Chapter 1
INTRODUCTION

1. INTRODUCTION

Job Portals are like the meeting points for the recruiters as well as the job seekers where each aims at meeting their individual requirements. The job seekers try to find a job opportunity where they can apply their knowledge, acquire new skills and grow as a professional. On the other hand, recruiters try to fill their job openings with the right candidate who has the perfect aptitude and qualification to handle the responsibilities efficiently. Job portals are like a market place where the demand meets the supply.

Government jobs portal is a web application designed to provide users with the information related to jobs in government sectors. This was chosen as India is one of the highest populated country and unemployment rate is also highest so, to help people in getting jobs and also to help the country was main objective.
In this a user has to register himself by providing complete education and qualifications details. The user will be informed about the related jobs for which he/she is eligible, and also by clicking on links available he can get the complete information about the interview schedule and the other details.

Here the details will be provided to the user based on the highest educational qualifications of the user and also if location is criteria given by user then all the jobs for that locations will be forwarded to the user. This job portal can prove to be very useful to users of different profile to search jobs on the basis of the qualifications. Every user can access through user id and apply fo9r multiple jobs at a time.

Also the admin will notify the user in advance about the upcoming schedules, pay packages, selection criteria etc. via email

Chapter 2

REVIEW OF LITERATURE

2. REVIEW OF LITERATURE

Internet has become an essential element in our lives allows us to communicate with virtually no barriers, make our purchases, inform and seek work. In fact, online job search has become an important tool in finding employment. Today there are many employment agencies and websites online that offer job search services enable that work to those who are unemployed right now. One of these portals is Employment Office.

The importance of sites such as Employment Office is not only about the possibility of finding a job through here you can also find all the news related to the workplace.

Focusing on the work of Internet job search, what we provide portals how Employment Office? Why are major job portals?

1. Most of these sites are free so you do not involve any cost to the user.

2. They personalize the job search so that each time a company issues a bid, the network automatically to candidates who meet the profile required by email or SMS.

3. Another advantage is the great transparency that offers the Web the public distribution of job offers the candidate makes a real prospect of its ability to access a particular job.

4. Interactivity in many portals candidates can decide what information your resume can be viewed or not companies.

5. A very important issue is that companies very actively involved in these portals starting them recruitment process at any time. These portals, as well as area candidates, companies have own area through which staff seek to fit their profile and needs.

6. On these sites you will also find a section for training. For example, in Office Jobs you can find a wide variety training and courses ranging from management, communication, sports, marketing, engineering, information technology, tourism. These courses help to improve your chances to find work and increase your professional skills.

Employment Office, like many other websites of job search [2], help to facilitate job search and that the negotiations between candidates and companies are carried out in an agile and fast. This is definitely a great service that tries to encourage and improve job search among the population and

meet their labor needs. The purpose of designing the online job portal is to give the job seekers a platform for finding a right and a satisfactory job according to their qualification [4]. It also connects the job seekers with the major industries [3]. It also provides job seekers to submit their CV and can apply for job posting and employer can select best employees from the available CV based on their payment option selection. This is basically a job portal where job seekers apply for jobs and when jobs are available they are notified by e-mail [6].

Job portal is prepared for provide all category of job and help to get various type of jobs. The main purpose of job portal is to provide facility of job seekers for getting quick jobs [1]. So, it enables applicants to search for jobs in a convenient manner and to enable employers to find a suitable candidate.

2.1 Scope of Literature

- Maintain job seeker and employer record.
- Maintain uploaded resume details.
- Provide customized job postings.
- Maintains job postings results and generate various reports

Chapter 3

Proposed Approach

&

System Architecture

3.1 Proposed Approach

There are dozens of web-portals available but user doesn’t know which one to use and where to register according to their qualification. In the various existing web-portals we have a large amount of redundant information which many a times confuses the user.

Hence we proposed the idea of govt. job portal where user will get notified with the jobs which are related to his/her chosen criteria.

Criteria may include the location, stream or the particular string (entered by user).the jobs are showed to home page of the user and these jobs are sorted according to the criteria entered.
The past data is archived and kept in database so that it can be used for future analysis purpose and user queries for date can be satisfied.

3.2 SYSTEM ARCHITECTURE

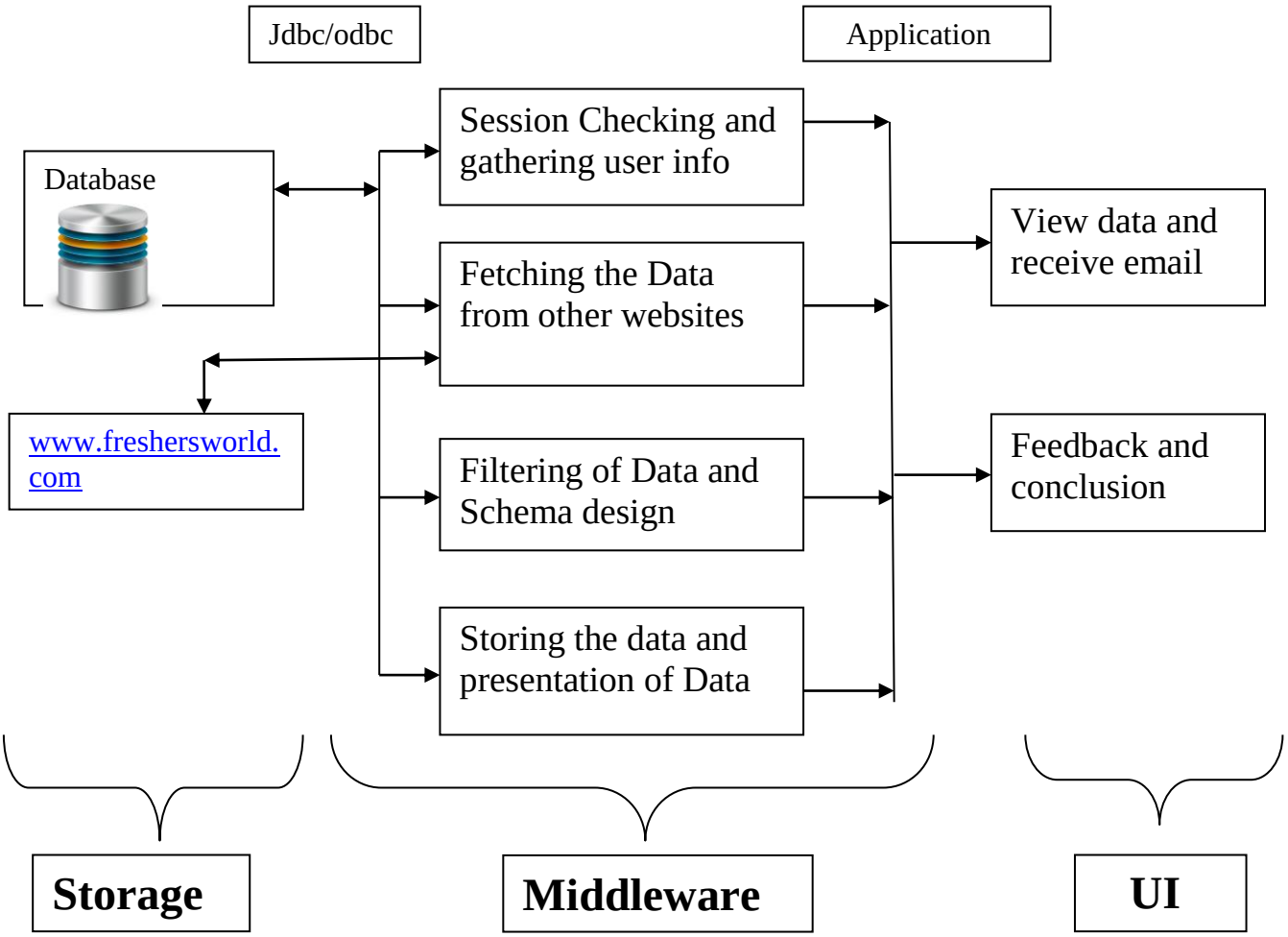


Fig 3.2: System Architecture

The system Architecture consists of the parts viz. The UI (User Interface), Middleware, and the storage. The UI consists of the JSP and Servlets through which the user will interact with the system or website.

The middleware consists of the main program code and logic which takes the data and input from the user and stores it in the database. Then according to the request of the admin it fetches the data from the website and stores it in the database. The stored data is then processed to filter relevant jobs from the data. The filtered data is stored in the database which is used as the master database for the website. The middleware also consists of the tender module which fetches the tenders available and displays it to the user. The filtering module filters the jobs for a user according to his/her location, interest and date. The feedback module takes feedback from the user and performs analysis on it and displays result of the analysis. The email module mails the available jobs to the registered user on their registered user id.

The oracle 10g is used as the backend database which stores the information from the personal details to the jobs fetched. The database stores the personal information, educational qualification, unfiltered jobs and the filtered jobs.

Chapter 4

System Description

4.1 Technology stack

The Government Jobs portal web application will be implemented in Java/J2EE and it will be hosted/deployed in free application server (i.e. Apache Tomcat).

Following technologies, tools and software's is used in **Government Jobs portal** web application

1. HTML [*Hyper Text Markup Language, It is used to create static web pages*].
2. JSP [*Java Server Pages, It is used to create dynamic web content*].
3. Servlet
4. CSS [*Cascading style sheet*].
5. Core Java.
6. Jdk 1.7 or above.
7. Apache Tomcat 8.0.3.0.
8. Oracle 10g.
9. Jdbc-odbc Connector jar for Oracle.
10. NetBeans IDE.

4.2 TECHNOLOGY DESCRIPTION

JSP:

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to servlet because it provides more functionality than servlet such as expression language, jstl etc.

A JSP page consists of HTML tags and JSP tags. The jsp pages are easier to maintain than servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tag etc.

SERVLET:

Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side without a face. Java servlets make many Web applications possible.

ORACLE 10g:

10g is Oracle's grid computing product group including (among other things) a database management system (DBMS) and an application server. In addition to supporting grid computing features such as resource sharing and automatic load balancing, 10g products automate many database management tasks. The Real Application Cluster (RAC) component makes it possible to install a database over multiple servers.

4.3 System Specification

Hardware:

IBM compatible , Intel Pentium 4,Intel core-i3 based PC with a monitor ,keyboard and mouse, system must have 1 GB Ram, Hard disk 80 GB or of available memory.

Operating System:

Windows XP or Windows 7 or Windows 8.

Software needed:

Standard web browser, Oracle 10g, Glassfish 3.0 with NetBeans 6.1 IDE or Upgrade version , MVC Architecture.

Standard browser:

Google chrome

4.4 Functional Architecture

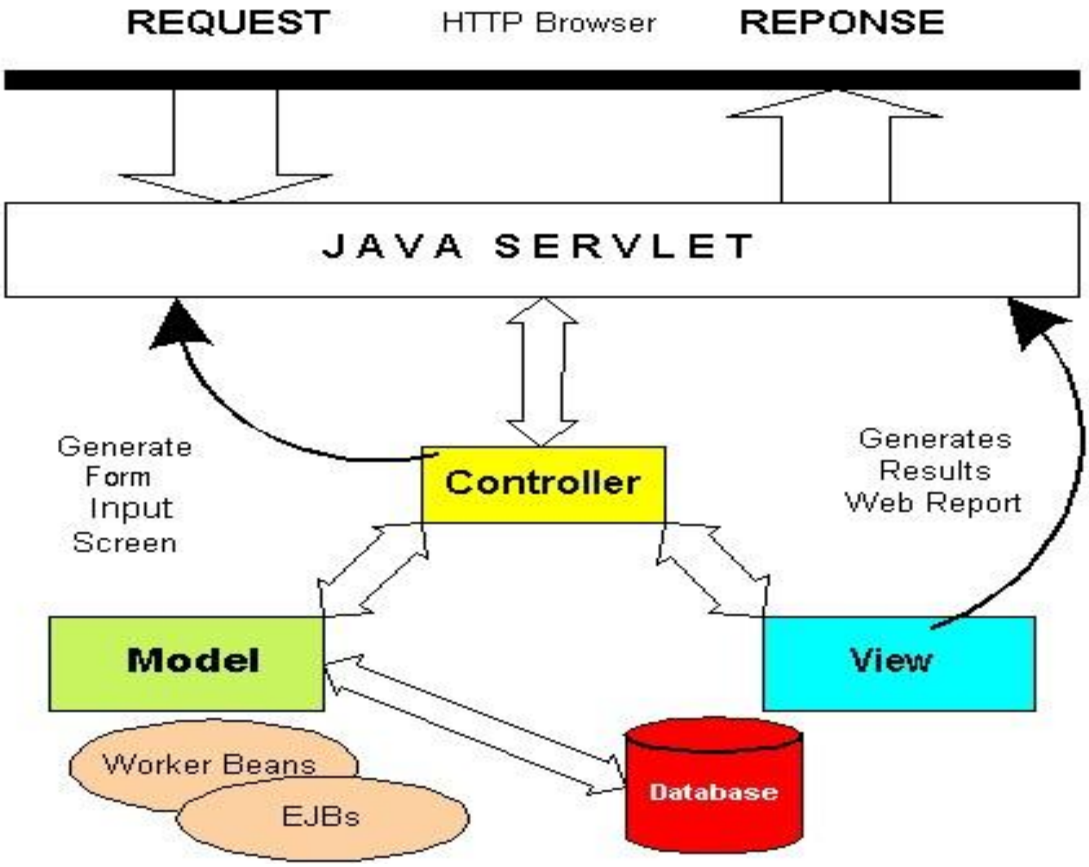


Fig 4.4: Functional Archtecture

The Government job Portal is based on the MVC (Model View Controller) design pattern. The MVC design pattern consists of three modules model, view and controller.

Model The model represents the state (data) and business logic of the application.

View The view module is responsible to display data i.e. it represents the presentation.

Controller The controller module acts as an interface between view and model. It intercepts all the requests i.e. receives input and commands to Model / View to change accordingly.

4.5 Functionality

Admin

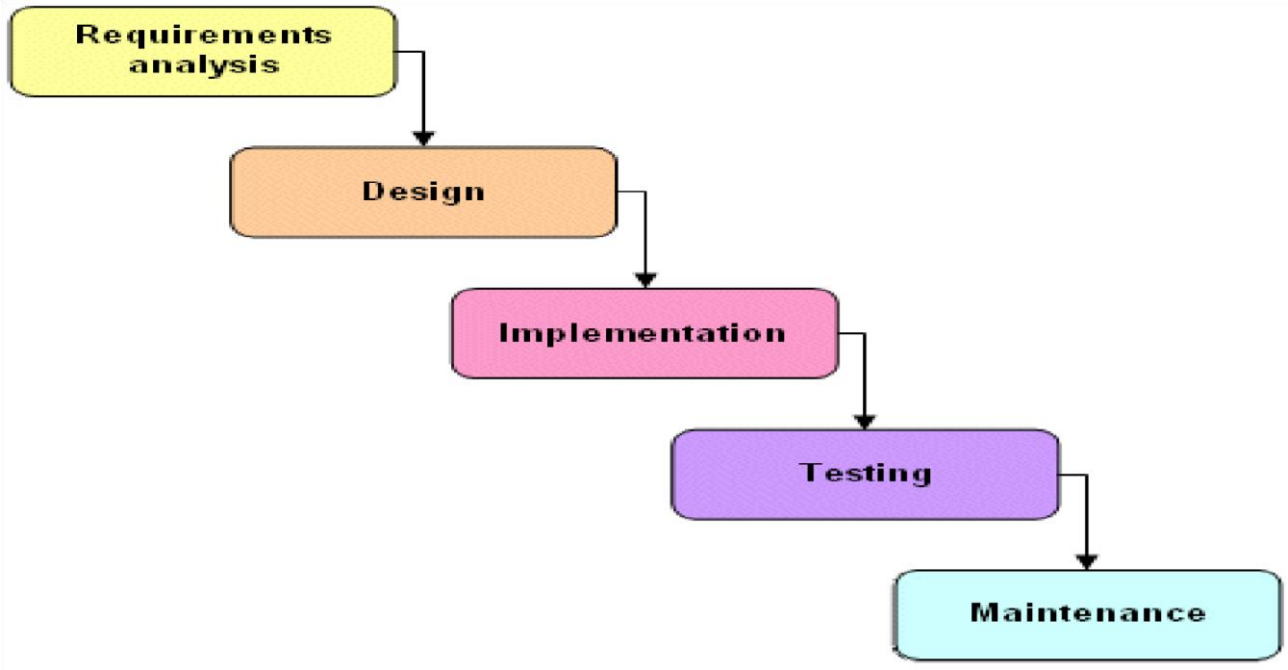
- Admin can log-in the system.
- View tenders
- Send email
- Update job database
- View analysis

Users

- User can register in the system
- User can log-in the system
- can view tender
- Can give feedback
- Search jobs based on location and stream.
- Update profile Details

4.6 METHODOLOGY

The **waterfall model** is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a cascade of waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance.



Waterfall Model:

The name of this model is justified by the diagrammatic representation which resembles a cascade of waterfall. It consists with intuitive set of phases. It has 5 phases:

1. Requirements
2. Design
3. Implementation
4. Verification
5. Maintenance

The different phases starting from feasibility study to integration and testing phase & delivery is known as developmental pan. At the end of developmental part, product is to be delivered to customer and maintenance commences after that.

An activity that's spans all phases of any software development is project management. Even though conveniently omitted in the life cycle diagram, project management nevertheless is an important activity in the life cycle and deals with the managing the effort at all stages of product development and maintenance.

4.7 COST ESTIMATION MODEL

Cocomo model for Government jobs Portal:

The government job portal has average complexity and fair flexibility. Thus it is classified as organic project.

- effort=2.4*(size)^1.05 PM
- Time to development=2.5 *(effort)^0.38Months

Where,

Effort = number of staff months PM

size = no of lines of code to complete product

time=total months

Effort=2.4*4.0^1.05=10.29 PM

Time= 2.5*10.29^0.38=6 months (development time)

4.8 DATA MODELS

Table 1: Job_Users

Column Name	Data Type	Nullable	Default	Primary Key
USERNAME	VARCHAR2(50)	No	-	1
FULLNAME	VARCHAR2(50)	Yes	-	-
PASSWORD	VARCHAR2(50)	Yes	-	-
				1 - 3

Table 2: Job_Education

Column Name	Data Type	Nullable	Default	Primary Key
COURSE	VARCHAR2(30)	Yes	-	-
BRANCH	VARCHAR2(50)	Yes	-	-
				1 - 2

Table 3: Job_Qualification

Column Name	Data Type	Nullable	Default	Primary Key
USERNAME	VARCHAR2(50)	Yes	-	-
QUALIFICATION	VARCHAR2(50)	Yes	-	-
STREAM	VARCHAR2(50)	Yes	-	-
PERCENT	NUMBER(5,2)	Yes	-	-
START_YEAR	NUMBER	Yes	-	-
END_YEAR	NUMBER	Yes	-	-
				1 - 6

Table 4: Fetched Data

Column Name	Data Type	Nullable	Default	Primary Key
DATA	VARCHAR2(2000)	Yes	-	-
				1 - 1

Table 5: Fetched jobs

Column Name	Data Type	Nullable	Default	Primary Key
COMPANY_NAME	VARCHAR2(100)	No	-	1
POST_DATE	VARCHAR2(100)	No	-	2
LOCATION	VARCHAR2(100)	No	-	3
POSITION	VARCHAR2(100)	No	-	4
ELIGIBILITY	VARCHAR2(500)	No	-	5
ON_DATE	VARCHAR2(100)	No	-	6
				1 - 6

Chapter 5

Results

&

Discussions

Results

Initially the user get the following screen after opening the Netbeans IDE as shown in fig 5.1

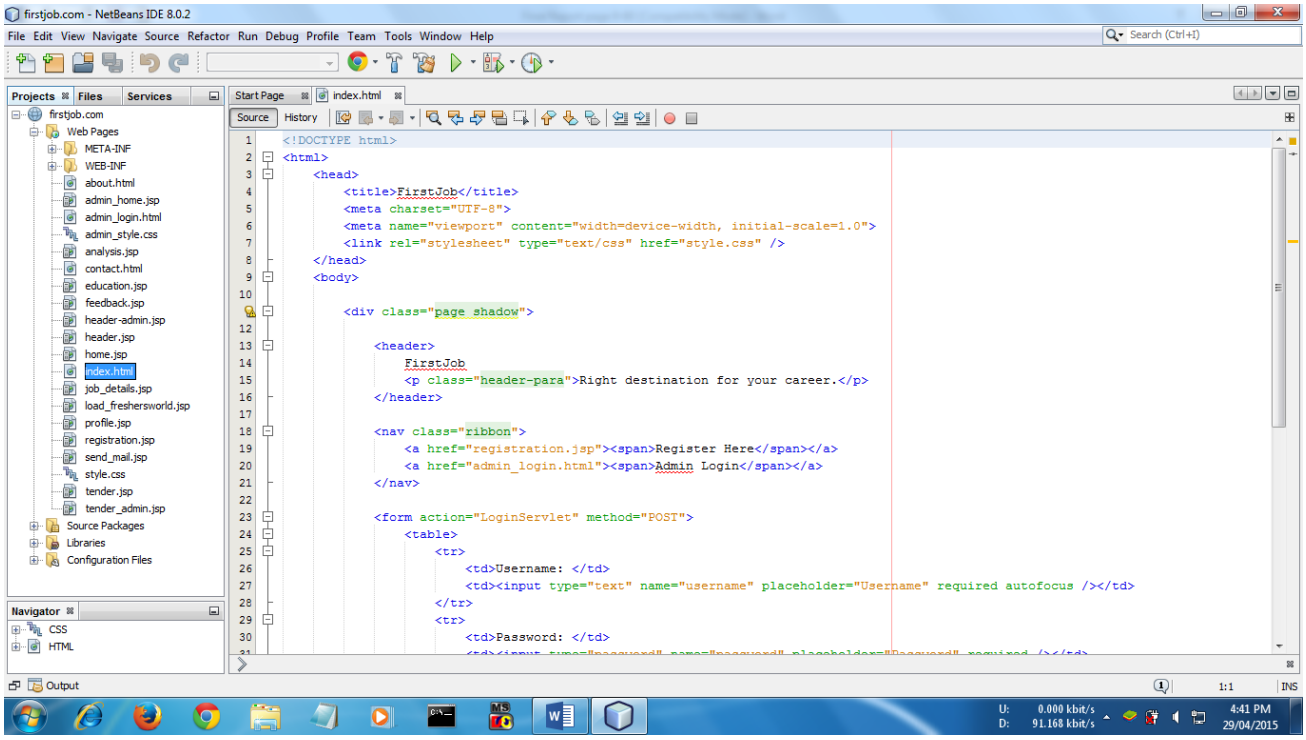


Fig 5.1: Main page

Screen-shot 1:

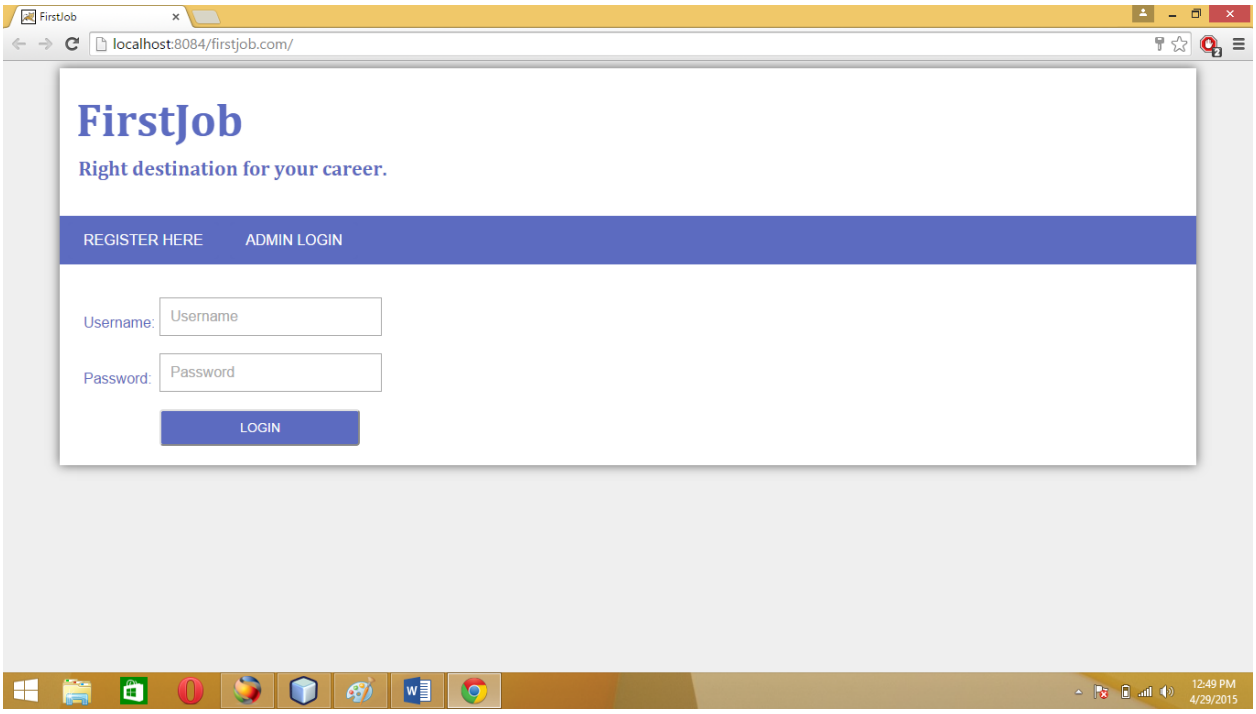


Fig5.2: Home Page

Source code:

Index.html

```
<! DOCTYPE html>
<Html>  <head>  <title>FirstJob</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<div class="page shadow">
<header>
FirstJob
<p class="header-para">Right destination for your career.</p>
</header>
<nav class="ribbon">
<a href="registration.jsp"><span>Register Here</span></a>
<a href="admin_login.html"><span>Admin Login</span></a>
</nav>
<form action="LoginServlet" method="POST">
<table>
<tr>
<td>Username: </td>
<td><input type="text" name="username" placeholder="Username" required
autofocus /></td>  </tr>
<tr>
<td>Password: </td>
<td><input type="password" name="password" placeholder="Password"
required /></td>
</tr>
<tr>  <td></td>
<td><input type="submit" value="LOGIN" /></td>
</tr>  </table>
</form> </div>
</body> </html>
```

```
<!DOCTYPE html>
<html>
<head>
<title>FirstJob</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>

<div class="page">

<header>
FirstJob
<p class="header-para">Right destination for your career.</p>
</header>

<nav class="ribbon">
<a href="registration.jsp"><span>Register Here</span></a>
<a href="index.html"><span>User Login</span></a>
</nav>

<form action="AdminLoginServlet" method="POST">
<table>
<tr>
<td>Username: </td>
<td><input type="text" name="username" placeholder="Username" required
autofocus /></td>
</tr>
<tr>
<td>Password: </td>
<td><input type="password" name="password" placeholder="Password" required
/></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="LOGIN" /></td>
</tr>
</table>
</form>

</div>
</body>
</html>
```

Screen-shot 2:

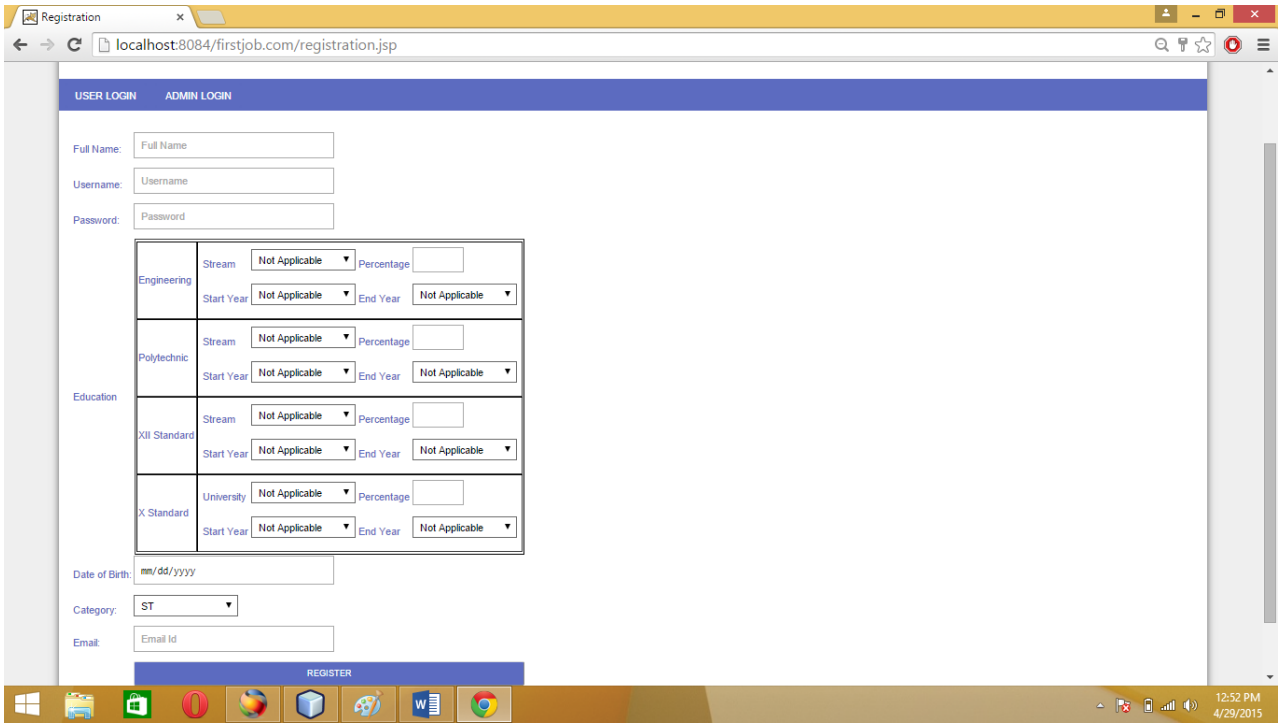


Fig 5.3: Registration page

Source code:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Registration</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>

<%
if (session != null) {
session.removeAttribute("username");
session.removeAttribute("user");
session.invalidate();
}
%>

<div class="page shadow">

<header>
FirstJob
<p class="header-para">Right destination for your career.</p>
</header>

<nav class="ribbon">
<a href="index.html"><span>User Login</span></a>
<a href="admin_login.html"><span>Admin Login</span></a>
</nav>

<form action="RegistrationServlet" method="POST">
<table>
<tr>
<td>Full Name:</td>
<td><input type="text" name="fullname" placeholder="Full Name" required autofocus
style="width: 300px"  /></td>
</tr>
<tr>
<td>Username:</td>
<td><input type="text" name="username" placeholder="Username" required
style="width: 300px" /></td>
</tr>
```

```
<td>Password:</td>
<td><input type="password" name="password" placeholder="Password" required
style="width: 300px" /></td>
</tr>
<tr>
<td>Education</td>
<td>
<jsp:include page="education.jsp" />
</td>
</tr>
<tr>
<td>Date of Birth:</td>
<td><input type="date" name="dob" placeholder="Date of Birth" required style="width:
300px" /></td>
</tr>
<tr class="category">
<td>Category:</td>
<td>
<select name="category">
<option value="0" selected>Open</option>
<option value="0" selected>OBC</option>
<option value="0" selected>SC</option>
<option value="0" selected>ST</option>
</select>
</td>
</tr>
<tr>
<td>Email:</td>
<td><input type="email" name="email" placeholder="Email Id" required style="width: 300px"
/></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="REGISTER" /></td>
</tr>
</table>
</form>

</div>
</body>
</html>
```

Screen-shot 3:

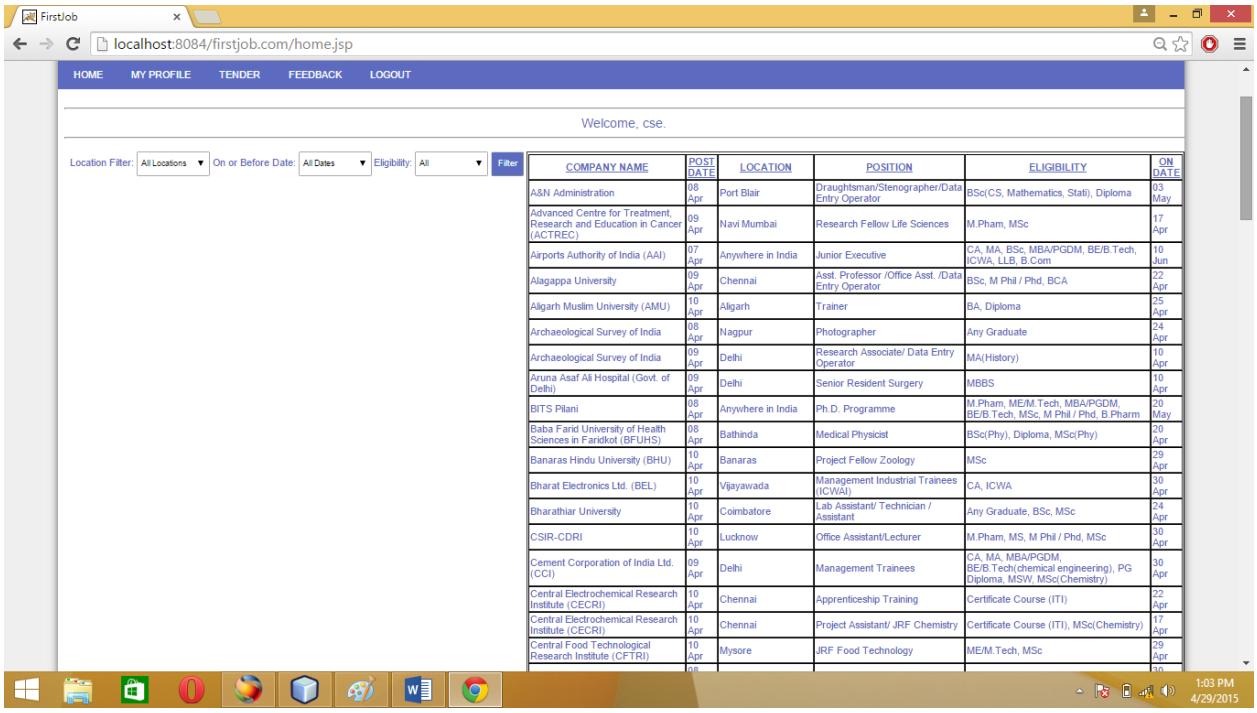


Fig 5.4: Job Details page

```
package crawler;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;


public class FreshersWorldJobDetails {

private final String link;
private String htmlCode = "";

public FreshersWorldJobDetails(String link) {
this.link = link;
}

public String getJobDetails() throws MalformedURLException, IOException {

// GET HTML CODE OF GIVEN LINK
URL url = new URL(link);
InputStream openStream = url.openStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(openStream));
String temp = reader.readLine();
while (temp != null) {
htmlCode = htmlCode + temp;
temp = reader.readLine();
}

// RETRIEVING JOB DETAILS PORTION FROM THE WEB PAGE
htmlCode = htmlCode.substring(htmlCode.indexOf("<div class=\"detail-container\">"),
htmlCode.indexOf("<span class=\"notice\">"));

// ADDING NATIVE STYLE
htmlCode = htmlCode.replace("<table ", "<table border='1' cellpadding='3' cellspacing='0'
style='border-color: #393939;' ");

// EXTRA FORMATTING STUFF
htmlCode = htmlCode.replace("<span>Apply Now</span>", "");
htmlCode = htmlCode.replace("<span style=\"color: #ffffff; font-family: verdana, geneva; font-
size: small;\">www.freshersworld.com</span>", "");
htmlCode = htmlCode.replace("<span style=\"color:
#ffffff;\">www.freshersworld.com</span>", "");
```

```
htmlCode = htmlCode.replace("<p class=\"company-weblink\">", "INFORMATION: <p class=\"company-weblink\">");
htmlCode = htmlCode.replace("<span class=\"text-right job_date\"><strong>Date of posting:</strong>", "</h2><h4><span class=\"text-right job_date\">Date of posting: ");
htmlCode = htmlCode.replace("</h2> <div id=\"job-specification\"", "</h4><div id=\"job-specification\"");

// FIXING RELATIVE URL'S
htmlCode = htmlCode.replace("<a href='/jobs", "<a href='http://www.freshersworld.com/jobs");
htmlCode = htmlCode.replace("<a href=\"/jobs", "<a href=\"http://www.freshersworld.com/jobs");

// REMOVING ADVERTISEMENT
String part1 = htmlCode.substring(0, htmlCode.indexOf("<script") - 1);
String part2 = htmlCode.substring(htmlCode.lastIndexOf("</script>") + "</script>".length());
htmlCode = part1 + part2;
return htmlCode;
}

}
```

Screen-shot 4:

Fig 5.5: Tender's page

Tender:

```
<%@page import="java.net.UnknownHostException"%>
<%@page import="java.net.ConnectException"%>
<%@page import="crawler.Tenders"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%!
String username = null;
%>
<!DOCTYPE html>
<html>
<head>
<title>FirstJob - Tenders</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<%
if (request.getSession(false) == null) {
response.sendRedirect("index.html");
}
username = (String) session.getAttribute("username");
if (username == null) {
response.sendRedirect("index.html");
}
%>
<div class="page">
<jsp:include page="header.jsp" />

<%
try {
out.println(new Tenders().getTable());
} catch (UnknownHostException ex) {
out.println("<hr><center><p>Please make sure you have good internet connection. "
+ "Refresh the page to reload information.</p></center><hr>");
} catch (ConnectException ex) {
out.println("<hr><center><p>Please make sure you have good internet connection. "
+ "Refresh the page to reload information.</p></center><hr>");
}
%>

</div>
</body>
</html>
```

Screen-shot 5:

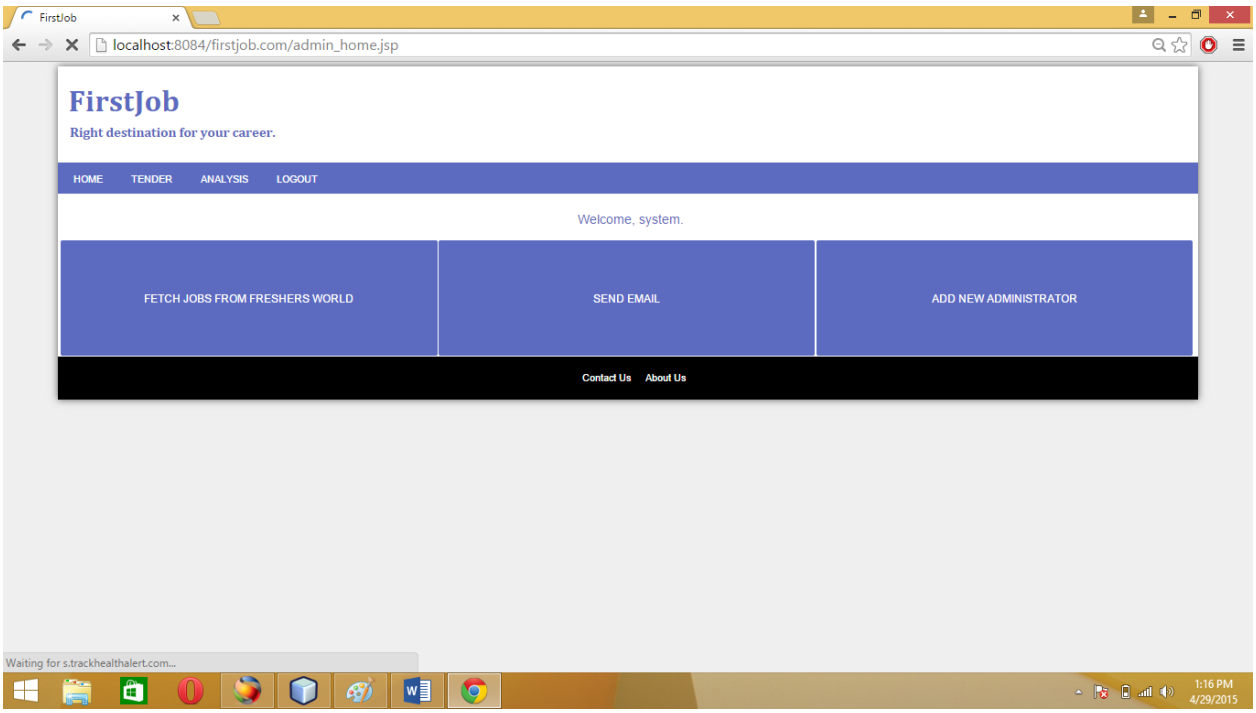


Fig 5.6: Admin Home page

```
public class FreshersWorld {

private static final String link = "http://www.freshersworld.com/jobs/category/Govt-Sector-job-vacancies";
private String htmlCode = "";
private final ArrayList<Job> jobList = new ArrayList<>();

public static void main(String[] args) throws IOException {
try {
FreshersWorld htmlParser = new FreshersWorld();
htmlParser.getFreshersWorldTable(null);
} catch (UnknownHostException unknownHostException) {
System.err.println("Unable to connect to " + link);

}
}
public String getFreshersWorldTable(User user) throws IOException {

// GET HTML CODE OF GIVEN LINK
URL url = new URL(link);
InputStream openStream = url.openStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(openStream));
String temp = reader.readLine();
while (temp != null) {
htmlCode = htmlCode + temp;
temp = reader.readLine();
}
// RETRIEVING THE TABLE PORTION FROM THE WEB PAGE
htmlCode = htmlCode.substring(htmlCode.indexOf("<table class=\"jobs_details table table-condensed search\">"));
htmlCode = htmlCode.substring(0, htmlCode.indexOf("</table>") + "</table>".length());

// ADDING NATIVE STYLE
htmlCode = htmlCode.replace("<table class=\"jobs_details table table-condensed search\">", "<table border='1' cellpadding='3' cellspacing='0' style='border-color: #393939;' class='jobs_details table table-condensed search'>");

// FIXING RELATIVE URL'S
htmlCode = htmlCode.replace("<a href='/jobs", "<a href='http://www.freshersworld.com/jobs");
htmlCode = htmlCode.replace("<a href=\"/jobs", "<a href=\"http://www.freshersworld.com/jobs");
```

```
// REMOVING NEXT PAGE DETAILS
try {
String nextPage = htmlCode.substring(htmlCode.indexOf("<div id=\"paginationStrip\">",
htmlCode.lastIndexOf("</div>")));
htmlCode = htmlCode.replace(nextPage, "");
} catch (StringIndexOutOfBoundsException ex) {
// NO NEXT PAGE DETAILS
}
htmlCode = htmlCode + "";
// FILTER IRRELEVANT JOB DETAILS
if (user != null) {
htmlCode = filterIrrelevantJobs(htmlCode, user);
} else {
filterEachJob(htmlCode);
}

try {
Database database = new Database();
database.connect();
database.addJobs(jobList);
database.disconnect();
} catch (SQLException | ClassNotFoundException ex) {
Logger.getLogger(FreshersWorld.class.getName()).log(Level.SEVERE, null, ex);
}
// RETURN FILTERED JOB DETAILS
return htmlCode;
}

public void filterEachJob(String htmlCode) {

// FIRST ROW OF TABLE IS HEADER ROW. REMOVE HEADER ROW SO THAT
FILTERING JOBS CAN BE STARTED.
String finalCode = htmlCode.substring(0, htmlCode.indexOf("</tr>") + "</tr>".length());
htmlCode = htmlCode.replace(finalCode, "");
// FILTERING EACH JOB DETAIL
String filter = htmlCode.substring(htmlCode.indexOf("<tr>"), htmlCode.indexOf("</tr>")
+ "</tr>".length());

while (filter != null) {
getJobAttributes(filter);
htmlCode = htmlCode.replace(filter, "");
try {
filter = htmlCode.substring(htmlCode.indexOf("<tr>"), htmlCode.indexOf("</tr>") +
"</tr>".length());
} catch (StringIndexOutOfBoundsException e) {
```

```
private String filterIrrelevantJobs(String htmlCode, User user) {

// FIRST ROW OF TABLE IS HEADER ROW. REMOVE HEADER ROW SO THAT FILTERING JOBS CAN BE STARTED.
String finalCode = htmlCode.substring(0, htmlCode.indexOf("</tr>") + "</tr>".length());
htmlCode = htmlCode.replace(finalCode, "");

// FILTERING IRRELEVANT JOB DETAILS
String filter = htmlCode.substring(htmlCode.indexOf("<tr>"), htmlCode.indexOf("</tr>") + "</tr>".length());

while (filter != null) {

boolean keepInfo = false;
System.out.println("\nFETCHED:" + filter);

try {
String eligibility = filter.substring(filter.indexOf("<td class=\"eligibility\">"));
eligibility = eligibility.substring(0, eligibility.indexOf("</td>") + "</td>".length());
if (user != null) {
if ((eligibility.contains("Diploma") || (eligibility.contains("Polytechnic"))) && (!user.getPolyStream().equals("Not Applicable"))) {
keepInfo = true;
}
if (((eligibility.contains("BE")) || (eligibility.contains("BE/B.Tech")) || (eligibility.contains("B.Tech"))) && (!user.getBeStream().equals("Not Applicable"))) {
keepInfo = true;
}}
getJobAttributes(filter);
if (keepInfo) {
finalCode += filter;
System.out.println("Status: Added");
} else {
System.out.println("Status: Removed");
}
} catch (StringIndexOutOfBoundsException ex) {
System.out.println("Status: Irrelevant");
}
htmlCode = htmlCode.replace(filter, "");
try {
filter = htmlCode.substring(htmlCode.indexOf("<tr>"), htmlCode.indexOf("</tr>") + "</tr>".length());} catch (StringIndexOutOfBoundsException e) {
break;
}}
```

```
if (finalCode.equals("<table border='1' cellpadding='3' cellspacing='0' style='border-color: #393939;' class='jobs_details table table-condensed search'><tr><th style=\"width: 160px;\">Company</th><th>Location</th><th>Position</th><thstyle=\"width: 220px;\">Eligibility</th><th style=\"width: 92px;\">Last Date</th></tr>")) {
finalCode = "<h3>Currently no jobs available.</h3>";
}

// RETURN FILTERED JOB DETAILS AND REMAINING HTML CODE
return finalCode + htmlCode;
}

private void getJobAttributes(String row) {
try {
row = row.substring(row.indexOf("<a href="));
String companyName = row.substring(row.indexOf(">") + 1, row.indexOf("</a>"));
row = row.substring(row.indexOf("<span class=\"post_date\">") + "<span class=\"post_date\">".length());
String postDate = row.substring(0, row.indexOf("</span>"));

row = row.substring(row.indexOf("<span class=\"location_name\"> <a href=") + 1 + "<span class=\"location_name\"> <a href=".length());
String location = row.substring(row.indexOf(">") + 1, row.indexOf("</a>"));

row = row.substring(row.indexOf("<td>") + "<td>".length());
String position = row.substring(0, row.indexOf("</td>"));

row = row.substring(row.indexOf("<td class=\"eligibility\">") + "<td class=\"eligibility\">".length());
String eligibility = row.substring(0, row.indexOf("</td>"));
eligibility = eligibility.replace("<span style='font-weight:normal'>", "");
eligibility = eligibility.replace("</span>", "");
row = row.substring(row.indexOf("<td class=\"date\">") + "<td class=\"date\">".length());
String date = row.substring(0, row.indexOf("</td>"));

System.out.println("DATA: " + companyName + "\t" + postDate + "\t" + location + "\t" + position + "\t" + eligibility + "\t" + date);

jobList.add(new Job(companyName, postDate, location, position, eligibility, date));

} catch (Exception exception) {
System.out.println("CANNOT PARSE: " + row);
}
}
}
```

Screen-shot 6:

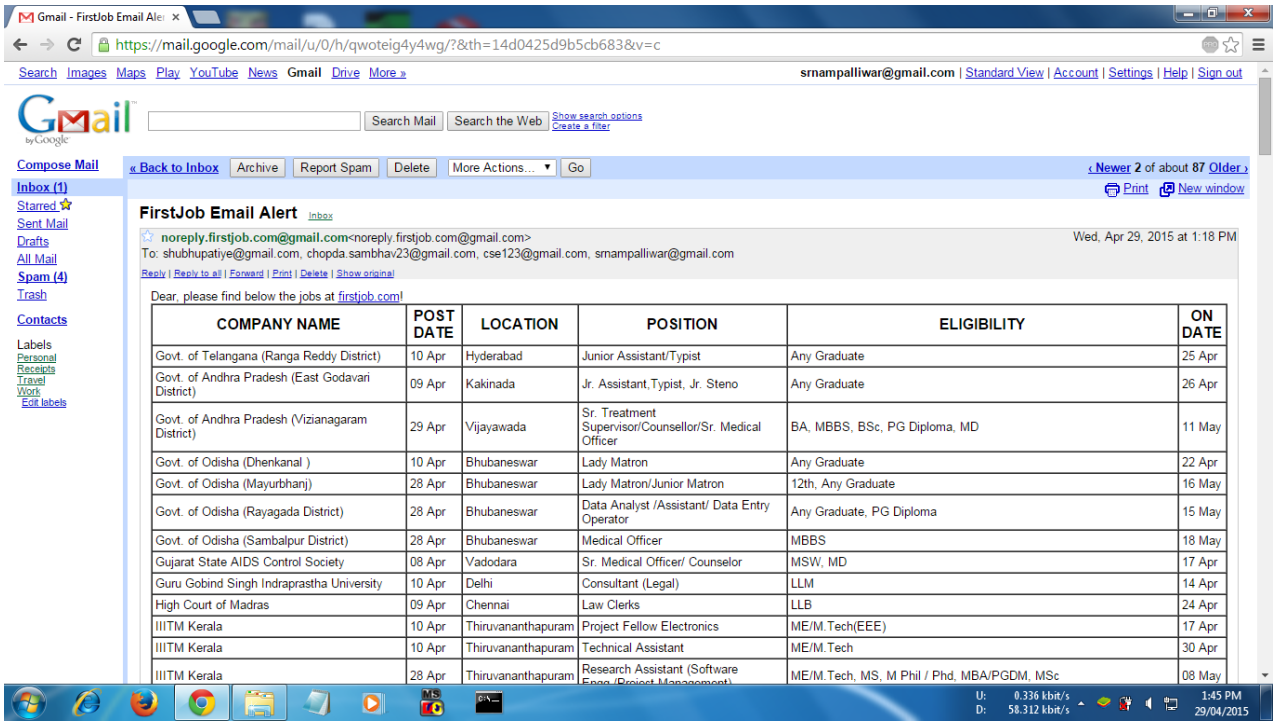


Fig 5.7: E-mail

```
<%@page import="email.EmailService"%>
<%@page import="java.util.ArrayList"%>
<%@page import="controller.Database"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%!
String username = null;
Database database = new Database();
%>
<!DOCTYPE html>
<html>
<head>
<title>FirstJob</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="style.css" />
<link rel="stylesheet" type="text/css" href="admin_style.css" />
</head>
<body>
<%
if (request.getSession(false) == null) {
response.sendRedirect("index.html");
}
username = (String) session.getAttribute("username");
if (username == null) {
response.sendRedirect("index.html");
}
%>
<div class="page">
<jsp:include page="header-admin.jsp" />
<%
try {
out.println("<hr><center>Sending Email...</center>");
database.connect();
ArrayList<String> recipients = database.getAllUserEmailId();
new EmailService().sendMessage(recipients);
database.disconnect();
out.println("<hr><center>Successfully Sent Emails!</center><hr>");
} catch (Exception ex) {
out.println("<hr><center><p>Please make sure you have good internet connection. "
+ "Refresh the page to reload information.</p></center><hr>");
}
%>
</div>
</body>
</html>
```

Chapter 6
Conclusion
&
Future Work

6.1 Conclusion

India is second largest populated country in the world and also has the largest percentage of unemployment. To help the country by reducing unemployment rate was the main objective of our project.

The government job portal has been designed successfully. Here we provide platform for employer and employee interaction. All the jobs are matched with the profile of user and then informed to seeker with the help of an email. The user can also filter jobs on the basis of location and also for specific profiles. The usefulness of a job portal depends upon the number of users getting benefited. The government Jobs portal provides user with the best available jobs according to user's highest qualification to his/her lowest qualification.

Lot of advancements in job portal is observed in last decade. Several Data source and computing related bottlenecks still exist. We have addressed many challenges and recent research areas in the field of data mining. The existing job portals, where provide a capital subscription, the Government jobs portal is free of cost. The other job portals does not provide relevant information based on qualification and also unable to provide the functionality of filtering the jobs based on multiple criteria of location, interest and date simultaneously, whereas the Government jobs portal provides all these functionality under one roof which makes our portal more efficient that existing ones.

Each module of the project has been successfully test, but after comparing this portal with the various portal available online, we have addresses many weakness of these online portals. Thus with all these functionality, the Government Jobs Portal would prove to be the one of the best portal in employment generation.

6.2 Future Work

For future development, we will first consider all previously assumed options which are not yet developed. This we can make the website livelier in action, so that can be considered as popular online job portal website.

At this some future development has to be taken into account, such as…..

- For analysis purpose, we can consider more clauses and use sentiment analysis as well as NLP to draw more useful conclusions.

- Resume submission, Search Resumes and applying Language processing on Resumes for filtering competent users.

We should look for the whole prospective in the near future for a successful, most popular and user-friendly website which will be frequently updated with some extra new features as well as attractive altering ornamentations in the user interface.

Chapter 7
References

7.0 References

[1] Divya Chaudhary, "Data Mining: Techniques and Algorithms", Volume 3, Issue 8, August 2013

[2] Pedro Domingos and Geoff Hulten, "Mining High-Speed Data Streams", KDD 2000.

[3] M. Hernandez and S. Stolfo, "Real-World Data is Dirty: Data Cleansing and The Merge/Purge Problem", *Data Mining and Knowledge Discovery*, Volume 2, Issue 1, 1998, 9-37.

[4] Chun-Nan Hsu and Graig A. Knoblock, "Discovering Robust Knowledge from Databases that Change", *Data Mining and Knowledge Discovery*, Volume 2, Issue 1, 1998, 69-95.

[5] Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In Proceedings of the Seventh international Conference on World Wide Web (WWW-7) (Brisbane, Australia). P. H. Enslow and A. Ellis, Eds. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 107-117.

[6] www.tutorialspoint.com/java/java_sending_email.htm

[7] http://stackoverflow.com/questions/28569865/extracting-url-parameters-in-java-servlet-from-a-request-placed-by-action-script

[8] http://stackoverflow.com/questions/5375028/extraction-of-html-tags-using-java

[9] https://www.connectionstrings.com/sql-server/

Books:

[1] Jason Hunter, William Crawford," Java Servlet Programming", 2nd Edition by O'Reilly Media Publications.

[2] Joel Murach and Andrea Steelman, "Murach's Java Servlets and JSP" 2nd Edition, January 2008.