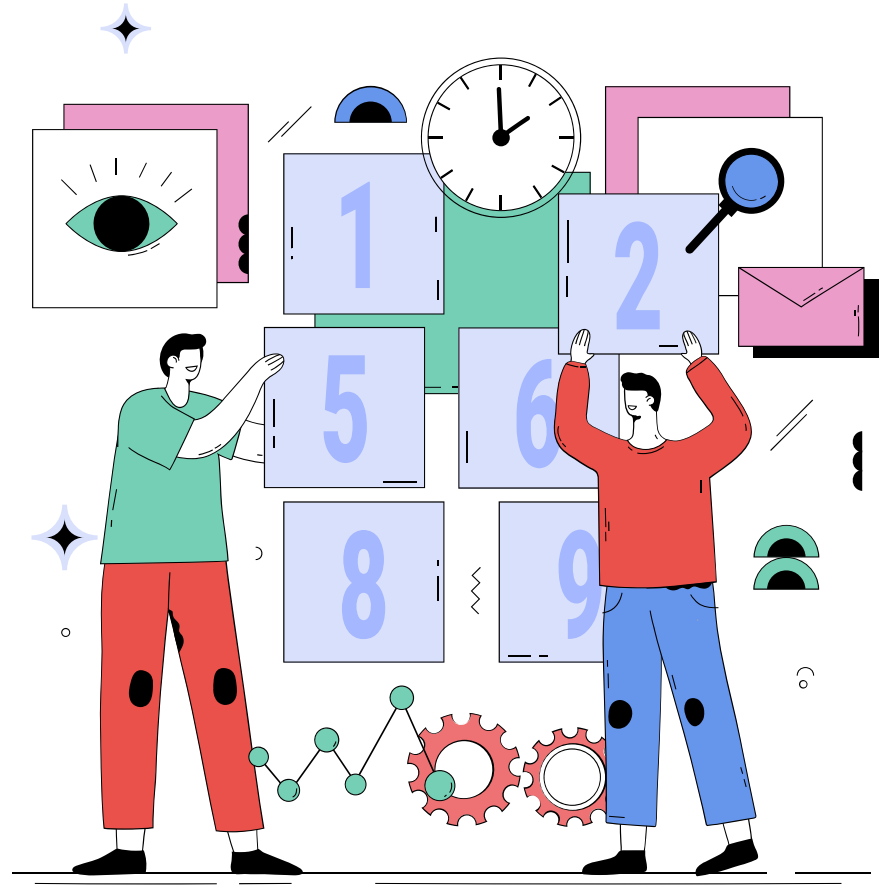
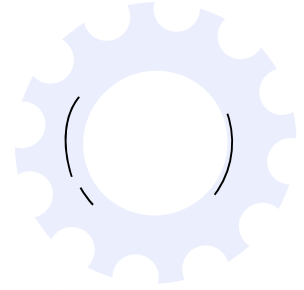
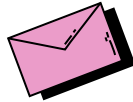
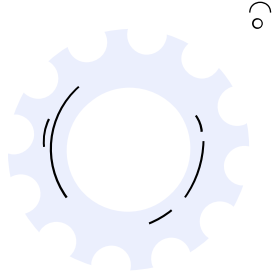


# Search Functionality Using Bluge



# PROBLEM STATEMENT

- To implement searching on documents in a website with less computational power and efficient method using Bluge.
- The existing search functionality in the targeted website uses Bleve, the motivation is to update the search functionality using Bluge



# EXISTING SYSTEM

Searching For: விரிவாக்கம்

Showing Results 1 - 5 of 5

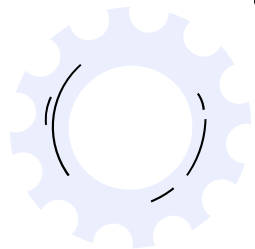
1. 02 - மழைப்பாடல் Chapter: 10 நூல் இரண்டு : கானல்வெள்ளி

para\_35 : ... மணவுறவில் உள்ளது. நம் படைபலமும் நிதிபலமும் பெருகும். கங்கை வணிகத்தை அதைக்கொண்டு விரிவாக்கம் செய்துகொள்ளமுடியும். நமக்கும் காந்தாரத்துக்கும் உறவு உருவானால் நம்மை வுத்ரியர்கள் அஞ்சுவார்கள். போரைத் தவிர...

2. 02 - மழைப்பாடல் Chapter: 16 பகுதி மூன்று : புயலின் தொடட்டில்

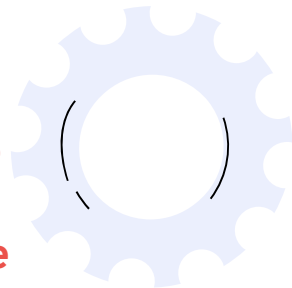
para\_14 : ...யையும் கணித்து அவன் செய்தியைச் சொல்லவேண்டும். கேட்பவர் உருவாக்கும் எதிர்வினைகளுக்கேற்ப தணிந்தும் நயந்தும் தேவையென்றால் மிஞ்சியும் தன் செய்தியை விரிவாக்கம் செய்யவேண்டும். அதன்பின்னர் அச்செய்தியை உறுதி...

# WHY MIGRATING FROM BLEVE?



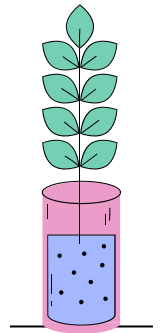
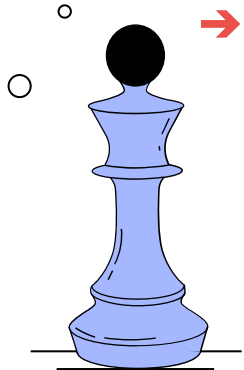
The design of Bluge is largely the same as Bleve. One of the biggest design changes of Bluge is that it now supports a portable Directory implementation. This Directory interface decouples the index implementation from many OS/filesystem details. This let us introduce a much more efficient in-memory-only index, something Bleve still struggles to offer today.

- The next major change is that Bluge supports accessing the index from multiple process (via OS locking primitives). Only one process can write at a time, but it is very useful architecturally to be able to search indexes from multiple processes, which is not possible with Bleve today.



# PROPOSED SOLUTION

- Data in website can be mapped with indexes. The purpose of storing indexes is to optimize speed and computational power to find relevant information for a search query.
- For example, if a website has a million words in it, to search for a word in it by looking onto every single word will be time consuming and memory usage will also be high.
- So , using Go language package called “ *Bluge* “ built on top of the existing package called “ *Bleve* ”.



# PROPOSED SOLUTION

## Search

Bluge



### Results

19 results

#### [Introduction](#) 1.027

... is **Bluge**? **Bluge** is an indexing/search library for Go. Developers can insert/update/delete documents in the index. Documents are composed of fields. **Bluge** support fields with text, numeric, date, and ...

#### [Code Walk-through - Indexing](#) 1.005

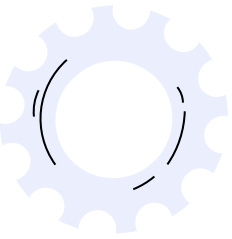

...ngle document. We step into the **Bluge** code and discuss the important details behind the indexing process. References Main Program  
<https://github.com/blugelabs/bluge/blob/f89eff45771cfe7cb151b01bc62...>

### Filter

#### Type

- ☐ Bluge Docs (11)
- ☐ Blog (6)
- ☐ Page (2)

# TECHNICAL CHALLENGES



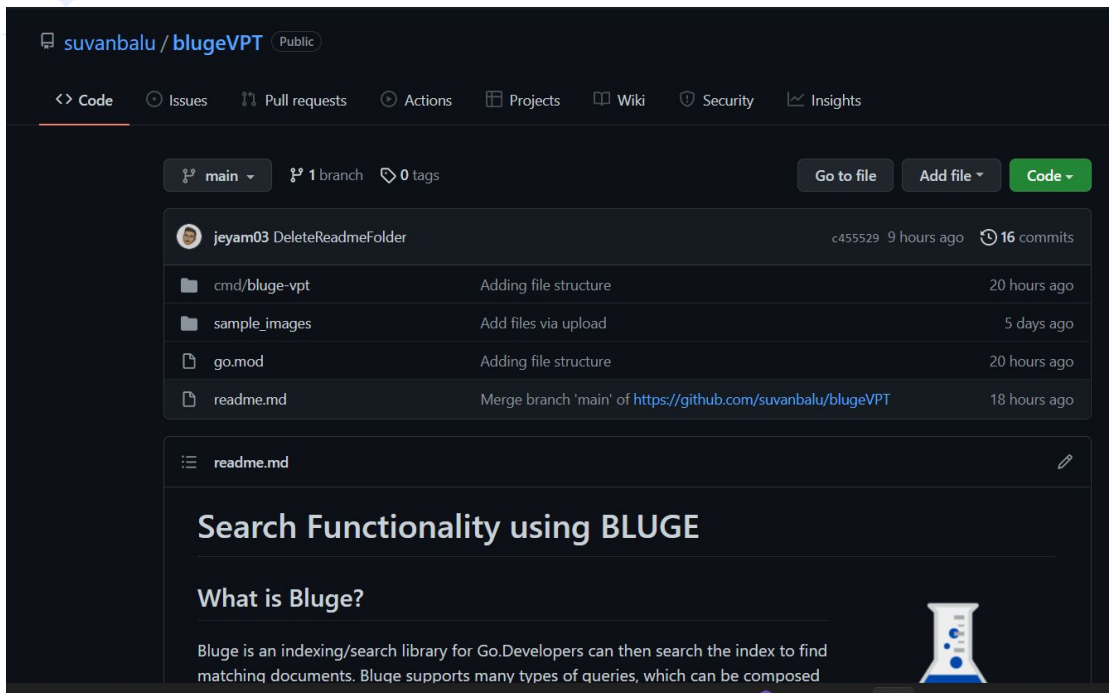
**Go is a new language for the team, we are currently learning it.**



**Implementation is for a website that has tamil text.**



# PROGRESS TILL NOW



suvanbalu / blugeVPT Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Add file Code

jeyam03 DeleteReadmeFolder c455529 9 hours ago 16 commits

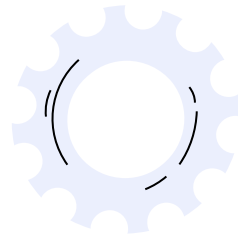
cmd/bluge-vpt	Adding file structure	20 hours ago
sample_images	Add files via upload	5 days ago
go.mod	Adding file structure	20 hours ago
readme.md	Merge branch 'main' of <a href="https://github.com/suvanbalu/blugeVPT">https://github.com/suvanbalu/blugeVPT</a>	18 hours ago

readme.md

## Search Functionality using BLUGE

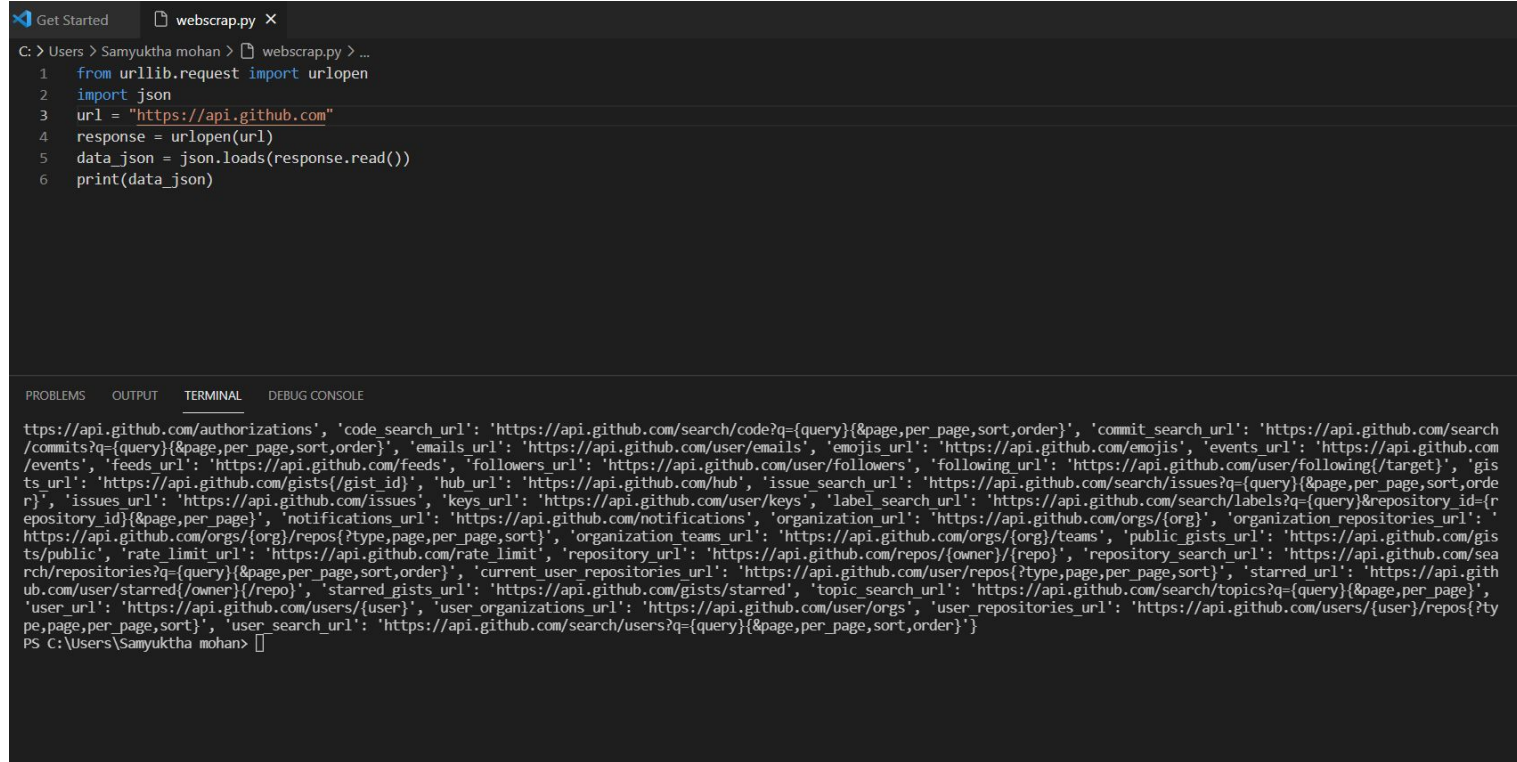
### What is Bluge?

Bluge is an indexing/search library for Go. Developers can then search the index to find matching documents. Bluge supports many types of queries, which can be composed



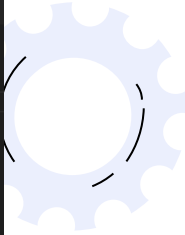


# PROGRESS TILL NOW



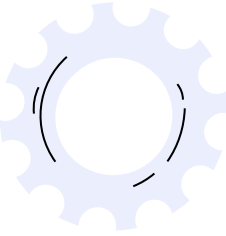
```
Get Started | webscrap.py X
C: > Users > Samyuktha mohan > webscrap.py ...
1 from urllib.request import urlopen
2 import json
3 url = "https://api.github.com"
4 response = urlopen(url)
5 data_json = json.loads(response.read())
6 print(data_json)

PROBLEMS | OUTPUT | TERMINAL | DEBUG CONSOLE
https://api.github.com/authorizations', 'code_search url': 'https://api.github.com/search/code?q={query}&page,per_page,sort,order}', 'commit_search url': 'https://api.github.com/search/commits?q={query}&page,per_page,sort,order}', 'emails url': 'https://api.github.com/user/emails', 'emojis url': 'https://api.github.com/emojis', 'events url': 'https://api.github.com/events', 'feeds url': 'https://api.github.com/feeds', 'followers url': 'https://api.github.com/user/followers', 'following url': 'https://api.github.com/user/following{/target}', 'gists url': 'https://api.github.com/gists{/gist_id}', 'hub url': 'https://api.github.com/hub', 'issue_search url': 'https://api.github.com/search/issues?q={query}&page,per_page,sort,order}', 'issues url': 'https://api.github.com/issues', 'keys url': 'https://api.github.com/user/keys', 'label_search url': 'https://api.github.com/search/labels?q={query}&repository_id={repository_id}&page,per_page}', 'notifications url': 'https://api.github.com/notifications', 'organization url': 'https://api.github.com/orgs/{org}', 'organization repositories url': 'https://api.github.com/orgs/{org}/repos?type,page,per_page,sort}', 'organization teams url': 'https://api.github.com/orgs/{org}/teams', 'public gists url': 'https://api.github.com/gists/public', 'rate limit url': 'https://api.github.com/rate_limit', 'repository url': 'https://api.github.com/repos/{owner}/{repo}', 'repository_search url': 'https://api.github.com/search/repositories?q={query}&page,per_page,sort,order}', 'current user repositories url': 'https://api.github.com/user/repos?type,page,per_page,sort}', 'starred url': 'https://api.github.com/user/starred{/owner}/{repo}', 'starred gists url': 'https://api.github.com/gists/starred', 'topic_search url': 'https://api.github.com/search/topics?q={query}&page,per_page}', 'user url': 'https://api.github.com/users/{user}', 'user organizations url': 'https://api.github.com/user/orgs', 'user repositories url': 'https://api.github.com/users/{user}/repos?type,page,per_page,sort}', 'user_search url': 'https://api.github.com/search/users?q={query}&page,per_page,sort,order}'
PS C:\Users\Samyuktha mohan>
```



# UPCOMING PLANS

1. Get the JSON directory for the website
2. Work on the code
3. Create good UI/UX for the search functionality



# TEAM

**Jeyam Palaniappan D**

**Samyuktha M S**

**Sashti Amar R A**

**Suvan Sathyendra B**

**T G Ashwin Kumar**

