WEB BASED MINING IN PERSONALISED ONLINE RECRUITING SYSTEM

Submitted in partial fulfillment of the requirements
of the degree of

B. E. Information Technology

By

Melita D'Souza 31

Guide

Ms. Vaishali Jadhav
Assistant Professor

Department of Information Technology
St. Francis Institute of Technology
(Engineering College)

University of Mumbai
2015-2016

WEB BASED MINING IN PERSONALISED ONLINE RECRUITING SYSTEM

Submitted in partial fulfillment of the requirements
of the degree of

B. E. Information Technology

By

Melita D'Souza 31

Guide

Ms. Vaishali Jadhav
Assistant Professor

Department of Information Technology
St. Francis Institute of Technology
(Engineering College)

University of Mumbai
2015-2016

CERTIFICATE

This is to certify that the project entitled **"Web Based Mining in Personalised Online Recruiting System"** is a bonafide work of **Melita D'Souza (31)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of B.E. in Information Technology.

Ms. Vaishali Jadhav
Guide

Mr. Pramod Shanbhag
Head of Department

Dr. A. K. Sen
Principal

Project Report Approval for B.E.

This project report entitled ***Web Based Mining in Personalised Online Recruiting System*** by ***Melita D'Souza*** is approved for the degree of ***B.E. in Information Technology.***

Examiners

1.----------

2.----------

Date:

Place:

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.

Melita D’Souza (31)

Date:

Abstract

Online recruiting methods have become an important part in the recruitment system. However, the lack of personal service in a web environment is one of development bottlenecks of online recruiting system.

A real world challenging task of the web master of an organization is to match the needs of user and keep their attention in their web site. So, only option is to capture the intuition of the user and provide them with the recommendation list. Most specifically, an online navigation behavior grows with each passing day, thus extracting information intelligently from it is a difficult issue. Web master should use Web Usage Mining (WUM) method to capture intuition. A WUM is designed to operate on web server logs which contain user's navigation. Hence, recommendation system using WUM can be used to forecast the navigation pattern of user and recommend those to user in a form of recommendation list. We propose a two tier architecture for capturing users intuition in the form of recommendation list. Intuition List consists of list of pages visited by user as well as list of pages visited by other users having similar usage profile. The practical implementation of proposed architecture and algorithm shows that accuracy of user intuition capturing is improved.

To elaborate, first, we analyze single online candidate's personal requirements. According to their requirements, a personalized recommendation system framework is proposed based on the technology of web usage mining. The system provides individual recommendations in accordance with the analysis of single job seeker's searching custom and interest, so the quality of service could be improved.

Contents

Acknowledgements

List of Figures

List of Tables

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	WUM	Web Usage Mining
2	WLF	Web Log File
3	PORS	Personalized Online Recruiting System
4	DB	Database
5	DFD	Data Flow Diagram

Chapter 1
Introduction

With the explosive growth of knowledge available on World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently. Meanwhile, the substantial increase in the number of websites presents a challenging task for web masters to organize the contents of websites to cater to the need of users. Analyzing and modeling web navigation behavior is helpful in understanding demands of online users. Following that, the analyzed results can be seen as knowledge to be used in intelligent online applications, refining website maps, and web based personalization system and improving searching accuracy when seeking information.

1.1 Description

Online recruiting websites tend to provide dozens of job opportunities; however, each candidate is often only interested in one position or several positions. Thus, lots of information on the website is useless to a single job seeker. How to find jobseekers' interests, and bring convenience and enlightenment for them in a web environment is an important necessity in today's fast moving world.

The job seekers leave many web access log files which reflect their searching customs and interests. In order to obtain single job seeker's searching custom and interest, this project presents a personalized recommendation system based on web log mining. It is to find not only single job seeker's frequent sequential paths that he had visited in the past, but also associated positions and pages that he might hadn't visited before. Thus different people could enjoy different recommendations in the improved online job searching environment.

In this project we are developing a website. The website will be used as a database which will be developed using HTML5, PHP5, JavaScript and CSS. We present architecture for capturing recommendations in the form of intuition list for user. Intuition List consists of list of pages visited by user as well as list of pages visited by other user having similar usage profile. The results represent

the improved accuracy of recommendations. Also, we use collaborative filtering, which is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).

1.2 Problem formulation

Consider an online job seeker who is looking for a job suiting his criteria for which he visits various websites to refer to their content. The colossus web offers umpteen information whereas the user is only interested in a subset of the information, which may confuse the user if it is not relevant thus disinteresting the user from visiting the websites with potentially useful data. Hence, to solve this issue we create a recommender system for online job seekers to help them.

- Access relevant information efficiently.
- Enjoy different recommendations based on their requirements and usage profile.
- View pages visited by other users having similar usage profiles.

1.3 Motivation

In today's scenario everyone is dependent on the Internet. People want everything easy and systematic. Web usage mining is a technique used to deduce useful and relevant information to guide the users. It is a cost-effective way of analyzing large amounts of data, especially when a human cannot analyze such datasets.

Diverse use of the Internet has made automatic knowledge extraction from web log files a necessity. WUM learns Web users' needs and preferences. This can improve the effectiveness of their Web sites by adapting the Information structure of the sites to the users' behavior. Collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.

1.4 Proposed Solution

The drawback of the existing system is that the user may have to visit and check multiple websites. If he is lucky enough he may get information satisfying his criteria. But there are also chances of the user exhausting him in an unending search and still not receiving the relevant information.

Web Usage Mining (WUM) is process of extracting knowledge from a Web user's access data, by exploiting Data Mining technologies. It can be used for different purposes such as personalization, system improvement and site modification. This project is an application of Web Usage Mining represented as a recommender system.

Thus, the main goal of the recommender system is to improve Web site usability. Typically, the Web usage mining prediction process is structured according to two components performed online and off-line with respect to Web server activity. Offline component builds the knowledge base by analyzing historical data, such as server access log file or web logs which are captured from the server, then these web logs are used in the online component for capturing the intuition list of the user so as to recommend page views to the user whenever he / she comes online for the next time.

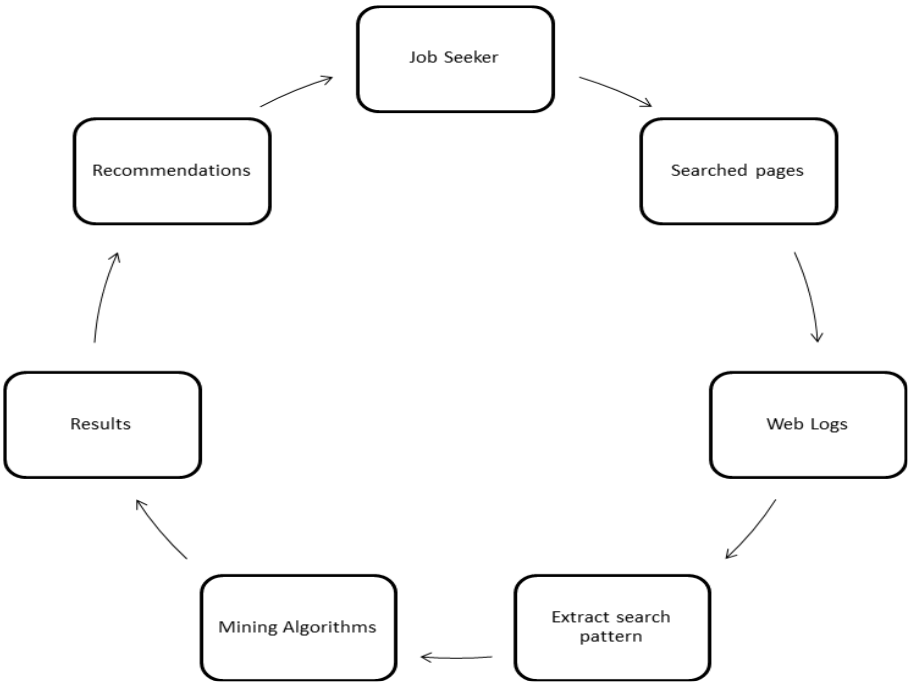


Fig. 1.1: Proposed System

It works as follows:

1. The user / job seeker registers with the system. The user-provided information is stored in the User Database.
2. After registration, the user logs in by providing the correct email-id and password. Then he views various links and all these links are stored in the web log file.
3. The system manages the users and the web log file.
4. The system performs data pre-processing, pattern discovery and pattern analysis on the data in the web log file.
5. Mining algorithms are applied to the data stored in the web log file.
6. Based on the above processes, the user will be recommended a list of webpages according to his behavior or search pattern. The recommendation list will consist of webpages based on users' historical pattern and that as compared to other users' pattern.

LONGEST COMMON SUBSEQUENCE ALGORITHM:

1. Capture the browsed pages for a user dynamically.
2. System compares browsed pages of a user with patterns of same user and other users in the knowledge base.
3. Check for the longest pattern in the knowledge base.
4. Compare both the sequences; Longest Common Subsequence (LCS) is obtained.
5. Consider the pages which are not present in the subsequence, these pages are the intuition pages for the user as they are visited by the user most frequently.
6. Recommendation list is given in the form of URI (Content) of the pages. Hence, recommendations/intuition list as compared to a user's historical pattern are captured.

1.5 Scope of the project

This project involves developing a system which benefits the job seeker. The scope of the project is to perform Web Usage Mining [5] process, specifically:

1. Technical Constraints:

A. Lack of Data

The system needs a lot of data from a single user to effectively make recommendations. Hence it should have a large number of user entries. The more items and user data a recommender system has that it can work with, the stronger is the chance of getting optimized recommendations.

B. Changing User Preferences/ Spread Out Data

Search trends are always changing. While a user may have a particular browsing intention today, his/her preference may change tomorrow. For example, today – java related/ IT company application development field. Tomorrow – management related/IT company management field.

2. End-User Constraints:

A. Visit a predefined minimum number of pages

- Minimum number of searched/ visited pages should be at least 8-10 to optimize the recommendations.
- Based on these number of jobs that the user searches and applies to, the recommendations will be given accordingly in a chronological order.

The scope of this project also involves collaborative filtering which is used to rate a particular searched instance and depending upon multiple users' searches and ratings, the ratings for the same searched instances can be predicted for another user.

Chapter 2

Review of Literature

Recently, several Web Usage Mining systems have been proposed to predicting user navigation behavior and their preferences. Here we review some of the most significant WUM systems and architecture that can be compared with our system.

Analog [1] is one of the first WUM systems. It is structured according to an offline and an online component. The off-line component build session clusters by analyzing past user activity recorded in server log files. Then the online component builds active user sessions which are then classified according to generated model. The classification allows to identify pages related to the ones in the active session and to return the requested page with a list of suggestions. This approach has several limitations, related to scalability. Nevertheless, architectural solution introduced was maintained in several other more projects.

In Mobasher et al [2] present Web personalizer, a system which provides dynamic recommendations, as a list of hypertext links, to users. The analysis is based on anonymous usage data combined with the structure formed by hyperlinks of the site. Data mining techniques (i.e. clustering, sequence pattern discovery and association rules) are used in preprocessing phase in order to obtain aggregate usage profiles. In this phase Web server logs are converted into clusters of visited pages, and cluster made up of set of pages with common usage characteristics. The online phase considers active user session in order to find matches among user's activities and discovered usage profiles. Matching entries are used to compute a set of recommendations which will be inserted into last requested page as list of hypertext links. Web Personalizer is a good example of two tier architecture for Personalization Systems. Baraglia and Palmerini proposed a WUM system called SUGGEST, that provide useful information to make easier the web user navigation and to optimize the web server performance [3]. SUGGEST adopts a two level architecture composed of offline creation of historical knowledge and online engine that understands user's behavior. As the request arrives at this system module it incrementally updates a graph representation of web site based on the active user sessions and classifies the active session using a graph partitioning algorithm.

In this project we use Web Usage Mining (WUM) for extracting knowledge from Web users' access data, by exploiting Data Mining technologies. It can be used for different purposes such as personalization, system improvement and site modification. A typical application of WUM is represented by so called recommender system [4]. The main goal of this recommender system is to improve Web site usability. Typically, the Web usage mining prediction process is structured according to two components performed online and off-line with respect to Web server activity. Offline component builds the knowledge base by analyzing historical data, such as server access log file or web logs which are captured from the server, then these web logs are used in the online component for capturing the intuition list of the user so as to recommend page views to the user whenever he / she comes online for the next time.

We present architecture for capturing recommendations in the form of intuition list for user. Intuition List consists of list of pages visited by user as well as list of pages visited by other user of having similar usage profile. The practical implementation of proposed architecture and algorithms shows that accuracy of user intuition capturing improves up to 85 percent for Live Session Window size of two, if numbers of page views having maximum weights are more in the navigation patterns of the user.

Chapter 3

System Analysis

3.1 Functional Requirements

A functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behavior, and outputs.

The system will include the following functional requirements:

- Authentication: The user should be authenticated to the system by providing the correct login credentials.
- Authorization: After authentication, the user is authorized to view other users' profiles but cannot alter and modify their data.
- Session tracking: The system keeps a track of the users' data i.e. ip address, log times and URLs.
- Historical data: The system performs data pre-processing, pattern discovery and pattern analysis on the tracked data to generate relevant information that is used by the recommender system.
- Recommendations: Based on this relevant information the user is provided with an intuition list .

3.2 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

The system will include the following non-functional requirements:

- Performance: The system should perform well by keeping in mind the response time and throughput.

- Accuracy: The results provide accurate user intuition.
- Capacity: The database should be able to hold multiple users' data.
- Availability: The users should be able to access the system whenever required.
- Reliability: In case the system is down for a while, its functionality will be replaced by the backup system.
- Security: The system should provide Confidentiality, Integrity and Authentication.

3.3 Specific Requirements

- **Software Requirements**

The software requirement specifications for developing this project are as follows:

1. Visual studio 2010
2. Microsoft SQL Server Version 2005/ My SQL
3. PHP/ASP/Java/Vb.Net/HTML/CSS
4. File Servers/ Xampp server
5. Windows OS

- **Hardware requirements:**

The minimum hardware requirement specifications for developing this project are as follows:

1. 256MB RAM
2. 80GB HDD
3. Intel 2.10 GHz Core i3 Processor

3.4 Use-Case diagram

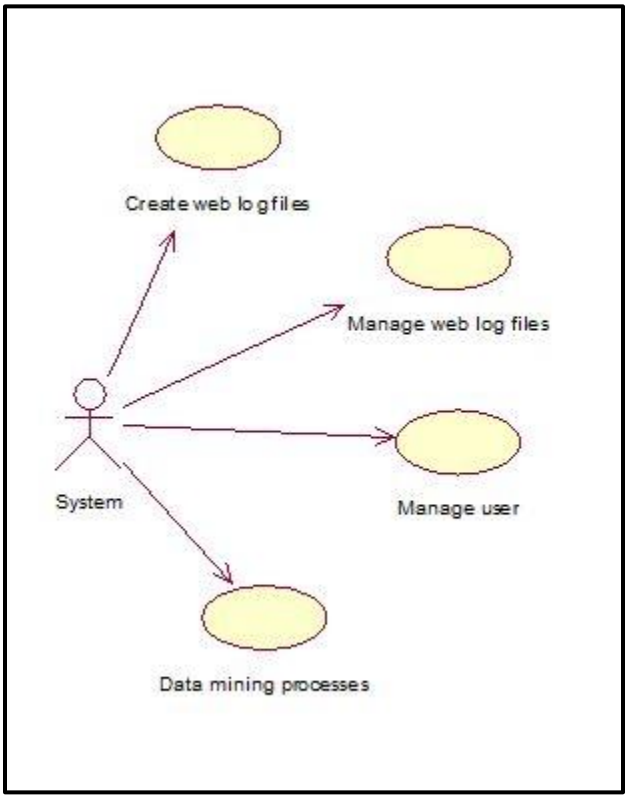


Fig. 3.4.1 Use-Case Diagram for system

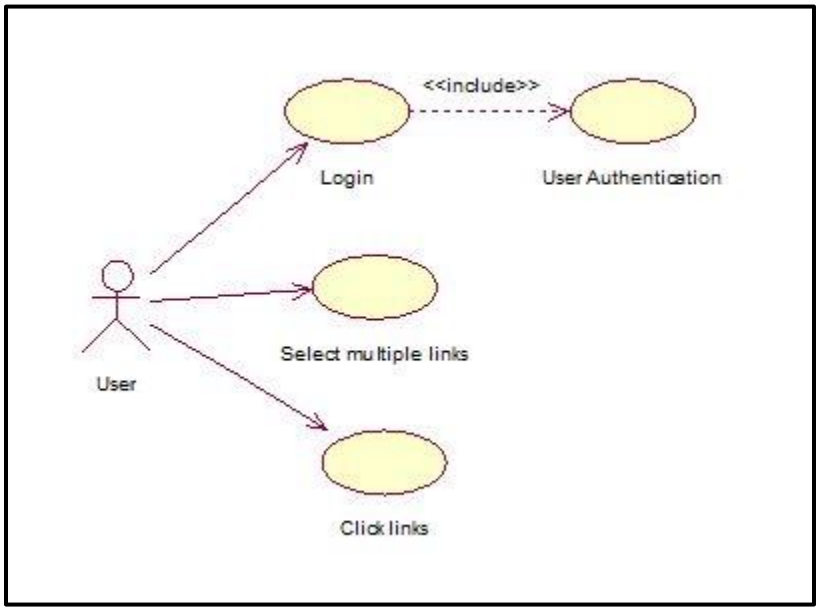


Fig. 3.4.2 Use-Case Diagram for user

Table 3.1 Use-Case Description

Use Case	Active Participation	Description
Create Account	User	For using the service user will first create an account
Login	User	To access the service user will logged in to the portal using user_id and password
Search Jobs	User	After successful login user will search for jobs links
Add User	Admin	Admin will verify the user and add that user to user database
Update User Details	Admin	The admin can add or delete or modify the data of user from the user database
Update Web log files	Admin	The admin can add or delete Web log files in database

Chapter 4

Analysis Modeling

4.1 Activity Diagram for Personalized Online Recruiting System

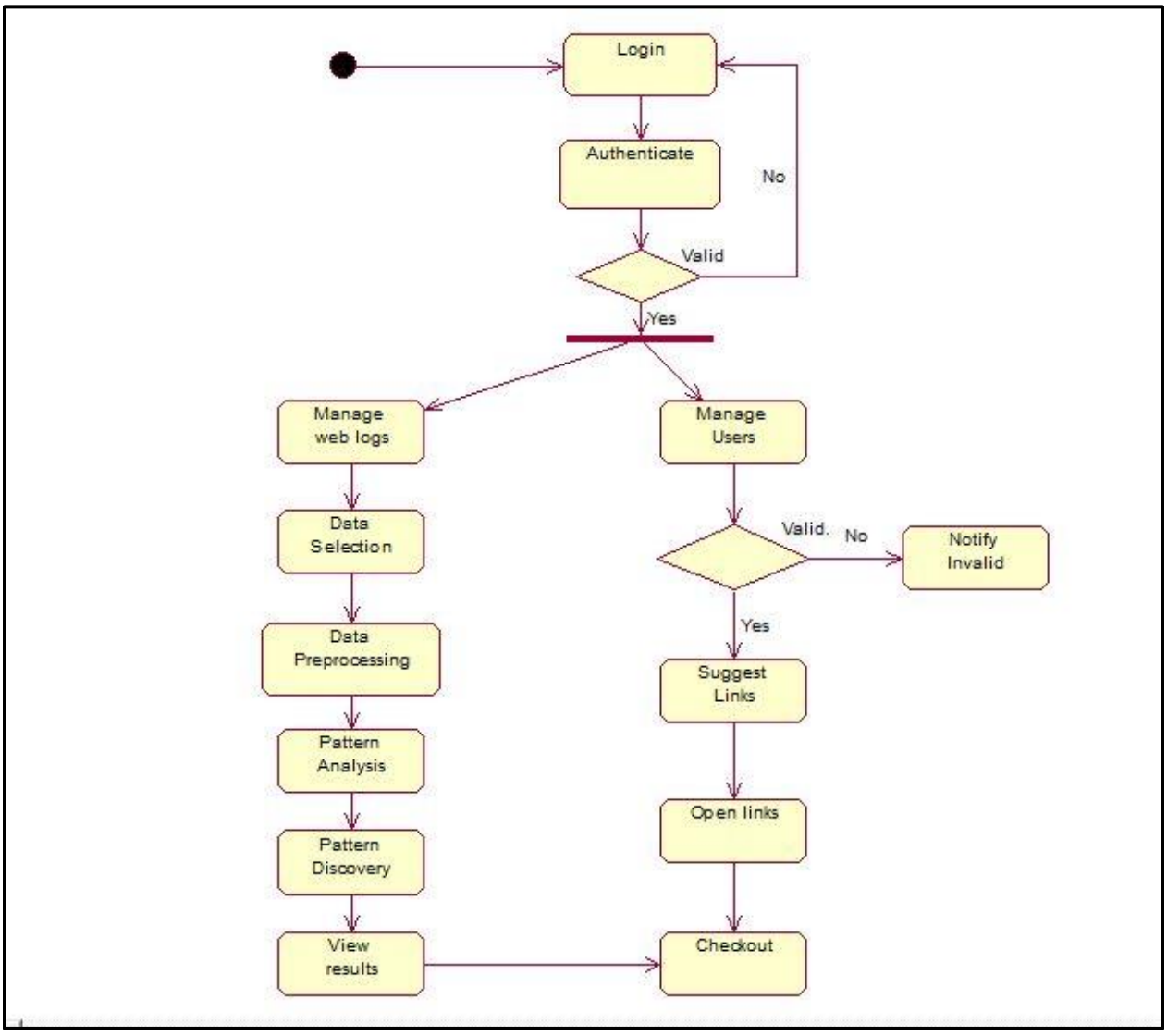


Fig. 4.1 Activity diagram for admin

The flow of the activity diagram is as follows:

1. The system authenticates the user .
2. The system manages the user and and also the data in the database.
3. System performs the data mining process like data selection, data preprocessing, data analysis and provide the result to user.
4. The user view the results
5. If the user chooses to exit, the activity comes to an end.

Context Level DFD

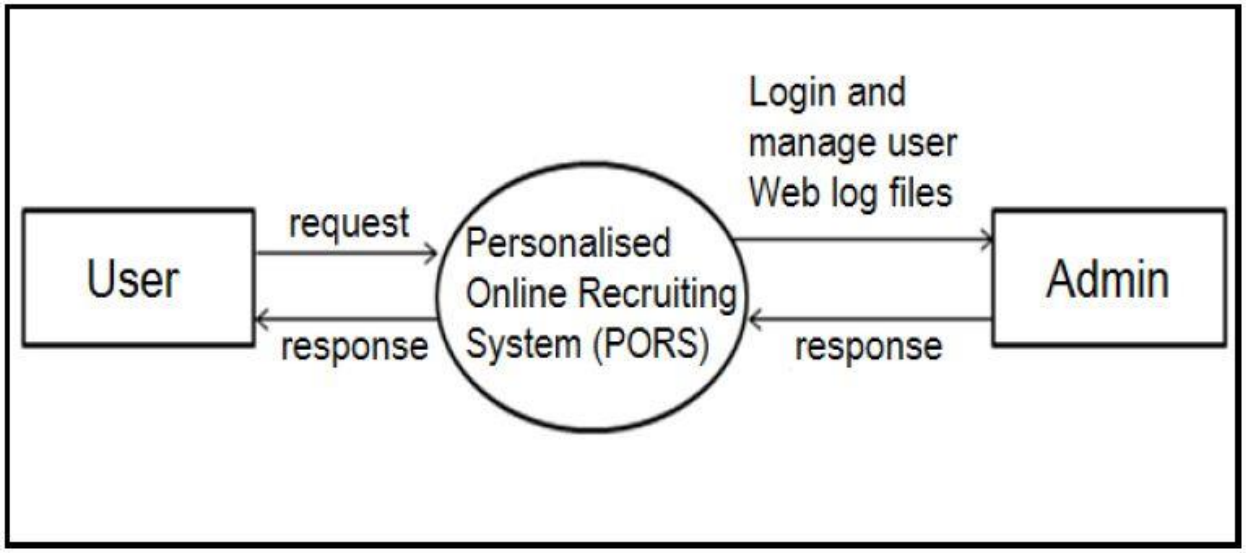


Fig. 4.2.1 DFD Level 0

1. User registers into the job portal. The user data is entered into the database.
2. The user search for various job of his choice.
3. The system manages data in database.
4. The system is responsible for updating the database regularly.

Level 1 DFD

For portal

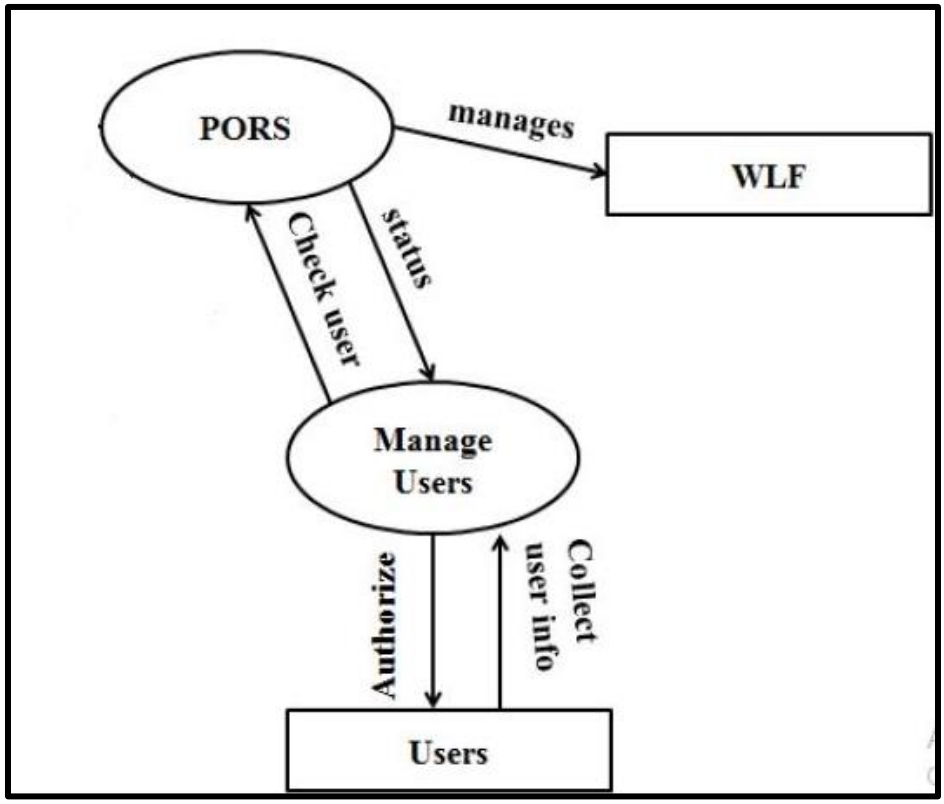


Fig. 4.2.2 Level 1 DFD for portal

The data flow for Level 1 DFD for Portal is:

1. The portal verify whether the registered user is a valid user or not.
2. The portal manages the registered user.
3. System manages the data in the database.

For User

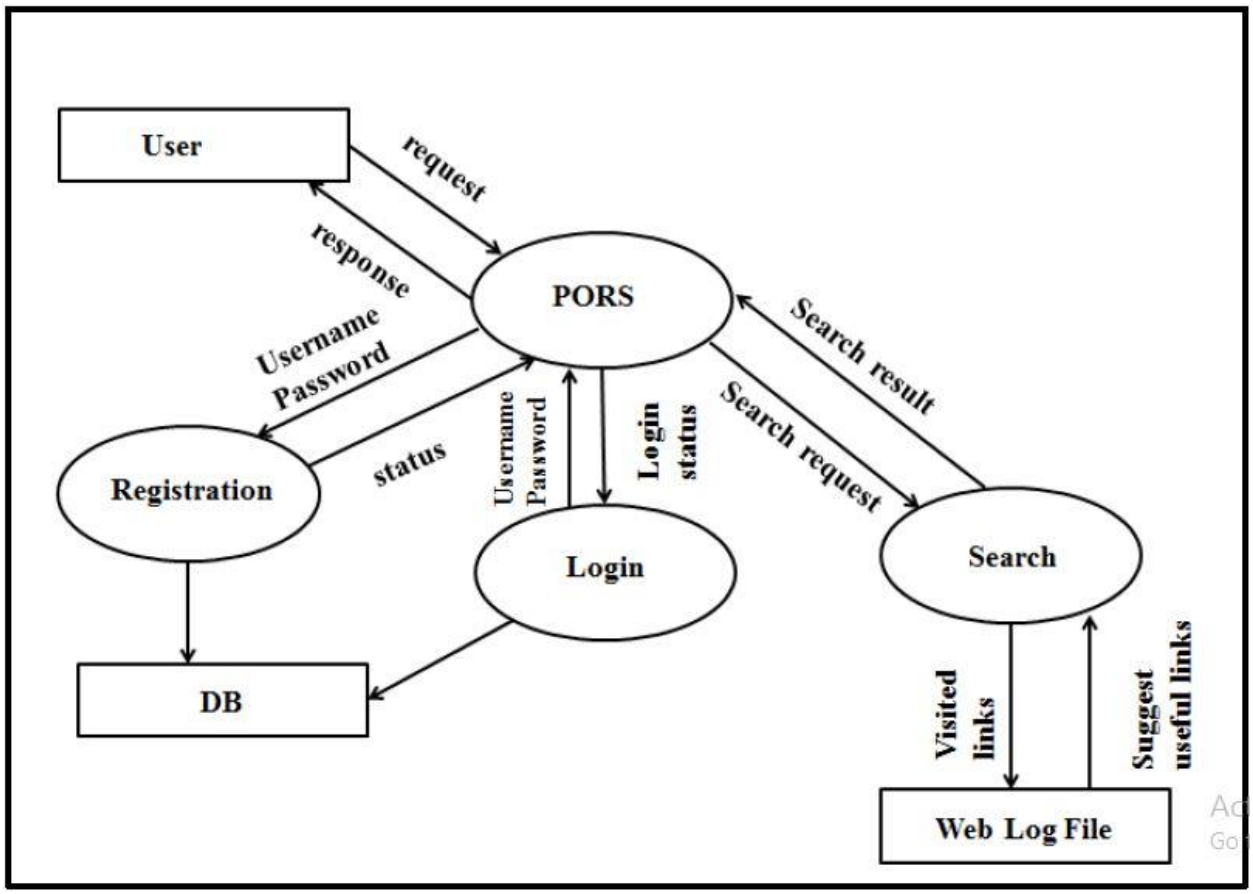


Fig. 4.2.3 Level 1 DFD for user

The data flow for Level 1 DFD for System is:

1. User register into the portal with a username and password.
2. Registered user logs into the portal through valid username and password.
3. Once user logs into the system, user searches for jobs.
4. User views various links and also gets recommendation for other links.
5. Users search data is stored as web log files in database.

Level 2 DFD (Portal)

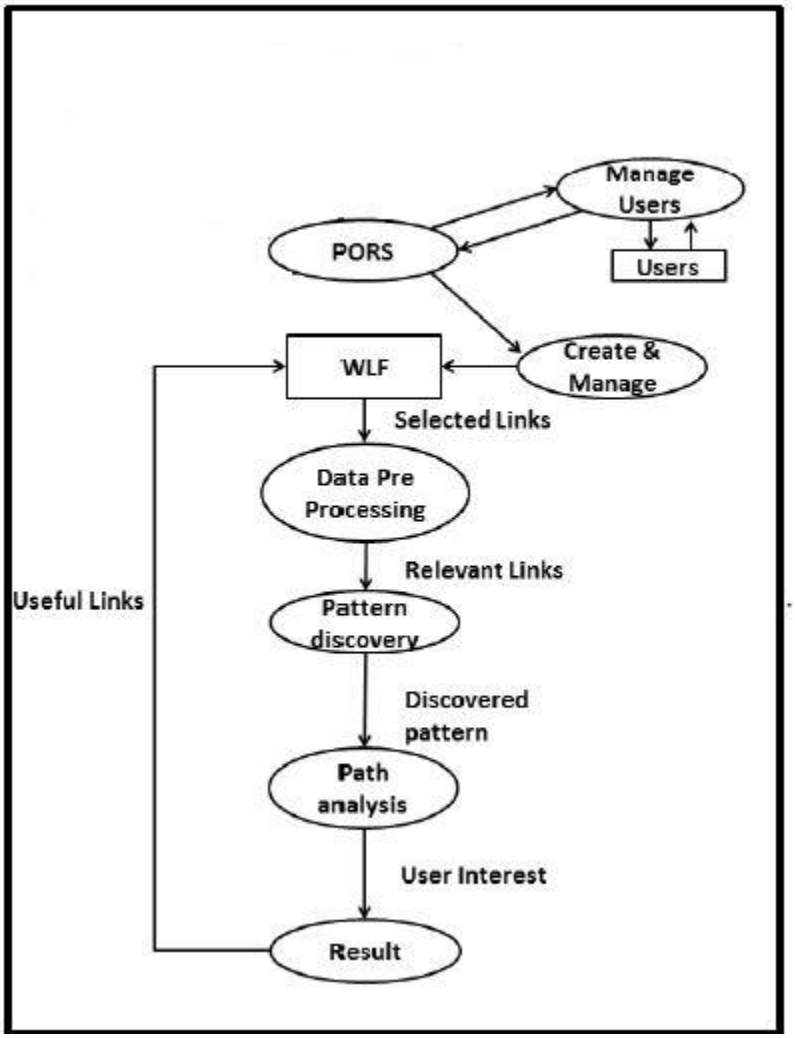


Fig. 4.2.4 DFD Level 2 for portal

The flow is:

1. Many user searches for links and users search data is stored in web log files.
2. Web log files are pre-processed to find relevant links.
3. Using pattern discovery mining, frequent patterns of links are discovered.
4. By path analysis, useful links are recommended to users.

Level 2 DFD (User)

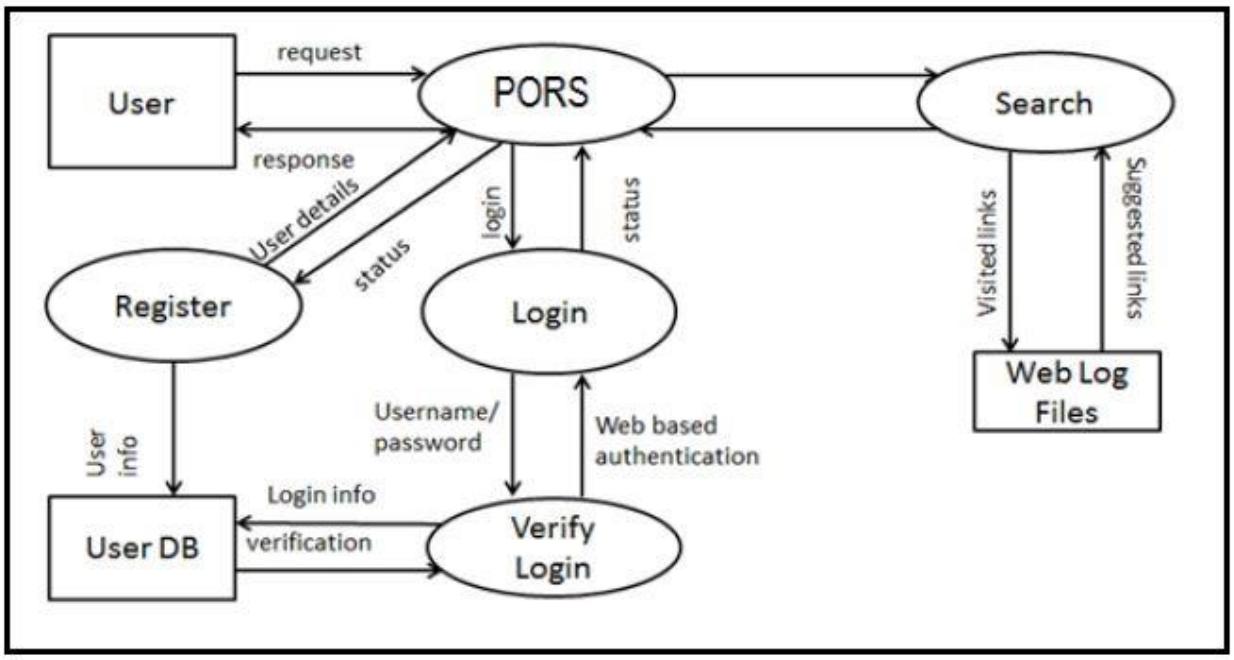


Fig. 4.4.5 Level 2 DFD (User)

The flow is:

1. User registers with the portal with a username and password.
2. User logs into the portal through valid username and password.
3. User search for job links.
4. Users search data is stored in web log data.
5. Web log is processed and useful links are recommended to user.

4.3 Sequence Diagram

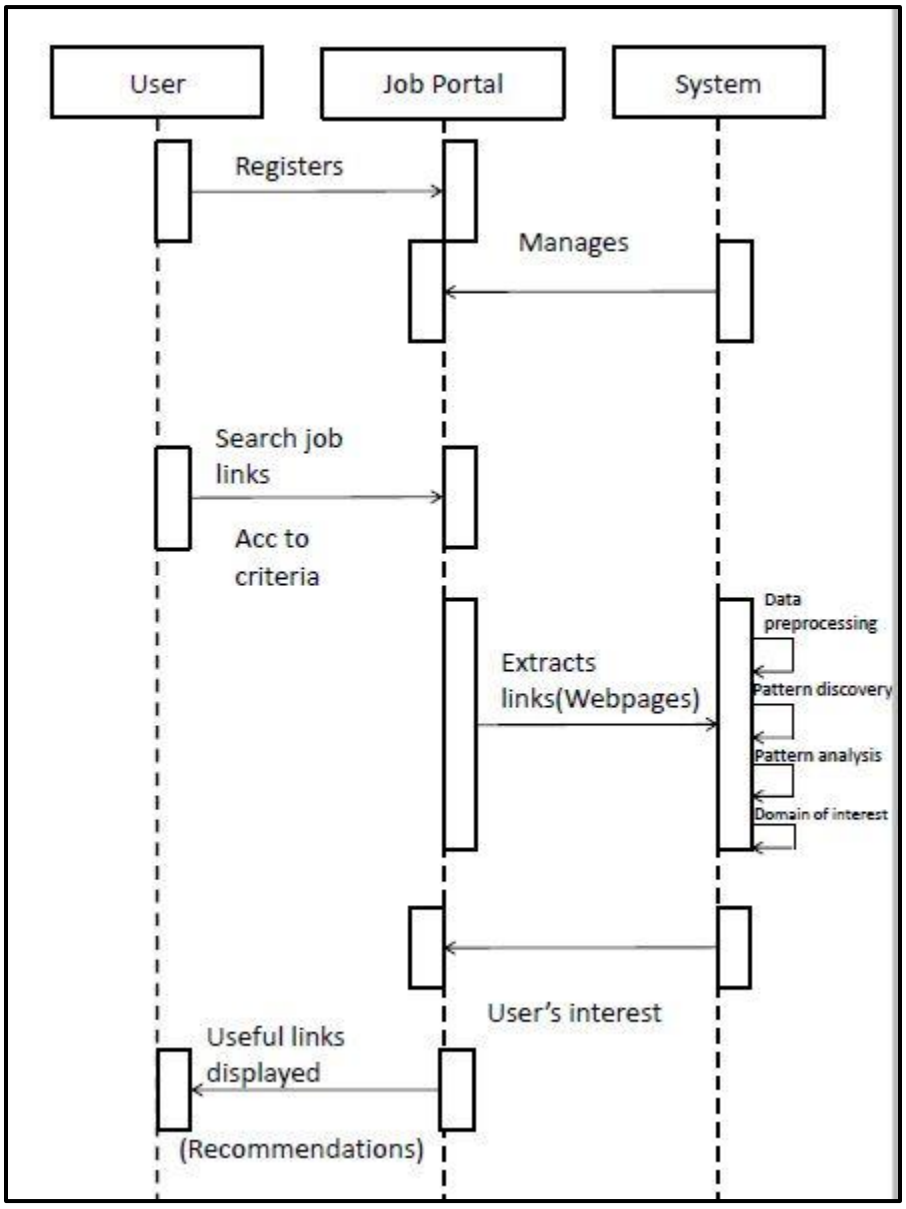


Fig. 4.3 Sequence Diagram

4.4 Class Diagram

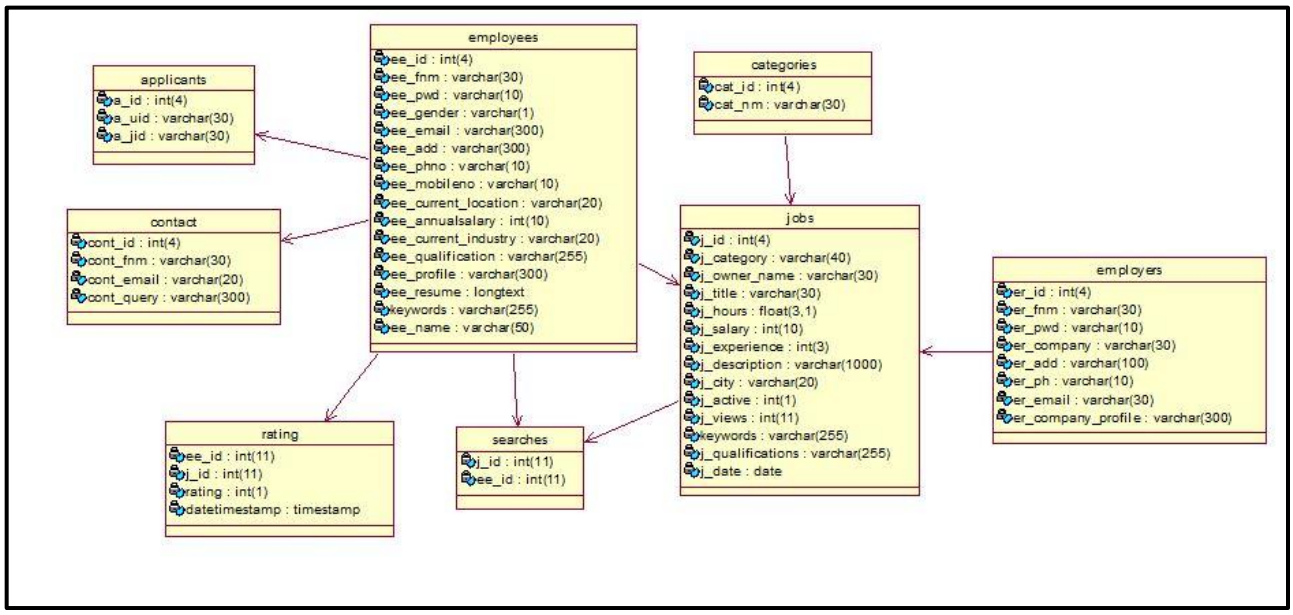


Fig. 4.4: Class diagram

4.5 Timeline Chart

Task Name	Start Date	End Date	Duration
Basic Requirement Study	07/13/15	07/27/15	11
Existing System Study	07/31/15	08/07/15	6
Planning Phase	08/10/15	08/31/15	16
Define the Problem	09/05/15	09/11/15	6
Study of Workflow	09/14/15	09/28/15	11
Write System Requirments	09/30/15	10/03/15	6
Analysis Phase	10/05/15	10/14/15	11
Data process modelIng	10/15/15	11/03/15	14
Design phase	11/07/15	11/26/15	15
System design(front end)	12/03/15	12/31/15	21
Program design(back end)	01/01/16	01/19/16	13
Program Coding and Unit Testing	01/26/16	02/29/16	25
System Testing	03/01/16	03/04/16	4
Implementation	03/08/16	03/11/16	4

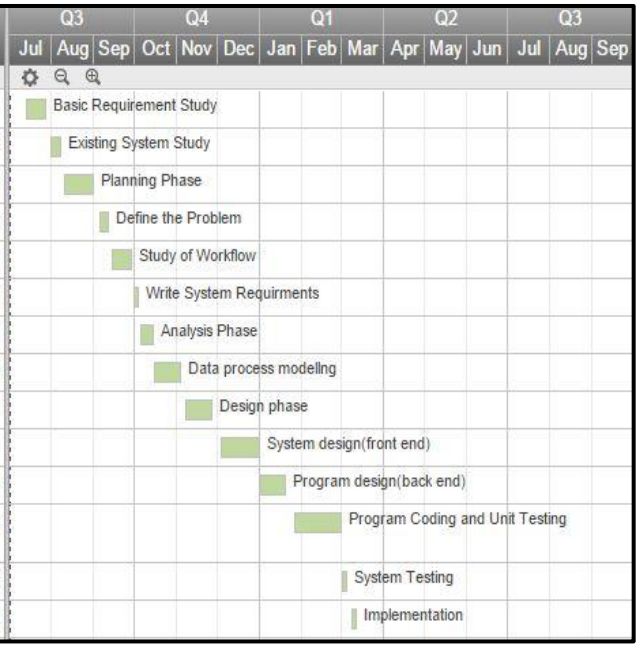


Fig. 4.4: Timeline chart

Chapter 5

Design

5.1 Architectural design of the system

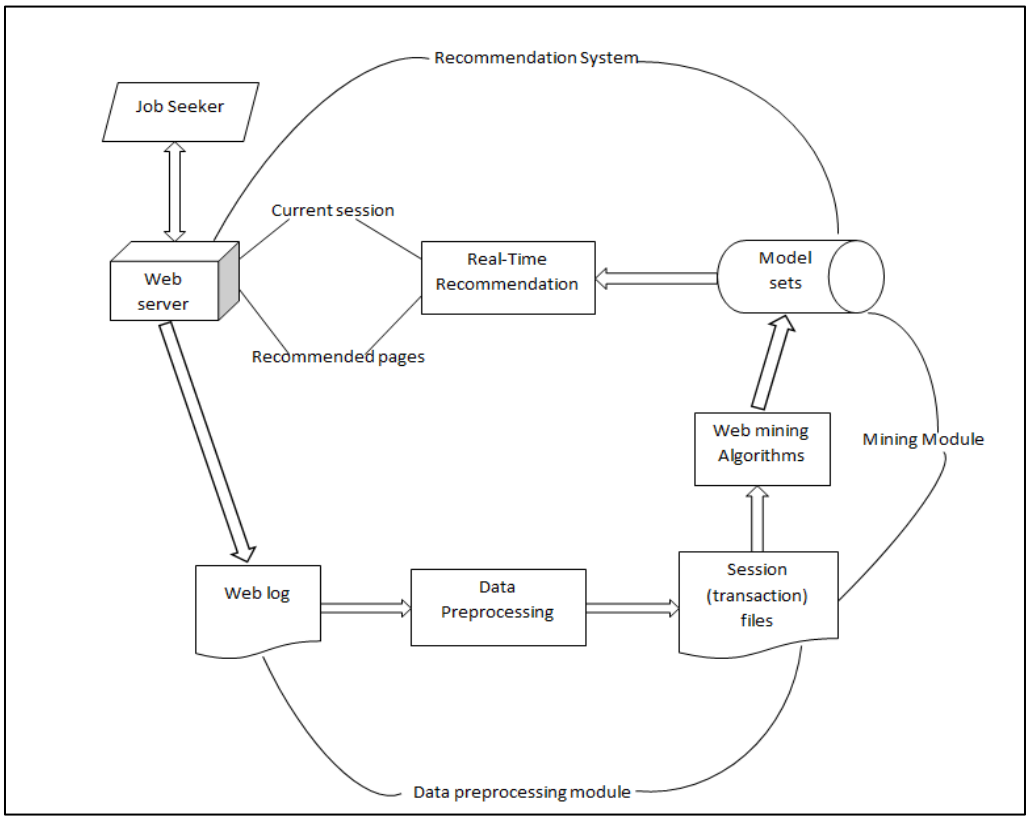


Fig 5.1 Architectural Design

The architectural design consists of three modules that further consist of various processes. The job seeker enters his information while registering with the website. This information is stored in the users' database. Once he logs in, he searches for various links. These links are then stored by the web server in the web log file. The first module is the Data Pre-Processing module that performs data preprocessing on the data in the web log file. This processed relevant data is then called the session file. The second module is the Mining module that includes the longest common subsequence algorithm that will be applied on the session files to get the model sets. The third and the last module is the recommendation module/system that provides real time recommendations to the currently active users.

5.2 User Interface Design

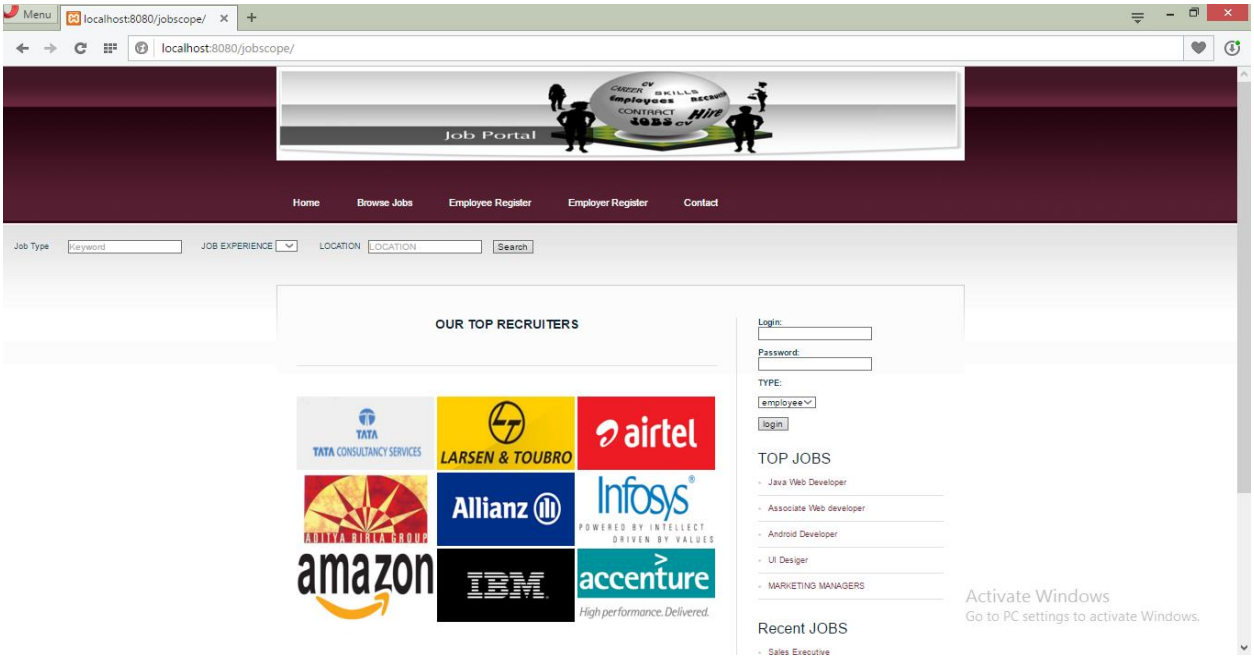


Fig. 5.2.1 Home page

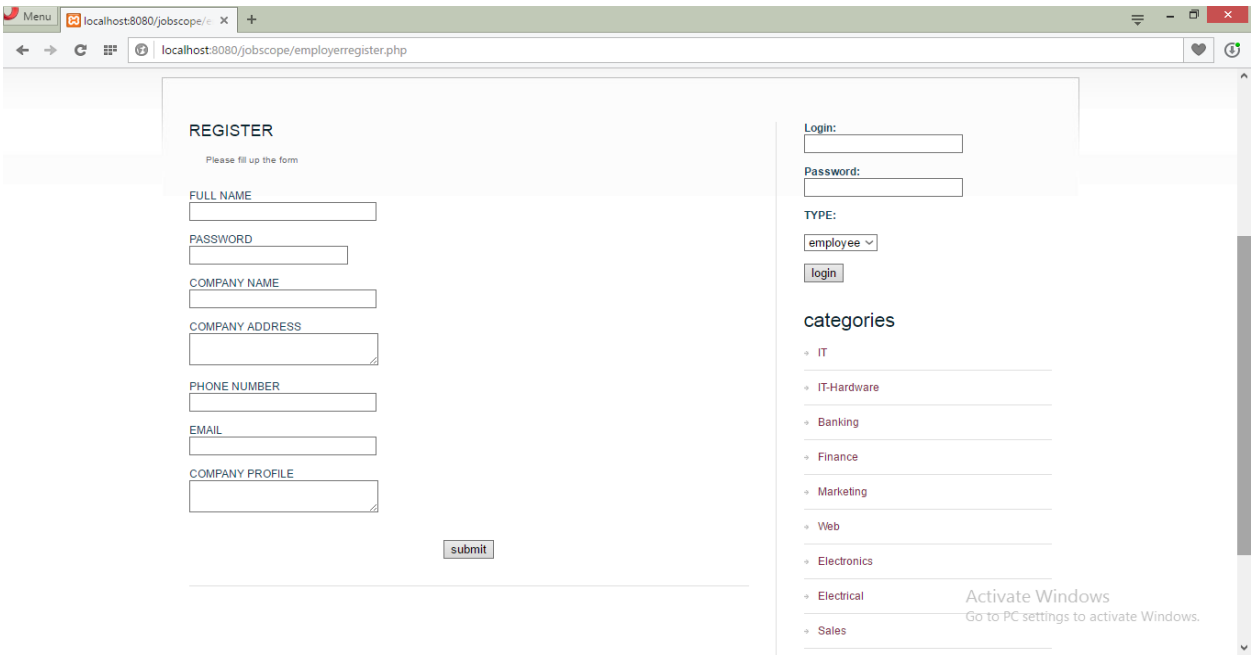


Fig 5.2.2 Employee Registration page

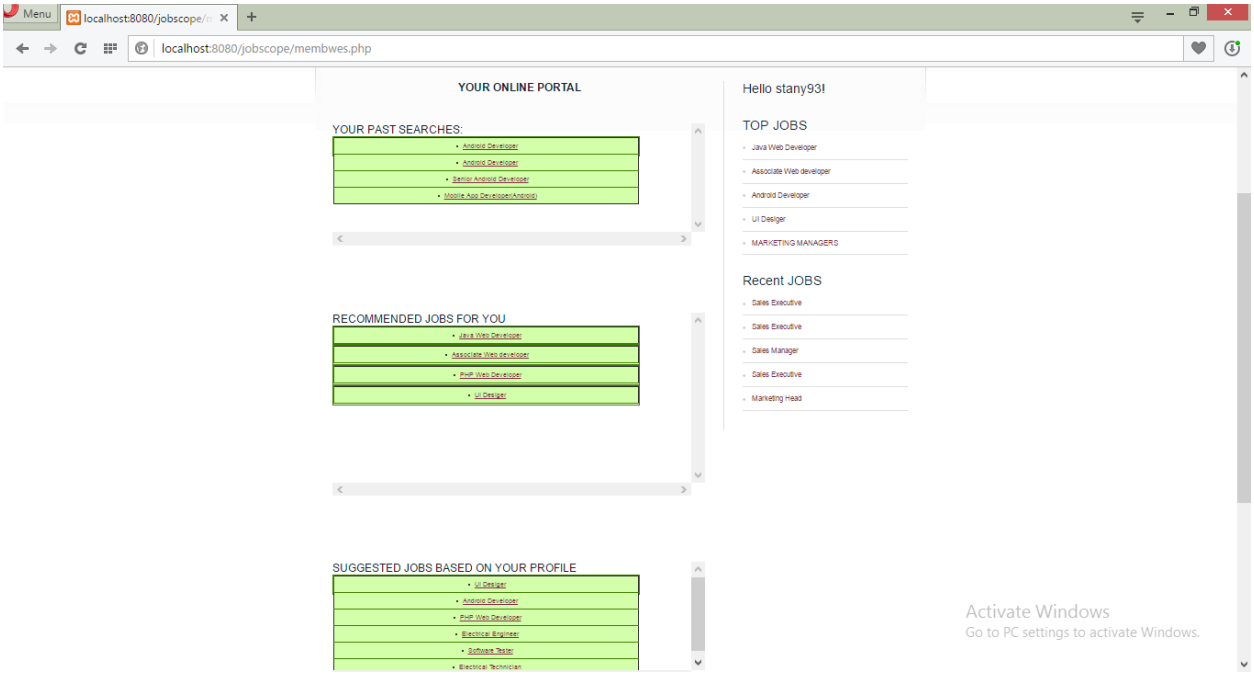


Fig. 5.2.3: Employee job portal page

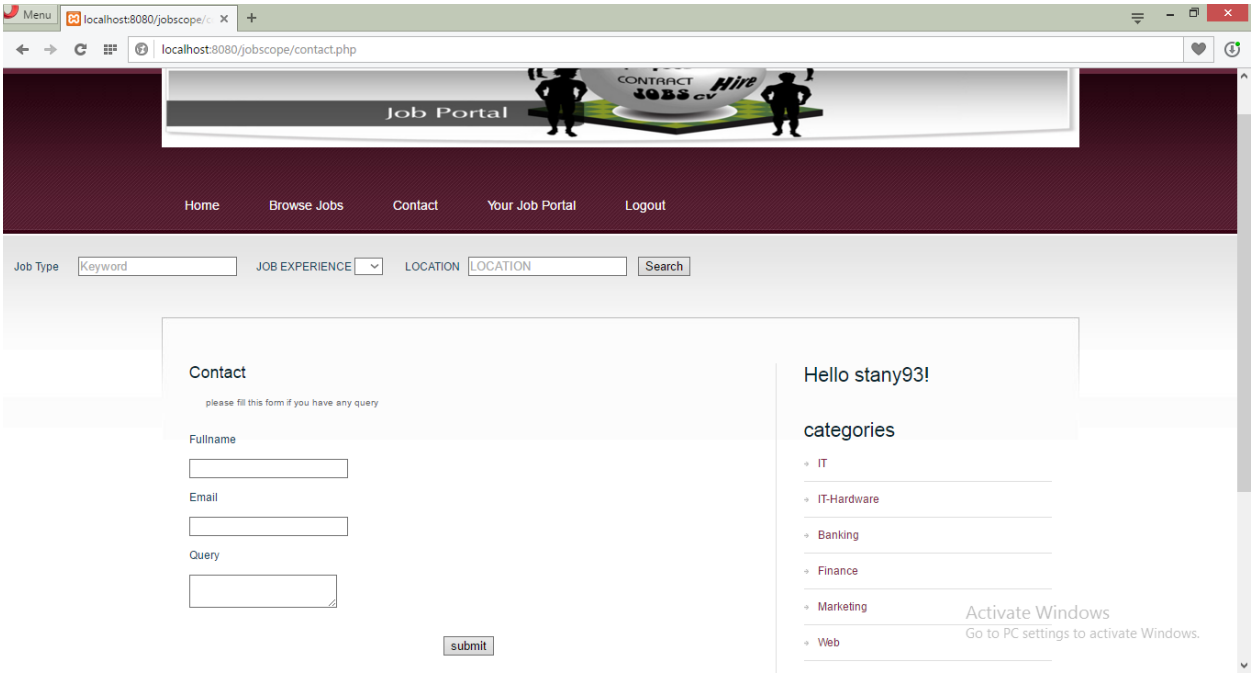


Fig. 5.2.4: Contact us

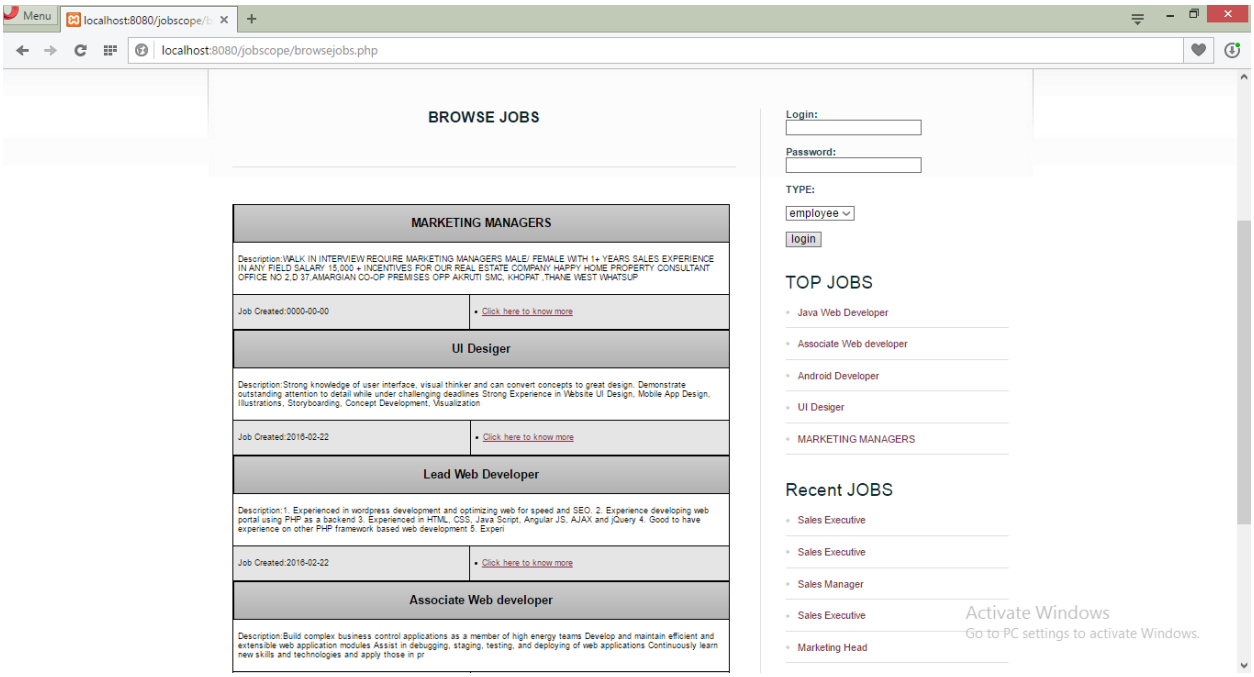


Fig. 5.2.4: Browse job page

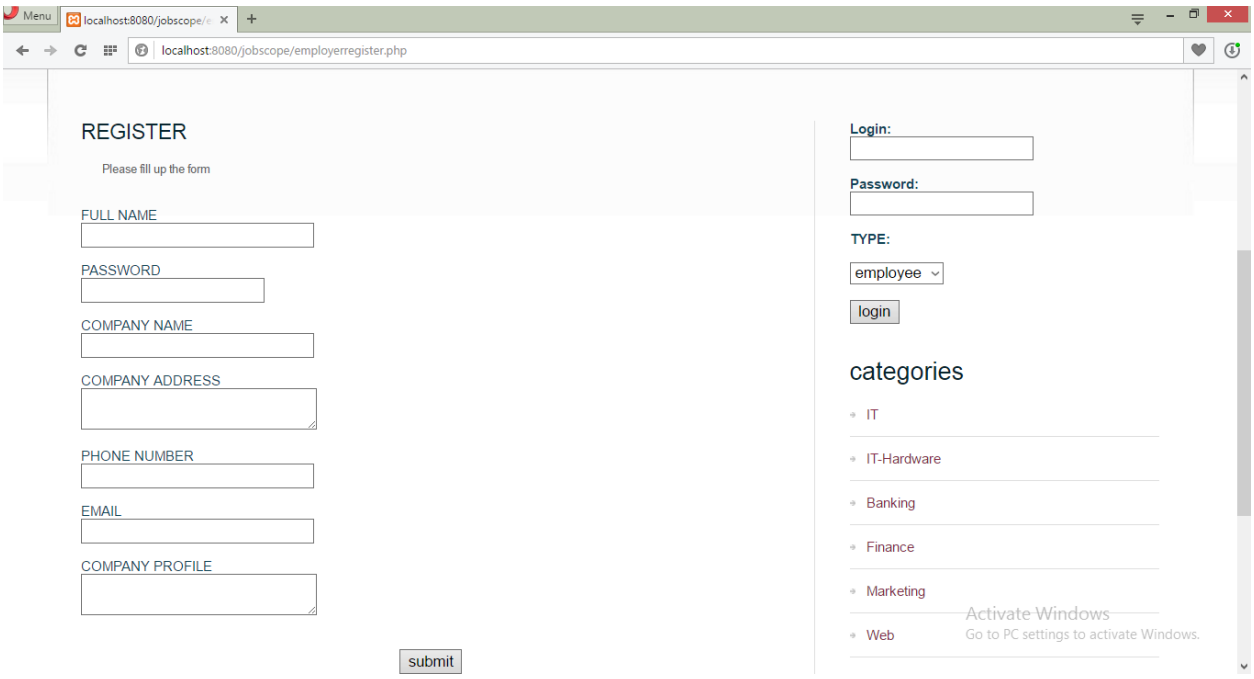


Fig. 5.2.5: Employer registration page

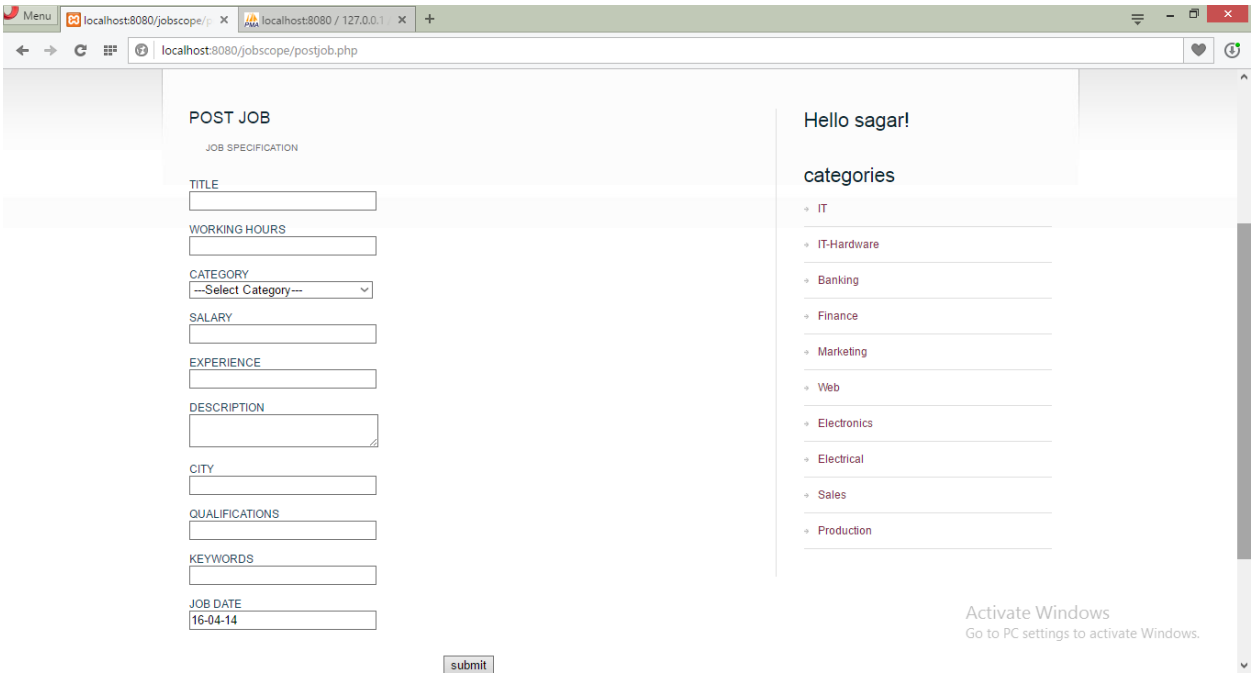


Fig. 5.2.6: Employer Post job page

Chapter 6

Implementation

6.1 Algorithm / Method used:

The longest common subsequence (LCS) problem is the problem of finding the longest subsequence common to all sequences in a set of sequences (often just two sequences). The longest common subsequence problem is a classic computer science problem, the basis of data comparison programs and is useful in recommender systems to compare between two or more users' searches and preferences.

Working of LCS Algorithm:
A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. Longest common subsequence (*LCS*) of 2 sequences is a subsequence, with maximal length, which is common to both the sequences.
In case multiple solutions exist, print any of them. It is guaranteed that at least one non-empty common subsequence will exist.

LONGEST COMMON SUBSEQUENCE ALGORITHM:

1. Capture the browsed pages for a user dynamically.
2. System compares browsed pages of a user with patterns of same user and other users in the knowledge base.
3. Check for the longest pattern in the knowledge base.
4. Compare both the sequences; Longest Common Subsequence (LCS) is obtained.
5. Consider the pages which are not present in the subsequence, these pages are the intuition pages for the user as they are visited by the user most frequently.
6. Recommendation list is given in the form of URI (Content) of the pages. Hence, recommendations/intuition list as compared to a user's historical pattern are captured.

Collaborative filtering also referred to as social filtering, filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. Collaborative filtering algorithms work by searching a large group of users or items and finding a smaller list from it with tastes similar to yours. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see.

Ram12 logs into the system. He performs searches either by entering the keyword in the job type bar or chooses from the top jobs at the extreme right of the page. Ram12's past and recent searches will be displayed to him. He will also be recommended jobs based on another user's similar search.

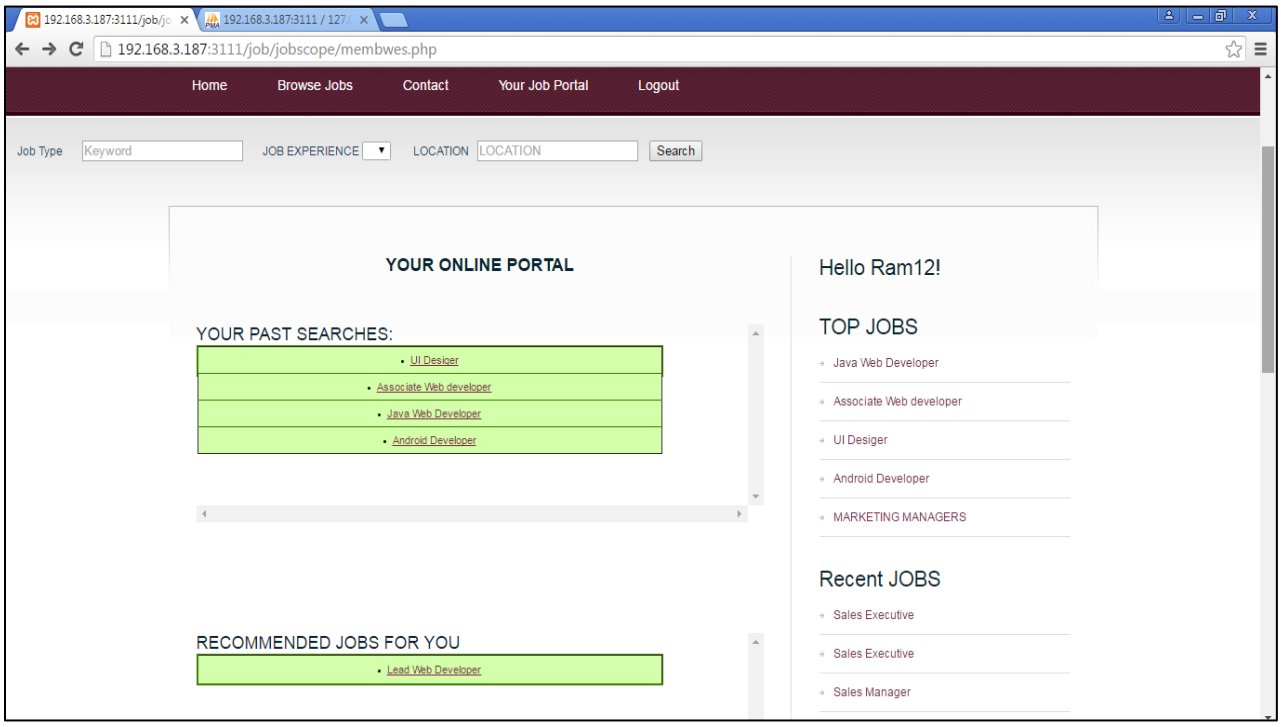


Fig. 6.1.1: Search & Recommendation section

Based on the ratings given by the other user to a particular job, and comparing it with Ram12's rating, Ram will be given recommendations based on the collaborative filtering algorithm. If Ram12 and the other user's rating are both above 3, the job will be recommended to the both of them.

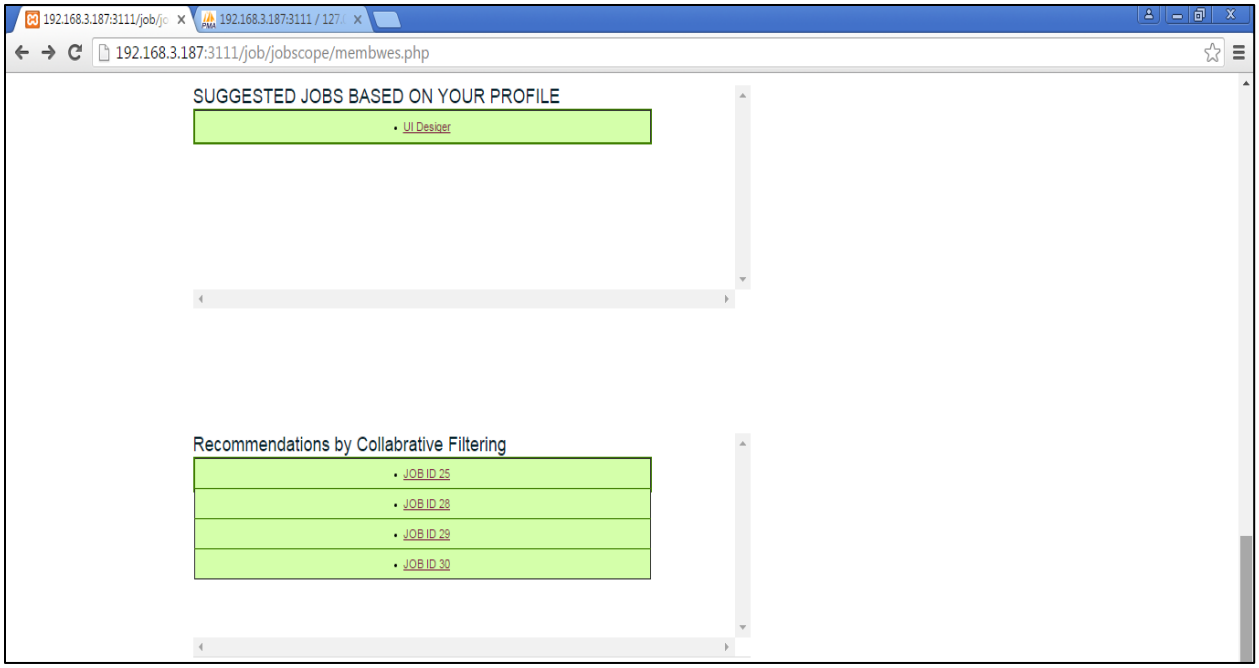


Fig. 6.1.2: Suggestion based on profile and collaborative filtering

A new user Beena registers with the system by providing the appropriate details on the register page as she is a new user. She then logs into the system by providing her credentials on the login page. When she checks her online job portal, she will receive a message on the screen saying 'You do not have any past searched history' as she hasn't previously visited any pages. Also she will not be provided any recommendations based on other users' past searches. This highlights the constraint that the user has to visit a predefined minimum number of pages after registration to receive accurate recommendations / suggestions.

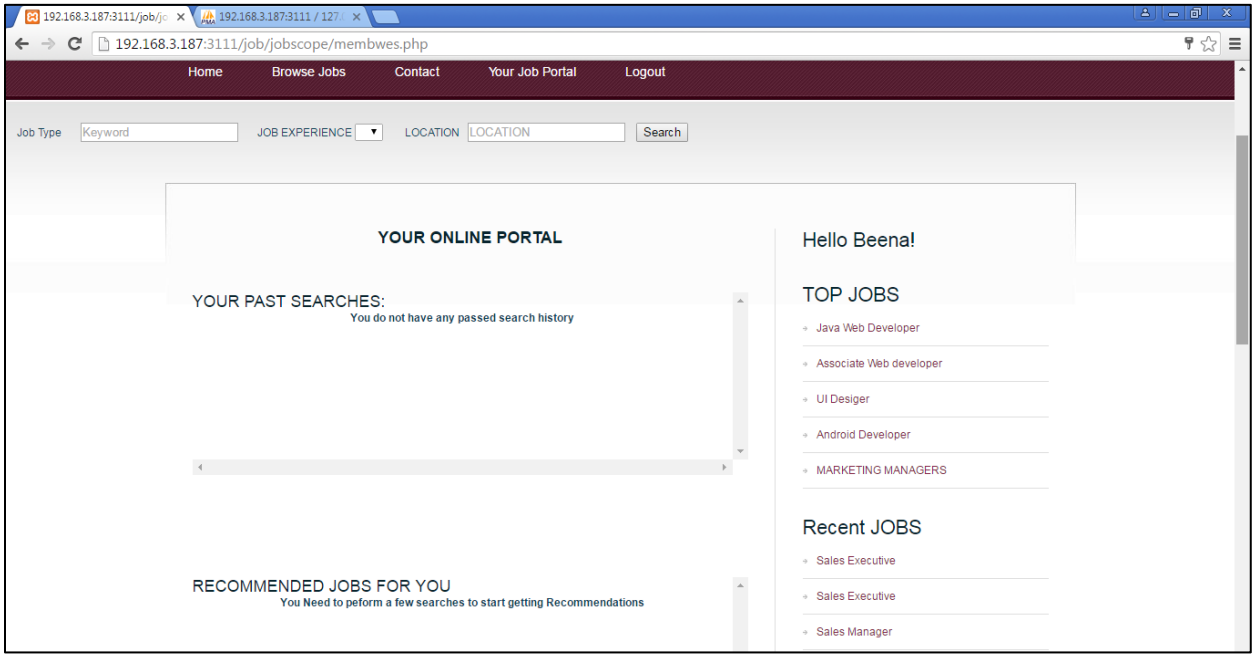


Fig. 6.1.3: Job portal of a user

User Beena searches for jobs on the portal which gets registered with the system.

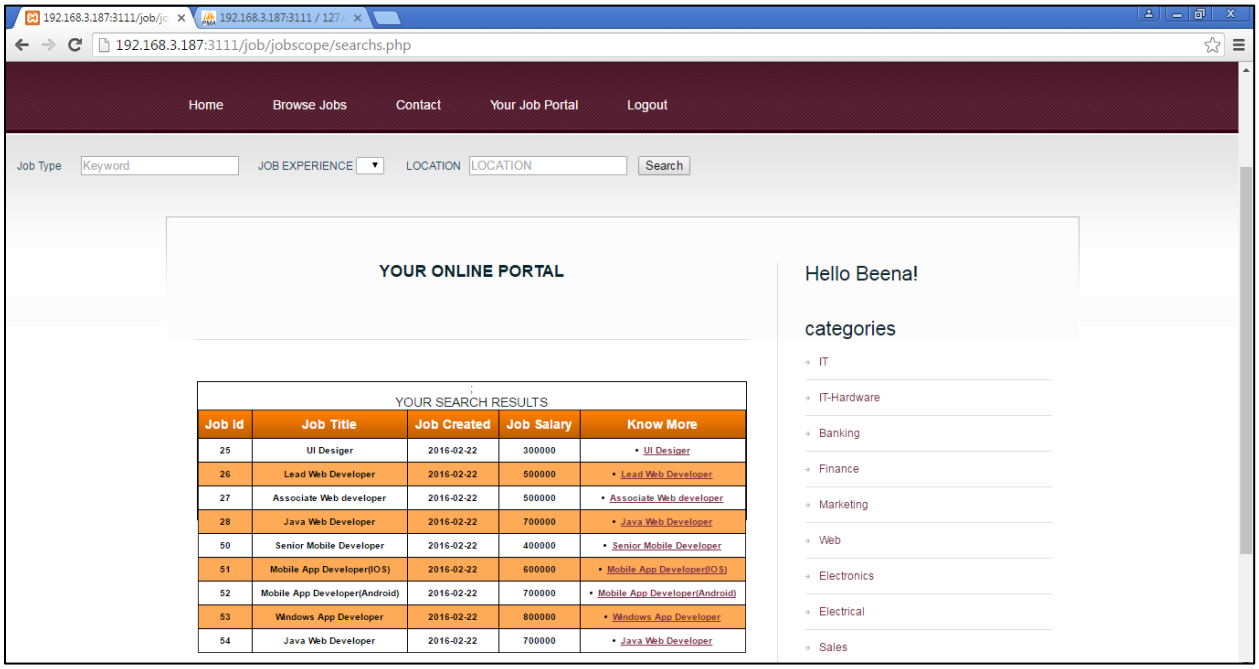


Fig. 6.1.4: Users search query result

Based on Beena's recent searches, she will be recommended those jobs as she has already searched them. She will also be recommended other users' pages Eg: Ram12 – UI Designer, Android developer and other users – Lead Web Developer, PHP Web developer, etc.

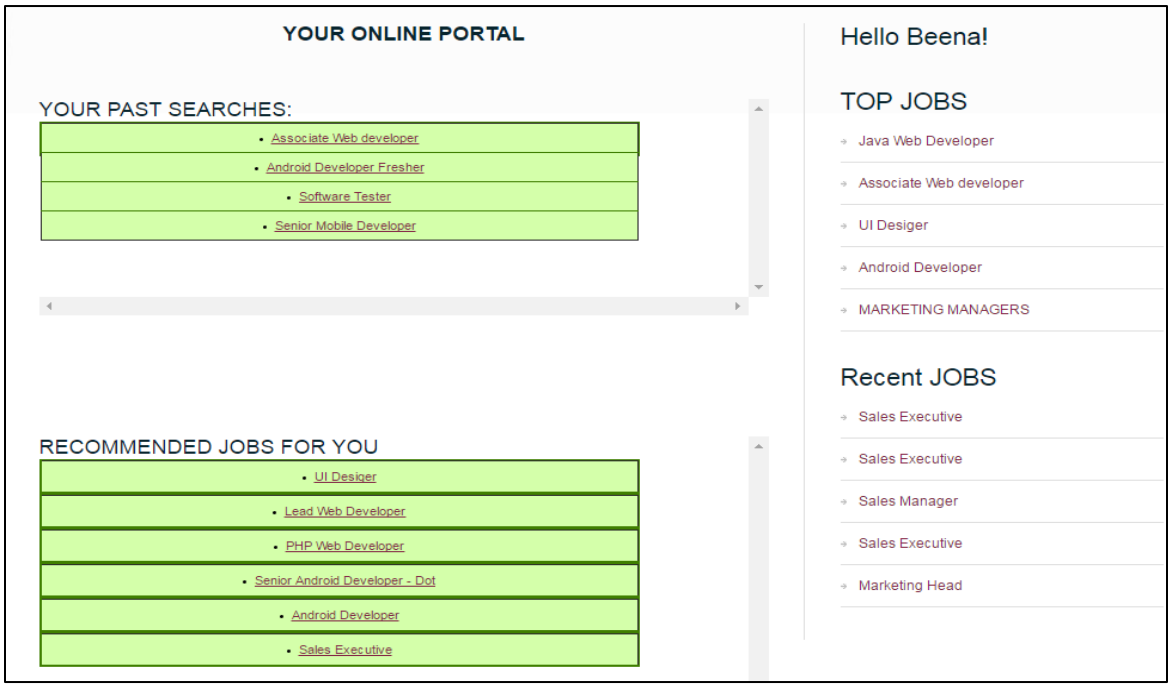


Fig. 6.1.5: Job portal of user

Based on the collaborative filtering algorithm, Beena will also be recommended jobs based on the comparison of ratings that she and another user gives a particular job. If the rating is good, the job will be recommended, else it will not.

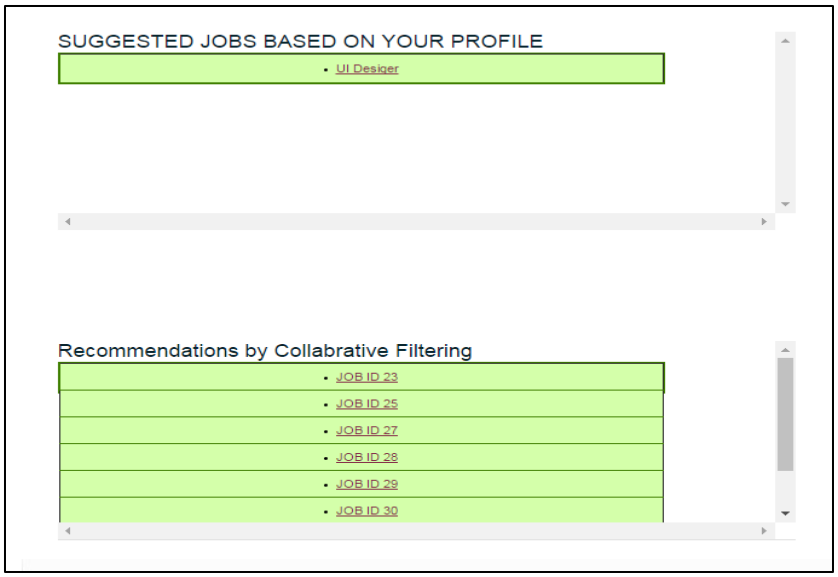


Fig. 6.1.6: job portal of user

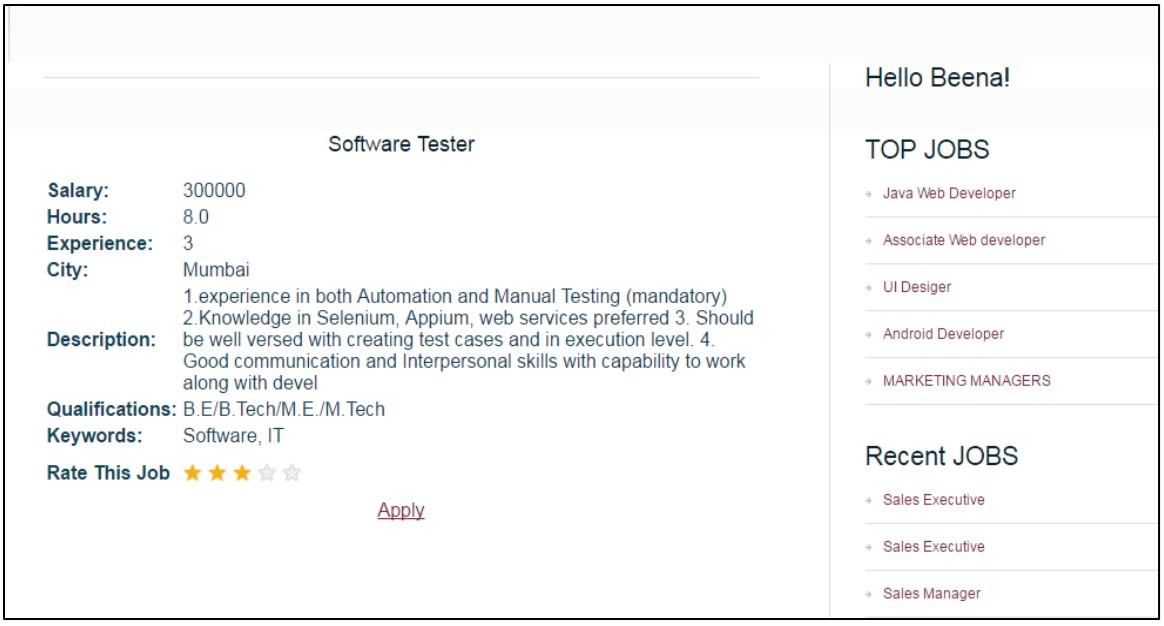


Fig. 6.1.7: Job details

6.2 Working of the Project :
Back-End Code

```
<?php session_start();
if(!isset($_SESSION['employee']))
{
        header('Location: index.php');
        exit;
}

$link=mysql_connect("localhost","jobscope","riddhi")or die("can not connect");
mysql_select_db("jobscope",$link) or die("can not select database");

$q="select * from jobs where j_active=1 order by j_id desc ";
$res=mysql_query($q,$link) or die ("can not select database");
$sempid=$_SESSION['eeid'];
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>

<?php
include("includes/head.inc.php");
?>
</head>
<body>
        <div id="logo">
                <?php
                include("includes/logo.inc.php");
                ?>
        </div>

        <div id="header-wrapper">
        <div id="header">
        <div id="menu">
                        <?php
                        include("includes/menu.inc.php");
                        ?>
        </div>
        <!-- end #menu -->
        <!-- end #search -->
        </div>
        </div>
```

```
<!-- end #header -->
<!-- end #header-wrapper -->

<div id="search">
                <?php
                include("includes/search.inc.php");
                ?>
</div>

<div id="wrapper">
<div id="page">
<div id="page-bgtop">
                <hr />
<!-- end #logo -->

<div id="content">
<div class="post">

        <h2 class="title"><center><b>YOUR ONLINE PORTAL</b></center></a></h2>

<div class="entry">
</div>

<div class="rsearch" style="height:200px;overflow:scroll;max-height:500px;"><h2>
YOUR PAST SEARCHES:</h2>


<?php
$link=mysql_connect("localhost","jobscope","riddhi")or die("can not connect");
mysql_select_db ("jobscope",$link) or die("can not select database");
$q="SELECT j_id,j_title from jobs where j_id IN (Select j_id from searches where ee_id
='$sempid')";
$res=mysql_query($q,$link) or die("cant connect");
if (mysql_num_rows($res) > 0)
        {
        echo"<div class=CSSTableGenerator1 style=width:500px;height:30px>";
        echo"<table border=1>";
        while($row=mysql_fetch_assoc($res))
                {
                echo '<tr > <td><li><a
                href="job_details.php?id='.$row['j_id'].'">'.$row['j_title'].'</a></li>
                </td></tr>';
                }
        echo"</table></div>";
        }
```

```
else
        echo "<center><b>You do not have any past search history</b></center>";
        mysql_close($link);
        ?>

</div>
<hr><br><br><br><br><br><br><br><br>
<div class="rrecommend" style="height:300px;overflow:scroll;max-
height:500px;"><h2>RECOMMENDED JOBS FOR YOU</h2>
<hr>

<?php
$link=mysql_connect("localhost","jobscope","riddhi")or die("can not connect");
mysql_select_db ("jobscope",$link) or die("can not select database");


/**********************************************************************
**********************************************************/


//No of unique employees
$sql = "select COUNT(DISTINCT ee_id) as NOR from searches ";
$sql1 = mysql_query($sql,$link) or die("cant connect");
$data=mysql_fetch_assoc($sql1);
$noofusers = $data['NOR'];

//echo "No of users is $noofusers";
$recom="";


/**********************************************************************
**********************************************************/

//Jobs searched by  current user
$q="SELECT distinct(j_id) from searches where  ee_id ='$sempid'";
$res=mysql_query($q,$link) or die("cant connect");

$userpages=array();   //Pages searched by user
while($row=mysql_fetch_assoc($res))
{
 $userpages[]=$row['j_id'];
}
//echo "Unique pages searched by user";
//print_r($userpages);
```

```
/*************************************************************************
*************************************************************/

//Users from where to compare for recommendation
$allusers = "SELECT DISTINCT(ee_id)  from searches  WHERE ee_id <>'$sempid'";
$allusers1 = mysql_query($allusers,$link) or die("cant connect");
$distinctusers=array();   // All distinct users

 while($row=mysql_fetch_assoc($allusers1))
{
 $distinctusers[]=$row['ee_id'];
}

//echo "Distinct users";
//print_r($distinctusers);

$largestratio=0;
$temporary=array();


/*************************************************************************
*************************************************************/

// Store other user's pages in temporary array
for($i=0; $i<count($distinctusers); $i++)
{
 $pages="SELECT distinct(j_id) from searches where  ee_id ='$distinctusers[$i]'";
 $pages1 = mysql_query($pages,$link) or die("cant connect");
 unset($temporary);
 while($row=mysql_fetch_assoc($pages1))
{
 $temporary[]=$row['j_id'];
}

$totusers =count($userpages); //No of pages user has searched

$tempcount=count($temporary);   //No of pages of compared user
$totalcount=$totusers+$tempcount;
$common=array_intersect($userpages,$temporary); //Common pages
$noofcommon=count($common);//No of common users
$ratio =$noofcommon/$totalcount;     //Compute common ratio . No of common pages/Total
no of pages of both users

 if($largestratio< $ratio)    //Compare with all remaining users.
{
 $largestratio= $ratio;
```

```
 $recom=$distinctusers[$i];//Store user which has highest common ratio
}

 /*
echo "<br>Total users $totusers <br>";
//var_dump($tempcount);
echo "Temp count $tempcount<br>";
echo "Total count $totalcount<br>";
echo "Common pages <br>";
print_r($common);
echo "No of common $noofcommon<br>";
echo "ratio is $ratio <br>";
echo "Largest ratio $largestratio <br>";
echo "Recommended user $recom <br>";
*/
}

/*************************************************************************
*************************************************************/

//Store recommeneded user's pages not common with current user
if ($recom<>'')
{
 $recomarray=array();
 $recpages="SELECT distinct(j_id) from searches where  ee_id ='$recom'";
 $recpages1 = mysql_query($recpages,$link) or die("cant connect");

 while($row=mysql_fetch_assoc($recpages1))
{
 $recomarray[]=$row['j_id'];
}

$recomdiffpages =array_diff($recomarray,$userpages);
$finalrecomdarray= array_filter($recomdiffpages);
$finalrecomdarray=array_slice($recomdiffpages,0);
//print_r($finalrecomdarray) ;

/*************************************************************************
*************************************************************/
//Display recommended pages by recommended user
 for($k=0; $k<count($finalrecomdarray);$k++)
{
$disrecpages="SELECT j_id,j_title from jobs where  j_id ='$finalrecomdarray[$k]'";
$disrecpages1 = mysql_query($disrecpages,$link) or die("cant connect");
if (mysql_num_rows($disrecpages1) > 0)
```

```
 {
  echo"<div class=CSSTableGenerator1 style=width:500px;height:30px>";
  echo"<table border=1>";

 while($row=mysql_fetch_assoc($disrecpages1))
{
     echo '<tr><td><li><a
     href="job_details.php?id='.$row['j_id'].'">'.$row['j_title'].'</a></li></td></tr>';
}
echo"</table></div>";
}
}
}

/**********************************************************************************
*****************************************************/

//Display message if searches not sufficient
else
echo"<center><b>You Need to peform a few searches to start getting
Recommendations</b></center>";
mysql_close($link);
?>
</div>
<hr><br><br><br><br><br><br><br><br>


<div style="height:200px;overflow:scroll;max-height:500px;">
<h2>SUGGESTED JOBS BASED ON YOUR PROFILE</h2>

<?php
$link=mysql_connect("localhost","jobscope","riddhi")or die("can not connect");
mysql_select_db ("jobscope",$link) or die("can not select database");
$q="(SELECT ee_annualsalary,ee_current_location,ee_qualification from employees where
ee_id ='$sempid')";
$res=mysql_query($q,$link) or die("cant connect1");
while($row=mysql_fetch_assoc($res))
     {
     //$eesal=$row['ee_annualsalary'];
     $eeloc=$row['ee_current_location'];
     $eequal=$row['ee_qualification'];
     }

     $q1="SELECT j_id,j_title from jobs where  j_city LIKE '%$eeloc%' AND j_qualifications
     LIKE '%$eequal%' ";
     $res1=mysql_query($q1,$link) or die("cant connect2");
```

```
//echo $q1;
if (mysql_num_rows($res1) > 0)
{
echo"<div class=CSSTableGenerator1 style=width:500px;height:30px;display:inline-
block;>";
echo"<table border=1>";
while($row=mysql_fetch_assoc($res1))
        {
        echo '<tr><td><li><a
        href="job_details.php?id='.$row['j_id'].'">'.$row['j_title'].'</a></li></td></tr>';
        }
echo"</table></div>";
}

else
echo "<center><b>We could not find any matches.</b></center>";
mysql_close($link);
?>

<br><br><br><br>
</div>
<hr><br><br><br><br><br><br><br><br><br>

<div style="height:200px;overflow:scroll;max-height:500px;">
<h2>Recommendations by Collabrative Filtering</h2></h2>

<?php
//$itemID=$_GET['id'];
$userID=$_SESSION['eeid'];
$link=mysql_connect("localhost","jobscope","riddhi") or die("cant connect");
mysql_select_db("jobscope",$link) or die("cant select db");
$collabarray=array();
$collap="SELECT distinct(j_id) from jobs";
$collap1 = mysql_query($collap,$link) or die("cant connect");


while($row1=mysql_fetch_assoc($collap1))
{
 $collabarray[]=$row1['j_id'];
}

echo"<div class=CSSTableGenerator1 style=width:500px;height:30px>";
echo"<table border=1>";
```

```
 for($t=0; $t<count($collabarray);$t++)
{
 $itemID=$collabarray[$t];
  //echo "item id is '$itemID'<br>";
 $denom = 0.0; //denominator
 $numer = 0.0; //numerator
 $k = $itemID;
//echo "k='$k'<br>";

$sql = "SELECT r.j_id, r.rating
 FROM rating r WHERE r.ee_id=$userID AND r.j_id <> $itemID";
 $db_result = mysql_query($sql, $link);
 //for all items the user has rated
 while ($row = mysql_fetch_assoc($db_result))
 {
 $j = $row["j_id"];
 //echo"j='$j'<br>";
 $ratingValue = $row["rating"];

 //echo "ratingvalue= '$ratingValue'<br>";
 //get the number of times k and j have both been rated by the same user
 $sql2 = "SELECT d.count, d.sum FROM dev d WHERE itemID1=$k AND itemID2=$j";
 $count_result = mysql_query($sql2, $link);
 //skip the calculation if it isn't found
  if(mysql_num_rows($count_result) > 0)
 {
 $count = mysql_result($count_result, 0, "count");
 //echo "Count='$count'<br>";
 $sum = mysql_result($count_result, 0, "sum");
 //echo "Sum ='$sum'<br>";
//calculate the average
 $average = $sum / $count;
//echo"Average = '$average'<br>";
//increment denominator by count
$denom += $count;

 //echo "Denominator ='$denom'";
//increment the numerator
 $numer += $count * ($average + $ratingValue);
 //echo" Numberator = '$numer'<br>";
 }
 }
  if ($denom == 0)
        $collab=0;
 else
        $collab=($numer / $denom);
```

```
//echo "collab rating is'$collab'<br>";
if($collab>=3)
        {
        $cpid=$collabarray[$t];
        //echo "collab rating is'$collab'<br>";
        //echo "Job id is '$collabarray[$t]'";
        $sql5 = "SELECT j_title FROM jobs  WHERE j_id=$cpid";
        $db_result5 = mysql_query($sql5, $link);
        $row5=mysql_fetch_assoc($db_result5);
        echo '<tr><td><li><a
        href="job_details.php?id='.$cpid.'">'.$row5['j_title'].'</a></li></td></tr>';
        }
}

echo"</table></div>";
//$collab=predict($userID,$itemID);
/*echo   $itemID;
echo  $collab;
*/
?>
</div>
</div>
</div>
<!-- end #content -->
<div id="sidebar">

<?php
include("includes/sidebar1.inc.php");
?>
</div>
<!-- end #sidebar -->
<div style="clear: both;"> </div>
</div>
</div>
</div>
<!-- end #page -->
<div id="footer-bgcontent">
<div id="footer">
<?php
include("includes/footer.inc.php");
?>
</div>
</div>
<!-- end #footer -->
</body>
</html>
```

```
/*****************************************************************************
********************************************************/

//Add Rating
    <?php session_start();
    if( !empty($_POST["id"])) {
    require_once("dbcontroller.php");
    $db_handle = new DBController();
    $j_id=$_POST['id'];
    $ee_id=$_SESSION['eeid'];
    $rating=$_POST['rating'];
    $result = mysql_query("SELECT * FROM rating WHERE j_id ='$j_id' AND  ee_id
    ='$ee_id' ");

    if( mysql_num_rows($result) > 0) {
    mysql_query("UPDATE rating SET rating = '$rating' WHERE j_id ='$j_id' AND  ee_id
    ='$ee_id' ");
    }

    else
    {
    mysql_query("INSERT INTO rating (ee_id,j_id,rating) VALUES ('$ee_id','$j_id','$rating'");
    }
    }

    $link=mysql_connect("localhost","jobscope","riddhi") or die("cant connect");
    mysql_select_db("jobscope",$link) or die("cant select db");

    $itemID=$_POST['id'];
    $userID=$_SESSION['eeid'];

    // Get all of the user's rating pairs
    $sql = "SELECT DISTINCT r.j_id, r2.rating - r.rating as rating_difference
            FROM rating r, rating r2
            WHERE r.ee_id=$userID AND
                    r2.j_id=$itemID AND
                    r2.ee_id=$userID;";
    $db_result = mysql_query($sql, $link);
    $num_rows = mysql_num_rows($db_result);
    //For every one of the user's rating pairs,
    //update the dev table
    while ($row = mysql_fetch_assoc($db_result))
    {
            $other_itemID = $row["j_id"];
            $rating_difference = $row["rating_difference"];
     //if the pair ($itemID, $other_itemID) is already in the dev table
```

```
//then we want to update 2 rows.
if (mysql_num_rows(mysql_query("SELECT itemID1
FROM dev WHERE itemID1=$itemID AND itemID2=$other_itemID", $link)) > 0)
{
                $sql = "UPDATE dev SET count=count+1,
                    sum=sum+$rating_difference WHERE itemID1=$itemID
                    AND itemID2=$other_itemID";
                mysql_query($sql, $link);
                    //We only want to update if the items are different
                if ($itemID != $other_itemID) {
                   $sql = "UPDATE dev SET count=count+1,
                       sum=sum-$rating_difference
                       WHERE (itemID1=$other_itemID AND itemID2=$itemID)";
                   mysql_query($sql, $link);
}
}

else
{
 //we want to insert 2 rows into the dev table
$sql = "INSERT INTO dev VALUES ($itemID, $other_itemID,1, $rating_difference)";
mysql_query($sql, $link);
//We only want to insert if the items are different
 if ($itemID != $other_itemID)
{
                    $sql = "INSERT INTO dev VALUES ($other_itemID,
                    $itemID, 1, -$rating_difference)";
                     mysql_query($sql, $link);
}
}
}
?>
```

Front-End Code

```
<?php session_start();
$link=mysql_connect("localhost","jobscope","riddhi")or die("can not connect");
mysql_select_db("jobscope",$link) or die("can not select database");
$q="select * from jobs where j_active=1 order by j_id desc ";
$res=mysql_query($q,$link) or die ("can not select database");
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<?php
include("includes/head.inc.php");
?>
</head>

<body>
<div id="logo">
<?php
include("includes/logo.inc.php");
?>
</div>

<div id="header-wrapper">
<div id="header">
<div id="menu">
<?php
include("includes/menu.inc.php");
?>
</div>
<!-- end #menu -->
<!-- end #search -->
</div>
</div>
<!-- end #header -->
<!-- end #header-wrapper -->

<div id="search">
<?php
include("includes/search.inc.php");
?>
</div>
```

```
<div id="wrapper">
<div id="page">
<div id="page-bgtop">
<hr />
<!-- end #logo -->
<div id="content">
<div class="post">
<h2 class="title"><center><b>OUR TOP RECRUITERS</b></center></a></h2>
<div class="entry">
</div>
</div>
<div id="s_results">

<a href="http://www.tcs.com/Pages/default.aspx" target="_blank">
<img  src="tata.png" height="110" width="200" /></a>
<a href="http://careers.larsentoubro.com/Client/index.aspx" target="_blank">
<img  src="lnt.png" height="110" width="200" /></a>
<a href="https://airtel.taleo.net/careersection/airtel_externalcareersection/default.ftl"
target="_blank">
<img src="airtel.png" height="110" width="200"/></a>
<a href="http://careers.adityabirla.com/jobs" target="_blank">
<img src="abg.png" height="110" width="200"/></a>
<a href="https://www.allianz.com/en/careers" target="_blank">
<img src="allianz.png" height="110" width="200"/></a>
<a href="https://jobopenings.infosys.com/" target="_blank">
<img src="infosys.png" height="110" width="200"/></a>
<a href="https://www.amazon.jobs/" target="_blank">
<img src="amazon.png" height="110" width="200"/></a>
<a href="http://www.ibm.com/in-en/" target="_blank">
<img src="ibm.png" height="110" width="200"/></a>
<a href="https://www.accenture.com/in-en/careers.aspx" target="_blank">
<img src="accenture.png" height="110" width="200"/></a>
 </div>
</div>
<!-- end #content -->
<div id="sidebar">

<?php
include("includes/sidebar1.inc.php");
?>
</div>
<!-- end #sidebar -->
<div style="clear: both;"> </div>
</div>
</div>
```

```
</div>
<!-- end #page -->
<div id="footer-bgcontent">
<div id="footer">

<?php
include("includes/footer.inc.php");
?>
</div>
</div>
<!-- end #footer -->
</body>
</html>
```

Employee Register:

```
<?php session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : Flowerily
Description: A two-column, fixed-width design for 1024x768 screen resolutions.
Version    : 1.0
Released   : 20090906
-->

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script type="text/javascript">
function updateTextInput(val){
        document.getElementById('cas').value=val;
}
</script>

<?php
include("includes/head.inc.php");
?>
<script type="text/javascript" src="jquery-1.2.6.min.js"></script>
<SCRIPT type="text/javascript">
```

```
<!--
pic1 = new Image(16, 16);
pic1.src="loader.gif";

$(document).ready(function()
{
$("#username").change(function()
 {
 var usr = $("#username").val();

if(usr.length >= 6)
{
$("#status").html('<img src="loader.gif" align="absmiddle"> Checking availability...');
$.ajax(
{
    type: "POST",
    url: "check.php",
    data: "username="+ usr,
    success: function(msg)
{

   $("#status").ajaxComplete(function(event, request, settings){

    if(msg == 'OK')
{
        $("#username").removeClass('object_error'); // if necessary
        $("#username").addClass("object_ok");
        $(this).html(' <img src="tick.gif" align="absmiddle">');
}

else
{
        $("#username").removeClass('object_ok'); // if necessary
        $("#username").addClass("object_error");
        $(this).html(msg);
}
});
}
});
}

Else
{
    $("#status").html('<font color="red">' +
    'This username is already taken</font>');
    $("#username").removeClass('object_ok'); // if necessary
```

```
   $("#username").addClass("object_error");
}
});
});
//-->

</SCRIPT>
</head>
<body>
<body>
<div id="logo">
        <?php
        include("includes/logo.inc.php");
        ?>
        </div>
<div id="header-wrapper">
        <div id="header">
        <div id="menu">
                <?php
                include("includes/menu.inc.php");
                ?>
                </div>
                <!-- end #menu -->

                <!-- end #search -->
        </div>
</div>
<!-- end #header -->
<!-- end #header-wrapper -->

                <div id="search">
                <?php

                include("includes/search.inc.php");
                ?>
                </div>

        </div>
<div id="wrapper">
        <div id="page">
                <div id="page-bgtop">
                        <hr />
                        <!-- end #logo -->
                        <div id="content">
                                <div class="post">
```

```
                                        <h2 class="title">REGISTER</a></h2>
                                        <p class="meta">Please fill up the form</p>
                                        <div class="entry">
                                                <form action="process_employee_register.php"
method="post" enctype="multipart/form-data">
                                                        USER NAME <br> <input type="text"
id="username" name="username" style="width:300px;" minlength="6"  required="required"
pattern="[A-Za-z0-9]+"   placeholder="UserName should include alphanumeric characters of min 6
characters">
<div id="status"></div>
                                                        <br><br>NAME <br> <input type="text"
name="nm1" style="width:230px;"  required="required" pattern="[A-Za-z]+"   placeholder="Name
should only include alphabets"/>
                                                        <br><br> PASSWORD<br><input
type="password" name="pwd" minlength=6 required>
                                                        <BR><BR>GENDER <BR> <INPUT TYPE
= "RADIO" VALUE="MALE" name="gender" required>MALE<INPUT TYPE = "RADIO"
VALUE="female"name="gender" required >FEMALE
                                                        <br><BR> EMAIL <BR> <INPUT TYPE =
"email" name="email" style="width:200px;" required>
                                                        <BR><BR> ADDRESS <BR>
<TEXTAREA name="addr" style="width:200px;" required></TEXTAREA>
                                                        <BR><BR> PHONE NO. <BR> <INPUT
TYPE = "TEXT" name="ph" style="width:200px;" placeholder="Optional">
                                                        <BR> <BR>MOBILE NO.<BR> <INPUT
TYPE = "TEXT" name="mobile" style="width:200px;" required>
                                                        <br><br>CURRENT LOCATION
<BR><INPUT TYPE="TEXT" name="cl" style="width:200px;"required>
                                                        <BR><BR>SALARY EXPECTING(PER
ANNUM)<BR><INPUT TYPE="range" name="rangeInput"  min="50000" max="1000000"
step="1000" onchange="updateTextInput(this.value);"/>
                                                        <input type="text" name="cas" id="cas"
value="" disabled>


        <BR><BR>QUALIFICATION<BR><INPUT TYPE = "TEXT" name="quali"
style="width:200px;" required>
                                                        <BR><BR>KEY SKILLS<BR>
<TEXTAREA name="keywords" style="width:200px;"required> </TEXTAREA>
                                                        <br><br>RESUME<br><input type="file"
name="resume" style="width:200px;">
                                                        <center><br><br> <input type="submit"
value="Submit" ></center>
                                                </form>

                                        </div>
```

```
                                </div>

                        </div>
                        <!-- end #content -->
                        <div id="sidebar">
                        <?php
                include("includes/sidebar.inc.php");
                ?>
                        </div>
                        <!-- end #sidebar -->
                        <div style="clear: both;"> </div>
                </div>
        </div>
</div>
<!-- end #page -->
<div id="footer-bgcontent">
        <div id="footer">
                <?php
                include("includes/footer.inc.php");
                ?>
        </div>
</div>
<!-- end #footer -->
</body>
</html>
```

Employer Register:

```
<?php session_start();
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--
Design by Free CSS Templates
http://www.freecsstemplates.org
Released for free under a Creative Commons Attribution 2.5 License

Name      : Flowerily
Description: A two-column, fixed-width design for 1024x768 screen resolutions.
Version    : 1.0
Released   : 20090906
-->

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<?php
include("includes/head.inc.php");
?>
</head>
<body>
<div id="logo">
<?php
include("includes/logo.inc.php");
?>
</div>
<div id="header-wrapper">
<div id="header">
<div id="menu">
<?php
include("includes/menu.inc.php");
?>
</div>
<!-- end #menu -->
<!-- end #search -->
</div>
</div>
<!-- end #header -->
<!-- end #header-wrapper -->
<div id="search">
<?php
include("includes/search.inc.php");
?>
</div>
<div id="wrapper">
<div id="page">
<div id="page-bgtop">
<hr />
<!-- end #logo -->
<div id="content">
<div class="post">
<h2 class="title">REGISTER</a></h2>
<p class="meta">Please fill up the form</p>
<div class="entry">
<form action="process_employer_register.php" method="post">
FULL NAME <br> <input type="text" name="nm" style="width:200px;">
<br><br> PASSWORD <br> <input type ="password" name="pwd">
<br><BR> COMPANY NAME <BR> <INPUT TYPE = "TEXT" name="cnm"
style="width:200px;">
<BR><BR> COMPANY ADDRESS <BR>
<TEXTAREA name="addr" style="width:200px;">
</TEXTAREA>
```

```
<br><br> PHONE NUMBER <br><input type="text" name="ph" style="width:200px;">
<BR><BR> EMAIL <BR> <INPUT TYPE = "TEXT" name="email" style="width:200px;">
<BR><BR>COMPANY PROFILE<BR>
<TEXTAREA name="profile" style="width:200px;">
</TEXTAREA>
<center><br><br> <input type="Submit" value="submit"></center>
</form>
</div>
</div>
</div>
<!-- end #content -->
<div id="sidebar">

<?php
include("includes/sidebar.inc.php");
?>
</div>
<!-- end #sidebar -->
<div style="clear: both;"> </div>
</div>
</div>
</div>
<!-- end #page -->
<div id="footer-bgcontent">
<div id="footer">

<?php
include("includes/footer.inc.php");
?>
</div>
</div>
<!-- end #footer -->
</body>
</html>
```

Login Page:

```
<?php session_start();
if(empty($_POST))
{
        exit;
}

if(empty($_POST['unm'])||empty($_POST['pwd'])||empty($_POST['cat']))
{
        header("location:errormsg.php");
}
```

```
if($_POST['cat']=='employee')
{


	$link = mysql_connect("localhost","jobscope","riddhi") or die("Cannot Connect");
	mysql_select_db("jobscope",$link) or die("Cant select db");

	$q = "select * from employees where ee_fnm = '".$_POST['unm']."'";

	$res = mysql_query($q,$link) or die("wrong query");

	$row = mysql_fetch_assoc($res);


	if(!empty($row))
	{
		if($_POST['pwd']==$row['ee_pwd'])
		{
			//login
			$_SESSION = array();

			$_SESSION['unm']=$row['ee_fnm'];
			$_SESSION['eeid']=$row['ee_id'];
			$_SESSION['cat']='employee';
			$_SESSION['status']=1;
			$_SESSION['employee']=1;

			header("location:membwes.php");
		}
		else
		{
			echo "Wrong Password";
		}
	}
	else
	{
		echo "No Such User";
	}

}

if($_POST['cat']=='employer')
{


	$link = mysql_connect("localhost","jobscope","riddhi") or die("Cannot Connect");
```

```
	mysql_select_db("jobscope",$link) or die("Cant select db");

	$q = "select * from employers where er_fnm = '".$_POST['unm']."'";

	$res = mysql_query($q,$link) or die("wrong query");

	$row = mysql_fetch_assoc($res);


	if(!empty($row))
	{
		if($_POST['pwd']==$row['er_pwd'])
		{
			//login
			$_SESSION = array();

			$_SESSION['unm']=$row['er_fnm'];
			$_SESSION['eid']=$row['er_id'];
			$_SESSION['cat']='employer';
			$_SESSION['status']=1;
			$_SESSION['employer']=1;
			header("location:index.php");
		}
		else
		{
			header("location:errormsg.php");
		}
	}
	else
	{
		header("location:errormsg.php");
	}


}
?>
```

Chapter 7

Testing

7.1 Test Cases:

Table 7.1 Test Cases

Test Case ID	Input	Output	Description
01	Fill username field with the desired employee in the registration form.	If the username already exists in the database, a warning appears in red.	Fill the username of the employee in alphanumeric characters in order to go to the next field.
02	Fill Full name of employee in the registration form.	Go to the next blank to be filled in the registration form.	Fill the name of the employee in the required field in order to go to the next field.
03	Enter your email-id.	It accepts the entered text and goes to the next blank if it satisfies the specifications, else displays a warning.	While entering the employee email-id there should be"@" and "." in it so that it is accepted and move to the next field.
04	Enter a non-existent user-id and/or password.	An error message will be displayed.	If the entered user has not registered previously, an error is shown.
05	Enter an existent user-id and password.	Successful login.	If the user is already registered, he successfully logs in by providing his correct user-is and password.

Test Case ID 1:

Fig. 7.1.1: Test Case 1 (Enter Username)

Fig. 7.1.2: Test Case 1 (Username verified)

Test Case ID 2:

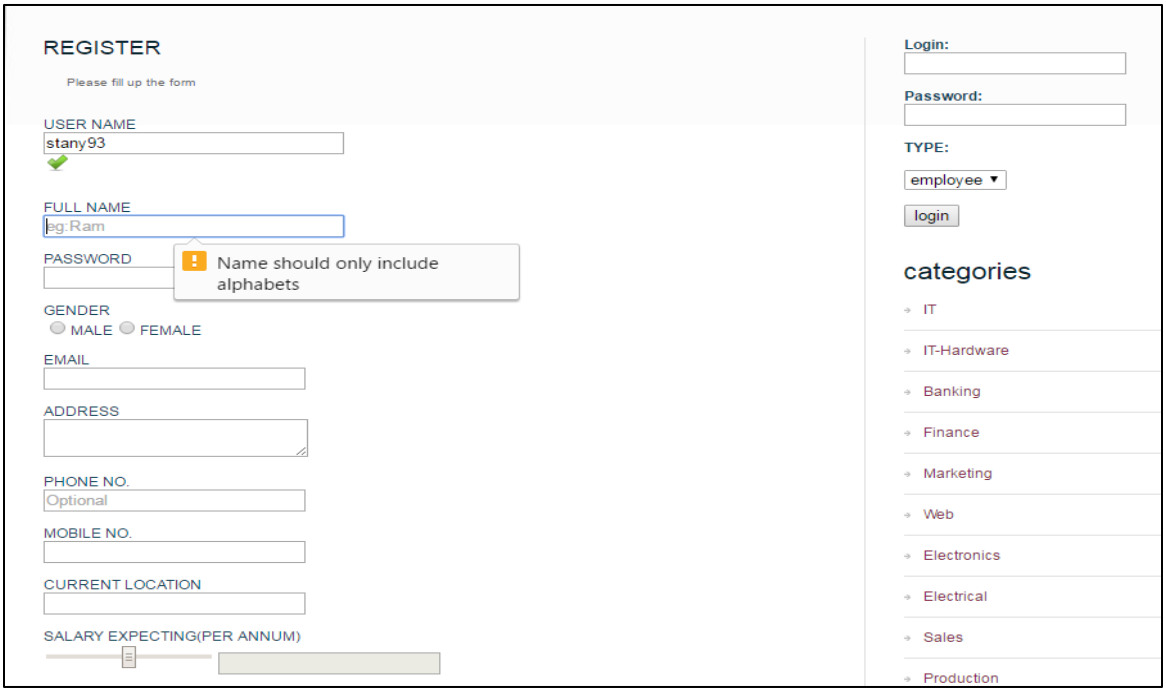


Fig. 7.1.3: Test Case 2 (Enter Fullname)

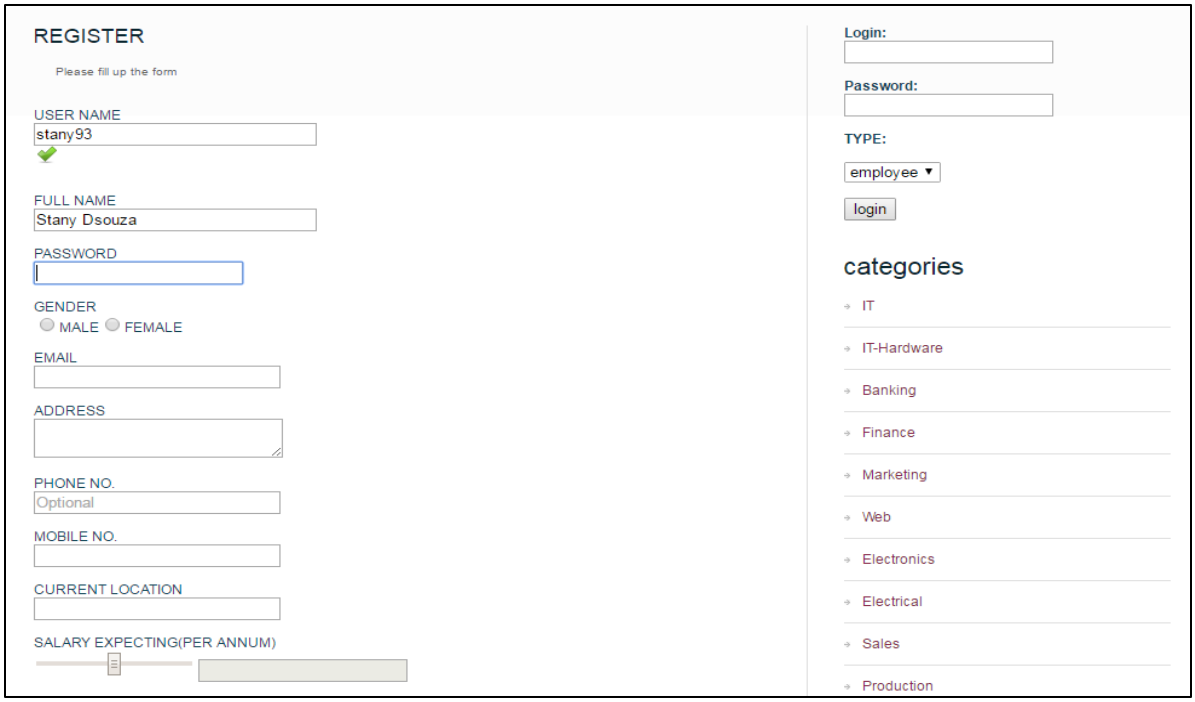


Fig. 7.1.4: Test Case 2 (Fullname verified)

Test Case ID 3:

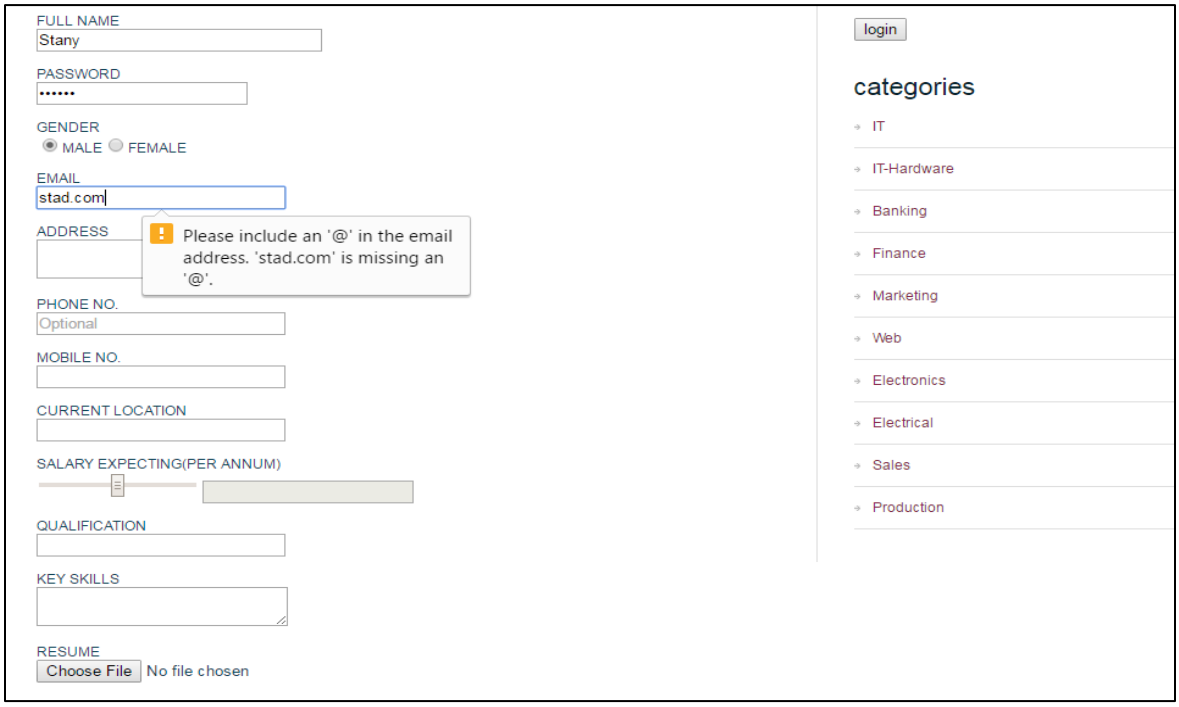


Fig. 7.1.5: Test Case 3 (Email-id verification)

Test Case ID 4:

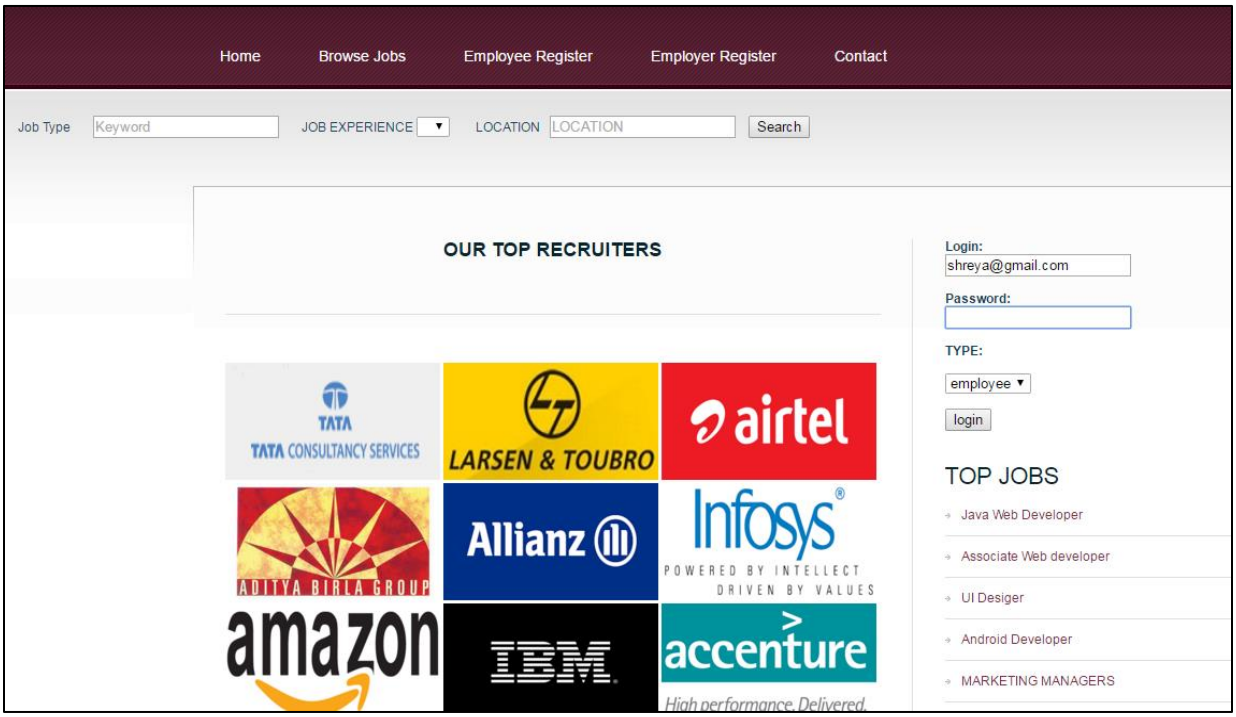


Fig. 7.1.6: Test Case 4 (Enter invalid login details)

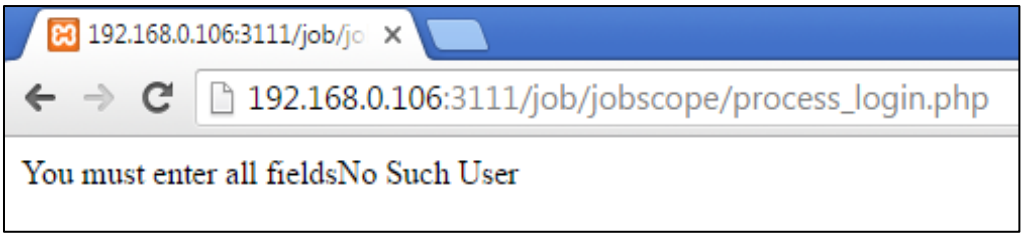


Fig. 7.1.7: Test Case 4 (Error message for non-existent user)

Test case 5:

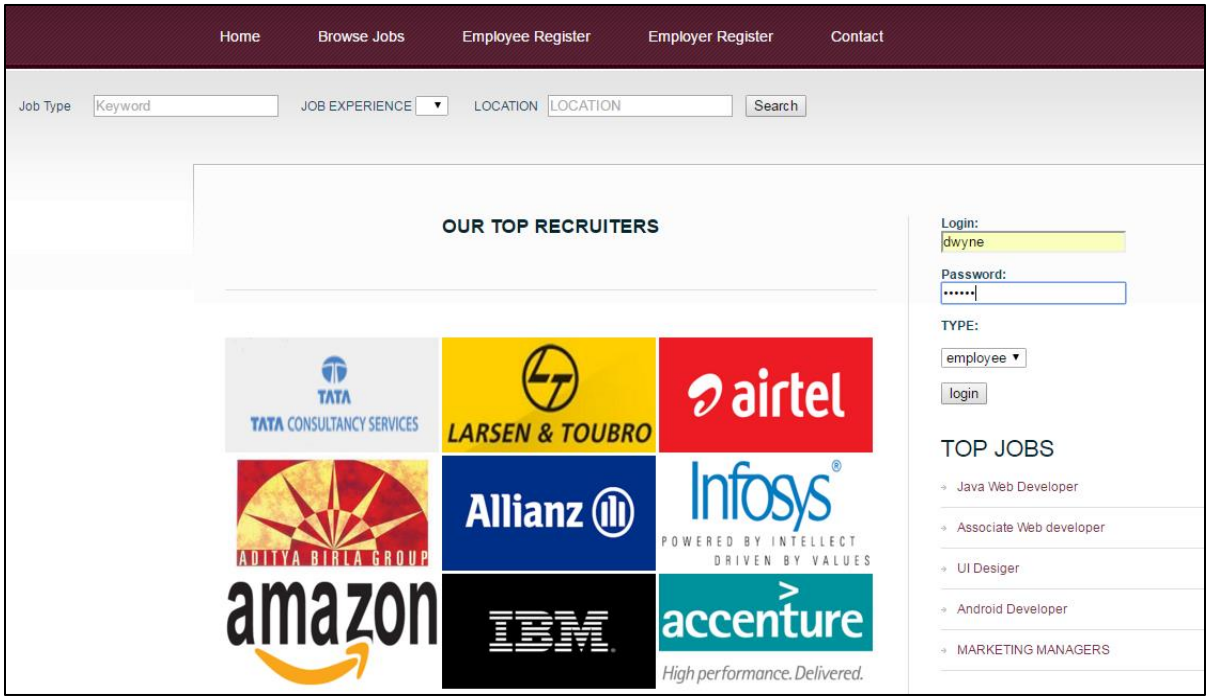


Fig. 7.1.8: Test Case 5 (Enter valid login details)

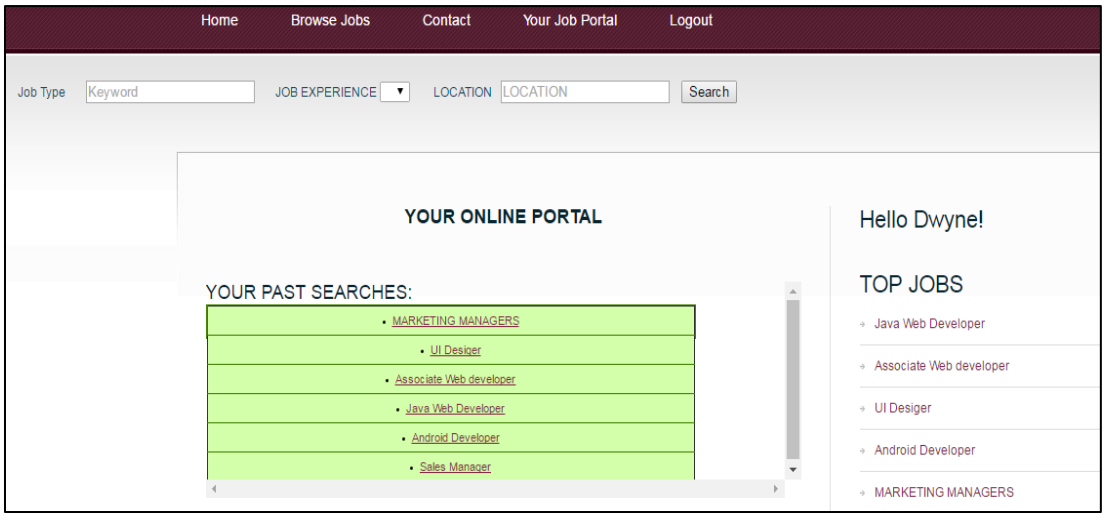


Fig. 7.1.9: Test Case 5 (Login page verification)

7.2 Types of Testing used

System testing is a critical phase implementation. Testing of the system involves hardware device and debugging of the computer programs and testing information processing procedures. Testing can be done with text data, which attempts to stimulate all possible conditions that may arise during processing. If structured programming Methodologies have been adopted during coding the testing proceeds from higher level to lower level of program module until the entire program is tested as unit. The testing methods adopted during the testing of the system were unit testing, integration testing, functional testing, performance testing and load/stress testing.

Unit Testing:

Unit testing focuses first on the modules, independently of one another to locate errors. This enables the tester to detect errors in coding and logical errors that is contained within that module alone. Those resulting from the interaction between modules are initially avoided.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure while at the same time to uncover the errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been detected by designing. It also tests to find the discrepancies between the system and its original objectives. Subordinate stubs are replaced one at time actual module. Tests were conducted at each module was integrated. On completion of each set another stub is replaced with the real module.

Functional Testing:

Functional testing is a technique in which all the functionalities of the program are tested to check whether all the functions that where proposed during the planning phase are full filled. This is also to check that if all the functions proposed are working properly. This is further done in two phases:

- One before the integration to see if all the unit components work properly
- Second to see if they still work properly after they have been integrated to check if some functional compatibility issues arise.

Performance Testing:

Expected Result

1. The user should be able to connect to the server properly without any problems.
2. The connection establishment between the device and the server should take minimal time.
3. The device should be able receive data from the server uninterruptedly.
4. Information provided by the application should be correct and as per the user's need.

Observation

1. Connection can be established easily provided that the server is on.
2. The connection with the server takes time as it uses Internet connection.
3. Receiving data from the server takes time.
4. Information coming from the database is correct.

Load/Stress Testing:

Expected Result

1. Response time should be unaffected irrespective of the no of users.
2. The introduction of the newer clients should not make the server to work hap hazardously.
3. Continuous use of the server by different clients should not result into the server getting slowed down.
4. Response time should not be degraded if there is congestion in network.

Observation

1. The speed of transmission was fine even when the newer clients were getting added. The response of the server was satisfying even with the introduction of newer client.

Chapter 8

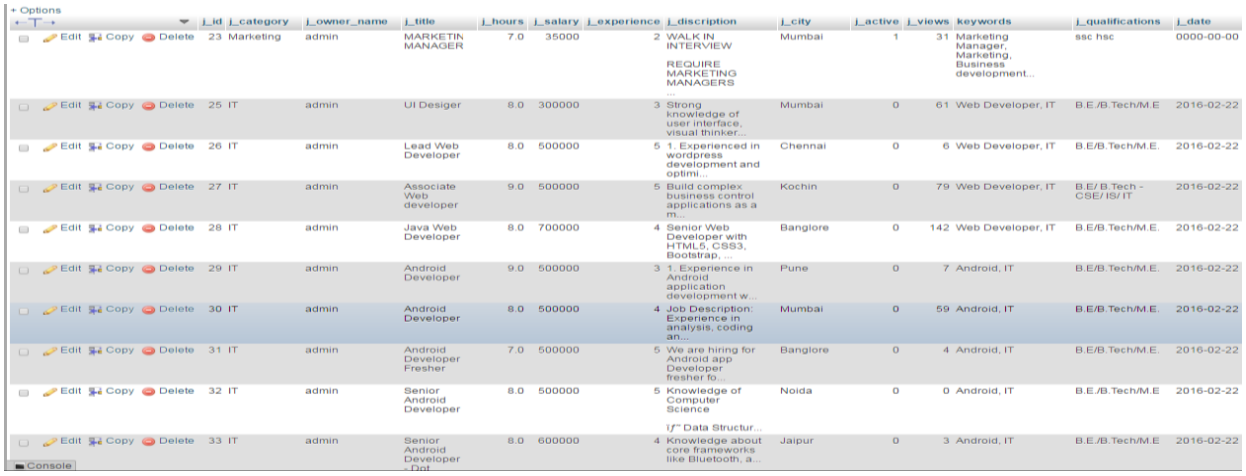
Results & Discussions

This section analyses the results of the experiments in order to show effective comparison between the considered algorithms in terms of accuracy. This section also includes a graphical analysis of the user demographics and the number of page views.

Accuracy

Step wise results are shown below for web log records from Jobscope database.

Step 1: Collection of web logs which are in raw or unprocessed form.14 attributes are shown below:



+ Options

	j_id	j_category	j_owner_name	j_title	j_hours	j_salary	j_experience	j_discription	j_city	j_active	j_views	keywords	j_qualifications	j_date
Edit Copy Delete	23	Marketing	admin	MARKETIN MANAGER	7.0	35000	2	WALK IN INTERVIEW REQUIRE MARKETING MANAGERS ...	Mumbai	1	31	Marketing Manager, Marketing, Business development...	ssc hsc	0000-00-00
Edit Copy Delete	25	IT	admin	UI Desiger	8.0	300000	3	Strong knowledge of user interface, visual thinker...	Mumbai	0	61	Web Developer, IT	B.E./B.Tech/M.E	2016-02-22
Edit Copy Delete	26	IT	admin	Lead Web Developer	8.0	500000	5	1. Experienced in wordpress development and optimi...	Chennai	0	6	Web Developer, IT	B.E/B.Tech/M.E.	2016-02-22
Edit Copy Delete	27	IT	admin	Associate Web developer	9.0	500000	5	Build complex business control applications as a m...	Kochin	0	79	Web Developer, IT	B.E/ B.Tech - CSE/ IS/ IT	2016-02-22
Edit Copy Delete	28	IT	admin	Java Web Developer	8.0	700000	4	Senior Web Developer with HTML5, CSS3, Bootstrap, ...	Banglore	0	142	Web Developer, IT	B.E/B.Tech/M.E.	2016-02-22
Edit Copy Delete	29	IT	admin	Android Developer	9.0	500000	3	1. Experience in Android application development w...	Pune	0	7	Android, IT	B.E/B.Tech/M.E.	2016-02-22
Edit Copy Delete	30	IT	admin	Android Developer	8.0	500000	4	Job Description: Experience in analysis, coding an...	Mumbai	0	59	Android, IT	B.E/B.Tech/M.E.	2016-02-22
Edit Copy Delete	31	IT	admin	Android Developer Fresher	7.0	500000	5	We are hiring for Android app Developer fresher fo...	Banglore	0	4	Android, IT	B.E/B.Tech/M.E.	2016-02-22
Edit Copy Delete	32	IT	admin	Senior Android Developer	8.0	500000	5	Knowledge of Computer Science i/" Data Structur...	Noida	0	0	Android, IT	B.E./B.Tech/M.E	2016-02-22
Edit Copy Delete	33	IT	admin	Senior Android Developer - Dot	8.0	600000	4	Knowledge about core frameworks like Bluetooth, a...	Jaipur	0	3	Android, IT	B.E./B.Tech/M.E	2016-02-22

Console

Fig. 8.1: Attributes of jobs

Step 2: Preprocessing is done for web log records. Cleansing, User and Session Identification, Content Retrieval and Path Completion applied on records. Thus, processed records for a user id 16 in user's sessionized form are as shown below.

EmployeeId	EmployeeName	EmployeeEmail	EmpoyeeLocation	EmployeeQualif	JobIdSearched	JobTitle	JobSalary	JobDate
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	30	Android OS Developer	500000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	51	Mobile App Developer(IOS)	600000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	52	Mobile App Developer(Android)	700000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	54	Java Web Developer	700000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	53	Windows App Developer	800000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	31	Android Developer Fresher	500000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	33	Senior Android Developer - Dot	600000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	25	UI Desiger	300000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	27	Associate Web developer	500000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	67	Hardware Trainee	500000	2016-02-23
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	65	Computer Hardware Engineer	500000	2016-02-23
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	70	Embedded Hardware Engineer - T	500000	2016-02-23
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	68	HIL- Hardware Job	500000	2016-02-23
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	42	Software Tester	300000	2016-02-22
16	Stany	stanydsouza93@gmail.com	Mumbai	B Engg	28	Java Web Developer	700000	2016-02-22

Fig. 8.2: Processed records for user id 16

Step 3: Pages are generated for the URI/pages/page view accessed by user.

Step 4: Consider user Id 16 having varying past searches. Based on these past searches a number of recommendations are generated. We apply Longest Common Subsequence algorithm on these searches. Thus the intuition list obtained is from the history of user's navigation pattern and of the user's whose usage pattern is same as the user.

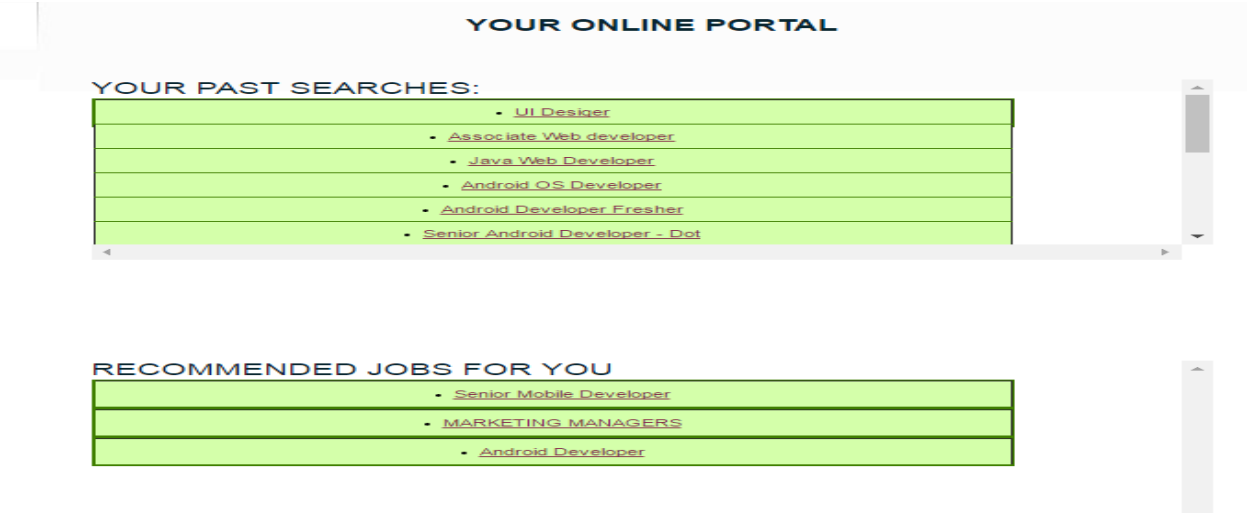


Fig. 8.3: Past searches of user id 16 with recommendations based on LCS

Step 5: Apply Collaborative filtering algorithm to get the recommendations based on user's rankings.

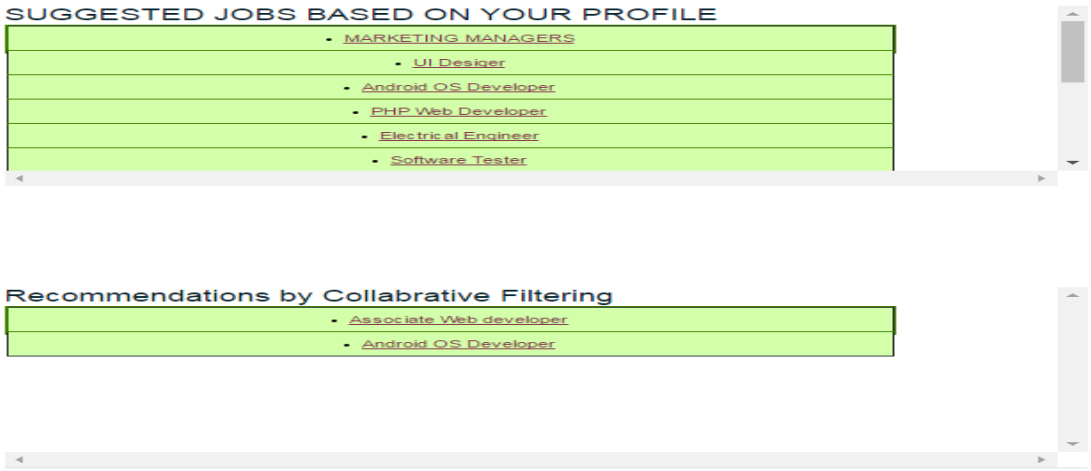


Fig. 8.4: Recommendations based on collaborative filtering

Thus, both the lists are combined into the Final Recommendation List of the user. Finally, Accuracy is calculated for the final recommendation list.

Accuracy measures the degree to which the recommendation system produces accurate recommendations. It is given by:

$$\text{Accuracy} = \frac{\text{Number of searches}}{\text{Number of recommendations}} * 100$$

Case 1: Accuracy of Longest Common Subsequence algorithm

Fig. 8.5: recommendation list for user id=16 and accuracy is 66.67%

$$\text{Accuracy} = \frac{\text{Number of searches}}{\text{Number of recommendations}} * 100 = 2/3*100 = 66.67\%$$

Case 2: Accuracy of Collaborative Filtering algorithm

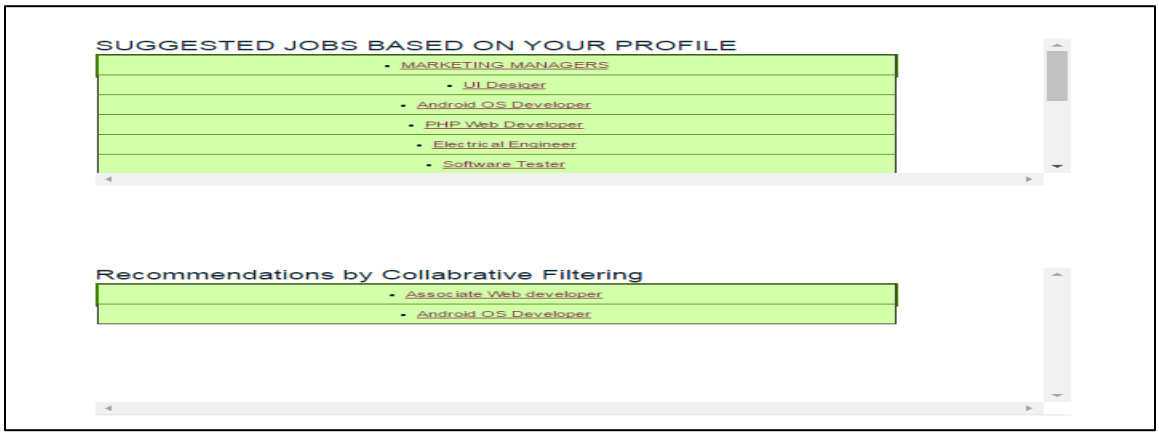


Fig. 8.6: recommendation list for user id=16 and accuracy is 100%

$$\text{Accuracy} = \frac{\text{Number of searches}}{\text{Number of recommendations}} * 100 = 3/3*100 = 100\%$$

Thus, from above cases we can prove that accuracy of the recommendation list increases if the number of page views is more in the user navigation pattern.

User Demographics

User demographics may influence how satisfied users are with recommendations.

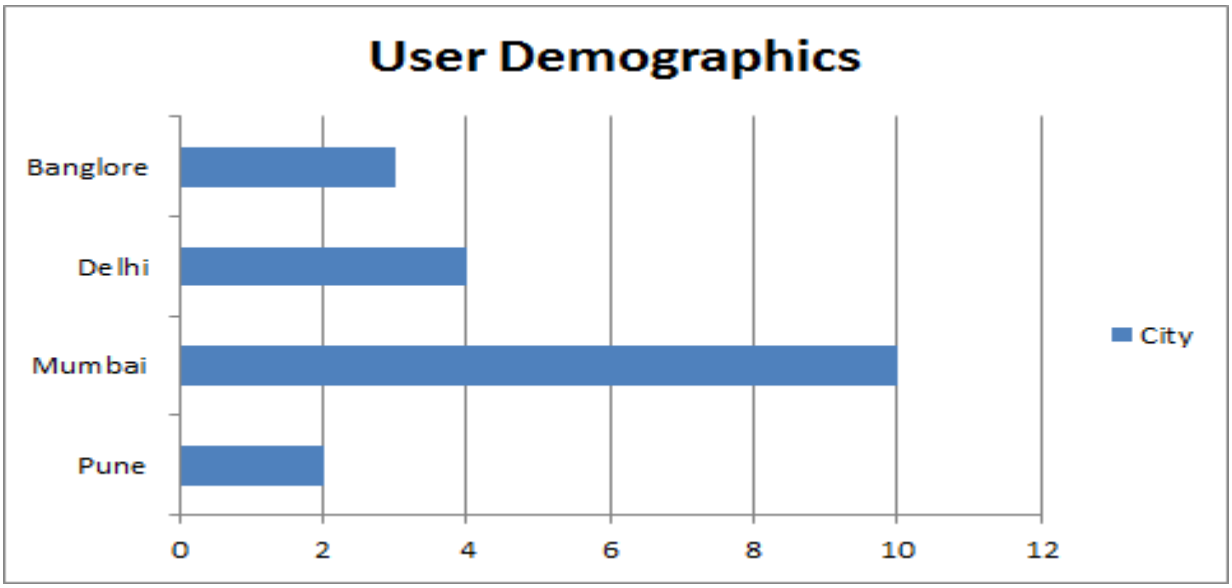


Fig. 8.7: User demographic

Number of page views

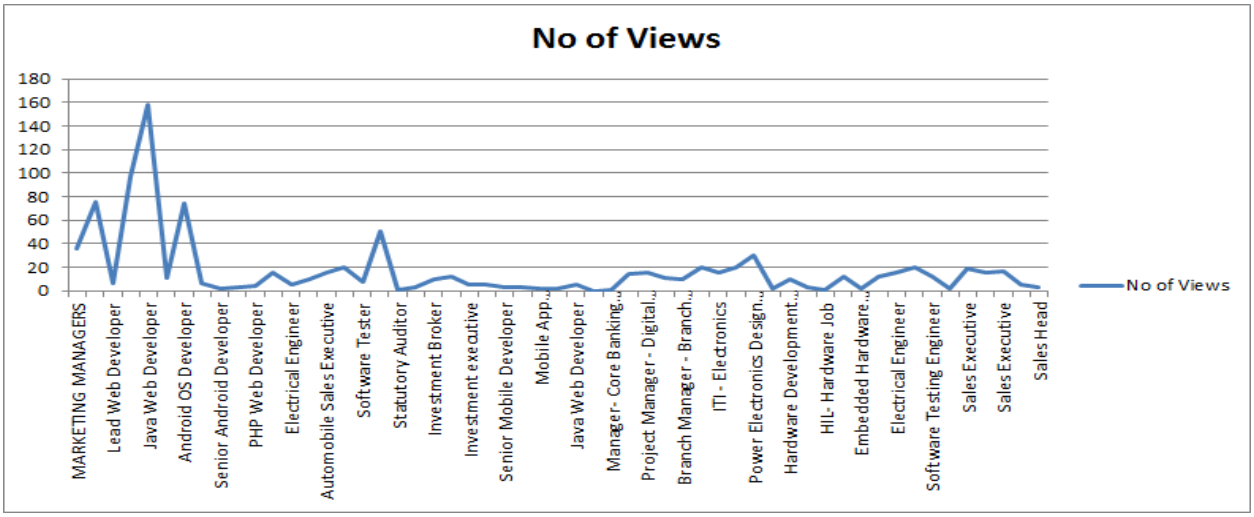


Fig. 8.8: Job Search trends

Chapter 9
Conclusions & Future Scope

Thus, we propose a two tier architecture for capturing users' intuition in the form of recommendation list containing list of pages visited by user and also list of pages visited by other users having similar usage profile. The practical implementation of proposed architecture and algorithms shows that the personalized recommendation system could find the job seekers' customs and interests and the accuracy of the recommendation list increases if the number of page views is more in the user navigation pattern. Therefore, we have developed a system which is beneficial for online job seekers to attain the relevant information or relevant webpages based on his search pattern. By using collaborative filtering, users can be given recommendations based on theirs and other users' ratings for a particular job. Based on these ratings any user can decide whether a particular job is worth searching or not. This system will leave aside all the drawbacks of the existing system and is one that is easy to use. The two main goals that will be achieved with this are to improve website usability and provide users convenience and suggestions in a web environment.

The project has a future scope when very large numbers of datasets need to be taken into consideration. Providing suggestions to the user on the basis of a dynamic scenario will be put into effect in the future.

Literature Cited

Journal Paper references:

[1] R. Baraglia and F. Silvestri, "Dynamic personalization of web sites without user intervention," Communications of the ACM,vol. 50, pp. 63-67, 2007.

[2] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on Web usage mining" Communications of the ACM, vol. 43, pp. 142-151, 2000.

[3] M. Jalali, N. Mustapha, A. Mamat, Md N. Sulaiman, “OPWUMP An architecture for online predicting in WUM-based personalization system”, In 13th International CSI Computer Science, Springer Verlag,2008. 307

[4] Dipa Dixit, Jayant Gadge,“Automatic Recommendation for Online Users Using Web Usage Mining”, Thadomal Shahani Engineering College, MU.

Web references:

[5] http://www.wikipedia.com

[6]Jiwae Han and Michelin Kamber,“Data Mining Concepts and Techniques”, Elsevier Publications, edition 2006.

Acknowledgements

We hereby take the privilege to present our report on "**Web-based mining in Personalized Online Recruiting System**". We are very grateful to our project guide Ms. Vaishali Jadhav for her constant guidance and for contributing her valuable time in the project from her busy schedule right from the project's inception.

We would like to thank the Director Rev. Bro. Melchior Tom and the Principal Dr. A.K. Sen for giving us the opportunity to work on the project. We would like to express our deep gratitude towards Mr. Bhavesh Pandya for his inspiration and timely guidance. His suggestions have greatly contributed for the betterment of our project.

Our special thanks to the Head of the Department Mr. Pramod Shanbhag, staff members and lab assistants for their co-operation and attention to the problems we faced.