

ATMIYA UNIVERSITY

RAJKOT



A

Report On

Web Crawler

Under subject of

PROJECT

B.TECH, Semester – VII

(Computer Engineering)

Submitted by:

1. Dhairya S. Patel - 190002082
2. Dhruvik A. Zatakiya - 190002129

Prof. Nirali Borad

(Faculty Guide)

Prof. Tosal M. Bhalodia

(Head of the Department)

Academic Year

(2022-23)

CANDIDATE'S DECLARATION

We hereby declare that the work presented in this project entitled “**Web Crawler**” submitted towards completion of project in **7th Semester** of B.Tech. (Computer Engineering) is an authentic record of our original work carried out under the guidance of “**Prof. Nirali Borad**”.

We have not submitted the matter embodied in this project for the award of any other degree.

Semester: 7th

Place: Rajkot

Signature:

Dhairya S. Patel – 190002082

Dhruvik A. Zatakiya - 190002129

**ATMIYA
UNIVERSITYRAJKOT**



CERTIFICATE

Date:

This is to certify that the “**Web Crawler**” has been carried out by **Dhruvik A. Zatakiya** under my guidance in fulfillment of the subject Project in COMPUTER ENGINEERING (7thSemester) of Atmiya University, Rajkot during the academic year 2022.

Prof. Nirali Borad

(Project Guide)

Prof.Tosal M.Bhalodia

(Head of the Department)

**ATMIYA
UNIVERSITYRAJKOT**



CERTIFICATE

Date:

This is to certify that the “**Web Crawler**” has been carried out by **Dhairya S. Patel** under my guidance in fulfillment of the subject Project in COMPUTER ENGINEERING (7th Semester) of Atmiya University, Rajkot during the academic year 2022.

Prof. Nirali Borad

(Project Guide)

Prof.Tosal M.Bhalodia

(Head of the Department)

ACKNOWLEDGEMENT

We have taken many efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to Prof. Nirali Borad for their guidance and constant supervision as well as for providing necessary information regarding the Mini Project titled “**Web Crawler**”. We would like to express our gratitude towards staff members of Computer Engineering Department, Atmiya University for their kind co- operation and encouragement which helped us in completion of this project.

We even thank and appreciate to our colleague in developing the project and people who have willingly helped us out with their abilities.

Dhairya S. Patel – 190002082

Dhruvik A. Zatakiya - 190002129

ABSTRACT

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion. Web crawling is an important method for collecting data on, and keeping up with, the rapidly expanding Internet. A vast number of web pages are continually being added every day, and information is constantly changing. This project is an overview of Web Crawler Supported Smart Search and the policies like searching, ranking, indexing involved in it.

INDEX

Certificate	3
Acknowledgement	5
Abstract	6
1. Introduction	10
1.1 Project Details: Broad Specifications	
1.2 Technology Used	
1.3 Project Planning	
2. Software Requirement and Specification	13
2.1 Purpose	
2.2 Scope	
2.3 Overall Description	
2.4 External Interface Requirements	
2.5 Non Functional Requirements	
3. Design & Planning	17
3.1 Use Case Diagram	
3.2 Data Flow Diagrams	
3.3 Network Activity Diagram	
3.4 Gantt Chart	
3.5 Home Page	
3.6 Crawl by Admin	
3.7 View Result by Admin	
3.8 Admin Manage	
3.9 Request Crawl by Employee	
4. Implementation	24
4.1 Implementation Environment	

4.2 Description of Modules	
4.3 Algorithm	
5. Testing	29
5.1 Testing Plan	
5.2 Testing Strategy	
5.3 Testing Methods	
5.4 Test Cases	
6. Limitations and Future Enhancements	33
6.1 Limitations	
6.2 Future Enhancements	
7. Conclusion	34

CHAPTER 1 :

INTRODUCTION

1. Introduction

1.1 Project Details: Broad Specifications

Web Crawler Supported Smart Search is a web application which allows normal users to search through local intranet and allows employees and admin to add seed link to be crawled. So, basically it is “*From Employees For Employees*” to share knowledge.

1.2 Technology Used

Front End: HTML, CSS, JS, Bootstrap

HTML: Hypertext Markup Language is the standard markup language for creating web pages and web applications. With Cascading Style Sheets and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.

JS: JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Bootstrap: Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

Binding : Flask Framework(Python) Flask is a micro web framework written in Python and based on the Jinja2 template engine. It is BSD licensed. The latest stable version of Flask is 0.12.2 as of May 2017.

Back End : Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Database : MongoDB is an open-source crossplatform document-oriented database program. Classified as NoSQL database program, MongoDB uses JSON-like documents with schemas .(JSON - Javascript Object Notation)

Indexing : Apache Solr is an open source enterprise search platform, written in Java, from the Apache Lucene project. Its major features include fulltext search, hit highlighting, faceted search, real-time indexing.

1.3 Project Planning

- **Requirement Gathering:** An analysis of search engine methodologies and how web crawler works in internet web structure.
- **System Design:** Data collection for system design. Database structure design. Preparation of function specification.
- **Systems Development:** Database server, web server, indexing server and four modules of systems will be developed in sequence. The four modules are spider module, indexer module, ranker module and query module.
- **Systems Testing:** Final systems testing, unit testing and integration testing will be performed.
- **Deployment:** System is currently deployed to local server only which is localhost.

CHAPTER 2 :

SOFTWARE REQUIREMENTS SPECIFICATION

2. Software Requirement Specifications

2.1 Purpose

This is a SRS document which refers to Web Crawler Supported Smart Search Release 2018 version 1. It describes the functionality and specification of how the web crawler will help in serving the employees of a firm in an efficient way.

2.2 Scope

This system is designed to crawl any website where employees and administrators can give any seed link to crawl that website and they also can use smart search to efficiently search across collected data. It can not search outside collected data and currently dynamic ranking is not supported.

2.3 Overall Description

Product Perspective

This software is developed as a part of course work the subject “System Design Practice”. The software aims to crawl/search online with ease. The web- app’s main perspective is towards efficient ranking, searching and provide a user-flexible system.

Product Functions

Crawling: When a spider is building its lists, it is called Web crawling. In this process spider crawls pages starting with seed link.

Ranking: It is the process of giving rank to collected links based on their inter relation(tree structure), the higher the rank the greater chance of it to be displayed on first page of search result.

Indexing: It is the process of giving index to crawled pages so that at search time they are accessed faster compare to traditional database queries.

Searching: When user enter any keyword(s) and the result is displayed, this process is called searching where links to be displayed are searched from already indexed data and most appropriate results are displayed first.

User Classes and Characteristics

Basically there are two types of end users i.e.,

Administrator: Administrator of a firm can manage which links to crawl and which to not, also she can add links to crawl and manage crawled data. She can manage employees too.

Employees: Employees of the firm can provide seed link to crawl and they can search for any detail related to its firm in user section of the system.

Operating Environment

The system is a web-app and not an android or iOS application because development in different operating environment will take time and cross-platform development is not the base of the development team and a web-app can be used by everyone in an efficient way. Therefore, after hosting this web-app can be accessible from any web browser.

Design and Implementation Constraints

The constraints are financial as well as at a corporate level. The system can only crawl allowed pages so if website owner or generally pages forbid some pages which require authentication or authorization will not be crawled.

Assumptions and Dependencies

System will not work if appropriate network is not available. Mainline network of system highly needs electricity to work on. System requires to be

given a seed link to crawl, it cannot crawl randomly any website found in between of any previous crawling job.

2.4 External Interface Requirements

User Interfaces

Application can be accessed through any browser interface. The software will be fairly compatible with Microsoft Internet Explorer Version 6 and above or other modern web browsers.

Hardware Interfaces

➤ Server Side:

- Operating System: Windows.
- Processor: Pentium 3.0 GHz or higher.
- RAM: 2GB or higher.
- Hard-Disk: 100GB or more.

➤ Client Side:

- Operating System: Windows 7,8, 8.1,10.
- Processor: Pentium 3.0 GHz or higher.
- RAM: 2GB or higher.

- **Software Interfaces**

➤ Client Side:

HTML5 supported Web Browsers, Windows 7, 8, 8.1, 10, MAC OS, Linux (All Flavors).

➤ Server Side:

Windows, MongoDB Database, Apache Solr(core), Python Interpreter.

2.5 Non Functional Requirements

Performance Requirements

The average response time for a user is 0.36 sec. The expected accuracy of output is 90%. For faster access we have provided solr indexing. Ranking is done in main memory so that is very fast.

Safety Requirements

If any testing purpose link of more than one domain name is given then it may lead the system to crash (e.g. <http://newsbytag.herokuapp.com>) If any link contains non-html page then it will be discarded immediately.

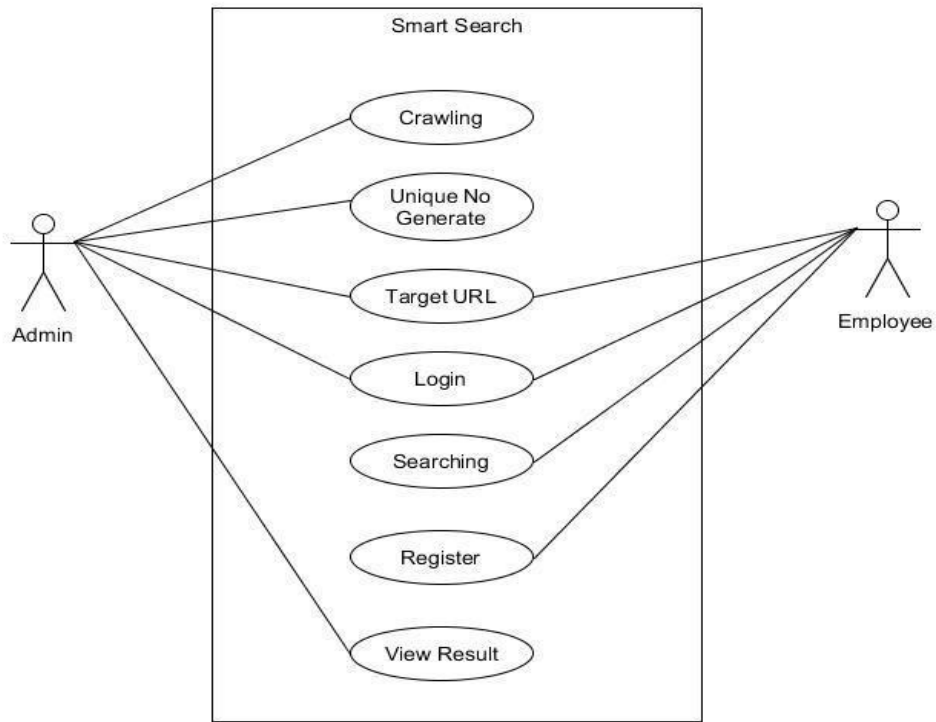
Software Quality Attributes

All the software modules are developed in python, which makes the system extensible and robust. Secondly the system will provide the user with easy to use and understandable GUI interface. For good interface we have used standard bootstrap library which also provides consistency.

CHAPTER 3: DESIGN & PLANNING

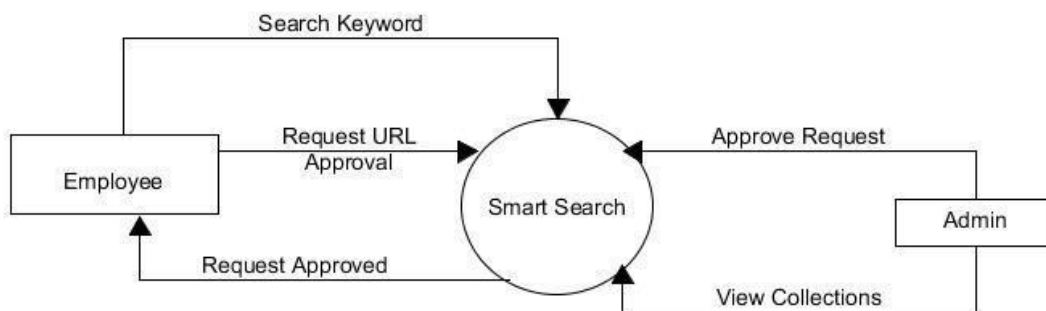
3. Design

3.1 Use-Case Diagram

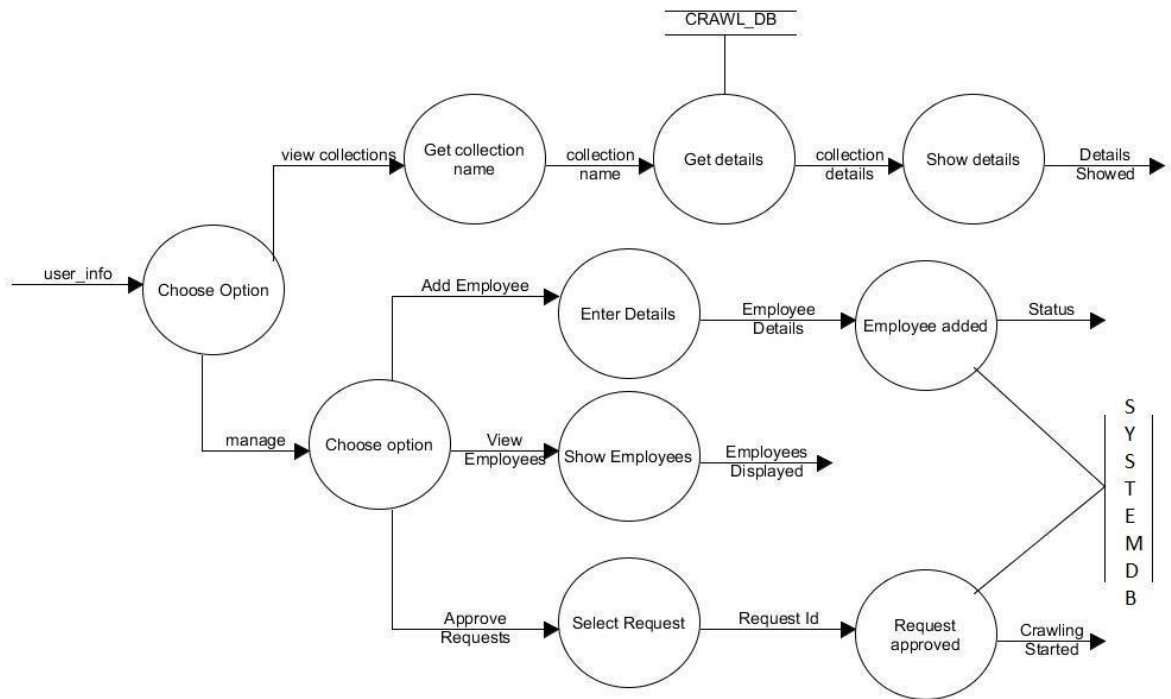


3.2 Data Flow Diagrams

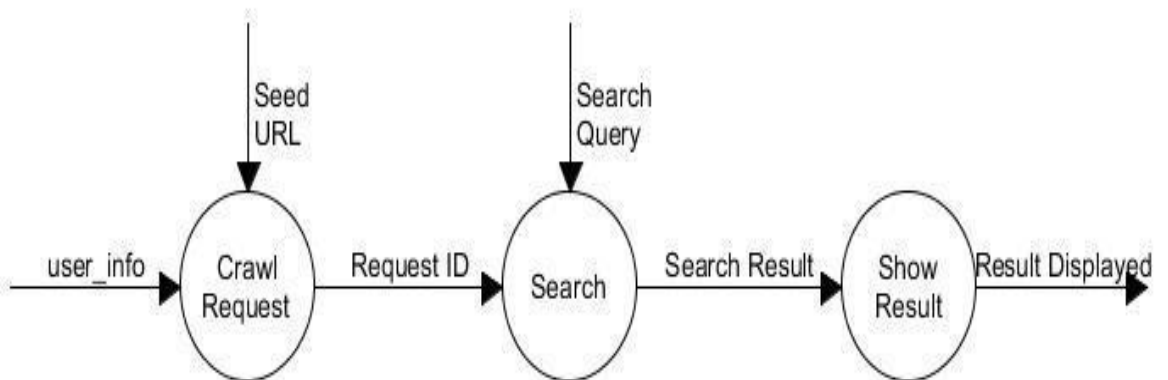
- Level - 0



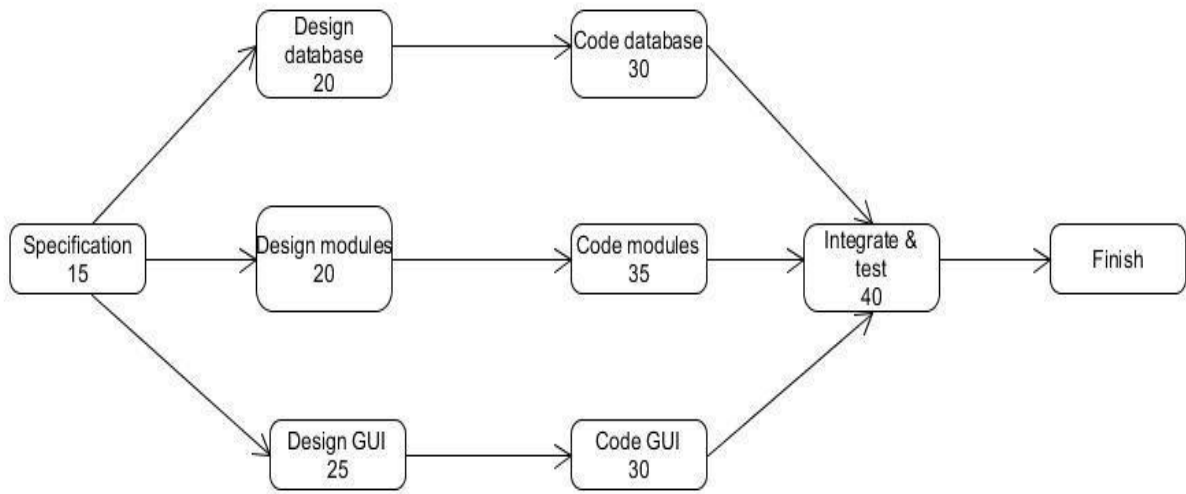
- **Level -1(a)**



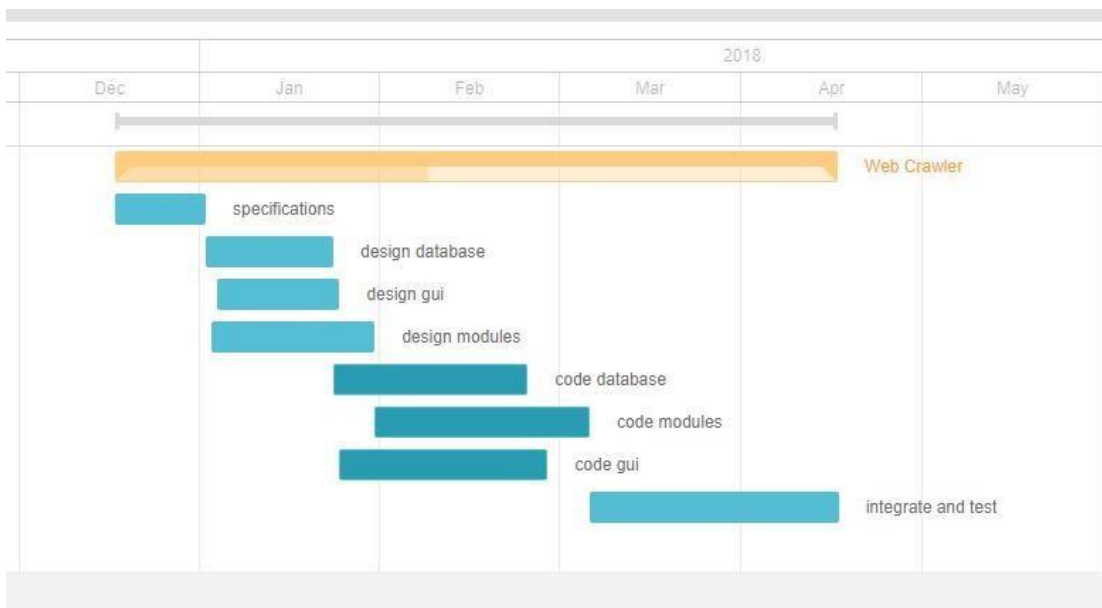
- **Level -1(b)**



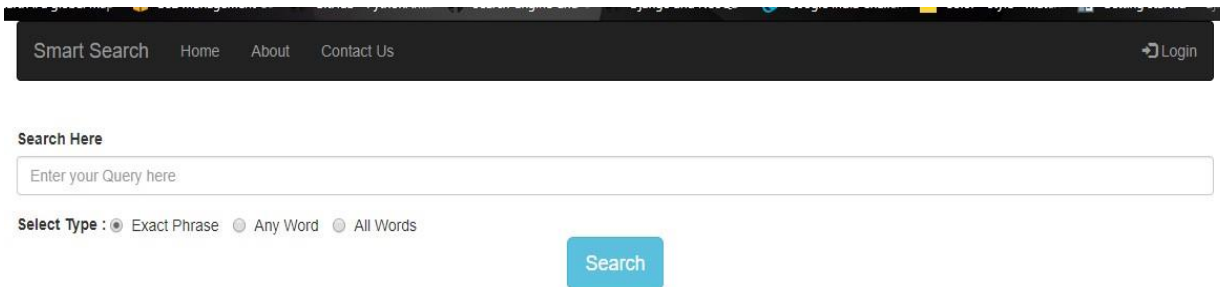
3.3 Network Activity Diagram



3.4 Gantt Chart

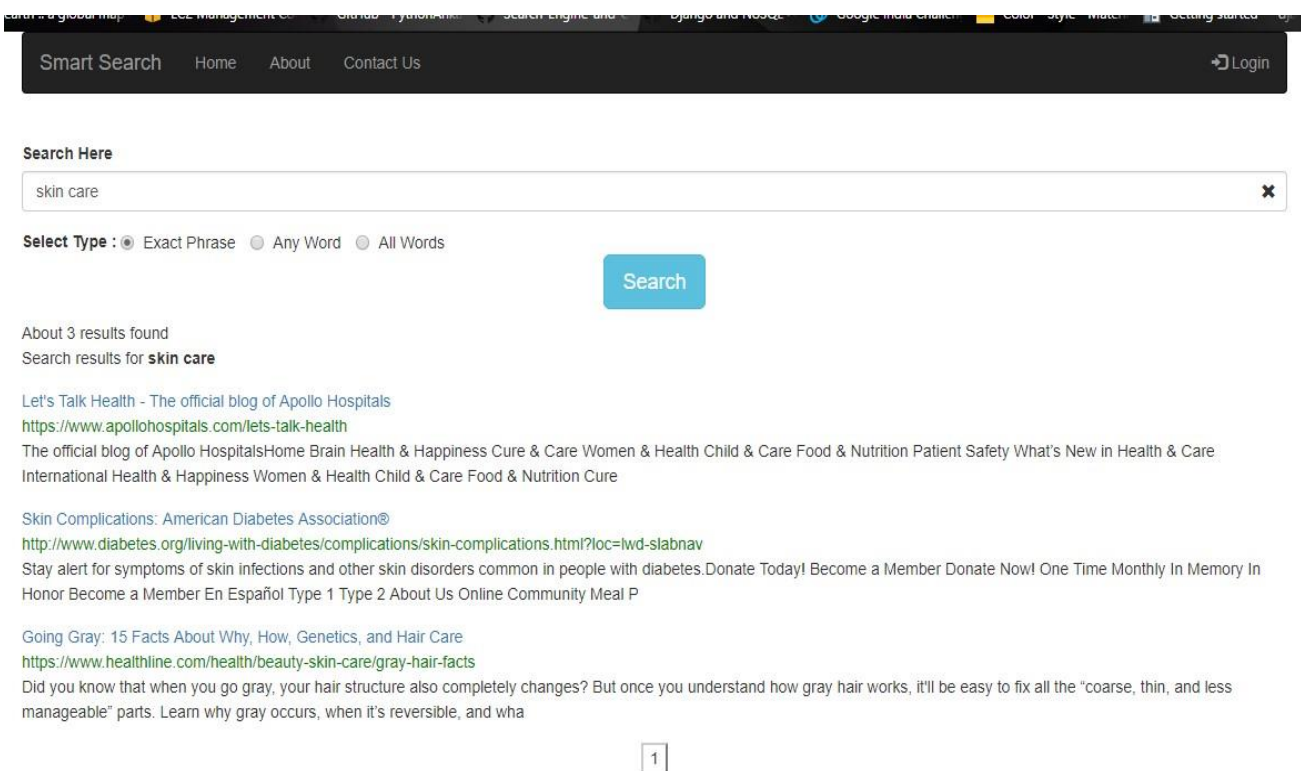


3.5 Home Page



The screenshot shows the top navigation bar with links for 'Smart Search', 'Home', 'About', and 'Contact Us', and a 'Login' button. Below the navigation bar is a search section titled 'Search Here' with a text input field containing the placeholder 'Enter your Query here'. Underneath the input field are three radio buttons for 'Select Type': 'Exact Phrase' (selected), 'Any Word', and 'All Words'. A blue 'Search' button is positioned to the right of the radio buttons.

Fig.3.5.1 Home Page



The screenshot shows the search results page. The search bar contains 'skin care' and has a close button (X). The 'Select Type' radio buttons are the same as in the previous screenshot. Below the search bar, it says 'About 3 results found' and 'Search results for skin care'. The first result is 'Let's Talk Health - The official blog of Apollo Hospitals' with the URL <https://www.apollohospitals.com/lets-talk-health>. The second result is 'Skin Complications: American Diabetes Association®' with the URL <http://www.diabetes.org/living-with-diabetes/complications/skin-complications.html?loc=lwd-slabnav>. The third result is 'Going Gray: 15 Facts About Why, How, Genetics, and Hair Care' with the URL <https://www.healthline.com/health/beauty-skin-care/gray-hair-facts>. At the bottom of the page, there is a small box containing the number '1'.

Fig.3.5.2 Search Results

Patient Speak Messages

<https://www.apollohospitals.com/patient-care/testimonial-messages>

Apollo Hospitals Ahmedabad Bengaluru Chennai Delhi Hyderabad Kolkata Mumbai Aragonda Bachel Bhubaneshwar Bilaspur Guwahati Indore Kakinada Karur Lavasa Madurai Mysore Nashik Nellore Pune Ranipet Thiruvannamalai Trichy Visakhapatnam Mobile Navig

Complete Guide on Diseases & Conditions - Apollo Hospitals

<https://www.apollohospitals.com/patient-care/health-and-lifestyle/diseases-and-conditions>

Apollo Hospitals offers a comprehensive overview of common diseases & conditions to increase your awareness of health and disease. Click on diseases to know more.

Let's Talk Health - The official blog of Apollo Hospitals

<https://www.apollohospitals.com/lets-talk-health>

The official blog of Apollo Hospitals Home Brain Health & Happiness Cure & Care Women & Health Child & Care Food & Nutrition Patient Safety What's New in Health & Care International Health & Happiness Women & Health Child & Care Food & Nutrition Cure

Best Super & Multispecialty Hospital in Navi Mumbai - Apollo Hospitals

<http://mumbai.apollohospitals.com>

Apollo Hospitals in Navi Mumbai is India's leading super speciality hospital. Our team of doctors provide the best of modern healthcare to ensure you stay healthy.

Apollo Health city in Hyderabad

https://hyderabad.apollohospitals.com/?utm_source=apollohospitals.com&utm_campaign=selectlocationtab&utm_medium=desktop

Apollo Health City is the leading Super Speciality hospitals in Hyderabad bring world-class healthcare treatment with highly qualified doctors for the best possible outcomes.

Best Hospital, Multi-Speciality Hospital in Chennai - Apollo Hospitals

<http://chennai.apollohospitals.com>

Apollo Hospitals Chennai is India's leading super speciality hospital. Our team of over 5000 doctors give you the best of modern healthcare to ensure you stay healthy.

Patient Information Guide

<https://www.apollohospitals.com/patient-care/patient-information-guide>

Apollo Hospitals Ahmedabad Bengaluru Chennai Delhi Hyderabad Kolkata Mumbai Aragonda Bachel Bhubaneshwar Bilaspur Guwahati Indore Kakinada Karur Lavasa Madurai Mysore Nashik Nellore Pune Ranipet Thiruvannamalai Trichy Visakhapatnam Mobile Navig

Fig.3.5.3 Search Results 2

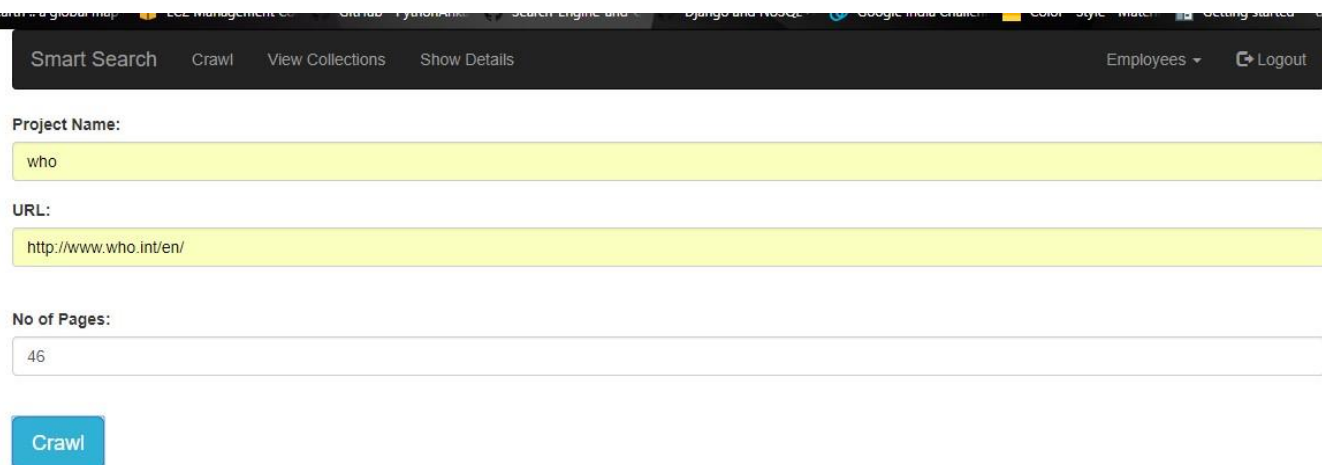


Fig.3.5.4 Admin Crawler

Smart Search Crawl View Collections Show Details Employees ▾ Logout

Project Name:

who

Submit

URL	Rank	Title
http://www.who.int/en/index.html	5.525193975199602	WHO World Health Organization
http://www.who.int/ar/index.html	3.701746700023844	منظمة الصحة العالمية منظمة الصحة العالمية
http://www.who.int/zh/index.html	3.701746700023844	世界卫生组织 世界卫生组织
http://www.who.int/fr/index.html	3.701746700023844	OMS Organisation mondiale de la Santé
http://www.who.int/ru/index.html	3.701746700023844	ВОЗ Всемирная организация здравоохранения
http://www.who.int/es/index.html	3.701746700023844	OMS Organización Mundial de la Salud
http://www.who.int/en	1.8932915525387	WHO World Health Organization
http://www.who.int/entity/en	1.8932915525387	WHO World Health Organization
http://www.who.int/entity/mediacentre/en	1.849296796910751	WHO WHO media centre: news, features, multimedia
http://www.who.int/about/en	1.833215955198467	WHO Who we are, what we do
http://www.who.int/topics/en	1.816189181620756	WHO Health topics
http://www.who.int/publications/en	1.816189181620756	WHO Publications
http://www.who.int/countries/en	1.816189181620756	WHO Countries
http://www.who.int/governance/en	1.816189181620756	WHO WHO's Governing Bodies
http://www.who.int/about/en/index.html	1.801054271773901	WHO Who we are, what we do
http://search.who.int/search?ie=utf8&site=who&lr=lang_en&client=_en_r&proxystylesheet=_en_r&output=xml_no_dtd&oe=UTF-8&accession=8&entry=3&uid=4&proxycustom=30ADVANCED%3F	1.754273641338171	Advanced search

Fig.3.5.5 Admin view results

Smart Search Crawl View Collections Show Details Employees ▾ Logout





Name	UserName	
dhruresh rajodiya	dhruresh	
Janam Desai	janam	
Prashan Kikani	kikani	
Vinit Pandya	vinit	

Fig.3.5.6 view employee

Smart Search Crawl View Collections Show Details Employees Logout

First Name:

Last Name:

User Name:

Password:

Confirm Password:

[Register Employee](#)

Fig.3.5.7 Add Employee

Project	Link	No. of Pages	Approve	Reject
diabetes	Go To Page	11	✓	✗
apollohospitals	Go To Page	5	✓	✗
cancer	Go To Page	5	✓	✗
drugs	Go To Page	5	✓	✗
everydayhealth	Go To Page	5	✓	✗
healthline	Go To Page	5	✓	✗
nih	Go To Page	5	✓	✗
freemedicaljournals	Go To Page	5	✓	✗
kidshealth	Go To Page	5	✓	✗

Fig.3.5.8 Approve Request

CHAPTER 4

IMPLEMENTATION DETAILS

4. Implementation

4.1 Implementation Environment

- Python environment in Windows 10 (version=3.6.2)
- MongoDB Client in Windows 10 (version=3.6.2)
- Apache solr in Windows 10 (version=7.2.1)

4.2 Description of Modules

➤ **Register / Log - In Module**

It is used to store information of employees who are new to the system and accessing the system for the first time and also authenticating the users before they can add link to the crawling job of the system.

Input : User's information or credentials

Output: Stored or Verified successfully

Processing: Check user's credentials in the database while logging in or store them in the database while registering new user.

➤ **Spider Module**

This module does crawling of urls from queue of pending crawling jobs.

Input : seed url's queue

Output: Crawled data stored in MongoDB

Processing: The spider first search in Db whether crawling of this project is started or is yet to be started. It then fetches the content of the current crawling page, we have configured it for fetching of tags like 'anchor'(it

gives reference of other pages from this page and gives keywords too), 'meta'(it gives details about content of that page), and 'title' tags. It then stores the anchor links fetched in the pending queue in MongoDB which are having domain name as the current working domain(Basically spider will not crawl out of the domain links). It then continues crawling by fetching next link from queue of current domain until maximum number of pages(links) is reached.

➤ **Indexer Module**

Input : Crawled data

Output: Indexed data stored at Apache solr's core.

Processing:

This module will be called in between any crawling job is finished and new job is to be started, this will index the data which is collected in previous crawling job. The indexing will be taken care by pysolr which is the light-weight wrapper for Apache solr.

➤ **Ranking Module**

Input : Crawled data

Output: Ranked pages stored in MongoDB

Processing:

This module will be called in between any crawling job is finished and new job is to be started, this will rank the pages of the domain which is just crawled. The ranking is based on page-rank algorithm which is described below. After ranking is completed it updates it in MongoDB for permanent storage.

➤ **Searching Module**

Input : Indexed data

Output: Most matching Urls.

Processing:

This module will be called when user search for any keyword in user section. This will find the documents which have the keyword in its fields, this finding is based on one of two options “any word” or “all words” ,there are different lucene queries for both of these options. After finding those records it will sort those records based on their ranks calculated in ranking module and display those urls with some description about content of that page(which was fetched from meta tags)

➤ **Admin Control Module**

Input : Pending projects for approvement **Output:**
Status after approvement.

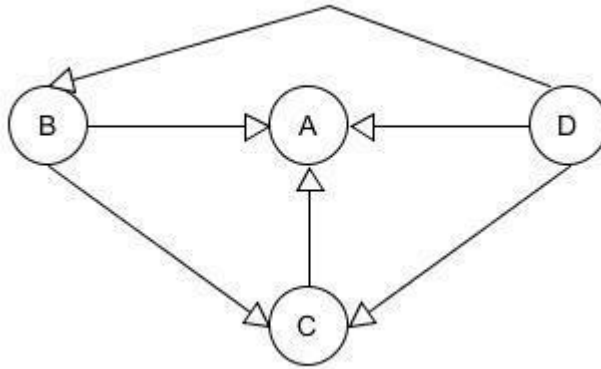
Processing:

This module is used by the administrator of the system in order to maintain control of the seed urls suggested by employees to add to crawling queue or not. Here, all the seed urls listed by the users come for verification .Thus, only after the admin approves the url, it can be listed for the crawling job. If admin rejects any url then it will be discarded. Admin can also modify number of pages to be crawled in particular job.

4.3 Algorithm

➤ Page Rank Algorithm

○ How Page Rank Works:



Assume a small universe of four web pages: **A**, **B**, **C** and **D**. The initial approximation of Page Rank would be evenly divided between these four documents. Hence, each document would begin with an estimated Page Rank of 0.25.

In the original form of Page Rank initial values were simply 1. This meant that the sum of all pages was the total number of pages on the web. Later versions of Page Rank would assume a probability distribution between 0 and 1. Here we're going to simply use a probability distribution hence the initial value of 0.25.

If pages **B**, **C**, and **D** each only link to **A**, they would each confer 0.25 Page Rank to **A**. All Page Rank i.e. $PR()$ in this simplistic system would thus gather to **A** because all links would be pointing to **A**.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

This is 0.75.

Again, suppose page **B** also has a link to page **C**, and page **D** has links to all three pages. The value of the link-votes is divided among all the outbound links on a page. Thus, page **B** gives a vote worth 0.125 to page **A** and a vote worth 0.125 to page **C**. Only one third of **D**'s Page Rank is counted for **A**'s Page Rank (approximately 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

In other words, the Page Rank conferred by an outbound link $L(v)$ is equal to the document's own Page Rank score divided by the normalized number of outbound links (it is assumed that links to specific URLs only count once per document).

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

In the general case, the Page Rank value for any page u can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

i.e. the Page Rank value for a page u is dependent on the Page Rank values for each page v out of the set B_u (this set contains all pages linking to page u), divided by the number $L(v)$ of links from page v .

CHAPTER 5

TESTING AND IMPLEMENTATION

5. Testing

5.1 Testing Plan

The testing is a technique that is going to be used in the project is black box testing ,the expected inputs to the system are applied and only the outputs are checked .

5.2 Testing Strategy

The development process repeats this testing sub process a number of the lines for the following phases.

- Unit Testing
- Integration Testing

Unit Testing tests a unit of code after coding of that unit is completed. Integration Testing tests whether the previous programs that make up a system, interface with each other as desired. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification.

Testing is carried out in such a hierarchical manner to that each component is correct and the assembly/combination of component is correct. Merely testing a whole system at end would most likely throw up errors in component that would be very costly to trace and fix. We have performed both Unit Testing and System Testing to detect and fix errors.

5.3 Testing Methods

We have performed Black-box testing for the testing

purpose. A brief description is given below:

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

5.4 Test Cases

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results
T01	Sign Up	<ol style="list-style-type: none"> 1. Go to Home page 2. Provide Information 	User Information	Stored Successfully	Success
T02	Log In	<ol style="list-style-type: none"> 1. Go to Home page 2. Enter Credentials 	User Credentials	Display Main Page	Main Page Displayed
T03	Add Employee	<ol style="list-style-type: none"> 1. Go to Main Page 2. Go to register employee 3. Enter 	Employee Information	Employee information stored successfully in database Success	Success

		empl oyee Detail s			
T04	View collection	1.Go to Views collection page 2.Enter Domain name as Project name	Collection of crawled data	Displays the data based on given project name.	Result Shown accordingl y
T05	Crawl	1. Go to Crawl Page 2. Provi de The proje ct name, doma in link and no pages to be crawl	Crawling start and request id generated	It Shows the request id Generated and Send email when crawling will done successfully.	Success
T06	Show details	1. Go to Show detail s page 2. Enter the doma in name of the link	Display Output data	It will list all the crawled link on that domain and it's rank and titles also	Successfu lly Displayed

		as proje ct name			
T07	Search Keyword	1. Home page 2. Enter any Keyw ord	Matched links listed	Display all the links related with input keyword from our database.	Displayed Successfu lly

CHAPTER 6

Limitations and Future Enhancements

6. Limitations and Future Enhancements

6.1 Limitations

- System is not able to provide the dynamic ranking based.
- If re-crawling of any website from scratch is needed then, admin has to delete previous record manually.

6.2 Future Enhancements

- Dynamic ranking of pages based on user's interest.
- “You may also like” feature in search section.

CHAPTER 7

CONCLUSION

7. Conclusion

Hereby, we declare that the functionality implemented in Web Crawler Supported Smart Search was performed by understanding all the modules. We have implemented crawling with solr indexing and ranking of pages. The searching facility is also provided with three options (Exact phrase, Any Word, All Words). After the coding was completed, comprehensive testing was performed and the results were provided in the report. Unit Testing of all modules were done and later, Integration Testing was also performed.