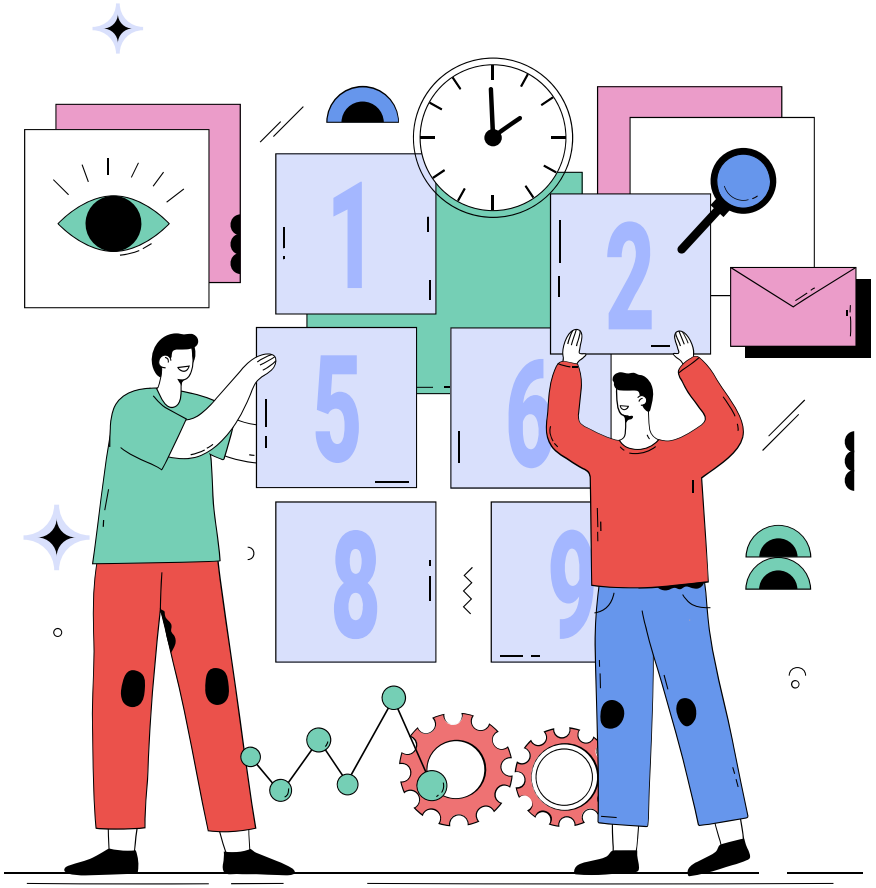
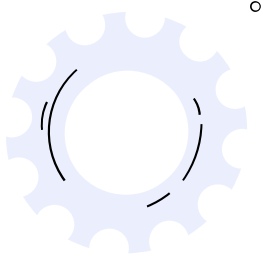
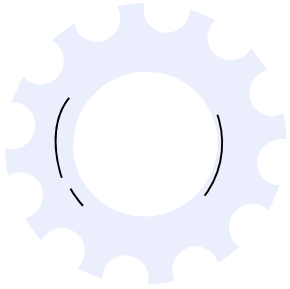
Search Functionality Using Bluge



PROBLEM STATEMENT



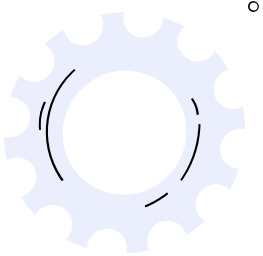


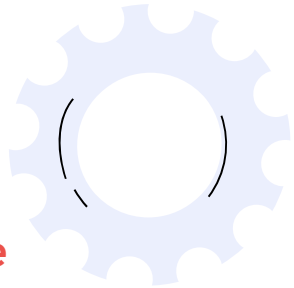
- To implement searching on documents in a website with less computational power and efficient method using Bluge.

- The existing search functionality in the targeted website uses Bleve, the motivation is to update the search functionality using Bluge

EXISTING SYSTEM

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.

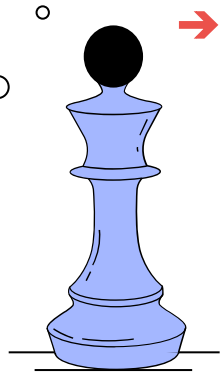
- The next major change is that Bluge supports accessing the index from multiple process (via OS locking primitives). Only one process can write at a time, but it is very useful architecturally to be able to search indexes from multiple processes, which is not possible with Bleve today.

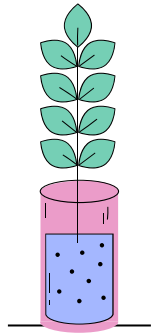
PROPOSED SOLUTION

- 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.

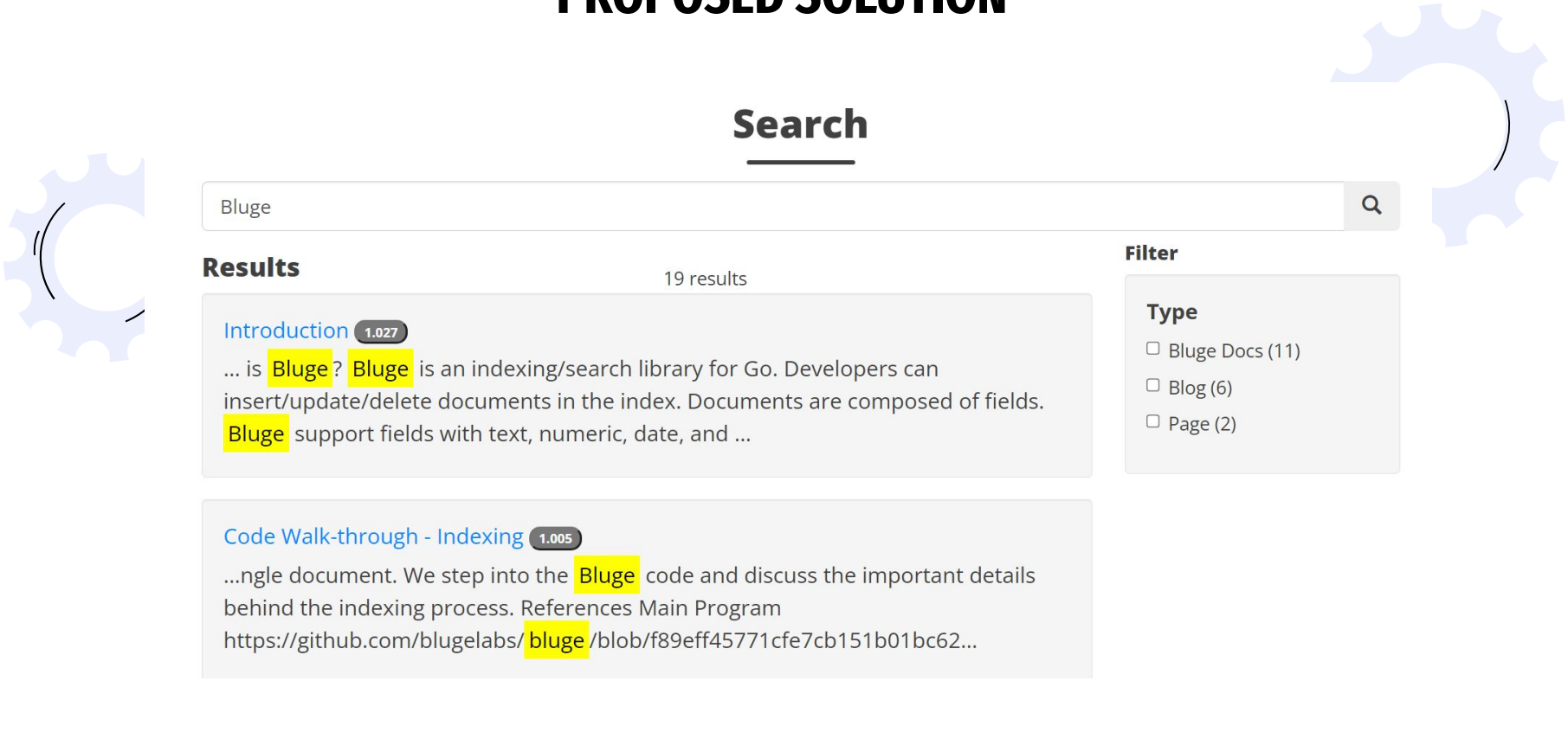
- For example, if a website has a million words in it, to search for a word in it by looking onto every single word will be time consuming and memory usage will also be high.

- So , using *Go* language package called " *Bluge* " built on top of the existing package called " *Bleve* ".





PROPOSED SOLUTION

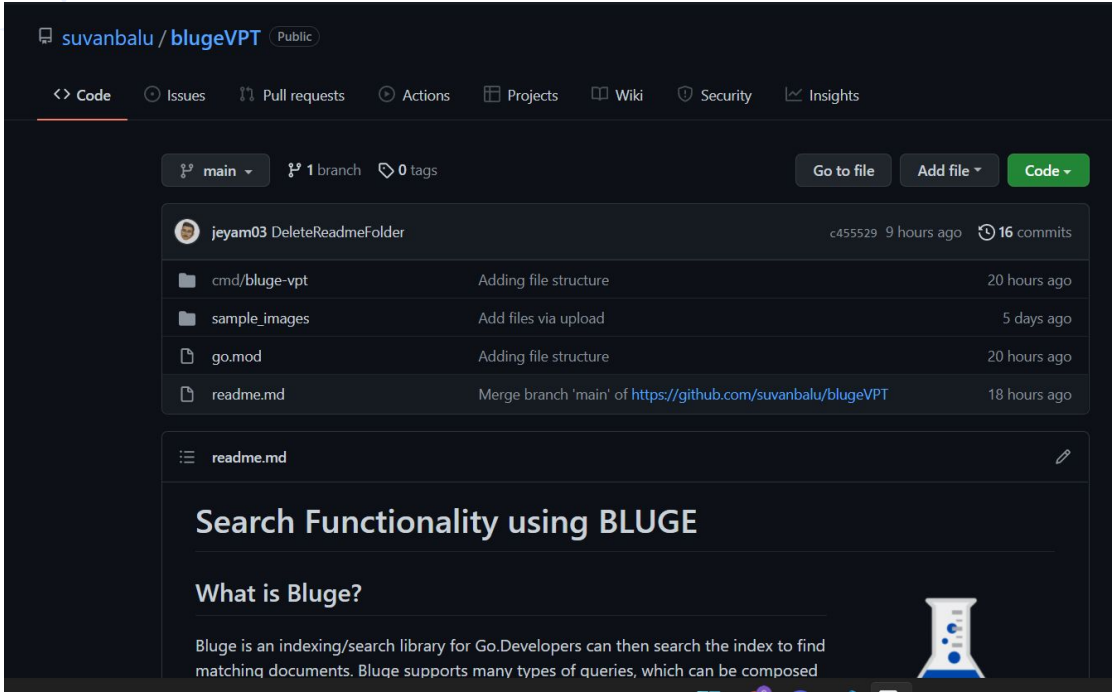


TECHNICAL CHALLENGES

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



PROGRESS TILL NOW

Get Started | webscrap.py

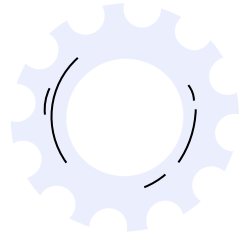
C: > Users > Samyuktha mohan > webscrap.py > ...

```
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

```
ttps://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}', 'gis
ts_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,orde
r}', '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={r
epository_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/gis
ts/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/sea
rch/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.gith
ub.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{?ty
pe,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 Sathyendira B
T G Ashwin Kumar