

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#) X

How to Create a Perfect README for Your GitHub Project



Sumuditha Lansakara · [Follow](#)

5 min read · Feb 5, 2024

56



...

[Open in app ↗](#)**Medium**

Search



Write



Introduction

A README file is the first thing people see when they come across your project on GitHub. It is a crucial file that introduces your project, provides instructions on how to use it, and gives necessary information to

contributors. In this blog post, we will go through the steps of creating a proper README file for your GitHub project. We will also introduce the README syntax and its basic features.

What is a README file?

A README file is a text file that contains information about your project. It is usually placed at the root of your project's repository and is the first file people see when they visit your project's page on GitHub. A good README file should include the following elements:

- A clear and concise title that describes your project
- An introduction that explains what your project is about and what problem it solves
- Installation instructions that explain how to set up and run your project
- Usage instructions that explain how to use your project
- Contributing guidelines that explain how others can contribute to your project
- License information that explains the licensing terms of your project

Basic Writing and Formatting Syntax

The README file uses the Markdown syntax for formatting. Markdown is a lightweight markup language that allows you to add formatting elements to plaintext text documents. Here are some basic Markdown syntax elements

that you can use in your README file:

- Headers: Use the `#` symbol followed by a space for headers. For example, `# Header 1`, `## Header 2`, and `### Header 3`.
- Emphasis: Use `*` or `_` for italic and `**` or `__` for bold. For example, `*italic*` or `_italic_` for italic and `**bold**` or `__bold__` for bold.
- Lists: Use `-`, `*`, or `+` for unordered lists and numbers followed by a period for ordered lists. For example,

```
- Unordered list item 1
- Unordered list item 2
1. Ordered list item 1
2. Ordered list item 2
```

- Links: Use `[link text](URL)` for links. For example, `[GitHub](https://github.com/)`.
- Images: Use `![alt text](URL)` for images. For example, `![GitHub logo](https://github.com/images/logos/octocat-logo-1200x630.png)`.
- Code: Use ``code`` for inline code and ````code block````` for code blocks. For example,

```
# This is inline code
```

```
```python
This is a code block
def hello_world():
 print("Hello, world!")
```

## Step-by-Step Guide to Creating a README File

Now that we have covered the basics of the README file and its syntax, let's go through the steps of creating a proper README file for your GitHub project.

### Step 1: Initialize Your Project

Before creating a README file, you need to initialize your project on GitHub. Here are the steps to do so:

1. Go to GitHub and sign in to your account.
2. Click on the “New” button on the top right corner of the page.
3. Fill in the necessary information for your project, such as the name, description, and public or private visibility.
4. Click on the “Create repository” button.

### Step 2: Create a README File

Once you have initialized your project, you can create a README file. Here are the steps to do so:

1. Navigate to your project's repository on GitHub.
2. Click on the “Create new file” button.
3. Name your file “README.md” (the “.md” extension stands for Markdown).
4. Write the necessary information for your README file using the Markdown syntax.

## Step 3: Add the Necessary Elements

As we mentioned earlier, a good README file should include the following elements:

1. Project Title: A clear and concise title that describes your project.
2. Introduction: An explanation of what your project is about, what problem it solves, and why it's useful.
3. Installation: Step-by-step instructions on how to set up and run your project, including any dependencies or prerequisites.
4. Usage: Examples and instructions on how to use your project, including any command-line arguments or configuration options.
5. Contributing: Guidelines for contributing to your project, including how to submit pull requests, report bugs, and request features.
6. License: Information about the licensing terms of your project, including any restrictions on commercial use or modification.

## Step 4: Final Review and Revision

- Conduct a thorough review of the entire README file.
- Ensure all information is accurate and up-to-date.
- Check for clarity and readability in the instructions.
- Seek feedback from colleagues or the community for improvements.

By following these point-form actions, you can systematically create a detailed and effective README file for your GitHub project.

---

---

```
Example README File
```

```
Project Title
```

Project Title is a description of the project and its purpose.

```
Introduction
```

Project Title is a project that does something useful. It was created to solve

```
Installation
```

To install Project Title, follow these steps:

1. Clone the repository: `git clone https://github.com/username/project-title`
2. Navigate to the project directory: `cd project-title`
3. Install dependencies: `npm install`
4. Build the project: `npm run build`
5. Start the project: `npm start`

## ## \*\*Usage\*\*

To use Project Title, follow these steps:

1. Open the project in your favorite code editor.
2. Modify the source code to fit your needs.
3. Build the project: **npm run build**
4. Start the project: **npm start**
5. Use the project as desired.

## ## \*\*Contributing\*\*

If you'd like to contribute to Project Title, here are some guidelines:

1. Fork the repository.
2. Create a new branch for your changes.
3. Make your changes.
4. Write tests to cover your changes.
5. Run the tests to ensure they pass.
6. Commit your changes.
7. Push your changes to your forked repository.
8. Submit a pull request.

## ## \*\*License\*\*

Project Title is released under the MIT License. See the [\[LICENSE\]\(https://www.example.com/licenses/MIT\)](#).

## ## \*\*Authors and Acknowledgment\*\*

Project Title was created by [\[Your Name\]\(https://github.com/username\)](#).

Additional contributors include:

- [\[Contributor Name\]\(https://github.com/contributor-name\)](#)
- [\[Another Contributor\]\(https://github.com/another-contributor\)](#)

Thank you to all the contributors for their hard work and dedication to the project.

## ## \*\*Code of Conduct\*\*

Please note that this project is released with a Contributor Code of Conduct. By contributing, you agree to abide by it.

## ## \*\*FAQ\*\*

**\*\*Q:\*\*** What is Project Title?

**\*\*A:\*\*** Project Title is a project that does something useful.

**\*\*Q:\*\*** How do I install Project Title?

**\*\*A:\*\*** Follow the installation steps in the README file.

**\*\*Q:\*\*** How do I use Project Title?

**\*\*A:\*\*** Follow the usage steps in the README file.

**\*\*Q:\*\*** How do I contribute to Project Title?

**\*\*A:\*\*** Follow the contributing guidelines in the README file.

**\*\*Q:\*\*** What license is Project Title released under?

**\*\*A:\*\*** Project Title is released under the MIT License. See the **[LICENSE]**([http://.../LICENSE](#))

## ## **\*\*Changelog\*\***

- **\*\*0.1.0:\*\*** Initial release
- **\*\*0.1.1:\*\*** Fixed a bug in the build process
- **\*\*0.2.0:\*\*** Added a new feature
- **\*\*0.2.1:\*\*** Fixed a bug in the new feature

## ## **\*\*Contact\*\***

If you have any questions or comments about Project Title, please contact **[You]**([http://.../You](#))

## ## **\*\*Conclusion\*\***

That's it! This is a basic template for a proper README file for a general proj

[Github](#)[Github Readme](#)[Project Readme](#)[Project Documentation](#)[Readme](#)



## Written by Sumuditha Lansakara

[Follow](#)

54 Followers · 5 Following

Product Engineer @MetanaHQ | Former β MLSA | Founding Ex-Co @APIIT FCS | Technical Writer | GDG | Postman Student Expert | 2xAzure | 2xFortinet | Gamer

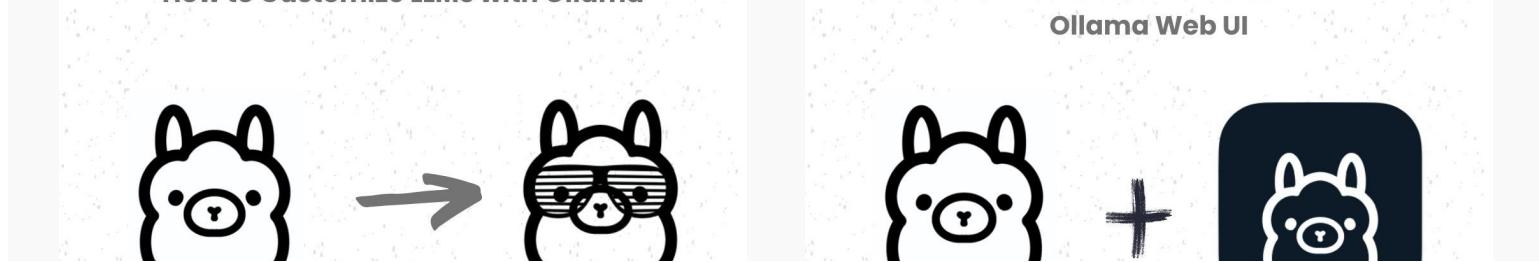
## No responses yet



Rain08

What are your thoughts?

## More from Sumuditha Lansakara





 Sumuditha Lansakara

## How to Customize LLMs with Ollama

“Create an image featuring a vibrant community of developers and users...”

Mar 9, 2024  105  1



...



 Sumuditha Lansakara

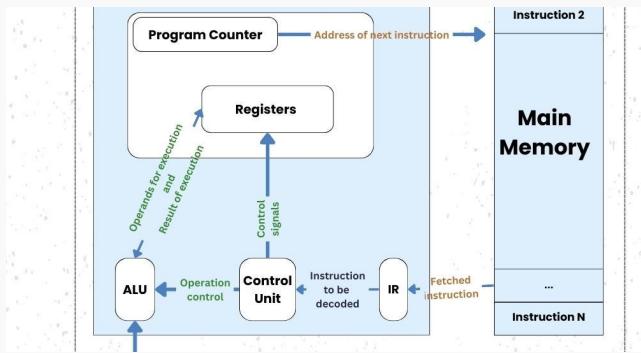
## How to Create a Self-Hosted LLM with Ollama Web UI

“In today’s interconnected world, accessibility and flexibility are paramount....”

Mar 10, 2024  23  1



...



 Sumuditha Lansakara

## What is fetch-decode-execute cycle ?

The central processing unit (CPU) is like the brain of the computer. It works by following ...

May 24, 2024



...

 Sumuditha Lansakara

## Building a Telegram Bot with Python and Gemini AI

Telegram bots are a great way to interact with users and automate tasks. In this tutorial,...

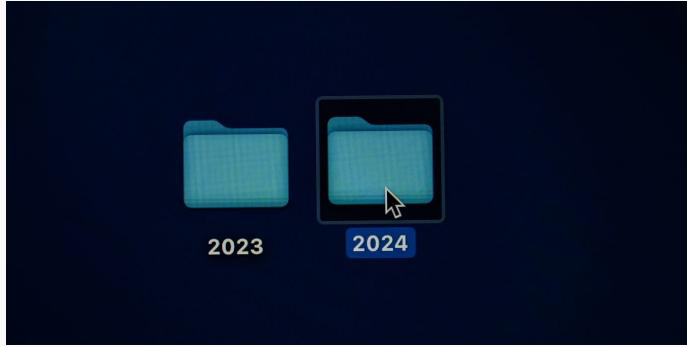
Mar 4, 2024  84



...

See all from Sumuditha Lansakara

## Recommended from Medium



 Teepiphat Phokaewkul

### How to Check all folder size using macOS Terminal

To check the sizes of all subfolders within a directory on macOS using the Terminal, you...

Nov 26, 2024  1



```

6 Func (I *Listener) listenToQueue(ctx context.Context, topic string, queue MessageQueue {
7 ...
8 // set the callback function for this event to commit the message to kafka
9 wrapper.SetCallback(func(c context.Context) {
10 ...
11 wrapper.CommitToKafka(c, queue, msg)
12 })
13 }
14
15 l.workerPool.Submit(func() {
16 ...
17 if err := l.eventHandler.HandleEvent(ctx, wrapper); err != nil {
18 log.Errorf(ctx, "Error handling event", log.Tags("topic", topic,
19 "errcode", err))
20 wrapper.OnCallback() // commit the message to kafka
21 }
22 }
23 }
24
25 Add to Chat GPT Edit HK
26 }
27
28 sighan := make(chan os.Signal, 1)
29 signal.Notify(sighan, syscall.SIGINT, syscall.SIGTERM)
30 <signal>
31
32 log.Info(ctx, "Received signal, shutting down...", log.Tags("topic", topic))
33 }
```

I added this code block, but copied it blindly from the example given in the segment/kafka-go docs. Is this the correct way to handle this? Or is there a better way?

Here's how to fix it:

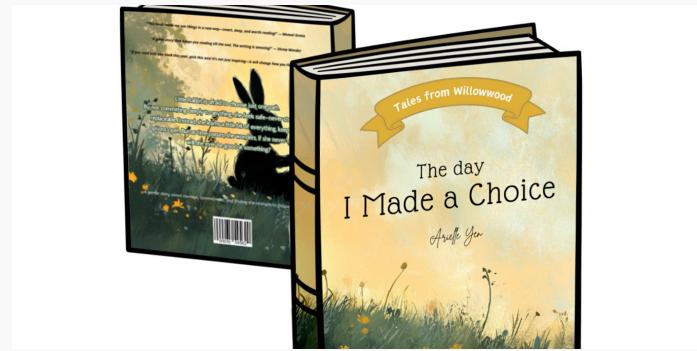
```

4 Func (I *Listener) Run(ctx context.Context) error {
5 ...
6 for topic, queue := range l.queue {
7 go l.listenToQueue(ctx, topic, queue)
8 }
9 <-ctx.Done()
10 return nil
11 }
```

func (I \*Listener) listenToQueue(ctx context.Context, topic string, queue MessageQueue {
12 ...
13 defer queue.Close()
14 for {
15 ...
16 }
17 }

 In Level Up Coding by Jacob Bennett

### The 5 paid subscriptions I actually

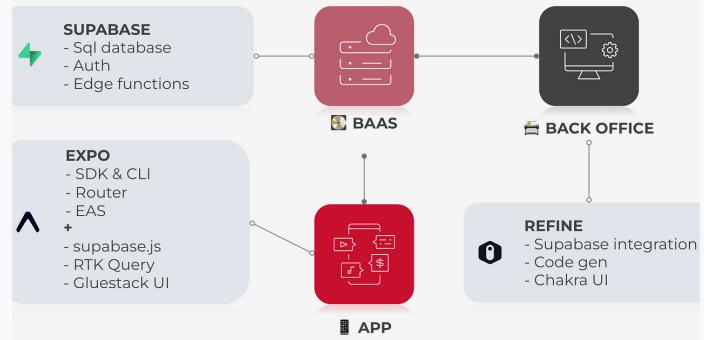


 Arielle Y.

### How I Use AI Tools to Make a Children's Storybook in Just 2...

I Made a Storybook with AI... Do You Think It's Perfect?

Feb 22  73  1



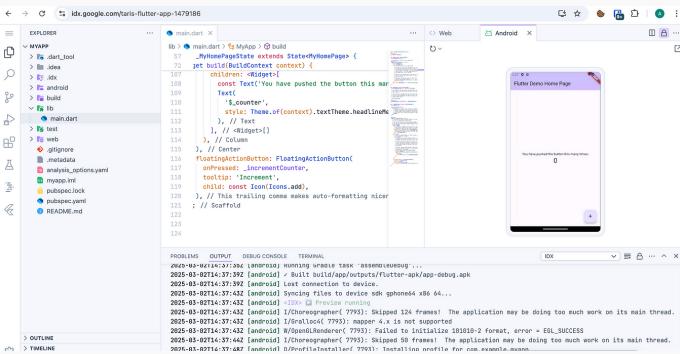
 Matteo Orlandi

### The Full-Stack Alchemist: Building

**use in 2025 as a Staff Software...**

## Tools I use that are cheaper than Netflix

Jan 7 12.1K 300



<CB/> In Coding Beauty by Tari Ibaba

# This new IDE from Google is an absolute game changer

This new IDE from Google is seriously revolutionary.

Mar 11 3.7K 200

# Mobile Apps with Supabase,...

As a frontend developer, I often find myself in situations where I need to visualize an idea,...

Oct 30, 2024



Fatima noor

# What is python? Introduction to python

---

— 1 —