

Zuzu is a static site generator that takes in markdown files and renders HTML pages. This blog has been written using this generator. This enables noobs like me to write blogs without having to learn a lot of code :P .

△ The latest version of zuzu might not be updated in the [original codebase](#), but instead all the features can be found here: [fuzzymf/fuzzymf.github.io](https://fuzzymf/fuzzymf.github.io). A couple of features have been added to the original codebase, such as RSS feed, PDF generation, and CSS/JS injection. The better codebase is the one mentioned above !

- [Features](#)
- [How does it work?](#)
- [Getting Started](#)
- [Next?](#)

## Features

Zuzu builds on a very basic problem statement: *I want to start my own blog without having to learn a lot of code*. The current version supports a ton of features:

- **Markdown:** Zuzu supports Github-flavoured markdown. Ramble on about your favourite topics, add code snippets, tables, and images, and Zuzu will render them for you.
- **Syntax Highlighting:** Zuzu uses highlight.js to beautify your code snippets. Just add your code in a code block and Zuzu will take care of the rest.
- **CSS/JS Injection:** Zuzu allows you to add your own CSS and JS files. Just add them to the `static` folder and Zuzu will inject them into your HTML files. Make sure to include the path in the markdown file. Example: `...<link rel="stylesheet" href="/static/css/style.css">...`
- **Front Matter:** Zuzu uses front matter to extract metadata from the markdown file. The front matter is written in YAML and is enclosed between `---`.

**Example:** ```` --- title: 'Zuzu' date: "29-05-2022" description: "Zuzu is a static site generator that converts all your markdown files into static html pages. It uses Github-flavoured Markdown CSS and highlight.js to beautify code snippets." tag: "tech"````

```
- RSS Feed: Zuzu generates an RSS feed for your blog. Just add the rss.xml file to the static folder and Zuzu will take care of the rest.
- PDF Generation: Zuzu uses md-to-pdf to convert your markdown files into PDFs. Just add the pdf link to your markdown file and Zuzu will take care of the rest. Example: [Download PDF](/blog/documents/zuzu.pdf).
- Configurable: Zuzu is highly configurable. You can change the theme, add your own CSS and JS files, and even add your own static folders. The config.toml looks like this:
```

## Homepage

`base_url = "https://anubhavp.dev/zuzu/"`

## template

```
template = "fuzzymf"
```

...

## Path to the directory containing the templates

```
template_dir = "templates/fuzzymf"
```

## Path to the directory containing the static files

```
static_dir = "static"
```

...

## Theme

```
templateHTML = "templates/fuzzymf/template.html"
```

```
## How does it work?
```

Zuzu leverages Node.js and a few packages to convert markdown files into HTML pages. It parses the markdown files, extracts the front matter, and converts the markdown content into HTML. The HTML is then injected into a template to generate the final HTML page. The final HTML page is then written to the output directory. The packages used are:

- [MarkdownIt](https://www.npmjs.com/package/markdown-it) \*Markdown parser done right.\*
- [MarkdownItAnchor](https://www.npmjs.com/package/markdown-it-anchor) \*Header anchors for markdown-it.\*
- [Glob](https://www.npmjs.com/package/glob) \*"Globs" are the patterns you type when you do stuff like `ls .js` on the command line, or put `build/` in a `.gitignore` file.\*
- [Gray-Matter](https://www.npmjs.com/package/gray-matter) \*Parse front-matter from a string or file.\*
- [Mkdirp](https://npmjs.com/package/mkdirp) \*Create Dirs if they do not exist.\*
- [Path](https://nodejs.org/api/path.html) \*Provides utilities for working with file and directory paths.\*
- [Highlight.js](https://highlightjs.org/) \*Highlight.js is a syntax highlighter written in JavaScript. It works in the browser as well as on the server. It works with pretty much any markup, and doesn't depend on any framework and has automatic language detection.\*
- [MDtoPDF](https://www.npmjs.com/package/md-to-pdf) \*Converts markdown to pdf.\*

```
- [RSS](https://www.npmjs.com/package/rss) *RSS parser.*
```

The generator consists of, but not limited to, the following functions:

- slugify: Converts a string into a URL-friendly slug.
- readFile: Reads a markdown file, parses its front matter, and renders its content to HTML.
- templatize: Replaces placeholders in a template string with actual data.
- saveFile: Saves content to a file, creating necessary directories if they don't exist.
- getOutputFilename: Generates the output HTML filename for a given input filename and output path.
- getOutputPdfname: Generates the output PDF filename for a given input filename and output path, ensuring the folder exists.
- processBlogFile: Processes a blog file, converting it to HTML and optionally to PDF, and updates a map of blog metadata.
- processDefaultFile: Processes a default file, converting it to HTML and optionally to PDF.
- buildBlogIndex: Builds the blog index from the map of blog metadata and updates the index file.
- copyAssets: Copies assets from the source folder to the destination folder.
- main: Orchestrates the processing of all markdown files, fetching metadata, and generating the static site.

The script follows this flow:

- Initialization: Import necessary modules and define utility functions.
- MarkdownIt Setup: Initialize MarkdownIt with plugins for rendering markdown content.
- Main Function:
  - Define paths for source content, output directories, and templates.
  - Fetch metadata from a remote URL.
  - Clean and create necessary directories.
  - Process markdown files in the source directory, converting them to HTML and optionally to PDF.
  - Build the blog index from processed blog files.
  - Copy static assets to the output directory.

Execution: The script is executed by calling the main function within an async IIFE, handling any errors that occur.

## ## Getting Started

To get started with Zuzu, clone the repository and install the dependencies:

```
```bash
git clone https://github.com/fuzzymf/zuzu.git
cd zuzu
npm install
```

This would be a good place to start your markdown files. The `content` folder contains all the markdown files. The `templates` folder contains the HTML templates. The `static` folder contains the CSS and JS files.

To generate the HTML files, run:

```
npm run generate
```

You can find two themes in the `templates` folder: `fuzzymf` and `initial`. To change the theme, change the `template` in the `config.toml` file, and the `templateHTML` in the `config.toml` file. Zuzu is highly configurable. You can add your own CSS and JS files to the `static` folder and include them in the markdown file, or better, start with your own `theme`.

## Next?

The [zuzu](#) repository is a good place to start. The latest version of zuzu might not be updated in the original codebase, but instead you can always find all the features maintained here: [fuzzymf/fuzzymf.github.io](https://fuzzymf/fuzzymf.github.io).

I am eagerly looking for someone to migrate all the features to the original codebase. Feel free to ping me if you're interested.

If you really dig zuzu, I would suggest you try Zuzu out, and then maybe contribute to the codebase. The codebase is open-source and contributions are welcome. Feel free to add comments, issues, or pull requests. Update the codebase, add new features, or even create your own theme. The world is your oyster, and zuzu is your playground.