



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech. CE Semester – VI

Subject: Service Oriented Computing

Project Title:

Job Portal System

By:

Ashish Kotecha (Roll No: CE064 Id:17CEUOS066)

Harsh Moradiya (Roll No: CE072 Id:17CEUOG056)

Guided By:

Prof. Ankit P. Vaishnav

CERTIFICATE



This is to certify that Service Oriented Computing project entitled “Job Portal” is the bonafide report of work carried out by

1) Ashish Kotecha (17CEUOS066)

2) Harsh Moradiya (17CEUOG056)

Of Department of Computer Engineering, Semester VI, academic year 2019-20, under our supervision and guidance.

Guide

Prof. Ankit P. Vaishnav

Assitant Professor of
Department of Computer
Engineering, Dharmsinh
Desai University, Nadiad.

HOD

Dr. C. K. Bhensdadia

Head of the Department of
Department of Computer
Engineering, Dharmsinh
Desai University, Nadiad.

Table of Contents

1. Abstract	4
2. Introduction	5
3. Software Requirements Specification	6
4. Design	11
5. Implementation	18
6. Testing	23
7. Screen-shots	25
8. Limitation and Future Extension	31
9. Conclusion	32
10. Bibliography	33

1. Abstract

This project is aimed at developing a web site for Online Job Portal. It is of great importance to the unemployed people. This site allocates the right job information for suitable persons. All the jobseekers as well as job providers information are stored in the database. Digital Job Portal is an application which connects employer and job seekers where employers are the source of the resources and the job seeker can find and apply for their targeted job. This document provides details about the entire software requirement specification for the online job portal.

2. Introduction

2.1 Brief Introduction

Job portal is wcf service through which you can connect any company with user. And as it is wcf service it is reusable component.

This system can be used as an Online Job Portal for the Placements providing to the un employees who are seeking for a job placement. Job Seeker logging into the system and he can should be able to upload their information in the form.

Visitors/Company representatives logging in may also access/search any information put up by Job Seeker. Using this information any company can find appropriate applicants and can hire them .So this service provides a better medium for company and user who want to find a job.

2.2 Tools/Technologies used

Technologies:

- 1) .Net Framework
- 2) WCF service
- 3) SQL Server Database

Tools:

Visual Studio

3. Software Requirement Specifications

3.1 Types of User

1. Company
2. End-User(Job Seekers)
3. Admin

3.2 System Function Requirement

R.1 Company Functionalities:

R.1.1 Post Job Details

Description: Company can register details about job title, vacancy, qualification and description about particular job.

Input: Details of job title, vacancy, qualification and description

Output: Success message.

R.1.2 View All applied user

Description: Company can view details of user by company.

Input: Company name.

Output: Display user information.

R.1.3 View User Application for job

Description: Company can view details of user by company.

Input: Job Id for particular job.

Output: Display user information.

R.1.3 Update Job Details

Description: Company can update details of job.

Input: Updated Details of job title, vacancy, qualification and description.

Output: Update Message.

R.1.4 Delete Job Details

Description: Company can delete details of job.

Input: Job id.

Output: Deleted Message.

R.2 User functionalities

R.2.1 Apply For Job

Description: Users can apply for a job in selected company for a particular job.

Input: Post name and Company name.

Output: Successfully Applied message.

R.2.2 Display Jobs

Description: User can view postwise job for a company

Input: Company Name

Output: Display different jobs.

R.2.3 Display Applied jobs

Description: User can view all applied jobs for all company.

Input: Username

Output: All applied jobs.

R.2.4 Withdraw applied jobs

Description: User can delete application for any post that he/she applied.

Input: User selection

Output: Withdraw job message.

R.3 Admin functionalities:

R.3.1 View All Job seekers

Description: Admin can view details of all jobseeker.

Input: User selection

Output: All Users details.

R.3.2 View All Company

Description: Admin can view details of all company.

Input: User selection

Output: All Company details.

R.3.3 Delete Job seekers

Description: Admin can delete any jobseeker.

Input: User name

Output: Deleted message.

R.3.4 Delete Company

Description: Admin can delete any company.

Input: Company name

Output: Deleted message.

R.3.5 View All Jobs

Description: Admin can view details of all posted job by company.

Input: User selection

Output: All Company Job details.

R.3.6 View All Job Application

Description: Admin can view details of all job application of jobseekers.

Input: User selection

Output: All Job Applications.

Common Functionalities (User, Admin, Company):

R.4 User Authentication-Sign Up

Input: User Details

Output: Data Stored Successfully

Description: User Enters Details Like User Id, Name, Password, Phone Number, E-Mail As Per Type Of User.

R.5 User Login

Input: User Credentials

Output: User Logged In Account/Error Message

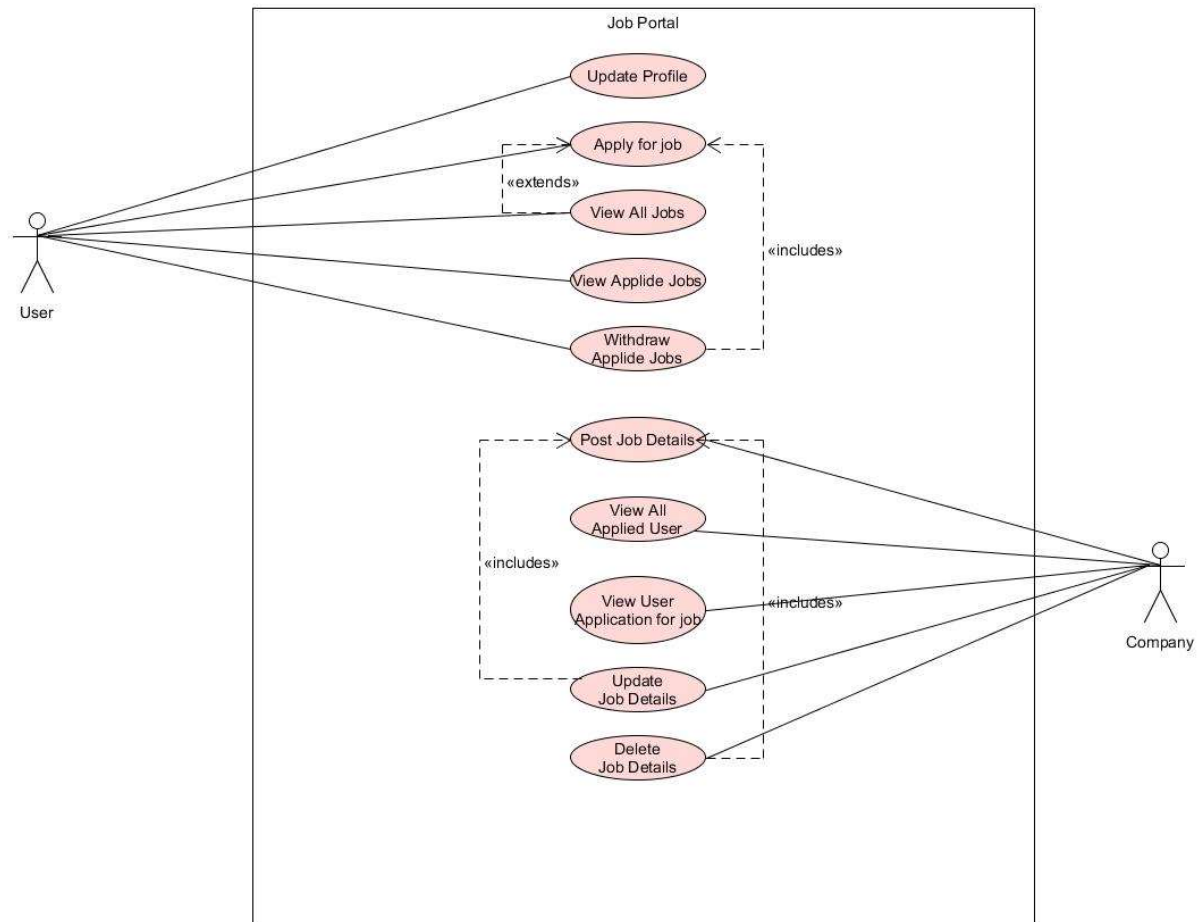
Description: User Enters The Username/Email And Password And Checks Into The Web Application By Validating In Database.

R.6 Log Out

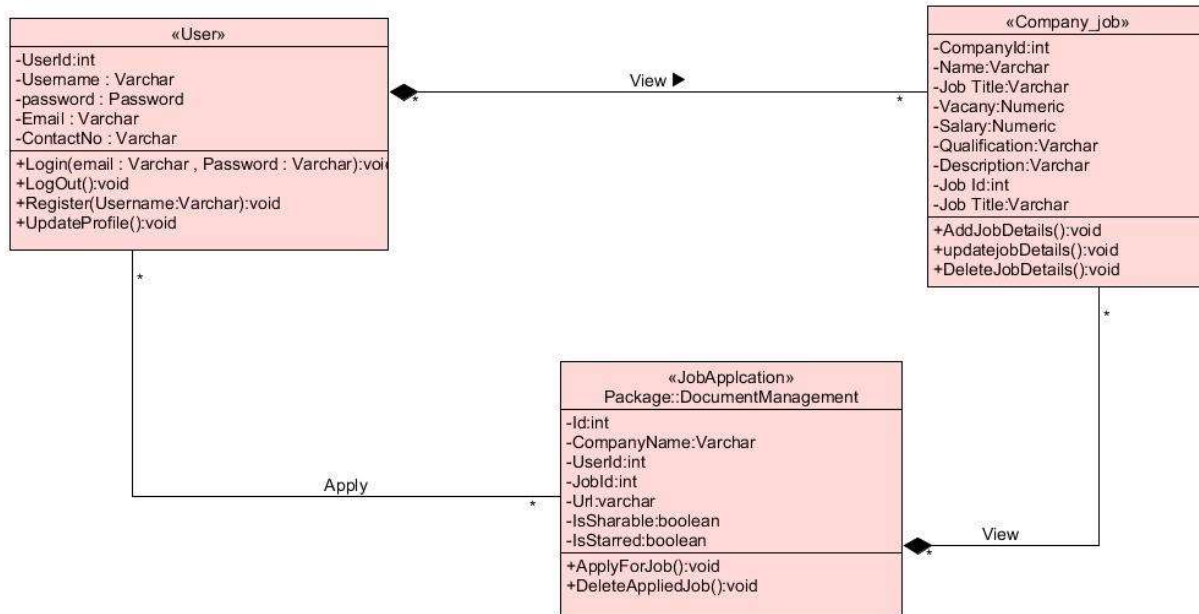
Description: User Logs Out Of The Website.

4. Designs

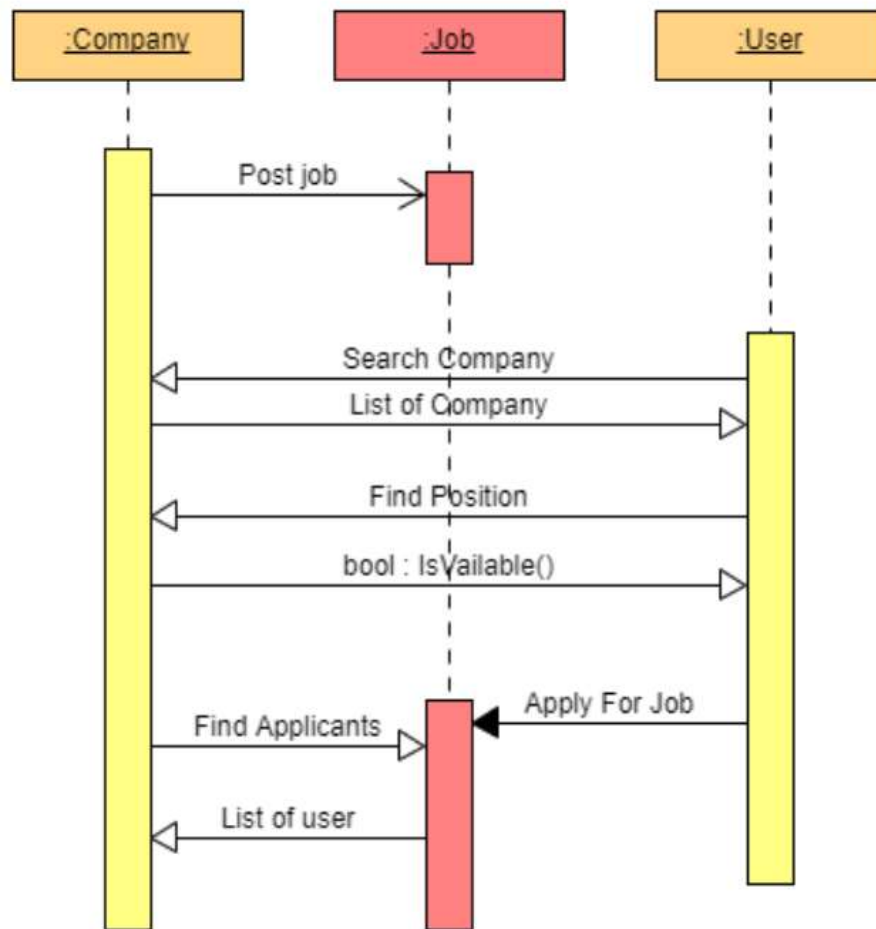
Use case Diagram:



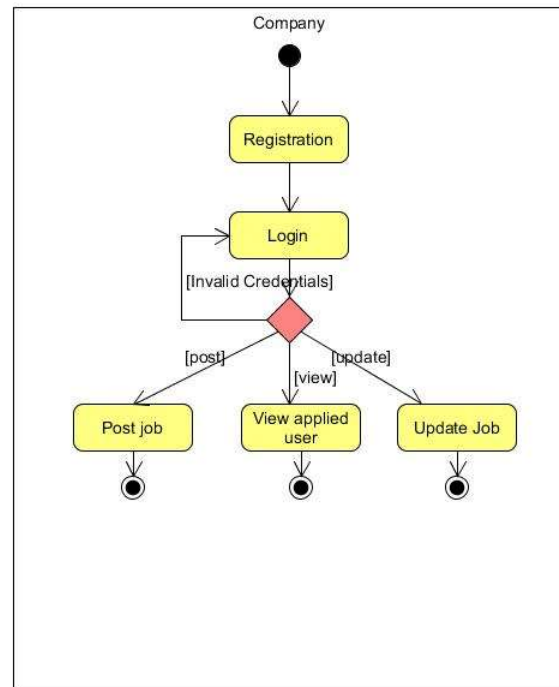
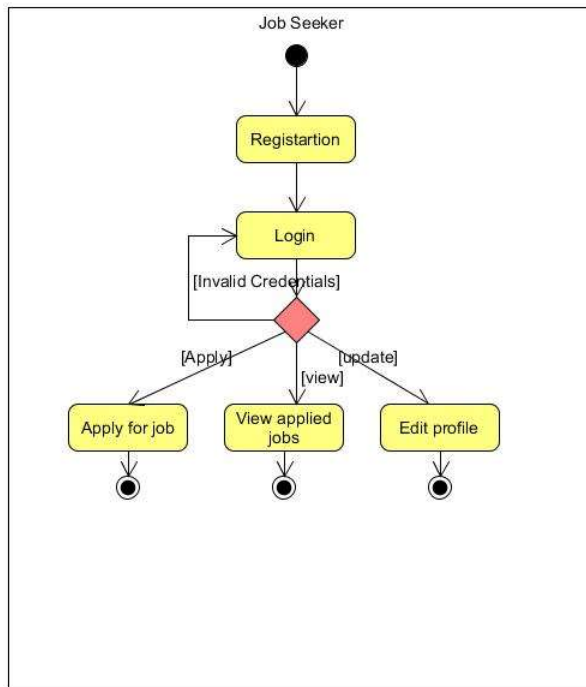
Class diagrams



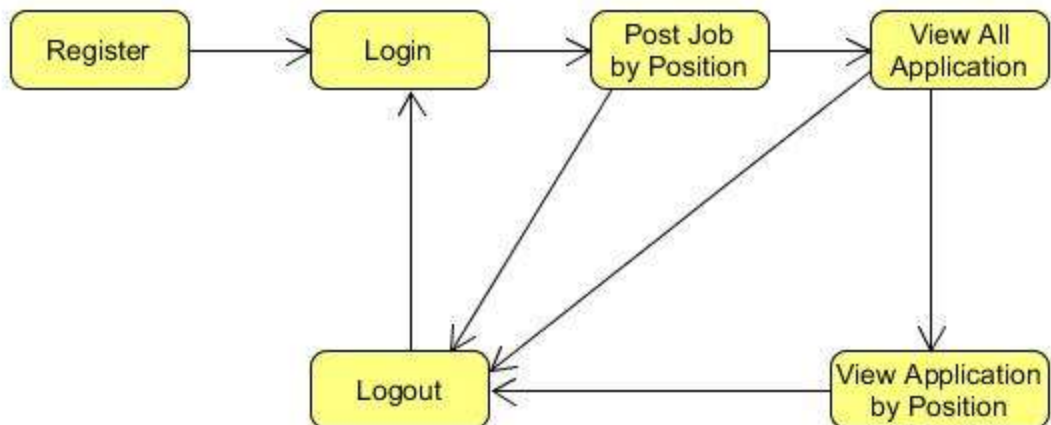
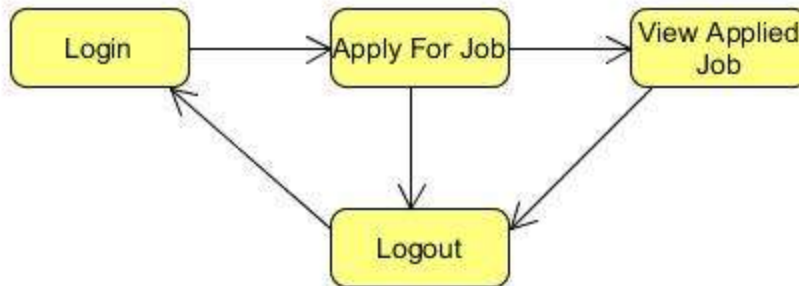
Sequence diagrams



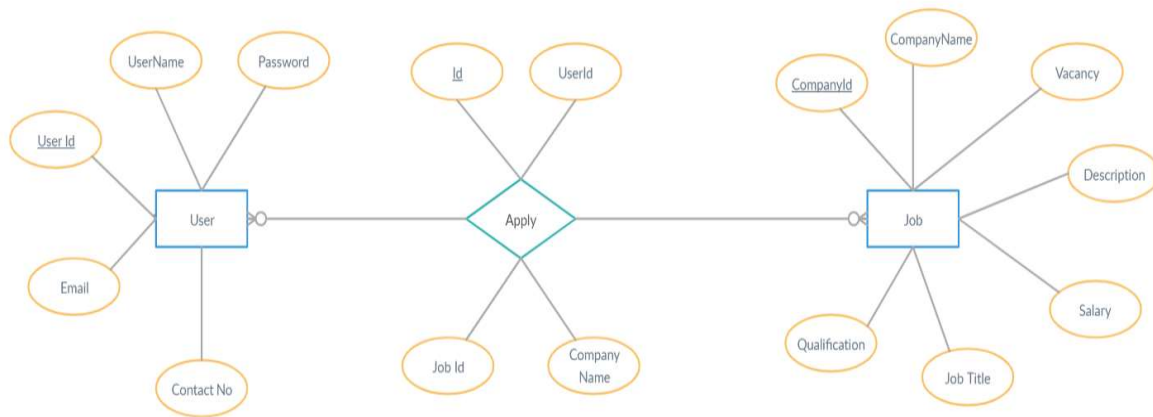
Activity diagrams:



State diagrams:



E-R Diagram:



Data Dictionary:

USER							
SR. NO	NAME	DATA TYPE	WIDTH	REQUIRED	UNIQUE	PK/ FK	REFERENCE TABLE DESCRIPTION
1	USERID	NUMERIC	20	YES	YES	PK	
2	PASSWORD	VARCHAR	20	YES	NO		
3	USERNAME	VARCHAR	20	YES	NO		
4	EMAIL	VARCHAR	20	YES	YES		
5	CONTACT NO	VARCHAR	20	YES	YES		

COMPANY							
SR. NO	NAME	DATA TYPE	WIDTH	REQUIRED	UNIQUE	PK/ FK	REFERENCE TABLE DESCRIPTION
1	ID	NUMERIC	20	YES	YES	PK	
2	NAME	VARCHAR	20	YES	NO		
3	JOB TITLE	VARCHAR	20	YES	NO		
4	JOB ID	VARCHAR	20	YES	NO		
5	VACANCY	NUMERIC	20	YES	NO		
6	SALARY	NUMERIC	20	YES	NO		
7	QUALIFICATION	VARCHAR	20	YES	NO		
8	DESCRIPTION	VARCHAR	20		NO		

JOB_APPLICATION							
SR. NO	NAME	DATA TYPE	WIDTH	REQUIRED	UNIQUE	PK/ FK	REFERENCE TABLE DESCRIPTION
1	ID	NUMERIC	20	YES	YES	PK	
2	USERNAME	VARCHAR	20	YES	NO		
3	COMPANY NAME	VARCHAR	20	YES	NO		
4	JOB ID	VARCHAR	20	YES	NO		

5. Implementation Details

1. Modules

- **User Module:**

User module is basically manage all the activity of users. In this module user can register and login by giving some basic details.

User can view his profile and can update it. User can view all the job post which is posted by company by position wise and user also can applied for that job User can see all the applied post. User can withdraw application from any position and also can update his/her profile .

- **Company Module:**

This module handles all the activity of the company. Company itself can register and can post a job by giving a job id and post name and other details . Any company can view all the users who have applied for any position which is currently open in that company. And if the company wants to withdraw that position opening then he can also withdraw it and also it can modify some of the details.

- **Admin Module:**

This module handles all the activity of the Admin. Admin can view all registered users as well as registered company. Also Admin can view details all job posted by company. Admin can also view all job application of all users. Admin can also delete the user as well as company.

2. Function Prototype

- Login:

```
string Login(User u);
```

- Register:

```
string AddUser(User u);
```

Users:

- Display All Jobs:

```
DataSet GetCompanyData();
```

- Apply to Job

```
string ApplyForCompany(JobApplication a);
```

- Update User:

```
string UpdateUser(User user);
```

- Withdraw Job:

```
string DeleteAppliedPosition(JobApplication a);
```

- Display Applied Jobs:

```
DataSet DisplayAppliedPost(string user);
```

Company:

- Add job details:

```
string InsertData(CompanyDetails c);
```

- Display User by Company

```
DataSet SelectUserbyCompany(string CompanyName);
```

- Display User job title wise for company:

```
DataSet SelectUserbyPostCompany(string CompanyName, string jobid);
```

- Update posted job details:

```
string UpdatePostedJob(CompanyDetails c, int id);
```

- Delete posted job details:

```
string DeletePostedJob(string name, string postname);
```

Admin:

- View User by Company

```
DataSet GetUsers();
```

- View User job title wise for company:

```
DataSet GetCompany();
```

- View All Job Application Details:

```
DataSet GetJobApplicationDetails();
```

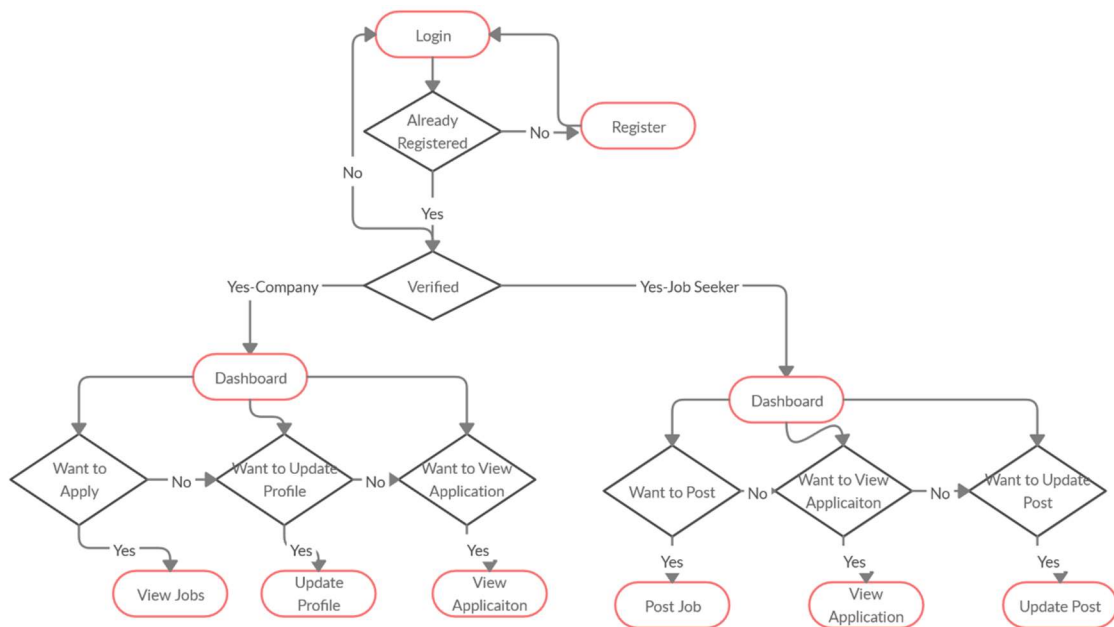
- Delete User:

```
string DeleteUser(string username);;
```

- Delete Company:

```
string DeleteCompany(string username);
```

3. Flow Chart



6. Testing

We performed a testing through the wcf test client. All the methods are tested on this test client . We try to test by using various test case and try to make our service as robust as we can We also try unit test on seperate method and then we perform integration testing by merging related method and perform test using various test cases. Then at last we have performed a system testing of whole service using client web app. For unit testing we used a test client and for integration testing and system testing we used a web client .

If user is already registered then system will shown error message “Username already exists”.

HOME ABOUT SERVICES CONTACT

Sign Up

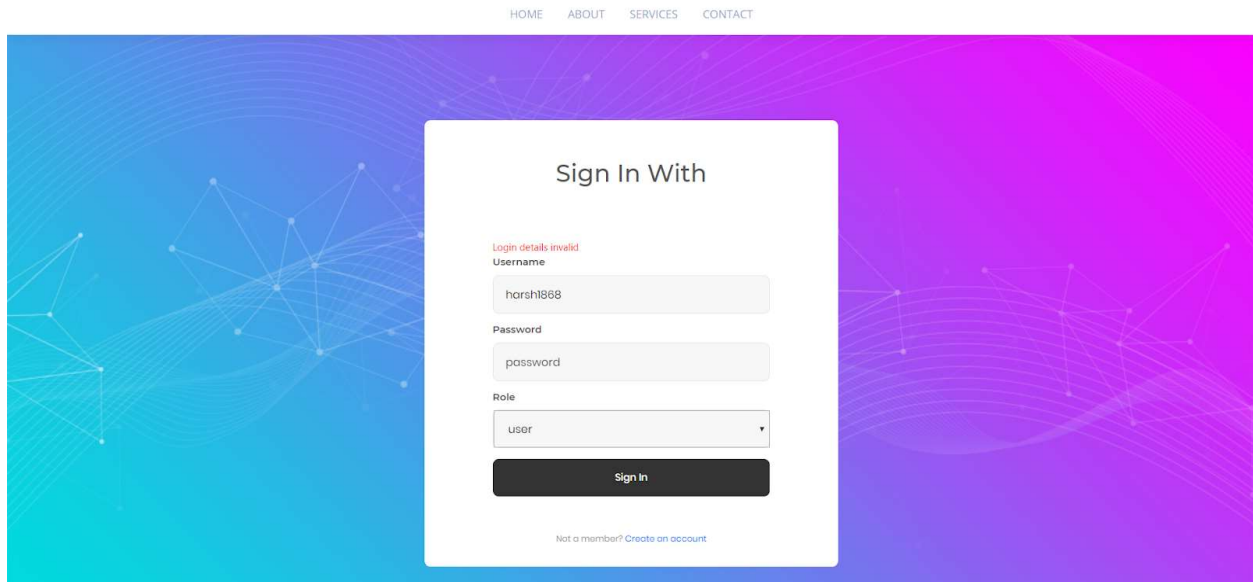
Username **Username already exists**

Email

Password

If user tries to login with wrong details then system will throw error “login invalid”.

HOME ABOUT SERVICES CONTACT



Sign In With

Login details invalid

Username

horsh888

Password

password

Role

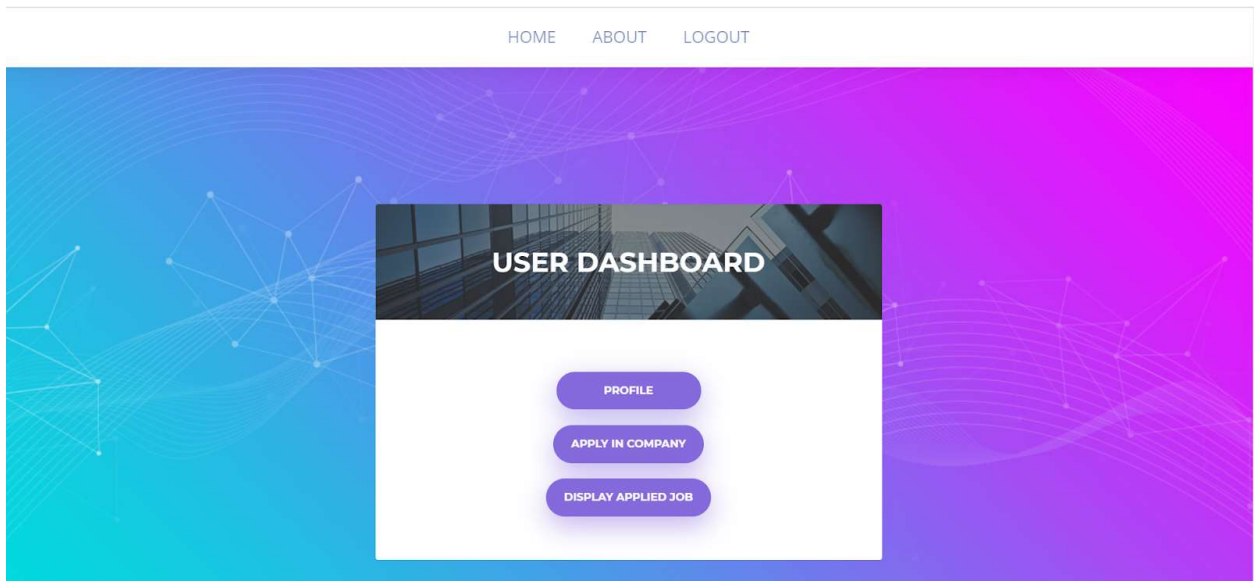
user

Sign In

Not a member? [Create an account](#)

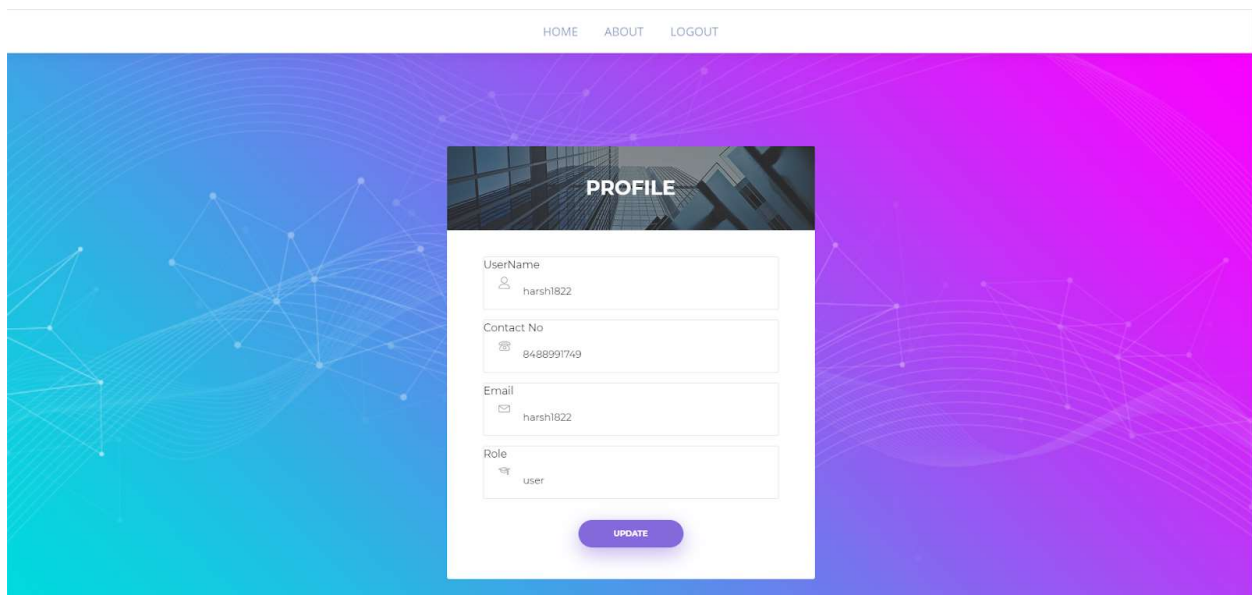
7. Screenshot

1. User Dashboard:



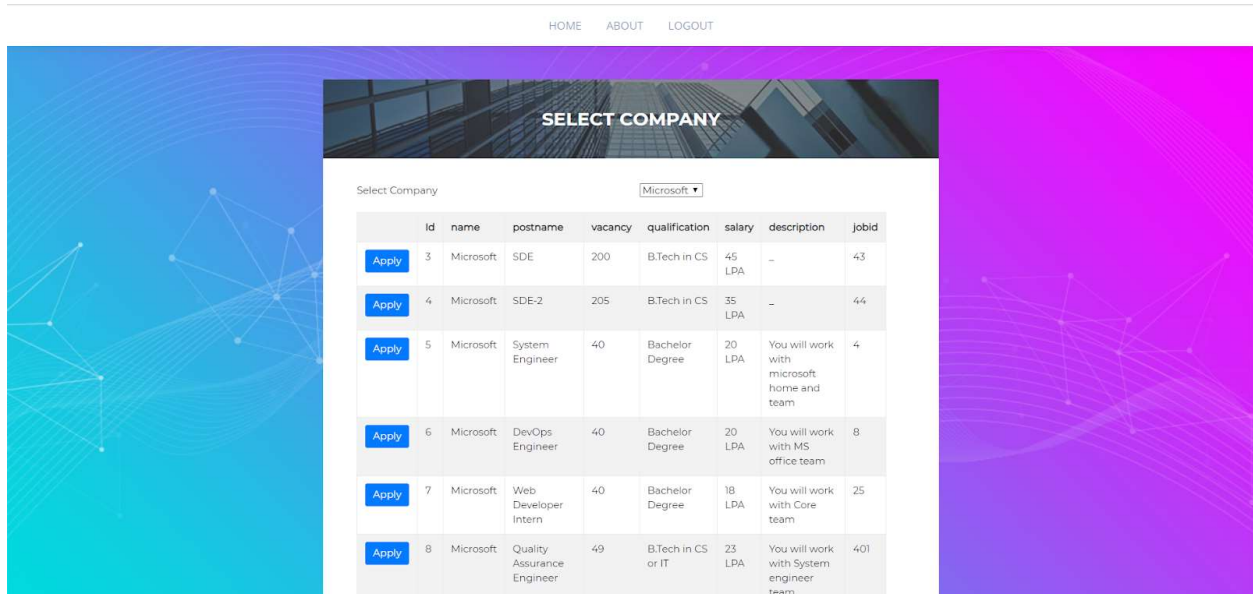
After login as user, user can see dashboard.

2. Update profile:



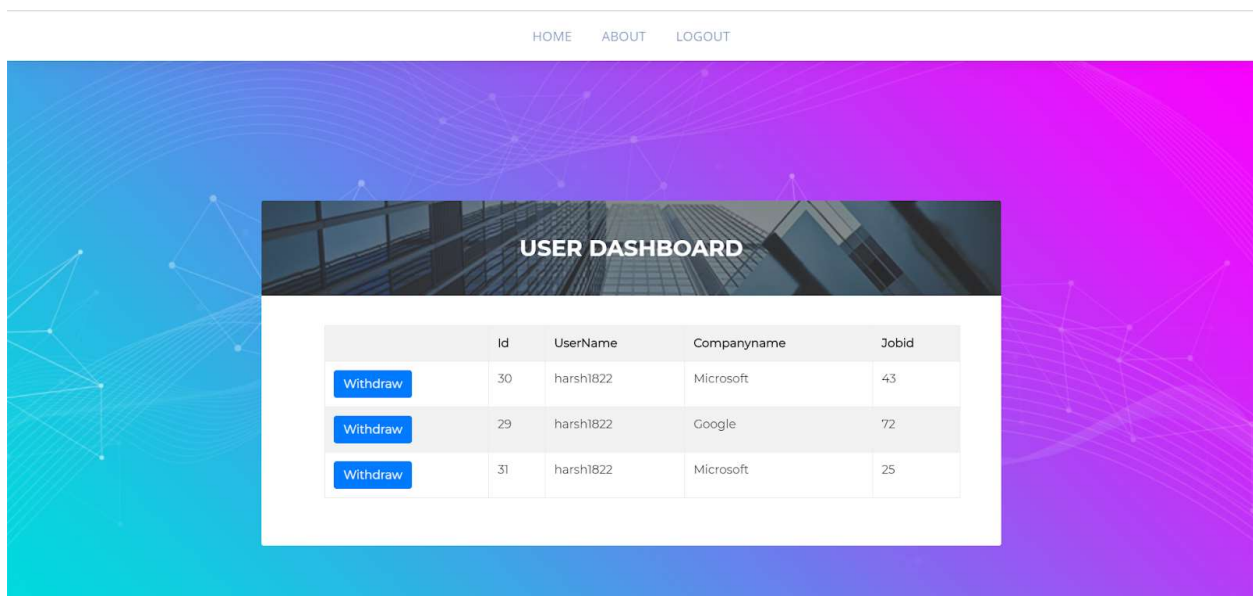
User can update his/her profile after successful login.

3. View/Apply Job:



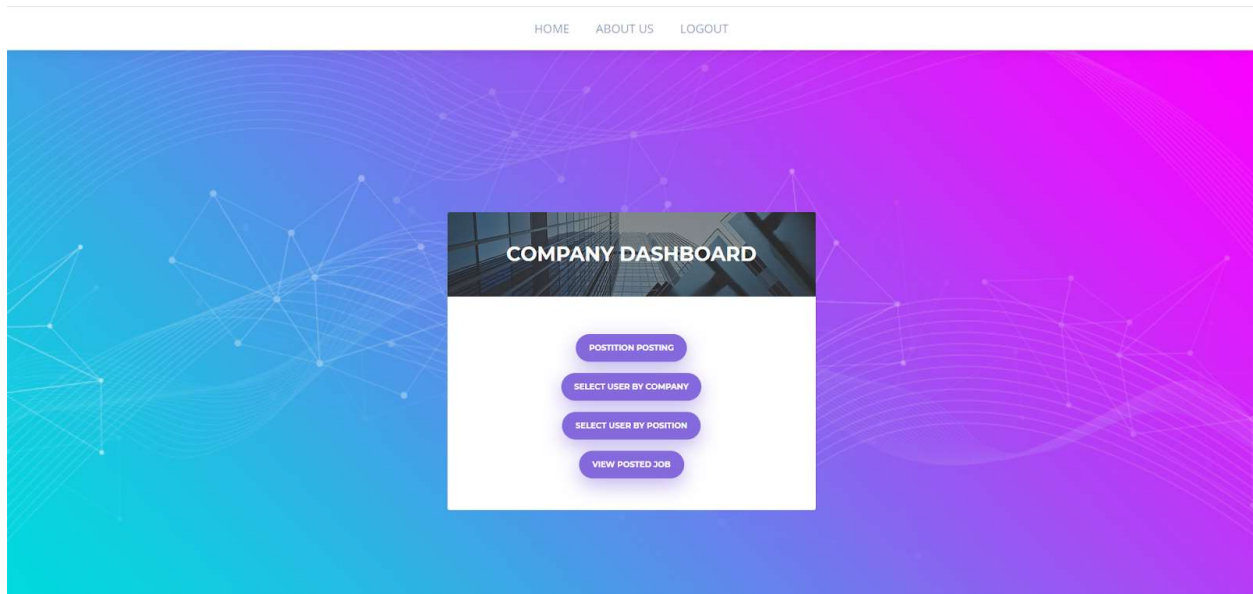
After clicking on Apply in company user can select company and view/apply for job.

4. View Applied Job:



After clicking on display applied job user can see all application made by him/her for particular job.

5. Company Dashboard:



After successful login as company, company can see dashboard.

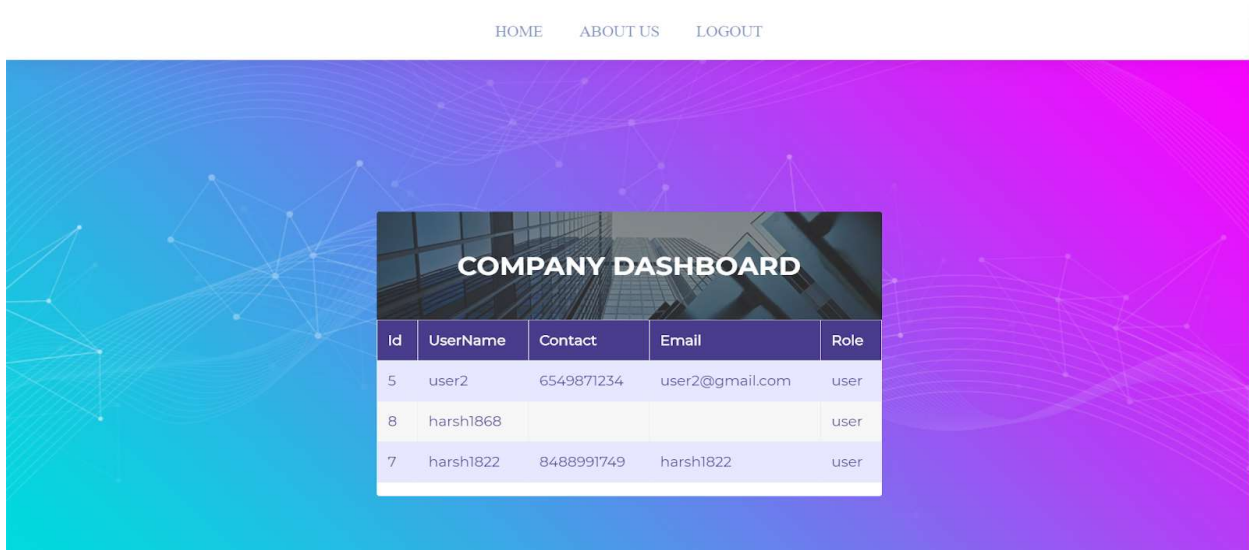
6. Post Job Details:

The screenshot displays a web application interface for posting a job. The top navigation bar includes 'HOME', 'ABOUT US', and 'LOGOUT'. The background features a blue-to-purple gradient with a network pattern. A central white box has a dark header labeled 'JOB POSTING'. Below the header, there are several input fields with icons and example text: 'JobId' (Eg. 72), 'Post Name' (Eg. SDE), 'Total Vacancy' (Eg. 101), 'Qualification' (Eg. B.tech in Cs, M.tech), and 'Expected Salary' (Eg. 18,00,000). At the bottom of the form is a 'Message' field with the placeholder 'Your comments...' and a purple 'POST' button.

Here company can post the description of job.

7. View Job Application:

HOME ABOUT US LOGOUT



The dashboard features a header with navigation links: HOME, ABOUT US, and LOGOUT. The main content area is titled 'COMPANY DASHBOARD' and displays a table of users. The table has five columns: Id, UserName, Contact, Email, and Role. It lists three users with their respective details.

Id	UserName	Contact	Email	Role
5	user2	6549871234	user2@gmail.com	user
8	harsh1868			user
7	harsh1822	8488991749	harsh1822	user

After clicking on select user by position, company can view all its applicants for particular job.

8. View/Update/Delete Posted Job:

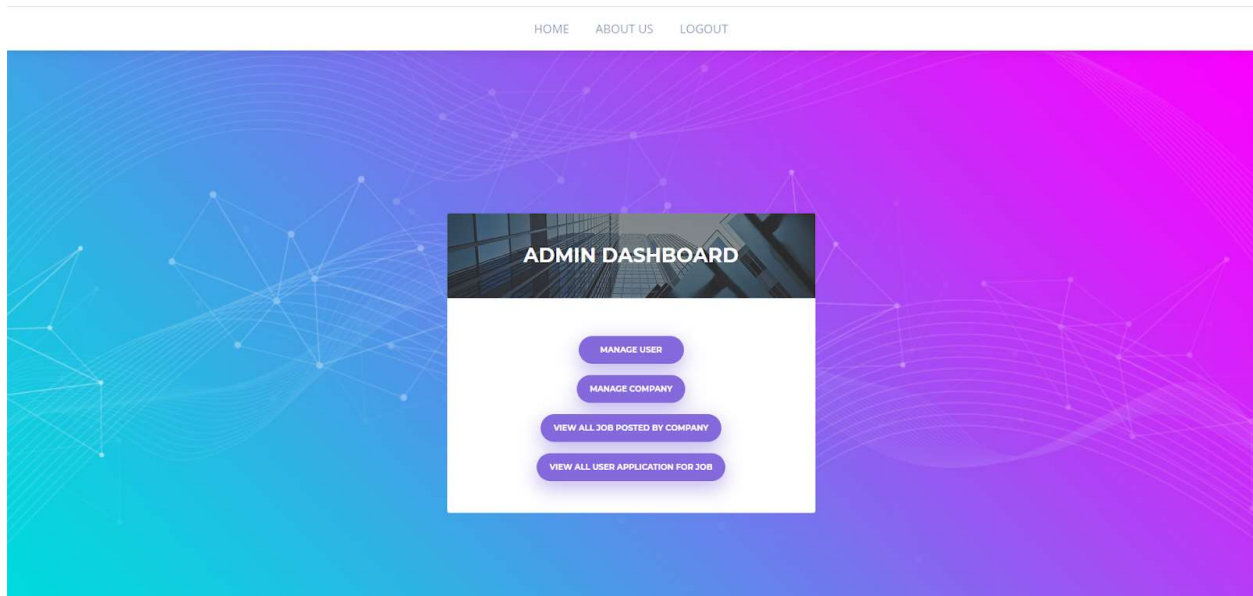


The dashboard features a header with navigation links: HOME, ABOUT US, and LOGOUT. The main content area is titled 'COMPANY DASHBOARD' and displays a table of posted jobs. Each job row includes 'Withdraw' and 'Update' buttons. The table has ten columns: Id, name, postname, vacancy, qualification, salary, description, and jobid.

		Id	name	postname	vacancy	qualification	salary	description	jobid
Withdraw	Update	3	Microsoft	SDE	200	B.Tech in CS	45 LPA	-	43
Withdraw	Update	4	Microsoft	SDE-2	205	B.Tech in CS	35 LPA	-	44
Withdraw	Update	5	Microsoft	System Engineer	40	Bachelor Degree	20 LPA	You will work with microsoft home and team	4
Withdraw	Update	6	Microsoft	DevOps Engineer	40	Bachelor Degree	20 LPA	You will work with MS office team	8
Withdraw	Update	7	Microsoft	Web Developer Intern	40	Bachelor Degree	18 LPA	You will work with Core team	25
Withdraw	Update	8	Microsoft	Quality Assurance Engineer	49	B.Tech in CS or IT	23 LPA	You will work with System engineer team	401
Withdraw	Update	12	Microsoft	Software Developer Engineer - 3	103	B.Tech in CS or IT	41 LPA	You will work with our core team.	723

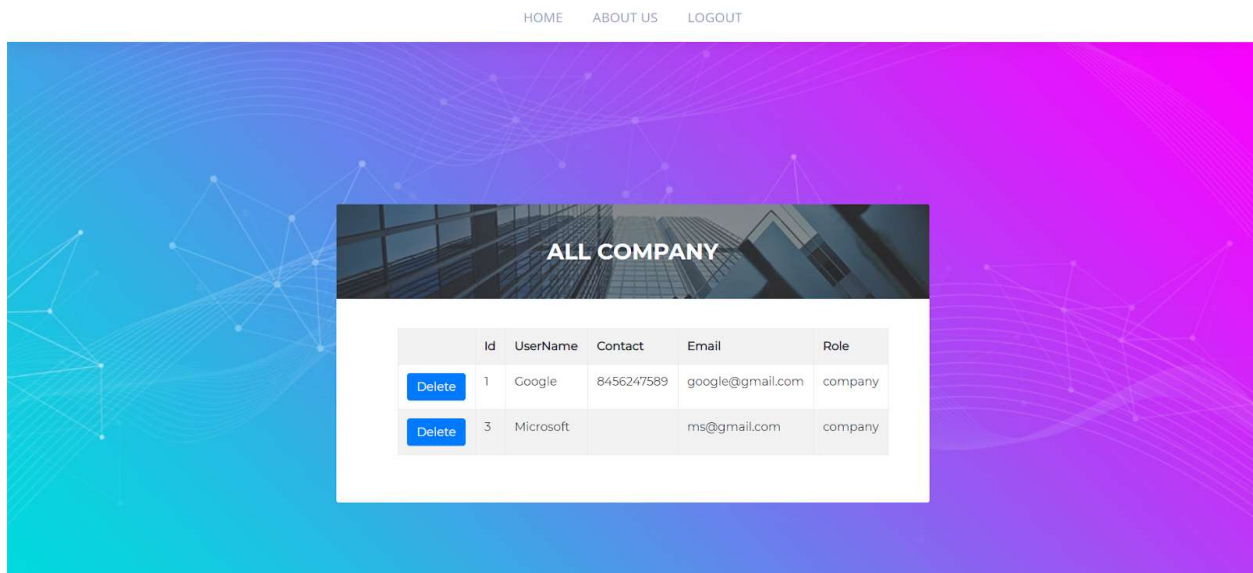
Company can view, update and delete details of posted job details.

9. Admin Dashboard



Admin can login with his/her password and view dashboard.

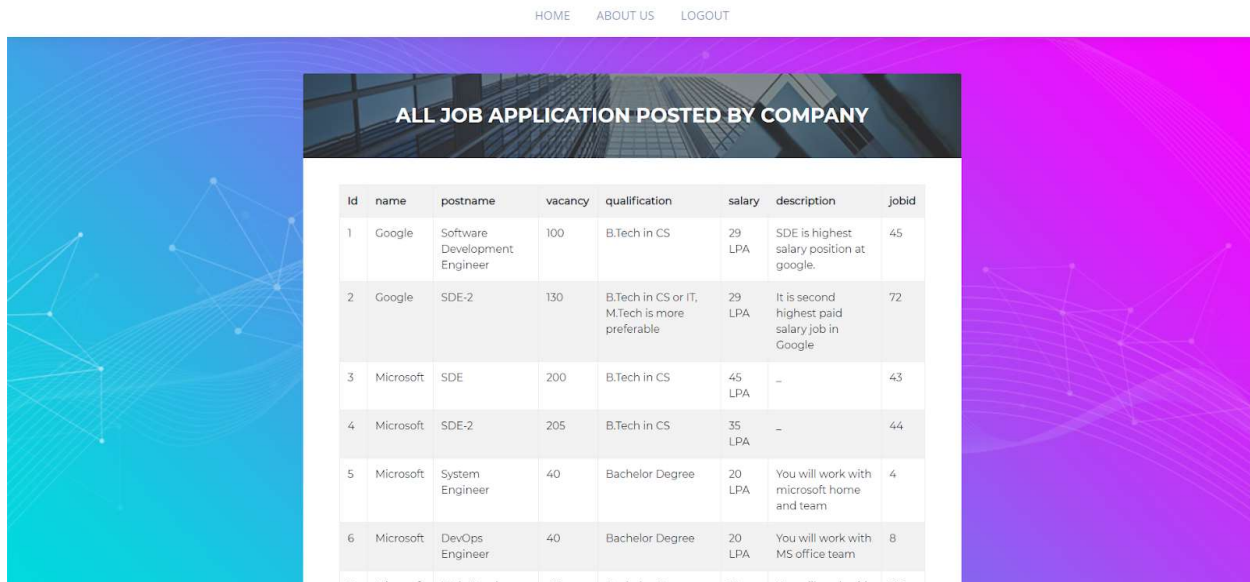
10. View All Company



Admin can view and delete all company details.

11. View All Job Application

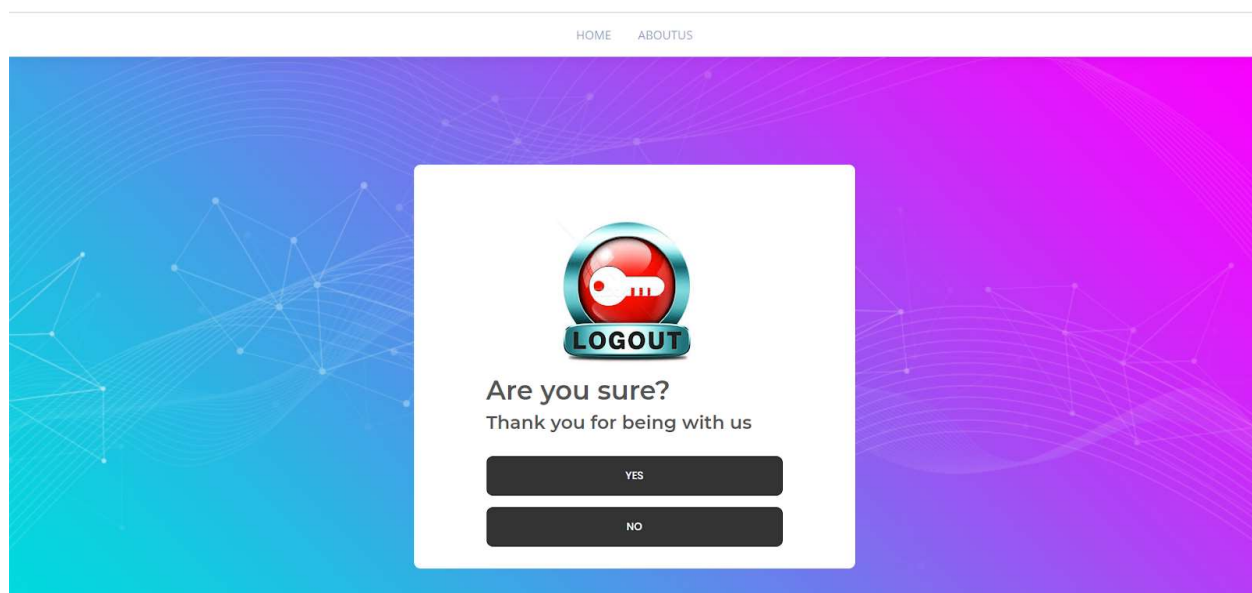
HOME ABOUT US LOGOUT



Id	name	postname	vacancy	qualification	salary	description	jobid
1	Google	Software Development Engineer	100	B.Tech in CS	29 LPA	SDE is highest salary position at google.	45
2	Google	SDE-2	130	B.Tech in CS or IT, M.Tech is more preferable	29 LPA	It is second highest paid salary job in Google	72
3	Microsoft	SDE	200	B.Tech in CS	45 LPA	-	43
4	Microsoft	SDE-2	205	B.Tech in CS	35 LPA	-	44
5	Microsoft	System Engineer	40	Bachelor Degree	20 LPA	You will work with microsoft home and team	4
6	Microsoft	DevOps Engineer	40	Bachelor Degree	20 LPA	You will work with MS office team	8
7	Microsoft	Web Developer	40	Bachelor Degree	18	You will work with	16

Admin can view all job posted by company.

12. Log out



After clicking on yes user will logged out from system.

8. Limitation and Future Extension

Limitations

1. Currently system can not find relevant jobs according to his/her profile.
2. There is no any authentication to check whether job seeker is eligible to apply for any job title or not.
3. There is no functionality to store resume on cloud storage.

Future Extension

1. We will provide recommendations and email updates for new job postings based on the job seeker's search history. We will implement the management of organizational documents in our system.
2. We can also provide templates for building the Resumes which might interest most applicants

9. Conclusion

The functionality of this system is developed after understanding whole system flow and all module and it is as per requirement.

The functionality which were successfully implemented :

1. Registration/Login
2. View Job
3. Apply for job
4. View applied job
5. View user who have applied for any position in company
6. Update profile
7. Withdraw application and posted job

10. Bibliography

Reference Links:

<https://stackoverflow.com/>

<https://forums.asp.net/>

<https://www.c-sharpcorner.com/>