive attenuation, or upstream amplifiers.
6	CMTS SNR (dB)	>= 30 dB	Low CMTS SNR causes intermittent upstream loss and high errors	Shows how clean the modem's upstream signal is when received at the CMTS.	Check for ingress, damaged coax, loose connectors, upstream noise sources.
7	Network Utilization	< 80% recomm...	High utilization causes congestion, latency, slow speeds	Represents load on upstream/downstream channels.	Check upstream noise, ISP congestion, or provisioning problems.
8	Correctable Codewords (Downstream)	Low to moderate	High numbers indicate noisy channel	Errors corrected by FEC. Lots of correctables indicate noise.	Inspect coax and connectors, reduce interference, check splitters, monitor trends.
9	Uncorrectable Codewords ...	Very low	High numbers cause packet loss and service ...	Errors that FEC cannot fix. Indicates severe noise or signal impairment.	Replace damaged cables and check connectors.

Figure 7: Information Chart Views

Note Editor/Preview

The main core feature of the program is the note editor/preview. It provides a powerful, integrated workspace for creating and editing notes within their respective categories found in the sidebar menu. With a user-friendly interface and rich-text editing capabilities, it allows users to easily format and organize their notes, ideas, and information.

Whether you're drafting detailed text, adding images, or organizing information with bullet points and headings, the note editor/preview offers the flexibility and tools needed for efficient note creation. Seamlessly blending functionality and ease of use, it ensures a smooth and productive experience.

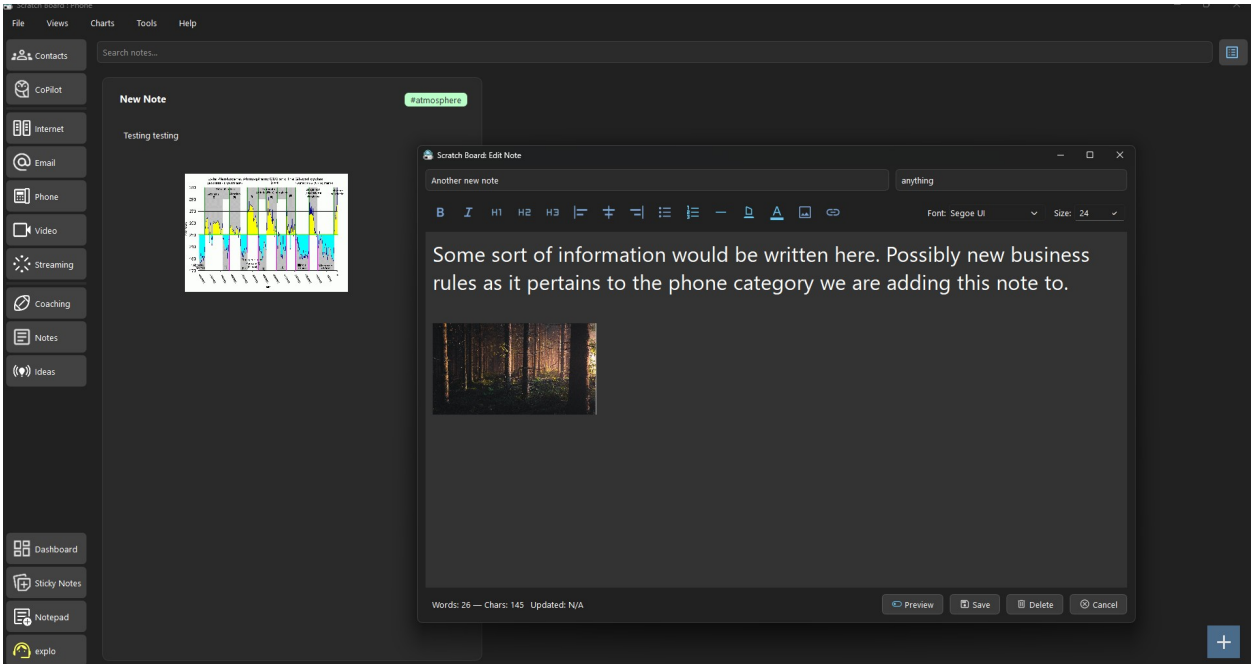


Figure 8: Note Editor/Preview

Enjoy!

https://github.com/quantum-yeti/scratchboard