"Resume Ranking using NLP and Machine Learning"

Project Report

Submitted in fulfillment of the requirements for the degree of

Bachelor of Engineering

by

Juneja Afzal Ayub Zubeda (12CO32)
Momin Adnan Ayyas Shaheen(12CO46)
Gunduka Rakesh Narsayya Godavari(12CO29)
Sayed ZainulAbideen Mohd Sadiq Naseem (13CO72)

Supervisor

Prof. Tabrez Khan

Co-Supervisor

Prof. Irfan Jamkhandikar

Department of Computer Engineering,
School of Engineering and Technology
Anjuman-I-Islam's Kalsekar Technical Campus
Plot No. 2 3, Sector -16, Near Thana Naka, Khanda Gaon,
New Panvel, Navi Mumbai. 410206
Academic Year : 2015-2016

CERTIFICATE

Department of Computer Engineering,

School of Engineering and Technology,
Anjuman-I-Islam's Kalsekar Technical Campus
Khanda Gaon,New Panvel, Navi Mumbai. 410206

This is to certify that the project entitled **Resume Ranking using NLP and ML** is a bonafide work of **Juneja Afzal Ayub Zubeda (12CO32), Momin Adnan Ayyas Shaheen (12CO46), Gunduka Rakesh Narsayya Godavari (12CO29), Sayed ZainulAbideen MohdSadiq Naseem (13CO72)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering in Department of Computer Engineering**.

Prof. Tabrez Khan
Supervisor

Prof. Irfan Jamkhandikar
Co-Supervisor

Prof. Tabrez Khan
Head of Department

Dr. Abdul Razak Honnutagi
Director

Project Approval for Bachelor of Engineering

This project entitled ***Resume Ranking using NLP and ML*** *by* ***Juneja Afzal Ayub Zubeda (12CO32), Momin Adnan Ayyas Shaheen (12CO46), Gunduka Rakesh Narsayya Godavari (12CO29), Sayed ZainulAbideen MohdSadiq Naseem (13CO72)*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering***.

Examiners

1. .....

2. .....

Supervisors

1.....

2. .....

Chairman

.....

DECLARATION

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Juneja Afzal Ayub Zubeda(12CO32)

Momin Adnan Ayyas Shaheen(12CO46)

Gunduka Rakesh Narsayya Godavari (12CO29)

Sayed ZainulAbideen MohdSadiq Naseem(13CO72).

ABSTRACT

Tittle: Resume Ranking using NLP and ML

Using NLP(Natural Language Processing) and ML(Machine Learning) to rank the resumes according to the given constraint, this intelligent system ranks the resume of any format according to the given constraints or the following requirement provided by the client company. We will basically take the bulk of input resume from the client company and that client company will also provide the requirement and the constraints according to which the resume should be ranked by our system. Beside the information provide by the resume we are going to read the candidates social profiles (like LinkedIn, Github etc) which will give us the more genuine information about that candidate.

Juneja Afzal Ayub Zubeda(12CO32)

Momin Adnan Ayyas Shaheen(12CO46)

Gunduka Rakesh Narsayya Godavari (12CO29)

Sayed ZainulAbideen MohdSadiq Naseem(13CO72)

B.E. (Computer Engineering)
University of Mumbai.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Keywords And Glossary

Keywords :

NLP, ML, MySQL, Elastic Search, Ranking, Scrapy, Scikit learn, SQL

Glossary :

A

ATS: Applicant Tracking System

D

DES: Data Envelopment Analysis

F

Free Grammer: Set of rules and grammer used analysing of programming languages,parsing and manage the stucture of the documnet.

G

GUI: Graphical User Interface,is a type of interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.

H

History log: A snapshot of file history is just a click away with the File History Log.

M

Metaphor: A figure of speech in which a word or phrase is applied to an object or action to which it is not literally applicable.
Metonymy: The substitution of the name of an attribute or adjunct for that of the thing meant.
Morphoms: It is a basic unit of meaning.
MySQL: SQL language.

N

NLIDB: Natural Language Interface for Database,interface to interact with the database system.
NLTK: Natural Language Toolkit,python library used to process the natural language.
NLP: Natural Language Processing,field of Artificial Intelligence used to process human natural language.

p

Presuppositions: Presuppositions in general are beliefs underlying a system. The presuppositions of NLP are beliefs that guide and have guided the development of NLP.
Php: Hypertext Preprocessor ,is a server-side scripting language designed for web development but also used as a general-purpose programming language.
Pattern: Web mining module for Python, with tools for scraping, natural language processing, machine learning, network analysis and visualization.
Phonology: The system of contrastive relationships among the speech sounds that constitute the fundamental components of a language.

Parser: A parser is a program, usually part of a compiler.

Q

Quantifications: Use of an indicator of quantity in linguistic term.

S

Speech tagging: Process of marking up a word in a text (corpus) as corresponding to a particular part of speech.
SQL: Structured Query Language is a special-purpose programming language designed for managing data held in a database management system .
Shallow: Things that aren't very deep,thus Deep linguistic processing is a natural language processing framework.
Software as Service (SaaS): SaaS is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network.
Scrapy: Python tool for Web-Crawlers
Scikit-learn: It is open source python library which contains optimized implementation of machine learning algorithm.

T

Token: Each seperate word of the senetence.
TK: Tk/Tcl has long been an integral part of Python. It provides a robust and platform independent windowing toolkit, that is available to Python programmers using the tkinter package.

CHAPTER 1

Introduction

1.1 Statement of Project

1.1.1Need of Resume Ranking System:

In the present system the candidate has to fill each and every information regarding there resume in a manual form which takes large amount of time and then also the candidates, are not satisfied by the job which the present system prefers according to there skills. Let me tell you a ratio of 5:1 means, If 5 people are getting job than out of that 5, only a single guy will be satisfied by his/her job. Let me tell you an example : If i am a good python developer and particular company hired me and they are making me work on Java so, my python skills are pretty useless. And on the other hand if there is vacant place in a company so according to owner of the company he/she will prefer a best possible candidate for that vacancy. So our system will act as a handshake between this two entities. The company who prefer the best possible candidate and the candidate who prefers the best possible job according to his or her skills and ability.

1.1.2Problems and Solution:

The problem is that the present are not much flexible and efficient and time saving. It requires candidate, to fill the forms online than also you might not get the genuine information of the candidate. Beside

Where our system which saves the time of the candidate by providing to upload there resume in any format preferable to the candidate beside all the information in the resume our system will detect all its activity from the candidate social profile which will give the best candidate for that particular job and candidate will also be satisfied because he will get job in that company which really appreciates candidates skill and ability. On the other hand we are providing same kind of flexibility to the client company.

1.2 Motivation

The current recruitment process are more tedious and time consuming which forces the candidates to fill all their skill and information manually. And HR team requires more man power to scrutinize the resumes of the candidates. So that motivated to build a solution that is more flexible and automated.

1.3 Objective and scope

1.3.1 Objective:

The major objective of our system is to take the current resume ranking system to other level and makes it more flexible for both the entity.

1) Candidates, who has been hired.
2) Client company, who is hiring the candidates.

Candidates, who has been hired :

Candidates who are searching for jobs after been graduated. Out of those, major number of candidates are so much desperate that they are ready to work on any post irrelevant to their skill set and ability. The main reason behind this unemployment is like a cancer to our society, if a guy

is not got place after been passed out for 1yr, society include relatives starting blaming that guy. Inspite of this reason the candidate are ready to work in any condition, on any post. So they don't have to face those situation.

Where our system help such candidates to get hired by such a company or an organisation who really worth their ability and their skill sets. Where our algorithm will work in such a way that with the help of the previous result and previous ranking constraints, it will try to optimize the current result, which we called it Machine Learning.

This will make sure that the relevant candidate is been hired for that particular vacancy. You can say best possible candidate.

Client company, who is hiring the candidates :

Like I am the owner of a particular organisation, obviously my aim would be to create such a team which is the best team in the world. It is like, if there is a vacancy of a java developer in my organisation. So, I won't prefer to hire a python developer and then make him learn Java. That will be pretty useless and time consuming for both that candidate and for the organisation too.

Where our system help the organisation to make out the best possible candidates list according to their given constraints and requirement for that particular vacancy.

This kind of approach, will help our hiring sector to improve like anything and make it more efficient as the relevant person is getting a relevant job. So there would be no regrets for both the entities, client company and that hired candidate. Hence satisfaction will be achieved.

1.3.2 Scope:

As we know Indian I.T sector is second largest candidate recruiting sector of our country. It contribute about 7.5% to our Gross Domestic Product(G.D.P) Our Proposed system is initially concerned with the I.T sector of our country. It is mainly going to deal the Indian I.T industry but if you talk about the pro version of our system it can be extended to various other commercial sector where, intake and elimination are in bulk like for Govermental Jobs.

1.4 System Architecture

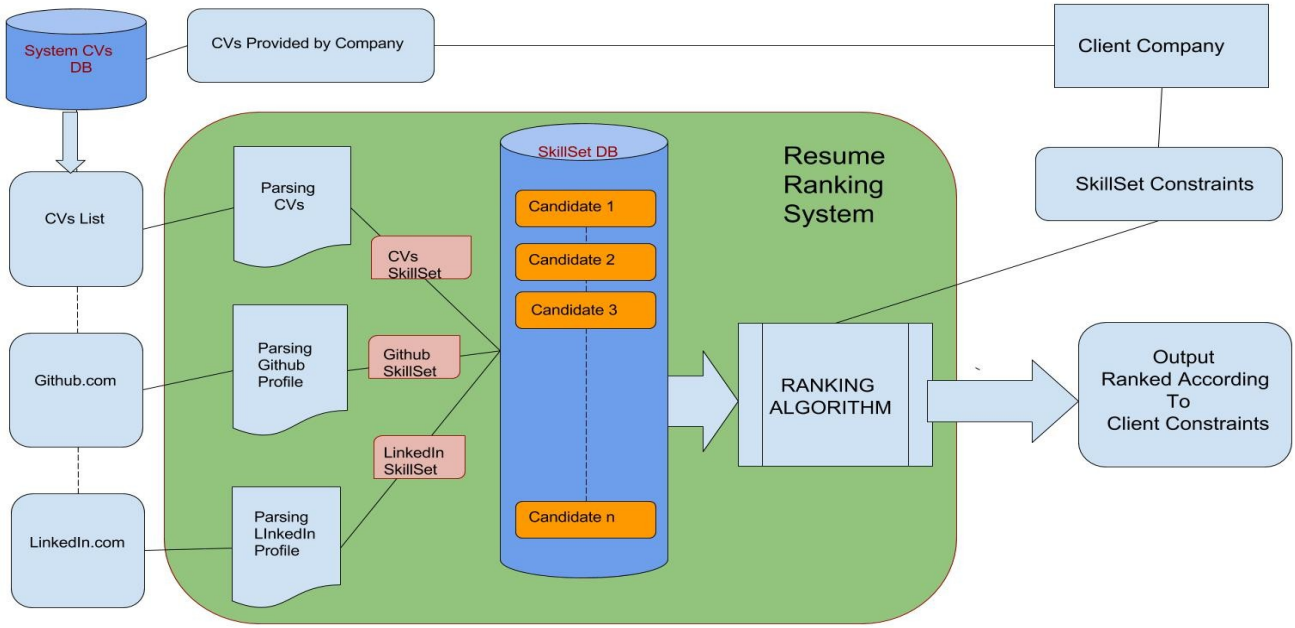


fig1.4.1. System Architecture

The System Architecture consists of two modules:

1. Outer World System
2. Resume Ranking System

1. Outer World System Consist Of:

1. Client Company.
2. System C.V's Data base.
3. Social Profile.

2. Resume Ranking System Consist Of:

1. Parser System.
2. Candidate Skillset Database.
3. Resume Ranking algorithm.

Client Company :

This is the client company who will provide us the bulk of the resume or C.V's with the specific requirement and constraints, according to which it should be ranked.

System C.V's Database :

This is the large database which is used to store the bulk of resumes provided by the client company in a distributed environment.

Social Profiles :

Social Profiles include LinkedIn Profile of the candidate, Github Profile of the Candidate. This social profile module can be extended to different community too.

Parser System :

Parsing system includes the parsing of the following candidate resume and their social profiles using NLP. That is without any manual interaction.Here, using Natural Language Processing the this is how we are going to parse the resume one at a time.

NLP (Natural Language Processing) requires following constraint for parsing :

- Morphological Analysis
- Syntactic Analysis
- Semantic Analysis

Morphological Analysis:

Morphology in linguistics is the study and description of how words are formed in natural language. In this phase the sentence is broken down into tokens- smallest unit of words, and determine the basic structure of the word.

For instance, unusually can be thought of as composed of a prefix un-, a
stem usual, and an affixly. composed is compose plus the inflectional affix -ed: a spelling rule means we end up with composed rather than composed.

a) Stop word removal:

Stop words are non context bearing words, also known as noisy words which are to be excluded from the input sentence to speed up the process.

b) Spelling check:

Three most popular method -

• Insertion : mistyping 'the' as 'th'
• Deletion: mistyping 'the' as 'ther'
• Substitution: mistyping 'the' as 'thw'

c) Token analyzer:

Each identified tokens can be represented as attribute token, value token, core token, multi-token, continuous token, etc.

• Attribute token- using metadata
• Core Token-first, all capital letters
• Numeric Token-digits , digits separated by decimal point
• Sentence Ending Markers- (. ? !)
• Value Token- (M.C.A, “mca”, ‘mca’)

Syntactic Analysis:

The objective of the syntactic analysis is to find the syntactic structure of the sentence.It is also called Hierarchical analysis/Parsing, used to recognize a sentence, to allocate token groups into grammatical phrases and to assign a syntactic structure to it.

a) Parse tree

Parser generates a parse tree with the help of syntactic analysis. A parse tree or parsing tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context free grammer.

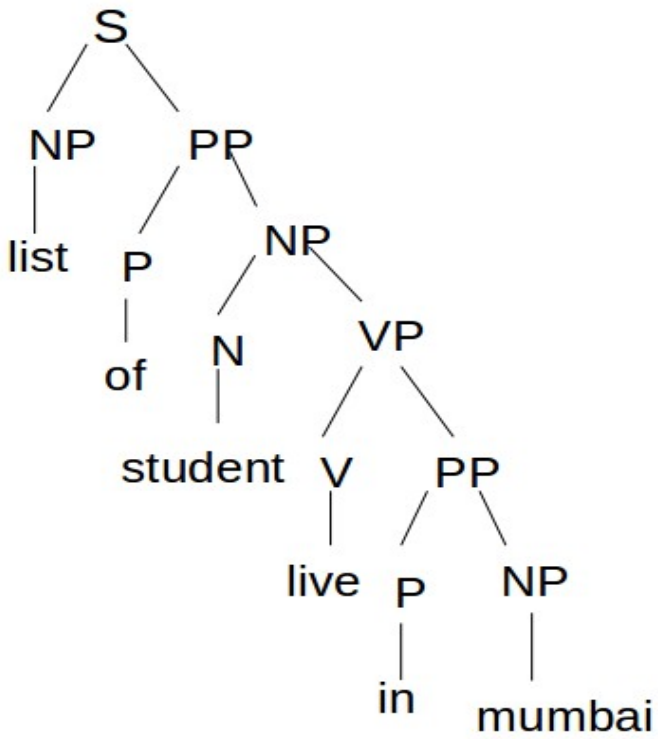


fig1.4.2. Parse Tree

Semantic Analysis :

Semantic Analysis is related to create the representations presentations for meaning of linguistics inputs. It deals with how to determine the meaning of the sentence from the meaning of its parts. So, it generates a logical query which is the input of Database Query Generator. It is another form of representation for user tokens and user input symbols in the form of semantic word.

CHAPTER 2

Literature Review

2.1 Case Study on talent acquisition

2.1.1 First Generation Hiring Systems

In this System the Hiring team would publish their vacancies and invite applicants. Methods of publishing were newspaper, television and mouth.
The interested candidates would then apply by sending there resumes. These resumes were then received and sorted by the hiring team and shortlisted candidates were called for further rounds of interviews.
The whole process would take lot of time and human efforts to find right candidate suitable for their job roles.

2.1.2 Second Generation Hiring Systems

As the industries have grown, there hiring needs has rapidly grown. To serve this hiring needs certain consultancy units have came into existence. They offered a solution in which the candidate has to upload their information in a particular format and submit it to the agency. Then these agencies would search the candidates based on certain keywords. These agencies were middle level organizations between the candidate and company. These systems were not flexible as the candidate has to upload there resume in a particular formats, and these formats changed from system to system.

2.1.3 Third Generation Hiring Systems

This is our proposed system, which allow the candidates to upload their resumes in flexible format. These resumes are then analyzed by our system, indexed and stored in a specific format. This makes our search process easy. The analyzing system works on the algorithm that uses Natural Language Processing, sub domain of Artificial Intelligence. It reads the resumes and understands the natural language/format created by the candidate and transforms it into a specific format. This acquired knowledge is stored in the knowledge base. The system acquires more information about candidate from his social profiles like Linkedin and Github and updates the knowledge base. Ranking

Attributes are:

1)Current Compensation
2)Expected Compensation
3)Education
4)Specialization
5)Location
6)Earliest Start Date
7)Work Gap
8)Total Experience
9)Relevant Experience
10)Communication
11)Current Employer
12)Stability
13)Education Gap

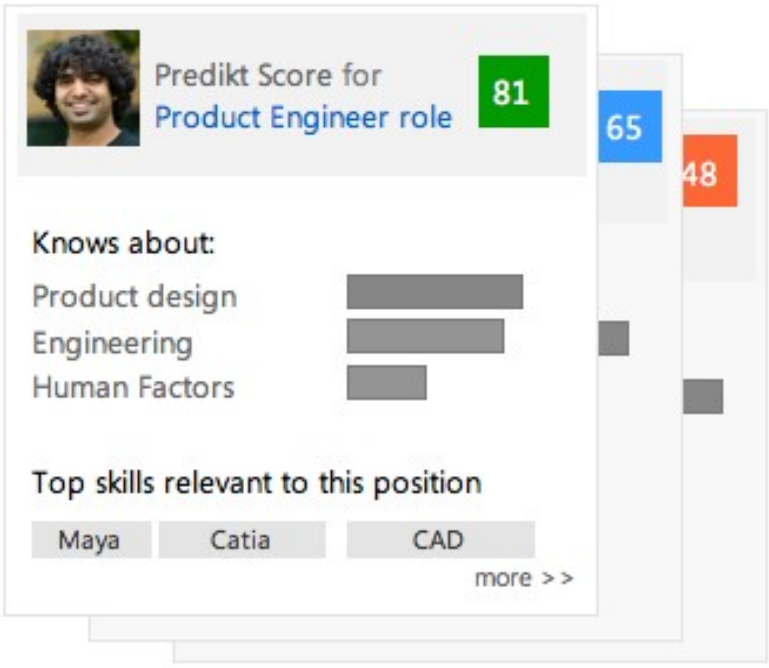


fig2.1. Desired Output

2.2 Intelligent searching

Put simply, Artificial Intelligence or "AI" is an add-on to system, complementing to provide the online recruitment solution . As the name suggests, AI enables a combination of an applicant-tracking system and an artificial intelligence resume parsing, searching and matching engine. The result is a supercharged tool giving incredibly accurate candidate matching to jobs, and ‘talent pool’ searching that makes other systems look like they’re from the stone-age.

2.2.1 Identifying “best” applicants in recruiting using data envelopment analysis

Selecting the most promising candidates to fill an open position can be a difficult task when there are many applicants. Each applicant achieves certain performance levels in various categories and the resulting information can be overwhelming. We demonstrate how data envelopment analysis (DEA) can be used as a fair screening and sorting tool to support the candidate selection and decision-making process. Each applicant is viewed as an entity with multiple achievements. Without any a priori preference or information on the multiple achievements, DEA identifies the non-dominated solutions, which, in our case, represent the ‘‘best’’ candidates. A DEA-aided recruiting process was developed that (1) determines the performance levels of the ‘‘best’’ candidates relative to other applicants; (2) evaluates the degree of excellence of ‘‘best’’ candidates’performance; (3) forms consistent tradeoff information on multiple recruiting criteria among search committee members, and, then, (4) clusters the applicants.

2.3 A Short Introduction to Learning to Rank

Learning to rank refers to machine learning techniques for training the model in a ranking task. Learning to rank is useful for many applications in Information Retrieval, Natural Language Processing, and Data Mining. Intensive stud-ies have been conducted on the problem and significant progress has been made. This short paper gives an introduction to learning to rank, and it specifically explains the fundamental problems, existing approaches, and future work of

learning to rank.

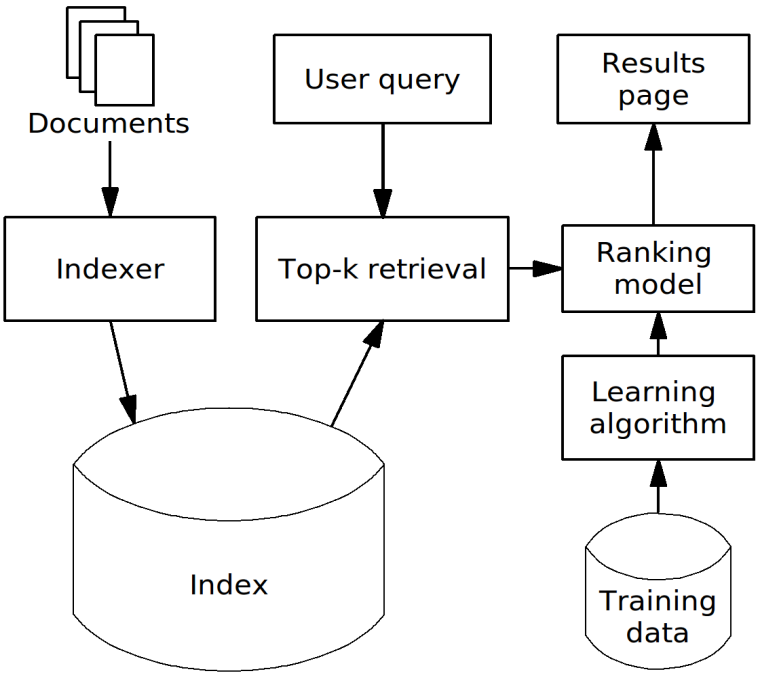


fig2.3. Overview of Ranking System

2.4 Weaknesses

1. Prior systems needed lot of human efforts and time.
2. Cost of hiring is high.
3. Potential candidate may loose the opportunity because of ambigious keyword matching.
4. Resumes needed to be in specific format.

2.5 How to overcome

1. Use of NLP to read resumes allow candidates the freedom to choose any format that's avalible to them.
2. Machine learning is used to rank candidates in accordiance to requirements
   Which reduces the efforts of sorting thousands of resumes.
3. Use of NLP can be used to get mean out of ambigius data.
4. Five benifits of A.I.
   - Goes Beyond Key Words
   - Fast and Accurate
   - Perfect For the New World of Social Recruiting
   - Customizes to your Needs
   - Gets Smarter

CHAPTER 3

Requirement Analysis

3.1 Software Requiremtents

The software reuirements in this project include:

- Python
- nltk
- ML
- PostgreSQL
- Elasticsearch

Python:

Python is used for creating backbone structure. Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, it uses whitespace indentation, rather than curly braces or keywords. Python has a large standard library, commonly cited as one of Python's greatest strengths.

WebCrawlers: Scrapy (Python Package)

Scrapy is an application framework for crawling web sites and extracting structured data which can be used for a wide range of useful applications, like data mining, information processing or historical archival. Even though Scrapy was originally designed for web scraping, it can also be used to extract data using APIs (such as Amazon Associates Web Services) or as a general purpose web crawler. Scrapy is controlled through the scrapy command-line tool, to be referred here as the “Scrapy tool” to differentiate it from the sub-commands, which we just call “commands” or “Scrapy commands”. The Scrapy tool provides several commands, for multiple purposes, and each one accepts a different set of arguments and options.

Natural Language Processing Tool :Natural Language Toolkit (NLTK) (Python Package)

NLTK was originally created in 2001 as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania. Since then it has been developed and expanded with the help of dozens of contributors. It has now been adopted in courses in dozens of universities, and serves as the basis of many research projects.
NLTK was designed with four primary goals in mind:

Simplicity

To provide an intuitive framework along with substantial building blocks, giving users a practical knowledge of NLP without getting bogged down in the tedious house-keeping usually associated with processing annotated language data .

Consistency

To provide a uniform framework with consistent interfaces and data structures, and easily guessable method names .

Extensibility

To provide a structure into which new software modules can be easily accommodated, including alternative implementations and competing approaches to the same task.

Modularity

To provide components that can be used independently without needing to

understand the rest of the toolkit.A significant fraction of any NLP syllabus deals with algorithms and data structures. On their own these can be rather dry, but NLTK brings them to life with the help of interactive graphical user interfaces that make it possible to view algorithms step-by-step. Most NLTK components include a demonstration that performs an interesting task without requiring any special input from the user. An effective way to deliver the materials is through interactive presentation of the examples in this book, entering them in a Python session, observing what they do, and modifying them to explore some empirical or theoretical issue.

Machine Learning tool :Scikit-learn (Python Package)

It is a Python module integrating classic machine learning algorithms in the tightly-knit scientific Python world (numpy, scipy, matplotlib). It aims to provide simple and efficient solutions to learning problems, accessible to everybody and reusable in various contexts: machine-learning as a versatile tool for science and engineering.

In general, a learning problem considers a set of n samples of data and try to predict properties of unknown data. If each sample is more than a single number, and for instance a multi-dimensional entry (aka multivariate data), is it said to have several attributes, or features.

We can separate learning problems in a few large categories:

- *Supervised learning* , in which the data comes with additional attributes that we want to predict .This problem can be either:
  –classification: samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of classification problem would be the digit recognition example, in which the aim is to assign each input vector to one of a finite number of discrete categories.
  – regression: if the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.
- *Unsupervised learning* , in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or thee dimensions for the purpose of visualization

PostgreSQL

PostgreSQL, often simply Postgres, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance. As a database server, its primary function is to store data securely, supporting best practices, and to allow for retrieval at the request of other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. PostgreSQL manages concurrency through a system known as multiversion concurrency control (MVCC). PostgreSQL includes built-in support for regular B-tree and hash indexes, and two types of inverted indexes: generalized search trees (GiST),generalized inverted indexes (GIN) and Space-Partitioned GiST (SP-GiST). Other storage features of PostgreSQL includes Referential integrity constraints including foreign key constraints, column constraints, and row checks, Binary and textual large-object storage, Tablespaces, Per-column collation, Point-in-time recovery, implemented using write-ahead logging, etc.

Elasticsearch DSL

It is a high-level library whose aim is to help with writing and running queries against Elasticsearch. It is built on top of the official low-level client (elasticsearch-py).
It provides a more convenient and idiomatic way to write and manipulate queries. It stays close to the Elasticsearch JSON DSL, mirroring its terminology and structure. It exposes the whole range of the DSL from Python either directly using defined classes or a queryset-like expressions.
It also provides an optional wrapper for working with documents as Python objects: defining mappings, retrieving and saving documents, wrapping the document data in user-defined classes.
To use the other Elasticsearch APIs (eg. cluster health) just use the underlying client.

3.2 Hardware Requiremtents

Linux: GNOME or KDE desktop GNU C Library (glibc) 2.15 or later, 2 GB RAM minimum, 4 GB RAM recommended, 1280 x 800 minimum screen resolution.

Windows: Microsoft R Windows R 8/7/Vista (32 or 64-bit) 2 GB RAM minimum, 4 GB RAM recommended, 1280 x 800 minimum screen resolution, Intel R processor with support for Intel R VT-x, Intel R EM64T (Intel R 64) Execute Disable (XD) Bit functionality.

3.3 Supportive Operating Systems :

The supported Operating Systems for client include:

- Windows xp onwards
- Linux any flavour.

Windows and Linux are two of the operating systems that will support comparative website.
Since Linux is an open source operating system, This system which is will use in this project is developed on the Linux platform but is made compatible with windows too.The comparative website will be tested on both Linux and windows. The supported Operating Systems for server include: The supported Operating Systems For server include Linux. Linux is used as server operating system. For web server we are using apache 2.0

CHAPTER 4

Project Design

4.1 Design Approach

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementationand its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.2 Software Architectural Designs

Our system follows the three tier architecture . First tier consist of GUI, Processing block and the Database.

GUI:

The GUI(Graphical User Interface) in our project deals with the interface for the user where the user will login and submit his resume in any formate(pdf,doc,docx,ect.) and social profiles links. The GUI provides a platform for the user to communicate with the database. It acts as a connector as well as communicator which connects the database and helps in transfer of data between the GUI and the database.

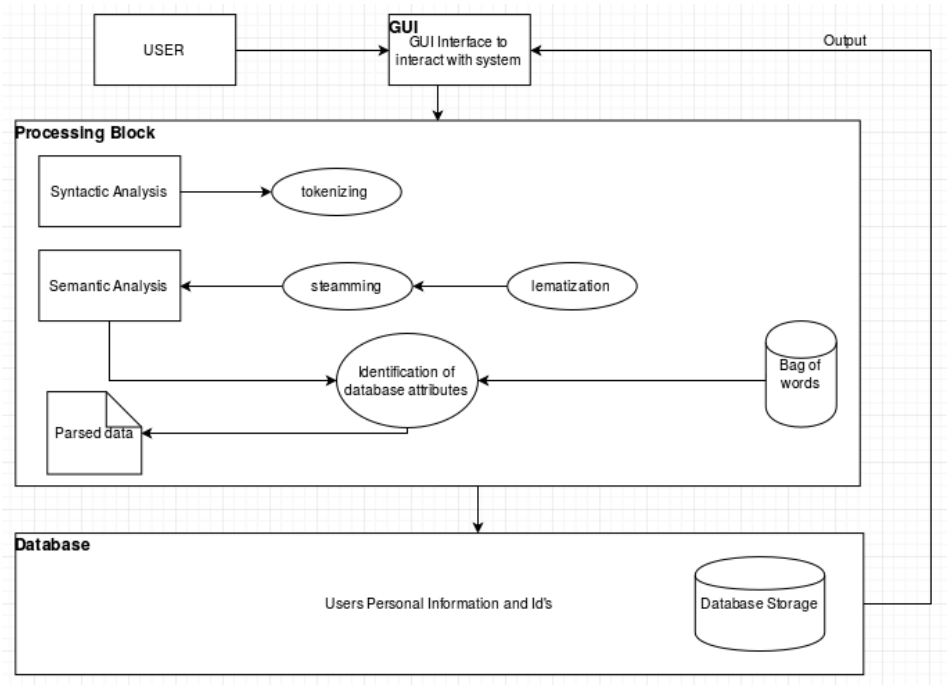
Processing block:

Processing block is the block where the actual processing of our project is done. This block connects the gui to the database i.e. it acts as a connector as well as communicator which connects the database and helps in transfer of data between the gui and the database. Its main function is to take input from resumes and social profile of the candidate and parse it to store the information and store it in the structured format(json), and database. After storing this information this system will give output using web application.

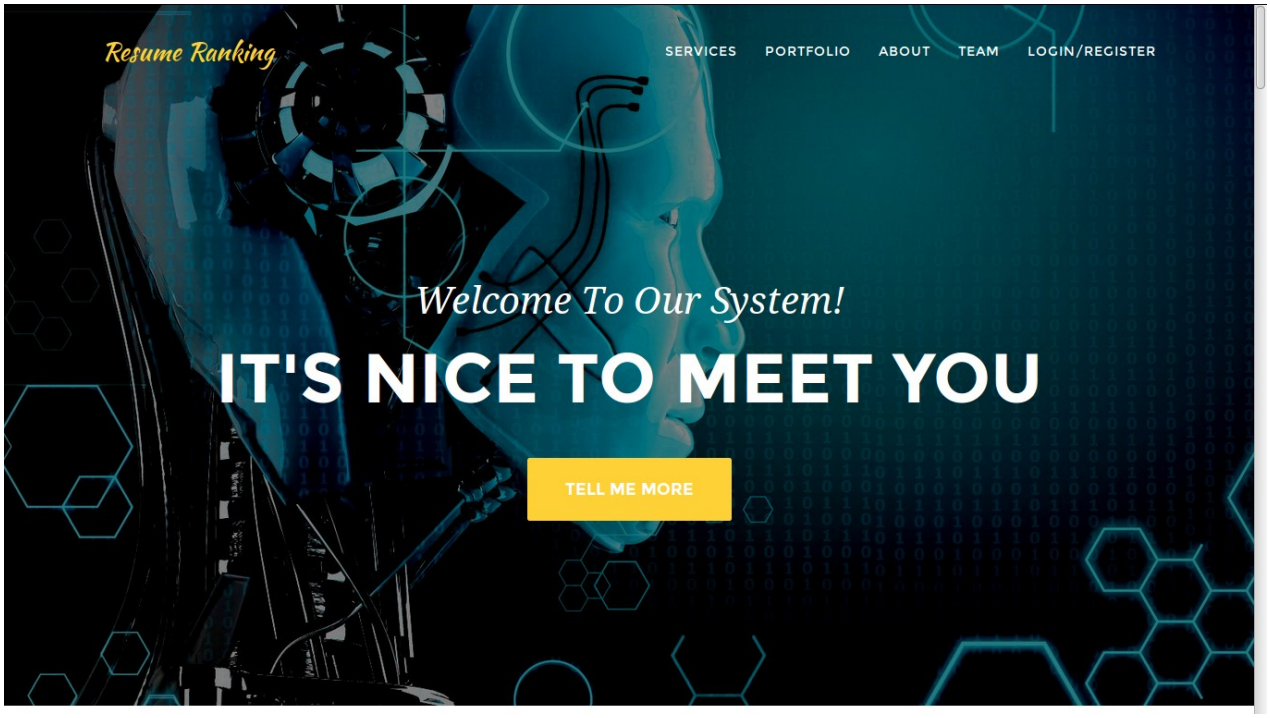
Database: Database tier is the tier used for the storage of data. This tier contains all the data that is need for the processing of the whole project. The data in this tier is related to the student information gathered form his/her resumes and social profiles.

Software Architecture Design

Fig4.2.1 Software Architecture Design



Front End Designs



4.2.2 Front End Designs

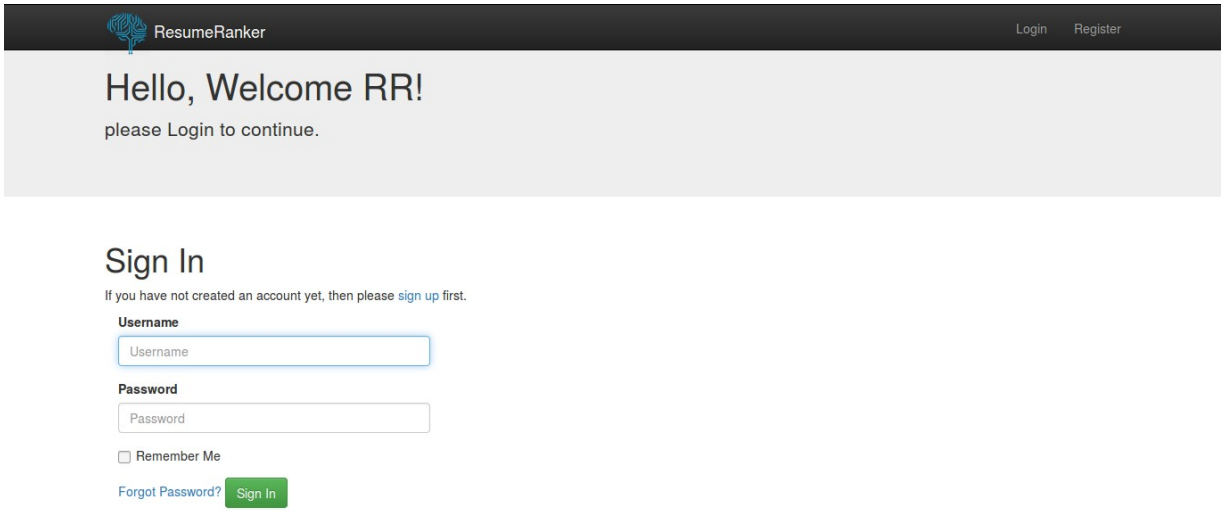


fig4.2.2 Front End

4.2.3 Component Diagram

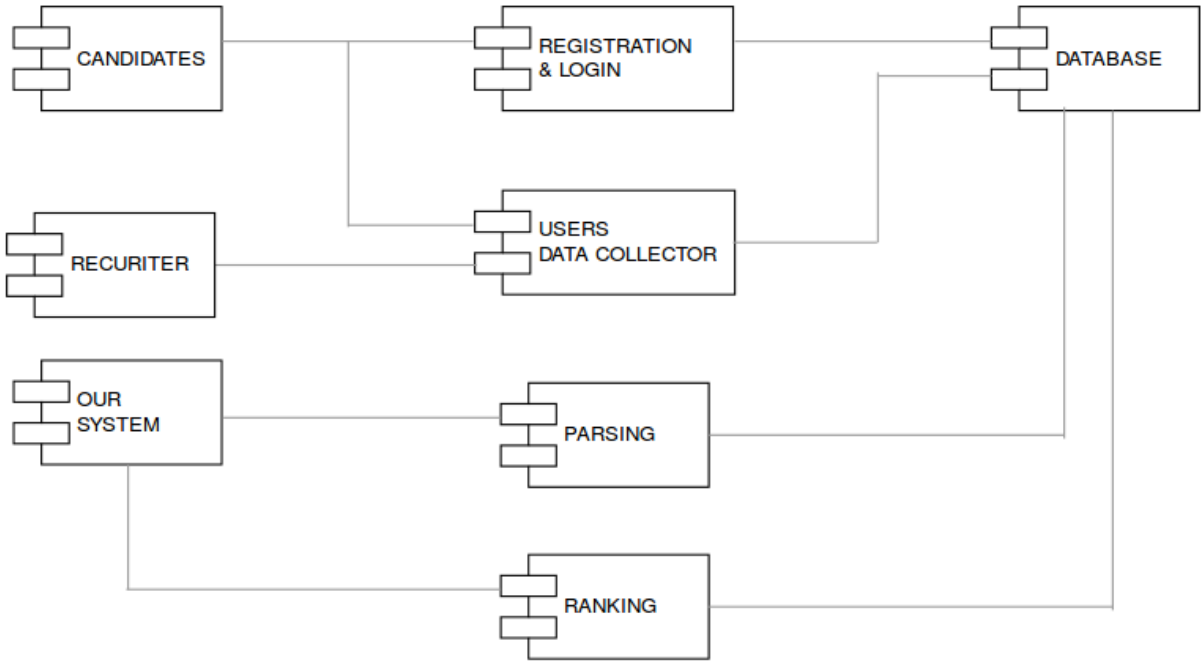


fig4.2.3 Component Diagram

4.2.4 Deployment Diagram

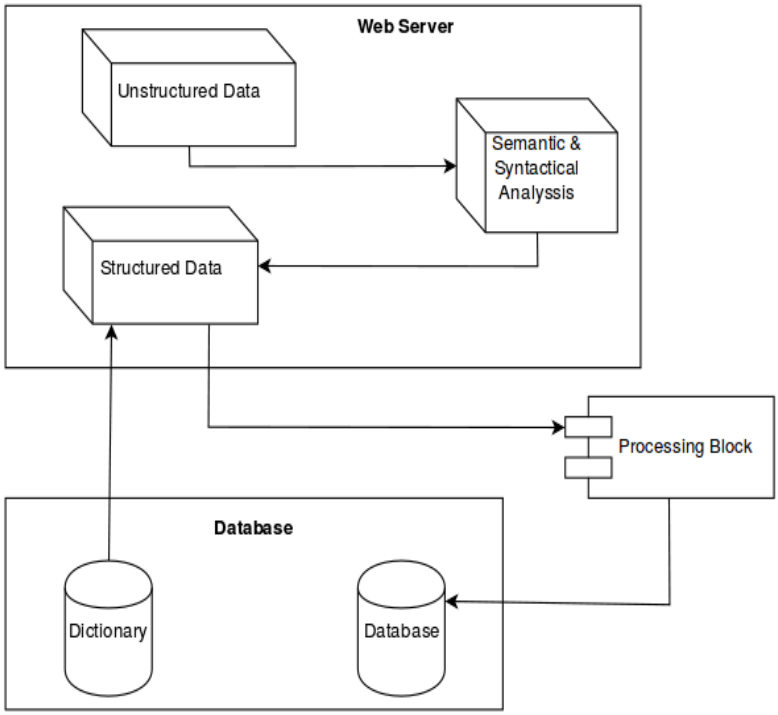


fig4.2.3 Deployment Diagram

4.2.5 State Chart Diagram

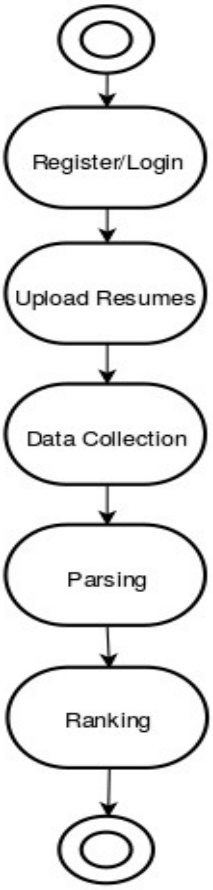


fig 4.2.3 State Chart Diagram

CHAPTER 5

Implementation Details

5.1 Assumptions and Dependencies

5.1.1 Assumptions

The following Assumption was taken into consideration:

- The Parser parses the resumes and convert them into the text file and then that text file is read to get the attributes of the candidate and store them in structured form in the json file.
- This json file contains the attributes of the student in ranked format and then it is read to show the out put to the student/employer.

5.1.2 Dependencies

The dependencies are as follows:

- For User interface Django webframework is used.
- Python programing language, NLTK, Apache Tika is used to parse the document and store the information in structured form.
- To get user information form linkedIn, github there api is used which provides data in structured format.

5.2 Implementation Methodologies

5.2.1 Modular Description of Project

Different modules or components created are domain establishment,data collection, parsing, ranking and database component. Parsing and Ranking is the heart of our system which is created using python, nltk, tika libraries. This component does the morphological analysis, syntactic analysis, semantic analysis and generates the parsed and ranked data of the candidate according to his/her skills. Then this information is stored in the database and retrived and shown to the users whenever required.

5.3 Detailed Analysis and Description of Project

Domain Establishment: This module is responsible for creating user accounts and database creation as the proposed system is domain independent and would be used by multiple users.

Registration or Login Module: If the new user want to interact with our system he needs to simply register into our system by completely filling details i.e. validation. If the user is already existing he needs to login.

Parsing & Ranking: Parsing module is responsible for parsing the document and storing it in json format which will later be used by the ranking module. Ranking module will then use the json file and rank the candidates information according to his/her skills and the information will be stored in the database.

Morphological Analysis: Morphology in linguistics is the study and description of how words

are formed in natural language. In this phase the sentence is broken down into tokens- smallest unit of words, and determine the basic structure of the word.

Syntactic Analysis: The objective of the syntactic analysis is to find the syntactic structure of the sentence.It is also called Hierarchical analysis/Parsing, used to recognize a sentence, to allocate token groups into grammatical phrases and to assign a syntactic structure to it.

Semantic Analysis: Semantic Analysis is related to create the representations presentations for meaning of linguistics inputs. It deals with how to determine the meaning of the sentence from the meaning of its parts.

5.3.1 Usecase Report

Title	Resume Ranking Using NLP and ML
Description	The current recruitment process are more tedious and time consuming which forces the candidates to fill all their skill and information manually. And HR team requires more man power to scrutinize the resumes of the candidates. So that motivated to build a solution that is more flexible and automated which will ease the burden on the employer for searching potential candidate and the burden of the cadidate to find job suitable to his/her interests.
Primary actor	Candidate in search of good job and Employer in search of potential candidate.
Pre-condition	There is no special requirement in submitting the resumes as our system is accepting different formats of resumes.
Post-condition	Candidate will see himself/herself ranked in his/her mentioned skills and the employer will get list of all potential candidate according to his/her need.
Main success scenario:	• Candidte submits resumes and social profile(linkedIn, stackoverflow, github) links. • The parser and ranker will parse and rank the candidate skills. • It is then stored in database and whenever required, it is retrived and displayed to users(employer and candidate).
Frequency of use	User can interact with the system at any time and get information.
System requirement	Normal, no specific requirement.

Table 5.3 Usecase Report

5.4 Class Diagram

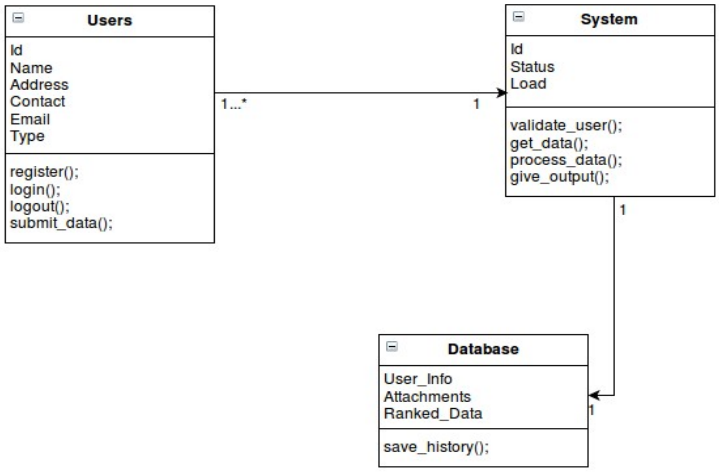


fig5.4 Class Diagram

5.4.1 Class Diagram Report

Title	Resume Ranking Using NLP and ML
Description	The current recruitment process are more tedious and time consuming which forces the candidates to fill all their skill and information manually. And HR team requires more man power to scrutinize the resumes of the candidates. So that motivated to build a solution that is more flexible and automated which will ease the burden on the employer for searching potential candidate and the burden of the cadidate to find job suitable to his/her interests.
Primary actor	Candidate in search of good job and Employer in search of potential candidate.
Pre-condition	There is no special requirement in submitting the resumes as our system is accepting different formats of resumes.
Post-condition	Candidate will see himself/herself ranked in his/her mentioned skills and the employer will get list of all potential candidate according to his/her need.
Django OR Web Application	The submited Resumes are firste parsed using python and the they are ranked and stored in database.
Python script	• It gets the resumes from web interface and pass it to the parser. • The parsed document is then ranked.
Database	Databse is used for retrieving the information whenever required and displayed on web inteface.

Table 5.4 Class Diagram Report

CHAPTER 6

Results and Discussion

6.1 Test cases and Results

When the candidate submits his/her resume it is ranked and stored in the database and is later retrived when required. We have tested our web application by considering following test case:

6.1.1 Unit Testing

We are firstly parsing the resumes and transforming them to text file and then reading them and parsing and storing the info in json format. After this we are ranking these resumes and storing them in our database. Later this data is shown to the user in web user interface.

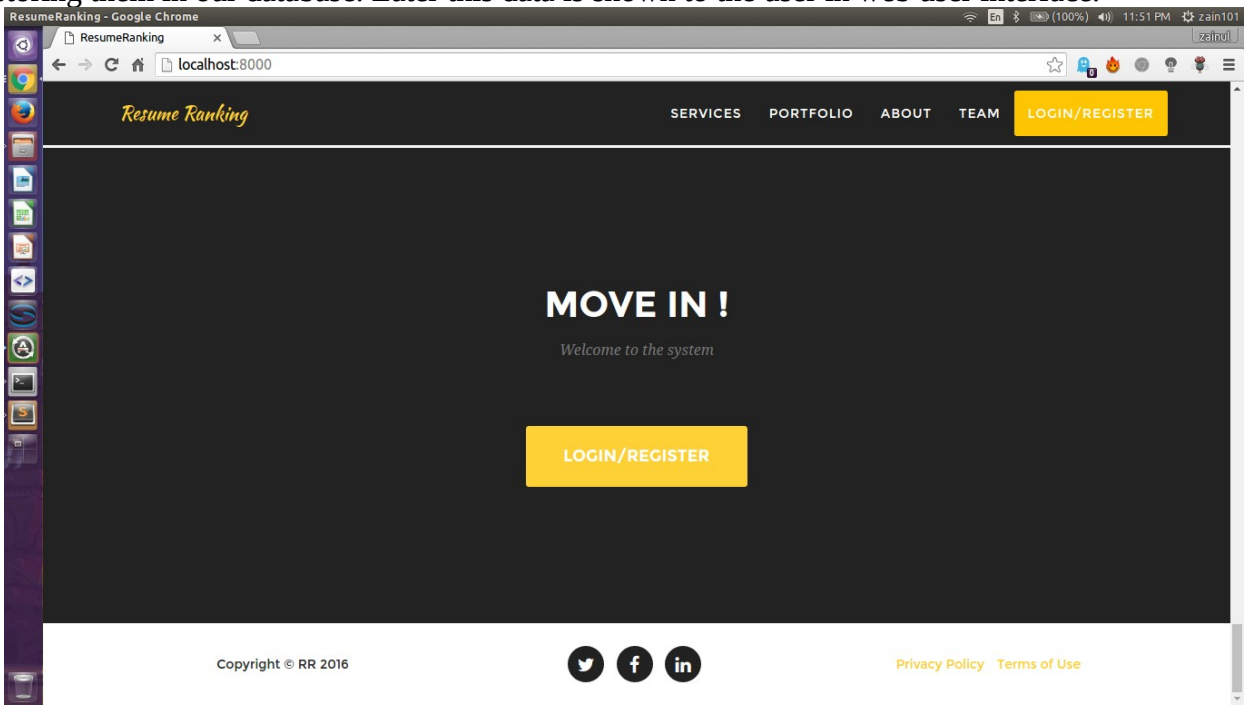


fig6.1.1 Home Page

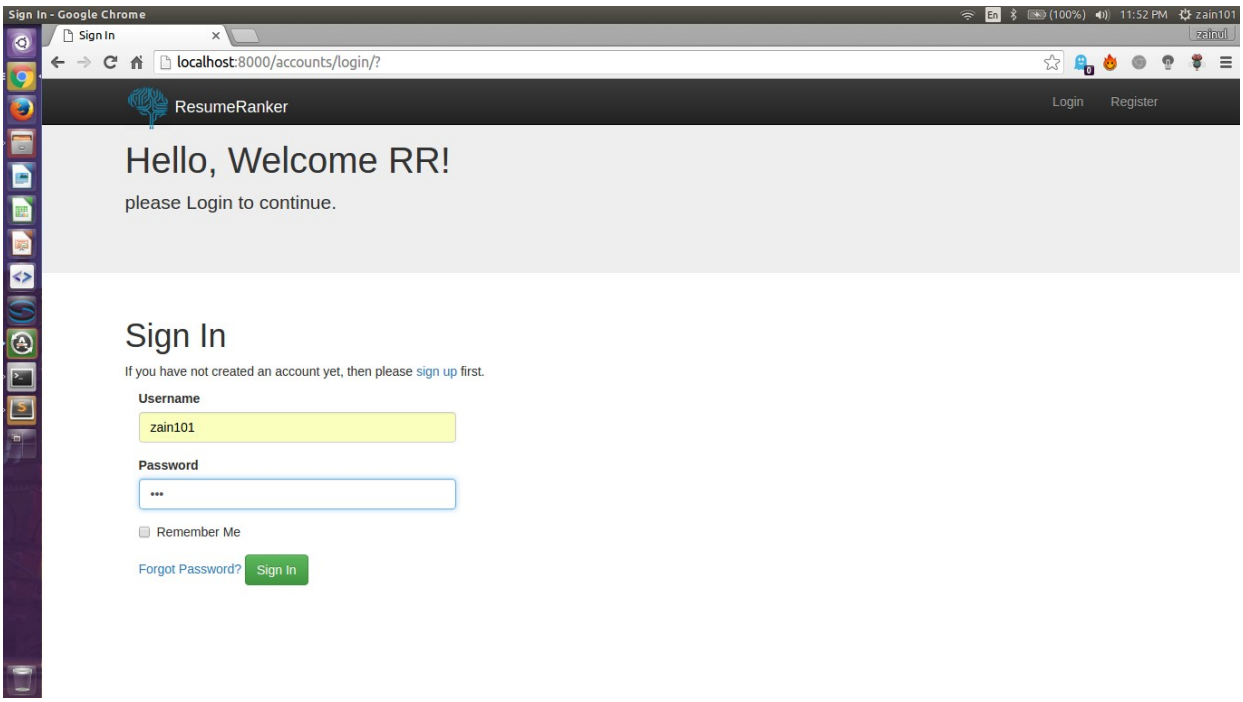


fig6.1.2 Login

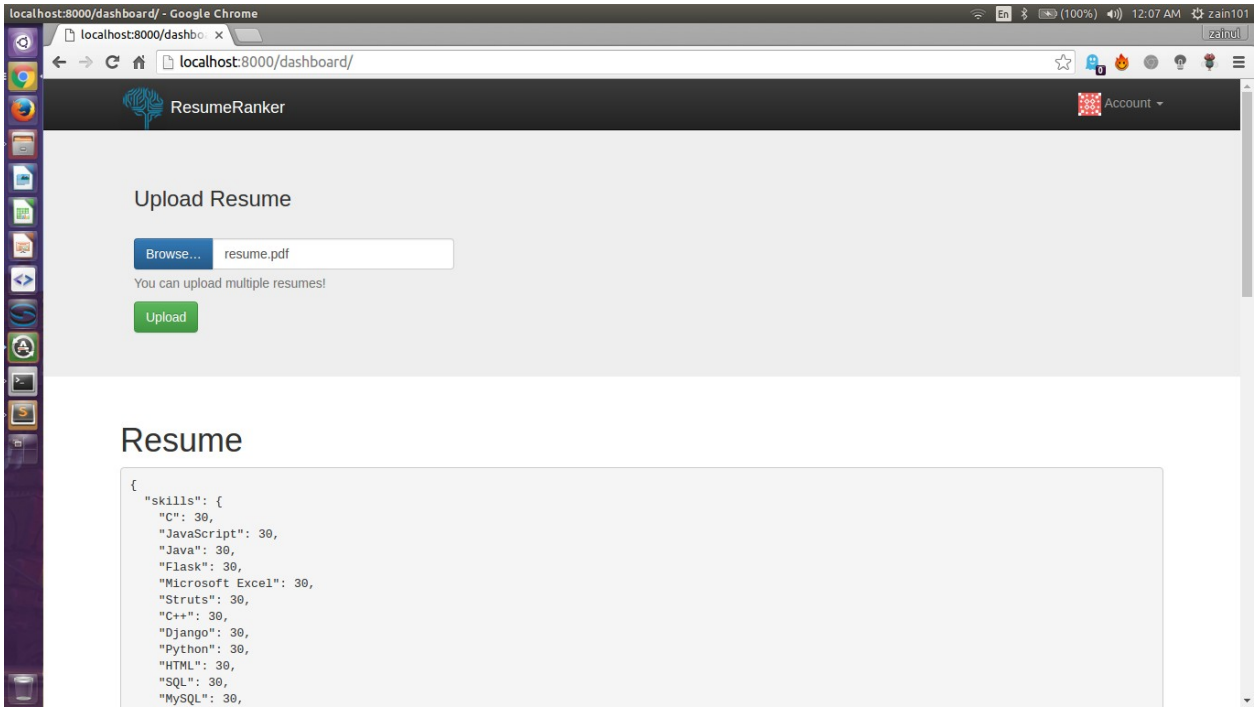


fig6.1.3 Upload Resume and giving Parsed and Ranked resume

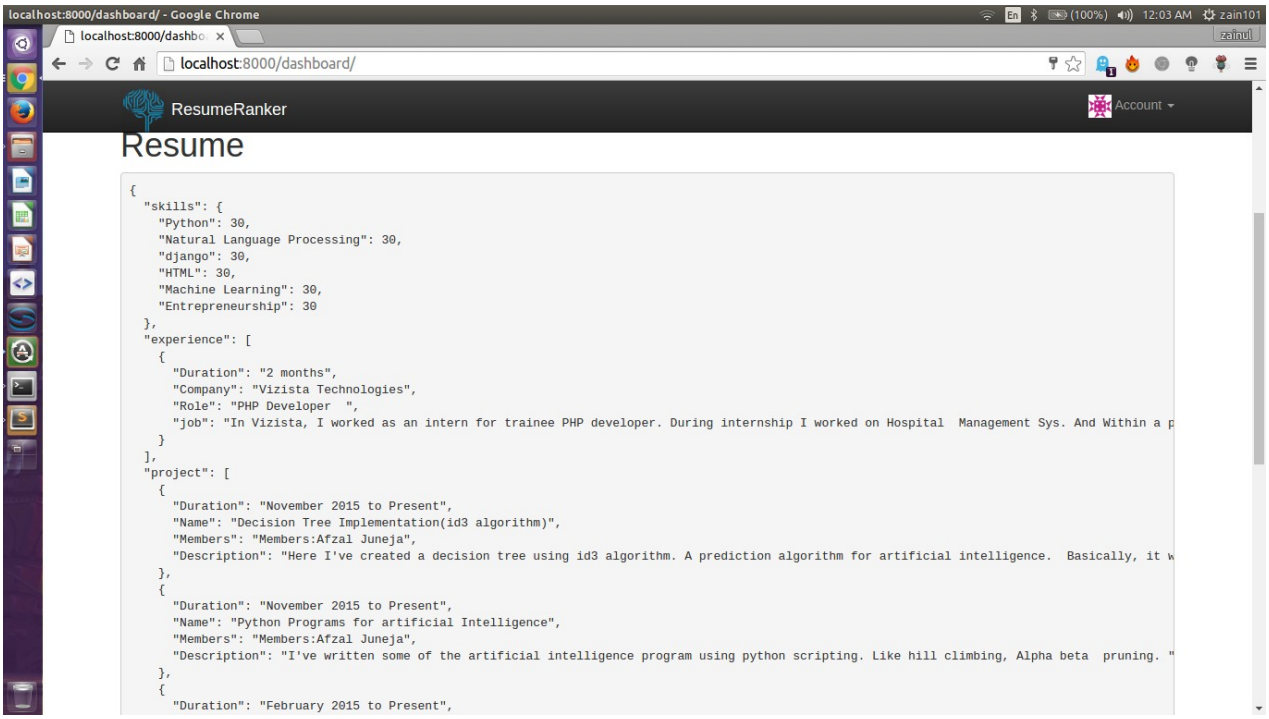


fig6.1.4 Upload Resume and giving Parsed and Ranked resume

CHAPTER 7

Project Time Line

7.1 Project Time Line Matrix

	Name	Duration	Start	Finish	Predecessors	Resource Names
1	1(a) Requirement gathering	7 days	1/1/16 8:00 AM	11/1/16 5:00 PM		PM;TL;TM1
2	1(b) Confirm requirements	1 day	11/1/16 8:00 AM	11/1/16 5:00 PM		TL;TM1;TM2
3	2(a) Front-end user interface	3 days	12/1/16 8:00 AM	14/1/16 5:00 PM		PM;TM3
4	2(b) Back-end database designing	3 days	15/1/16 8:00 AM	19/1/16 5:00 PM		TM2
5	3(a) Front-end coding	20 days	20/1/16 8:00 AM	16/2/16 5:00 PM		TM2;TM3
6	3(b) Database creation	20 days	17/2/16 8:00 AM	15/3/16 5:00 PM		TM2;TM3
7	3(c) Coding for screens,tables	40 days	16/3/16 8:00 AM	10/5/16 5:00 PM		TL;TM1
8	3(d) Creation of test cases	10 days	12/5/16 8:00 AM	25/5/16 5:00 PM		TL;TM1
9	4(a) Unit testing	3 days	26/5/16 8:00 AM	30/5/16 5:00 PM		TM1;TM2
10	4(b) System testing	3 days	31/5/16 8:00 AM	2/6/16 5:00 PM		TM1;TM3
11	4(c) Alpha and Beta testing	3 days	3/6/16 8:00 AM	7/6/16 5:00 PM		TL;TM1;TM2
12	5(a) Deployment	1 day	8/6/16 8:00 AM	8/6/16 5:00 PM		PM

fig7.1 Time Line

7.2 Project Time Line Chart

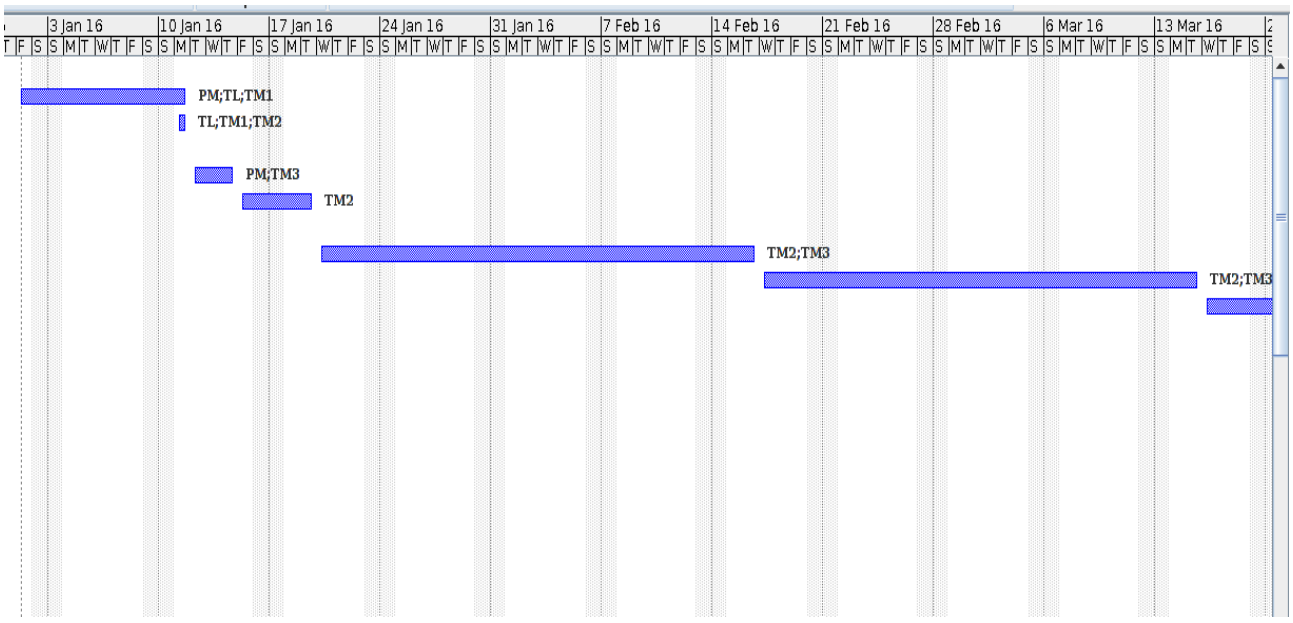


fig7.2.1 Time Line Chart

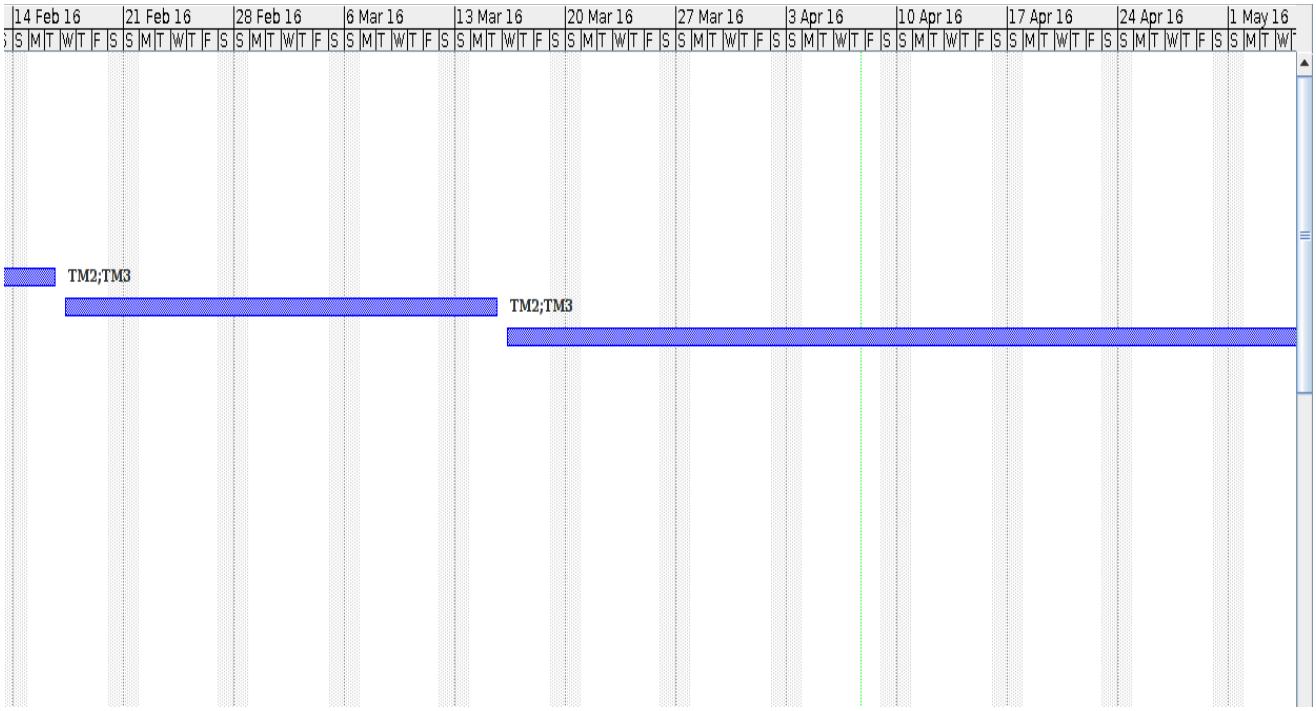


fig7.2.2 Time Line Chart

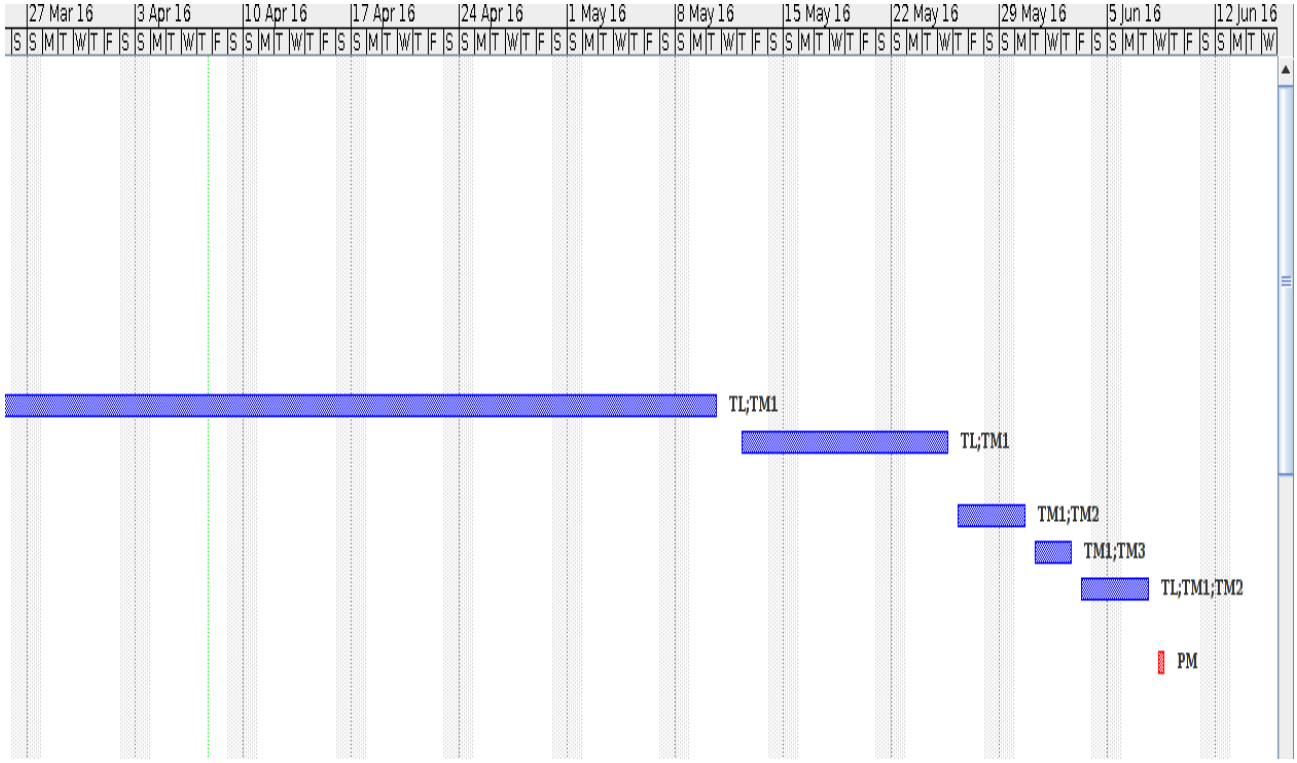


fig7.2.3 Time Line Chart

CHAPTER 8

Task Distribution

8.1 Distribution of Workload
8.1.1 Scheduled Working Activities

Activity	Time Period	Comment
Requirement gathering	8 days	Requirement gathering was to be done through searching on internet and taking the ideas, sharing the views among group members.
Planning	4 days	Planning was done by reviewing of literature of IEEE papers and by taking the walkthrough.
Design	6 days	Designing was accomplished by creating UML diagram, charts.
Implementation	70 days	Implementation was started with creating the backend, script and then frontend.
Testing	9 days	Testing has been done by performing unit testing,alpha nad beta testing, integrated testing and system testing.
Deployment	1 days	Deployment phase has been done by installing project on the server.

Table 8.1 Scheduled Working Activities

8.1.2 Members actvities or task

Member	Activity	Time period	Start date	End date	Comment
M1,M2, M3,M4	Requirement gathering	4 days	1/1/16	4/1/16	M1 and M2 has perfomed the seaching for project requirement on the internet by reviewing the related literature and by anlysing the related prject which is already available in the market. Regularly inform to the other member of team.
M1,M2, M3,M4	Analysing of the requirement	3 days	5/1/16	7/1/16	M1, M2, M3, M4 done the requirement analysing of project by sharing the ideas, and by discussing on related information which is gather by the M1, And M2. M3 and M4 has created the list of requirement after every meeting

M1,M2, M3,M4	Finalizing the requirement	1 day	8/1/16	8/1/16	Whole team finalized the requirement. M1 and M3 has created a list of finalise requirement.
M1,M2, M3,M4	Planning	4 days	9/1/16	12/1/16	Planning was done by walk-through and by analysing the available applications. M2 and M3 creates a list of funtion which will be implement in the project. Each and every module were discuss in every group meeting and M1 and M2 created a blue print for project .
M3,M4	Frontend design	3 days	13/1/16	15/1/16	M3 and M4 created the UML diagram for frontend of the system and data flow diagrams and informed to the whole team respectively.
M1,M2	Backend design	3 days	13/1/16	15/1/16	M1 and M2 created the UML diagram for backend of the system and data flow diagrams and informed to the whole team reapectively.
M3,M4	Installation of tools and technology for frontend	4 days	16/1/16	19/1/16	M3 and M4 installed the all the require tools and packages which is used for frontend design.
M1,M2	Installation of tools and technology for backend	4 days	16/1/16	19/1/19	M1 and M2 installed the all the require tools and packages which is used for backend design.
M3,M4	Implementa-tion of GUI	6 days	20/1/16	25/1/16	M3 and M4 created the GUI of the project and informed to other member.
M1	Implementa-tion of script for parsing	6 days	24/1/16	29/1/16	M1 implemented the script for parsing using nltk packages and explained the code to other team members.
M2	Implementa-tion of script for ranking	6 days	30/1/16	4/2/16	M2 implemented script ranking using and explained the code to other team members.
M3,M4	Implementa-tion of bag of words	7 days	26/1/16	2/2/16	M3 and M4 implemented the bag of words/dictionaries using database tables and

					attributes, synonyms,etc and informed to other team members.
M1,M2	Implementa-tion of script for mapping with dictionaries	25 days	5/2/16	29/2/16	M1 and M2 coded the script for mapping the keys and values to the appropriate dictionaries of tables and attributes and explained the code to other team members.
M3,M4	Connectivity of GUI with script	8 days	4/2/16	11/2/16	M3 and M4 did the connectivity of script with Django.
M4	Database connectivity	3 days	12/2/16	14/2/16	M4 has done the database connectivity with script and Django.
M3,M4	GUI connectivity	3 days	15/2/16	17/2/16	M3 and M4 created the connectivity GUI with database.
M3	Data gathering into database	3 days	18/2/16	20/216	M3 gathered the data required in the database with respect to the domain selected.
M1,M2, M3,M4	Integration of all modules	15 days	5/3/16	19/3/16	M1, M2,M3 and M4 integrated all the module. Implemented whole system properly.
M1,M2	Unit testing	5 days	20/3/16	25/3/16	M1 and M2 performed the unit testing and noted down results and discussed with other members of team.
M3,M4	Functional testing	4 days	26/3/16	29/3/16	M3 and M4 performed the functional testing and noted down results and discussed with other members of team.
M1,M2, M3,M4	Deployment	1 day	30/3/16	30/3/16	M1, M2,M3 and M4 Deployed the system live for use.

Table 8.2 Member Activities and Task

CHAPTER 9

Conclusion and Future Scope

9.1 Conclusion

Our system will provide better and efficient solution to current hiring process. This will provide potential candidate to the organisation and the candidate will be successfully be placed in an organisation which appreciate his/her skillset and ability.

9.2 Future Scope

The application can be extended further to other domains like Telecom, Healthcare, E-commerce and public sector jobs.

REFERENCES

[1] IEICE TRANS. INF. & SYST., VOL.E94–D, NO.10 OCTOBER 2011
Special Section on Information-Based Induction Sciences and Machine Learning ***A Short Introduction to Learning to Rank,*** Hang LI

[2] Identifying ‘‘best’’ applicants in recruiting using data envelopment analysis ***Sharon A. Johnson, JoeZhu***. http://www.sciencedirect.com/science/article/pii/S0038012102000484

[3] **Referenced Links :**

Jessica Simko , ***“****How Hiring Managers Make Decisions”*
http://www.careerealism.com/hiring-managers-decisions/
Vinayak Joglekar , “*Ranking Resumes using MachineLearning****”***
https://vinayakjoglekar.wordpress.com/2014/06/24/ranking-resumes-using-machine
Peter Gold , *“Artificial Intelligence Recruiting”*
https://www.linkedin.com/pulse/artificial-intelligence-recruiting-peter-gold
Turbo Ricruit ,*“Automated Application Processing”,”Better candidate experience”,”Matching Job Descriptions to Resumes”*
http://www.turborecruit.com.au/benefits-of-artificial-inteligence-for-recruitment/

[4] ***Rchillies***, http://www.rchillies.com

[5] ***Belong.co***, http://www.belong.co

[6] ***ALEX System*** , http://www.hireability.com/alex/

[7] ***Turbo Ricruit*** , http://www.turborecruit.com.au/intelligent-searching/

[8] ***Revolvy*** , http://www.revolvy.com/main/index.php?s=Parse%20tree

[9] Student Thesis “***Information Quality Management inInformation Extraction:A Survey***"..http://www.rn.inf.tudresden.de/uploads/Studentische_Arbeiten/Belegarbeit_Jansen_Nicolas.pdf

CHAPTER 10

Appendix I

10.1 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of “don't repeat yourself”.

Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models. Django was designed to help developers take applications from concept to completion as quickly as possible. It takes security seriously and helps developers avoid many common security mistakes. Some of the busiest sites on the Web leverage Django’s ability to quickly and flexibly scale.

10.1.1 Features of Django

- Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses "views", the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper(ORM) that mediates between data models and a relational database system for processing HTTP requests with a web templating system and a regular-expression-based URL dispatcher.
- The main Django distribution also bundles a number of applications in its "contrib" package, including an extensible authentication system, the dynamic administrative interface, etc.
- Django's configuration system allows third party code to be plugged into a regular project, provided that it follows the reusable app conventions.
- Django can be run in conjunction with Apache, NGINX using WSGI, Gunicorn, or Cherokee using flup(a Python module).Django also includes the ability to launch a FastCGI server, enabling use behind any web server which supports FastCGI, such as Lighttpd or Hiawatha.
- Django officially supports four database backends: PostgreSQL, MySQL, SQLite, and Oracle. Microsoft SQL Server can be used with django-mssql on Microsoft operating systems, while similarly external backends exist for IBM DB2, SQL Anywhere and Firebird.
- Django may also be run in conjunction with Jython on any Java EE application server such as GlassFish or JBoss.

ACKNOWLEDGMENT

I would like to take the opportunity to express my sincere thanks to my guide **Prof. Tabrez Khan**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

I am grateful to him for his timely feedback which helped me track and schedule the process effec tively. His time, ideas and encouragement that he gave is help me to complete my project efficiently.

I would also like to thank **Dr. Abdul Razak Honnutagi**, AIKTC, Panvel, for his encouragement and for providing an outstanding academic environment, also for providing the adequate facilities.

I am thankful to **Prof. Tabrez Khan**, HOD, Department of Computer Engineering, AIKTC, School of Engineering, Panvel and all my B.E. teachers for providing advice and valuable guidance.

I also extend my sincere thanks to all the faculty members and the non-teaching staff and friends for their cooperation.

Last but not the least, I am thankful to all my family members whose constant support and encour agement in every aspect helped me to complete my project.

Juneja Afzal Ayub Zubeda (12CO32)

Momin Adnan Ayyas Shaheen (12CO46)

Gunduka Rakesh Narsayya Godavari (12CO29)

Sayed ZainulAbideen MohdSadiq Naseem (13CO72)

(Department of Computer Engineering)
University of Mumbai.