Project Report

on

Online Job Portal

By

Siddhi Sachin Kale

Indira College of Commerce and Science, Pune

M.Sc.(Comp.Sci.)-I

TABLE OF CONTENTS

5.3 Conclusions

5.4 Bibliography

1) INTRODUCTION

- Online Job Portal is a web-based system which allows interactive job vacancy portal for candidates.
- This web-based system is a dynamic site requiring constant updates from jobseekers as well as the recruiters regarding job vacancies. It is build usingASP.net framework.
- This project is aimed at developing an online search Portal for the job seekers.

1.1) Existing System and Need for System

- The present system requires applicants to search through print and visualmedia for job opportunities.
- Applicants need to apply for jobs using conventional methods and appear forinterview on a specified date at a specified location.
- Employers need to advertise the vacancies and sort all applicant details,conduct selection procedures and complete the formalities.
- This approach is tedious and requires much effort and resources
- Online Job Portal will provide the fast operation and low cost expense thanold system.
- Hence, the purpose of the project is to build an application program toreduce the manual work for managing the jobs, vacancies and resume.

1.2) Scope of Work

- This system will help the recruiter to keep track of all job applicationseffectively.
- An interactive friendly interface will be provided to facilitate differentservices.
- It may help for collecting perfect management in detail in a very short periodof time.
- It will also reduce the cost of collecting management & also the collection procedure will go on smoothly.
- The basic scope contains:
  - Jobseeker's area
  - Company's area
  - Administrator's area

1.3) Operating Environment – Hardware and Software Hardware Requirement-

PROCESSOR: - Core 2 Duo or Athlon X2 at 2.4 GHzRAM:- 4

GB RAM

HARD DISK: - 250 GB

Software Requirements: - Operating

System: - Windows10Web Server: -

IIS 7

Dot Net Framework: - Dot Net Framework 3.5

Database: - MYSQL

2) PROPOSED SYSTEM

2.1) Proposed System

In the proposed system, we have following modules.

Admin: -

In this module the Admin can view all the users, recruiters and jobs. He can also see the resumes uploaded and download it. Admin will have the right to delete the jobseeker, recruiter and jobs.

Job Seeker: -

In this module Job Seeker registers himself and upload his resume and fill the profile and after login he will search for the job on various conditions. He can update his profile and resume. The jobseeker can filter the jobs according to the time and type.

Recruiter: -

In this module Recruiter registers himself and his company. After login he will addnew job. He will provide all the details of the job. He will be able to see who has applied to his job and can see the resume. He will see the details of the jobseeker and can contact him directly by mail.

Authentication: -

This module contains all the information about the authenticated user. User withouthis username and password can't enter into the login. He needs to first signup.

Reports: -

This module contains all the information about the reports generated by the adminbased on the particular job seeker, particular job provider and all jobs generated by the job providers.

2.2) Objectives of System

Online Job Portal is developed for easy communication between jobseekers and recruiters. Both Jobseekers and Recruiters can get benefited through this system.

This Software is fully integrated with Jobseeker and Recruiter Relationship Management and developed in a manner that is easily manageable, time and cost saving that shows relieving one from manual works .This feature's helps in many ways like saving cost, time and paperwork, this shows the paperless environment.

2.3) User Requirements

The online job portal is designed to connect job seekers with recruiters. The system is designed for three types of users, including job seekers, recruiters, and admin.

1) Job seekers: The job seeker is the primary user who creates a profile and looks for job vacancies. Specifically, the system should allow the job seeker to:

- Create a profile and upload a resume and cover letter.
- Search for job vacancies.
- Apply for job vacancies by submitting the resume.
- Edit and update the profile and resume.

2) Recruiters: The recruiters are secondary users who create company profiles and post job vacancies. Specifically, the system should allow the recruiters to:

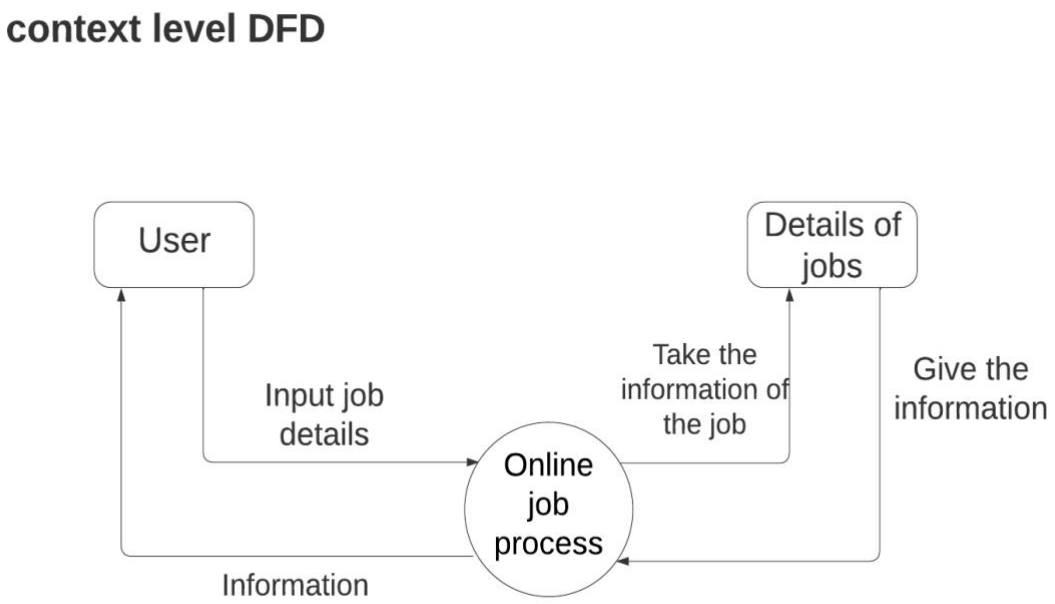
- Create a company profile and post job vacancies.
- Manage the job vacancies, applications, and candidate data.

3) Admin: The admin is a special user login that has maximum privileges. The admin can manage job vacancies, job seekers, recruiters, and other activities on the portal. Specifically, the system should allow the admin to:

- Manage users.
- Manage job vacancies, applications, and candidate data.
- Monitor the portal's activities and performance.

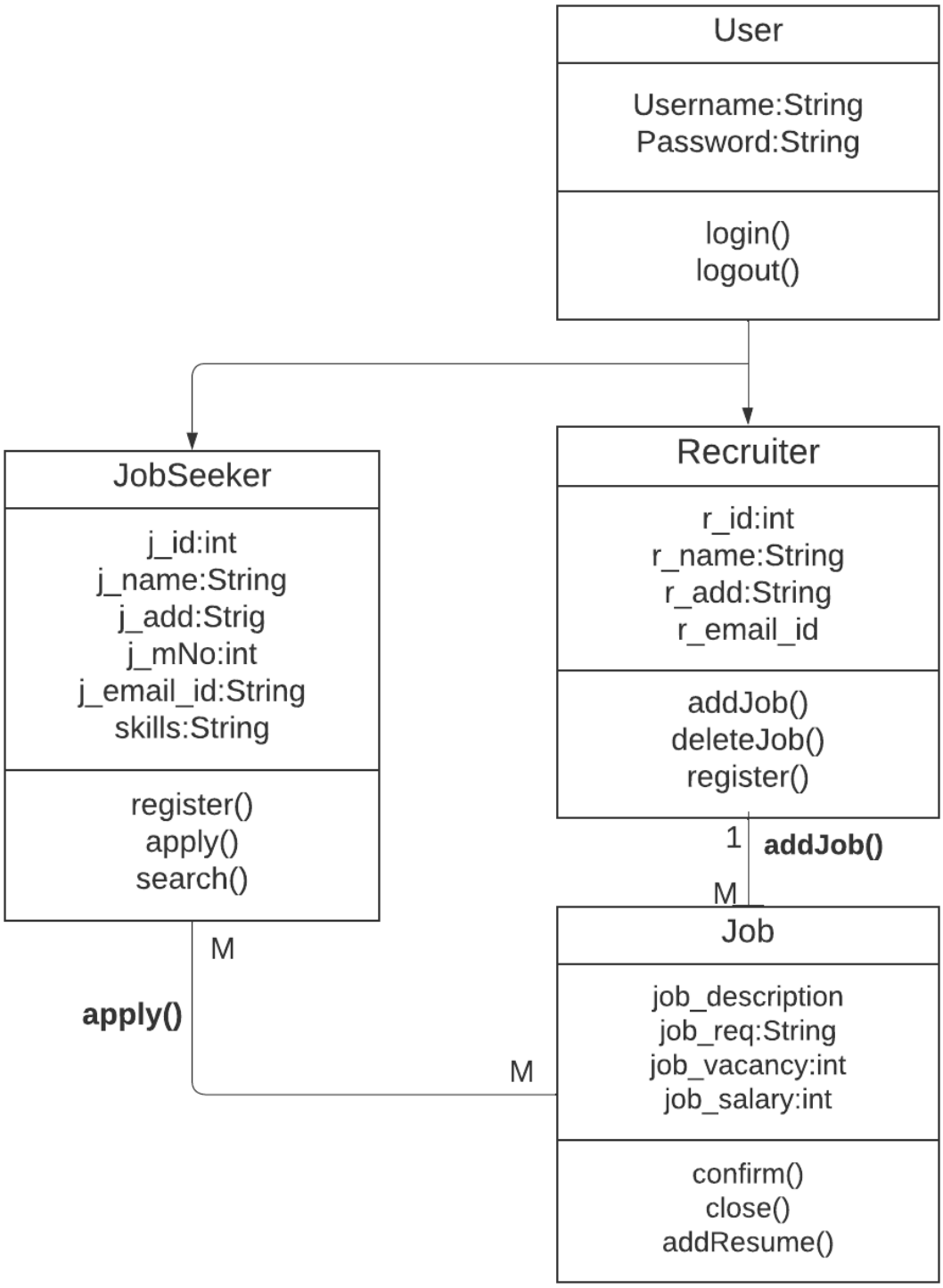
3) ANALYSIS & DESIGN

3.1) Data Flow Diagram (DFD)



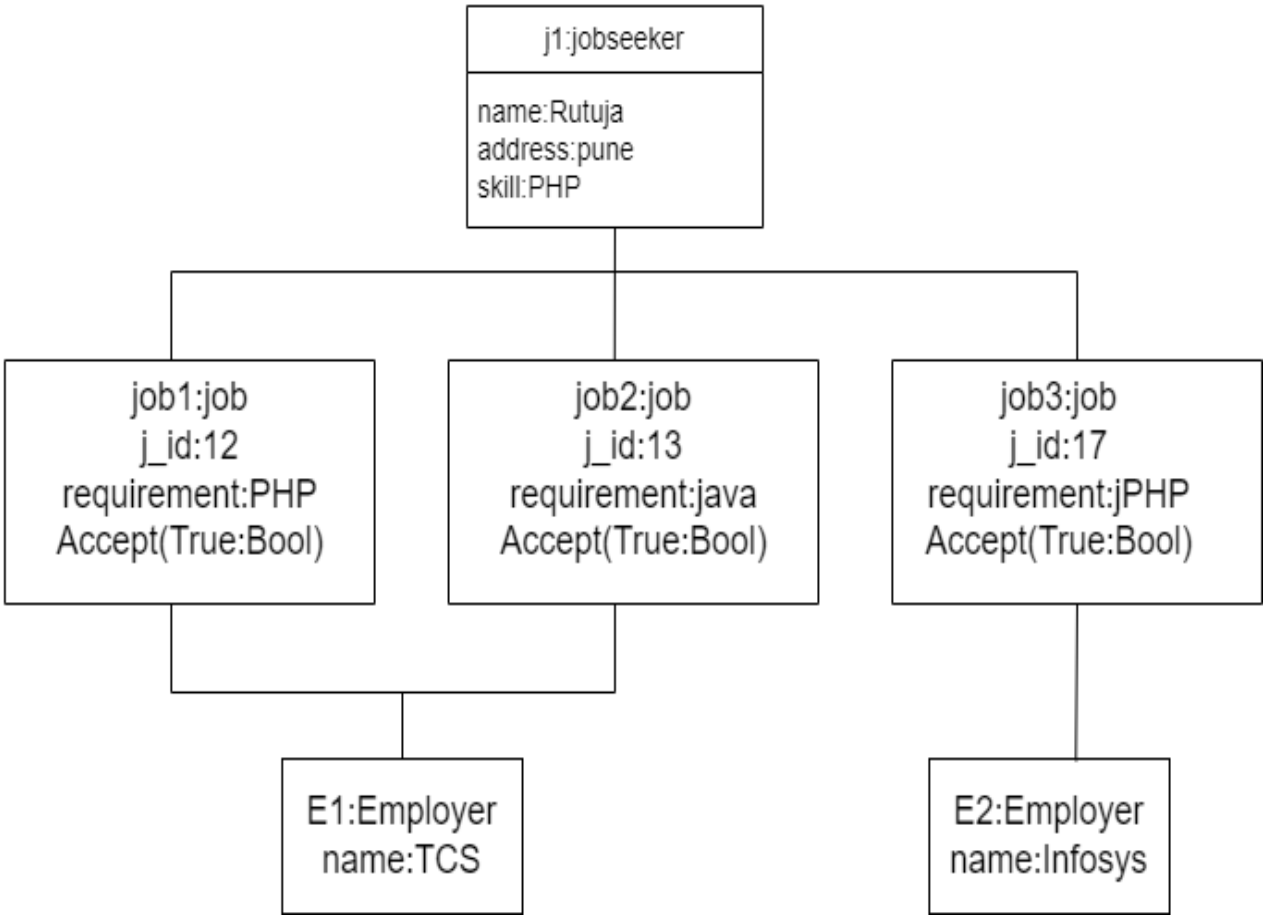
3.2) UML Diagrams:-

3.2.1 Class Diagram



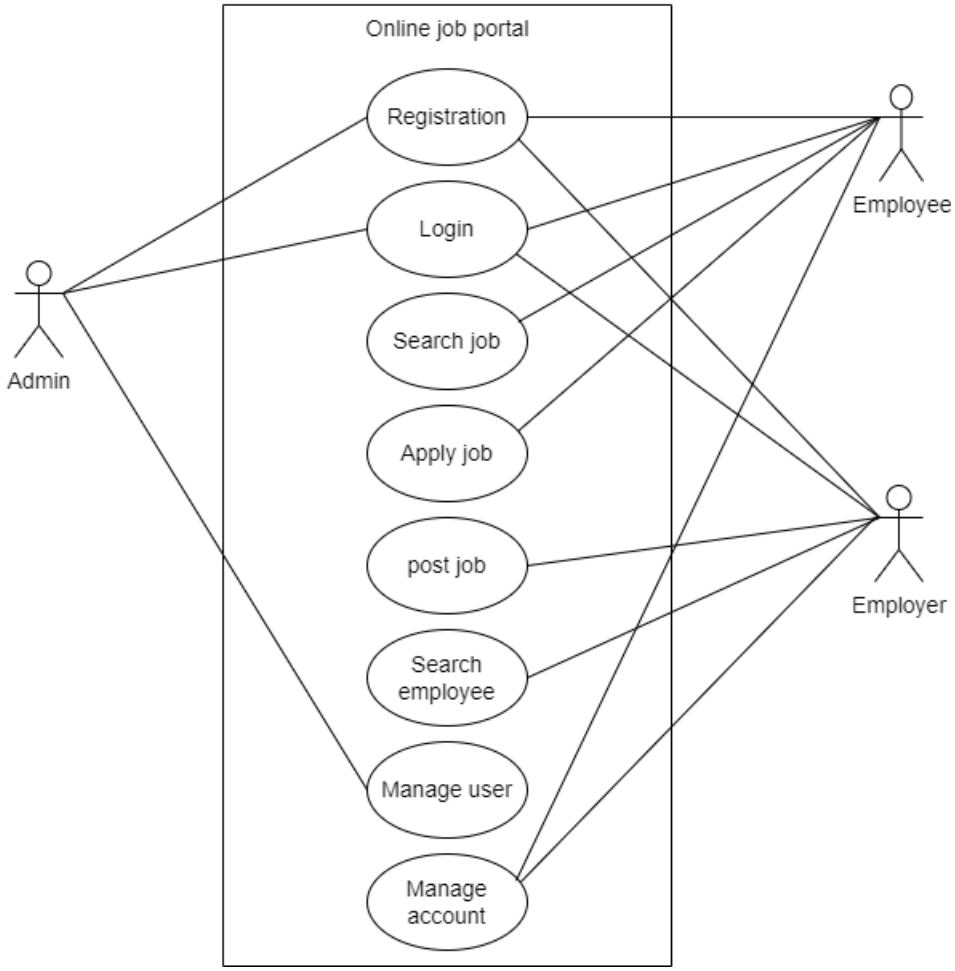
3.2.2. **Object diagram**:-

Object diagram:

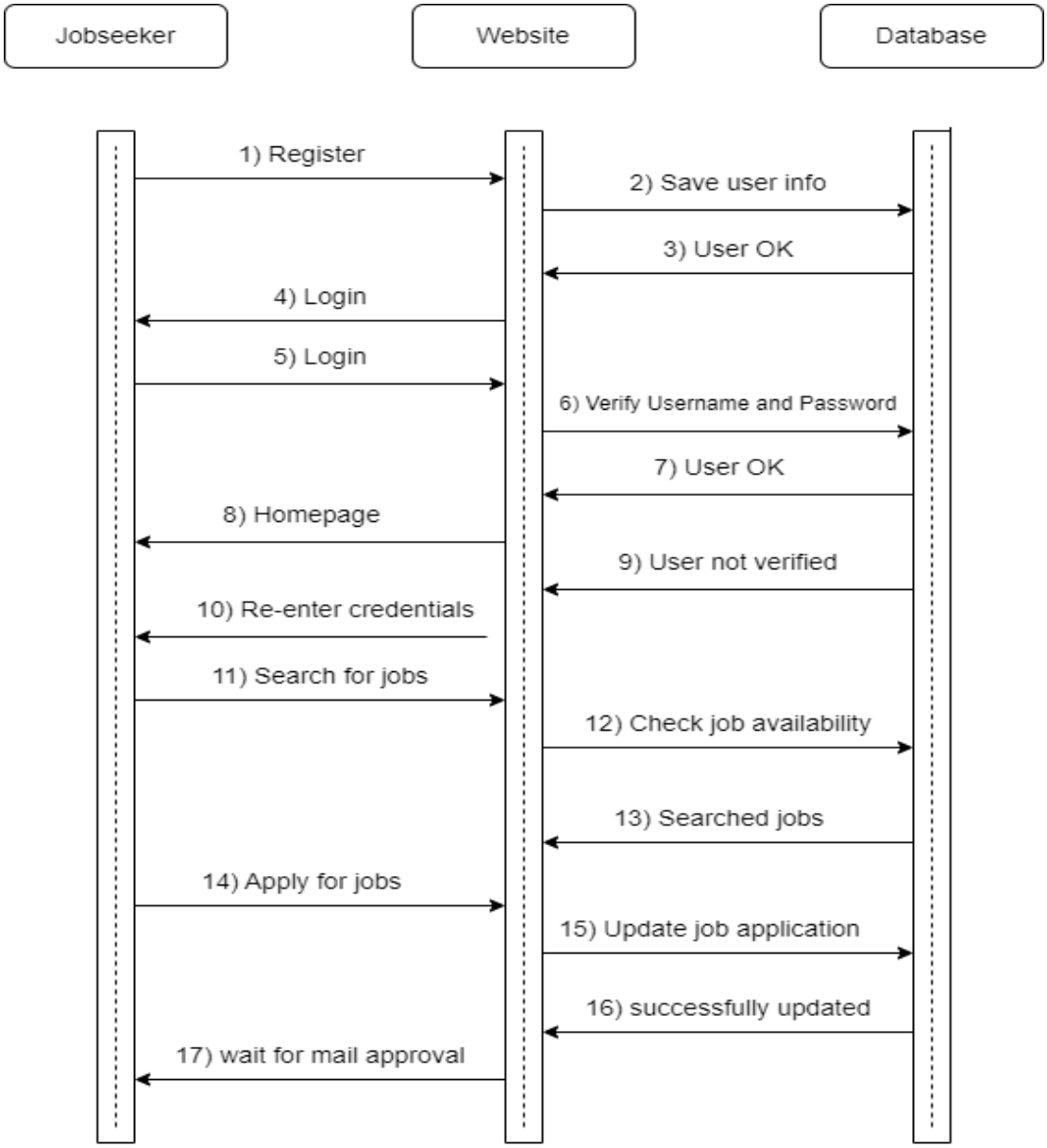


3.3.3 Use Case Diagram:-

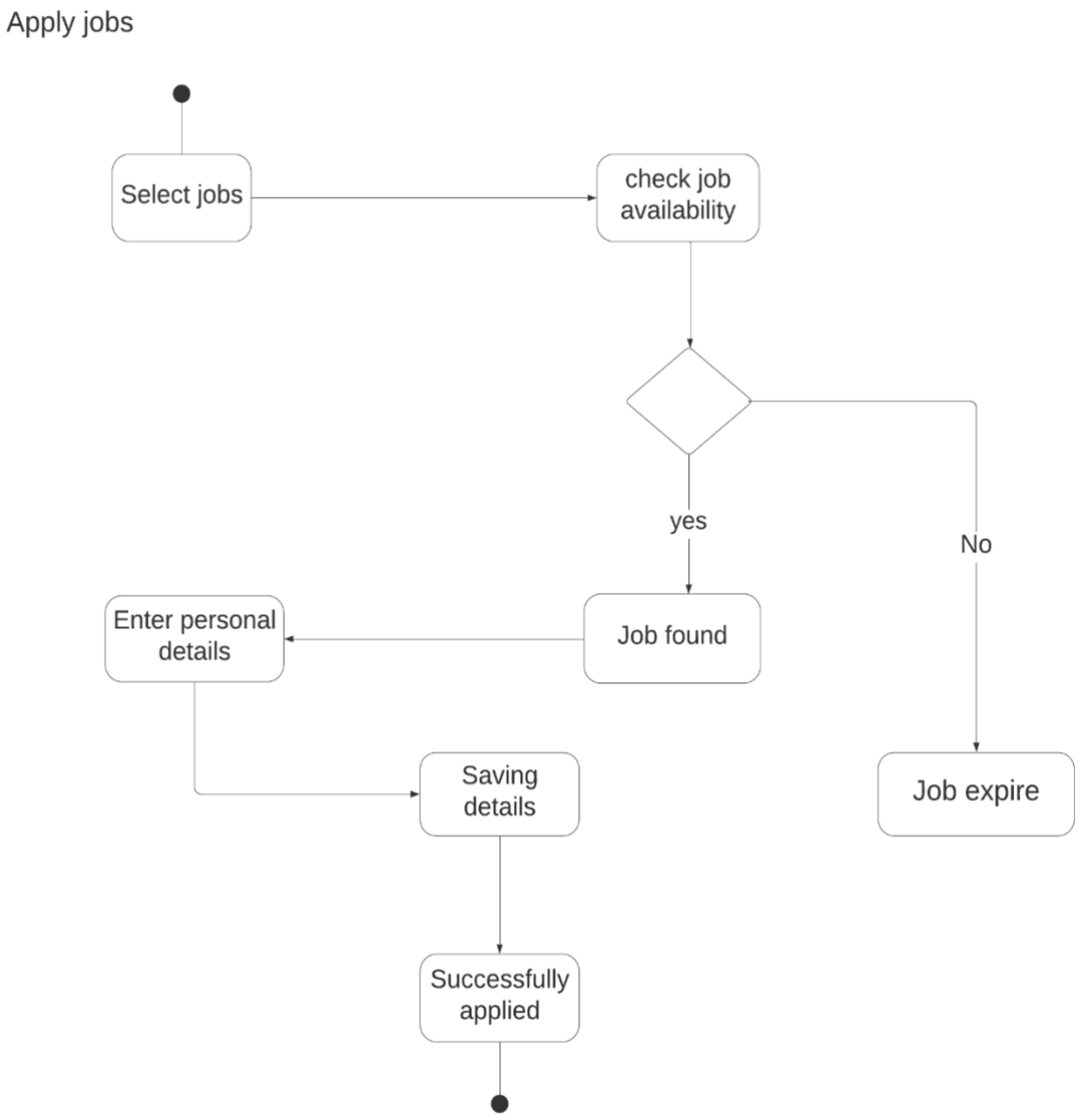
Use Case diagram



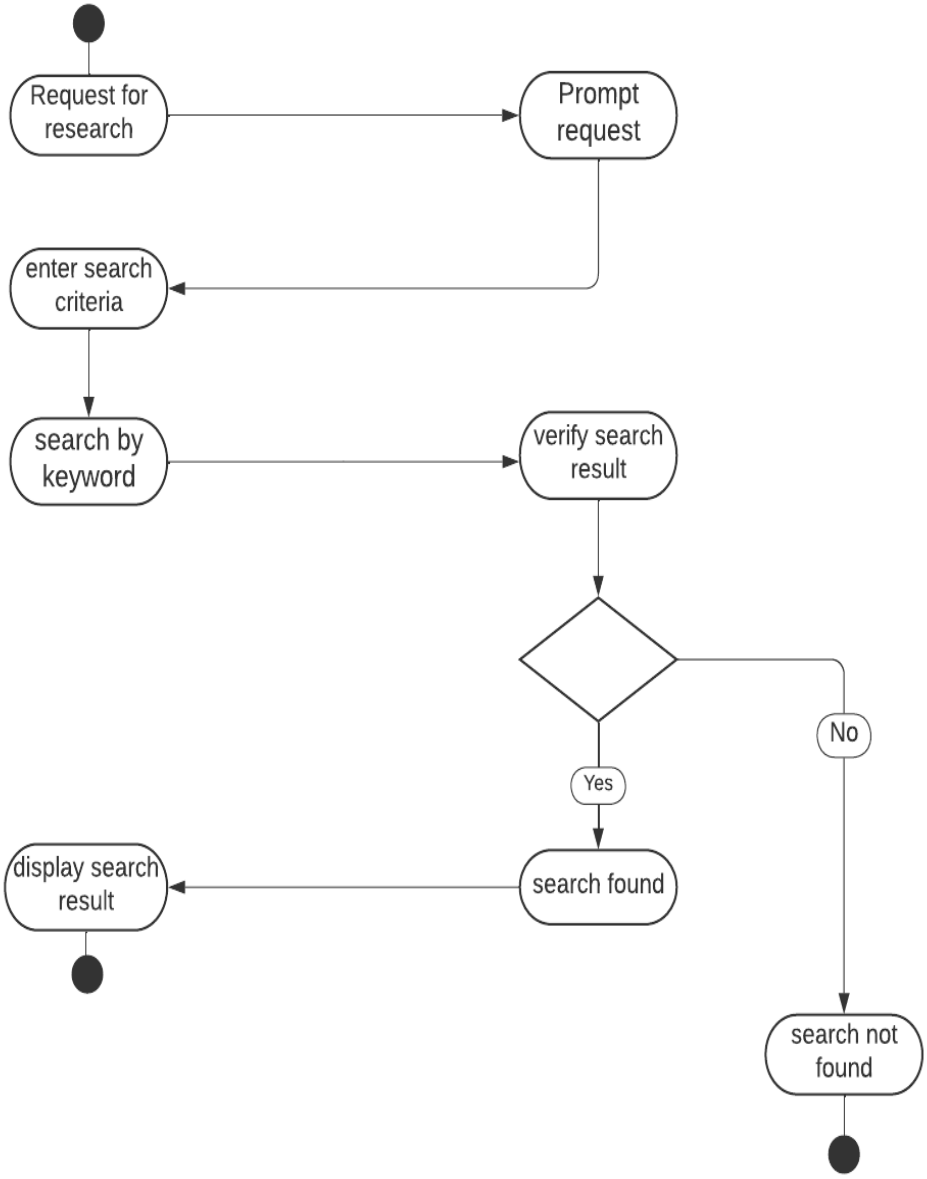
3.3.4 **Sequence Diagram**:-



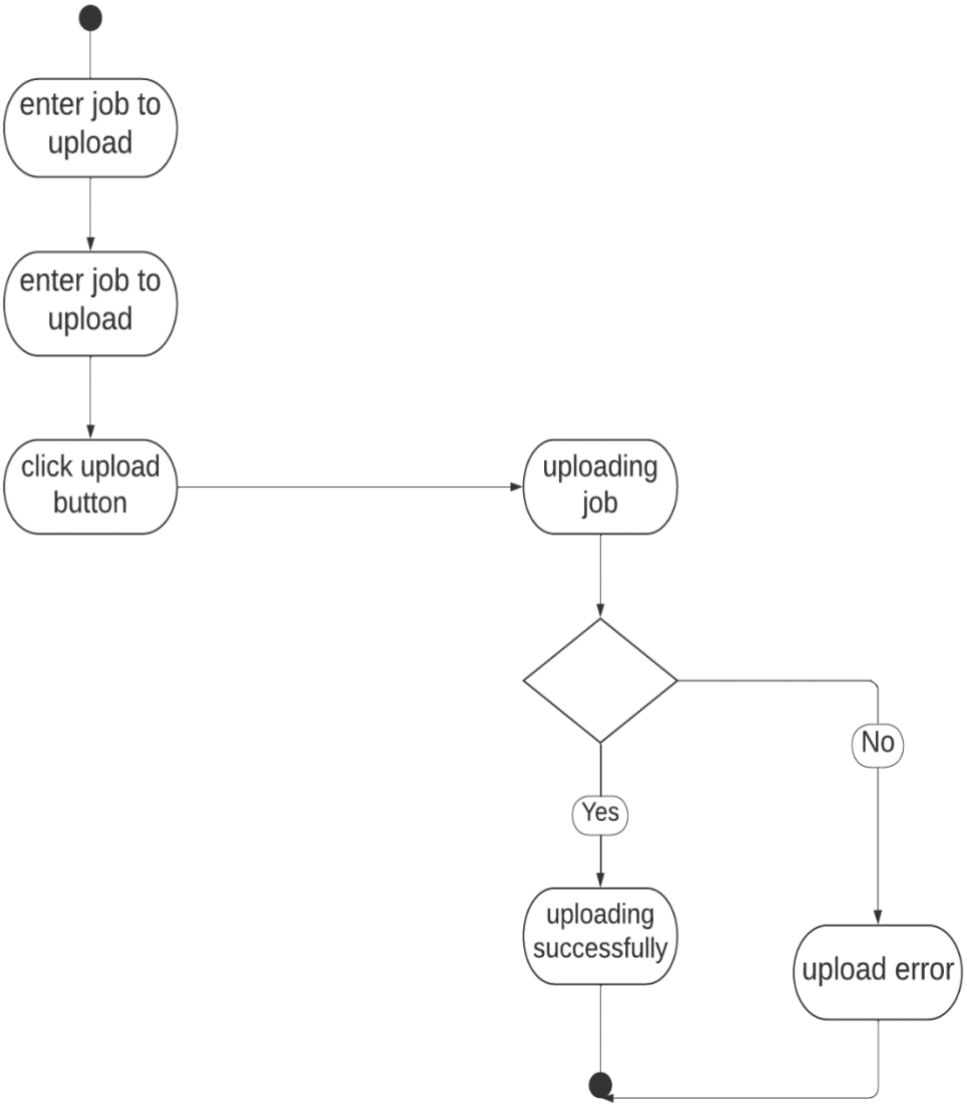
3.3.5.Activity Diagram



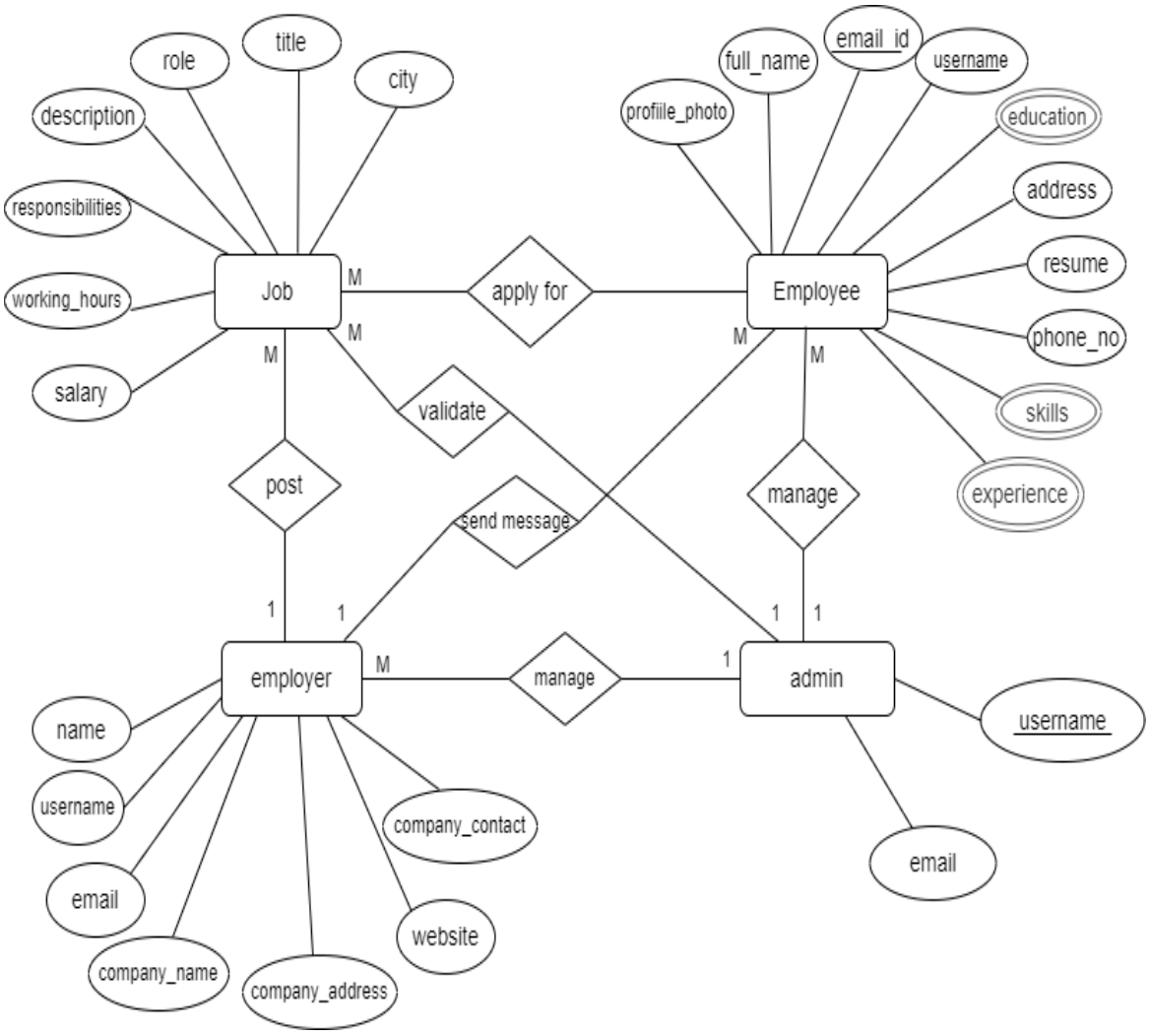
Search jobs


Request for research
Prompt request
enter search criteria
search by keyword
verify search result
No
Yes
display search result
search found
search not found

Upload jobs


enter job to upload
enter job to upload
click upload button
uploading job
No
Yes
uploading successfully
upload error

3.3) Entity Relationship Diagram (ERD)



3.5) Data Dictionary

Database :- mydb

Table Name:- User

Field	Type	Null	Key	Default	Extra
UserId	int	NO	PRI	NULL	auto_increment
Username	varchar(50)	YES	UNI	NULL	
Password	varchar(50)	YES		NULL	
Email	varchar(50)	YES		NULL	
Mobile	varchar(50)	YES		NULL	
TenthGrade	varchar(50)	YES		NULL	
TwelfthGrade	varchar(50)	YES		NULL	
GraduationGrade	varchar(50)	YES		NULL	
PostGraduationGrade	varchar(50)	YES		NULL	
Phd	varchar(50)	YES		NULL	
Workson	varchar(100)	YES		NULL	
Experience	varchar(50)	YES		NULL	
Resume	text	YES		NULL	

Field	Type	Null	Key	Default	Extra
Address	varchar(500)	YES		NULL	
Name	varchar(50)	YES		NULL	

Table Name:- Recruiter

Field	Type	Null	Key	Default	Extra
UserId	int	NO	PRI	NULL	auto_increment
Username	varchar(50)	YES	UNI	NULL	
Password	varchar(50)	YES		NULL	
Email	varchar(50)	YES		NULL	
Mobile	varchar(50)	YES		NULL	
CompanyName	varchar(80)	YES		NULL	
Post	varchar(50)	YES		NULL	
Qualifications	varchar(500)	YES		NULL	
Experience	varchar(50)	YES		NULL	
Address	varchar(500)	YES		NULL	
Name	varchar(50)	YES		NULL	

Table Name: Jobs

Field	Type	Null	Key	Default	Extra
JobId	int	NO	PRI	NULL	auto_increment
Title	varchar(50)	YES		NULL	
Description	text	YES		NULL	
Qualification	varchar(50)	YES		NULL	
Experience	varchar(50)	YES		NULL	
Specialization	text	YES		NULL	
LastDateToApply	date	YES		NULL	
Salary	varchar(50)	YES		NULL	
JobType	varchar(50)	YES		NULL	
CompanyName	varchar(200)	YES		NULL	
CompanyImage	varchar(500)	YES		NULL	
Website	varchar(100)	YES		NULL	
Email	varchar(50)	YES		NULL	

Field	Type	Null	Key	Default	Extra
Address	text	YES		NULL	
CreateDate	datetime	YES		NULL	
NoOfPost	int	YES		NULL	

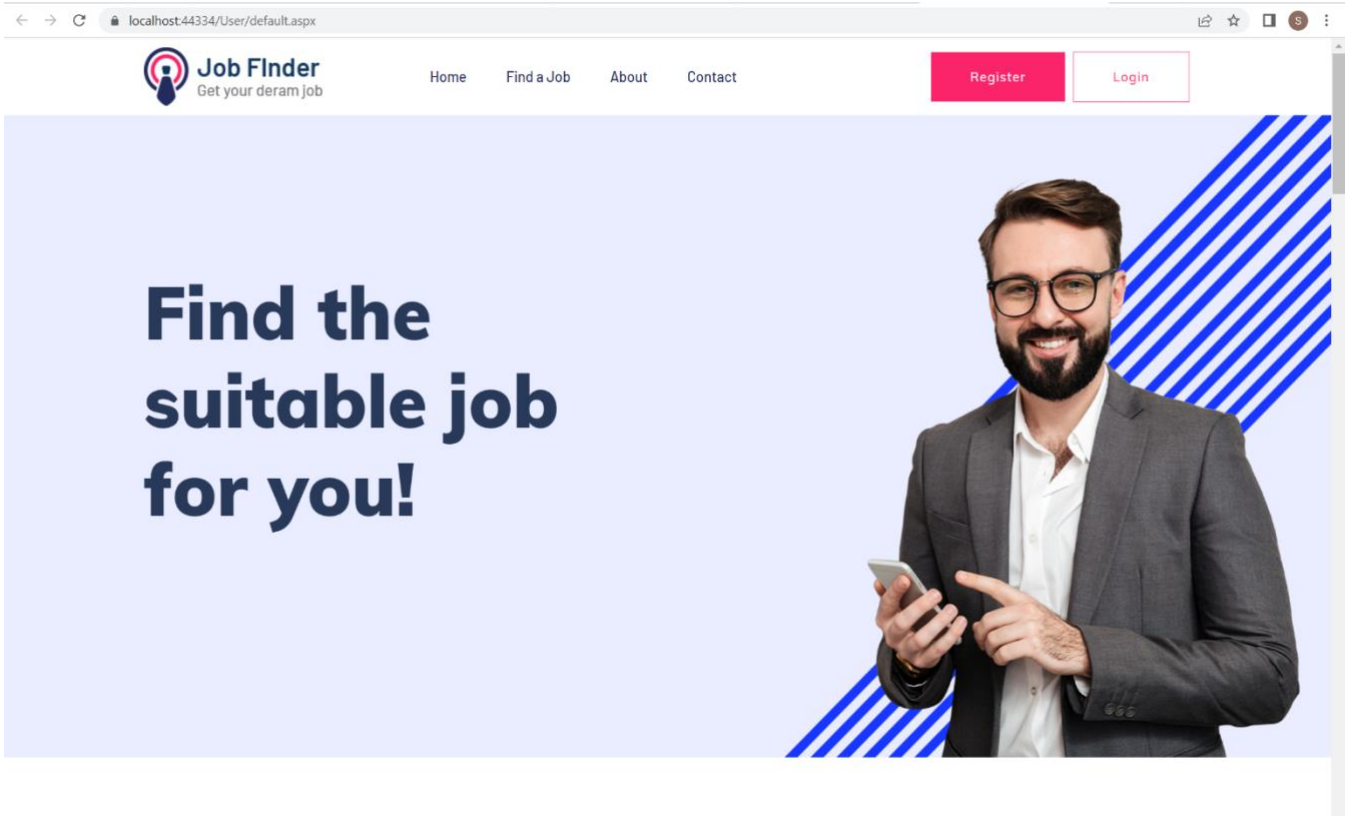
Table Name : Appliesjobs

Field	Type	Null	Key	Default	Extra
AppliedJobId	int	NO	PRI	NULL	auto_increment
JobId	int	YES		NULL	
UserId	int	YES		NULL	

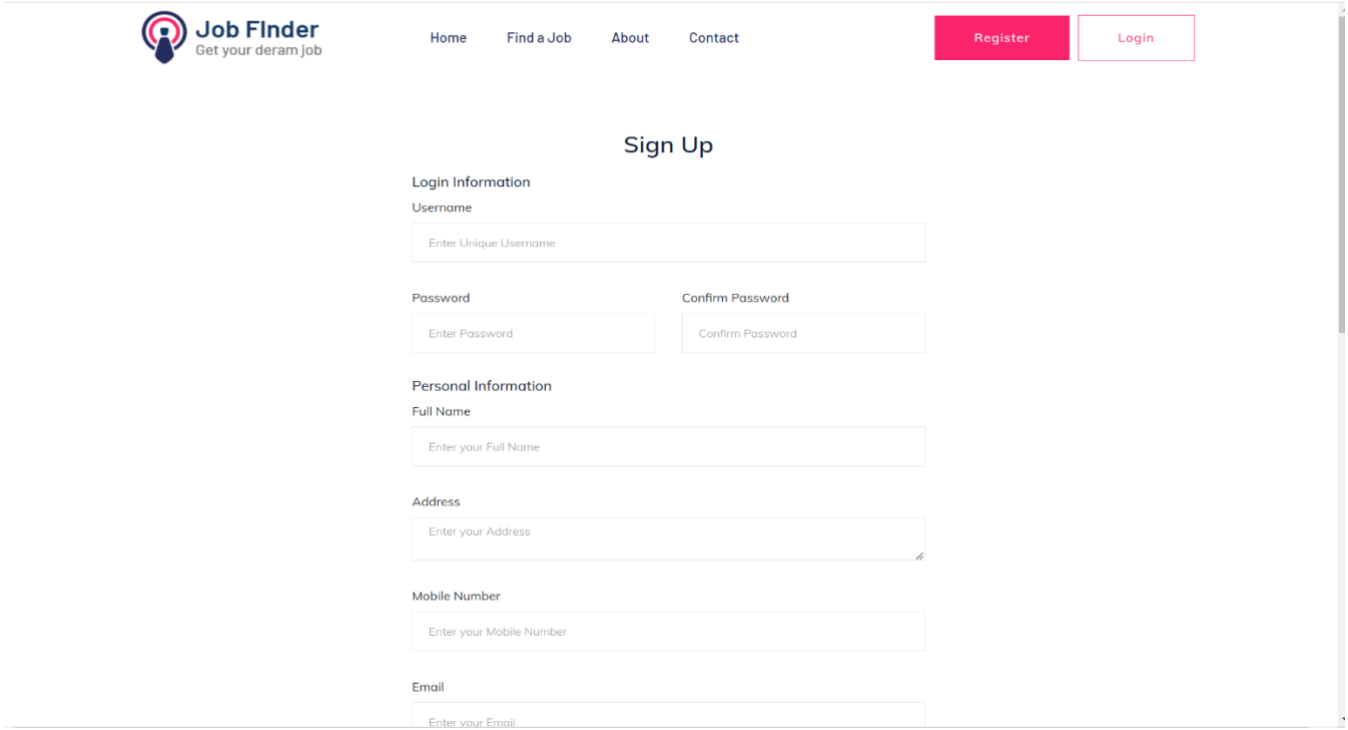
4) USER MANUAL

4.1) Menu Screens:

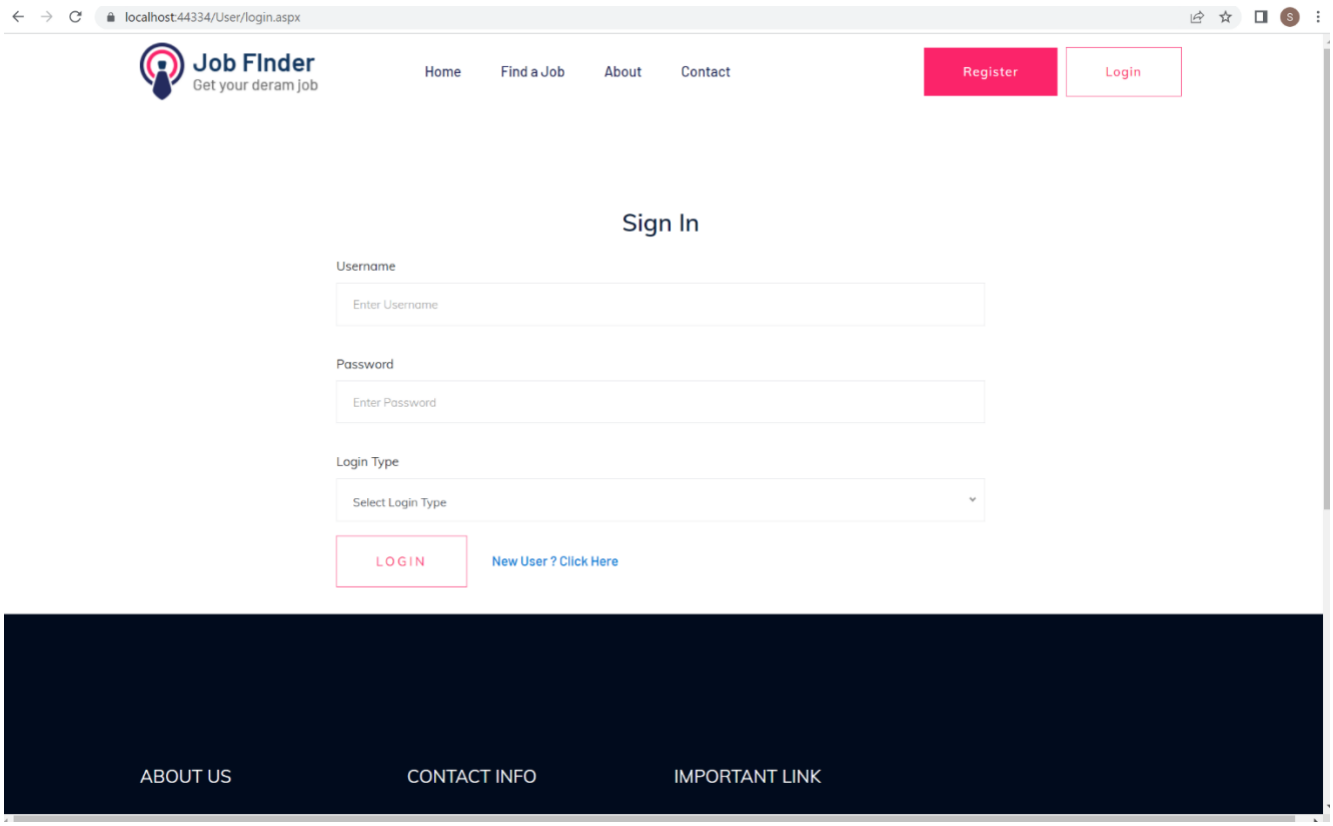
Homepage :



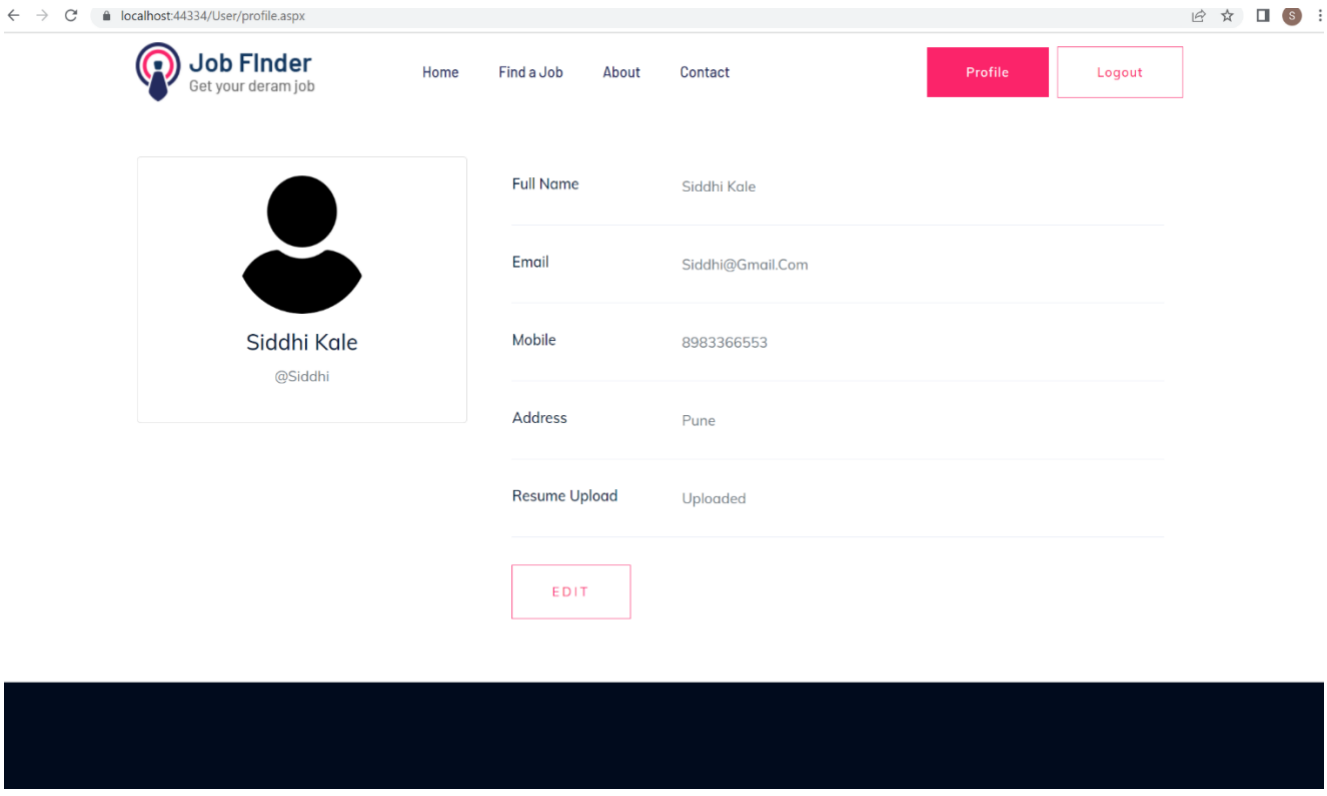
Register Page :



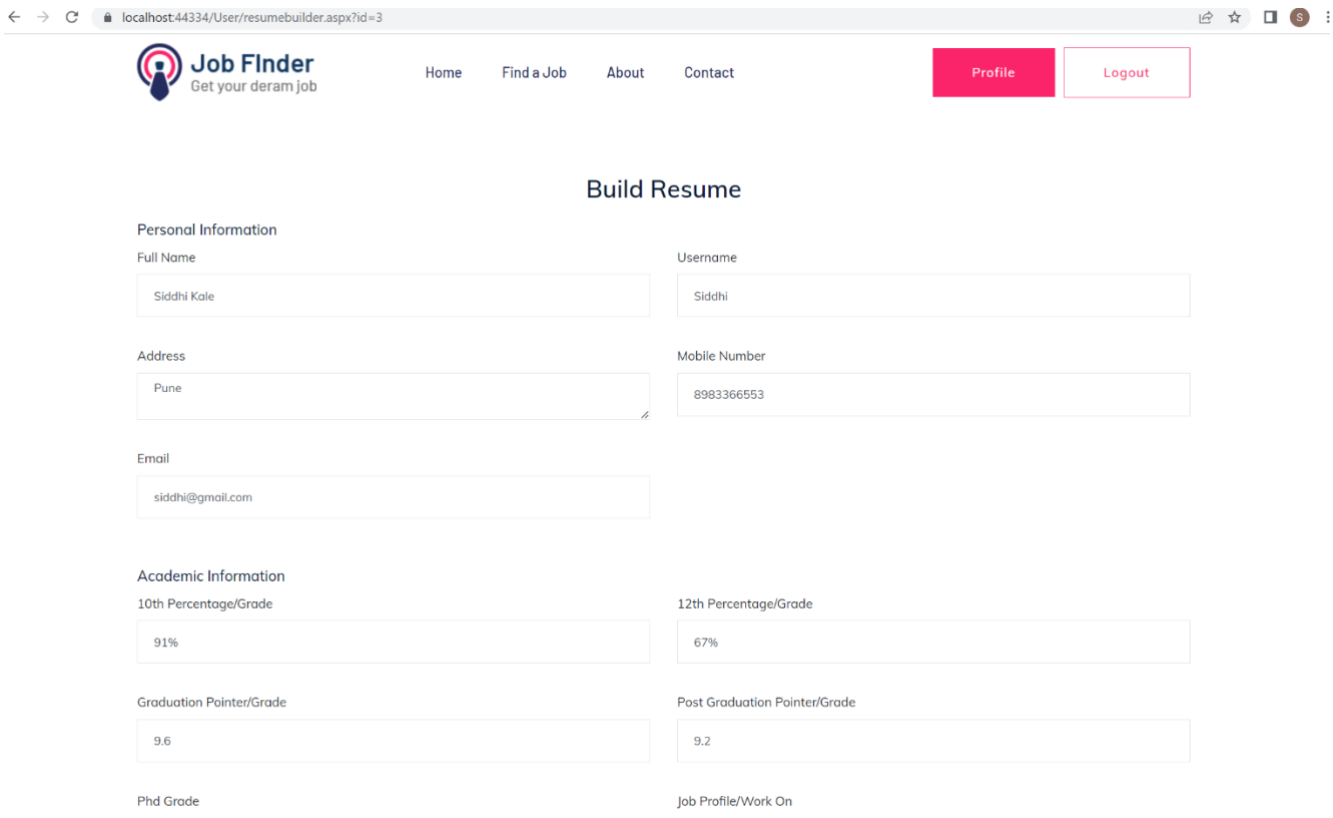
Login Page:



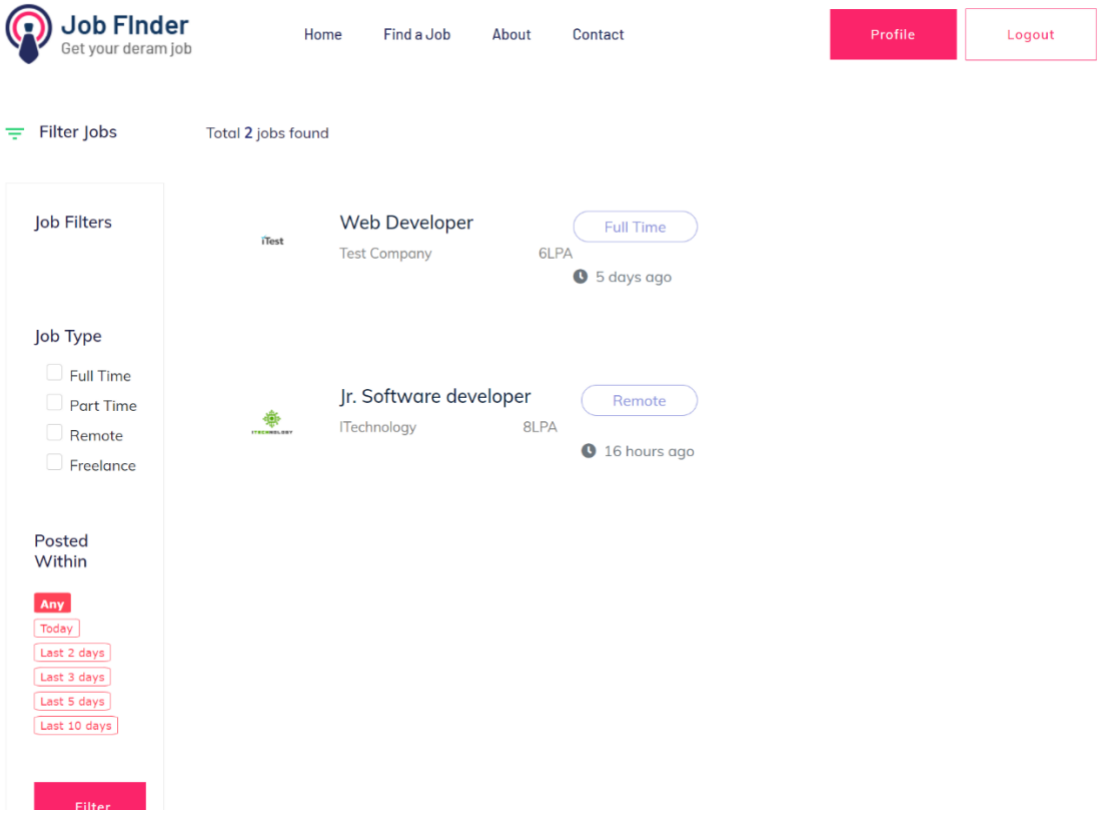
Jobseeker Profile Page :



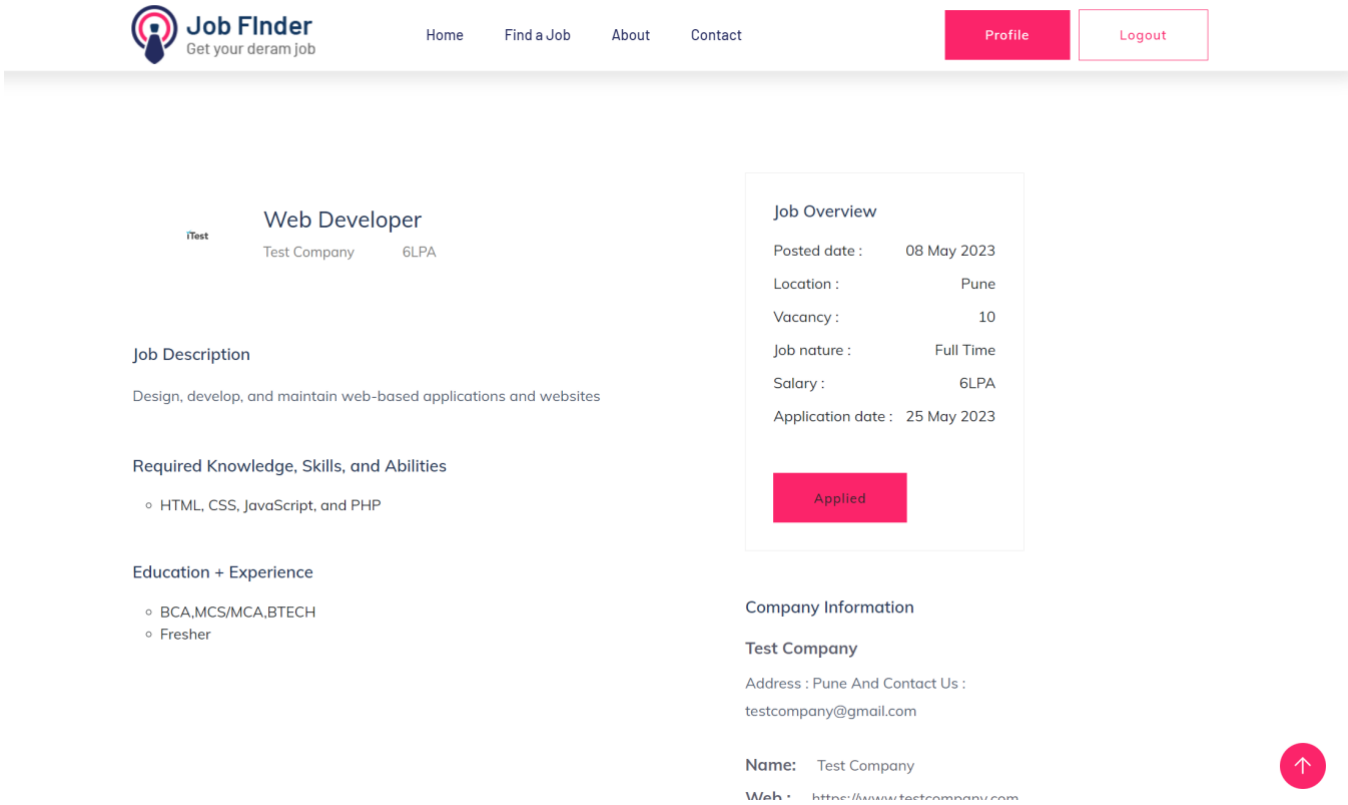
Jobseeker Profile edit page:



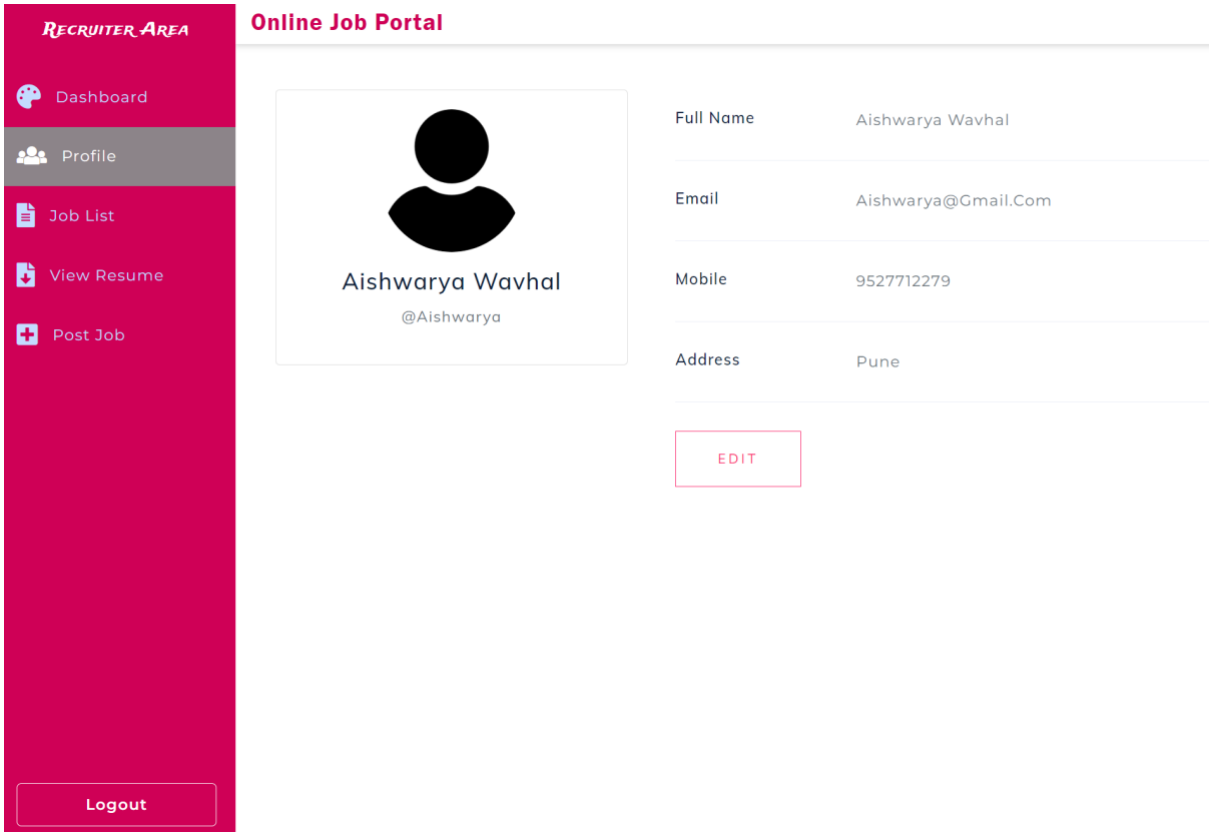
Job List Page:



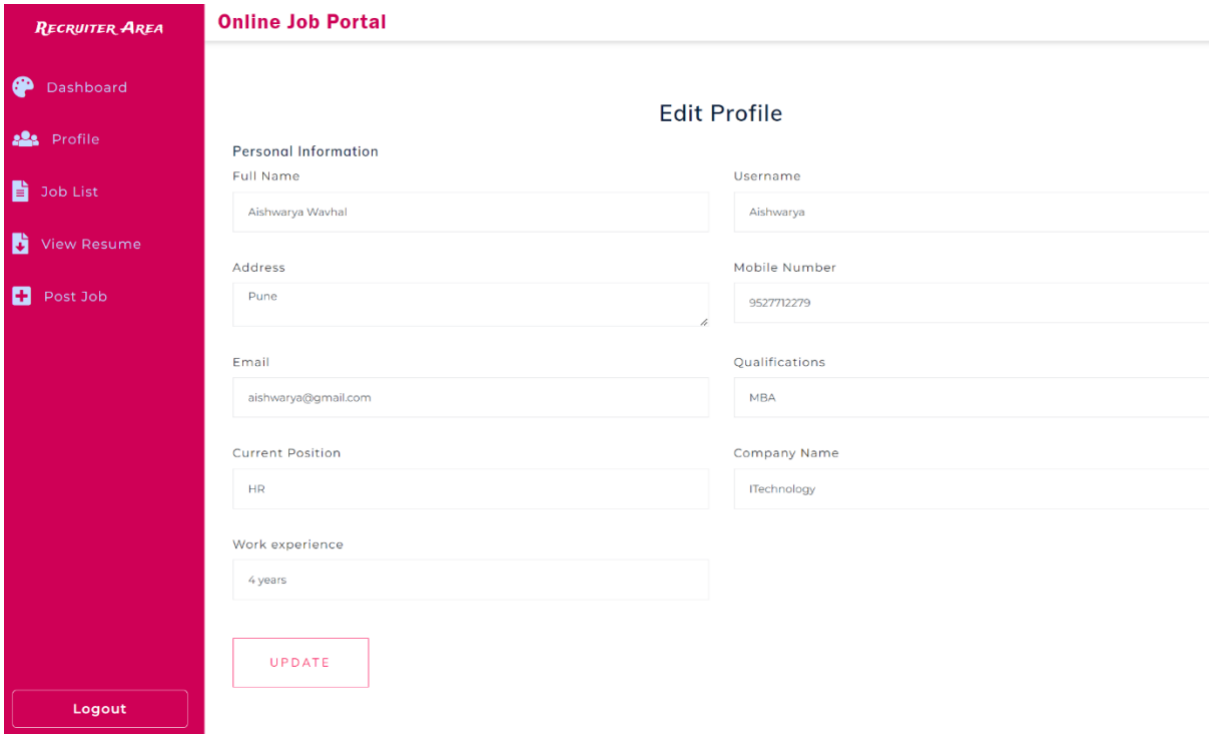
Job details and apply :



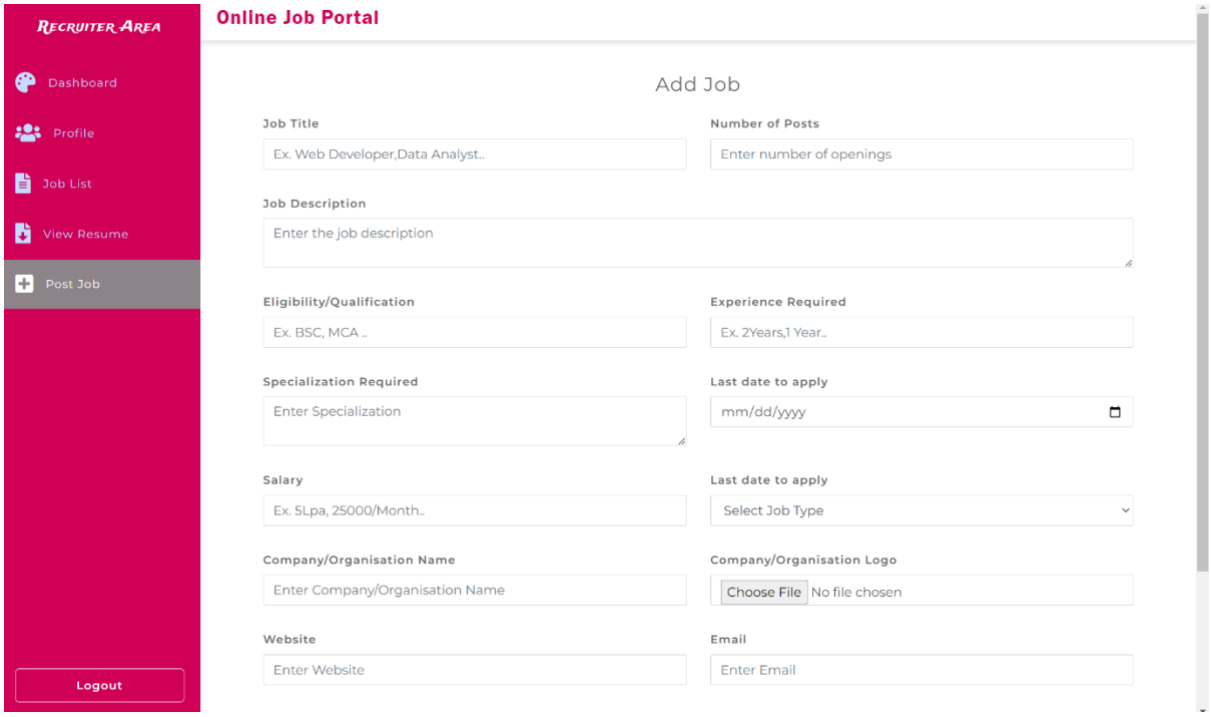
Recruiter Profile:



Recruiter Profile edit :



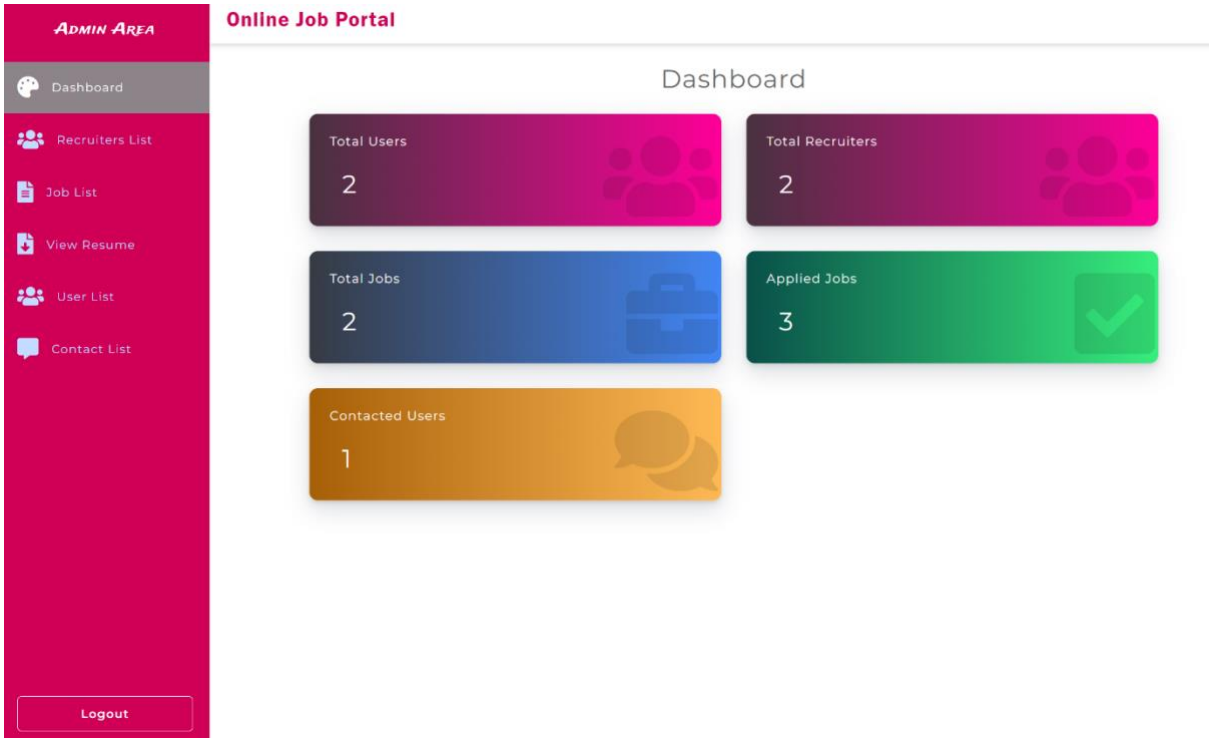
Add/Post job :



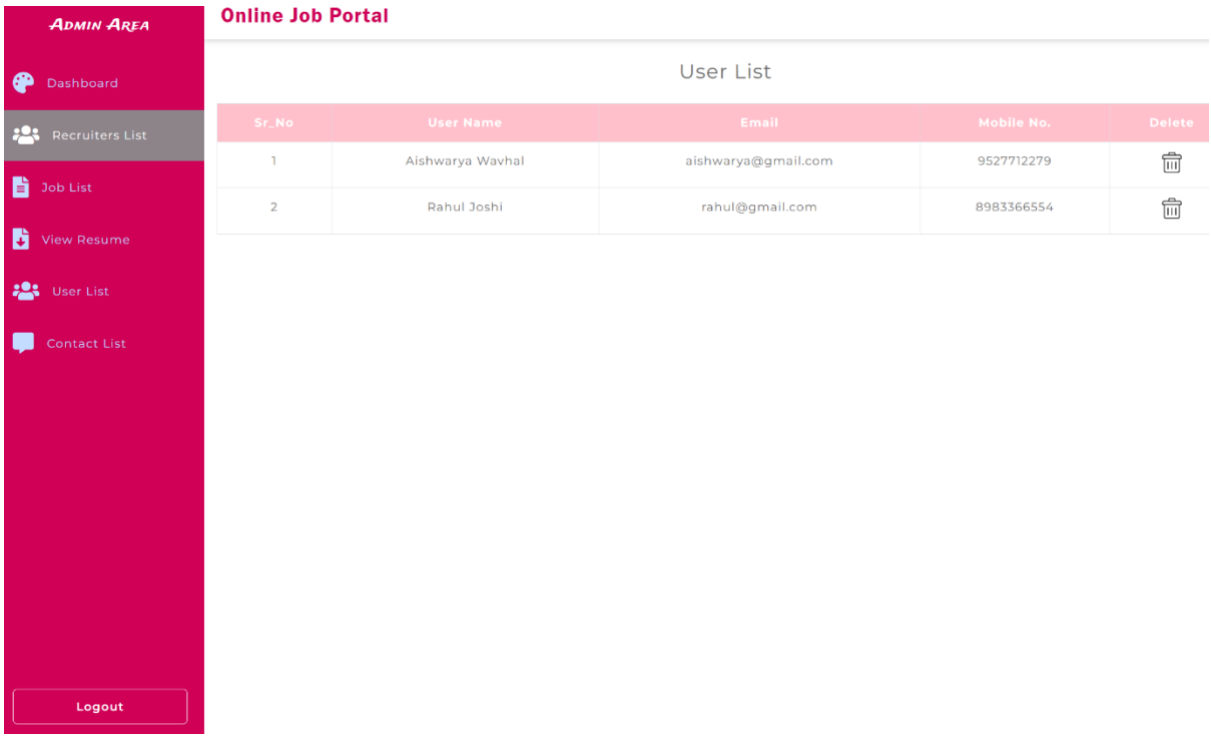
View Jobs :

View Resume :

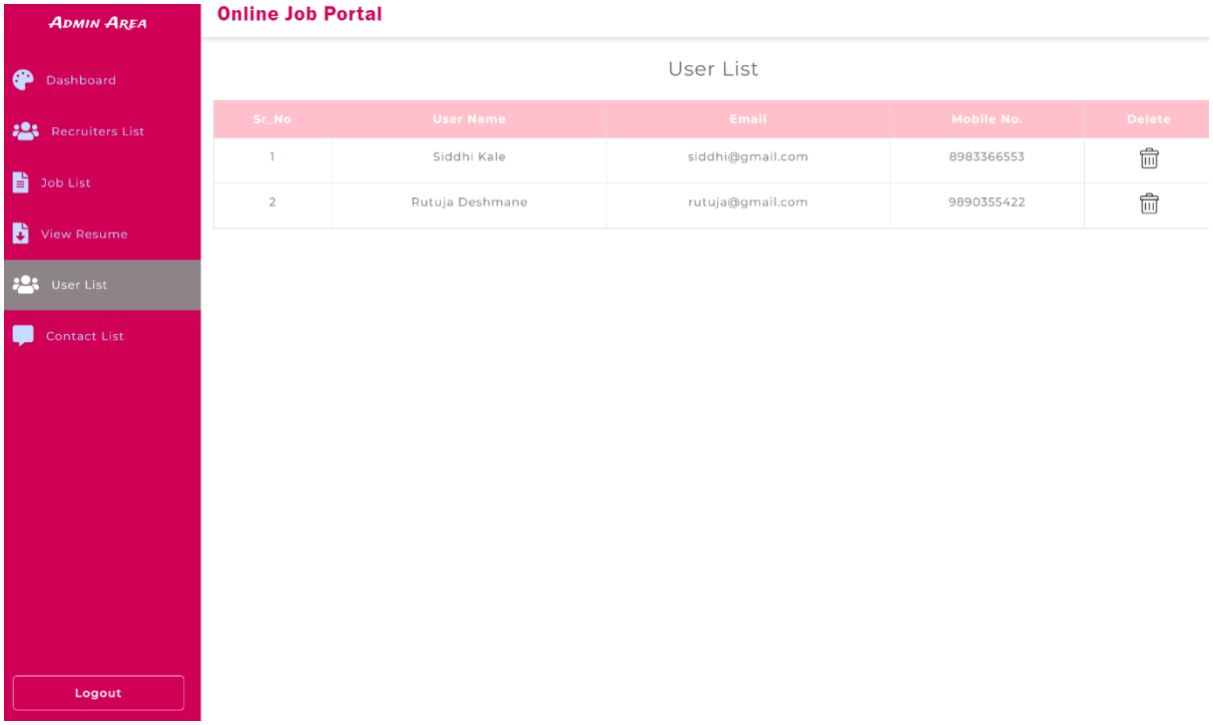
Admin Dashboard :



Recruiters list :



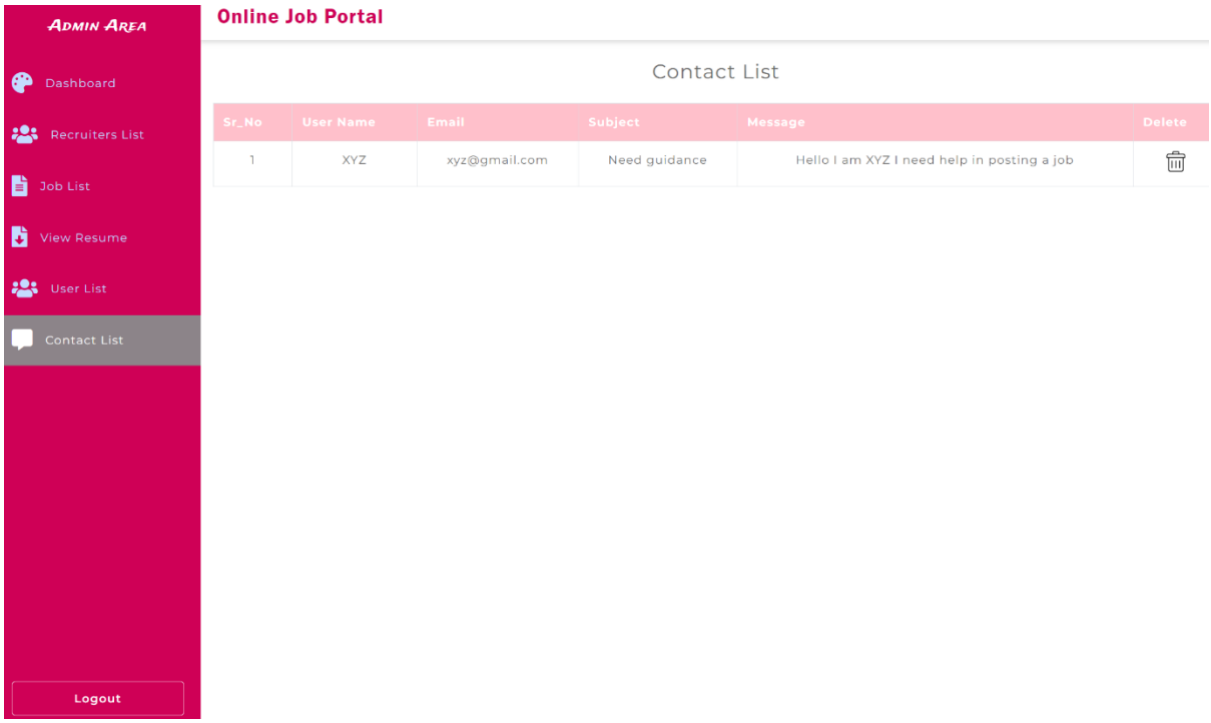
Jobseekers list :



Job List :

View Resume :

Contact List :



4.2) Project Code:

Login code:

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplicationChanges.User
{
   public partial class login : System.Web.UI.Page
   {
      MySqlConnection con;
      MySqlCommand cmd;
      MySqlDataReader sdr;
      string str = ConfigurationManager.ConnectionStrings["LocalMysql"].ConnectionString;
      string username, password = string.Empty;
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void btnLogin_Click(object sender, EventArgs e)
      {
         if (Page.IsValid)
         {
            try
            {
               if (ddlLoginType.SelectedValue == "Admin")
               {
                  username = ConfigurationManager.AppSettings["username"];
                  password = ConfigurationManager.AppSettings["password"];
                  if (username == txtUserName.Text.Trim() && password ==
txtPassword.Text.Trim())
                  {
                     Session["admin"] = username;
                     Response.Redirect("../Admin/admindashboard.aspx", false);
                  }
                  else
                  {
                     showErrorMsg("Admin");
                  }

               }
```

```
            else if(ddlLoginType.SelectedValue == "User")
            {
                con = new MySqlConnection(str);
                string query = "Select * from User where Username=@Username and
Password=@Password";
                cmd = new MySqlCommand(query, con);

                cmd.Parameters.AddWithValue("@Username", txtUserName.Text.Trim());
                cmd.Parameters.AddWithValue("@Password", txtPassword.Text.Trim());


                con.Open();

                sdr = cmd.ExecuteReader();
                if (sdr.Read())
                {
                    Session["user"] = sdr["Username"].ToString();
                    Session["userId"] = sdr["UserId"].ToString();
                    Response.Redirect("default.aspx", false);


                }
                else
                {
                    showErrorMsg("User");
                }
                con.Close();
            }
            else
            {
                con = new MySqlConnection(str);
                string query = "Select * from Recruiter where Username=@Username and
Password=@Password";
                cmd = new MySqlCommand(query, con);

                cmd.Parameters.AddWithValue("@Username", txtUserName.Text.Trim());
                cmd.Parameters.AddWithValue("@Password", txtPassword.Text.Trim());


                con.Open();

                sdr = cmd.ExecuteReader();
                if (sdr.Read())
                {
                    Session["recruiter"] = sdr["Username"].ToString();
                    Session["recruiterId"] = sdr["UserId"].ToString();
                    Response.Redirect("../Recruiter/recruiterdashboard.aspx", false);


                }
```

```
                else
                {
                    showErrorMsg("Recruiter");
                }
                con.Close();

            }

        }

        catch (Exception ex)
        {
            Response.Write("<script>alert('" + ex.Message + "');</script>");
            con.Close();


        }

    }

}

private void showErrorMsg(string userType)
{
    lblMsg.Visible = true;
    lblMsg.Text = "<b>" + userType + "</b> Credentials are incorrect...";
    lblMsg.CssClass = "alert alert-danger";
}
}
}
```

Register Code:

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplicationChanges.User
{
    public partial class register : System.Web.UI.Page
    {
```

```
        MySqlConnection con;
        MySqlCommand cmd;
        string str = ConfigurationManager.ConnectionStrings["LocalMysql"].ConnectionString;
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnRegister_Click(object sender, EventArgs e)
        {
           if (Page.IsValid) // Check if all validations pass
           {
              try
              {
                 con = new MySqlConnection(str);
                 if (ddlRegisterType.SelectedValue =="User")
                 {
                    string query = "insert into User
(Username,Password,Name,Address,Mobile,Email) values
(@Username,@Password,@Name,@Address,@Mobile,@Email)";

                    cmd = new MySqlCommand(query, con);

                    cmd.Parameters.AddWithValue("@Username", txtUserName.Text.Trim());
                    cmd.Parameters.AddWithValue("@Password",
txtConfirmPassword.Text.Trim());
                    cmd.Parameters.AddWithValue("@Name", txtFullName.Text.Trim());
                    cmd.Parameters.AddWithValue("@Address", txtAddress.Text.Trim());
                    cmd.Parameters.AddWithValue("@Mobile", txtMobile.Text.Trim());
                    cmd.Parameters.AddWithValue("@Email", txtEmail.Text.Trim());
                 }
                 else
                 {
                    string query = "insert into Recruiter
(Username,Password,Name,Address,Mobile,Email) values
(@Username,@Password,@Name,@Address,@Mobile,@Email)";

                    cmd = new MySqlCommand(query, con);

                    cmd.Parameters.AddWithValue("@Username", txtUserName.Text.Trim());
                    cmd.Parameters.AddWithValue("@Password",
txtConfirmPassword.Text.Trim());
                    cmd.Parameters.AddWithValue("@Name", txtFullName.Text.Trim());
                    cmd.Parameters.AddWithValue("@Address", txtAddress.Text.Trim());
                    cmd.Parameters.AddWithValue("@Mobile", txtMobile.Text.Trim());
                    cmd.Parameters.AddWithValue("@Email", txtEmail.Text.Trim());
                 }

                 con.Open();
```

```
            int r = cmd.ExecuteNonQuery();
            if (r > 0)
            {
                lblMsg.Visible = true;
                lblMsg.Text = "Registered Successfully.";
                lblMsg.CssClass = "alert alert-success";
                clear();
            }
            else
            {
                lblMsg.Visible = true;
                lblMsg.Text = "Cannot save record right now, Please try after sometime";
                lblMsg.CssClass = "alert alert-danger";

            }
        }


        catch (MySqlException ex)
        {
            if (ex.Number == 1062) // MySQL error number for duplicate entry
            {
                lblMsg.Visible = true;
                lblMsg.Text = "The username '" + txtUserName.Text.Trim() + "' already
exists. Please choose a different username.";
                lblMsg.CssClass = "alert alert-danger";
            }
            else
            {
                Response.Write("<script>alert('" + ex.Message + "');</script>");
            }
        }
        catch (Exception ex)
        {
            Response.Write("<script>alert('" + ex.Message + "');</script>");


        }
        finally
        {
            con.Close();
        }
    }
    else
    {
        lblMsg.Visible = true;
        lblMsg.Text = "Check the entries...";
        lblMsg.CssClass = "alert alert-danger";
    }
}
```

```
        private void clear()
        {
            txtUserName.Text = String.Empty;
            txtConfirmPassword.Text = String.Empty;
            txtFullName.Text = String.Empty;
            txtAddress.Text = String.Empty;
            txtMobile.Text = String.Empty;
            txtEmail.Text = String.Empty;
            txtPassword.Text = String.Empty;


        }
    }
}
```

ViewResume Code:

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplicationChanges.Admin
{
    public partial class viewresume : System.Web.UI.Page
    {
        MySqlConnection con;
        MySqlCommand cmd;
        DataTable dt;
        string str = ConfigurationManager.ConnectionStrings["LocalMysql"].ConnectionString;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["admin"] == null)
            {
                Response.Redirect("../User/login.aspx");
            }
            if (!IsPostBack)
            {
                ShowAppliedJob();
            }
        }
        private void ShowAppliedJob()
```

```
        {
            string query = string.Empty;
            con = new MySqlConnection(str);
            query = @"SELECT
  ROW_NUMBER() OVER (ORDER BY aj.AppliedJobId) AS Sr_No,
  aj.AppliedJobId,
  j.CompanyName,
  aj.JobId,
  j.Title,
  u.Mobile,
  u.Name,
  u.Email,
  u.Resume
FROM AppliesJobs aj
INNER JOIN Jobs j ON aj.JobId=j.JobId
INNER JOIN User u ON aj.UserId=u.UserId;";
            cmd = new MySqlCommand(query, con);

            MySqlDataAdapter sda = new MySqlDataAdapter(cmd);
            dt = new DataTable();
            sda.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();

        }



        protected void GridView1_PageIndexChanging1(object sender,
GridViewPageEventArgs e)
        {
            GridView1.PageIndex = e.NewPageIndex;
            ShowAppliedJob();
        }

        protected void GridView1_RowDeleting1(object sender, GridViewDeleteEventArgs e)
        {
            try
            {
                GridViewRow row = GridView1.Rows[e.RowIndex];
                int appliedJobId = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);

                con = new MySqlConnection(str);
                cmd = new MySqlCommand("delete from AppliesJobs where AppliedJobId=@id",
con);
                cmd.Parameters.AddWithValue("@id", appliedJobId);
                con.Open();
                int r = cmd.ExecuteNonQuery();
                if (r > 0)
                {
```

```
            lblMsg.Text = "Resume details deleted successfully";
            lblMsg.CssClass = "alert alert-success";


        }
        else
        {
            lblMsg.Text = ("Cannot delete the record");
            lblMsg.CssClass = "alert alert-danger";

        }
        GridView1.EditIndex = -1;
        ShowAppliedJob();

    }
    catch (Exception ex)
    {

        Response.Write(ex.Message);
    }
    finally
    {
        con.Close();

    }
}

protected void GridView1_RowDataBound1(object sender, GridViewRowEventArgs e)
{

    e.Row.Attributes["onCLick"] =
Page.ClientScript.GetPostBackClientHyperlink(GridView1, "Select$" + e.Row.RowIndex);
    e.Row.ToolTip = "Click to view job details";
}

protected void GridView1_SelectedIndexChanged1(object sender, EventArgs e)
{
    foreach (GridViewRow row in GridView1.Rows)
    {
        if (row.RowIndex == GridView1.SelectedIndex)
        {
            HiddenField jobId = (HiddenField)row.FindControl("hdnJobId");
            Response.Redirect("JobList.aspx?id" + jobId.Value);
        }
        else
        {
            row.BackColor = ColorTranslator.FromHtml("#ffffff");
            row.ToolTip = "Click to select this row";
        }
    }
```

```
      }
   }
}
```

5) Limitations and Enhancement

Limitations :

As with any project, there may be some limitations and drawbacks to an online job portal. Here are some potential areas to consider:

- Limited Reach: Although online job portals have a wider reach than traditional job search methods, they may still have limitations in terms of geographic coverage, industry-specific jobs, or access to certain types of jobs.

- Competition: There are many other online job portals available, which means that it may be difficult to stand out from the competition. This could impact the number of users and job postings on your site.

- No Screening : One limitation of online job portals is the lack of screening of job applicants. Since anyone can apply for a job with just a few clicks, there is no guarantee that the applicants possess the necessary qualifications or experience for the job. This can lead to a large number of irrelevant or unqualified applications, making it difficult for employers to find the right candidate. Additionally, it can be time-consuming for employers to sift through large numbers of applications to find suitable candidates, which can be a significant drawback of online job portals.

Enhancements :

There is ample scope of enhancement and adding functionalities to this application.

This application can be extended to send automated interview scheduling through acceptance/rejection of Resume. Companies can delete jobs once the job availably period is over automatically. The application can have a job recommendation system based on the frequent search results of different users. The portal can also send email notifications to candidates about certain job availabilities. There can be a feedback or review section for the application.

The User functionality can be extended to give the user options to save the job and later apply, to upload multiple documents. The application can be more scalable by extending the search functionality based on country, city or area.

While this application meets the basic requirements of a job portal eliminating few of the traditional challenges faced like time, money and effort, it can be extended to make the application more dynamic and robust. The User Interface can be made more attractive and user friendly. We can dig through more AngularJS magic capabilities to add additional features to the UI.

5.3) Conclusion :

In conclusion, the online job portal project is a useful platform that provides easy access to job listings for job seekers and facilitates the recruitment process for employers. The system has a user-friendly interface, making it easy to navigate and search for job listings. With features such as resume creation and job application submission, the system streamlines the hiring process and increases efficiency for both job seekers and employers. However, the system has limitations, including the lack of screening, which could result in unqualified applicants applying for jobs. Despite these limitations, the online job portal remains a valuable tool for job seekers and employers alike, and its implementation can greatly benefit college students searching for job opportunities.

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in ASP.NET web based application and MYSQL, but also about all handling procedures related with online job portal.

It also provided knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

5.4) Bibliography:

- ASP.NET Documentation. Microsoft. https://docs.microsoft.com/en-us/aspnet/
- C# Programming Guide. Microsoft. https://docs.microsoft.com/en-us/dotnet/csharp/
- MySQL Documentation. Oracle. https://dev.mysql.com/doc/
- Online tutorials, guides, and documentation on ASP.NET, C#, and MySQL