A PROJECT REPORT

on

“ Fing Search Engine ”

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Sahil Sheikh **B-68**

Pranjal Sonawane **B-69**

Sayali Thombare **B-70**

Supervisor

Ms. Rajashree Gadhave

Department of Computer Engineering

MES Pillai’s HOCL College of Engineering and Technology

University of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled "**Fing Search**" is a bonafide work of **Sahil Sheikh (B-68) , Pranjal Sonawane (B-69) & Sayali Thombare (B-70)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "**Computer Engineering**".

Ms. Rajashree Gadhave

(Supervisor)

Ms.Rohini Bhosale

(Head of Department)

Dr. J.W. Bakal

(Principal)

Contents

Abstract

Search Engine project will able to provide users required information at one particular place by using the words and patterns entered by the user during their search operation. All the information will be provided over the browser screen where users can select appropriate link filtered by the search query. Whatever the information presented to the user can be in any form by default such as it may be in the form of web pages. The search query will provide listing of web pages as per their occurrence during search operations. This search mechanism will work on the concept of tags and meta tags which are used while writing the contents under the particular web pages. If the user's query don't matched with the tags and meta tags then it will go for summary section to match the given words in order to present the exact output or results. Upon going through web pages an index file will be created where listing of pages will be done by the system and present them as per their index number. Keyword density for a particular post under a particular web page will also help the system to set the priority for indexing the web page.

Acknowledgement

It gives immense pleasure in bringing out this synopsis of the project entitled “ Fing Search ”.In performing our project we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much Pleasure. We would like to show our gratitude to our mentor Ms. Rajashree Gadhave and project coordinator Ms. Snehal Chitale involved in this project and for giving us a good guideline for numerous consultations and team members itself, have made valuable comment suggestions on this proposal which gave us inspiration to improve our project. We thank all the Team members for their help directly and indirectly to complete our project .

List of Abbreviations

NLP :- Natural Language Processing

DB :- DataBase

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Introduction

Fing search engine is a software system that is designed to carry out web searches. They search the web in a systematic way for particular information specified in a textual web search query. In this project we are going to build a search engine which can show us the search results it fetched from a few selected web sites. Fing provides all basic need for searching requirement to the user , also added lot of functionality to make user experience better and having clear and simple user interface .Fing search is also features thought of the day on home page .

1.2 Motivation

Search engines are programs that make it easy for people to search the internet for a relevant web page. The three main functions of a search engine are collecting information about webpages, categorizing those webpages, and creating an algorithm that makes it easy for people to find relevant web pages.

1.3 Problem Statement & Objectives

Because every time, every time more and more web pages are added to the Internet to achieve top Ranking Search Engine.
The relevance of search engine algorithms are proprietary. Because of the mysterious nature of search engine relevancy algorithms, the process of achieving a higher ranking in organic search engine results.

1.4 Organization of the Report

The report is divided in 4 sections: The first section contains references and summarization. It contains the abstract, abbreviations, acknowledgement. The second section contains Introduction, Problem statement, objectives, Motivations. The third section contains the literature survey and the mini project contribution. The last section contains the full explanation about the proposed system. It contains details about the architecture and framework, the algorithm and process design, and the hardware and software details, experiment, results, conclusion and future work. Lastly we have references.

CHAPTER 2

LITERATURE SURVEY

2.1 Survey of Existing System

Search engines work by crawling hundreds of billions of pages using their own web crawlers. These web crawlers are commonly referred to as **search engine bots** or **spiders**. A search engine navigates the web by downloading web pages and following links on these pages to discover new pages that have been made available.

2.2 Limitation Existing system or research gap

The search engine might show way too much irrelevant information on your screen. Sometimes you even can not find anything useful from search results. It wastes your time to pick up useful information from seas of search results.

CHAPTER 3

PROPOSED SYSTEM

3.1 Introduction

• Fing search engine is a software system that is designed to carry out web searches. They search the web in a systematic way for particular information specified in a textual web search query.

• In this project we are going to build a search engine which can show us the search results it fetched from a few selected web sites.

3.2 Architecture/ Framework

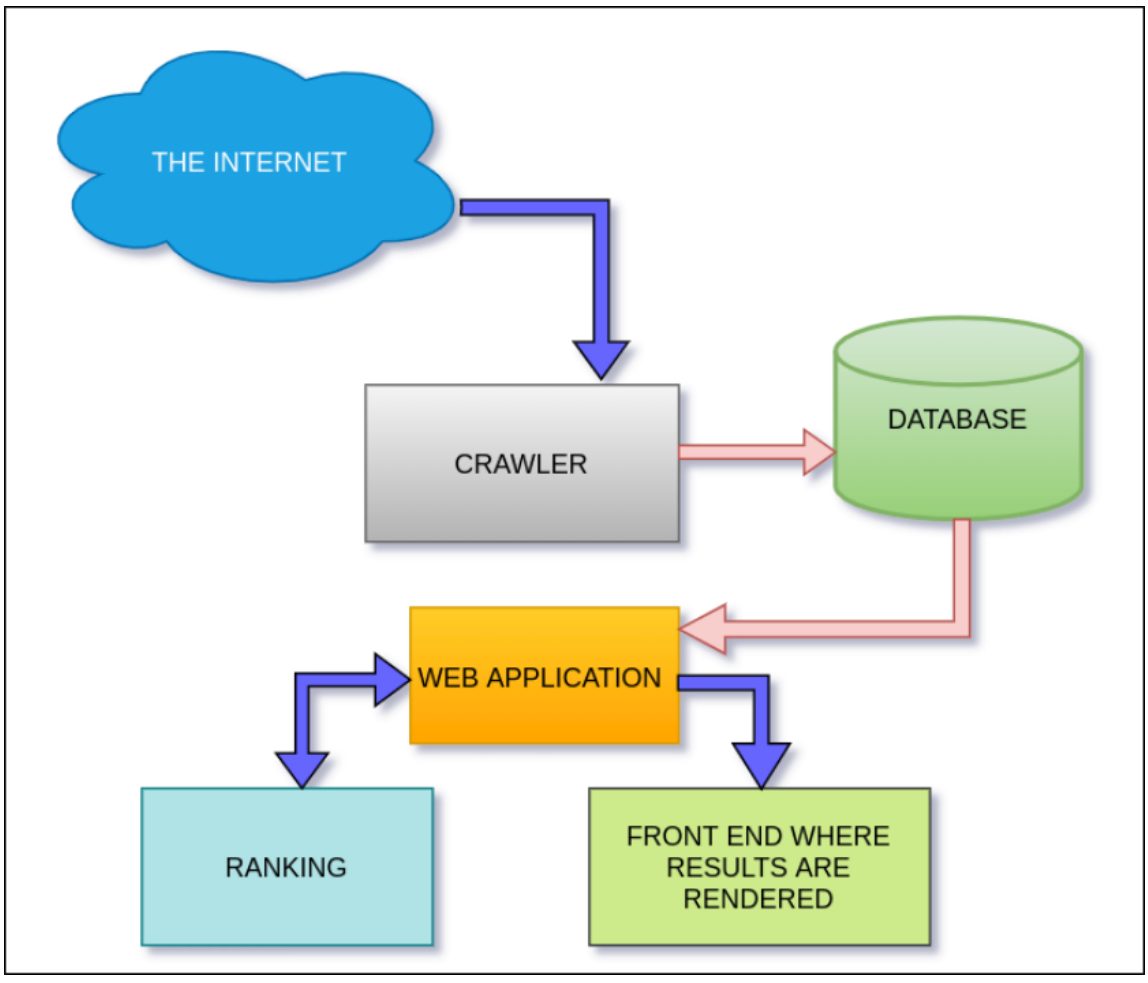

Fig 1.1 Framework

3.3 Algorithm and Process Design

APPROACH FOR BUILDING THE SEARCH ENGINE

A search engine performs four basic processes:

CRAWLING:

Web search engines get their information by crawling from site to site. The crawler is provided with an entrypoint from which it starts collecting the links and text data and storing them in the database.

INDEXING:

Indexing means associating the data found on the web pages with the domain it was found on and HTML fields. The way data is stored in the database is a major contributor to the efficiency of the search engine.

SEARCHING:

As the name implies searching means to search the database for relevant results to the search query.

RANKING:

Ranking means to rank the search results found from the above operation in order of their relevance to the user. The better ranking system results in a better search experience.

WEB CRAWLER

The word “crawler” itself can be intimidating to many people but it is basically a script having a few lines of code.

A web crawler, spider, or search engine bot downloads and indexes content from all over the Internet. They're called web crawlers because crawling is the technical term for automatically accessing a website and obtaining data via a software program.

The crawler goes from page to page and stores the data fetched from it in the database, so that the information can be retrieved when it's needed.

APPROACH TO BUILD THE CRAWLER

- We are going to use the following python libraries to achieve the task

    1 requests library to fetch the pages.

    2 beautifulsoup4 to parse the response received from the response object.

    3 pymongo to connect to mongodb where we are going to store the data.
- Yes that’s it, that’s all we need.
- We will build a python class named Crawler inside the crawler.py file.
- The first thing we want to do is to make a connection with our database using “pymongo” library.

    client = pymongo.MongoClient(connect_url_to_mongodb)

    db = client.name_of_database

- After the connection is made we are going to define two methods inside the class “Crawler” named start_crawling and crawl.
- Both of the methods mentioned above are going to take two arguments:
- url (string containing the url to the page we want to parse)
- depth(integer parameter to control the number of pages your program crawls)

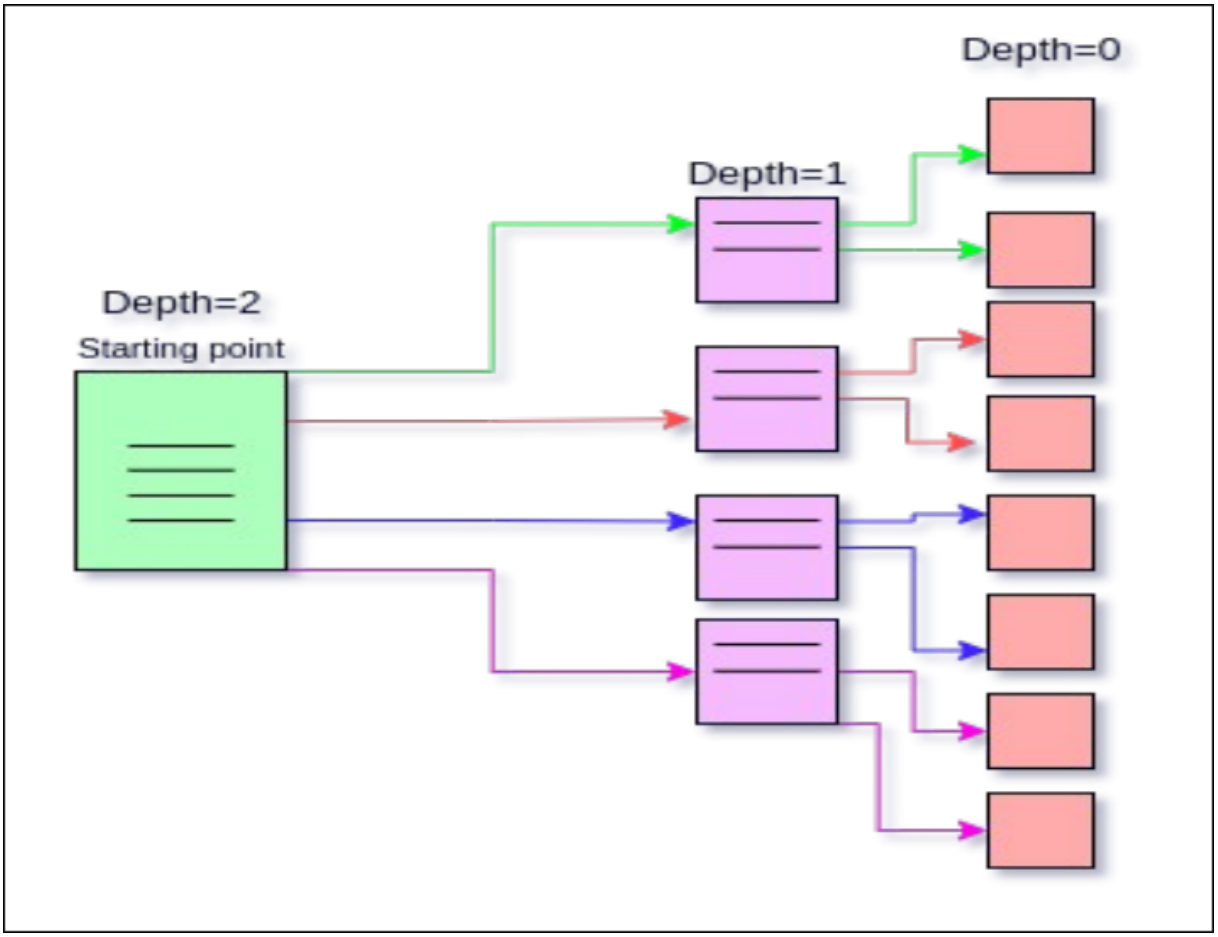

Fig 1.2 Crawler Depth Search

Ranking Mechanism

Once the search results are fetched from the database, next comes sorting them in order of relevance. To achieve this, first, the search query is separated into different keywords, after which, each search result is checked for the number of keywords present in it, and ranked according to it.

Algorithm

- Get the search query after the preprocessing and store them in an array keywords
- Get the search results from the database and store them in an object results.
- Check for the number of keywords present in each result in the object results.
- For each keyword in the title of the result, it is given score +2, and for each keyword in the description of the result, it is given score +1.
- Sort the results object in the descending order of score of each result.

The Web App

We are going to make the web app using flask library in python. There will be two pages one the homepage and the other the search results page.

Basic structure of the web application:

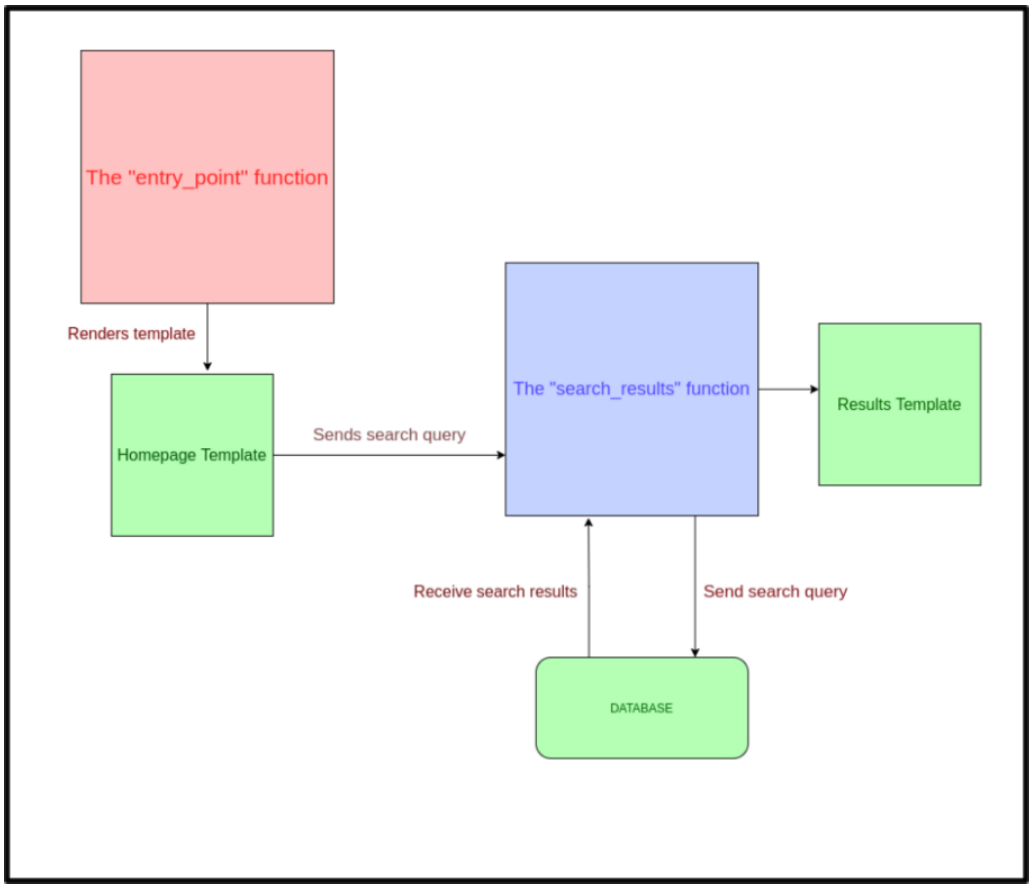

Fig 1.3 Structure of Web application

3.4 Details of Hardware & Software

This project is built in Visual Studio Code. The project required minimal processing capability.

- Python

3.5 Experiment and Results

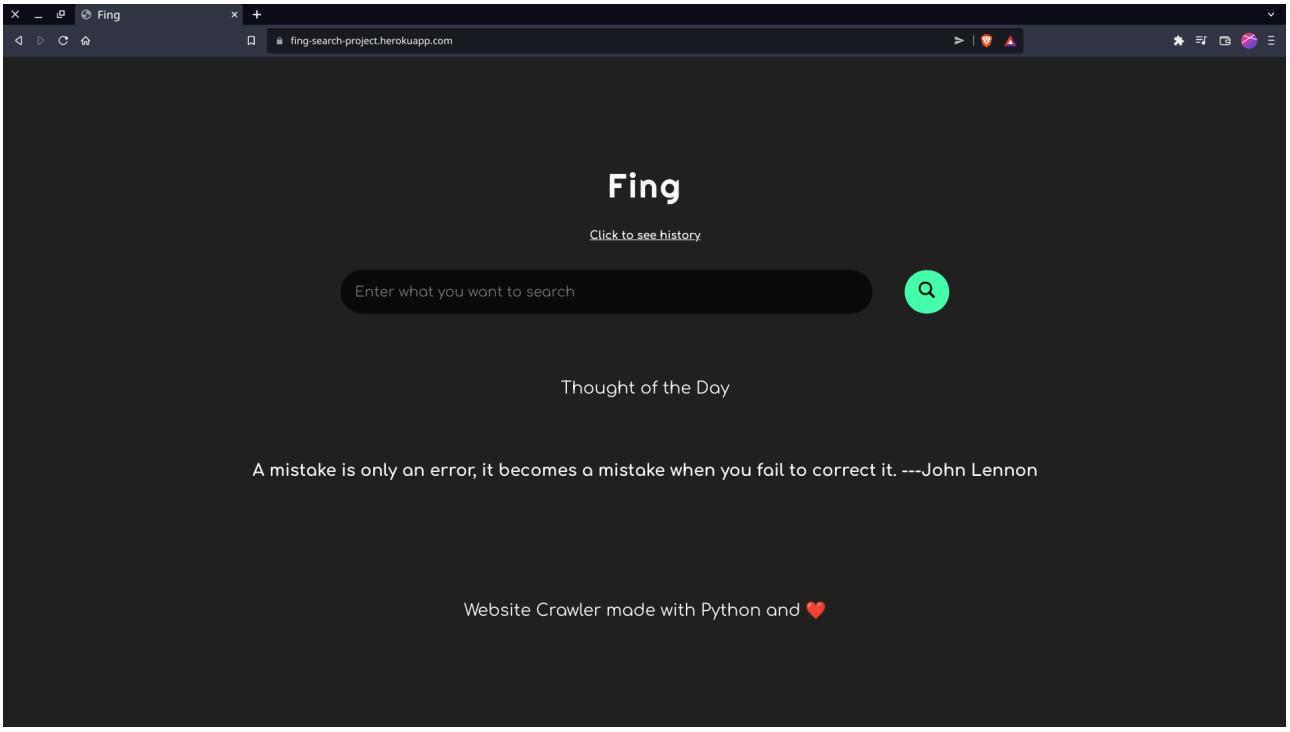

Fig 1.4 Home Page

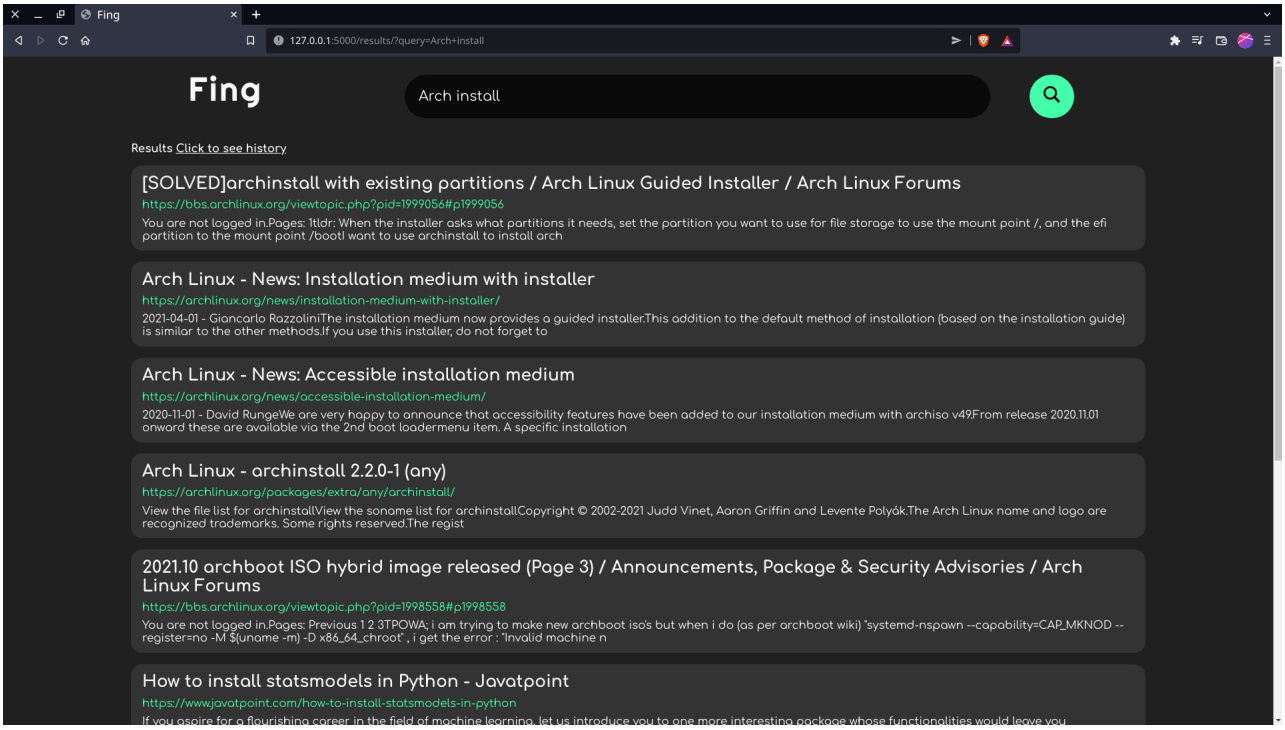

Fig 1.5 Search Results

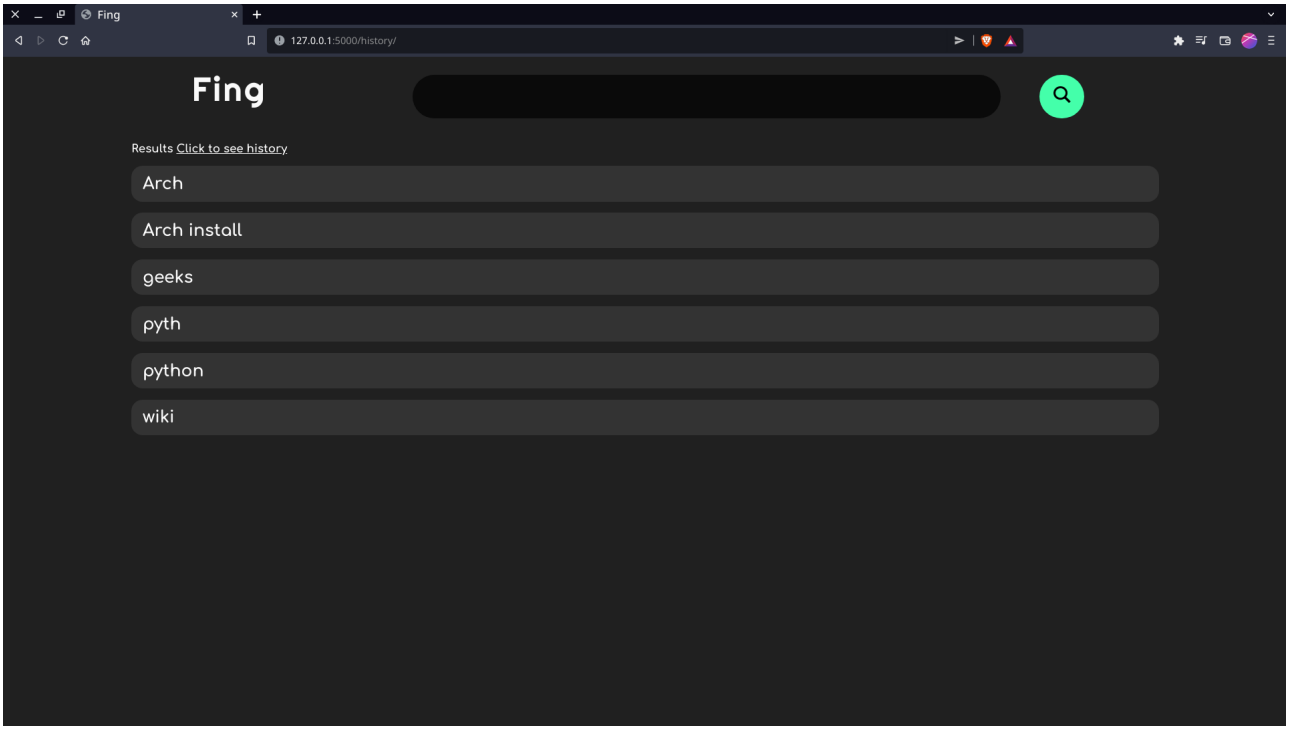

Fig 1.6 Search History

3.6 Conclusion and Future work

Fing provide internet-based search services, providing accessibility to the world's online information.This project met all its original intended requirements and goals and was overall a great success .In Future work we will use ML and NLP and different indexing algorithm so user will more and more accurate result every time they searches also including voice assistance and many more features .

References

- https://www.deepcrawl.com/knowledge/technical-seo-library/how-do-search-engines-work/
- https://flask.palletsprojects.com/en/2.0.x/
- https://en.wikipedia.org/wiki/PageRank
- https://pythonhosted.org/Flask-paginate/
- https://www.mongodb.com/blog/post/getting-started-with-python-and-mongodb