**Building Search Engine Using Machine Learning Technique**

**1. INTRODUCTION:**

World Wide Web is actually a web of individual systems and servers which are connected with different technology and methods. Every site comprises of the heaps of site pages that are being made and sent on the server. So if a user needs something, then he or she needs to type a keyword. Keyword is a set of words extracted from user search input. Search input given by user may be syntactically incorrect. Here comes the actual need for search engines. Search engines provide you a simple interface to search user query and display the results in the form of the web address of the relevant web page. Figure 1 focuses on three main components of search engine. 1) Web crawler Web crawlers help in collecting data about a website and the links related to them. We are only using web crawler for collecting data and information from WWW and store it to our database. 2) Indexer Indexer which arranges each term on each web page and stores the subsequent list of terms in a tremendous repository. 3) Query Engine It is mainly used to reply the user’s keyword and show the effective outcome for their keyword. In query engine, Page ranking algorithm ranks the URL by using different algorithms in the query engine. Fig. 1. Block Diagram of Search Engine [1] This paper utilizes Machine Learning Techniques to discover the utmost suitable web address for the given keyword. The output of PageRank algorithm is given as input to machine learning algorithm. The section II discusses the related work in search engine and PageRank algorithm. In section III Objective is explained. Section IV deals with proposed system which is based on machine learning technique and section V contains the conclusion.

**1.1 Objective of the project:**

The web is the huge and most extravagant wellspring of data. To recover the information from World Wide Web, Search Engines are commonly utilized. Search engines provide a simple interface for searching for user query and displaying results in the form of the web address of the relevant web page, but using traditional search engines has become very challenging to obtain suitable information. This paper proposed search engine using Machine Learning technique that will give more relevant web pages at top for user queries.

**2. LITERATURE SURVEY:**

**“A Review Paper on Various Search Engines (Google, Yahoo, Altavista, Ask and Bing)”,**

Search engines are used in the web as a tool for information retrieval. As the web is a huge repository of heterogeneous and unstructured data so to filter out relevant information from unnecessary ones search engines are needed. Search engines usually consists of crawling module, page repository, indexing module, querying module and ranking module. The inter communication between these modules describes the working methodology of a search engine. This paper aims to focus on the comparative analysis of five major search engines i.e Google, Yahoo, Altavista, Ask and Bing in a tabular form based on some features. The features include search operator, search web, search images, search videos, search news, search maps, search books, advance search, change background, change search settings, display number of results, shopping, translation services, multi-language support, questions/answers, directory, advertising programs, business services, themes, case sensitive, finance, safe search, search pad, career and preferences. Google stands out as the best search engine amongst all search engines, which works on Page Rank algorithm. Page Rank is a numeric value which determines the importance of a web page by calculating the number of backlinks.

“**Keyword Focused Web Crawler”**

Users and uses of internet is growing tremendously these days which causing an extreme trouble and efforts at user side to get web pages searched which are as per concern and relevant to user's requirement Generally users approach to search web pages from a large available hierarchy of concepts or use a query to browse web pages from available search engine and receive results based on search pattern where few of the results are relevant to search and most of them are not. Web crawler plays an important role in search engine and act as a key element when performance is considered. This paper includes domain engineering concept and keyword driven crawling with relevancy decision mechanism and uses Ontology concepts which ensures the best path for improving crawler's performance. This paper introduces extraction of URLs based on keyword or search criteria. It extracts URLs for web pages which contains searched keyword in their content and considers such pages only as important and doesn't download web pages irrelevant to search. It offers high optimality comparing with traditional web crawler and can enhance search efficiency with more accuracy.

**“Contrastive Study of Simple PageRank, HITS and Weighted PageRank Algorithms: Review”**

Web mining is a very important research subject. It's basically an application of data mining to find concealed information on web. Internet has been providing us with boundless source of information according to our need. In recent times search tools have emerged as one of the requisite tools for person who do navigation on net or rely on web. But with expanding usage of net, it is stretching hastily in its material. With this reckless augmentation in information material, there comes a daunting task in organizing the information according to people's demands. The plight is like “drowning in data but starving for knowledge”. So to avoid the challenging scenario we have techniques to extract or filter information which have great relevance to user's query. This paper actually deals with some of those algorithms and their comparative exploration based on various parameters which will succeed in removing difficulty in ranking appropriate content to user. Techniques that has been discussed here with apt example are Simple PageRank which is based on link structure mainly forward links mainly followed by Google after that Weighted PageRank has been explained which also based on link approach but here both backward and forward links are used to rank the pages, finally HITS (Hypertext Induced Topic Search) has been scrutinized which work on both content and link structure ofweb.

**“A machine learning approach to web page filtering using content and structure analysis”**

As the Web continues to grow, it has become increasingly difficult to search for relevant information using traditional search engines. Topic-specific search engines provide an alternative way to support efficient information retrieval on the Web by providing more precise and customized searching in various domains. However, developers of topic-specific search engines need to address two issues: how to locate relevant documents (URLs) on the Web and how to filter out irrelevant documents from a set of documents collected from the Web. This paper reports our research in addressing the second issue. We propose a machine-learning-based approach that combines Web content analysis and Web structure analysis. We represent each Web page by a set of content-based and link-based features, which can be used as the input for various machine learning algorithms. The proposed approach was implemented using both a feedforward/backpropagation neural network and a support vector machine. Two experiments were designed and conducted to compare the proposed Web-feature approach with two existing Web page filtering methods — a keyword-based approach and a lexicon-based approach. The experimental results showed that the proposed approach in general performed better than the benchmark approaches, especially when the number of training documents was small. The proposed approaches can be applied in topic-specific search engine development and other Web applications such as Web content management.

**“Comparative Study of Page Rank and Weighted Page Rank Algorithm”,**

Nowadays, it has been seen that the upcoming information system is the World Wide Web. Most of the web content is not structured so collecting and analyzing such data is very tedious. Web servers worldwide generate a huge amount of information depending upon web users’ browsing activities. It therefore becomes difficult for the user to select the relevant information from the results displayed by the search engines after performing various steps. In this case, Web page ranking algorithms play an important role in ranking web pages so that the user could retrieve the page which is most related to the user’s query. Some page ranking algorithms are HITS, Sensitive PageRank Algorithm, PageRank and weighted PageRank. In this paper, two popular web page ranking algorithms are compared which are: Weighted PageRank algorithm and PageRank algorithm. The paper highlights their difference, strengths, weaknesses.

**“Page Ranking Algorithms – A Comparison”,**

The growth in the number of websites has been increasing tremendously over the years and the data over the web has been increasing accordingly. Retrieving the required information from the web thereby fulfilling the needs of the web user has become a challenging job for website owners. This paper looks into the insights of the various ranking algorithms and their comparative study.

**“Comparative Analysis of Different Page Ranking Algorithms”**

Search engine plays an important role in internet, to retrieve the relevant documents among the huge number of web pages. However, it retrieves more number of documents, which are all relevant to your search topics. To retrieve the most meaningful documents related to search topics, ranking algorithm is used in information retrieval technique. One of the issues in data miming is ranking the retrieved document. In information retrieval the ranking is one of the practical problems. This paper includes various Page Ranking algorithms, page segmentation algorithms and compares those algorithms used for Information Retrieval. Diverse Page Rank based algorithms like Page Rank (PR), Weighted Page Rank (WPR), Weight Page Content Rank (WPCR), Hyperlink Induced Topic Selection (HITS), Distance Rank, Eigen Rumor, Distance Rank Time Rank, Tag Rank, Relational Based Page Rank and Query Dependent Ranking algorithms are discussed and compared

**“A Comparative Analysis of Web Page Ranking Algorithms”**

Web is expending day by day and people generally rely on search engine to explore the web. In such a scenario it is the duty of service provider to provide proper, relevant and quality information to the internet user against their query submitted to the search engine. It is a challenge for service provider to provide proper, relevant and quality information to the internet user by using the web page contents and hyperlink between the web pages. This paperdeals with analysis and comparison of web page ranking algorithms based on various parameter to find out their advantages and limitations for the ranking of the web pages. Based on the analysis of different web page ranking algorithms, a comparative study is done to find out their relative strengths and limitations to find out the further scope of research in web page ranking algorithm.

**“Web Page Ranking Using Machine Learning Approach”**

This article gives an overview of the currently available literature on web page ranking algorithm using machine learning. Web page ranking algorithm, a well-known approach to rank the web pages available on cyber world. It helps us to know -- how the search engine exactly works and how a machine learn itself while giving priority to the page that which page is important to successfully fulfills the user query need and which page is worth less. Machine learning approach also helps us in understanding the complex part of page priority criteria in most popular search engines like Google, yahoo, AltaVista, dog pile and many more search engines like that. Page ranking mainly unrevealed the structure of web. This article gives an overview of available literature in the field of web page ranking algorithm and it also highlights the main point based on machine leaning approach in web page ranking algorithm.

**“Wikipedia Search Engine: Interactive Information Retrieval Interface Design”**

As information becomes more ubiquitous and the demands that searchers have on search systems grow, there is a need to support search behaviors beyond simple lookup. Information seeking is the process or activity of attempting to obtain information in both human and technological contexts. *Exploratory search* describes an information-seeking problem context that is open-ended, persistent, and multifaceted, and information-seeking processes that are opportunistic, iterative, and multitactical. Exploratory searchers aim to solve complex problems and develop enhanced mental capacities. Exploratory search systems support this through symbiotic human-machine relationships that provide guidance in exploring unfamiliar information landscapes.

Exploratory search has gained prominence in recent years. There is an increased interest from the information retrieval, information science, and human-computer interaction communities in moving beyond the traditional turn-taking interaction model supported by major Web search engines, and toward support for human intelligence amplification and information use. In this lecture, we introduce exploratory search, relate it to relevant extant research, outline the features of exploratory search systems, discuss the evaluation of these systems, and suggest some future directions for supporting exploratory search. Exploratory search is a new frontier in the search domain and is becoming increasingly important in shaping our future world.

Table of Contents: Introduction / Defining Exploratory Search / Related Work / Features of Exploratory Search Systems / Evaluation of Exploratory Search Systems / Future Directions and concluding Remarks

**3. SYSTEM ANALYSIS**

**3.1 Existing system:**

Keyword based Search engines are unable to give relevant search results as they do not know the exact meaning of the keywords used. This paper compares both keyword and semantic search engines. Semantic Web is advanced version of the current web. It represents information meaningfully for machines and humans.

**Disadvantages:**

* Complex queries or words that have dual meaning.
* Long search queries.
* Users not familiar with important keywords to retrieve best results.

**Proposed system:**

World Wide Web is actually a web of individual systems and servers which are connected with different technology and methods. Every site comprises of the heaps of site pages that are being made and sent on the server. So if a user needs something, then he or she needs to type a keyword. Keyword is a set of words extracted from user search input. Search input given by user may be syntactically incorrect. Here comes the actual need for search engines. Search engines provide you a simple interface to search user query and display the results in the form of the web address of the relevant web page.

**Advantages:**

* It give a user more relevant web address for user query
* it works intelligently,
* it works efficiently
* relevant web page at the top of result, according to user need

**Modules Information:**

To implement this project author has designed following modules

In this paper author has implemented following modules

1. Admin module: admin can login to application using username and password as admin and then accept or activate new users registration and then train SVM and XGBOOST algorithm
2. Manager module: manager can login to application by using username and password as Manager and Manager and then upload dataset to application
3. New User Signup: using this module new user can signup with the application
4. User Login: user can login to application and then perform search by giving query.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

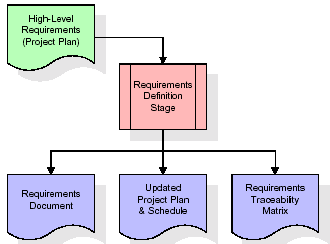
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

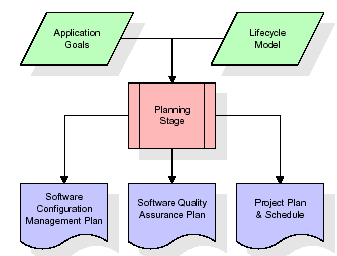
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

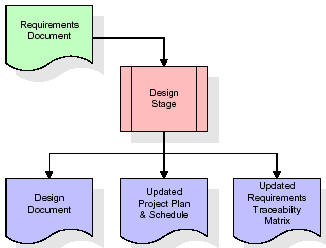
The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

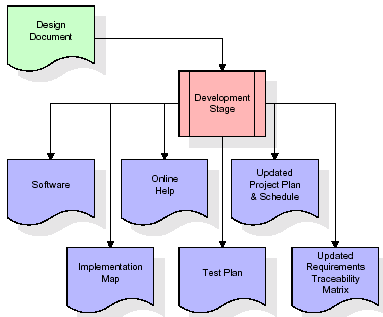
**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

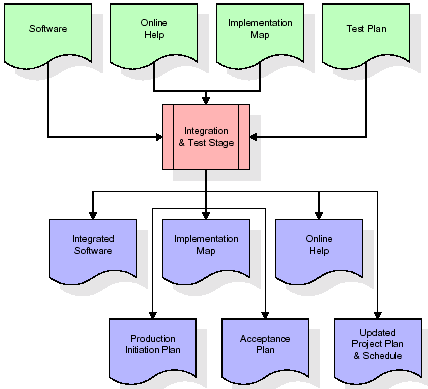
The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

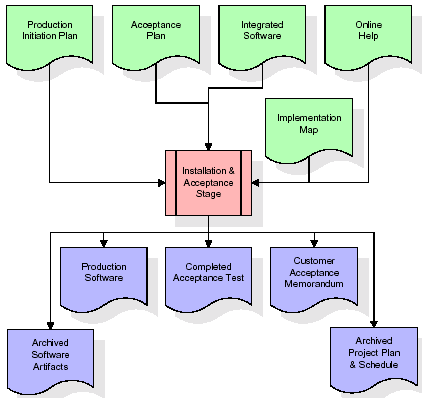


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python

**4. SYSTEM DESIGN**

**4. SYSTEM DESIGN**

**CLASS DIAGRAM:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

****

**USECASE DIAGRAM:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**SEQUENCE DIAGRAM:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**COLLABORATION DIAGRAM:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.

****

**COMPONENT DIAGRAM:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**DEPLOYMENT DIAGRAM:**

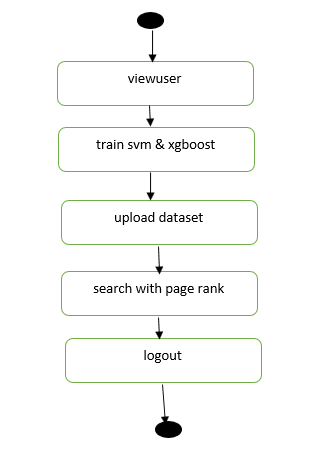
A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**ACTIVITY DIAGRAM:**

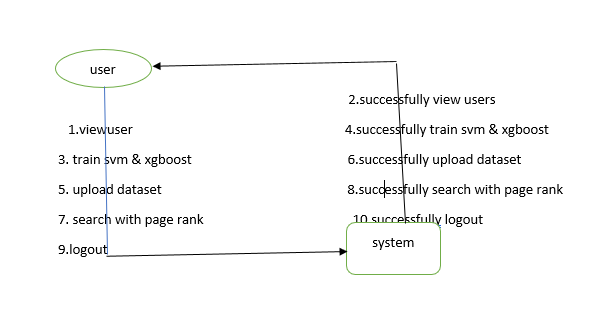
Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent.



**Data flow :**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.



**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import os

from django.core.files.storage import FileSystemStorage

import pymysql

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import accuracy\_score

from xgboost import XGBClassifier

from sklearn import svm

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from string import punctuation

from nltk.corpus import stopwords

import nltk

from nltk.stem import WordNetLemmatizer

from sklearn.feature\_extraction.text import TfidfVectorizer

import numpy as np

from numpy import dot

from numpy.linalg import norm

global uname

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

def cleanNews(doc):

tokens = doc.split()

table = str.maketrans('', '', punctuation)

tokens = [w.translate(table) for w in tokens]

tokens = [word for word in tokens if word.isalpha()]

tokens = [w for w in tokens if not w in stop\_words]

tokens = [word for word in tokens if len(word) > 1]

tokens = [lemmatizer.lemmatize(token) for token in tokens]

tokens = ' '.join(tokens)

return tokens

X = np.load("model/X.npy")

Y = np.load("model/Y.npy")

URLS = np.load("model/URLS.npy")

tfidf\_vectorizer = TfidfVectorizer(stop\_words=stop\_words, use\_idf=True, smooth\_idf=False, norm=None, decode\_error='replace', max\_features=3000)

tfidf = tfidf\_vectorizer.fit\_transform(X).toarray()

df = pd.DataFrame(tfidf, columns=tfidf\_vectorizer.get\_feature\_names())

print(str(df))

print(df.shape)

X = df.values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

def Train(request):

if request.method == 'GET':

output = ''

font = '<font size='' color=black>'

arr = ['Algorithm Name','Accuracy','Precision','Recall','FSCORE']

output += '<table border="1" align="center"><tr>'

for i in range(len(arr)):

output += '<th><font size="" color="black">'+arr[i]+'</th>'

output += "</tr>"

svm\_cls = svm.SVC()

svm\_cls.fit(X, Y)

predict = svm\_cls.predict(X\_test)

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

a = accuracy\_score(y\_test,predict)\*100

output += '<tr><td><font size="" color="black">SVM</td><td><font size="" color="black">'+str(a)+'</td><td><font size="" color="black">'+str(p)+'</td><td><font size="" color="black">'+str(r)+'</td><td><font size="" color="black">'+str(f)+'</td></tr>'

xgb\_cls = XGBClassifier()

xgb\_cls.fit(X, Y)

predict = xgb\_cls.predict(X\_test)

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

a = accuracy\_score(y\_test,predict)\*100

output += '<tr><td><font size="" color="black">XGBoost</td><td><font size="" color="black">'+str(a)+'</td><td><font size="" color="black">'+str(p)+'</td><td><font size="" color="black">'+str(r)+'</td><td><font size="" color="black">'+str(f)+'</td></tr>'

context= {'data':output}

return render(request, 'ViewUsers.html', context)

def VerifyUser(request):

if request.method == 'GET':

global uname

username = request.GET['t1']

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'searchengine',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "update signup set status='Accepted' where username='"+username+"'"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

output = username+" account activated"

context= {'data':output}

return render(request, 'AdminScreen.html', context)

def SearchQueryAction(request):

if request.method == 'POST':

query = request.POST.get('t1', False)

qry = query

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

arr = ['Query','Search URL','Rating']

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

query = query.strip().lower()

query = cleanNews(query)

vector = tfidf\_vectorizer.transform([query]).toarray()

vector = vector.ravel()

for i in range(len(X)):

score = dot(X[i], vector)/(norm(X[i])\*norm(vector))

if score > 0.2:

output += "<tr><td>"+font+qry+"</td>"

output += '<td><a href="'+URLS[i]+'" target="\_blank">'+font+URLS[i]+"</td>"

output += "<td>"+font+str(score)+"</td>"

context= {'data':output}

return render(request, 'ViewOutput.html', context)

def ViewUsers(request):

if request.method == 'GET':

global uname

output = '<table border=1 align=center width=100%>'

font = '<font size="" color="black">'

arr = ['Username','Password','Contact No','Gender','Email Address','Address','Status']

output += "<tr>"

for i in range(len(arr)):

output += "<th>"+font+arr[i]+"</th>"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'searchengine',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select \* FROM signup")

rows = cur.fetchall()

for row in rows:

username = row[0]

password = row[1]

contact = row[2]

gender = row[3]

email = row[4]

address = row[5]

status = row[6]

output += "<tr><td>"+font+str(username)+"</td>"

output += "<td>"+font+password+"</td>"

output += "<td>"+font+contact+"</td>"

output += "<td>"+font+gender+"</td>"

output += "<td>"+font+email+"</td>"

output += "<td>"+font+address+"</td>"

if status == 'Pending':

output += '<td><a href="VerifyUser?t1='+username+'">Click Here</a></td>'

else:

output += "<td>"+font+status+"</td>"

context= {'data':output}

return render(request, 'ViewUsers.html', context)

def UploadDatasetAction(request):

if request.method == 'POST':

global uname

dataset = request.FILES['t1']

dataset\_name = request.FILES['t1'].name

fs = FileSystemStorage()

fs.save('SearchEngineApp/static/files/'+dataset\_name, dataset)

output = dataset\_name+' saved in database'

context= {'data':output}

return render(request, 'UploadDataset.html', context)

def UploadDataset(request):

if request.method == 'GET':

return render(request, 'UploadDataset.html', {})

def SearchQuery(request):

if request.method == 'GET':

return render(request, 'SearchQuery.html', {})

def UserLogin(request):

if request.method == 'GET':

return render(request, 'UserLogin.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def AdminLogin(request):

if request.method == 'GET':

return render(request, 'AdminLogin.html', {})

def ManagerLogin(request):

if request.method == 'GET':

return render(request, 'ManagerLogin.html', {})

def Signup(request):

if request.method == 'GET':

return render(request, 'Signup.html', {})

def AdminLoginAction(request):

global uname

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

if username == 'admin' and password == 'admin':

uname = username

context= {'data':'welcome '+username}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':'login failed'}

return render(request, 'AdminLogin.html', context)

def ManagerLoginAction(request):

global uname

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

if username == 'Manager' and password == 'Manager':

uname = username

context= {'data':'welcome '+uname}

return render(request, 'ManagerScreen.html', context)

else:

context= {'data':'login failed'}

return render(request, 'ManagerLogin.html', context)

def UserLoginAction(request):

global uname

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

index = 0

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'searchengine',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username,password, status FROM signup")

rows = cur.fetchall()

for row in rows:

if row[0] == username and password == row[1] and row[2] == "Accepted":

uname = username

index = 1

break

if index == 1:

context= {'data':'welcome '+uname}

return render(request, 'UserScreen.html', context)

else:

context= {'data':'login failed or account not activated by admin'}

return render(request, 'UserLogin.html', context)

def SignupAction(request):

if request.method == 'POST':

username = request.POST.get('t1', False)

password = request.POST.get('t2', False)

contact = request.POST.get('t3', False)

gender = request.POST.get('t4', False)

email = request.POST.get('t5', False)

address = request.POST.get('t6', False)

output = "none"

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'searchengine',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select username FROM signup")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

output = username+" Username already exists"

break

if output == 'none':

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'searchengine',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "INSERT INTO signup(username,password,contact\_no,gender,email,address,status) VALUES('"+username+"','"+password+"','"+contact+"','"+gender+"','"+email+"','"+address+"','Pending')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

output = 'Signup Process Completed'

context= {'data':output}

return render(request, 'Signup.html', context)

**6. TESTING:**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | | **Actual** |
| 01 | Admin module | Verify  Admin module or not | If Admin module may not upload | we cannot do any further operations | we can do further operations | | High | High |
| 02 | Manager module | Verify Manager module or not | If Manager module n may not be Done | we cannot do any further operations | we can do further operations | | High | High |
| 03 | New User Signup | Verify New User Signup or not | If New User Signup may not be done | we cannot do any further operations | we can do further operations | | High | High |
| 04 | User Login | Verify User Login Model or not | If User Login may not Run | We cannot run  operation | We can Run the Operation | | High | High |

**FUNCTIONAL REQUIREMENTS:**

**1.SOFTWARE REQIREMENTS:**

**System Atributes:**

Uname

**Data base Requirements:**

No need

**2.USECASE:**

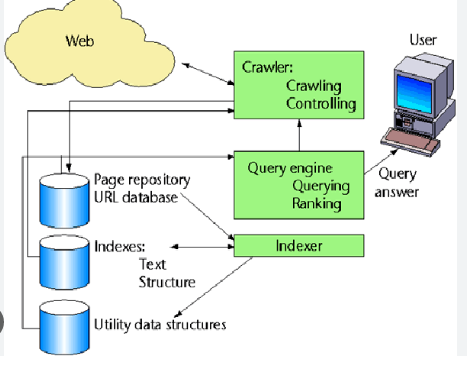
* Use cases - Use cases describe the interaction between the system and external users that leads to achieving particular goals.
* Each use case includes main elements:

1. Admin module
2. Manager module
3. New User Signup
4. User Logi**n**

**3.User Stories:**

In this paper author is using machine learning algorithms called SVM and XGBOOST to predict search result of given query and building search engine with machine learning algorithms. To train this algorithm author is using website data and then this data will be converted to numeric vector called TFIDF (term frequency inverse document frequency). TFIDF vector contains average frequency of each words

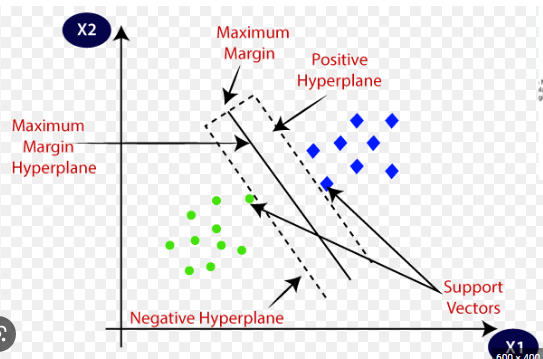
**4.Work down Structure:**

****

**5.Prototype:**

* python 3.7.0 or 3.7.4
* opencv-python==4.5.1.48
* keras==2.3.1
* tensorflow==1.14.0
* protobuf==3.16.0
* h5py==2.10.0
* sklearn-extensions==0.0.2
* scikit-learn==0.22.2.post1
* Numpy
* Pandas

**6.Models and Diagrams:**

****

**NON-FUNCTIONAL REQUIREMENT:**

1. **Usability:**  Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.(how it was handle entire project easy)
2. **Security:**the quality or state of being secure: such as. a : freedom from danger : safety. b : freedom from fear or anxiety. c : freedom from the prospect of being laid off job security.(
3. **Readability**: Readability is the ease with which a reader can understand a written text.
4. **Performance**: the execution of an action. : something accomplished : deed, feat. : the fulfillment of a claim, promise, or request : implementation. 3. : the action of representing a character in a play.
5. **Availability**: the quality or state of being available trying to improve the availability of affordable housing. 2 : an available person or thing.
6. **Scalability:**Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

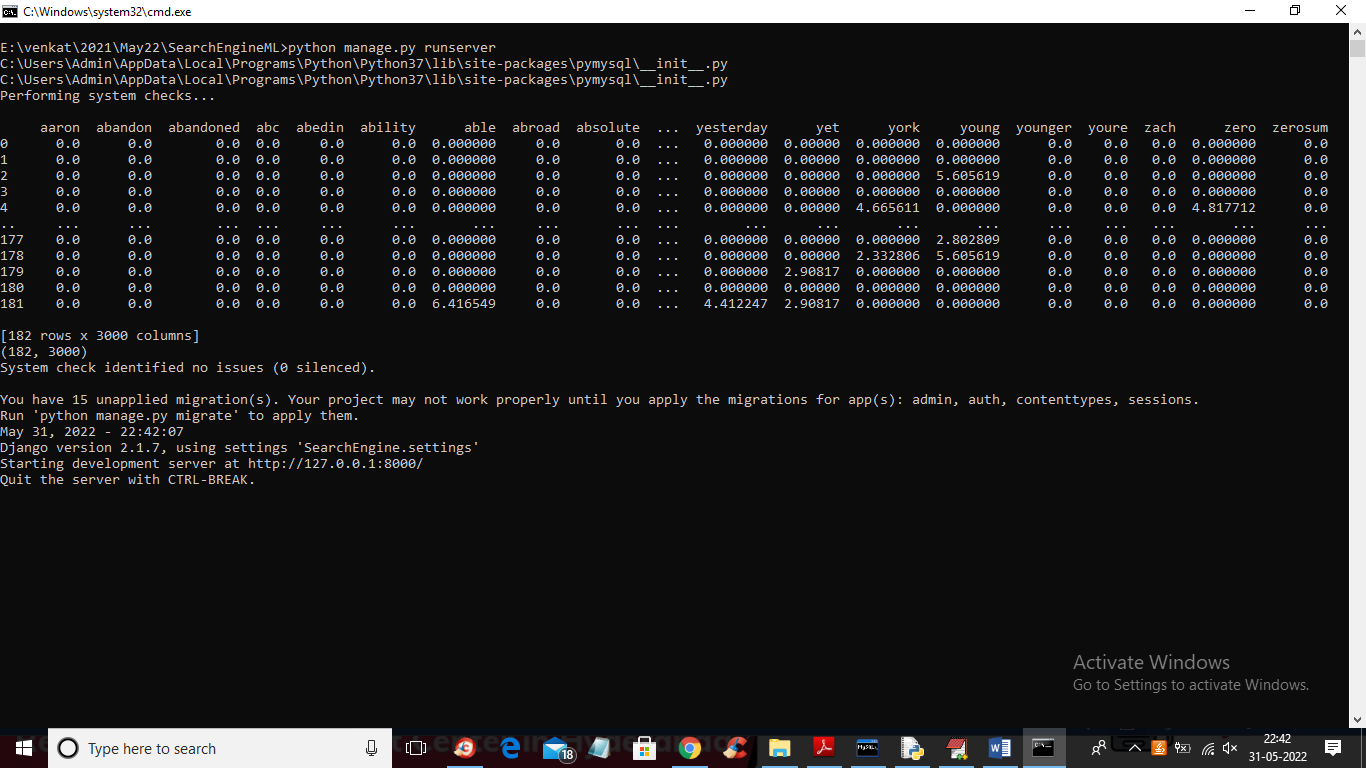
**7. SCREENSHOTS:**

Building Search Engine Using Machine Learning Technique

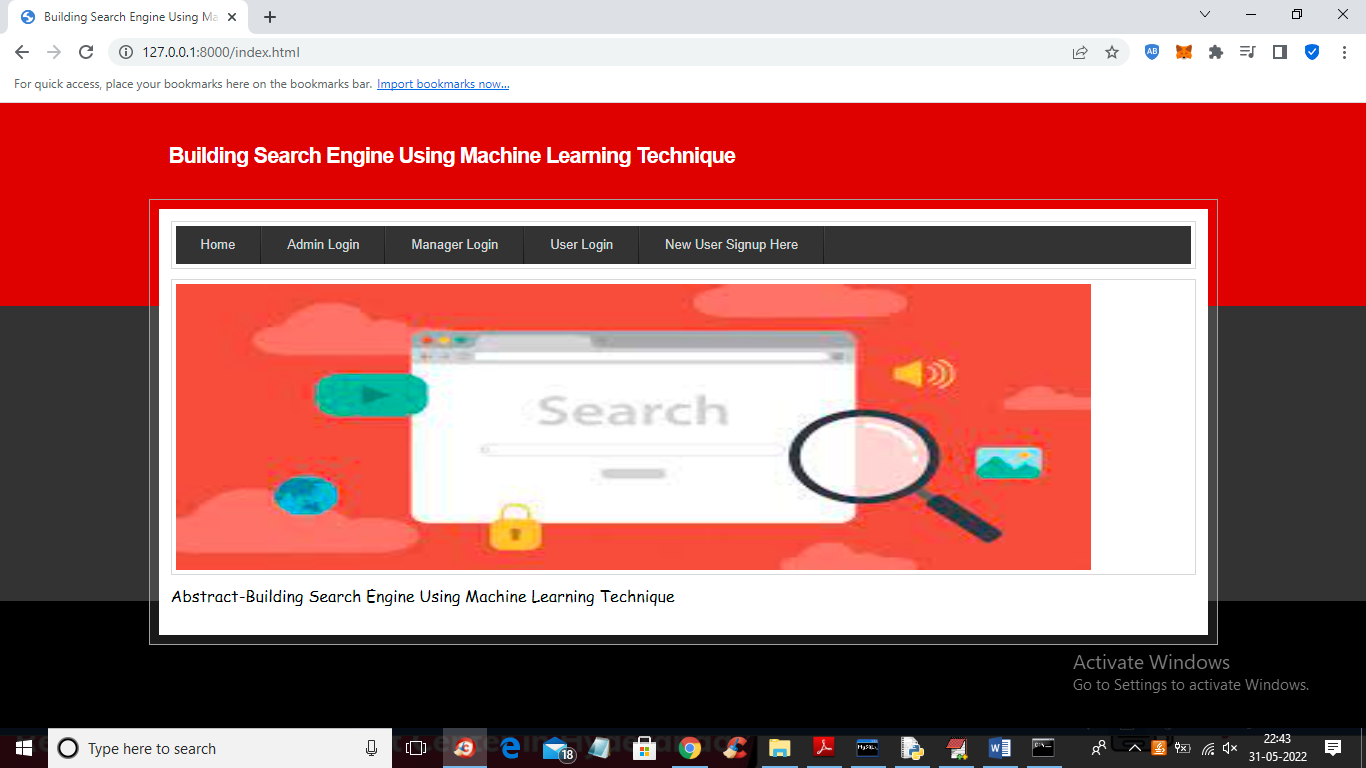
In this paper author is using machine learning algorithms called SVM and XGBOOST to predict search result of given query and building search engine with machine learning algorithms. To train this algorithm author is using website data and then this data will be converted to numeric vector called TFIDF (term frequency inverse document frequency). TFIDF vector contains average frequency of each words.

To run project install MYSQL and python 3.7 and then copy content from DB.txt file and paste in MYSQL to create database.

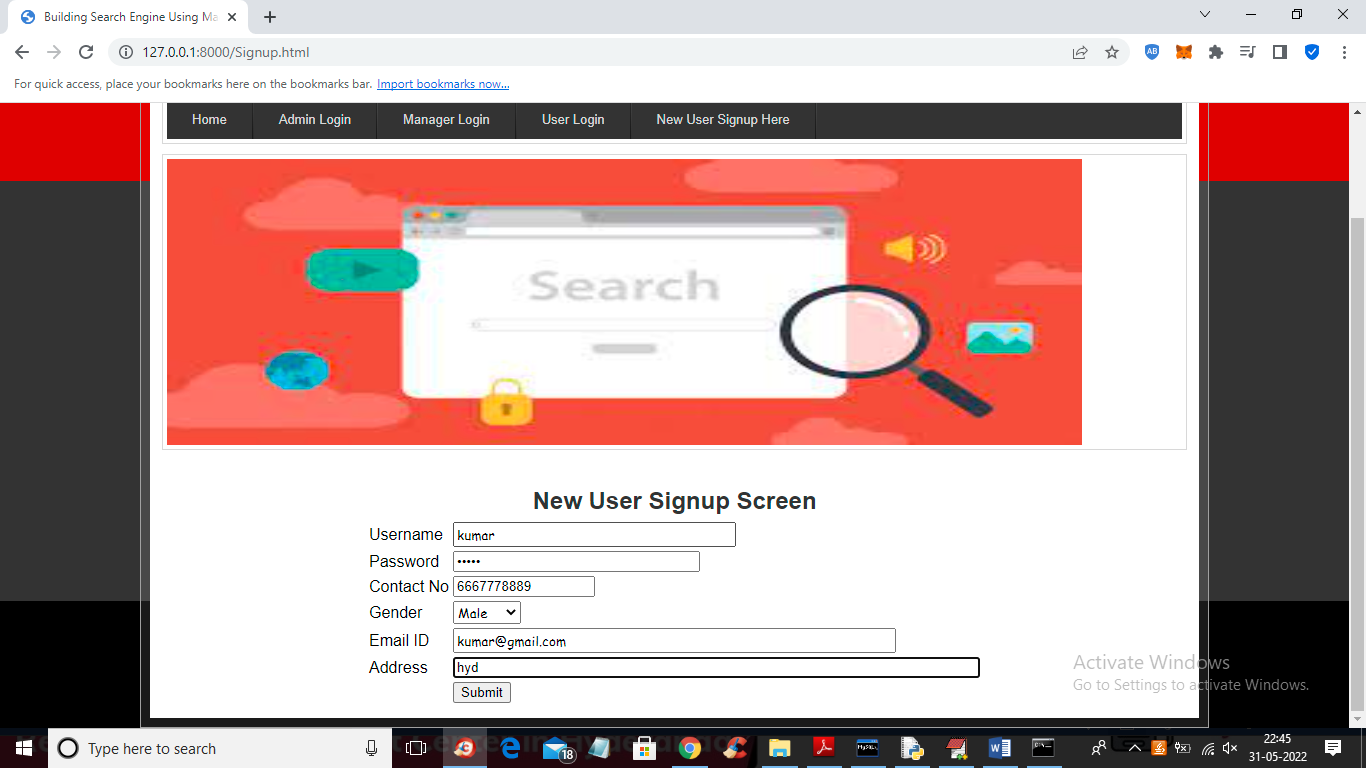
Now double click on ‘run.bat’ file to start python DJANGO server and get below screen



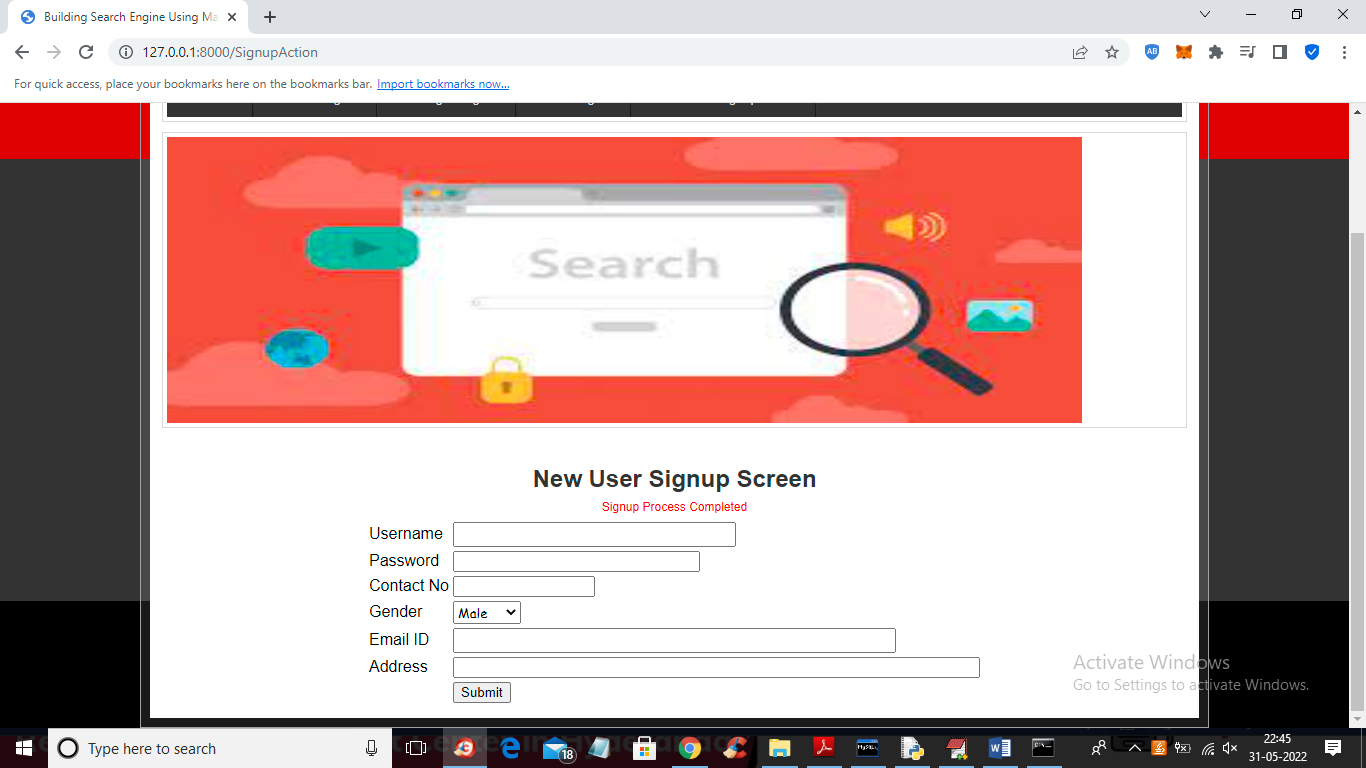
In above screen server started and build a vector from dataset where first row showing word and remaining rows contains TFIDF word frequency. Now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



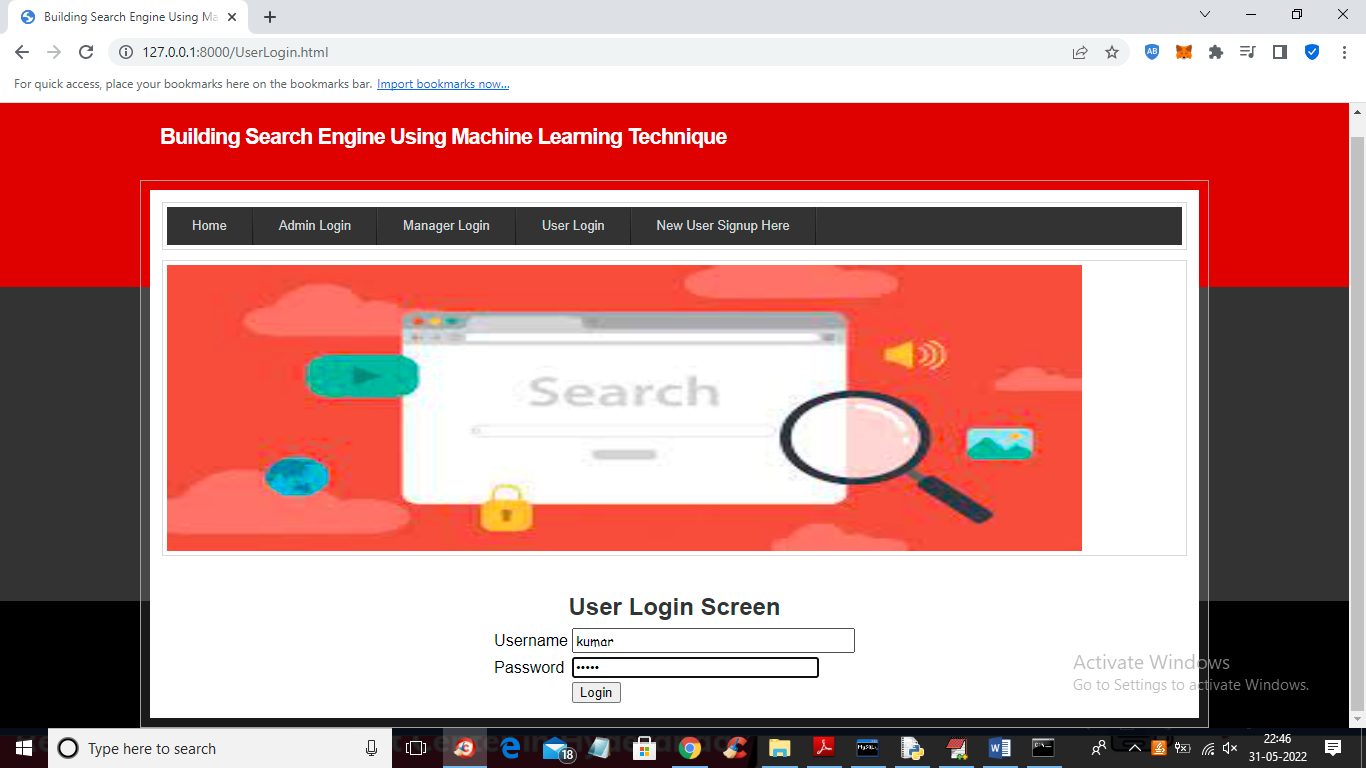
In above screen click on ‘New User Signup Here’ link to get below screen



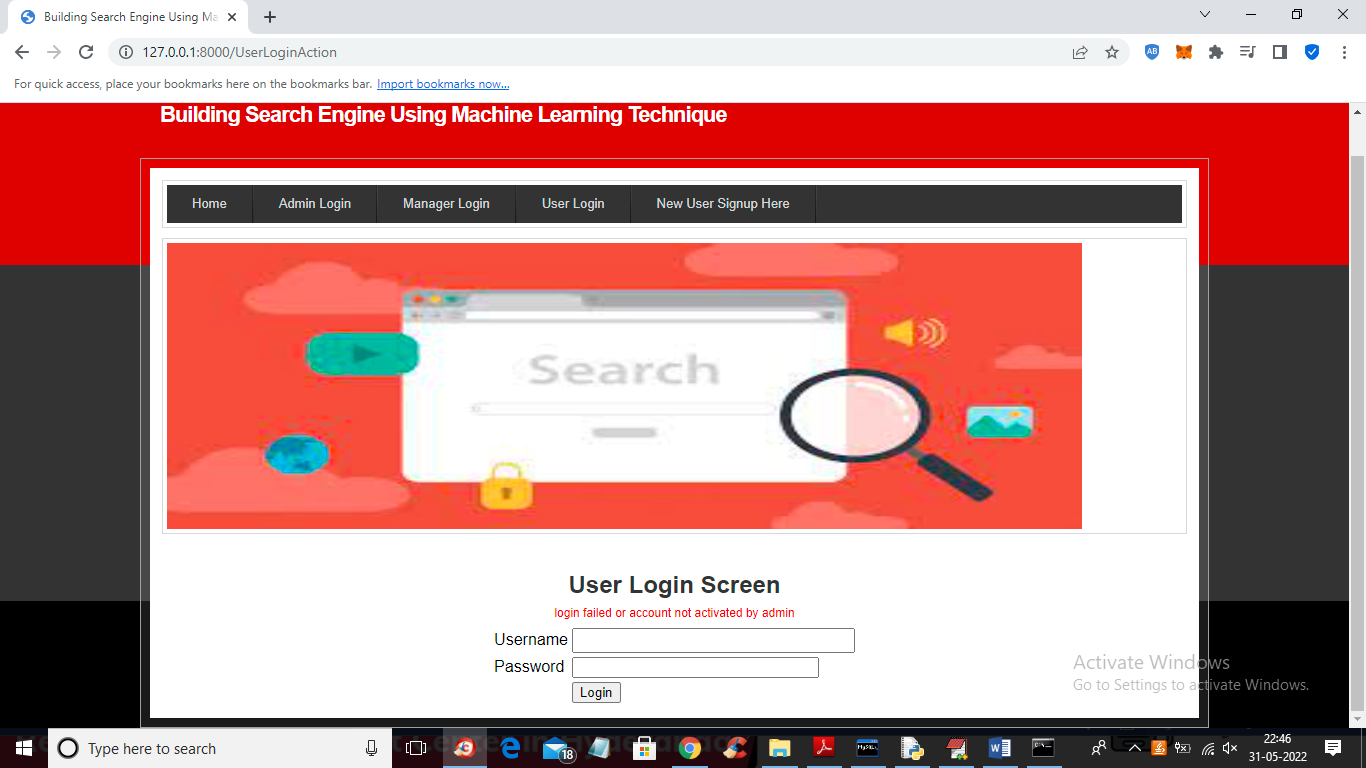
In above screen user is signing up and then press button to get below output



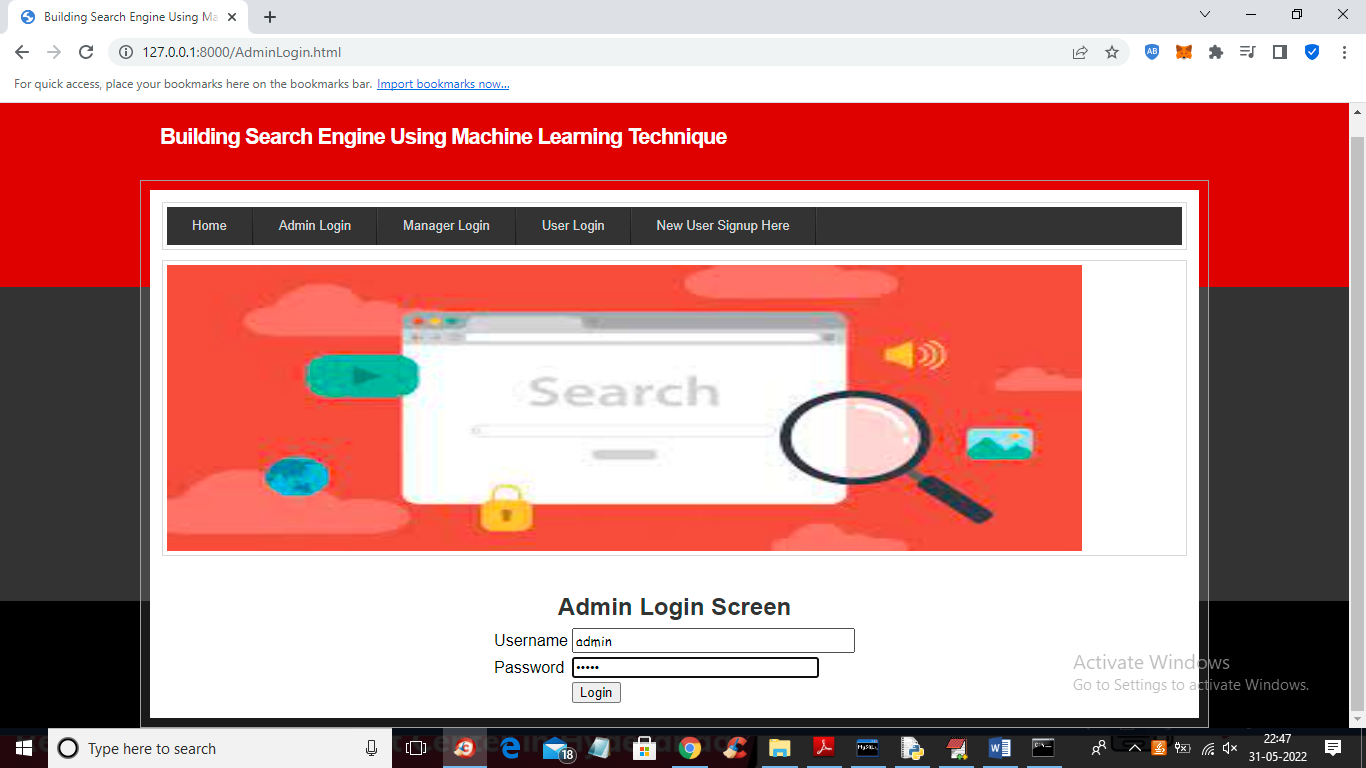
In above screen user signup process completed and now click on ‘User Login’ to get below screen



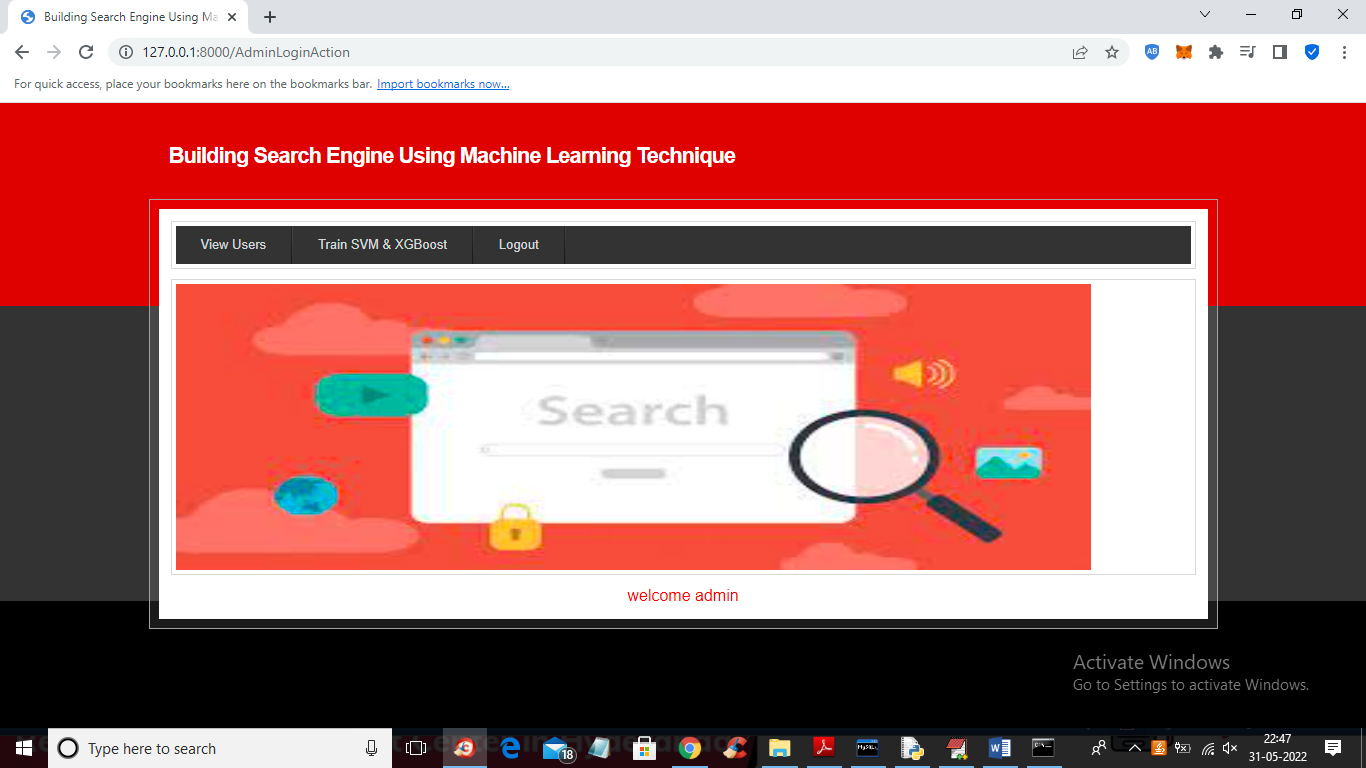
In above screen user is login and will get below output



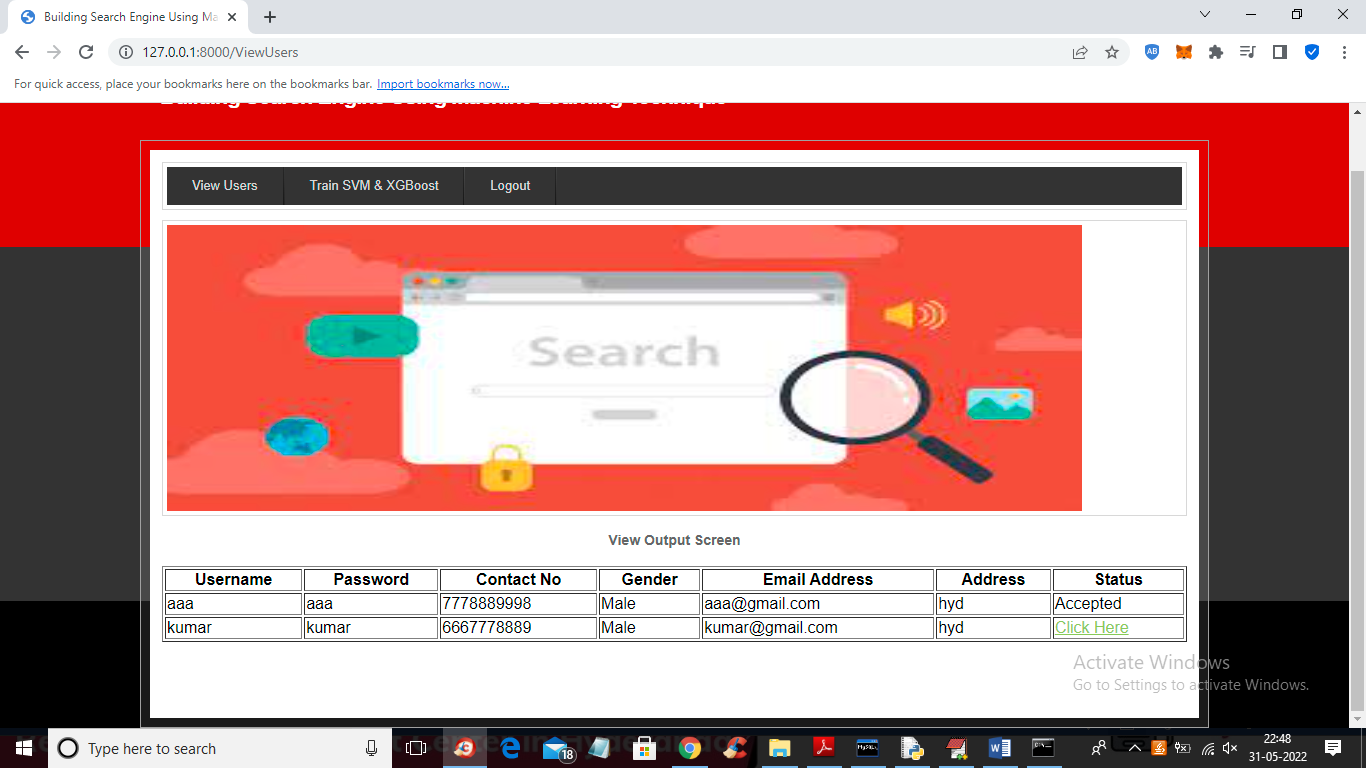
In above screen we gave correct login but account not activated by admin so click on ‘Admin Login’ link to login as admin and then activate user



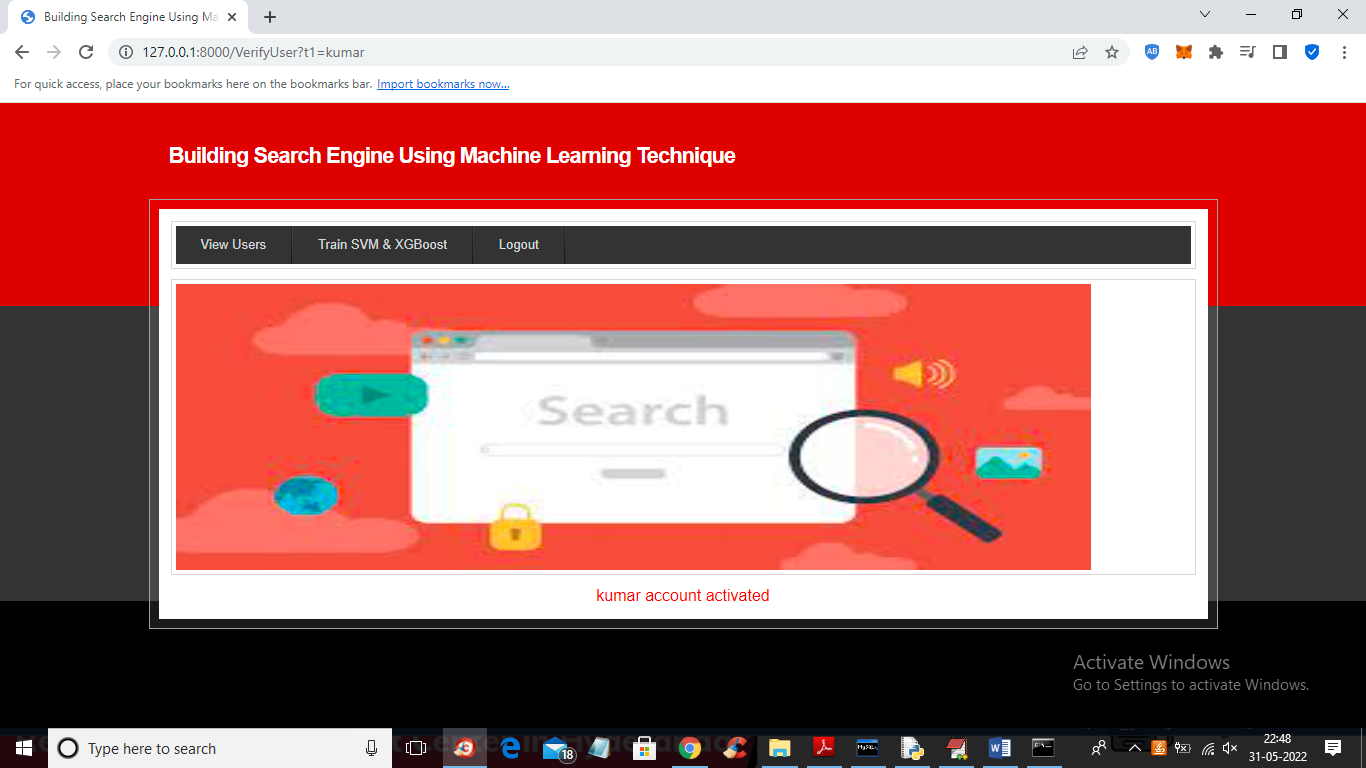
In above screen admin is login and after login will get below screen



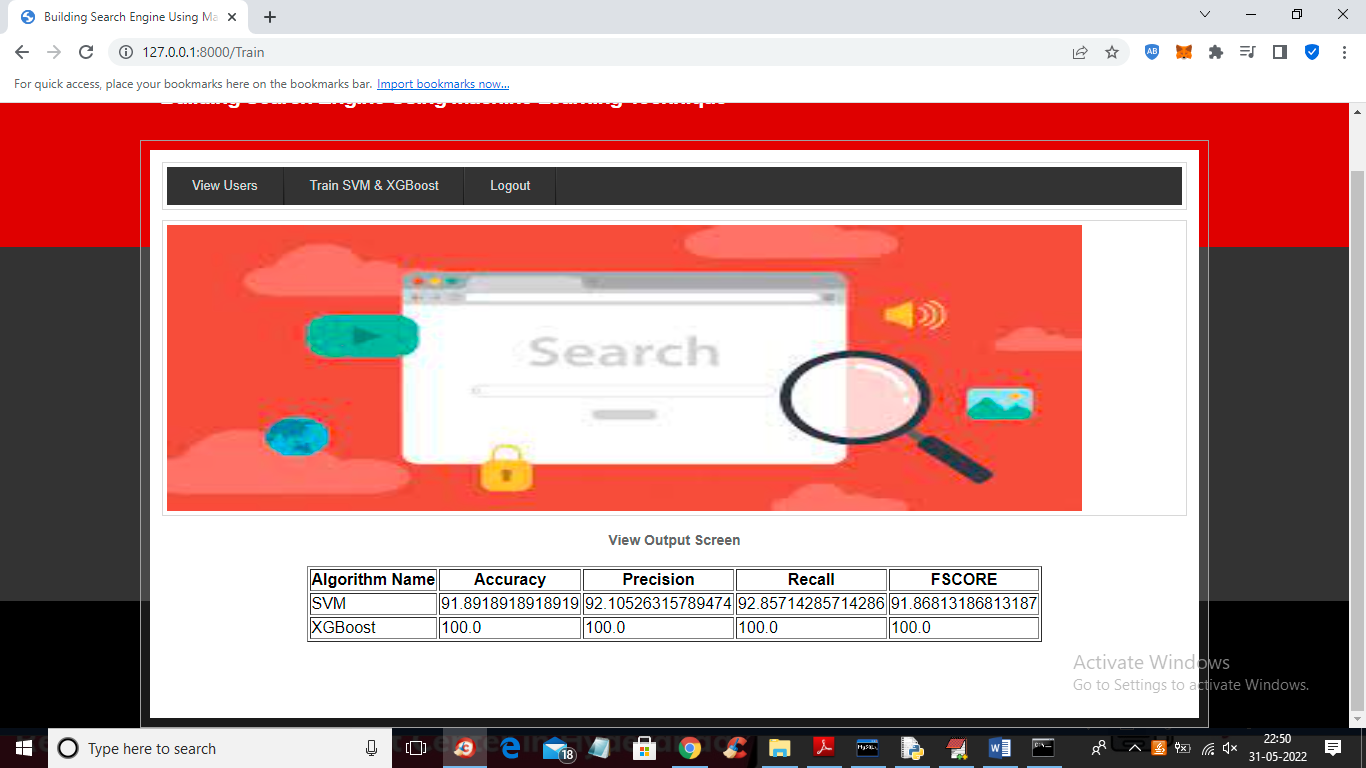
In above screen admin can click on ‘View Users’ link to view all users



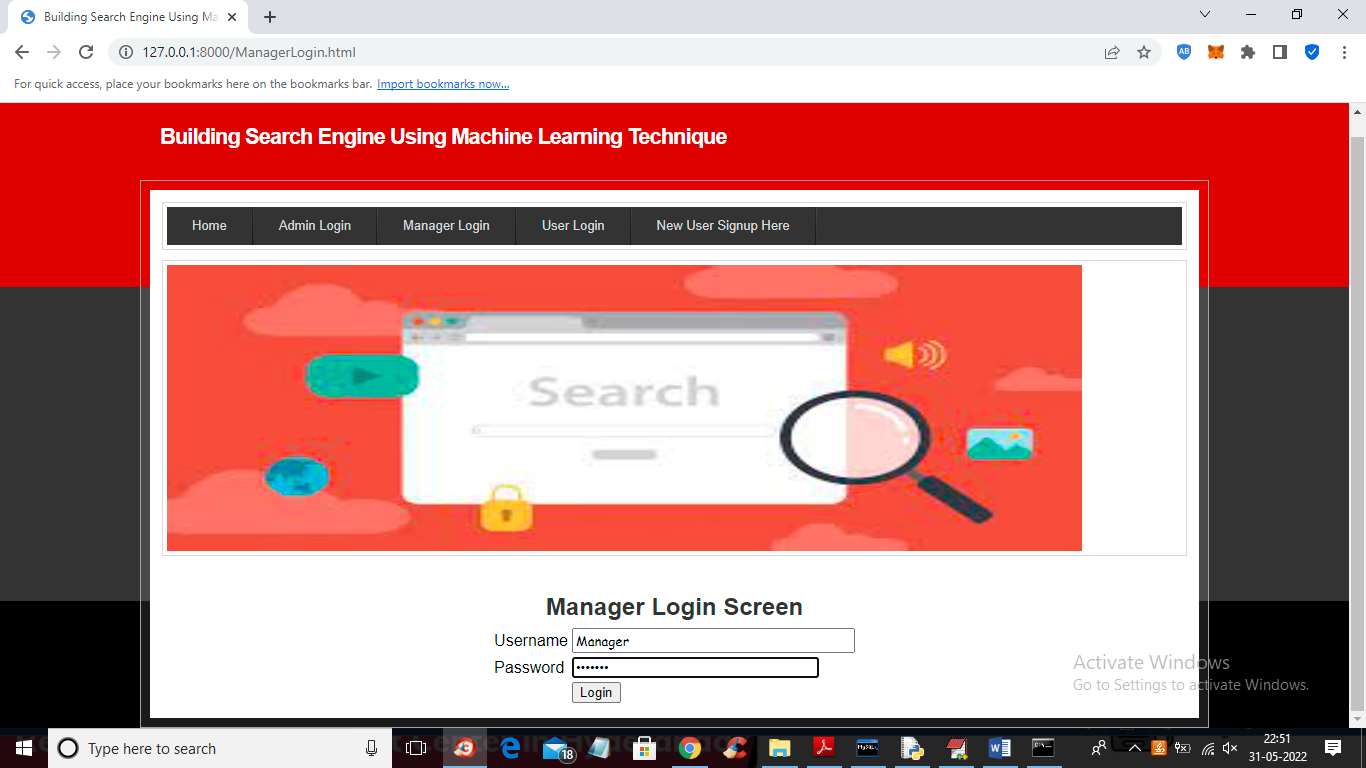
In above screen admin can click on ‘Click Here’ link to activate that user account



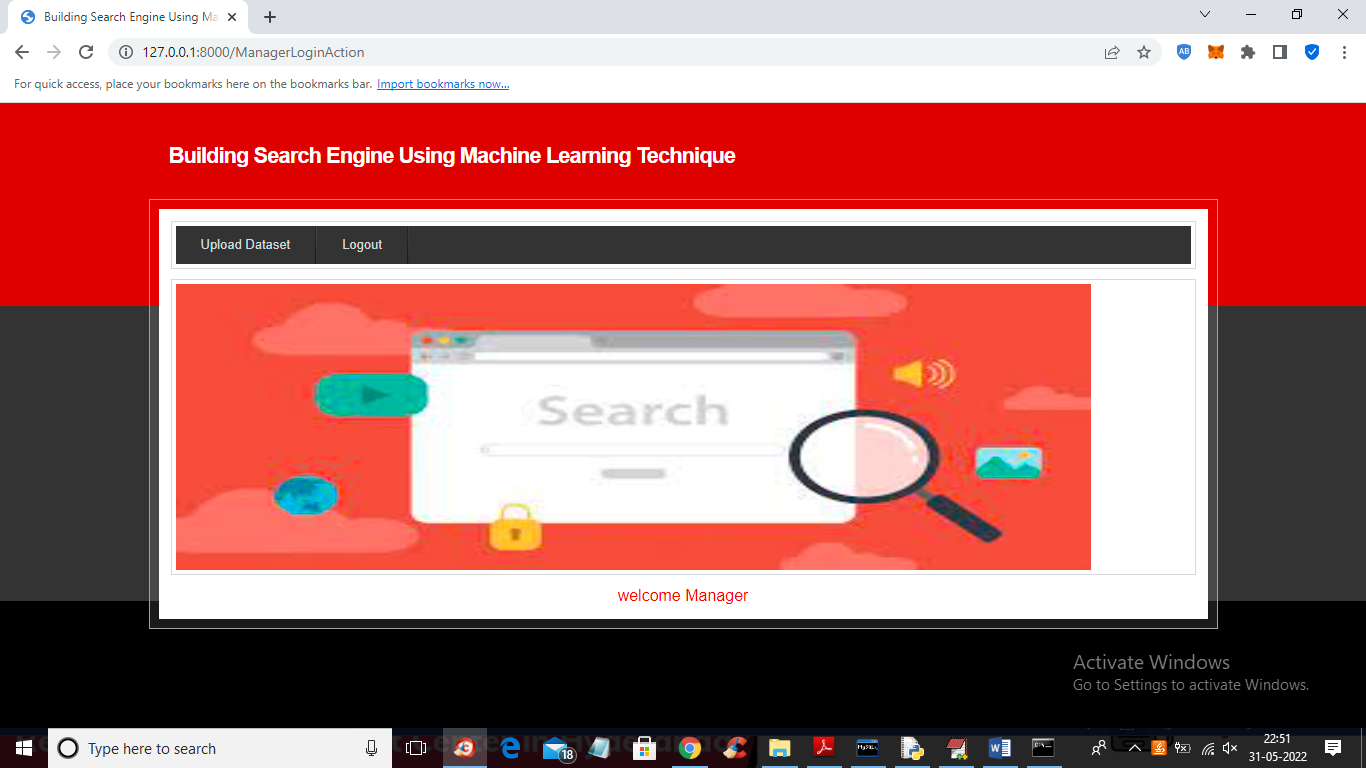
In above screen we can see admin activated kumar user account and now admin can click on ‘Train SVM & XGBOOST’ link to train machine learning SVM and XGBOOST algorithm and get below output



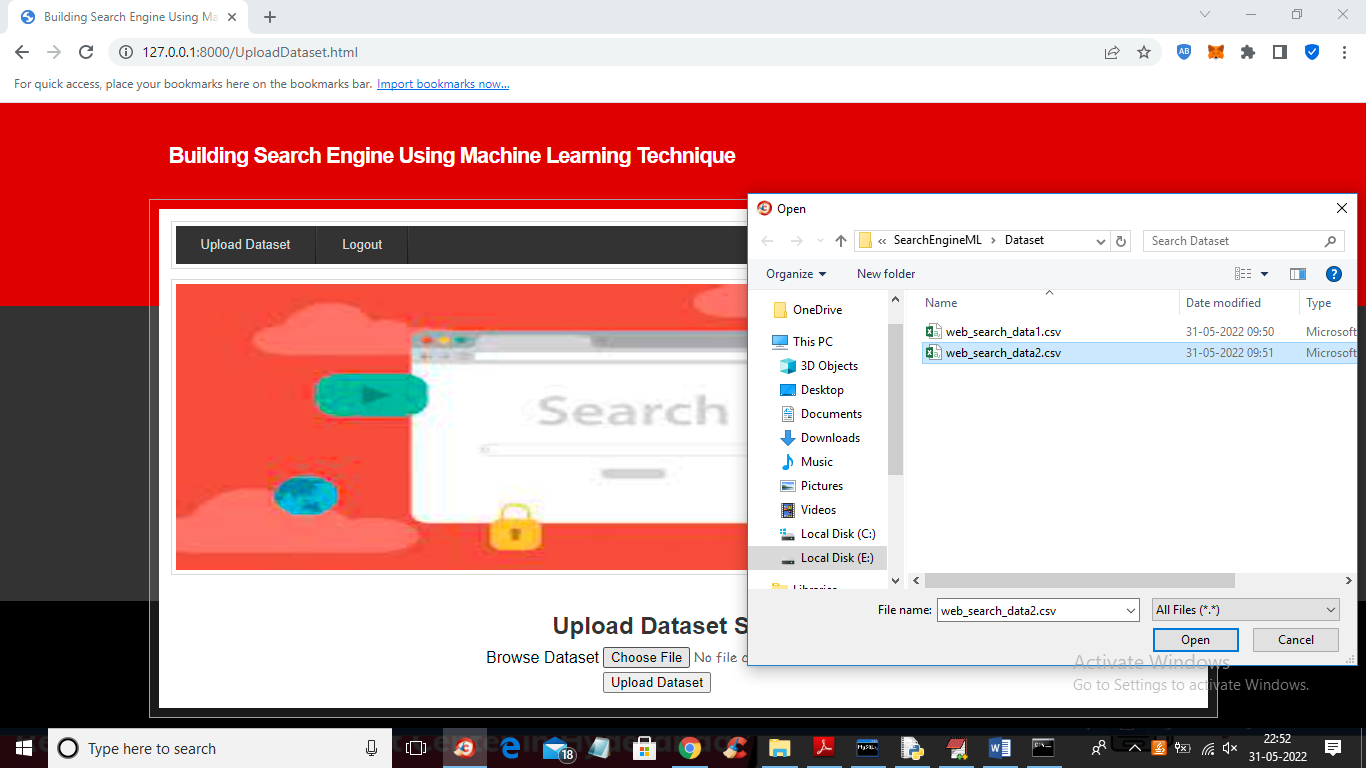
In above screen we can see SVM and XGBOOST accuracy and in both algorithms XGBOOST got high accuracy and now logout and login as Manager



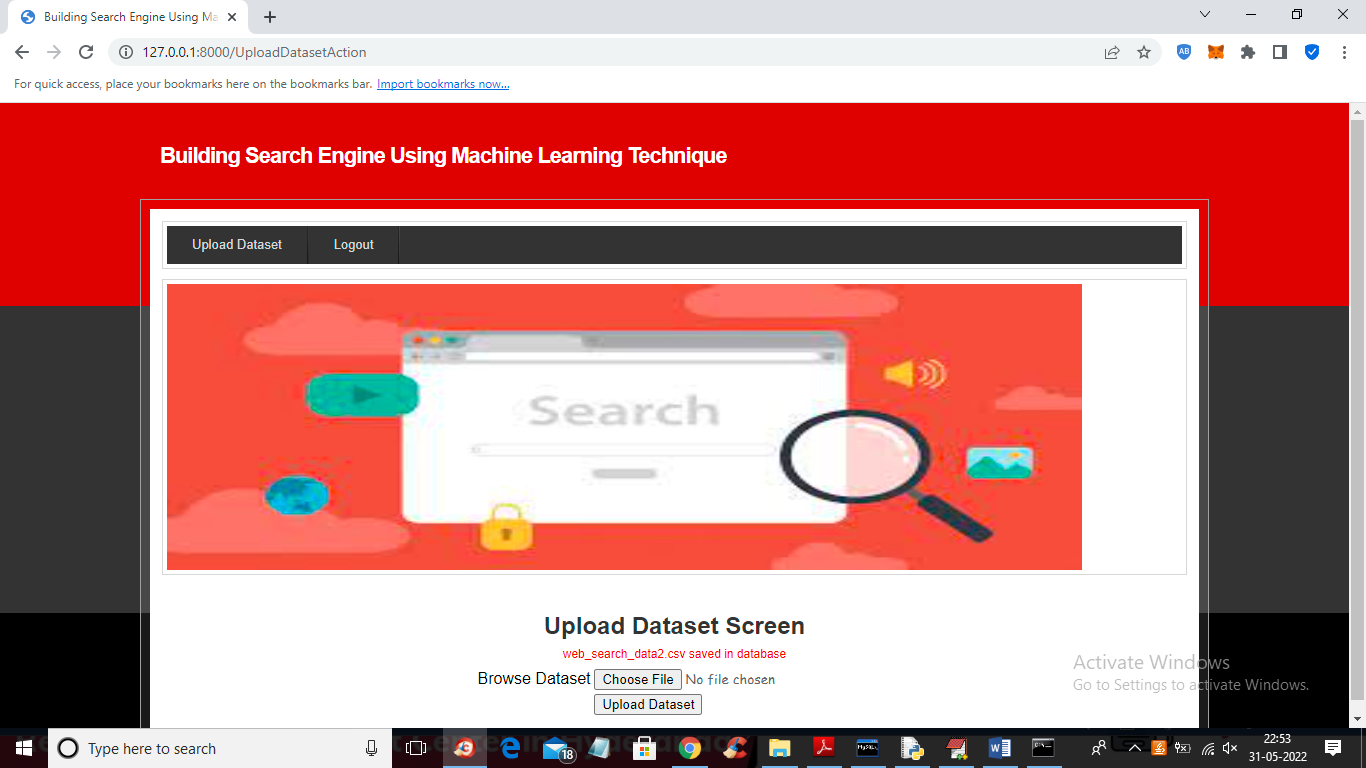
In above screen manager is login and after login will get below screen



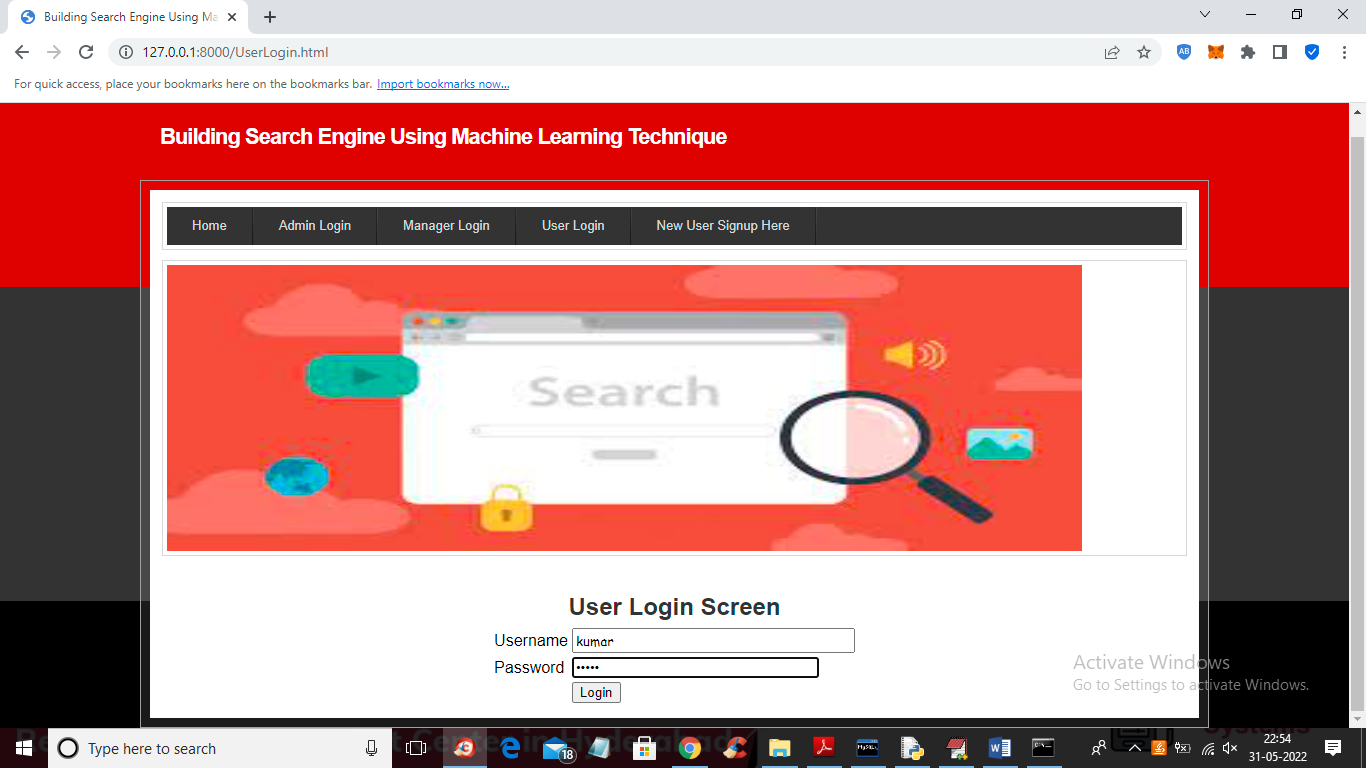
In above screen manager can click on ‘Upload Dataset’ link to upload dataset or documents



In above screen manager is browsing and uploading dataset and this file you can find inside ‘Dataset’ folder and now press button to saved dataset at server database



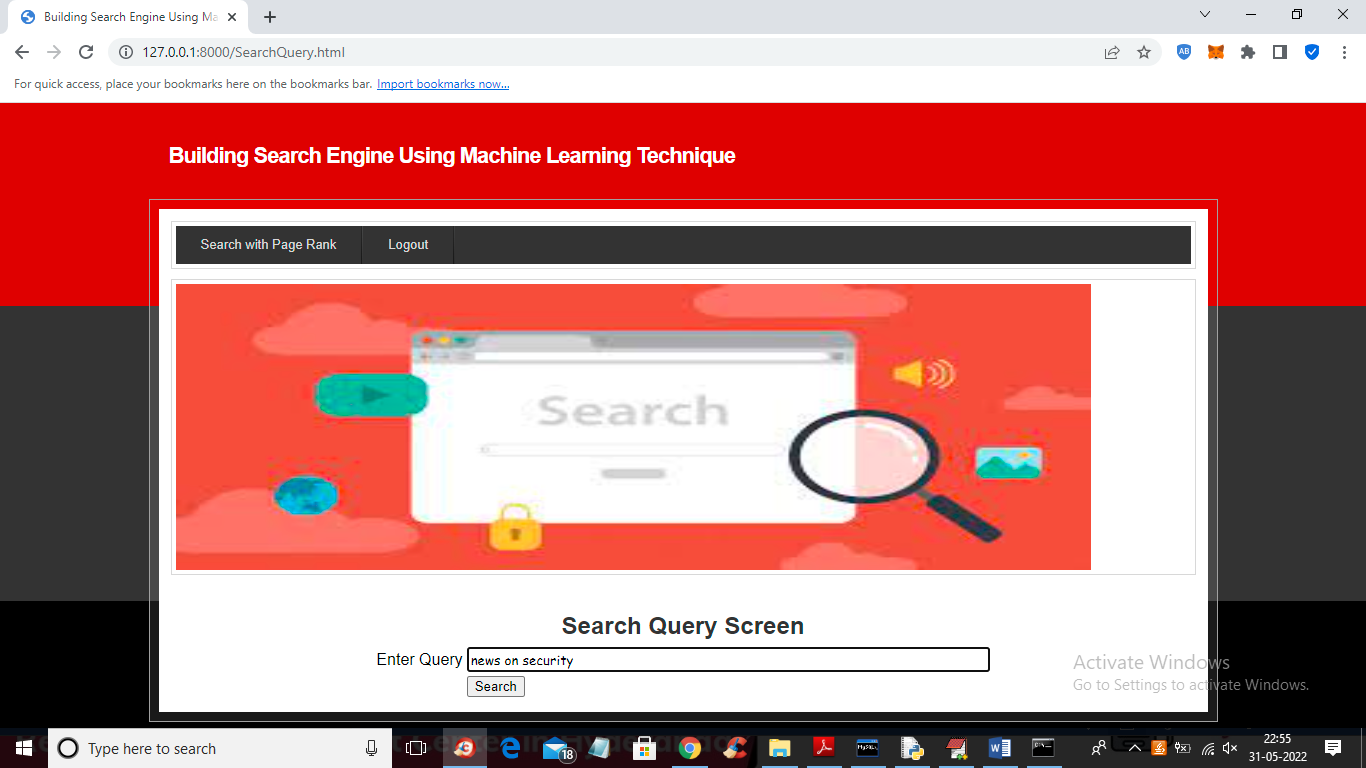
In above screen dataset file saved in database and now logout and login as user to perform search



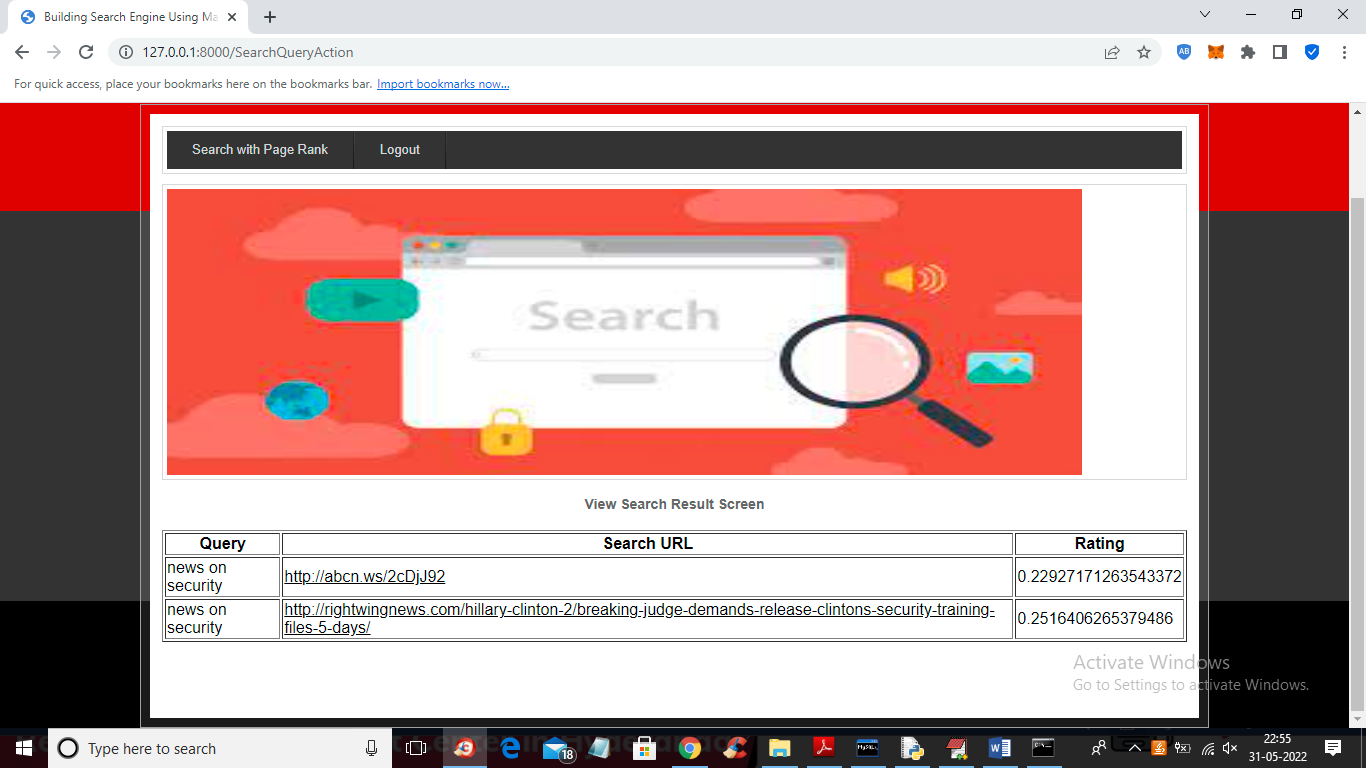
In above screen user is login and after login will get below output



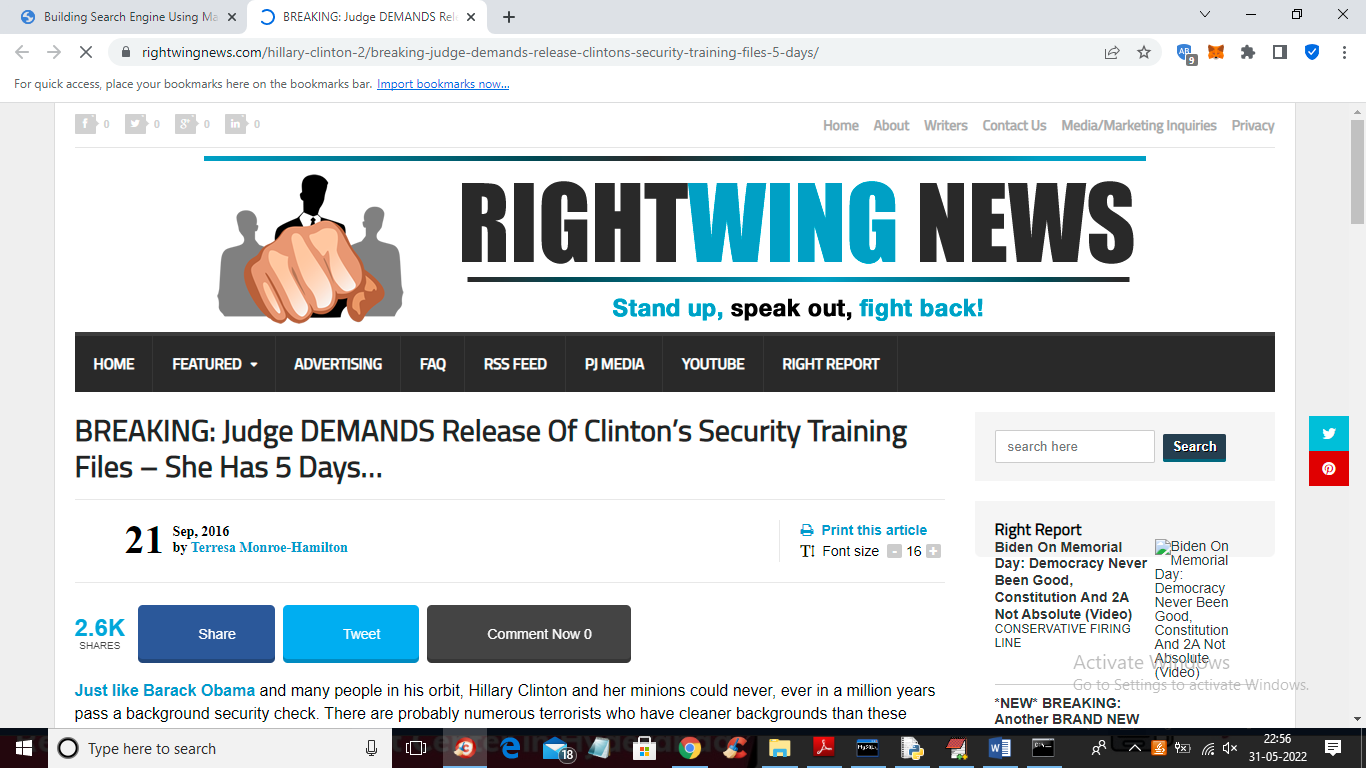
In above screen user can click on ‘Search with Page Rank’ link to search any data



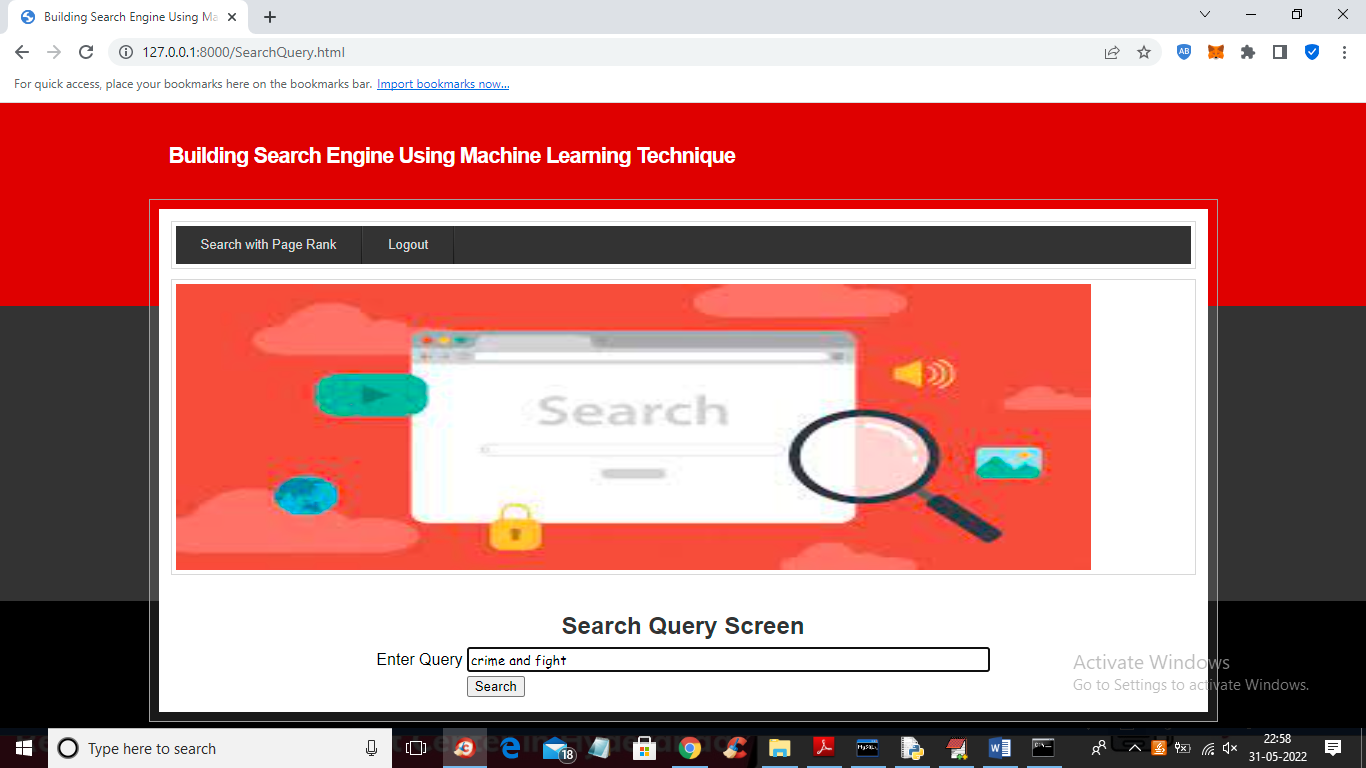
In above screen I entered query as ‘news on security’ and press button to get below search result



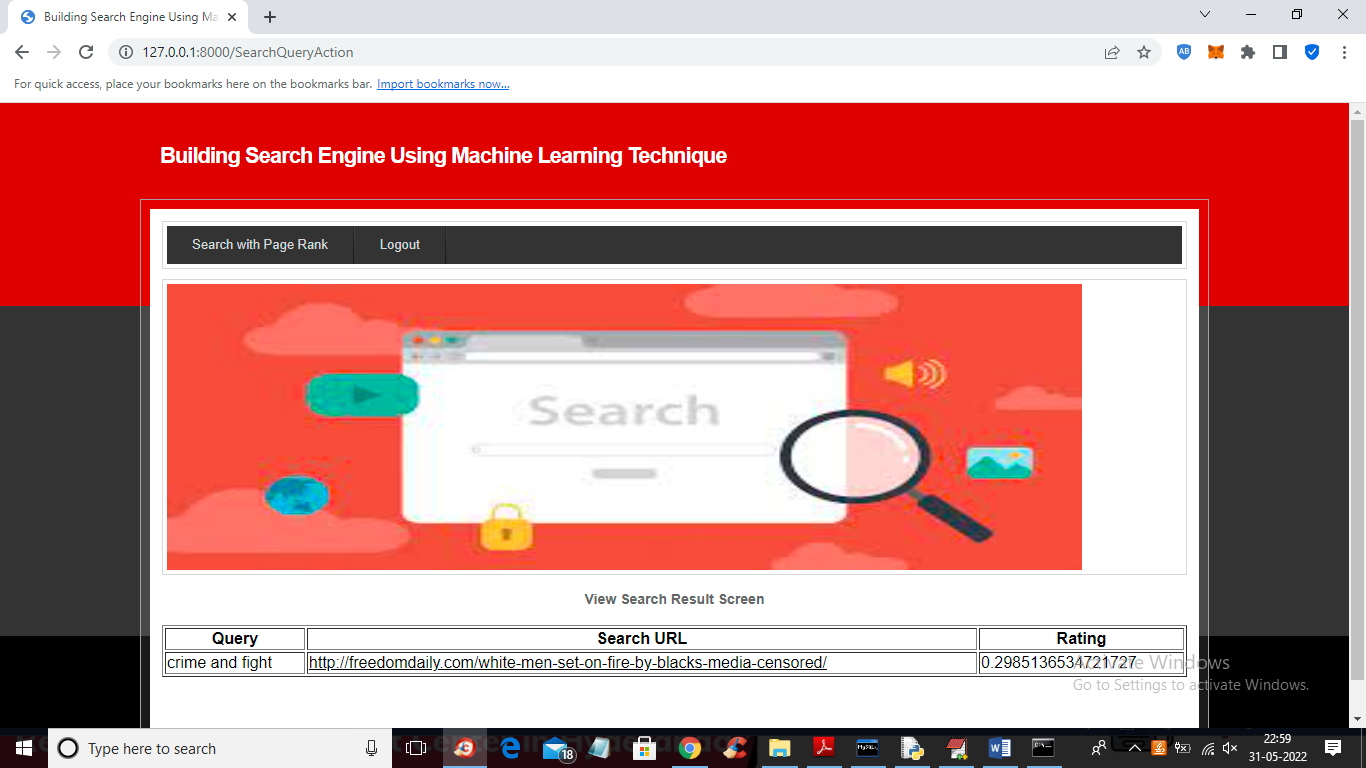
In above screen machine learning algorithm predicts two URLS for given query and user can click on those URLS to visit page



In above screen by clicking on URL link user can visit and view page. Similarly user can give any query and if query available in dataset then he will get output



For above query we got below result



**8. CONCLUSION:**

Search engine is very useful for finding out more relevant URL for given keyword. Due to this, user time is reduced for searching the relevant web page. For this, Accuracy is very important factor. From the above observation, it can be concluded that XGBoost is a best in terms of accuracy than SVM and ANN. Thus, Search engine built using XGBoost and PageRank algorithm will give better accuracy.

**9. REFERENCES:**

[1] Manika Dutta, K. L. Bansal, “A Review Paper on Various Search Engines (Google, Yahoo, Altavista, Ask and Bing)”, International Journal on Recent and Innovation Trends in Computing and Communication, 2016.

[2] Gunjan H. Agre, Nikita V.Mahajan, “Keyword Focused Web Crawler”, International Conference on Electronic and Communication Systems, IEEE, 2015.

[3] Tuhena Sen, Dev Kumar Chaudhary, “Contrastive Study of Simple PageRank, HITS and Weighted PageRank Algorithms: Review”, International Conference on Cloud Computing, Data Science & Engineering, IEEE, 2017.

[4] Michael Chau, Hsinchun Chen, “A machine learning approach to web page filtering using content and structure analysis”, Decision Support Systems 44 (2008) 482–494,scienceDirect,2008.

[5] Taruna Kumari, Ashlesha Gupta, Ashutosh Dixit, “Comparative Study of Page Rank and Weighted Page Rank Algorithm”, International Journal of Innovative Research in Computer and Communication Engineering, February 2014.

[6] K. R. Srinath, “Page Ranking Algorithms – A Comparison”, International Research Journal of Engineering and Technology (IRJET), Dec2017.

[7] S. Prabha, K. Duraiswamy, J. Indhumathi, “Comparative Analysis of Different Page Ranking Algorithms”, International Journal of Computer and Information Engineering, 2014.

[8] Dilip Kumar Sharma, A. K. Sharma, “A Comparative Analysis of Web Page Ranking Algorithms”, International Journal on Computer Science and Engineering, 2010.

[9] Vijay Chauhan, Arunima Jaiswal, Junaid Khalid Khan, “Web Page Ranking Using Machine Learning Approach”, International Conference on Advanced Computing Communication Technologies, 2015.

[10] Amanjot Kaur Sandhu, Tiewei s. Liu., “Wikipedia Search Engine: Interactive Information Retrieval Interface Design”, International Conference on Industrial and Information Systems, 2014.

[11] Neha Sharma, Rashi Agarwal, Narendra Kohli, “Review of features and machine learning techniques for web searching”, International Conference on Advanced Computing Communication Technologies, 2016.

[12] Sweah Liang Yong, Markus Hagenbuchner, Ah Chung Tsoi, “Ranking Web Pages using Machine Learning Approaches”, International Conference on Web Intelligence and Intelligent Agent Technology, 2008.

[13] B. Jaganathan, Kalyani Desikan,“Weighted Page Rank Algorithm based on In-Out Weight of Webpages”, Indian Journal of Science and Technology, Dec-2015.