IBM PROJECT REPORT

SKILL AND JOB RECOMMENDER APPLICATION

TEAM ID: PNT2022TMID30600

Submitted by

SUSHMITHA. G	**Reg. No: (613019104085)**
NARMADHA.J	**Reg. No: (613019104050)**
SURUTHI. R	**Reg. No: (613019104083)**
SOWMYA.M	**Reg. No: (613019104079)**

TABLE OF CONTENTS

ABSTRACT

Most businesses now use Internet-based recruiting portals as their main hiring method. Such platforms save the time and expense associated with hiring new employees, but they have problems with outdated information retrieval strategies based search approaches. As a result, a sizable number of applicants passed up the chance to be hired. The recommender system technology is used successfully in e-commerce applications to address issues linked to information overload. It helps customers identify products that fit their personal preferences. Numerous recommender system ideas have been put out in an effort to enhance the functionality of e- recruiting. This proejct will provide a survey of the electronic hiring process and current suggestion building methodologies for matching individuals to job. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skill set. Users will interact with the Chabot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

CHAPTER - 1

1.INTRODUCTION

1.1 PROJECT OVERVIEW

Water is the principal source for shipping energy to each cell in the body and is additionally Finding jobs that best suits the interests and skill set is quite a challenging task for the job seekers. The difficulties arise from not having proper knowledge on the organization's objective, their work culture and current job openings. In addition, finding the right candidate with desired qualifications to fill their current job openings is an important task for the recruiters of any organization. Job recommendation system has certainly made job seeking convenient to job seekers. This is the solution where recruiter as well as the job seeker meets aiming at fulfilling their individual requirement. To develop end-to-end web users are the cheapest as well as the fastest source of communication reaching wide range of audience on just a single click irrespective of their geographical distance. As simplified recruitment process which makes way convenience for job seekers to access job portal on the go, as our app script is built to support numerous business models as per the industry requirements. In this decade people are using their smart phone rather than web portals for job seeking online. So it is the right way for career portal which balances the gap between recruitment board and job searcher candidates. On the job seeker panel, the user can register their profile using system. Here, a job seeker searches for the positions that interest him and submits an application. Due to the abundance of job boards, candidates typically choose the one that offers the best services to them, including skills, creating a job profile, and suggesting new positions to job seekers.

1.1 PURPOSE

Job recommendation is primarily aimed at supporting the discovery of jobs that may interest the user. It should be dynamic in order to cater to the changing preferences of the user. The proposed system will help the user to overcome these difficulties by matching their skills and other details with appropriate companies suitable for respective user. The proposed system consists of user dataset with various attributes and company dataset with company details.

CHAPTER – 2
LITERATURE SURVEY

2.1.1. TITLE: Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms
AUTHOR: F. M. Javed Mehedi Shamrat

Business intelligence and analytics are data management techniques used in organisations to gather historical and current data utilising software and statistics. To provide insights for enhanced decision-making by analysing unprocessed data. In the current financial environment, it is necessary to be analytical and seek for the simplest method or intelligent business model in order to survive and develop one's own firm. The main goal is to assess how well different machine learning algorithms work with the system of an online job portal. This proposed module includes three phases, including the Clusters similar kind of job search phase (CSK), which creates a visual graph displaying clusters of similar types of jobs that job seekers have searched for on the website of the job portal, the email notifications send phase (ENS), which is in charge of sending email notifications to job seekers when a job circular is posted on the website, and the extract the job circular phase (EJC), which is the method for finding relevant job postings. The outcome demonstrates the effective grouping of related job searches, sending of email notifications to particular individuals, and information extraction from the web. The primary goal of the essay is to group together verses from the Holy Qur'an. The K-means technique was used by the authors of the research to mine the text from the Holy Qur'an and count the number of steam- and unsteam-steamed words in each cluster. The final illustration displays the various densities within each cluster. The authors suggested text document vectorization. After that, in order to provide the optimum results, the initial seed points should be chosen as widely apart as feasible.

2.1.1 TITLE: Designing and Implementation Of A Graduate Job Portal System.

2.1.2 AUTHOR : Zamiwe Tembo

This project was aimed at designing and developing the Graduate Job Portal System for the Graduates from various Colleges and Universities around Zambia and even beyond. The main aims of this portal are to connect to the industries and acts as an online recruitment to support the students to find the right jobs after graduation and to help companies to recruit sound and appropriate talented men and women of the Republic of Zambia. Furthermore, the system enhances the understanding concept and importance of the Graduate Job Portal for students in the universities. A survey has been conducted to identify the challenges students face after graduation when looking for employment and gathered the requirements which have been incorporated in to the system. After graduation, there is only one most important thing in the mind of any graduate and that is to find a job, settle and live a good life, but the paradox of reality defines life as a coin with two faces, that is, graduating is one thing and finding a job is completely another thing and the two are not related. For a country like Zambia where major companies or employers and the large population are aligned along the line of rail, media coverage is centred upon the large population living along the line of rail and some selected towns, mushrooming of registered and unregistered colleges producing thousands of graduates into the system every year, these and many economic issues have made it hard for a learned or erudite person, especially one who has profound knowledge of a particular field from recognized institutions to find suitable jobs. In addition, vices such as corruption which have reached alarming levels in developing countries in which Zambia is not an exception, over population as well as unstable political atmosphere has resulted into companies employing wrong or incompetent staff members.

2.1.3 TITLE: Recruitment And Selection Process With Reference Using Job PortalFramework.

2.1.4 AUTHOR: Ankit Bhatnagar1 ,Nitish Kajla2 ,Mahesh Kumar Gupta3

The project's goal is to create an online search gateway for the college's or company's placement department. With the proper login information, this system, which is a web application, may be accessed both inside and outside the company. To manage the scholar data related to placement, the placement department frequently uses these technologies as a web employment site. Students submit their information as a resume. The application manages numerous modules and the reports that go along with them. This means that a poor distribution of or lack of information about employment chances prevents people from learning about new job prospects, which is one of the causes of the lack of jobs. This indicates that although there are additional positions available, job searchers are not aware of them. Here, our website aids job searchers in their search for employment. The Internet has altered many parts of our lives today, including how we hunt for jobs. The goal of creating this website was to save both the candidate and the employer time. We offer two alternatives on our website: the first is to search for employment, and the second is to search for employees. Assume that if a person is seeking for work, they must select the option to search for positions, provide their contact information, and upload their resume[2]. On the other hand, if a business is seeking for an employee, it should select the option to search for employees and fill out the necessary information.

2.1.5 **TITLE:** Shared Values of E-Recruitment Portal: Determinant Factors of Job-Seekers' Intention to use Job Portals

AUTHOR: Aradhana Patra,Munjarin Rahman.

In Malaysia, job portals are a popular tool for locating employment. in Malaysia, a job. Due to technology advancements, traditional job searching is no longer used. advancements. Consequently, this study has been carried out to concentrate on the need and behaviour various ages, genders, educational backgrounds, and jobseekers from Malaysia through a survey questionnaire on the internet. an analytical or quantitative strategy was adopted. 104 job searchers took part in the poll, shared their opinions, and comments regarding their use of Malaysian online job hunting resources. When there a few open-ended inquiries for a more thorough and thorough research investigation. this academic undertaking addressed five important factors: Usability, User Experience, Performance Expectancy, and Performance Quality. Subjective Norm and Credibilityas Important Influences on Behavioural Intention to utilise a job site jobseekers. This study immediately advances the Job Portals' effectiveness and user-friendliness. With lower job search expenses and a more secure network, job hopefuls now have more opportunities to explore employment. By accelerating two-way interaction, it makes it easier for job applicants to comprehend the hiring process and gives more information about the company. E-recruitment, a subset of e-HRM, is a company's e-business initiative that uses webbased electronic technology to carry out human resources operations and procedures. According to this report, Job Street is the most popular job board in Malaysia, followed by LinkedIn. Both of these job sites place a strong emphasis on the needs of their customers and strive to effectively address the majority of the problems encountered by job seekers.

2.1.6 **TITLE:** Inequality in online job searching in the age of social media

AUTHOR: Gökçe Karaoglu, Eszter Hargittai & Minh Hao Nguyen

Better digital abilities may be crucial for successful job searching as hiring procedures increasingly move online. Digital inequality, on the other hand, raises concerns about who is most likely to be

able to seek for jobs online, especially on social media, given that it implies that people utilise the Internet in different ways and to varying degrees of skill. This essay investigates online job searching, covering the function of digital job-search abilities. The findings indicate that online job-seeking activities are influenced by sociodemographic traits (such as age, race, education, and income) as well as online experiences, using social media, and having better digital job-search skills. These results demonstrate the existence of digital disparities in online job searching, including variations due to social media usage. Additionally, the majority of these research INFORMATION, COMMUNICATION & SOCIETY 1827 did not take into account the importance of digital job-search abilities. However, Puckett and Hargittai (2012) showed that people with higher-level Internet abilities were more likely to use the Internet for finding information about jobs, indicating that this is a domain worth additional investigation. The study focused on the job-searching experiences of college students. They lacked assessments that were primarily focused on job-search abilities, a gap that this paper fills. The literature on the possible use of social media for job searching and the importance of digital job-search abilities is reviewed in the sections that follow.

CHAPTER – 3

SYSTEM ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal isput forth with a very general plan for the project and some cost estimates. Duringsystem analysis the feasibility study of the proposed system is to be carried out. Thisis to ensure that the proposed system is not a burden to the company. For feasibilityanalysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

Economical Feasibility
Technical Feasibility
Social Feasibility

3.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the researchand development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The

developed system must have a modest requirement, as only minimal or null changesare required for implementing this system.

3.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user.This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism,which is welcomed, as he is the final user of the system.

3.2 EXISTING SYSTEM

Water quality monitoring (WQM) sensor technology has improved in recent years. Sensorized equipment that can The existing system is handled manually. The system follows large number of paper work for maintaining job details and user can be difficult to search the part time jobs in manual process. In current system the student or user don't know about part time jobs details or company/office details and location. In this existing system takes lots of time for searching particular jobs information.

3.2.1 DISADVANTAGES

- Poor communication between user and company officer, so here intimating about new job is a hard task.
- Know the company job vacancy information is very difficult
- Immediate responseto the queries is difficult.
- More stationary use so they are expensive.
- Manual system is takes more time.
- Existing system is manually, so it increases the chances of errors.

3.3 PROPOSED SYSTEM

The proposed system is developed after a detailed study about the requirements requested by theuser. Proposed system is a computerized one, where all the limitations of manual system are compensated. jobs details of web application for skill based Job application system have simplified the working information and make a user friendly environment, where the user is provided with much flexibility to manage effectively. It helps the admin to generate desirable interface more quickly and also to produce better results.

Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. Many times, people who lack industry knowledge are unclear about what exactly they need to learn in order to get a suitable job for them. We address the problem of recommending suitable jobs to people who are seeking a new job.

3.3.1 ADVANTAGES

- User can easily know about the company details.
- Automation of existing manual information systems.
- Reduction of manual processing
- Users will interact with the Chabot and can get the recommendations based on their skills.
- Keep track of daily information exchange at the server by the administrator.

Increase in processing and transfer speeds of information over the network.

- Decrease in processing time

3.4 SYSTEM REQUIREMENTS

3.4.1 HARDWARE REQUIREMENTS

System : Pentium i3 Processor.

- Hard Disk:500 GB.
- Monitor:15'' LED

- Input Devices:Keyboard, Mouse
- RAM :4 GB

3.4.2 SOFTWARE REQUIREMENTS

- Operating system : Windows 10.
- Coding Language : Python
- Web Framework : Flask

3.5 LANGUAGE SPECIFICATION

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similarto PERL and PHP.

- **Python is Interactive** − You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

Easy-to-learn − Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs scripting.

Apart from the above-mentioned features, Python has a big list of good features, feware listed below

It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code forbuilding large applications.
- It provides very high-level dynamic data types and supports dynamic typechecking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., isavailable on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ isthe version you need to install.

- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

  Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there aresome definite differences between the languages.

FirstPythonProgram

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up thefollowing prompt −

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter −

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result −

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is nolonger active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file −

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to runthis program as follows −

```
$ python test.py
```

This produces the following result −

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication inworld wide web. Different methods of data retrieval from specified URL are definedin this protocol.

The following table summarizes different http methods −

Sr.No	Methods & Description	
1	**GET** Sends data in unencrypted form to the server.	Most common method.
2	**HEAD** Same as GET, but without response body	
3	**POST** Used to send HTML form data to server.method is not cached by server.	Data received by POST
4	**PUT** Replaces all current representations of theuploaded content.	target resource with the

5	**DELETE** Removes all current representations of the target resource given by aURL

By default, the Flask route responds to the **GET** requests. However, this preferencecan be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us createan HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request app=Flask(__name__)

@app.route('/success/<name>')
```

```
def success(name):

return'welcome %s'% name

@app.route('/login',methods=['POST','GET']) def login():

ifrequest.method=='POST': user=request.form['nm'] return

redirect(url_for('success',name= user)) else:

user=request.args.get('nm') return redirect(url_for('success',name=

user)) if__name__=='__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, entername in the text field and click **Submit**.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by −

```
user = request.form['nm']
```

Change the method parameter to **'GET'** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by −

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.
It is used for:

- web development (server-side),
- software development,
- mathematics,

- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready softwaredevelopment.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi,etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer linesthan some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed assoon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or afunctional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anythingother than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar forPython or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open thecommand line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you candownload it for free from the following website: https://www.python.org/

Python Quickstart

Python is an interpreted programming language, this means that as a developer youwrite Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

Where

```
C:\Users\Your Name>python helloworld.py
```

"helloworld.py" is the name of your python file.
Let's write our first Python file, called helloworld.py, which can be done in any texteditor.

helloworld.py

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directorywhere you saved your file, and run:

C:\Users\

```
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

Your Name>python helloworld.py

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not towrite the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:C:\Users*Your*

Name>python

From there you can write any python, including our hello world example from earlierin the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information. >>>
print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!") Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writingdirectly in the Command Line: >>> print("Hello, World!")Hello, World!

Or by creating a python file on the server, using the .py file extension, and runningit in the Command Line:

C:\Users*Your Name*>python myfile.py

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important. Python uses indentation to indicatea block of code. Example if 5 > 2: print("Five is greater than two!")

Python will give you an error if you skip the indentation: Exampleif 5 >

2:print("Five is greater than two!")

Comments

Python has commenting capability for the purpose of in-code documentation. Comments start with a #, and Python will render the rest of the line as a comment:Example

Comments in Python:

#This is a comment.print("Hello, World!")Docstrings

Python also has extended documentation capability, called docstrings.Docstrings

can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:Example

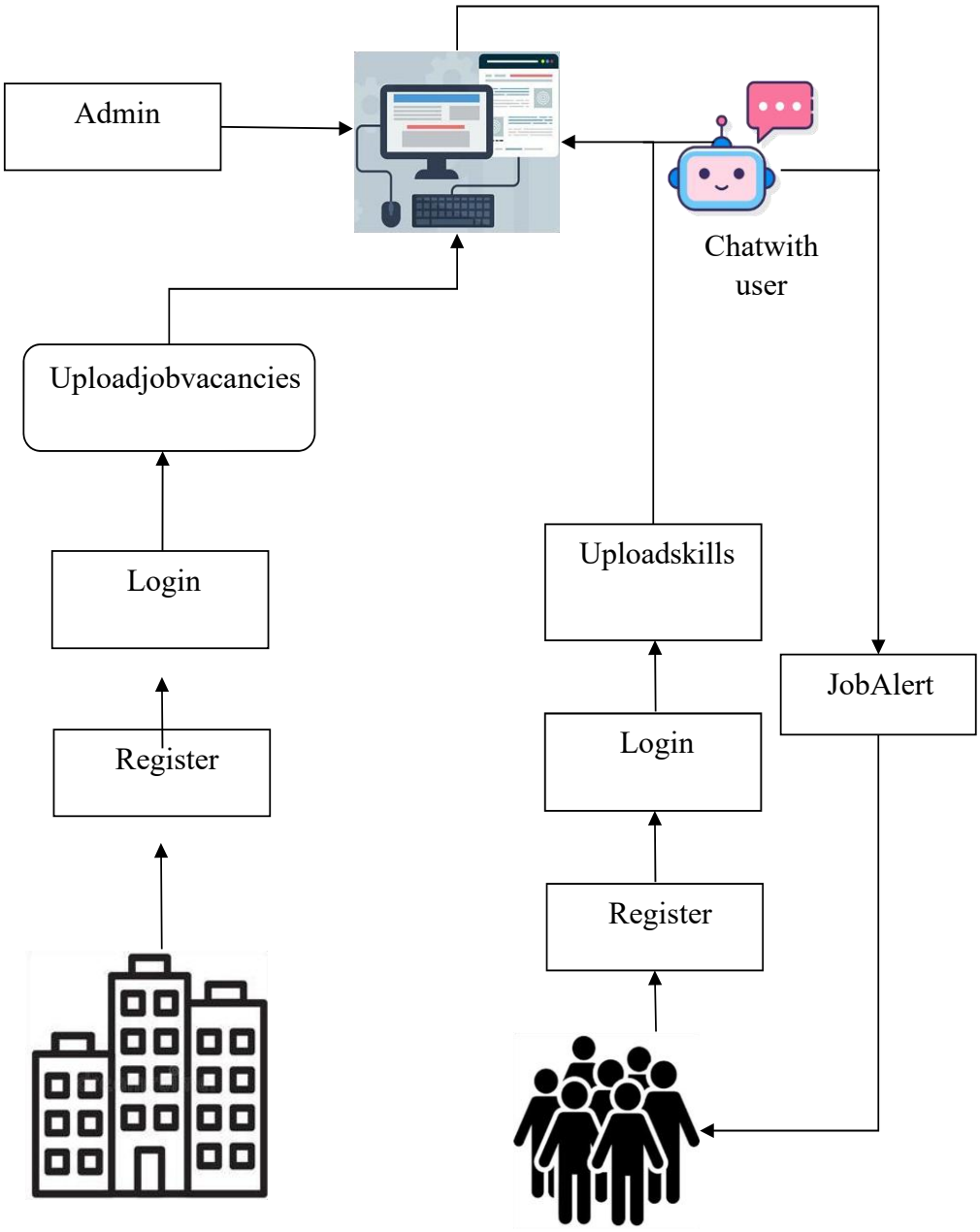
Docstrings are also comments:

"""Thisis a multiline docstring."""print("Hello, World!")

CHAPTER-4

SYSTEM DESIGN

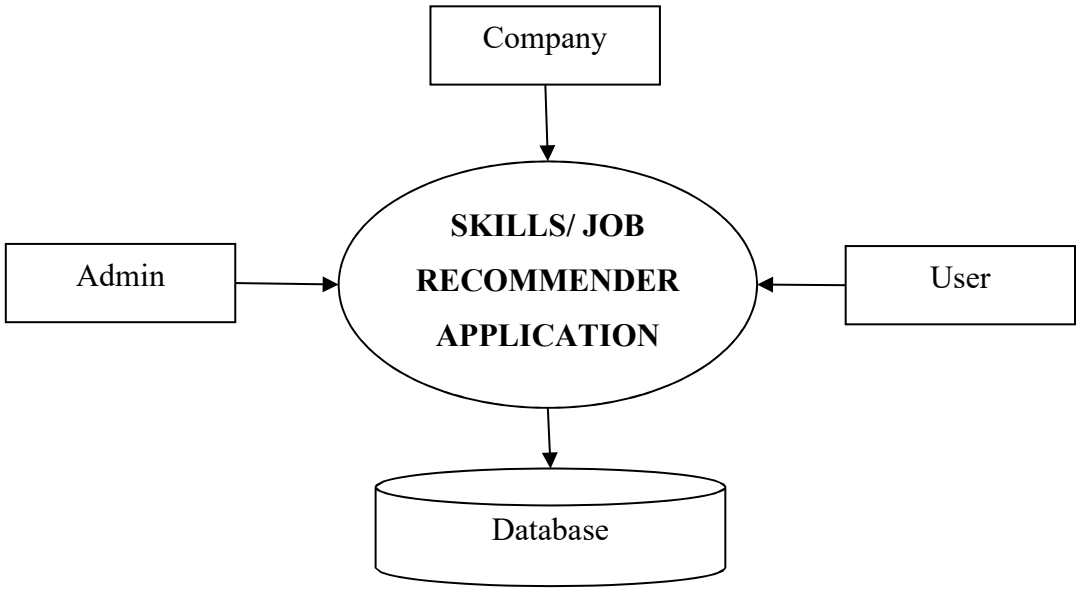
4.1 SYSTEM ARCHITECTURE



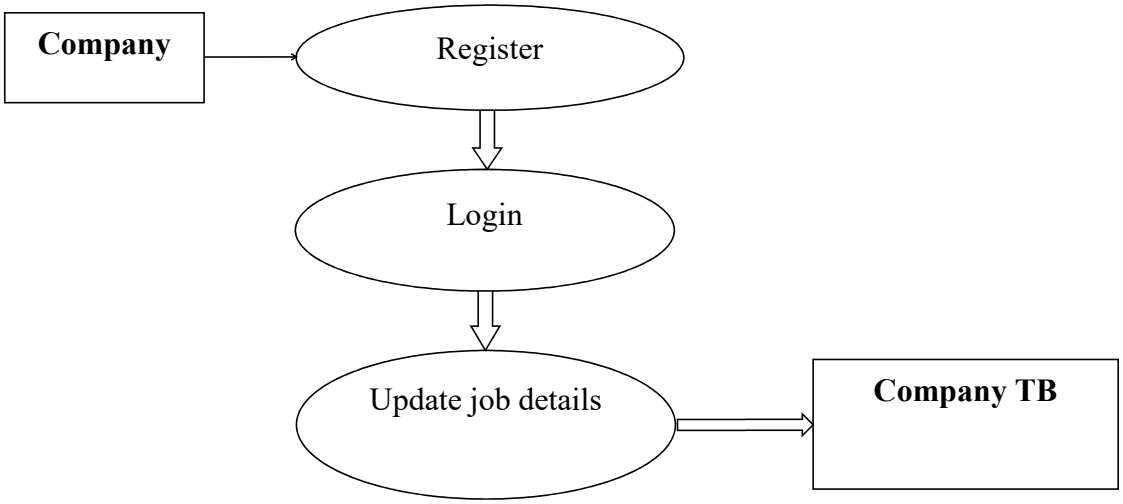
4.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism thatcan be used to represent a system in terms of input data to the system, variousprocessing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. Itis used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with thesystem and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depictsinformation flow and the transformations that are applied as data moves frominput to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a systemat any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

LEVEL 0



LEVEL 1



LEVEL -2

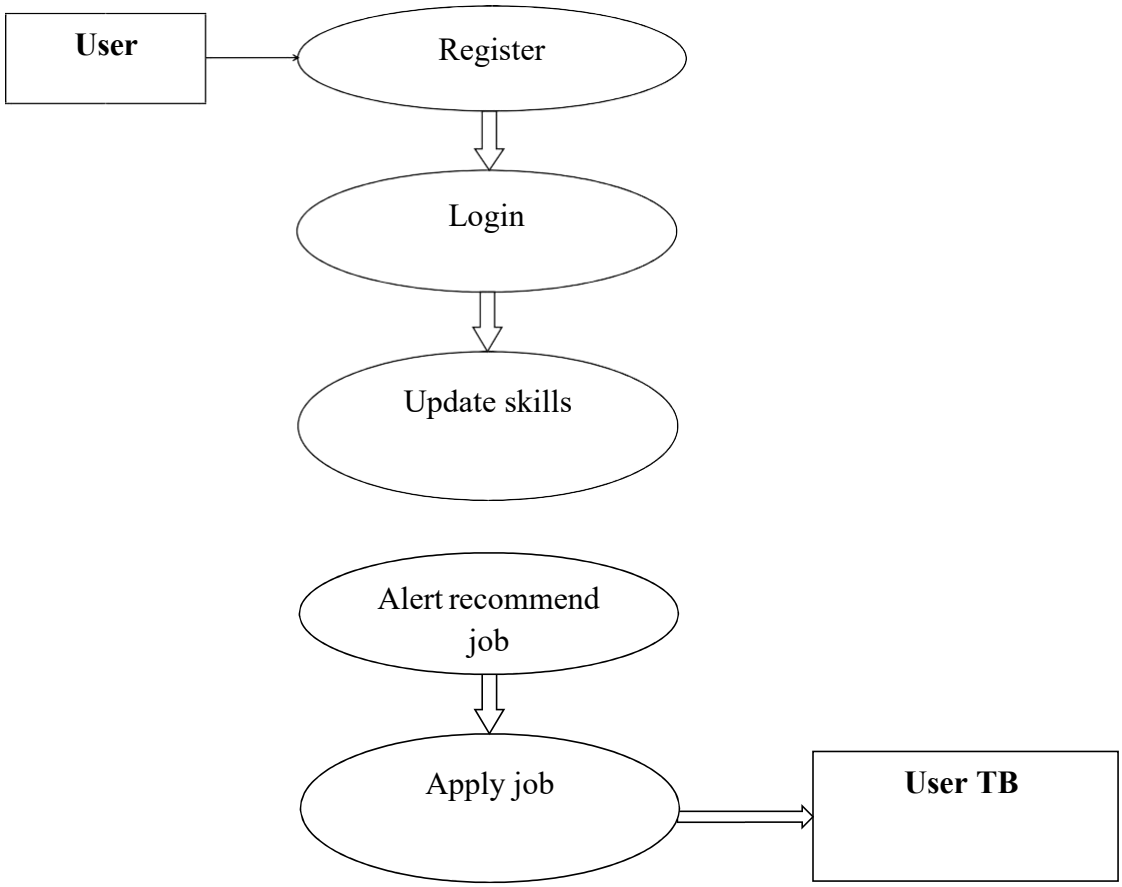


Figure:4.2 Data Flow Diagram

COMPONENTS & TECHNOLOGIES

S. No	Component	Description	Technology
1.	User Interface	Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in theapplication	Python
3.	Application Logic-2	Logic for a process in theapplication	SMTP
4.	Application Logic-3	Logic for a process in theapplication	SMTP

5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	Local host
7.	File Storage	File storage requirements	Local host
8.	External API-1	Purpose of External API used inthe application	-
9.	External API-2	Purpose of External API used inthe application	-
10.	Machine Learning Model	Purpose of Deep Learning Model	Classifies and detect theimages with high accuracy
11.	Infrastructure (Server /Cloud)	Application Deployment on LocalSystem / Cloud	Local Server Configuration
		Local Server Configuration: Cloud Server Configuration :	

APPLICATION CHARACTERISTICS

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworksused	PYCHARM
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	-
3.	Scalable Architecture	Justify the scalability ofarchitecture (3 – tier, Microservices)	Able to respond the changes in an application
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	The system must always befunctional

5.	Performance	Design consideration for the performance of the application (number of requests per sec, use ofCache, use of CDN’s) etc.	Takes no longer time toresponse

1.2 DATABASE SCHEMA

A table is a data structure that organizes information into rows and columns. It can be used to bothstore and display data in a structured format. For example, databases store data in tables so that information can be quickly accessed from specific rows. Websites often use tables to display multiple rows of data on page. Spreadsheets combine both purposes of a table by storing and displaying data in a structured format.

Databases often contain multiple tables, with each one designed for a specific purpose. For example, a company database may contain separate tables for employees, clients, and suppliers.

Each table may include its own set of fields, based on what data the table needs to store. In databasetables, each field is considered a column, while each entry (or record), is considered a row. A specific value can be accessed from the table by requesting data from an individual column and row.

Company table

Field	**Type**
companyname	nvarchar(50)
Regno	nvarchar(50)
Mobile	nvarchar(50)
Email	nvarchar(50)
Website	nvarchar(50)
Address	nvarchar(50)
Username	nvarchar(50)
Password	nvarchar(50)

Job table

Field	Type
Company name	nvarchar(50)
Contact no	nvarchar(50)
Address	nvarchar(50)
Location	nvarchar(50)
Vacancy	nvarchar(50)
Job	nvarchar(50)
Department	nvarchar(50)
Website	nvarchar(50)
Cname	nvarchar(50)

Registration table

Field	Type
Name	nvarchar(50)
Gender	nvarchar(50)
Age	nvarchar(50)
Email	nvarchar(50)
Phone	nvarchar(50)
Address	nvarchar(50)
Degree	nvarchar(50)

Department	nvarchar(50)
Username	nvarchar(50)
Password	nvarchar(50)

4.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized generalpurpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to expressthe design of software projects.

4.4 GOALS

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so thatthey can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the coreconcepts.
3. Be independent of particular programming languages and developmentprocess.

4.5 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms ofactors, their goals (represented as use cases), and any dependencies between those use cases.

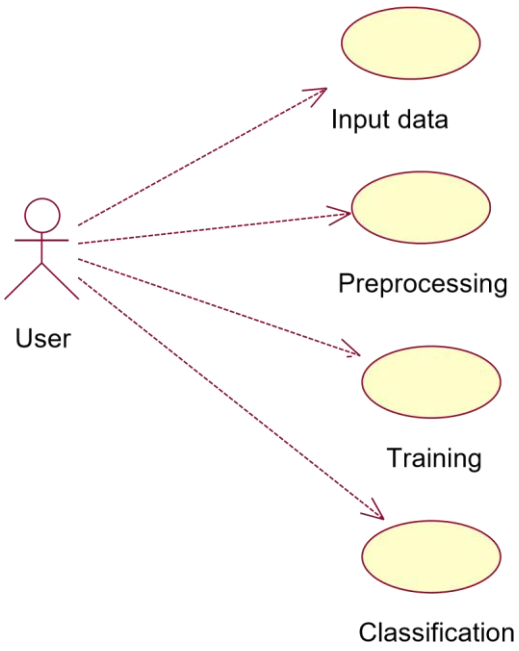


Figure:4.3 Use Case Diagram

4.6 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML)is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

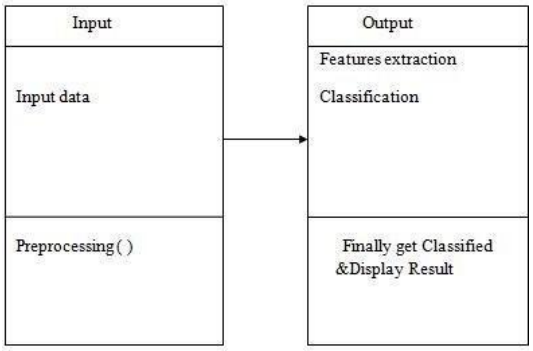


Figure:4.4 Class Diagram

4.7 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interactiondiagram that shows how processes operate with one another and in what order. It is

a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

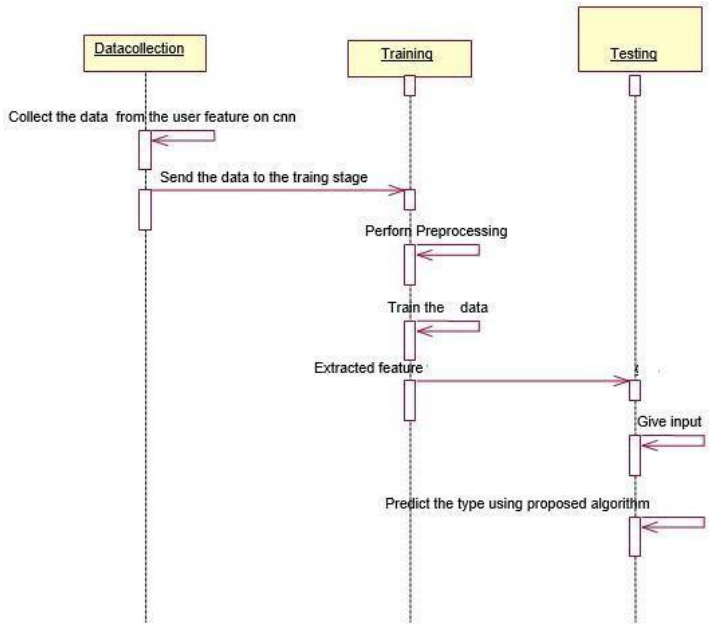


Figure:4.5 Sequence Diagram

4.8 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

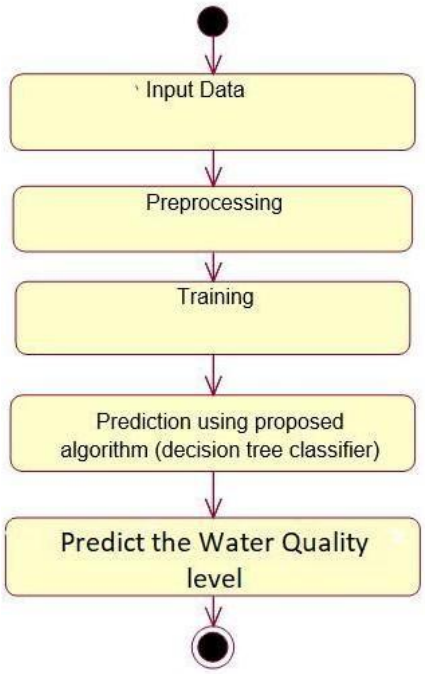


Figure:4.6 Activity Diagram

CHAPTER - 5

MODULE DESCRIPTION

5.1 MODULES

- **Create interface**

  This module offered a framework for job platform application to the user, to get answers without any human assistance. Admin can train keywords with answers for future processing. Chatbots are such kind of computer programs that interact with users using natural languages.

- **Registration**

  There is registration form available where new user can create their account by providing required information to the system. The registration form details are like name, email, gender, mobile number, address, and etc. These details are stored in the database. And then can getting to the username and password in the system. After the login process, the user can login in the system using username and password.

- **Update job details**

  The company can register to this application, the registered details like company name, id, email address; mobile number etc. after the registration process, the company can update the job details.

- **Update skills**

  The user can upload the skill details to this application. And the user will interact with the Chabot and can get the recommendations based on their skills.

- **Recommend job with alert**

  After updating the skills details, the system will recommend the job openings based on the user skills.

- **Apply job**

  After get the job alert, the user can apply the job through this application.

CHAPTER – 6
SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performsaccurately to the documented specifications and contains clearly defined inputs andexpected results.

6.1.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully

unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combinationof components.

6.1.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are availableas specified by the business and technical requirements, system documentation, anduser manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.Systems/Procedures:

interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining toidentify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.1.4 SYSTEM TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test.

System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.1.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at leastits purpose. It is purpose. It is used to test areas that cannot be reached from a blackbox level.

6.1.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as mostother kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

.you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works

6.1.7 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing tobe conducted as two distinct phases.

6.1.8 TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

6.1.9 TEST OBJECTIVES

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

6.1.10 FEATURES TO BE TESTED

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

CHAPTER – 7

CONCLUSION & FUTURE ENCHANCEMENT

7.1 CONCLUSION

In this essay, we suggested a structure for the duty of job recommendations. The use of a variety of text processing and recommendation methods in accordance with the preferences of the job recommender system creator is permitted by this framework, which also makes it easier to comprehend the job suggestion process. Furthermore, we make a new dataset with profiles of job seekers and open positions publicly accessible. The coding is done in a simplified and easy to understandable manner so that other team trying to enhance the project can do so without facing much difficulty. The documentation will also assist in the process as it has also been carried out in a simplified and concise way.

7.2 FUTURE ENHANCEMENT

In future we can develop this project in android application. This system is developed such a way that additional enhancement can be done without much difficulty. The renovation of the project would increase the flexibility of the system. Also the features are provided in such a way that the system can also be made better and efficient functionality

- Try to all user contact with online.
- Add more features in site future.

APPENDIX 1

A1. SAMPLE CODE

SOURCE CODE

App.py

```
from flask import Flask, render_template, flash, request, session
from flask import render_template, redirect, url_for, request

importjson from
json2html import *
importrequests

import ibm_db importpandas import
ibm_db_dbi
fromsqlalchemyimportcreate_engine

engine = create_engine('sqlite://',
                       echo = False)
dsn_hostname="98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"
dsn_uid = "tvd24047" dsn_pwd = "C0fhAXeLsuoQuvel"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB" dsn_port = "30875"
dsn_protocol = "TCPIP"
dsn_security = "SSL"

dsn = (
```

```
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port, dsn_protocol,
dsn_uid, dsn_pwd,dsn_security)



try:
    conn = ibm_db.connect(dsn, "", "") print ("Connected to database: ", dsn_database, "as user: ",
    dsn_uid, "on host: ", dsn_hostname)

except: print ("Unable to connect: ",
    ibm_db.conn_errormsg() )

app = Flask(_name_) app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



@app.route("/")
defhomepage(): return
render_template('index.html')
```

```
@app.route("/Home") defHome():
return render_template('index.html')

@app.route("/AdminLogin")
defAdminLogin():
   return render_template('AdminLogin.html')

@app.route("/NewUser") defNewUser():
return render_template('NewUser.html')

@app.route("/NewCompany")
defNewCompany():
   return render_template('NewCompany.html')

@app.route("/UserLogin")
defStudentLogin(): return
render_template('UserLogin.html')

@app.route("/CompanyLogin")
defCompanyLogin():
   return render_template('CompanyLogin.html')
        @app.route("/Search") defSearch(): return
                    render_template('Search.html')



@app.route("/AdminHome")
defAdminHome():

   conn = ibm_db.connect(dsn, "", "") pd_conn =
   ibm_db_dbi.Connection(conn) selectQuery =
```

```
    "SELECT * from regtb " dataframe =
    pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()
    return render_template('AdminHome.html', data=data)




@app.route("/ACompanyInfo")
defACompanyInfo():

    conn = ibm_db.connect(dsn, "", "") pd_conn =
    ibm_db_dbi.Connection(conn) selectQuery =
    "SELECT * from companytb " dataframe =
    pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')

    data = engine.execute("SELECT * FROM Employee_Data").fetchall()


    return render_template('ACompanyInfo.html', data=data)



@app.route("/AjobInfo")
defAjobInfo():
```

```
    conn = ibm_db.connect(dsn, "", "") pd_conn =
    ibm_db_dbi.Connection(conn) selectQuery =
    "SELECT * from jobtb " dataframe =
    pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()

    return render_template('AjobInfo.html', data=data)


@app.route("/SCompanyInfo")
defSCompanyInfo():


    conn = ibm_db.connect(dsn, "", "") pd_conn =
    ibm_db_dbi.Connection(conn) selectQuery =
    "SELECT * from jobtb " dataframe =
    pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()
    return render_template('SCompanyInfo.html', data=data)


@app.route("/CompanyHome")
defCompanyHome():
    return render_template('CompanyHome.html')


@app.route("/UserHome")
defUserHome():
```

```
    uname= session['uname']

    conn = ibm_db.connect(dsn, "", "") pd_conn =
    ibm_db_dbi.Connection(conn) selectQuery = "SELECT * FROM regtb
    where Username='"+ uname +"' " dataframe =
    pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine, if_exists='append') data
    = engine.execute("SELECT * FROM Employee_Data").fetchall()

    return render_template('UserHome.html', data=data)



@app.route("/CJobInfo")
defCJobInfo():


    cname= session['cname']
    conn = ibm_db.connect(dsn, "", "") pd_conn =
    ibm_db_dbi.Connection(conn) selectQuery = "SELECT * FROM
    jobtb where Cname='"+ cname +"' " dataframe =
    pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()



    return render_template('CJobInfo.html', data=data)
```

```
@app.route("/adminlogin", methods=['GET', 'POST']) defadminlogin(): error
= None if request.method == 'POST': if request.form['uname'] == 'admin'or
request.form['password'] == 'admin': conn = ibm_db.connect(dsn, "", "")
pd_conn = ibm_db_dbi.Connection(conn) selectQuery = "SELECT * FROM
regtb "

        dataframe = pandas.read_sql(selectQuery, pd_conn)
        dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
        data = engine.execute("SELECT * FROM Employee_Data").fetchall()

        return render_template('AdminHome.html', data=data)

    else:
     return render_template('index.html', error=error)
@app.route("/userlogin", methods=['GET', 'POST'])
defuserlogin():
  error = None if
  request.method == 'POST':

    username = request.form['uname']
    password = request.form['password']

    conn = ibm_db.connect(dsn, "", "")

    pd_conn = ibm_db_dbi.Connection(conn)

    selectQuery = "SELECT * from regtb where UserName='" + username + "' and password='"
```

```
+ password + """ dataframe =
        pandas.read_sql(selectQuery, pd_conn)

        ifdataframe.empty:
            data1 = 'Username or Password is wrong' return
            render_template('goback.html', data=data1)
        else:
            print("Login")
                        selectQuery = "SELECT * from regtb where UserName='" + username + "' and
password='" + password + """ dataframe =
            pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',
                        con=engine,
                        if_exists='append')

            # run a sql query print(engine.execute("SELECT * FROM
            Employee_Data").fetchall())

                    return render_template('UserHome.html', data=engine.execute("SELECT * FROM
Employee_Data").fetchall())



@app.route("/companylogin", methods=['GET', 'POST'])
defcompanylogin(): error = None if request.method ==
'POST':

        uname = request.form['uname']
        password = request.form['password']
        session['cname'] = uname
```

```
conn = ibm_db.connect(dsn, "", "")

pd_conn = ibm_db_dbi.Connection(conn)

            selectQuery = "SELECT * from companytb where UserName='" + uname + "' and
password='" + password + "'" dataframe =
pandas.read_sql(selectQuery, pd_conn)

ifdataframe.empty:
    data1 = 'Username or Password is wrong' return
    render_template('goback.html', data=data1) else:
    print("Login")
        selectQuery = "SELECT * from companytb where UserName='" + uname + "' and
password='" + password + "'" dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data',
                con=engine,
                if_exists='append')

    # run a sql query print(engine.execute("SELECT * FROM
    Employee_Data").fetchall())

      return render_template('CompanyHome.html', data=engine.execute("SELECT * FROM
Employee_Data").fetchall())

@app.route("/NewStudent1", methods=['GET', 'POST'])
defNewStudent1(): if request.method == 'POST':

    name = request.form['name']
    gender = request.form['gender']
```

```
        Age = request.form['Age'] email =
        request.form['email'] pnumber =
        request.form['pnumber'] address =
        request.form['address'] Degree =
        request.form['Degree'] depat =
        request.form['depat'] uname =
        request.form['uname'] passw =
        request.form['passw']
    conn = ibm_db.connect(dsn, "", "")

    insertQuery = "insert into regtb values('" + name + "','" + gender + "','" + Age + "','" + email +
"','" + pnumber + "','" + address + "','" + Degree + "','" + depat + "','" + uname + "','" + passw +
"')"
    insert_table = ibm_db.exec_immediate(conn, insertQuery)

    sendmsg(email, "Successfully registered this website")

    data1 = 'Record Saved!' return
    render_template('goback.html', data=data1)




@app.route("/newcompany", methods=['GET', 'POST'])
defnewcompany(): if request.method == 'POST':

        cname = request.form['cname']
        regno = request.form['regno']
        mobile = request.form['mobile']
```

```
        email = request.form['email']
        Website = request.form['Website']
        address = request.form['address']
        uname = request.form['uname']
        passw = request.form['passw'] conn
        = ibm_db.connect(dsn, "", "")

                                                insertQuery = "insert into companytb
values('"+cname+"','"+regno+"','"+mobile+"','"+email+"','"+Website+"','"+address+"','"+uname
+"','"+passw+"')" insert_table = ibm_db.exec_immediate(conn,
        insertQuery)

        data1 = 'Record Saved!' return
        render_template('goback.html', data=data1)




@app.route("/newjob", methods=['GET', 'POST'])
defnewjob():
    if request.method == 'POST': cnn =
        session['cname'] cname =
        request.form['cname'] cno =
        request.form['cno'] Address =
        request.form['Address']
        JobLocation = request.form['JobLocation']
        Vacancy = request.form['Vacancy']
```

```
        Job = request.form['Job']
        Department = request.form['depat']
        website = request.form['website']

        conn = ibm_db.connect(dsn, "", "")

            insertQuery = "insert into jobtb values('" + cname + "','" + cno + "','" + Address + "','" +
JobLocation + "','" + Vacancy + "','" + Job + "','" + Department + "','" + website + "','"+cnn+"')"
insert_table = ibm_db.exec_immediate(conn, insertQuery)

        conn = ibm_db.connect(dsn, "", "") pd_conn = ibm_db_dbi.Connection(conn)
        selectQuery1 = "SELECT * FROM regtb where Department='" + Department + "'"
        dataframe = pandas.read_sql(selectQuery1, pd_conn)

        dataframe.to_sql('regtb', con=engine, if_exists='append')
        data1 = engine.execute("SELECT * FROM regtb").fetchall()

        foritem1indata1:
           Mobile = item1[5] Email = item1[4] sendmsg(Email,"Jop


    Title"+Job + " More Info Visit Website") data = 'Record Saved!'


    return render_template("goback.html", data=data)




@app.route("/jobsearch", methods=['GET', 'POST'])
defjobsearch():
```

```
if request.method == 'POST':
  jobname = request.form['name'] url
  = "https://linkedin-jobs-
  search.p.rapidapi.com/"

  payload =  { "search_terms":
     jobname,"location":
     "india", "page":
  "1" } headers
  = {
     "content-type": "application/json",
     "X-RapidAPI-Key": "b045b9af95msha8d7c3160785729p1674cdjsnbdf4adbf9868",
     "X-RapidAPI-Host": "linkedin-jobs-search.p.rapidapi.com"
  }

  response = requests.request("POST", url, json=payload, headers=headers)

  print(response.text)

  infoFromJson = json.loads(response.text)

  df = pandas.json_normalize(infoFromJson)

  df.to_sql('regtb', con=engine, if_exists='append') data1 =
  engine.execute("SELECT * FROM regtb").fetchall()



  return render_template('Search.html',data=data1)
```

```
#send grid

defsendmsg(Mailid,message):
    importsmtplib
    fromemail.mime.multipartimportMIMEMultipart
    fromemail.mime.textimportMIMEText
    fromemail.mime.baseimportMIMEBase
    fromemailimportencoders

    fromaddr = "sampletest685@gmail.com"
    toaddr = Mailid

    # instance of MIMEMultipart
    msg = MIMEMultipart()

    # storing the senders email address
    msg['From'] = fromaddr

    # storing the receivers email address
    msg['To'] = toaddr

    # storing the subject
    msg['Subject'] = "Alert"

    # string to store the body of the mail
    body = message

    # attach the body with the msg instance
    msg.attach(MIMEText(body, 'plain'))
```

```
    # creates SMTP session s =
    smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

    # Authentication
    s.login(fromaddr, "hneucvnontsuwgpj")

    # Converts the Multipart msg into a string
    text = msg.as_string()

    # sending the mail
    s.sendmail(fromaddr, toaddr, text)

    # terminating the session if__name___

== '__main__': app.run(host='0.0.0.0',

debug='TRUE')

job.py

importrequests

importjson

importpandasaspd

from json2html
```

```
import * url =

"https://linkedin-

jobs-

search.p.rapidapi.co

m/"


payload = {
    "search_terms": "python programmer",
    "location": "india",
    "page": "1"
} headers = {
    "content-type": "application/json",
    "X-RapidAPI-Key": "b045b9af95msha8d7c3160785729p1674cdjsnbdf4adbf9868",
    "X-RapidAPI-Host": "linkedin-jobs-search.p.rapidapi.com"
}

response = requests.request("POST", url, json=payload, headers=headers)

print(response.text)

infoFromJson = json.loads(response.text)
print(json2html.convert(json = infoFromJson))
#data = json.loads(elevations) df =
pd.json_normalize(infoFromJson)
print(df)
```

VIDEO LINK:
https://youtu.be/gAHDv7LNZDE

REFERENCE

1. 1. M. Simić, G. M. Stojanović, L. Manjakkal, and K. Zaraska, "Multisensor system for remote environmental (air F. M. Javed Mehedi Shamrat, Implementation of an Intelligent OnlineJob Portal Using Machine Learning Algorithms, 2020.
2. Zamiwe Tembo, Designing And Implementation Of A Graduate Job Portal System, 2019.
3. Ankit Bhatnagar1 ,Nitish Kajla2 , Mahesh Kumar Gupta3,Recruitment And Selection Process With Reference Using Job Portal Framework, 2021.
4. Aradhana Patra,Munjarin Rahman,Shared Values of E-Recruitment Portal: Determinant Factors of Job-Seekers' Intention to use Job Portals, 2020.
5. Gökçe Karaoglu, Eszter Hargittai & Minh Hao Nguyen,Inequality in online job searching in the age of social media, 2021.