

Book

Rakesh Venkat

2025-11-30

Table of contents

1 Overview

2 Documentation hub.

This site collects Technical Docs, Architecture Notes, and Design Decisions for all my projects.

3 How to Use This Site

- Use the **sidebar** or the project cards above to jump into a specific project.
- Within each project:
 - Start with the *Overview* chapter
 - Then read Architecture / Workflow chapters
 - Dive into Code Snippets, Testing, and Future Work as needed

This site is primarily written for **future me** — but if you're reading it later as a user or contributor, I hope it gives you clear insight into how everything fits together

4 Roadmap & Future Projects

This hub is designed to grow over time as more tools and experiments appear.

Planned additions include:

- Experimental tools around Rust / Ratatui / docs automation
- Libraries for working with generated JSON data
- More utilities for automating documentation workflows

When a new project is created, it will get:

- Its own folder (e.g. `new-project/`)
- Its own mini “book” of chapters
- A card on this page linking into its docs

5 About Me

I'm **Rakesh V**, a developer and tinkerer who enjoys building tools that make it easier to think, automate, and explore ideas.

I maintain the projects in this “book” — most of them in Rust, some around Ratatui, and others around documentation and tooling.

If you'd like to know more about my background, how these tools came about, and how to support my work, see:

[Read more in “Author” section](#)

6 Projects Overview

6.0.1 **ratcrate-core**

The **engine** behind the Ratatui ecosystem index.

- Scans the crates.io sparse index
- Finds all crates depending on **ratatui**
- Enriches them with crates.io metadata
- Applies smart caching (version + timestamp based)
- Produces **ratcrate.json** for other tools to consume

[Read the ratcrate-core docs »](#)

6.0.2 **ratcrate-cli**

A command-line interface for exploring the Ratatui ecosystem.

- Reads **ratcrate.json** produced by **ratcrate-core**
- Search and filter Ratatui crates from the terminal
- Show stats, links, and crate details in a compact way

[Read the ratcrate-cli docs »](#)

6.0.3 ratcrate-tui (*planned*)

A **Ratatui-based TUI** for browsing Ratatui crates.

Very meta. Very fun.

- Terminal UI built with Ratatui
- Interactive search and filters
- Crate detail views with stats and links
- Powered by `ratcrate-core` data

[Read the ratcrate-tui docs »](#)

6.0.4 doc2quarto

A helper tool for turning rough notes, markdown, or code comments into **structured Quarto documentation**.

- Convert notes into `.qmd` skeletons
- Generate chapter structures
- Insert placeholders for diagrams/screenshots
- Ideal for standardizing docs across all projects

[Read the doc2quarto docs »](#)

6.0.5 QBanner

A **Ratatui-based TUI** for browsing Ratatui crates.

Very meta. Very fun.

- Terminal UI built with Ratatui
- Interactive search and filters

- Crate detail views with stats and links
- Powered by `ratcrate-core` data

[Read the Qbanner Libraryi docs »](#)

6.0.6 QSwitch (Planned)

A helper tool for turning rough notes, markdown, or code comments into **structured Quarto documentation**.

- Optimized workflow

[Coming Soons »](#)

7 Author

8 About the Author

Hi, I'm **Rakesh Venkat** — a developer, researcher, and lifelong tinkerer who likes building tools that:

- Automate away repetitive work
- Make it easier to explore ideas
- Give people nicer ways to learn from and navigate code

Most of the projects in this book revolves around:

- **Rust** and terminal tools
- **Ratatui** TUIs (very meta)
- **Documentation** and knowledge organization
- Small research & experiments that later become libraries or tools

I think of this book as a logbook of the systems I'm building and how they fit together.

8.1 What I'm Trying to Do With These Projects

The broader goals behind these tools:

- Build **high-leverage tools** that save time for other developers.
- Collect and organize knowledge (like the Ratatui ecosystem) so people don't have to rediscover the same things repeatedly.
- Treat documentation as a first-class artifact, not an afterthought.