

STUDENT PLACEMENT RELATIONSHIP MANAGEMENT

A PROJECT REPORT

Submitted by

AISHWARYA T (1413003)

KARTHIKEYAN V (1413044)

KOWSIC RAJ V (1413051)

PRAVEENRAJ M S (1413075)

in partial fulfillment for the requirement of award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



K.S.R. COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University Chennai and Approved by AICTE)



TIRUCHENGODE

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2018

K.S.R. COLLEGE OF ENGINEERING (AUTONOMOUS)

TIRUCHENGODE – 637 215

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this project report “**STUDENT PLACEMENT RELATIONSHIP MANAGEMENT**” is the bonafide work of “**AISHWARYA (1413003), KARTHIKEYAN V (1413044), KOWSIC RAJ V (1413051) and PRAVEENRAJ M S (1413075)**” who carried out the project work under my supervision.

Signature

**Dr. A. RAJIV KANNAN M.E., Ph.D.,
HEAD OF THE DEPARTMENT**

Professor

Department of CSE

K.S.R College of Engineering,
(Autonomous)

Tiruchengode - 637215.

Signature

**Dr. E. BABY ANITHA M.E., Ph.D.,
SUPERVISOR**

Assistant Professor

Department of CSE

K.S.R College of Engineering,
(Autonomous)

Tiruchengode - 637215.

Submitted for Project viva-voce held on

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We feel highly honored to extend our sincere gratitude to our beloved Founder cum Chairman, **Lion Dr. K.S. RANGASAMY, MJF, K.S.R** Educational Institutions and our Chairman cum Managing Trustee **Mr. R. SRINIVASAN, BBM., MISTE.,** for providing all facilities to complete this project work.

We would like to acknowledge the constant and kind support provided by our Principal **Dr. P. SENTHIL KUMAR, M.E., Ph.D.,(IITM),** who supported us in all the endeavors and been responsible for inculcating us all through our career.

We act sense of gratitude to **Dr. A. MAHABUB BASHA, M.E., Ph.D.,** Director of Computer Science and Engineering department for offering suggestions at the right time.

We feel highly elated to thank our respectable Head of the Department, **Dr. A. RAJIV KANNAN, M.E., Ph.D., MCSI., MISTE.,** who guided us and was a pillar of support for the successful completion of the project.

We are great indebted to our Project Coordinator **Mr. K. KUMARESAN, M.E., (Ph.D.),** of our department for his valuable suggestions and guidance to our project.

We are the most fortunate in having the opportunity to work under the internal guide **Dr. E. BABY ANITHA, M.E., Ph.D.,** express our sincere thanks to her. This project has brought out the hidden talents within us. It is a pleasure to express our gratefulness to our beloved parents for providing their support and confidence to us for the completion of the project and our heartfelt thanks to our entire department faculty members, beloved friends, directly and indirectly who helped us during the tenure of the project.

ABSTRACT

The project entitled student placement relationship management is a web based application development system using integrated development environment. The main objective of this web application is to decrease the paper work and easier record maintenance by having a particular website for placement process. In this project mainly based on Customer Relationship Management (CRM). The student placement relationship management is a Job Portal developed for creating an interactive job vacancy for candidates. The web application is to be conceived in its current form as a dynamic site-requiring constant updates both from the seekers as well as the companies. It enables jobseekers to post their resume, search for jobs and view personal job listings. It will provide various companies to place their vacancy profile on the site and also have an option to search candidate resumes. Based on the criteria given by the company, job offers are sent to the job seekers.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	IV
	LIST OF FIGURES	VIII
	LIST OF TABLES	IX
	LIST OF ABBREVIATIONS	X
1	INTRODUCTION	1
	1.1 FUNCTIONAL REQUIREMENTS	2
	1.2 NON-FUNCTIONAL REQUIREMENTS	2
2	PROBLEM DESCRIPTION	3
	2.1 EXISTING SYSTEM	3
	2.2 DRAWBACKS OF EXISTING SYSTEM	3
	2.3 PROPOSED SYSTEM	4
	2.4 ADVANTAGES OF PROPOSED SYSTEM	4
3	SYSTEM SPECIFICATION	5
	3.1 HARDWARE SPECIFICATION	5
	3.2 SOFTWARE SPECIFICATION	5
4	SOFTWARE SPECIFICATION	6
	4.1 FRONT END	6
	4.1.1 HTML	6
	4.1.2 CSS	7
	4.1.3 Java script	7
	4.2 BACK END	8
	4.2.1 J2EE	8
	4.2.2 SQL	9

5	SYSTEM ANALYSIS	10
5.1	REQUIREMENT ANALYSIS	10
5.2	DATA COMMUNICATION	10
6	MODULE DESCRIPTION	11
6.1	ADMIN	11
6.1.1	Add new user	11
6.1.2	Update existing user	11
6.1.3	Delete existing user	11
6.1.4	Change password	11
6.1.5	Send notification	12
6.1.6	Logout	12
6.2	COMPANY	12
6.2.1	Post jobs	12
6.2.2	View student record	12
6.2.3	Logout	12
6.3	COLLEGE	12
6.3.1	View record	12
6.3.2	Update details	12
6.3.3	Logout	13
6.4	STUDENT	13
6.4.1	Update resume	13
6.4.2	View job record	13
6.4.3	Logout	13
7	DESIGN	14
7.1	USE CASE DIAGRAM	14
7.2	SEQUENCE DIAGRAM	15

	7.3 DATAFLOW DIAGRAM	16
8	TESTING	17
	8.1 UNIT TESTING	17
	8.2 INTEGRATION TESTING	17
	8.3 SYSTEM TESTING	17
	8.3.1 White box testing	18
	8.3.2 Black box testing	18
	8.4 SYSTEM IMPLEMENTATION	18
9	CONCLUSION AND FUTURE ENHANCEMENTS	19
	9.1 CONCLUSION	19
	9.2 FUTURE ENHANCEMENT	19
10	APPENDIX	20
	10.1 SOURCE CODE	20
	10.2 DATABASE TABLES	30
	10.3 SCREENSHOTS	33
11	REFERENCES	37

LIST OF FIGURES

Figure No.	Name of the Figure	Page No.
7.1	Use Case Diagram	14
7.2	Sequence Diagram	15
7.3	Dataflow Diagram	16
10.1	Login page	33
10.2	Student Registration	33
10.3	Student Home Page	34
10.4	Student Personal Information	34
10.5	Document upload	35
10.6	College profile	35
10.7	Job Posting	36
10.8	Add company or college	36

LIST OF TABLES

Table No.	Title of the Table	Page No.
10.1	Student Details	30
10.2	Academic Details	30
10.3	Company Details	31
10.4	College Details	31
10.5	Graduation Details	32
10.6	Document upload	32

LIST OF ABBREVIATIONS

ACRONYM	ABBREVIATION
CSS	Cascading Style Sheet
GB	Giga Bytes
HD	Hard Disk
HTML	Hyper Text Markup Language
OS	Operating System
RAM	Random Access Memory
J2EE	Java Enterprise Edition
JDBC	Java Database Connectivity
ODBC	Open Database Connectivity
EJB	Enterprise Java Bean
SQL	Structured Query Language
API	Application Program Interface
CRM	Customer Relationship Management

CHAPTER 1

INTRODUCTION

The project entitled **STUDENT PLACEMENT RELATIONSHIP MANAGEMENT** is a web based application development system using integrated development environment. We planned to develop this application to decrease the paper work and easier record maintenance by having a particular website for placement process. Customer Relationship Management (CRM) is an approach to manage a company's interaction with current and potential customers. This web application is developed for managing interaction between jobseekers, college and company. Here jobseekers are students from various colleges, institutes or professionals. Apart from this there will be an admin for the application to make changes to the database content. This application can be used to automate the workflow of the job seekers and company. This project's main idea is to develop a centralized application connected to database which will maintain placement details. Student placement relationship management application will reduce paperwork and maintain record in a more efficient and systematic way.

1.1 FUNCTIONAL REQUIREMENTS

Admin

- Create, update and delete user details after login
- Can change login password
- View student status.
- Send notifications to students.
- Logout from the system

Company

- To post jobs.
- To view the student records.
- Logout from the system.

College

- To view records.
- To update details.
- Logout from the system.

Student

- To post their resume
- To apply for jobs.
- To view the applied records.
- Logout from the system.

1.2 NON –FUNCTIONAL REQUIREMENTS

- Secure access of confidential data.
- 24x7 availability.
- Browse testing and support for Mozilla Firefox and Google Chrome.

CHAPTER 2

PROBLEM DESCRIPTION

2.1 EXISTING SYSTEM

- Before creating this website, all jobseekers to send their resumes or information through postal mails or they use person to person contacts with each other. It will take long time to send their requirements through this type of communications.
- Here there May error occurs in the process. The administration faces the problems to collect all the information from clients and consultants to analyze the requirement in the corresponding clients. Administration has to send requirements information to different consultants and jobseekers.
- These records are entered in manual process will take long time, separate workers need to maintaining the databases. All the details are stored via separate databases.

2.2 DRAWBACKS OF EXISTING SYSTEM

- Duplication of jobs
- Time consumption is high.
- Difficult to data maintenance and search.
- Data can be loss.
- No proper feedback or intimations
- Takes more time to apply a job

- Heavy work load because maintained in the form of files or records.

To avoid all these limitation and make the working more accurately the system need to be computerized.

2.3 PROPOSED SYSTEM

- The main objective of the proposed system is to provide a user-friendly interface.
- Here all job seekers send their resumes or information through our site. It does not consume much of time. It is very easier to modify if any error occurs in the process. It is also very easier to administrator to collect information from clients and consultants.
- Proper intimations to job seekers through Emails is possible.
- Once the details are fed into the computer there is no need for various persons to deal with separate sections.
- Only a single person is enough to maintain all the reports.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- Large volume of data can be stored with ease.
- Maintenance of record is flexible.
- Records stored are updated now and then.
- Stored data and procedures can be easily edited.
- Reports can be generated with ease.
- Accurate calculations are made and less man power required.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATION

PROCESSOR	:	Dual core and above
PROCESSOR SPEED	:	2.80 GHz
RAM	:	2 GB or more
HARD DISK	:	20 GB or more

3.2 SOFTWARE SPECIFICATION

TARGETED OS	:	Windows 7/8/8.1/10
FRONT END	:	HTML, CSS, JS
BACK END	:	J2EE, SQL
WEB SERVER	:	Tomcat, SMTP, XAMPP
BROWSER	:	Chrome, Mozilla, IE
IDE	:	Eclipse

CHAPTER 4

SOFTWARE SPECIFICATION

4.1 FRONT END

4.1.1 HTML

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as “heading”, “paragraph”, “table”, and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

Hyper Text: Hyper Text simply means “Text within Text”. A text has a link within it, is a hypertext. Every time when you click on a word which brings you to a new webpage, you have clicked on a hypertext.

Markup language: A markup language is a programming language that is used make text more interactive and dynamic. It can turn a text into images, tables, links etc.

An HTML document is made of many HTML tags and each HTML tag contains different content.

4.1.2 CSS

- CSS stands for **Cascading Style Sheets**
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External style sheets are stored in **CSS files**

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs and variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

4.1.3 JAVASCRIPT

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming style. It has an API for working with text, array, dates, regular expression, and basic manipulation of the DOM, but does not include any I/O such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and database, and non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop application, including desktop widgets.

4.2 BACK END

4.2.1 J2EE

- J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online.
- The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitier, Web-based applications.
- At the client tier, J2EE supports pure HTML, as well as Java applets or applications. It relies on Java Server Pages and servlet code to create HTML or other formatted data for the client.
- Enterprise JavaBeans (EJBs) provide another layer where the platform's logic is stored. An EJB server provides functions such as threading, concurrency, security and memory management. These services are transparent to the author.
- Java Database Connectivity (JDBC), which is the Java equivalent to ODBC, is the standard interface for Java databases.
- The Java servlet API enhances consistency for developers without requiring a graphical user interface.

4.2.2 SQL

SQL consists of a data definition language, data manipulation language and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data control. SQL is a standard language for storing, manipulating and retrieving data in databases.

CHAPTER 5

SYSTEM ANALYSIS

5.1 REQUIREMENT ANALYSIS

Since the design of a new or revised system cannot begin until the analyst fully understands the existing system, this stage cannot be omitted in software development. Hence we are going to start with the analysis of the existing manual system under the following heading: data collection, data storage, data communication and manipulation and system cost.

Figure 10.8 DATA COMMUNICATION

The process like – sorting, comparing, analysis and calculation are to be performed on the records which are stored on files. Difficulties arise when it actually comes to manipulation of data. Processing data manually consumes a lot of time; for instance, sorting of folders into years of admission. Hence the access time or process time of the present system is extremely high and results in staffs spending more time than is necessary.

Now, with the availability of high speed modern electronic information processing machines, this need for improving the speed of transaction processing could be easily accomplished.

CHAPTER 6

MODULE DESCRIPTION

6.1 ADMIN

In Admin module, users logged into the system are capable of doing the following operations. Here the admin is responsible for analyse, insert, update, delete and view the details of the other users. He/she can change the details of the other users.

6.1.1 Add new user

Admin can add new company or college by giving separate id and name as input. The default password on creation is “KSRCE” for all created new user. Here, the new company and college are added. The admin only have rights to do this for a security purpose.

6.1.2 Update existing user

Admin can update existing student or company or college details by giving their id. Using this module admin can update the basic details.

6.1.3 Delete existing user

Admin can delete student or company or college name or details from the system if they are not need. All details of student or company or college are deleted permanently from the system. i.e., if anyone has relieved their details will be erased.

6.1.4 Change Password

Admin can change login password using this module. So if the password is forgotten by the student or company or college can be changed by admin.

6.1.5 Send notification

Admin can send notifications to the shortlisted students or job seekers via email system.

6.1.5 Logout

Admin should logout securely from the system.

6.2 COMPANY

In company module, users logged into the system are capable of doing the following operations. Here the company member is responsible for posting jobs and view the details of the student in the system.

6.2.1 Post jobs

Company can post jobs by filling the form as input. The records can be viewed by other users in the system.

6.2.2 View student record

Company can view all records of the students by giving separate id or name or skills as input or the records can be searched through other user details.

6.2.3 Logout

Company should logout securely from the system.

6.3 COLLEGE

6.3.1 View record

College can view all records of the students and jobs in the system.

6.3.2 Update details

College can update their details like name, location and other details.

6.3.3 Logout

College should logout securely from the system.

6.3 STUDENT

In Student module, users logged into the system are capable of doing the following operations. Here the student can post jobs and can view the details of the applied jobs and search for jobs.

6.3.1 Update resume

Students can update their resume through designed forms by giving the proper details.

6.3.2 View job record

Students can view the jobs applied by company and apply for those jobs. This is also used get notifications if their resume is selected by the company.

6.3.3 Logout

Student should logout securely from the system.

CHAPTER 7

DESIGN

7.1 USE CASE DIAGRAM

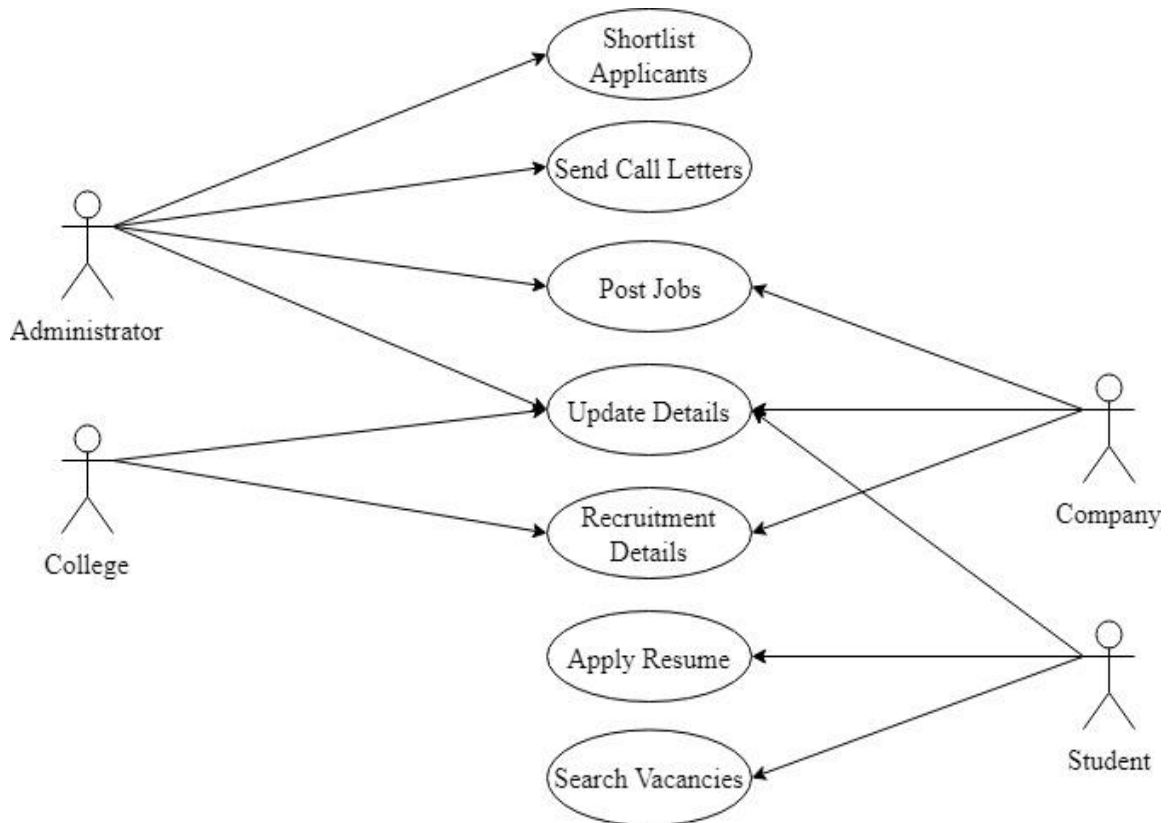


Figure 7.1 Use Case Diagram

In this management system, the actors are student, college, company and administrator. The detailed description of use case is represented in the use case diagram.

7.2 SEQUENCE DIAGRAM

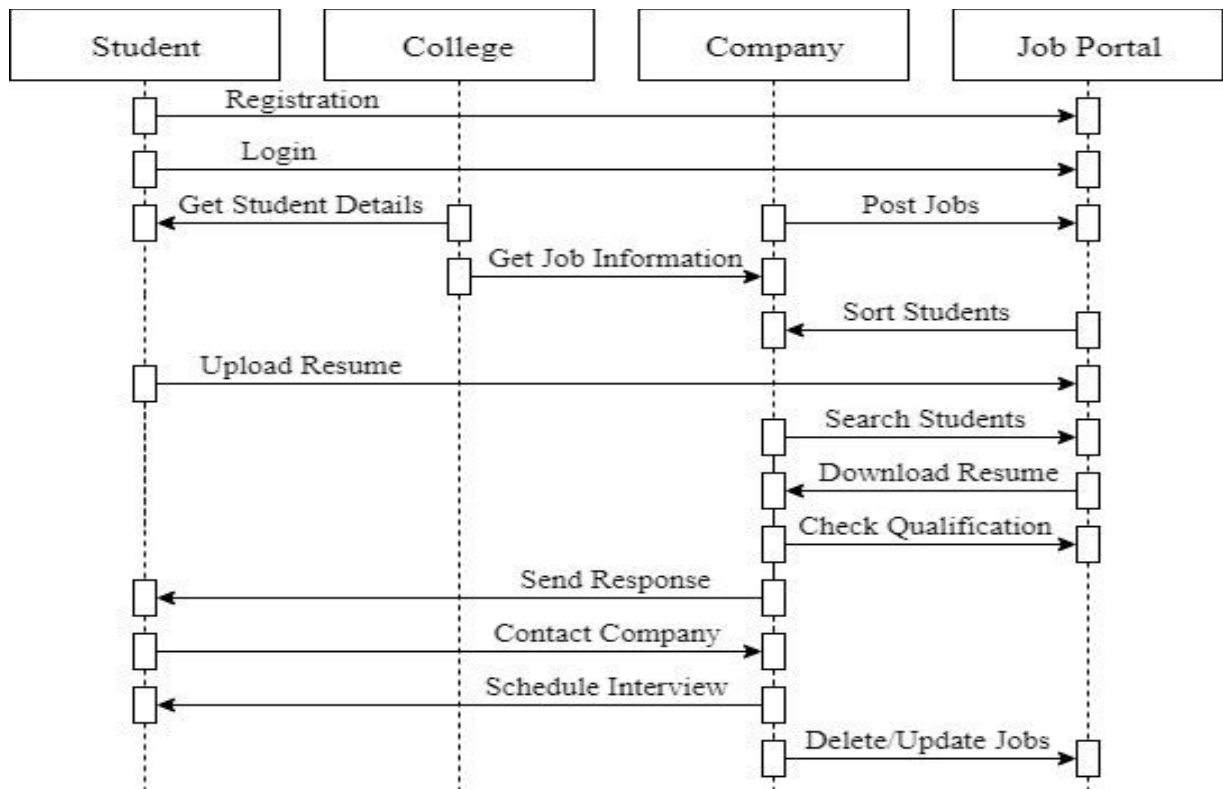


Figure 7.2 Sequence Diagram

Sequence diagram are easy and intuitive way of describing the behavior of a system by viewing the interaction between the system and its environment. A sequence diagram shows an interaction arranged in a time sequence.

7.3 DATAFLOW DIAGRAM

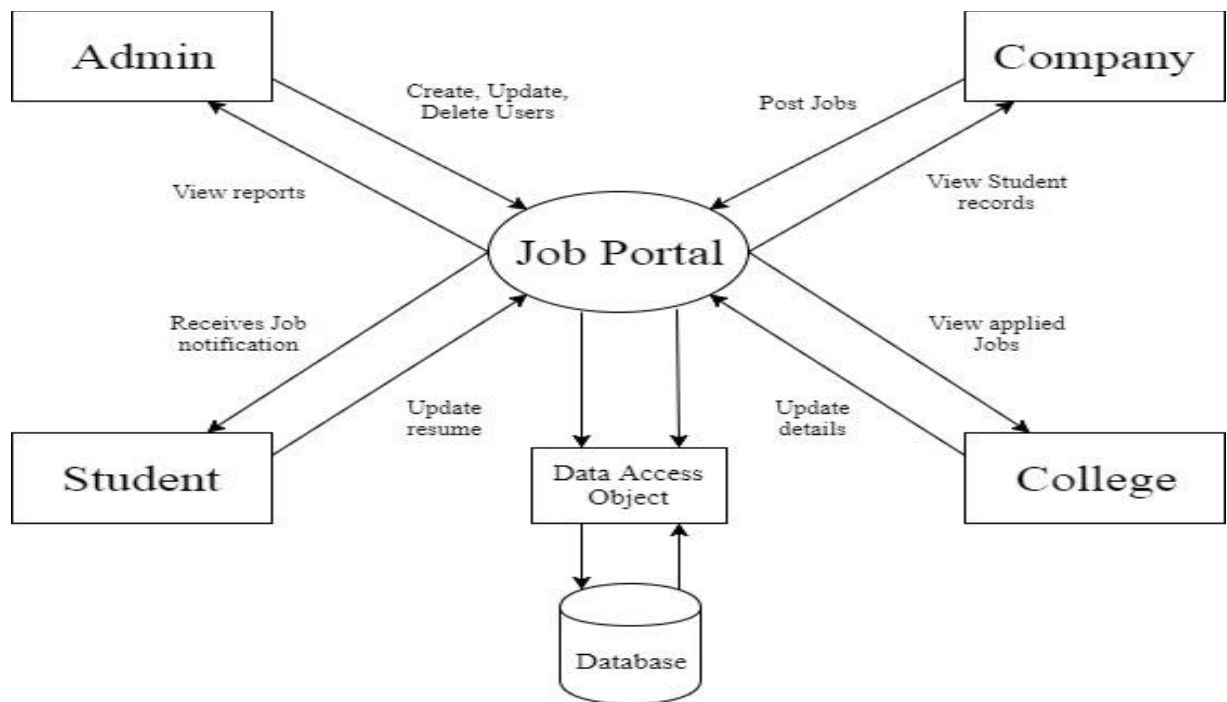


Figure 7.3 Dataflow Diagram

Dataflow diagrams maps out the flow of information for process or system. The above dataflow diagram describes the data flow through various modules.

CHAPTER 8

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. There are various types of test. Each test type addresses a specific testing requirement.

8.1 UNIT TESTING

In unit testing, we have to test the programs making up the system. For this reason, Unit testing sometimes called as Program testing. Unit testing on the modules are independently of one another, to locate errors. This enables to detect errors in coding and logic in the module. The testing was carried out during programming stage itself.

8.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

8.3 SYSTEM TESTING

System testing ensures that the entire integrated software system meets the requirements. It tests a configuration to ensure known and predictable results. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. In this testing it is based on the

coding to assign or performs the function by using the methods and data for the program to be run.

8.3.1 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner coding, structure and language of the software.

8.3.2 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box must be written from a definitive source document, such as specification or requirements document. The test provides inputs and responds to outputs without considering how the software works.

8.4 SYSTEM IMPLEMENTATION

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. The aim of the system illustration was to identify any malfunction of the system. After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 CONCLUSION

The project titled as “Student Placement Relationship Management” is a web based application. This software provides facility for create, update and delete staff details after login. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus, the system has fulfilled all the objectives identified and is able to replace the existing system.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software.

9.2 FUTURE ENHANCEMENTS

In future, we can use photo reorganization instead of using heterogeneous database more over high speed, accuracy and non-redundant data are the main advantages of the system. Data entry errors can be minimized through validity checks. After the verification only, the data are placed in the permanent database.

CHAPTER 10

APPENDIX

10.1 SOURCE CODE

LOGIN PAGE

```
<!DOCTYPE html>

<html>

<head>

<title>Main Page</title>

<link href="_css/Frontpage.css" rel="stylesheet" type="text/css" />

<link href="_css/FrontPageFooter.css" rel="stylesheet" type="text/css" />

<script type="text/javascript">

    function initAll(){

        window.location="Student/SignUp.jsp?content=none";

    }

</script>

</head>

<body >

    <div class ="myheader">

        <div class="logo" > <h1>

            <a href="index.jsp">Job Portal<br />

            <small >where you get your dream job...</small></a>

            </h1>

        </div>

        <pre class="menu"> <h1> <a href="index.jsp" >Home Page</a> <a
href="Student/SignUp.jsp?content=none" >Sign Up</a> <a href="index.jsp" >About Us</a> <a
href="index.jsp" >Contact Us</a> </h1></pre>

    </div>

    <div class="mainContent">

        <div class="divtableclass">

            <table align="center" class="tb">
```

```

<tr> <td>

<pre>  <form id="login" action="CheckId" method="POST" >

Login as: <input type="radio" name="grp1" value="Student" checked="checked"> Student <input type="radio"
name="grp1" value="College">College<input type="radio" name="grp1" value="Company">Company

Registration ID: <input type="text" name="regId" >

Password:      <input type="password" name="password">

<input type="Submit" value="Sign In" class="signup_btn" >      <input type="button" value="Student Sign
Up" name="sreg" class="signup_btn" onclick="initAll()"/>

</form> </pre></td>

</tr>

</table>

</div> </div>

<div class ="footer"> </div>

</body>

</html>

```

Login Servlet

```
@WebServlet(name="CheckId", urlPatterns={"/CheckId"})

public class CheckId extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html;charset=UTF-8");

        PrintWriter out = response.getWriter();

        try
        {

            String regId=request.getParameter("regId");

            String pass=request.getParameter("password");

            String role=request.getParameter("grp1");

            String query=null;

            String url=null;

            int n1=-2;

            String d=new Date().toString();

            boolean status=false;

            if(role.equals("Student")){

                query="select * from StudentMaster where StudentID=" + regId + " and Password=" + pass + "";

                url="Student/Home.jsp";

            }else if(role.equals("College")){

                query="select * from CollegeReg where CollegeID=" + regId + " and Password=" + pass + "";

                url="College/Home.jsp";

            }else if(role.equals("Company")){

                query="select * from CompanyReg where CompanyID=" + regId + " and Password=" + pass + "";

                url="Company/Home.jsp";

            }

            java.sql.ResultSet rs=ConnPack.ConnectionFactory.getInstance().getResultSet(query);

            if(rs.next()){

                javax.servlet.http.HttpSession session=request.getSession();

                session.setAttribute("regId", regId);

                session.setAttribute("role", role);

                response.sendRedirect(url);

            }

        }

    }

}
```



```

        if(role.equals("College")){
            String query1="update CollegeProfile set LastAccessDate='"+d+"' where CollegeId='"+regId+"'";
            n1=ConnPack.ConnectionFactory.getInstance().setData(query1);
        }else if(role.equals("Company")){
            String query2="update CompanyProfile set LastAccessDate='"+d+"' where
CompanyId='"+regId+"'";
            n1=ConnPack.ConnectionFactory.getInstance().setData(query2);
        }else if(role.equals("Student")){
            String query3="update Personal set LastAccessDate='"+d+"' where StudentId='"+regId+"'";
            n1=ConnPack.ConnectionFactory.getInstance().setData(query3);
        }
        if(n1 == 0 || n1 == -1)
            out.println("query not executed");
    }
    else
        response.sendRedirect("Invalid.jsp");
}
catch(Exception ex)
{
    ex.printStackTrace();
}
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}
}

```

Student Bean

```
public class StudentBean
{
    String name,email,dob,countryOfResidence,contact;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getDob() {
        return dob;
    }
    public void setDob(String dob) {
        this.dob = dob;
    }
    public String getCountryOfResidence() {
        return countryOfResidence;
    }
    public void setCountryOfResidence(String countryOfResidence) {
        this.countryOfResidence = countryOfResidence;
    }
    public String getContact() {
        return contact;
    }
    public void setContact(String contact) {
        this.contact = contact;
    }
}
```

Data Access Object

```
public class ConnectionFactory
{
    private Connection con;
    private static ConnectionFactory conn;
    private ConnectionFactory()
    {
    }
    static
    {
        conn=new ConnectionFactory();
    }
    public static ConnectionFactory getInstance()
    {
        return conn;
    }

    public int setData(String query,String DSN)
    {
        int n=0;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","");
            Statement stmt=con.createStatement();
            n=stmt.executeUpdate(query);
        }
        catch(Exception ex)
        {
            return -1;
        }
        return n;
    }
}
```

```

public ResultSet getResultSet(String query,String DSN)
{
    ResultSet rs=null;
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","");
        Statement stmt=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
        ResultSet.CONCUR_UPDATABLE);
        rs=stmt.executeQuery(query);
    }
    catch(Exception ex)
    {
        System.out.print("Conn fail in resultset");
        return null;
    }
    return rs;
}

public Vector getData(String query,String DSN)
{
    Vector V=new Vector();
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","");
        Statement stmt=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
        ResultSet.CONCUR_UPDATABLE);
        ResultSet rs=stmt.executeQuery(query);
        ResultSetMetaData rsmd=rs.getMetaData();
        int colcount=rsmd.getColumnCount();
        while(rs.next())
        {
            Vector sub=new Vector();

```

```
        for(int i=1;i<=colcount;i++)
            sub.add(rs.getString(i));
        V.add(sub);
    }
}
catch(Exception ex)
{
    System.out.println("Conn fail");
    return null;
}
return V;
}
}
```

JSP Page

```
<% @page import="Beans.StudentCompanySearchBean"%>
<% @page import="java.util.LinkedList"%>
<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<% String nameFromSession=(String)session.getAttribute("UserName");
    String searchBy = (String)request.getParameter("searchby");
    String searchValue=(String)request.getParameter("searchvalue");
    SearchResultDao obj= new SearchResultDao();
    LinkedList<StudentCompanySearchBean> searchBeanObj=obj.getSearchDetails(searchBy,searchValue);
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<table border="1" width="100%" style="text-align:center;">
<tr bgcolor= #4CAF50;>
<th>Serial Number</th>
<th>Name</th>
<th>Email</th>
<th>Gender</th>
<th>Date of Birth</th>
<th>Country</th>
<th>Contact</th>
<th>Secondary Percentage</th>
<th>Senior Secondary Percentage</th>
<th>Degree</th>
<th>Branch</th>
<th>College</th>
```

```

        <th>Year of Passing</th>

        <th>College Percentage</th>

    </tr>

    <% int i=1;

        for(StudentCompanySearchBean x:searchBeanObj)
        {
    %>

            <tr>

                <td> <% out.print(i++); %>

                <td> <% out.print(x.getName()); %></td>

                <td> <% out.print(x.getEmail()); %></td>

                <td> <% out.print(x.getGender()); %> </td>

                <td> <% out.print(x.getDob()); %></td>

                <td> <% out.print(x.getCountryOfResidence()); %></td>

                <td> <% out.print(x.getContact()); %></td>

                <td> <% out.print(x.getSecValue()); %></td>

                <td> <% out.print(x.getSenSecValue()); %></td>

                <td> <% out.print(x.getDegree()); %></td>

                <td> <% out.print(x.getBranch()); %></td>

                <td> <% out.print(x.getCollege()); %></td>

                <td> <% out.print(x.getYearOfPassing()); %></td>

                <td> <% out.print(x.getCollegeValue()); %></td>

            </tr>

    <%

        }

    %>

</table>

</body>

</html>

```

10.2 DATABASE TABLES

Table 10.1 Student Details

FIELD	TYPE	SIZE	CONSTRAINT
studentid	varchar	10	Primary key
studentname	varchar	30	Not null
gender	varchar	10	Not null
dateofbirth	date		Not null
adderss	varchar	30	Not null
collegeid	varchar	10	Foreign Key (College)
contact	varchar	15	Not null, Unique
emailed	varchar	30	Not null, Unique
password	varchar	20	Not null

Table 10.2 Academic Details

FIELD	TYPE	SIZE	CONSTRAINT
academicid	int	11	Primary key, Auto increment
studentid	varchar	10	Foreign key (Student)
secyearofpassing	varchar	4	Not null
secpercent	int	3	Not null
sensecyearofpassing	varchar	4	Not null
sensecpercent	int	3	Not null

Table 10.3 Company Details

FIELD	TYPE	SIZE	CONSTRAINT
companyid	varchar	10	Primary key
companyname	varchar	30	Not null
companyaddress	varchar	30	Not null
contact	varchar	15	Not null
website	varchar	30	Not null
lastaccessdate	date		Null
password	varchar	30	Not null

Table 10.4 College Details

FIELD	TYPE	SIZE	CONSTRAINT
collegeid	varchar	30	Primary key
collegename	varchar	30	Not null, Unique
university	date		Not null
collegeaddress	varchar	10	Not null
contact	varchar	15	Not null, Unique
website	varchar	30	Not null, Unique
password	varchar	20	Not null

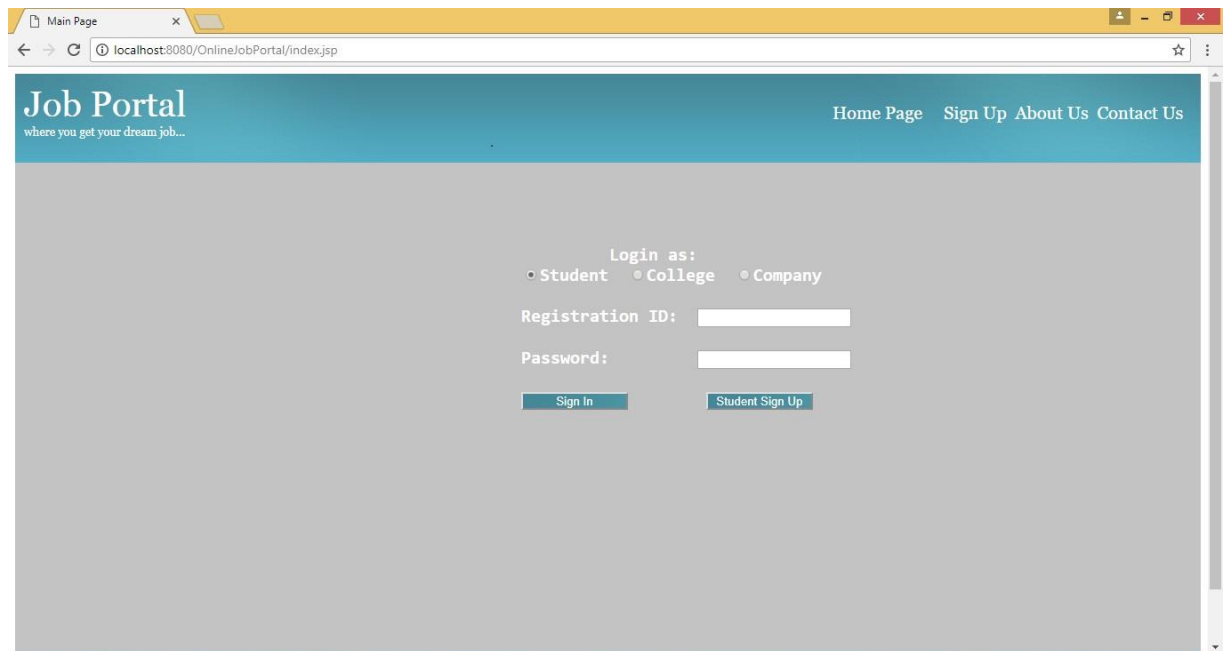
Table 10.5 Graduation Details

FIELD	TYPE	SIZE	CONSTRAINT
graduationid	int	11	Primary key, Auto increment
studentid	varchar	10	Foreign key (Student)
degree	varchar	10	Not null
branch	varchar	20	Not null
collegeid	varchar	30	Foreign Key (College)
yearofbegin	varchar	4	Not null
yearofpassing	varchar	4	Not null
percentage	int	3	Not null

Table 10.6 Document Upload

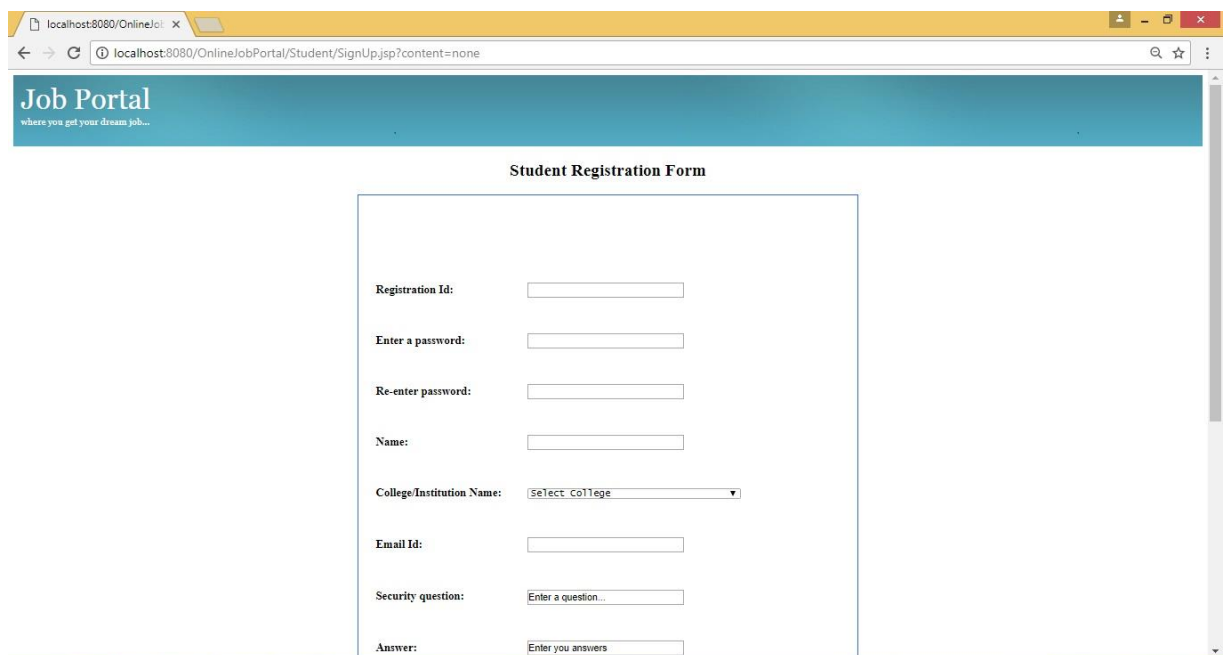
FIELD	TYPE	SIZE	CONSTRAINT
documentid	int	11	Primary key, Auto increment
studentid	varchar	10	Foreign key (Student)
docname	varchar	20	Not null, Unique
doclink	varchar	30	Not null, Unique
docdesc	varchar	50	Null

10.3 SCREENSHOTS



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/OnlineJobPortal/index.jsp'. The page has a blue header with the text 'Job Portal' and the tagline 'where you get your dream job...'. To the right of the header are links: 'Home Page', 'Sign Up', 'About Us', and 'Contact Us'. The main content area is light gray and contains a login form. The form has a section 'Login as:' with three radio buttons: 'Student', 'College', and 'Company'. Below this are two input fields: 'Registration ID:' and 'Password:'. At the bottom of the form are two buttons: 'Sign In' and 'Student Sign Up'.

Figure 10.1 Login Page



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/OnlineJobPortal/Student/SignUp.jsp?content=none'. The page has a blue header with the text 'Job Portal' and the tagline 'where you get your dream job...'. The main content area is light gray and contains a registration form titled 'Student Registration Form'. The form is enclosed in a white box with a gray border. It contains several input fields: 'Registration Id:', 'Enter a password:', 'Re-enter password:', 'Name:', 'College/Institution Name:' (with a dropdown menu showing 'select college'), 'Email Id:', 'Security question:' (with a placeholder 'Enter a question...'), and 'Answer:' (with a placeholder 'Enter you answers').

Figure 10.2 Student Registration

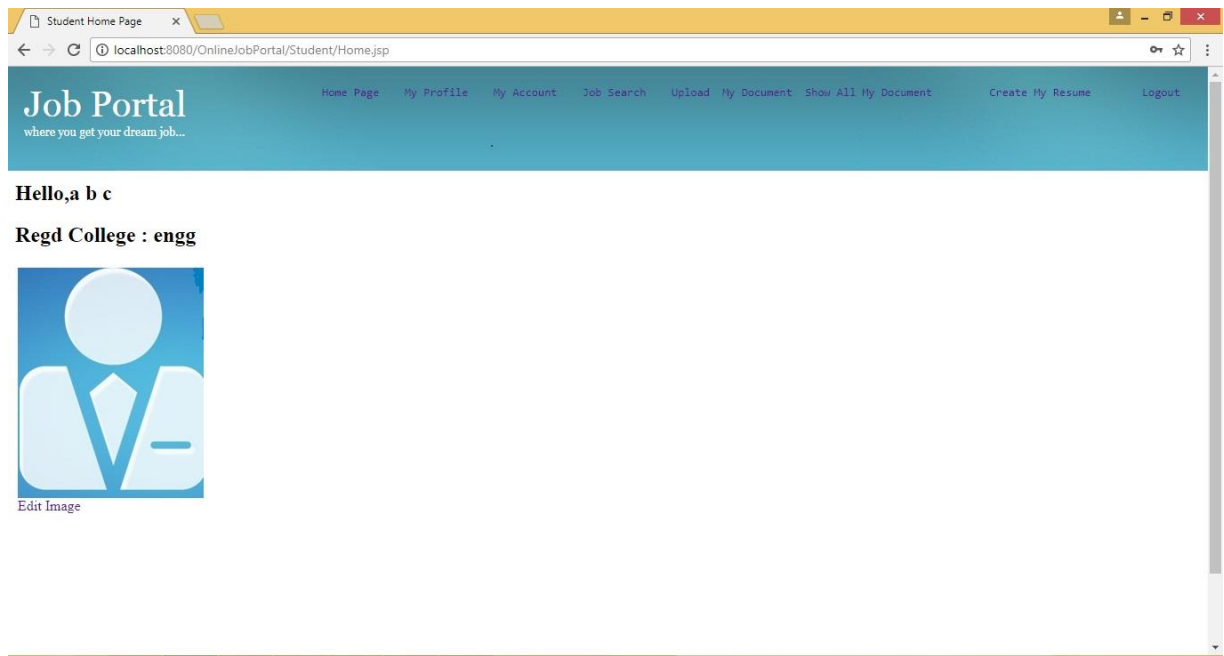


Figure 10.3 Student Home Page

Figure 10.4 Student Personal Information Form

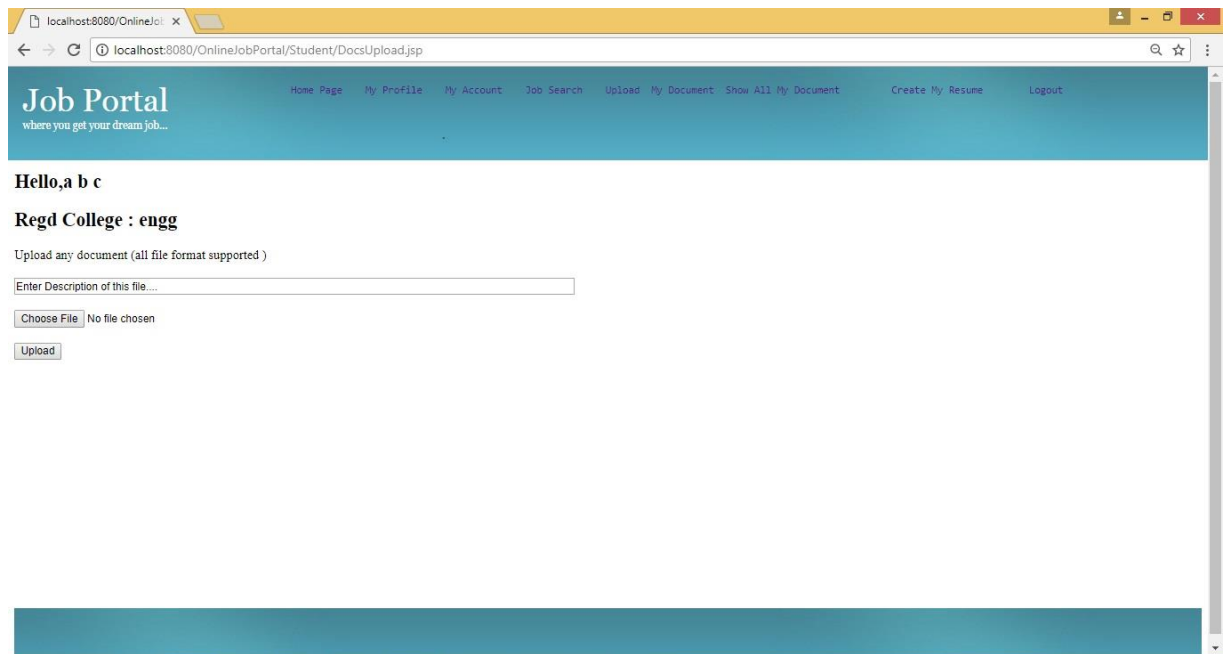


Figure 10.5 Document Upload

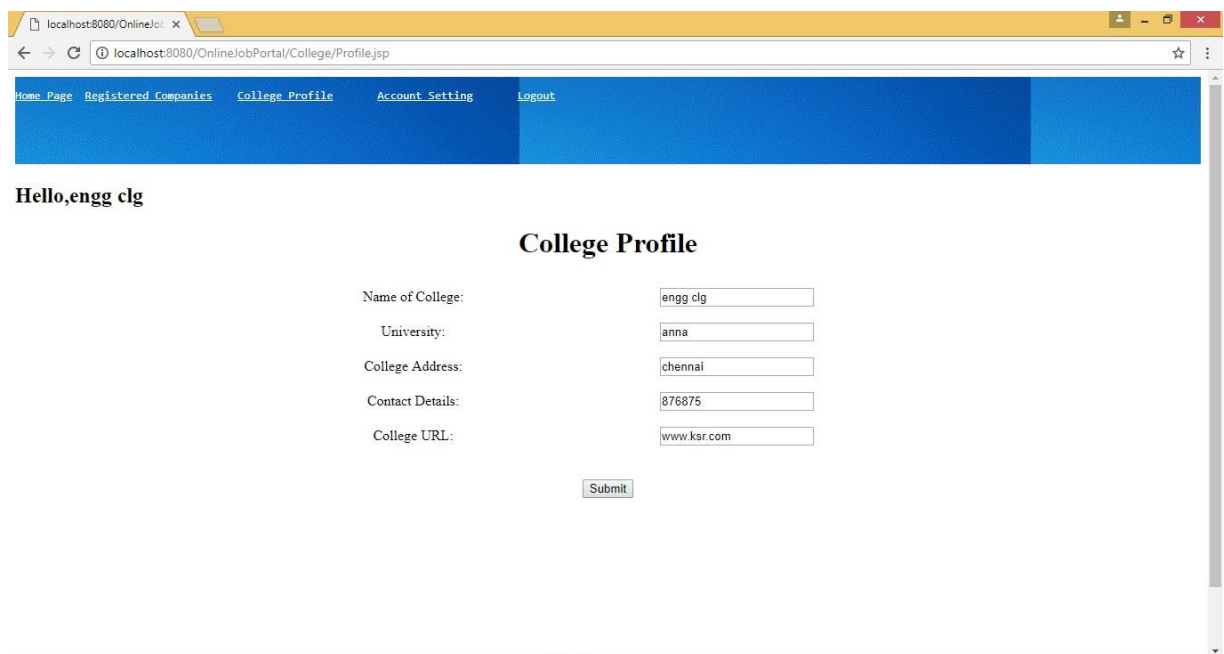


Figure 10.6 College Profile

localhost:8080/OnlineJob x

localhost:8080/OnlineJobPortal/Company/JobPosting.jsp

Home Page Registered Colleges Company Profile Company Account Job Posting Student Search Logout

Hello,google

Job Posting

CompanyID:

JobId:

Designation Post:

Package:

Location:

Last Date to Apply:

Secondary Percentage:

Senior Secondary Percentage:

Graduation Details

Degree:

Branch:

Selected Items:

Figure 10.7 Job Posting

localhost:8080/OnlineJob x

localhost:8080/OnlineJobPortal/Admin/Registration.jsp

Home Page My Account New Registration Logout

Hello,Admin

Select Registration Type

☐ Company

☒ College

Enter Registration Id

Please Fill Entry

Enter Password

Confirm Password

Figure 10.8 Add Company or college

REFERENCES

JOURNAL:

- [1] Deepinder Kaur, I.S. AlShawi, (12-23, JUNE 2016) "Your internet connection to your next job", IEEE Journal, vol. 41, no. 2, pp. 12-23.
- [2] ACCENTURE (2017): "How Much Are Customer Relationship Management Capabilities Really Worth? What Every CEO Should Know". <http://www.accenture.com/>.
- [3] Herbert Schildt, (2017)"Java – The Complete Reference", Tenth Edition, Tata McGrawHill, Publishing Company Limited.
- [4] Rafter, R. Bradley, K. & Smyth, B. (2016), "Automated collaborative filtering applications for online recruitment services", Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, pp-363-368.
- [5] Rob, P. & Coronel, C. (2015), "Database Systems: Design, implementation and management", Eight edition, Boston: Course Technology.
- [6] Sellami,S., Benharkat,A. & Amghar,Y. (2017), "Study of challenges and techniques in large scale", In ICEIS:10th International Conference on Enterprise Information Systems, Barcelona, Spain.
- [7] Vivek Kumar Sehgal, Akshay Jagtiani, Meha Shah (2013), "Job Portal- A Web Application for Geographically Distributed Multiple Clients", First International Conference on Artificial Intelligence, Modelling & Simulation, pp-199-204.

WEB REFERENCE:

<http://www.w3schools.com>

<http://www.javatpoint.com/java-tutorial>

<http://www.servlets.com>

<http://www.stackoverflow.com>

<https://dev.mysql.com>

<https://www.salesforce.com>