

ZERO TO HERO IN

# TECHNICAL WRITING



# SURAJ VISHWAKARMA

MAKING CONSISTENT INCOME BY WRITING ARTICLES

# Disclaimer

The content of this eBook is intended for informational purposes exclusively. The eBook aims to provide you with a guide that will assist you in gaining comprehension of the field of technical writing. The focus is mainly on writing technical articles.

Reproduction, transmission, or sale of any portion of this publication in any format is prohibited without explicit prior written authorization from the author. Any trademarks or registered trademarks mentioned within this guide are the property of their respective owners.

By reading this eBook, you agree that I and/or my company is not responsible for the success or failure of your career/business decisions relating to any information presented in this eBook.

Please note that the modifications above are tailored to the title "Zero to Hero in Technical Writing: Making Consistent Income." If you require any further adjustments or have specific requirements, feel free to let me know.

For more information, contact: [surajvishwakarma625@gmail.com](mailto:surajvishwakarma625@gmail.com)

# Acknowledgments

I am profoundly grateful to the individuals who have played an indispensable role in the realization of this book, "Zero to Hero in Technical Writing: Making Consistent Income."

[Rishabh Singh](#), your feedback throughout the journey of crafting this book has been invaluable. He has helped me in finalizing the name of the ebook. He is a UI/UX designer who also designed the cover of the book.

[Bentil](#), being a technical writer has helped with structure, and recommendations for additional topics have been a guiding light. Your input has not only enhanced the coherence of the book but has also enriched its depth, making it a more comprehensive resource.

To everyone who has contributed their time, expertise, and encouragement, I express my deepest appreciation. This book would not have come to fruition without your unwavering support.

Thank you all for being an integral part of this endeavor.

# About Author

[Suraj Vishwakarma](#) is a full-stack developer and technical writer. He has been writing code since 11th Grade and started writing technical articles the past 3 years. He is publishing articles on [dev.to](#) and its website, [surajondev.com](#).

During his journey, he was able to meet and collaborate with different people and organizations to produce quality and well-optimized blog posts.

His collaboration includes:

- daily.dev
- LogRocket
- MedusaJS
- Documatic
- Accepted as Writer on freeCodeCamp
- And many others

He has published a book before with the title “50+ Awesome Tools for Web Developers”. It has garnered more than 2300 sales with 4.9★.

He is very grateful that you have got this book. He wishes you on your technical writing journey.

# Table of Contents

1. Introduction	5
2. Why should you write technical writing?	7
3. How to start your blog?	10
4. What should you write?	23
5. Researching the topic	26
6. Writing style to follow	29
7. Markdown	33
8. Things to do before publishing the article	38
9. Things to do after publishing the article	40
10. Tools to Use	42
11. Challenges of Writing Technical Articles	51
12. Earning money through Technical Writing	58
13. Last Note	66

# Introduction



Technical Writing is about writing articles, documentation, emails, technical reports, and [other content](#) related to computer hardware and software, architecture, software engineering, and other fields that help in building software. In this book, our focus is on article writing as a Technical Writer.

The profession of technical writing is rooted back to World War 2. But the rise of the profession is equivalent to the rise of the internet, especially the Web2 era. There were tons of websites on the internet. Every website needs content. The ones that are related to software needs someone that can write documentation of tools and article to promote the tool/product. This has created a great opportunity for writers who know about technical stuff. At the same, this was restricted to internal teams and someone with personal websites other.

The second time, the profession has its boost with the rise of platforms like Medium. It provides a place for writers to create and publish articles online with their tools. It won't require you to depend on others or to have your own website. Although Medium was for every kind of writer. During this period, platforms like [dev.to](#), Hashnode, and Hackernoon that focus only on technical writing also have a rise. Now, developers especially prefer these platforms to Medium.

The platforms have made it quite easy to publish technical articles. They have their own audience along with that search engine traffic is also quite good. Writing technical articles has become easy with few things to follow to break into technical writing. We are going to look

into those that you can follow to write articles that reach more developers.

We are in this period, where we have multiple options to write and publish articles online. My objective in creating this book is to help beginner developers break into technical writing. Technical writing can be or can not be your profession, but it is worth a try for every developer. Let's look into the reasons that should interest you in writing technical articles.

# Why should you write technical writing?

I mentioned what technical writing is. You can pursue it as a side hustle or a profession. This choice depends on what best fits you. A developer at the beginning of their career can pursue it as a profession. However, someone who is already writing code and has experience can write articles as a side hustle. Both scenarios will be beneficial. Now let's explore some benefits of it.

## Building Audience

By starting a blog on a platform and on your website, you will start reaching an audience (other developers most likely) through the articles on it. On the platform, they can interact with the posts by giving their reactions and commenting on their thoughts about the articles. They can follow the platform to receive more articles from you. By doing this, you will create an audience for your blog that will read, react, and comment on your blog posts.

## Learning Opportunity

While researching the articles, you will learn a lot about the topic, enhancing your knowledge of the domain. I learned many things just by writing articles. Additionally, the community will interact with your

posts to share their experiences and opinions. You will also learn a lot from other perspectives on the posts.

## Helping Developers

The best way to solidify a piece of knowledge is by sharing it with others. When you write an article about a topic, it will only deepen your knowledge about it. Also, with your knowledge and experience, you can help other developers who are at the stage where you once were. So, it's a win-win situation for both the reader and the creator.

## Generating Income

There are various blogs that pay writers for their articles. Tech startups need technical writers to spread the word about their products. We will discuss such blogs that pay well for writing articles. You can do a part-time/full-time job as a Technical writer. With this point, I just want to highlight that you can generate good income as a side hustler or a job seeker.

## Potential Job Opportunities

A technical writer job is quite valuable. However, someone who wants to explore more opportunities from here can add more skills, such as communication, management, and public speaking. This will open doors for opportunities such as Developer Relations, Developer Advocate, Community Manager, and Content Manager. There will be more such jobs that you can adapt from technical writing by adding more skills to it.

The domain is quite promising. The opportunity is substantial in it, as blogs and startups are always looking for skilled writers to join their platforms. I will not recommend everyone pursue it because some enjoy sharing, while others do not. However, I will definitely recommend everyone to at least give it a try. See how it goes for you. I hope the benefits of technical writing have excited you to start a blog today. But please wait for a moment before commenting on the book, as I will guide you through some processes that are specially tailored for my blogs. Whether you are new or already have a blog, this guide will help you sort out lots of details.

# How to start your blog?

The next big question after the so much benefit of writing technical articles is “How to start?”. Let’s deal with this question. The first thing that comes to your mind is where to put those articles. As it is important to know where you can get eyeballs for your articles. Writing is a Passion that should be followed despite the reach but you need reach. You want your articles to be read. That’s it become important to decide the platform to write. In technical, you have lots of options. But major, there are two options to put your articles online. Let’s look into them:

## Platforms

Platforms are like social media for writers. They are places where creators publish posts, and other users read them. Readers can become creators, and vice versa. You can create and publish your articles on platforms like [Medium](#), [DEV](#), [Hashnode](#), [Hackernoon](#), [Tealfeed](#), and others.

### Pros:

- These platforms have active users who are in search of good articles to read.
- They have high rankings for their websites in Google searches. This makes it easier to rank your articles on Google using these platforms.

- You don't need to worry about managing your own website with various features, especially for a technical-based blog.
- There is no cost to start writing on these platforms.

### **Cons:**

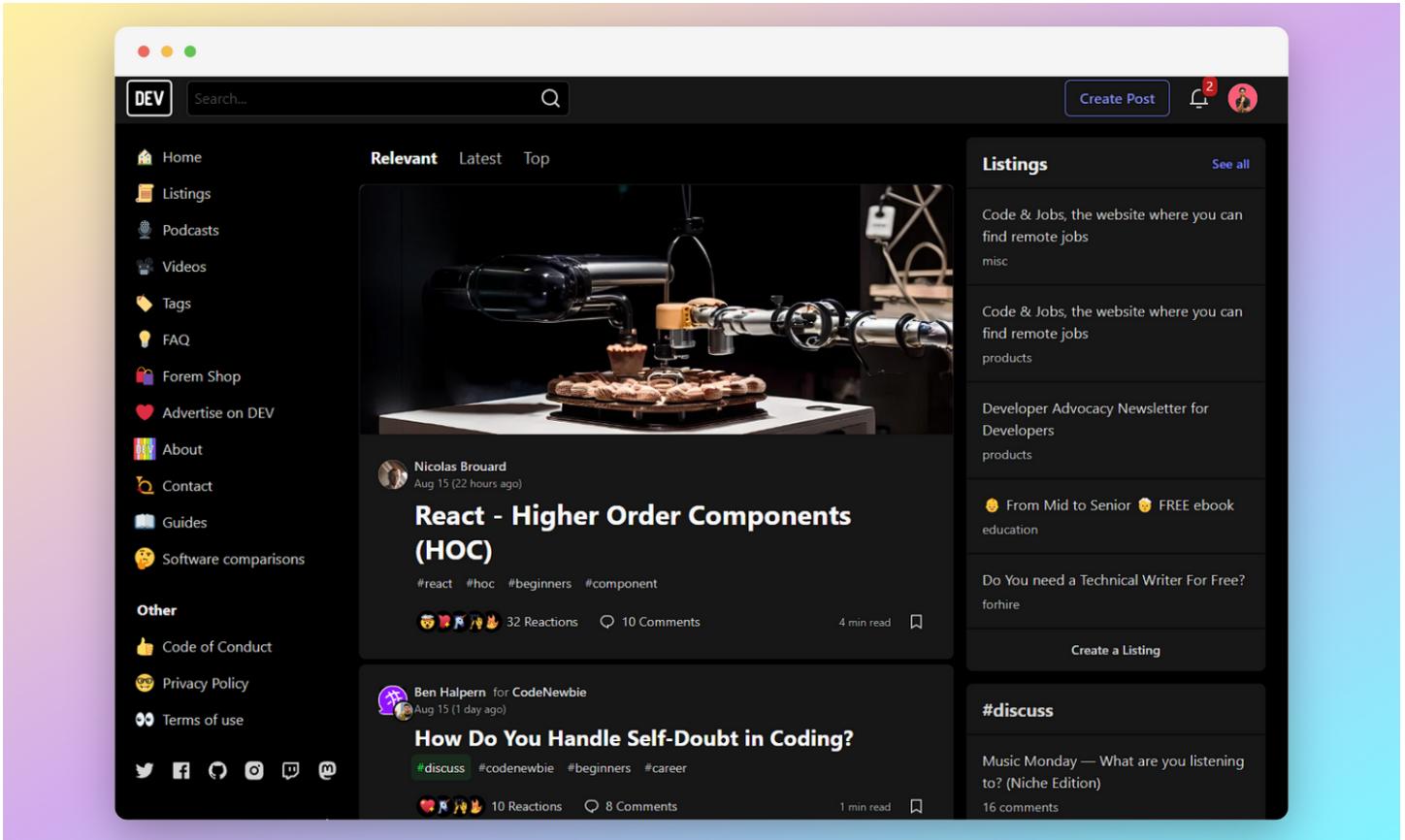
- You become dependent on the platform. You rely on the platform for new features and must comply with its content policies.
- There's a risk of being shadow-banned from the platform for various reasons.

## **Platforms to Write Technical Articles**

Let's list down some of the platforms where you can write technical articles. Some of the below platforms are focused only on technical and some are focused on the general article. The list is from Ascending to Descending terms based on my preference for writing articles. Now, let's start it.

### **Dev.to**

*A constructive and inclusive social network for software developers. With you every step of your journey.*

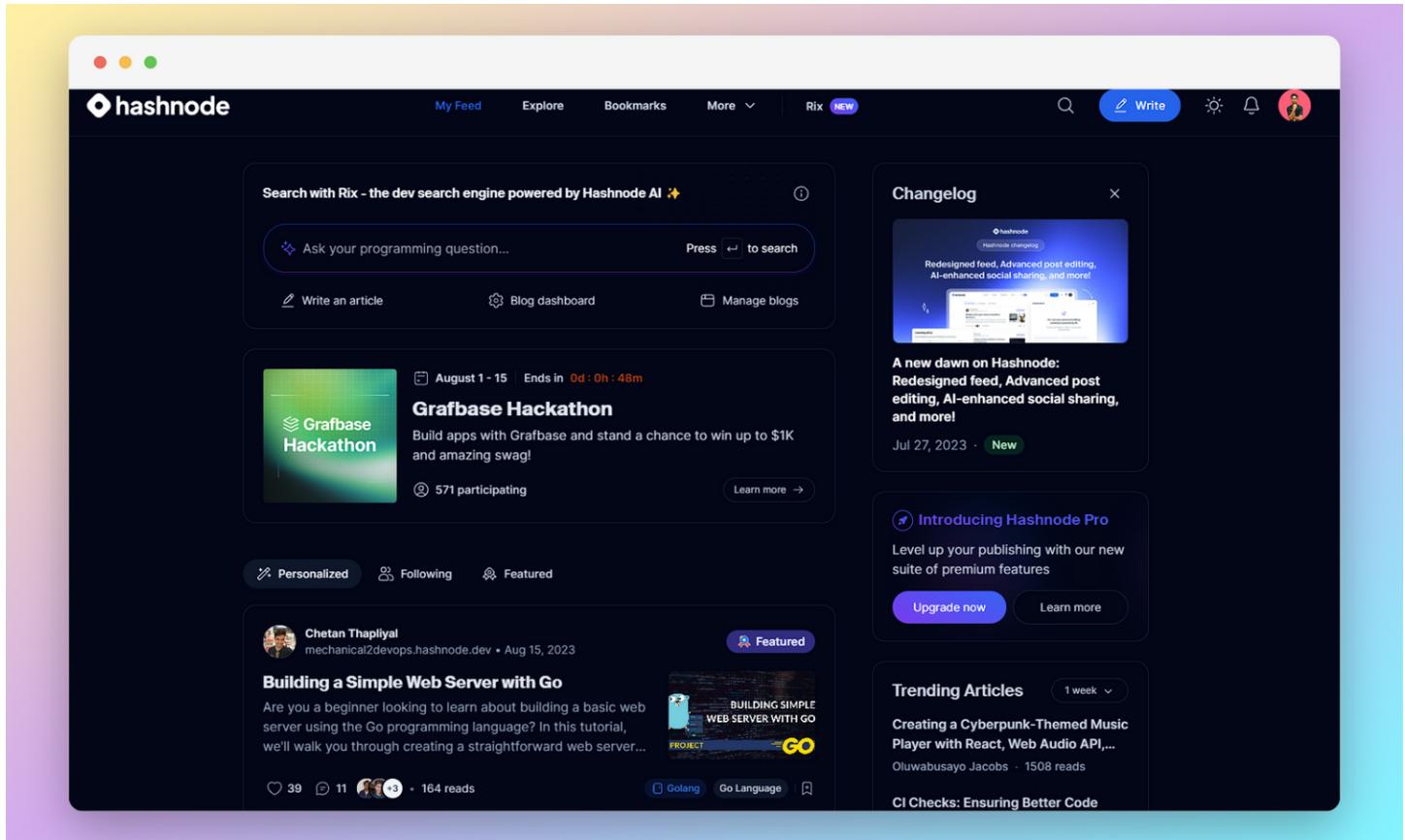


The best platform I have used for writing technical articles. From my initial days, I am on the platform. The best part is that your first article can give you tons of reach even when you don't have any followers. On the platform number of followers doesn't matter a lot, the content matter on [dev.to](#) the platform.

There are lots of improvements on the platform. The UI is simple making most of the articles getting discovered. This was the first platform where I wrote my first technical article. In my third article, I was able to attract 200+ reactions to the post. So, its algorithm is quite the best fitting even for beginners.

## Hashnode

*Start a blog for free instantly and share your ideas with people in tech, developers, and engineers. Hashnode is a free blogging platform.*



The second platform that comes to my mind for writing technical articles is Hashnode. They have a pretty good-looking UI. You can connect your domain or use their sub-domain for writing articles. The UI and SSL are provided by the Hashnode. With your own domain, you can have more control over the UI and feel of the blog platform. But still, you can't fully customize the website.

The reason for being on second is that it is a little bit hard to get reach the platform as a beginner. Your article has to be exceptionally great to

be featured on the top of the blog and lead to more reach. The UI looks good but pinning features posted at the top hampered the reach of other articles. That's what I think of having a low reach for beginner writers on the platform.

In spite of such things, You can give it a try for writing articles as I have seen some awesome reach for some creators.

---

## Medium

*Medium is an open platform where readers find dynamic thinking, and where expert and undiscovered voices can share their writing on any topic.*

The screenshot shows the Medium 'For you' feed. At the top, there are navigation links: '+', 'For you', 'Following', 'Self Driving Cars', 'Privacy', 'Gadgets', 'Future', and 'Digital Life'. Below this, the first article is by Dmitry Kruglov in 'Better Programming' (Updated Nov 7, 2022), titled 'The Architecture of a Modern Startup'. It discusses the trade-off between pragmatism and speed. The second article is by Adam Dannaway in 'UX Planet' (Mar 14), titled '16 little UI design rules that make a big impact', which is a UI design case study. The third article is by Prashant Gupta (May 14), titled 'Next.js 13.4: A Game-Changer for Front-End Development!', which covers the launch of Next.js 13.4. On the right side, there is a 'Staff Picks' section featuring articles by Megan Finley Horowitz, Chris Messina, and Anna Mercury. Below that is a 'Recommended topics' section with categories like 'Visual Design', 'Product Management', 'Book Excerpts', 'Software Development', 'Entrepreneurship', and 'Web3'. There is also a 'Who to follow' section featuring a profile for Zoe Chew.

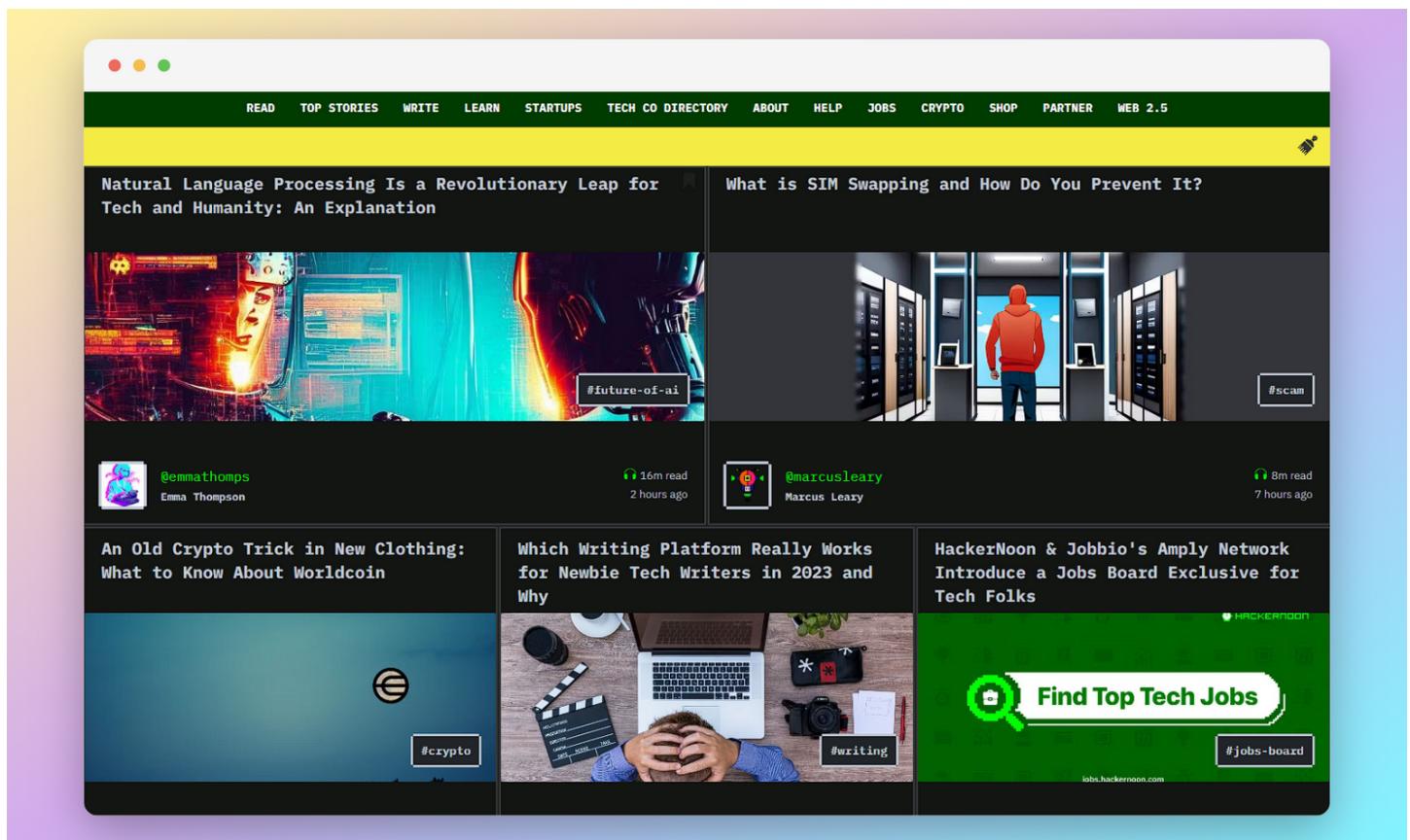
Medium is the place for putting general articles. You can put your technical articles on it too. It is very easy to write and publish articles on the platform.

I had given it a try in my initial days but couldn't continue on it. It does not support markdown(We will talk about it later). But I have seen some awesome growth on Medium for technical articles. So, I would recommend you to explore the platform.

---

## Hackernoon

*How hackers start their afternoon. HackerNoon is a free platform with 25k+ contributing writers. 100M+ humans have visited HackerNoon to learn about technology*

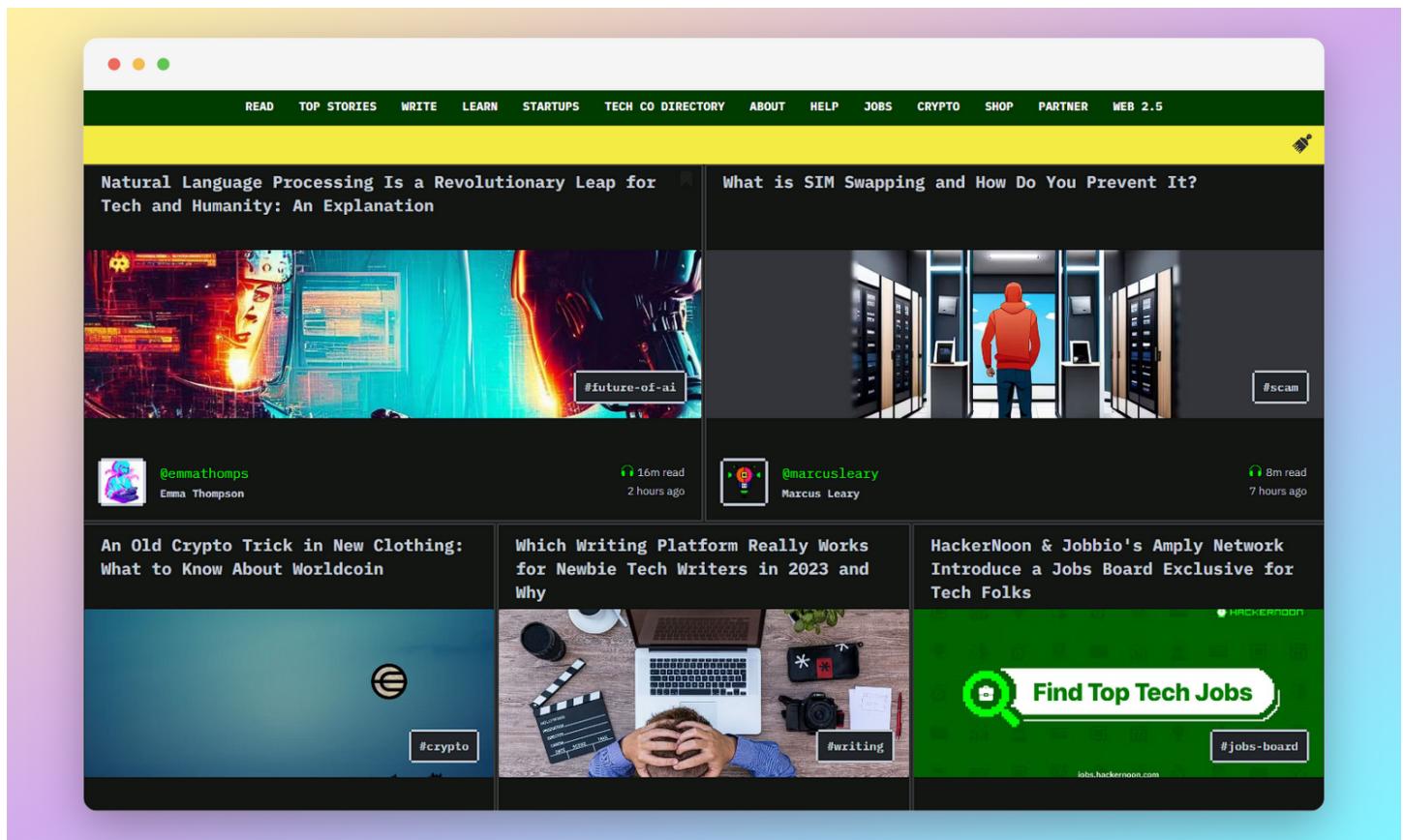


This platform is focused on technical writing. I have seen the platform's article ranking well on search engines. I haven't tried it for writing articles but I will recommend you explore it further. Find if it fits you or not.

---

## Tealfeed

*Tealfeed is a knowledge-sharing platform that keeps users feed informative, connects people of similar interests, and empowers creators.*



Tealfeed is a new platform that serves as a general blogging platform. I have been in contact with their CEO regarding the future of the tealfeed and it looks awesome.

I also publish my articles on this platform. You can get some good reach through their platform and Google search engine.

## Own Website

We have looked at a lot about publishing articles on platforms. Now it's time to look into the second option which is to build your own website or use any CMS(Content Management System) provider to manage the building, hosting, and features of the website.

### Pros:

- Having your own website will give you the power to customize as per your liking as you will not be dependent on any platform.
- You can add any features to your website without any dependence.
- Customize UI as per your choice.

### Cons:

- It is costlier as you have to buy a subscription to a CMS, hosting, and other plans
- You have to build the SEO of your website from scratch.

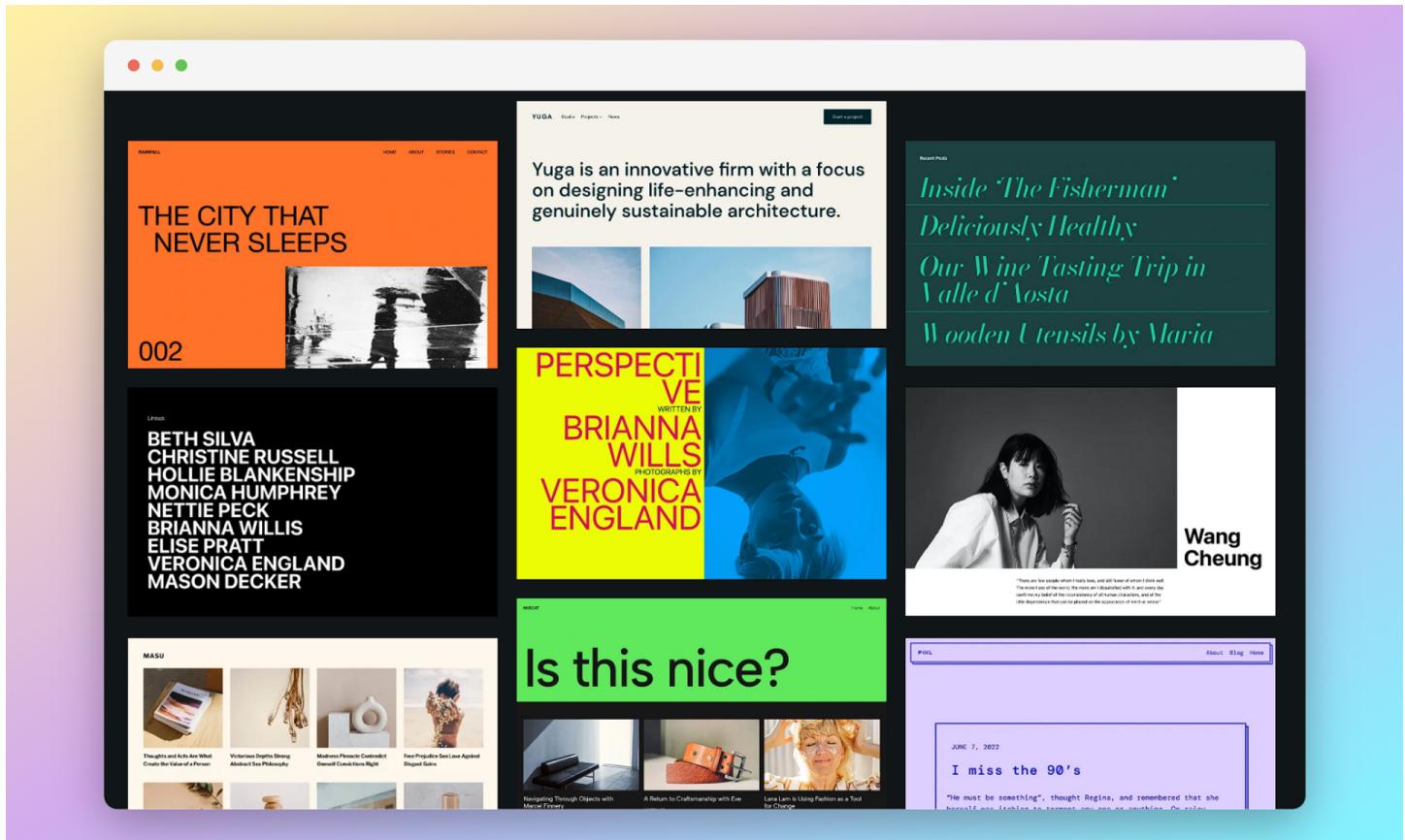
## CMS Website to Write Technical Articles

Now, it's time to list down some CMS websites that can help you in hosting and managing your website. I will mention if it has any free tier under the sub-domain. Let's get started with the *list*.

# Wordpres.com

**Plan:** Free

*Create a free website or build a blog with ease on [WordPress.com](https://www.wordpress.com). Dozens of free, customizable, mobile-ready designs and themes. Free hosting and support.*



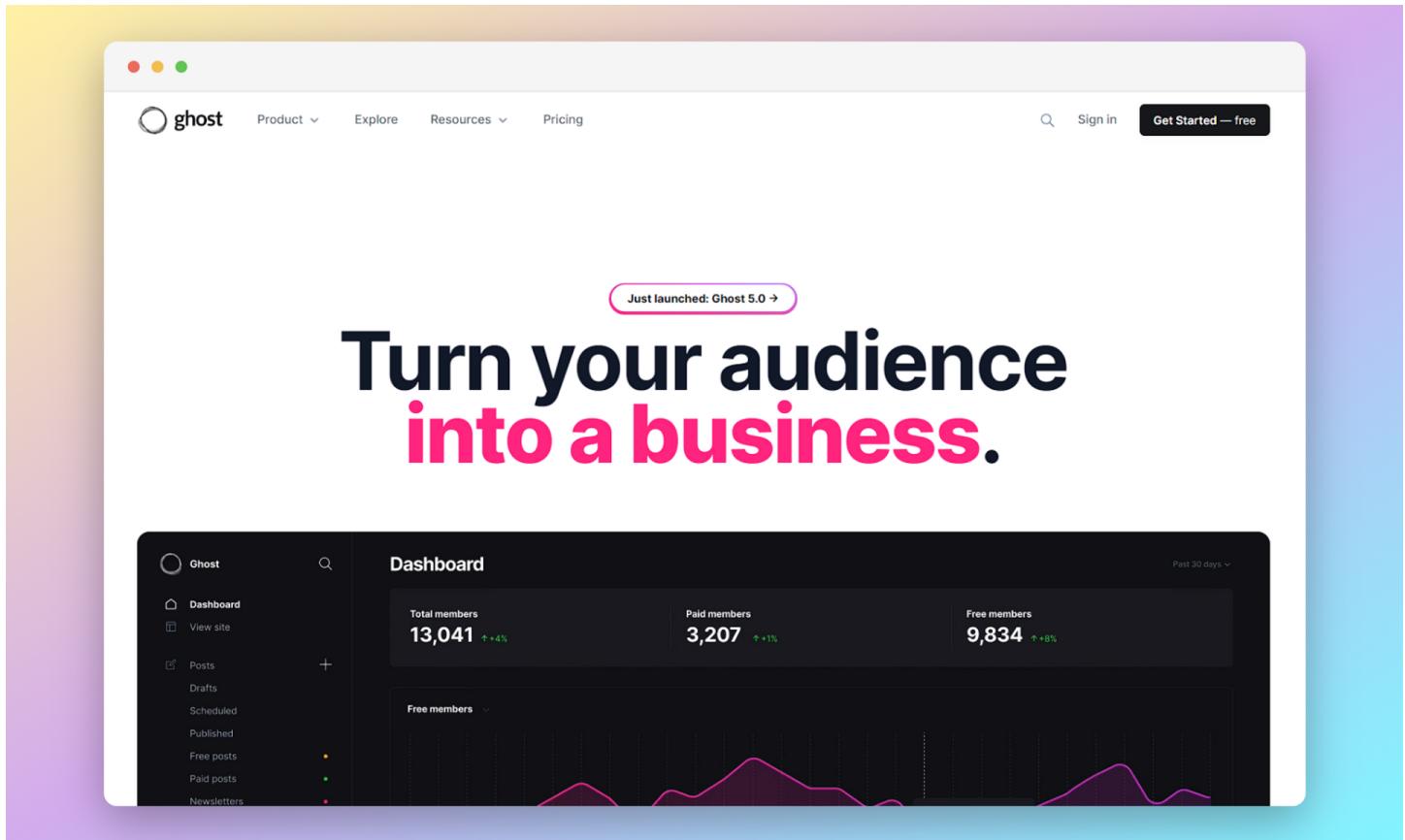
My website, [surajondev.com](http://surajondev.com) is hosted and managed through WordPress. I have been using the platform for publishing my article. It is quite popular and there are a variety of features. A few of them can be a free plan under a sub-domain, analytics features, support for lots of plugins, and more.

The one thing that I don't like is the theme. They are limited and are not mobile-friendly. But the support other than it has is phenomenal.

## Ghost.org

**Plan:** Paid(14 days free plan)

*The world's most popular modern publishing platform for creating a new media platform. Used by Apple, SkyNews, Buffer, OpenAI, and thousands more.*



Another interesting CMS website is [Ghost.org](#). freeCodeCamp uses ghosts to publish articles. It's good if you ask me. The UI is clean, there are options for plugging, analytics is awesome and much more. You can run a newsletter using the platform.

The only thing that drags it is they don't have any free plan. You can still get a 14-day trial. I will recommend you use [Ghost.org](#) if you have finance with you.

I have listed down my view about these two ways of putting your article on the internet. But this kind of comparison creates more chaos as you will lose something if you chose between the two. But what if I told you that you can choose both? Here in writing an article, you can choose both. You will get reach from the platform and independence for your blog. This can be done by publishing your article on both domains. Let me guide you on this.

## **Posting on the platform and own website**

This is my way of posting articles online. I publish my articles on platforms such as [DEV.to](#), Hashnode, and Tealfeed. Along with that, I have my own website that is [surajondev.com](#) for posting those articles. Let's learn more about this method in question-answer format.

### **How to manage the SEO of the article as it will publish on multiple websites?**

Most of the writing platform has an input for canonical URL while posting the article. A canonical URL points to the webpage that is the original copy of the article/webpage. By adding the URL of the article that you post on your website on the platform, you will tell the search engine that the URL mentioned in the canonical URL is the original copy and the article on the platform is a duplicate of that.

## Post options

### Canonical URL

Change meta tag `canonical_url` if this post was first published elsewhere (like your own blog).

<https://surajondev.com/2022/12/22/6-awesome-ways-to-start-a-website/>

I can't say that the canonical URL will do its job 100% as articles from your platform reach higher rankings on search engines than your website. But this is a good practice while posting articles on multiple platforms.

### What if I don't have the finance to start my own website?

You can take advantage of CMS which provides a free subdomain. I started my website with a subdomain from WordPress. You can also find other websites that offer sub-domain. Initially, my website with a subdomain was called [surajondev.wordpress.com](https://surajondev.wordpress.com). It is still active but redirects the user to [surajonde.com](https://surajonde.com). With this, you will attain most of the features of having your own website.

I will also recommend you upgrade to the full fledge website when you have finance with you.

## **How to migrate from a sub-domain to full fledge website?**

It's easy to migrate when you will use the same CMS. It will redirect to your domain from the sub-domain if someone visits that website. Also, there will not be much loss in the SEO, as I see no decline in traffic after migrating. After migrating, you will also need to change the Canonical URL from each article from the platform.

# What should you write? 🤔

One of the most asked and as well as most searched questions is “What should I write?” while starting to write articles. In my early days as a technical writer, I used to write about topics that I was learning. It was easy as I was already having knowledge about the topic. It also enhances my knowledge as I write and also engage with other developers through my article's comment section. As I grow, I started finding topics that I have to learn and then write. In this case, I was able to explore more topics.

You can take a look into the following methods to find topics to write an article on it.

## Write about what you are learning

My first recommendation whenever someone asked me about this is to write what am learning. In the beginning, we learn a lot about any tech stack/framework/tool. You can convert that into a blog by adding your point and way of explaining things.

This allows you to gain more knowledge about the topic and you will retain it for a longer time. Also, you will be able to hear from the community about your learning and points to add to your note.

# Social Media Post

You can take inspiration from Twitter, LinkedIn, and Instagram posts for the topic idea. Don't copy all the information from the post. If you have anything more to say or you want to add your points about that topic then write an article.

While taking inspiration, you should give credit to the original source.

## Convert the topic into series

It's not about finding topics but generating multiple topics from one. For instance, you are writing about the NextJS .env variable. You can create a series of NextJS to discuss NextJS. This will lead to having many topics from just one domain.

I avoid breaking a tutorial article into multiple series as some articles I find on the internet, I would like them to find all the information on the same page.

## Maintaining a note about topics

We sometimes encountered topic ideas in our brain but later forget that or lost in the realm of imagination. So, having a topic sheet with the topic ideas will help you in maintaining ideas.

I use a notion template to manage my blog ideas. If you use notion then you can check the template here:

<https://www prototion com/notion-template/Blog-Tracker-a4tGHDAESUHHuT9q>

# Tools

I don't use a lot of tools to find topic ideas but sometimes I use [AnswerThePublic](#). It is a tool to find what questions are public asking on search engines. Its visual representation and ability to export into an Excel sheet are what I like most about the tool. You can find some good topics to write an article on.

You can also use Google's auto-suggest when searching on the platform. The topics in this are quite relevant.

## AI Tools

There is no way, we cannot leave AI tools from any field. The rise of so many AI tools and the benefit that they can provide you is mind-blowing. You should use AI tools such as ChatGPT, Bard, Bing AI, or any other tool to explore different topics.

You can enter prompts such as "Topics to write an article on X", "Trendy article ideas on X", or anything you would like.

One more thing, [Andrew Baisden](#) has written an article, [100 blog topic ideas for your next article](#), from here you can take some topic ideas.

# Researching the topic



Once you have found the perfect topic to write an article on, the next step is to research it in terms of content to write. This should be performed even if you know about the topics as you can learn more about the topic. This will help you in better structuring and writing the article. You can look into the following methods to do research on the topic:

## Documentation

You should look into the documentation of the tools/library/framework that you want to write. You should get knowledge from the documentation first as it is the original source of information. You can get lots of ideas about the format and structure of the article through documentation.

## Reading Articles from other developers

You can take a look at the articles that are related to the article to know about how others are writing and the topics that they are covering. You can find inconsistencies or missing topics from their article so that you can add your own. You can also quote their article in your own articles if you find some interesting information from their articles.

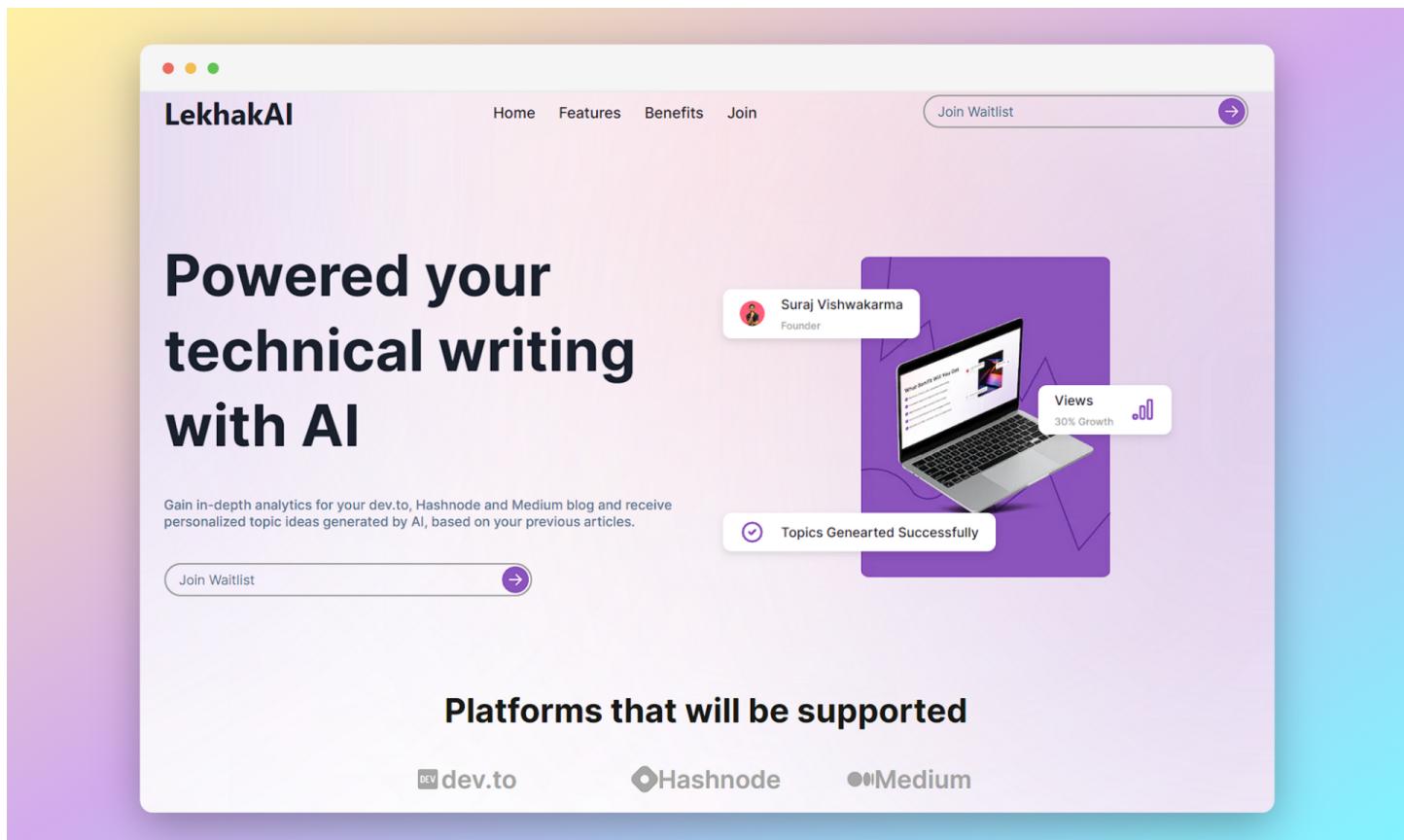
# Writing Code

If you are creating a tutorial-based article such as “Building Music Player in React” then you should first create the project by yourself. In those processes, you will learn what challenges and difficulties you are facing. You can talk about that in your blog. Also, by writing the whole code you can provide the codebase to the readers of the article.

## AI Tools

You can search further about the article through different tools such as ChatGPT and Bard. This will give more information in a simpler format that can help you further extend your articles.

To help a technical writer to do well research on topics such as finding article ideas, keywords, topics-include in articles, and other article-related things, I am launching a [LekhakAI](#). It is currently under production. But will be live soon, you can sign up for the waitlist.



These methods are only for researching and gaining knowledge on the topic. You should never ever try to copy or use the other's work with just a few modifications. Whenever you want to mention another article's quotes, make sure to give a proper quotation.

I would also not recommend you copy-pasting stuff from text-generating tools. You should use it just for research on the topic. There are platforms that support articles published directly from text-generating AI tools. You should read their policy on this before publishing any such kind of article.

# Writing Style to follow



Technical writing requires a distinct writing style that focuses on clarity, conciseness, and accuracy. As a technical writer, your goal is to communicate complex information in a way that is easily understood by your target audience.

## Use Clear and Precise Language

Technical writing should be free from ambiguity and jargon. Use clear, straightforward language to convey your message effectively. Avoid unnecessary technical terms or acronyms unless they are widely understood within your specific industry or audience.

## Organize Information Logically

Present information in a logical and structured manner. Divide the complete article into different headings, sub-headings, and bullet points for better understanding. I see a lot of articles without headings and well structure. It is quite hard for the reader to find a useful section.

## Giving a gist of the article in the introduction

In the introduction section of the article, you can start with the reason for writing the article. You can also mention what your article will help with. This can hook the reader in the article. You should also mention

the topics that you are going to cover. This will set the mind of the readers with the information that they are going to get. So that they don't set any unrealistic expectations from the article.

## **Use code blocks effectively**

As a technical writer, you will need to add code blocks to your article. Always mention the programming language to code blocks for proper syntax highlighting. In markdown, you can add this after adding "```" and then the name of the programming language. For instance, ```javascript for a code block of JavaScript.

## **Visualize Information**

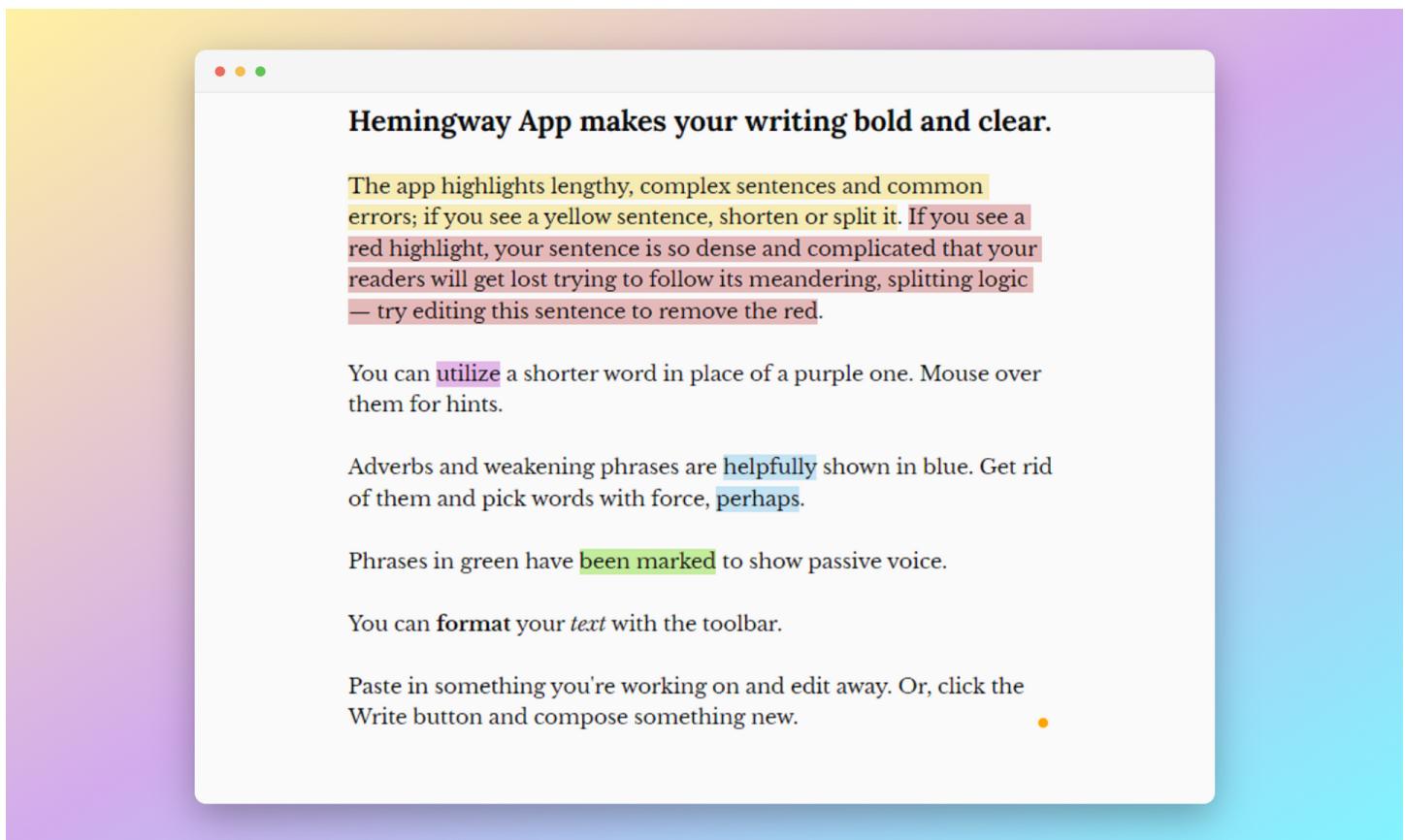
Complex technical concepts can often be better understood through visuals. Incorporate diagrams, charts, tables, and illustrations to supplement your text. Visual aids help readers grasp information more easily and enhance the overall clarity of your content. Try to add relevant images and screenshots to the article. Add proper captions to these images for accessibility purposes. Adding images will also increase the scrollable area of your article.

## **Provide Context and Background**

Always consider the knowledge level of your audience. Provide sufficient context and background information to ensure readers can follow along, even if they are not experts in the subject matter. You can add links to keywords to external articles so that readers can learn about it more. Strike a balance between assuming too much prior knowledge and over-explaining basic concepts.

# Proofread and Edit

Technical writing demands accuracy and precision. Pay attention to grammar, punctuation, and spelling. Use appropriate style guides and follow any established company or industry-specific guidelines. Proofread your work thoroughly to eliminate errors and ensure the highest quality of your writing. These can be achieved by reading the article once it is completed. This will let you in finding many errors that you miss while writing the article. Work on those errors. You can also use Grammarly for reducing grammar errors. And use [hemingwayeditor](#) for chopping lengthy sentences.



# Making your article more beginner-friendly

I see lots of articles that are straightforward without explaining much of the code. I can understand sometimes it's hard to explain everything. But at least, you should give links to other resources for further explanation. Making your articles beginner-friendly by explaining things from a perspective that might not know my keywords. This will also reach more developers.

As a technical writer, your primary focus should be on providing accurate, concise, and easily understandable information. By adopting a clear, organized, and reader-centric writing style, you can effectively communicate complex technical concepts to your intended audience.

# Markdown



I have mentioned many a time about the markdown in the book. Let's look into this.

*Markdown can simply define as a markup language that is used for formatting and structuring plain text documents.*

For developers, it is easy to follow as it is a few syntaxes for different text formatting that can be added in a simple text document. The extension of markdown is `.md`. Platforms like Hashnode and DevTo support markdown formatting of the blog. It also gives a visual tool to format the article.

Let's look into some common syntax for different tags while formatting articles with Markdown. Along with that, I will give some tips on how to properly use those.

## Heading

You can create headings using hash `#` symbols. There are six levels of headings you can create, each represented by a different number of hash symbols.

## Syntax:

```
# Heading 1  
## Heading 2  
### Heading 3  
#### Heading 4  
##### Heading 5  
##### Heading 6
```

I would suggest using heading 2 for your main headings. As heading 1 results in the larger text when rendering. Use heading 3 for sub-headings.

## Basic text formatting

You can basic formatting to text such as bold, italic, and strikethrough.

```
**Bold** // adding bold  
*Italic* // adding italic  
~~Strikethrough~~ // adding strikethrough
```

Use bold for grabbing the reader's attention to a particular word. Italics can be used for quotes.

# Bullet points

Bullets can be used to provide a listing.

## Syntax:

```
- Main Bullet  
    - Sub Bullet
```

Use these to mention topics that you will cover in the introduction.

# Image

You can add images by providing the caption and URL of the image.

## Syntax:

```
![Image description](Image URL)
```

Always use the caption to describe the image. On platforms, DevTo and Hashnode, you can upload images by drag-n-drop and uploads. It provides the syntax like already having the URL of the image. You just need to provide the caption.

# Link

You can provide links to any text using similar syntax as for images with a little modification. You just need to remove the `!` exclamatory sign from the syntax.

## Syntax:

```
[Link Name](Link URL)
```

You should provide links to docs and further topics that users can look at after reading the article.

## Code

I do mention the code block tag a little bit in the writing section. It's one of the necessary tag to know as a technical writer.

### Syntax:

```
```Name of the programming language  
// Your code goes here
```

You should mention the name of the programming language for syntax highlighting.

## Quote

It's one of those tags that you might not need in every article but still require in some.

### Syntax:

```
> Your Quote
```

This should be used to quote others' work. You can also use this to add notes for the readers. Notes can be additional information on any topic.

## Embeds

Embeds is an external link that renders its content in the article. You can add Codepen, Codesandbox, GitHub repository, etc. The syntax and URL support are quite specific to the platform. You can check [DevTO embed](#) and [Hashnode embed](#) for more about it.

# Things to do before publishing the article



## Generating meaningful titles and cover image

The title and cover image will attract an audience to your blog. So, use them wisely. You can frame your article that creates curiosity. Instead of "Building an application", you can use "How I build an application". For the tooling article, you can use the number in your title. For instance, 10 GumRoad products for Developers and 40+ Useful Resources for Mastering the Web.

As for the cover image, you should include both an image and the second title of your article. For the second title, you can text generation tools. You can use [canva](#) for creating a template for your article. Be consistent in the design of your thumbnail. Pick a template and just add an image and title for each article. Don't try to create completely different thumbnails for different articles.

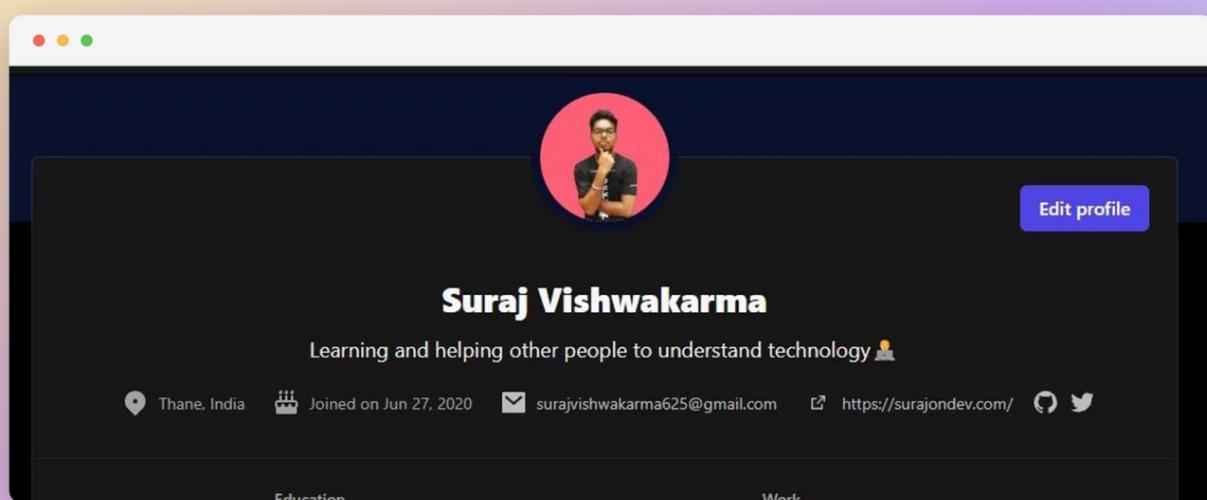
## Adding article details

- **Canonical URL:** If you are republishing the article, add the canonical URL to the article.
- **Tags:** Add relevant tags to the article to show that the article will be shown to the right audience.

- **Publishing Date:** Almost every platform and CSM website supports scheduling. You can schedule the post to go live at your preferred time. As for finding a preferred time, it depends on the audience. As per my posting, I like to post on Monday and Thursday at 5 PM IST. You can use my timing or experiment on your blog to find the perfect time for posting an article for maximum reach.
- **Excerpt:** It is shown as a description of the article when the user explores the article. Add those if the platform supports them.

## Mention your contact info

You should mention your contact such as Email on the platforms where you write articles. In this way, any recruiter can hire you for your technical writing skill.



# Things to do after publishing the article



## Interact with the community

You should interact with the comments that you will get on your post. Make sure it's a healthy comment. You can also interact with a comment on other developers' posts on platforms. Interacting with other developers will help you know more topics and learn furthermore. It also builds a good relationship with other creators on the platform.

## Promote on social media

If you have social media accounts then you can share the link to the article. I would suggest sharing article links on Twitter and LinkedIn will yield more reach. You can also convert your long post into small threads to post on Twitter. In the last, tweet you can give a link to the complete article for more information.

Reddit is also a good platform to share your article. Make sure you follow their guidelines before posting the link to any article.

## Create a publishing plan

After publishing and analyzing a few articles, you can create a publishing plan that includes:

- Types of articles you want to publish
- Frequency of posting articles (Once a week is a good start)
- Day and time of publishing article
- Platforms on which you want to publish

As for what I do here is my publishing plan.

I publish my article on 4 platforms. Here are those:

- [surajondev.com](http://surajondev.com) (My blogging website)
- [dev.to/surajondev](https://dev.to/surajondev) (My [dev.to](https://dev.to) profile)
- surajondev.hashnode.dev (My Hashnode-powered self-hosted blog)
- [tealfeed.com/surajondev](https://tealfeed.com/surajondev) (My tealfeed account)

I aim for publishing two posts weekly. It depends on time, you can also aim for once a week. Don't go below this. Whatever you choose, maintain consistency in writing those articles.

As for the day, I post one article on Monday and another on Thursday. The gap gives me time to write an article. Both days are quite good for publishing articles. For timing, I publish my articles sharp at 5 PM IST as I mentioned before. I feel it's a great time for publishing the article. I scheduled the article as I can't post on so many platforms at the same time.

# Tools to Use



Tools won't help you in improving your writing skills drastically but they can improve your productivity and efficiency in writing articles. I have used a variety of tools in my writing journey but few of them stick with me. I don't use a lot while writing but whatever I used is here.

## LekhakAI

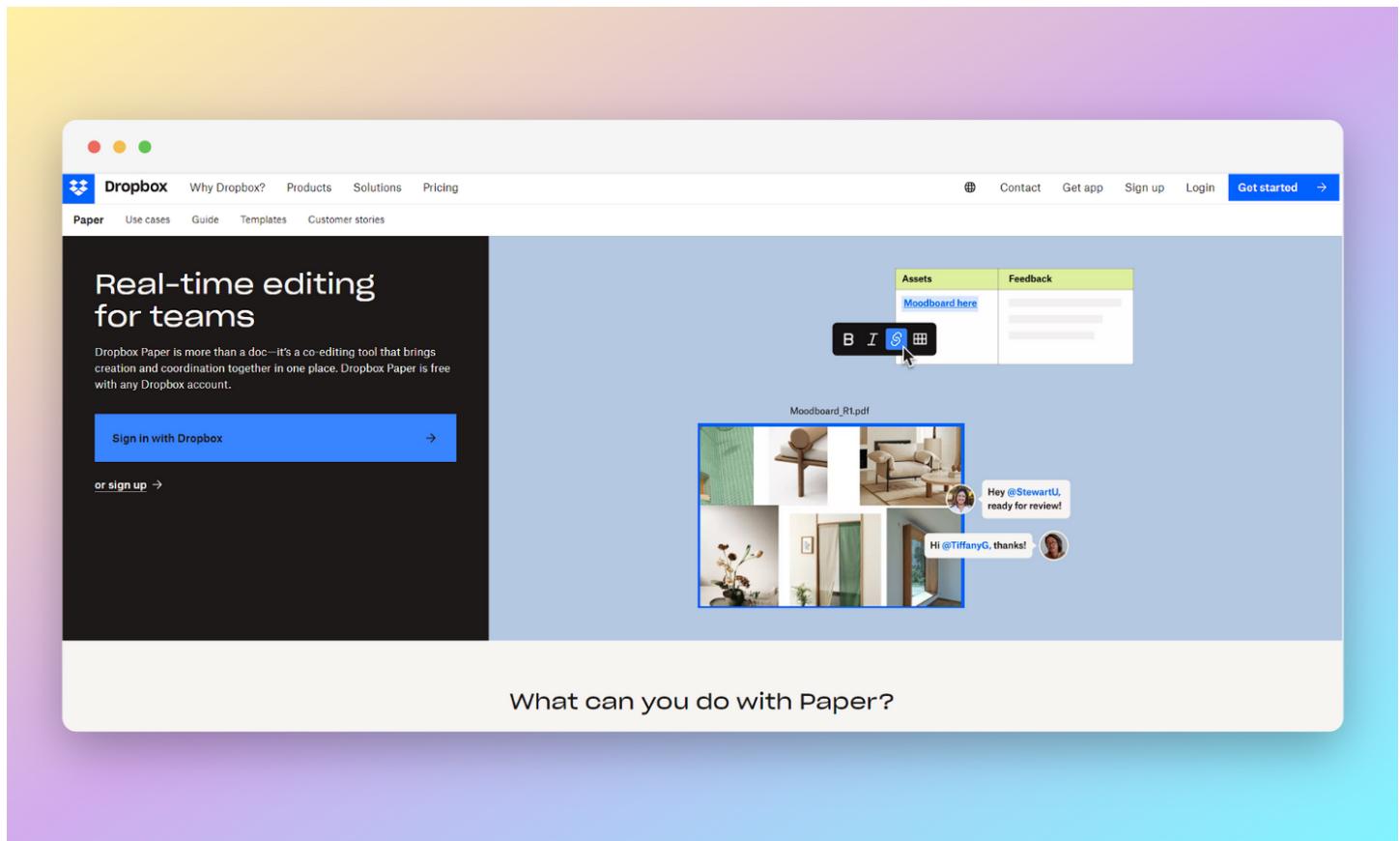
The screenshot shows the homepage of LekhakAI. At the top, there's a navigation bar with 'Home', 'Features', 'Benefits', and 'Join' buttons, along with a 'Join Waitlist' button. Below the navigation, the main headline reads 'Powered your technical writing with AI'. A subtext below it says: 'Gain in-depth analytics for your dev.to, Hashnode and Medium blog and receive personalized topic ideas generated by AI, based on your previous articles.' To the right of the headline, there's a purple graphic featuring a laptop displaying a dashboard with a profile picture of 'Suraj Vishwakarma Founder', a progress bar for 'Views 30% Growth', and a checkmark for 'Topics Generated Successfully'. At the bottom of the page, there's a section titled 'Platforms that will be supported' with icons for 'dev.to', 'Hashnode', and 'Medium'.

As I mentioned about this, I am building a platform that will help technical writers on platforms like DevTo and Hashnode extract more

details about their analytics. It will also help in generating topic ideas that are more likely to get more reach on the platform.

It is currently under development. You can join the waitlist. Once the application is completed, you will get the email to signup for it.

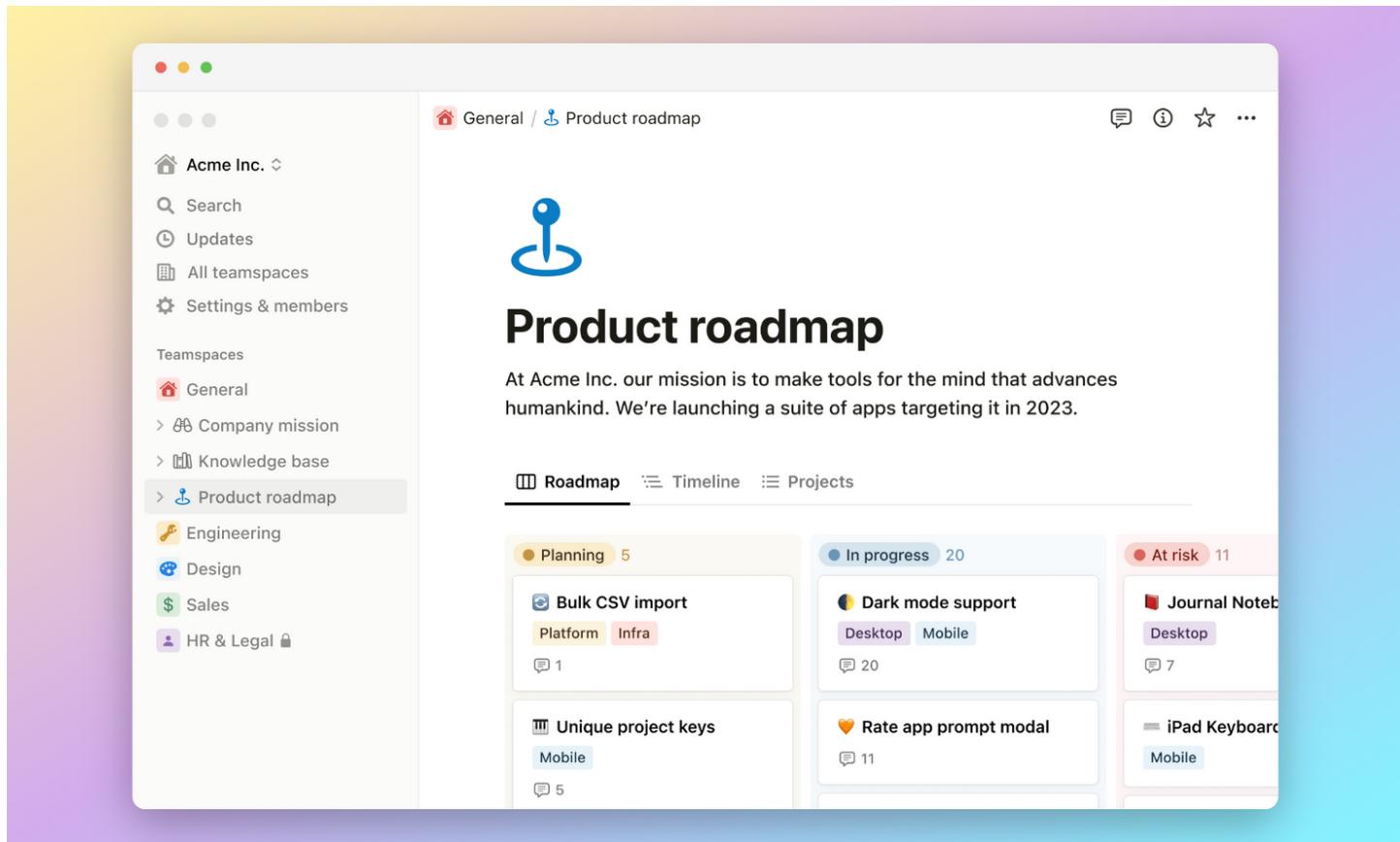
## Dropbox Paper



I use it for writing articles. The platform is very well known in the writing community for sharing and collaborating on writing projects. It has support for markdown, the preferred way of writing text for developers.

Recently, It got support from Grammarly which makes it even more awesome.

# Trello/Notion



Use any of this to manage your topic ideas, upcoming project, or anything related to your writing. Choosing between these depends on what you prefer, Trello has a great board for project management while Notion can do this along with providing other views too.

I would recommend Notion as you can do more on the platform in terms of views, customizing the look, or simplifying managing it.

# Grammarly

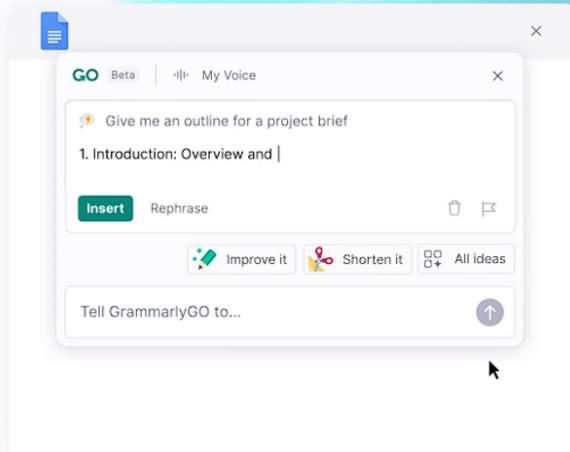
## Personalized AI, Everywhere You Write

Instantly generate clear, compelling writing while maintaining your unique voice.

[Get Grammarly](#) It's free

[Sign up with Google](#)

By signing up, you agree to the [Terms and Conditions](#) and [Privacy Policy](#). California residents, see our [CA Privacy Notice](#).



One app that helps me in solving my grammar errors is Grammarly. You might know it or use it if you write frequently. It's good and helped a lot. For people like me, non-native English speakers, it's an awesome tool.

I used their free plan which just helped with grammar. Premium plans helped in tone and restructuring the sentence. I feel like the premium plan will change your voice and way of writing articles, I suggest using the free plan only.

Sometimes, I also use [Hemingway](#) for cropping lengthy sentences.

# Canva



A place where I design my thumbnail and illustration. They have tons of designs that you can use for free. It's quite easy to customize a design or you can start from scratch.

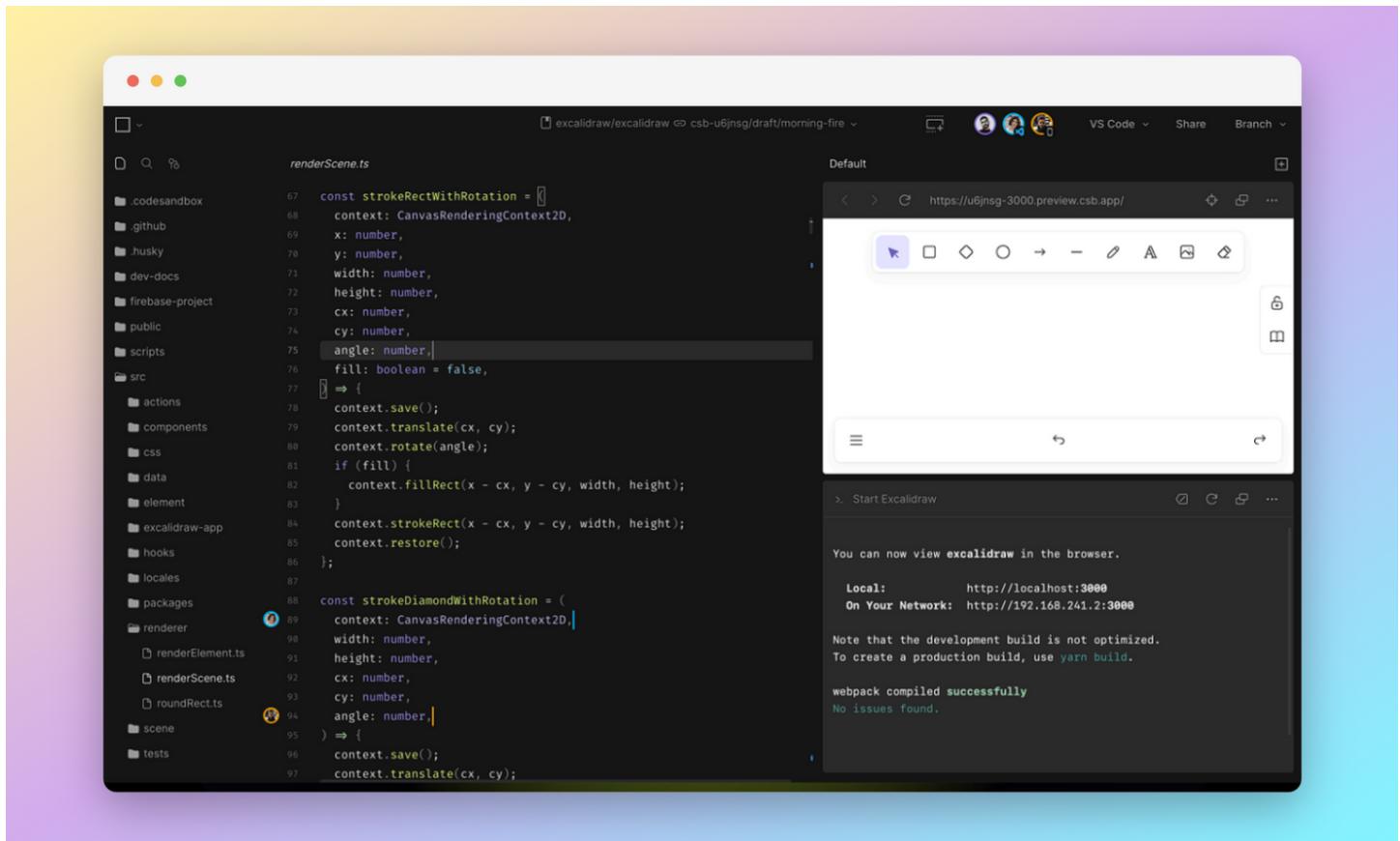
I have used a lot of tools like Canva but nothing is like Canva. The variety of templates is impressive.

# carbon



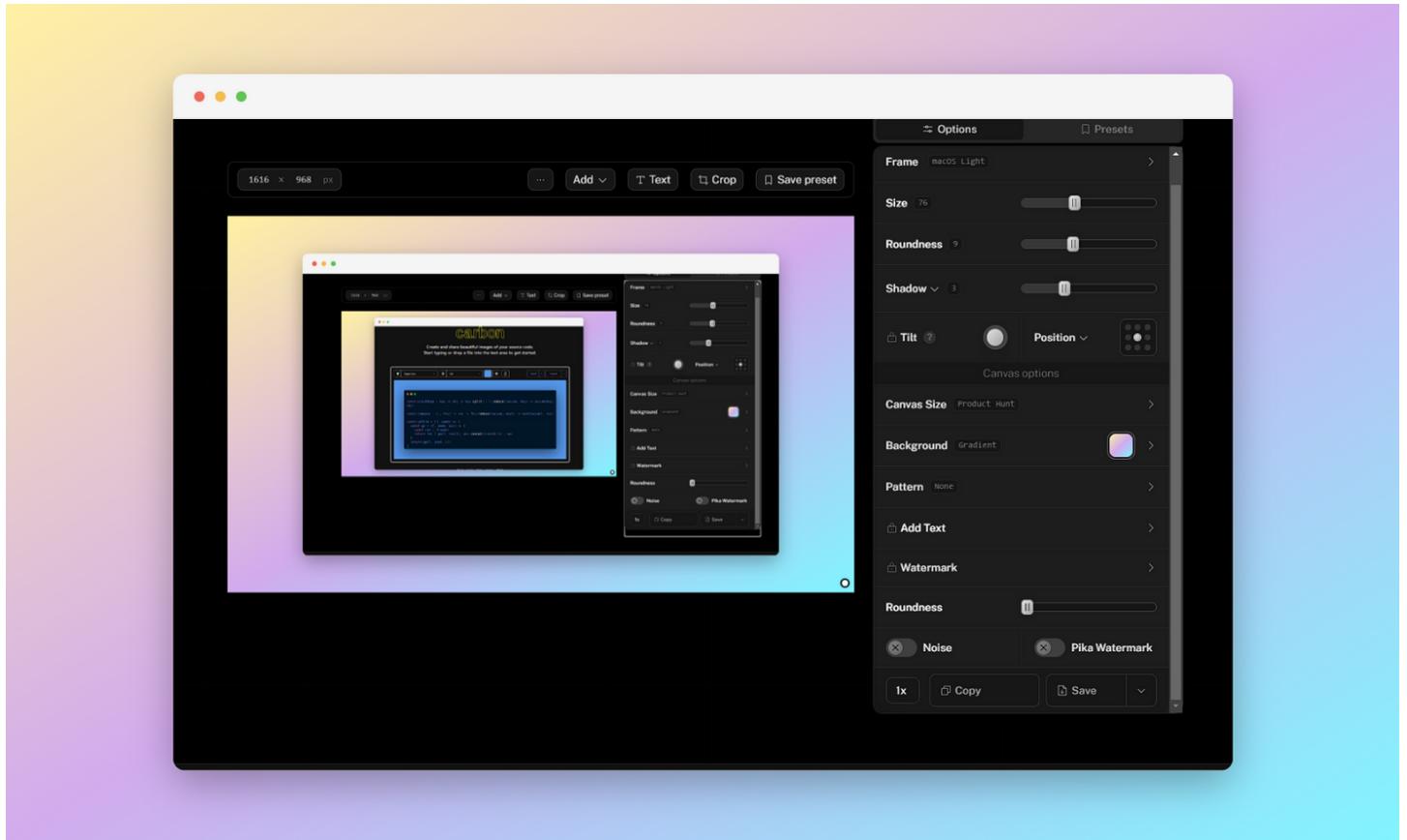
I used this tool to add beautiful images of my code. I prefer, adding code as a text. So, that others can easily copy it and test it. But still, sometimes I used to add code as an Image. I used this when I have to explain syntax or code that is less likely to be copied.

# CodePen/CodeSandBox



Whenever I need to add the complete project with the code to the article, I looked forward to these platforms. They both have awesome embedded support that can view in the article directly. If the project contains only HTML, CSS, and JS then CodePen is preferred. Otherwise, with a complex project with React or NextJS, CodeSanbox is useful.

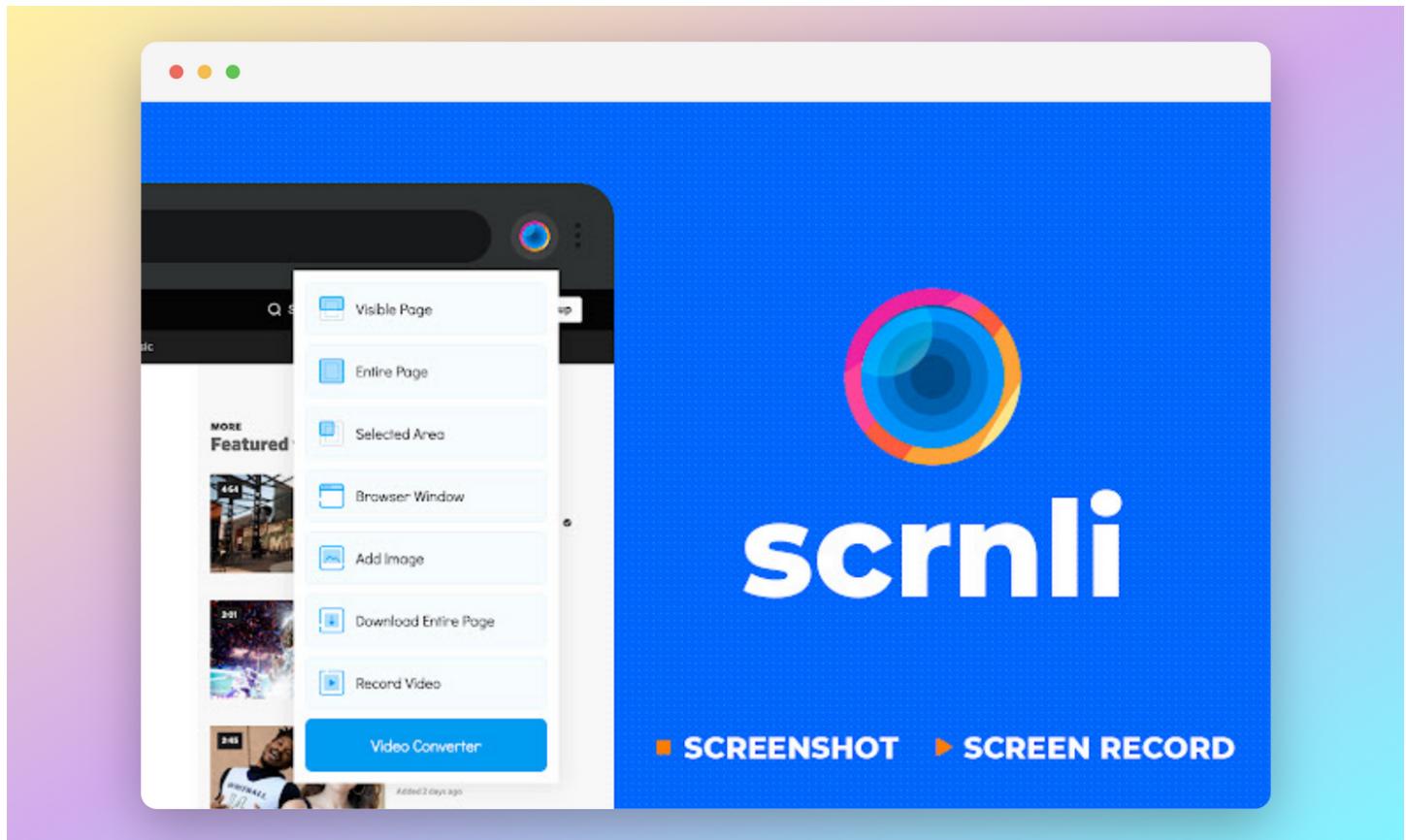
# Pika Style



I recently discovered this tool that provides a beautiful background to screenshots. You can provide a link to capture screenshots too. I use this in tooling articles where I discuss tools for developers. I used Pika Style to capture screenshots of those websites to display in my article.

All these beautiful backgrounds in this section are generated using the Pika style.

# Scrnli



It is a Chrome extension to capture your screen. You can take screenshots of the page with custom size or full you can capture the entire website. You can also use it to record Chrome extensions. All this you can use in your article to explain things visually.

# Challenges of Writing Technical Articles



In this chapter, I would like to discuss some challenges that you might encounter while started writing technical writing. I find myself in this challenge sometimes, I do follow certain methods to overcome those.

So, let's look into those.

## Writer's block

Writing content continuously can drain you but you have to maintain consistency in your writing. There is a time when you have lots of motivation and you love writing content. But there is also a time when you felt less motivated to write content and share it with the world.

The period in which an author felt less motivated in writing which led to creating no or a slowdown in creating content. This is called Writer's Block. I recently felt writer's block which led to creating less and remaining less active on any platform.



There is no particular technique that guarantees the overcome of writer's block. Every person is different and there can be various methods that help people overcome it.

Today, I am going to share my method of overcoming writer's block.

## **Break**

Sometimes you need a break from the work. Taking a break will re-fill your energy and motivation to work again. You can take a walk, chat with your friends or anything. This break can go up to 2-3 days depending on the situation. But make sure, you don't get used to breaks.



I took a break and went to spend quality time with my friends. Spending time with my friends energizes me. We also went for walks.

## Read

Reading articles from other authors will inspire you to write again. It will also keep your mind engaged with other stuff. You can read articles, books, novels, etc. whatever you like.



During this period, I read various articles along with a book. The book I read was Rumours of Spring: A Girlhood in Kashmir. Great book about Kashmir and you connect with the author.

## Write Anything (but less)

One thing that you should do during this period is write anything but break it into small fractions. You do not have to write for hours just write for 5 minutes or even less. Writing little will help you in initializing the motivation for writing. Write whatever you like, you don't have to stick with your work. You can write about your day, things you want, life, your imperfect thoughts, or anything you always wanted to write.



I used to write topics for my articles. Just topics nothing more. I also started writing a summary of the book I had read.

All these things have worked for me and I was able to back at writing.

# **Procrastination & Inconsistency**

Creating content online demands consistency from the creator. You need to have proper consistency in terms of the frequency of publishing articles. If you decide you want to create one every week 1 article then stick to that.

I see people say creating a perfect article even if it takes time but creating a perfect article/content is a myth. Creating content on a consistent basis will eventually lead to better content. With consistent writing, you will improve your writing skill gradually.

Procrastination is one of the reasons that breaks consistency. You can deal with both of them. Let's look into it.

## **Procrastination**

One effective strategy to combat procrastination is to break down your writing tasks into smaller, manageable steps. By setting specific goals for each writing session, you can create a sense of accomplishment as you complete these milestones. Additionally, consider using techniques like the Pomodoro Technique, where you work in focused intervals followed by short breaks. This method can help maintain your focus and gradually build your momentum.

## **Inconsistency**

Maintaining consistency in your writing routine is crucial for steady progress. To tackle inconsistency, establish a dedicated writing schedule. Choose a time of day when you feel most alert and creative, and stick to it as consistently as possible. Create a publishing plan and write according to that. Setting reminders or alarms can help you stay

on track. Remember, it's okay if life occasionally interferes with your schedule; the key is to get back on track as soon as possible.

It's important to acknowledge that both procrastination and inconsistency are natural challenges that writers face. By implementing these strategies and being patient with yourself, you can gradually build habits that minimize their impact on your technical writing journey. Keep your long-term goals in mind and celebrate the small victories along the way. You've got this!

# Earning money through Technical Writing 💰

Technical writing pays quite well, you pursue it as the main source of your income. You can write sponsored on your blog, guest post on their platform, or collaborate with organizations.

There are various ways of making money through technical articles. Let's look into those.

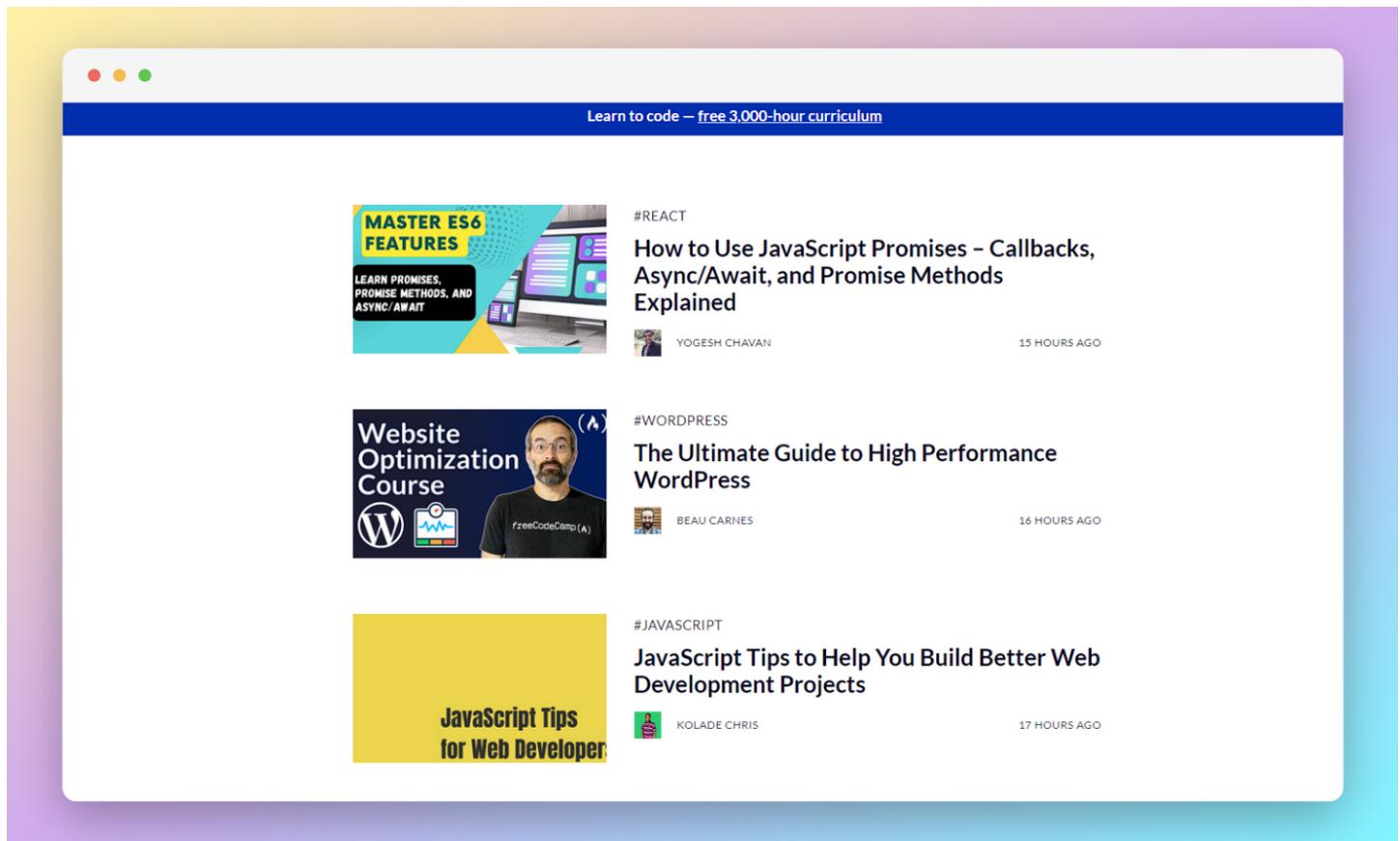
## Paid Collaborative Article

Collaboration is one of the important ways of getting more exposure to technical writing for the cooperative. You can learn a lot through their constructive feedback and style guides for your article.

It can also lend you freelance work too. With my first collaboration with [daily.dev](#), I got the awesome opportunity to work with LogRocket through that article. There are many benefits of such work. You can do this merely for exposure but sometimes they pay you too for your time invested in the article. So, in both situations is a win-win for you. Also, these kinds of collaboration is based on a freelance contract.

Let's look into some of the platforms where you can write collaborated articles.

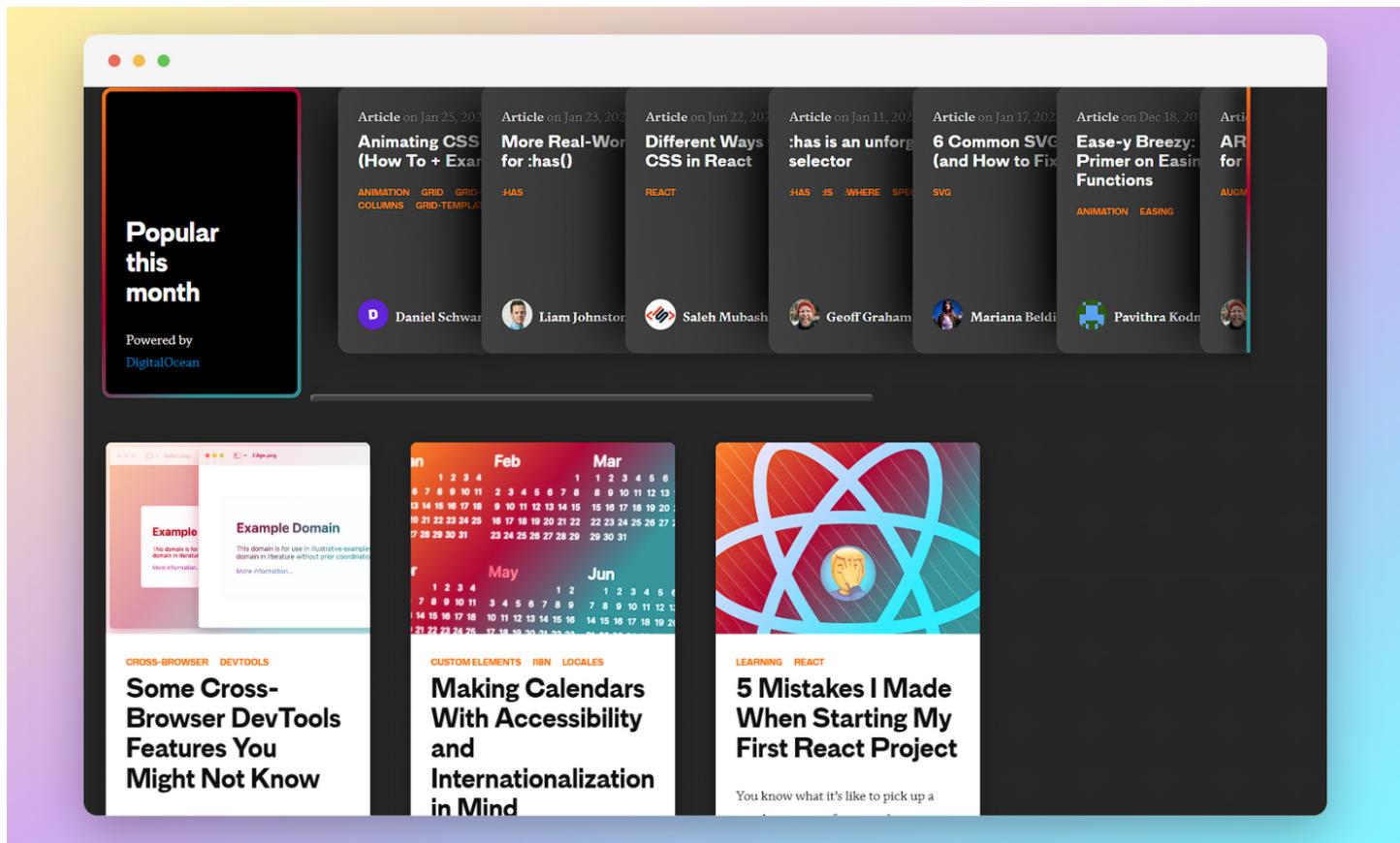
# freeCodeCamp



If you are a software developer, you definitely heard of them. They have a publication where they publish articles in collaboration with other developers. It's not easy to get selected as an author on their website. In the last issue, [I got selected as an Author for freeCodeCamp and How you can do it?](#) I have discussed methods that you can implement to get selected on their platform. If you get selected it is awesome for your portfolio and get amazing exposure.

Read this article to knowing their style guide and form to fill out for their publication: [The freeCodeCamp Publication Style Guide](#)

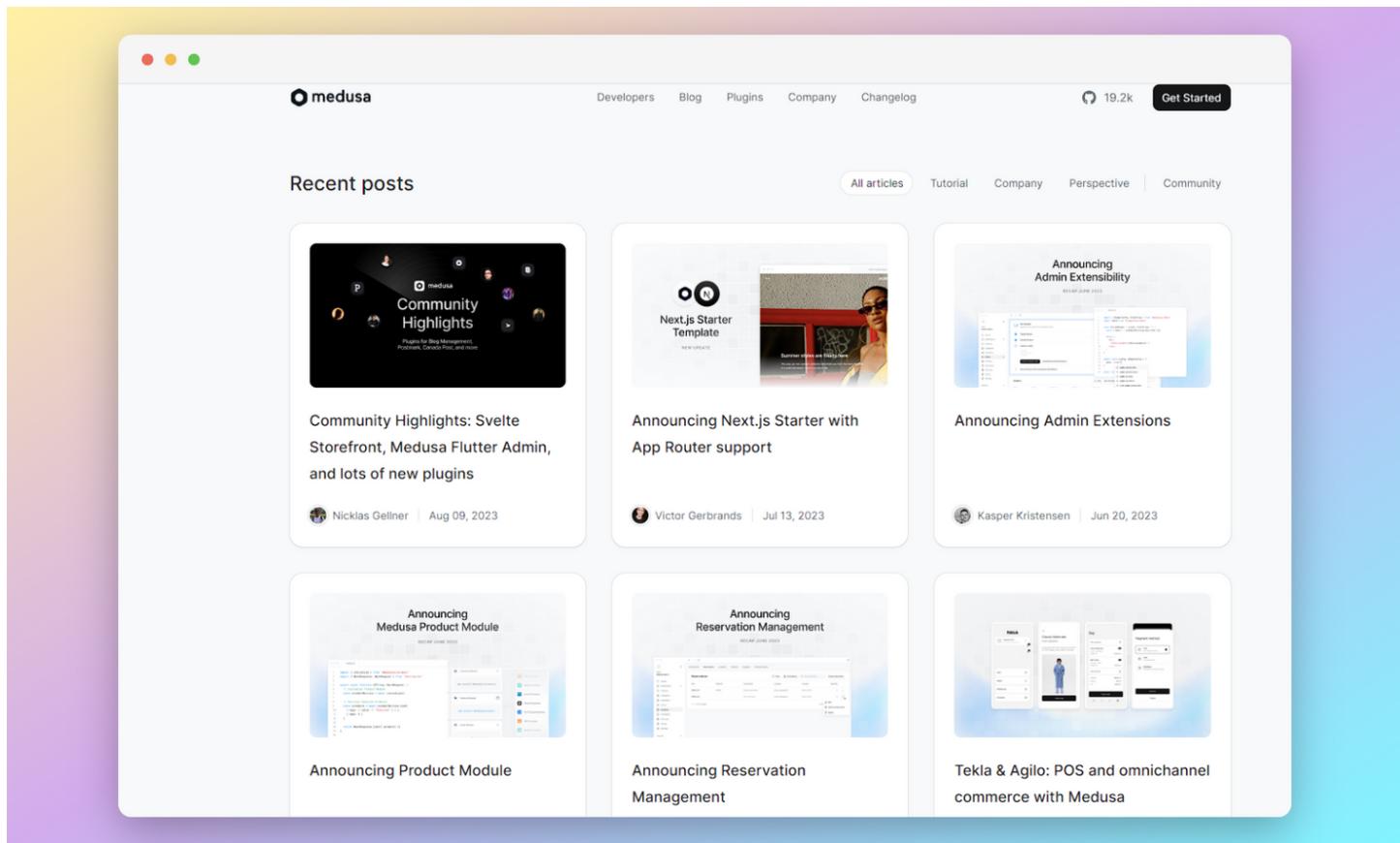
# CSS-Tricks



CSS-tricks is one such publication like freeCodeCamp. While freeCodeCamp is general tech-based blogging and CSS-Tricks is more oriented towards web development, especially the CSS.

It's also an awesome platform with massive reach. You can try writing them by reading their styles and filling out their form from here: [Guest Writing for CSS-Tricks](#).

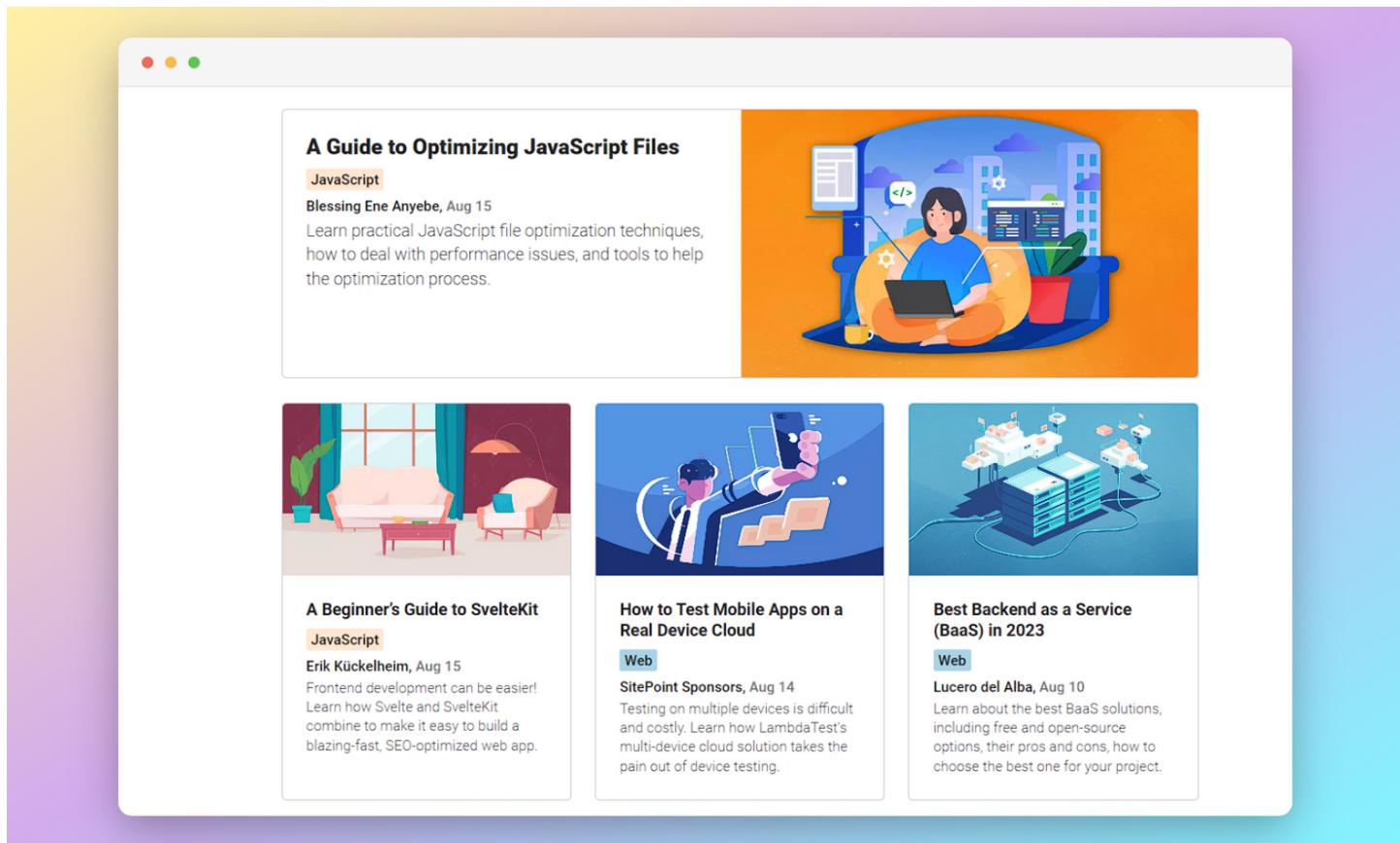
# MedusaJS



Medusa is a headless commerce and an open-source alternative to Shopify. I recently published an article in collaboration with them, [How to Integrate Discord as Notification Provider in Medusa](#).

It was amazing working with them; they learned a lot through their style guide and feedback. You can read this notion template for more details: [Write for Us - Medusa](#).

# SitePoint



Another great platform for writing collaborative technical blog posts. They publish books, courses, and articles for web developers.

Visit this website for learning their style guides and form for the collaborative post: [Write for SitePoint](#).

You can find such kind of website on the internet a tech-based organization that has a blog. Most likely they have some page for guest writing.

# **Cold Emailing for Collaboration**

You can send a cold email/DM to the organization or person that associates with the organization for the paid collaboration. This collaboration can be for writing articles either on your platform or as a guest writer on theirs.

There are chances that this kind of cold emailing can land you freelance work or even a job if they find it helpful.

## **Jobs in Technical Writing**

Due to the rise in startups based on developer-based tooling, they need writers that can write technical articles. The salary they are also quite good. According to Glassdoor, \$85,000/yr is the average salary of a technical writer in the USA.

Mostly there are two types of writing work that are involved in such jobs that are writing technical articles and documentation.

### **Writing Articles**

In this job, you will be asked to write technical articles for their blog. They most likely provide you with the content idea and keywords to use for the article. But sometimes, you have to do it all by yourself. It depends on the organization. This job might also involve editing articles of other writers (most likely articles of guest writers).

## **Documentation**

As the name goes, it involves writing documentation of the organization. You will charge with interacting with developers to write proper documentation for their tools.

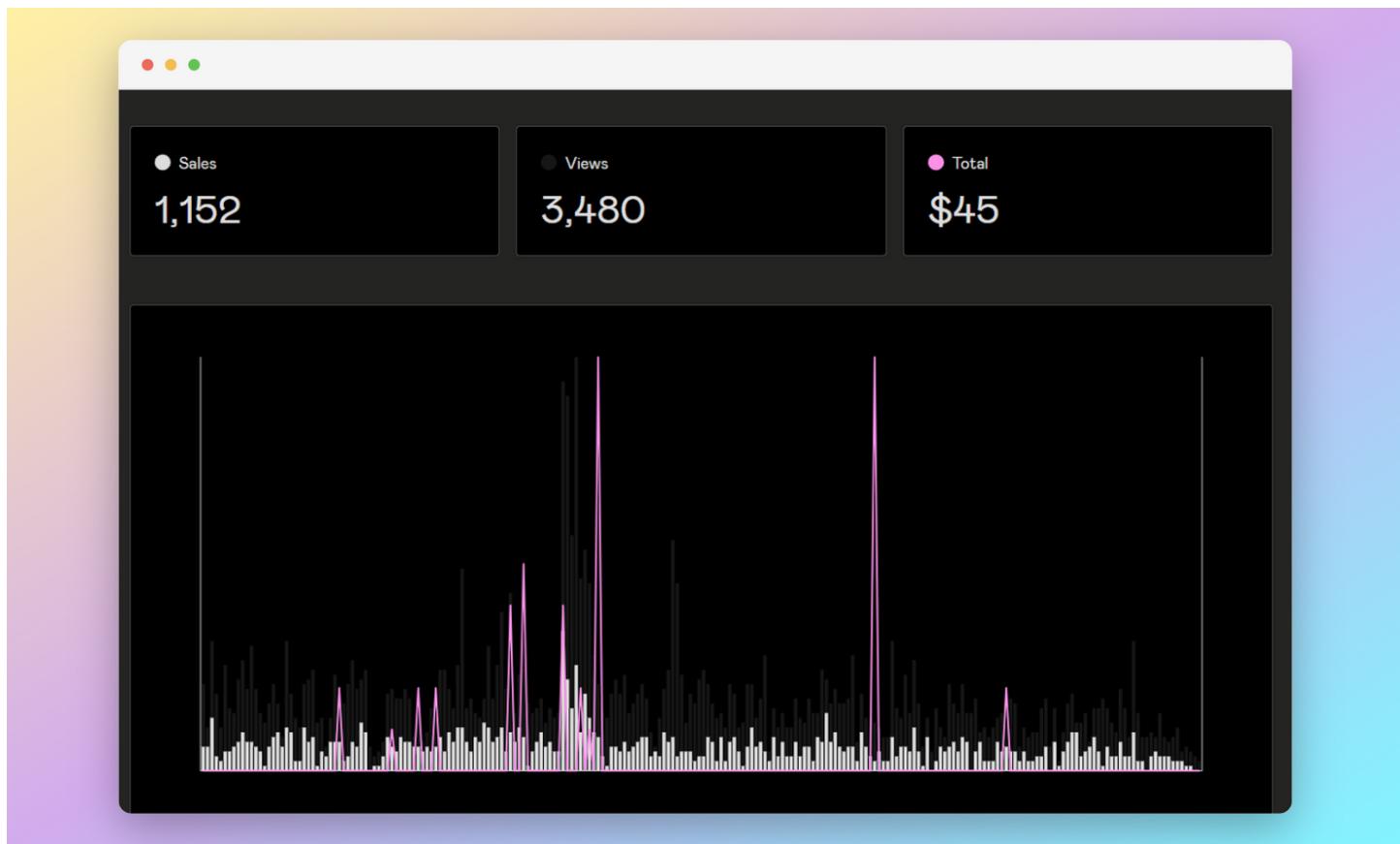
## **Monetizing Article**

Platforms like Medium offer a subscription model where readers pay a monthly fee to access premium content. By placing your articles behind a paywall, you can earn a share of the subscription revenue based on the engagement your articles receive.

One of the significant advantages of the subscription model is the potential for a steady and predictable income stream. As you will not depend on the client.

## **Selling Digital Product**

After building an audience, you can start selling digital products such as the Notion template or GumRoad ebook to your audience and to other people on the internet. You can create and sell Notion templates that showcase their expertise in organizing and presenting information effectively. These templates could cater to various needs, such as project management, content planning, knowledge bases, or personal organization.



You can create comprehensive ebooks that delve into specific technical topics, offering detailed explanations, tutorials, examples, and best practices. One such example of this ebook.

One thing to make sure of is that you should not use your audience to sell mindless things or cheap things. Whatever you sell make sure it provides value to your audience.

# Last Note



Congratulations! You've reached the end of this comprehensive eBook on technical writing for crafting impactful and engaging technical articles. Throughout this journey, we've delved into various facets of the technical writing process, equipping you with the knowledge and tools to excel in the realm of sharing your expertise with a broader audience. We have touched on so many topics that you will encounter on your technical writing journey. Here are some of the topics to remind you on our journey:

- Benefits of writing a technical article
- Platforms and Websites to write articles
- What and How should you write technical articles?
- Understanding Markdown syntax
- Pre Writing and Post writing stuff
- Tools to use
- Problem and challenges
- Methods to earn money through technical writing

Remember, the journey of a technical writer is one of continuous growth. Each article you write, whether it's a tutorial, a deep-dive analysis, or a troubleshooting guide, contributes to your growth as a writer and a professional. Embrace the iterative nature of the craft, seeking ways to refine your skills, adapt to new technologies, and stay abreast of industry trends. Also, make sure you try to be consistent when publishing articles, it is hard but you should try it.

Whether you choose to pursue technical writing as a full-fledged profession or as a rewarding side venture, the rewards are abundant. Sharing your knowledge not only solidifies your own understanding but also empowers others to learn, grow, and innovate. Your articles have the potential to become valuable resources, aiding developers in their projects, troubleshooting, and exploration of new technologies.

Lastly, I want to thank you for purchasing the book. I hope that this ebook will provide great benefits on your writing journey. I want to get one more favor if you haven't provided a review then please do it on the GumRoad.

Happy writing, and may your words spark inspiration and insight in the minds of countless developers around the world!

## Connect with Me

Let's connect and stay informed on all things tech, innovation, and beyond!

[Twitter](#)

[LinkedIn](#)