

ONLINE JOB SEARCH PORTAL

A Project Report

Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

SIDDHI SANTOSH SHIRKE
527

Under the esteemed guidance of

Ms. Anisha Asirvatham
Assistant Professor



NAGINDAS KHANDWALA COLLEGE(Autonomous)

(Affiliated to University of Mumbai)

MUMBAI, 400 064

MAHARASHTRA

2021-22

PNR No.:

Roll no: 527

1. Name of the Student: Siddhi Santosh Shirke
2. Title of the Project: Online Job Search Portal
3. Name of the Guide: Ms. Anisha Asirvatham
4. Teaching/Industry experience of the Guide: 11 years
5. Is this your first submission? No

Signature of the Student:

Signature of the Guide:

Date:

Date:

Signature of the Coordinator:

Date:

NAGINDAS KHANDWALA COLLEGE(Autonomous)

(Affiliated to University of Mumbai)

MUMBAI, 400 064
MAHARASHTRA

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project titled, "**Online Job Search Portal**", is bonafide work of **Siddhi Santosh Shirke** bearing Seat No: **527** submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE** in **INFORMATION TECHNOLOGY** from the University of Mumbai.

Internal Guide

Coordinator

External Examiner

Date:

College Seal

ABSTRACT

“JOB PRAISE” will be an online job searching portal. Job Search Portal is a web application that allows Job Seekers to find available jobs and Employers to find eligible job seekers for their company. Online Job Search Portal act as a bridge of communication between organizations and applicants. Here employers will be able to post their jobs and review applications. Job Seekers can search for jobs in any field. Can upload their resumes.

The project aims to make job searching easy for job seekers and make the employer's work easy in finding the right candidate for the company. The system will ensure security, will be reliable, maintainable, and available 24/7 the only condition is that the user would require a good connection to the internet to access the website. Every user of the system has to log in to the system with a registered email id and password to ensure security and authentication.

The main purpose of my website will be to provide an easy and efficient way of searching for jobs and candidates. Also, one particular candidate can apply to multiple jobs. The job seeker could apply for any job that is best suited for him/her and will also get updates for various job openings dependent upon his/her qualification. The employer can also post multiple job openings at the same time and update or delete the post as per the requirement. The employer uses the system to find the right candidate at any time according to his/her convenience

ACKNOWLEDGEMENT

The “Online Job Portal” project has been a unique experience for me. I am very thankful to the Nagindas Khandwala College Department of Information Technology, who gave us the opportunity to accomplish this project. I would like to thank all those who give guided and associated me in the completion of this project.

I express my sincere gratitude to our Director madam Dr. Mrs. Ancy Jose, I/c. Principal Prof. Dr. Moushumi Datta, Coordinator Dr. Sindhu P.M, and Project Guide Ms. Anisha Asirvatham for her invaluable support and guidance during the course of the project. I am also very much thankful to the Authorities for including the project work as part of the syllabus, without which I would not have gained the experience of developing software like this.

Finally, I would like to express my gratitude to my friends for their support and guidance throughout this venture. Last but not least I would thank my family whose support, motivation, and encouragement without it this would not have been possible.

DECLARATION

I hereby declare that the project entitled, "**Online Job Search Portal**" done at the **place where the project done**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as a final semester project as part of our curriculum.

SIDDHI SANTOSH SHIRKE

TABLE OF CONTENTS

| | |
|---|----|
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 BACKGROUND: | 1 |
| 1.2 OBJECTIVES: | 1 |
| 1.3 APPLICABILITY:..... | 2 |
| CHAPTER 2: GAP ANALYSIS/ DRAWBACK OF EXISTING SYSTEM..... | 3 |
| CHAPTER 3: REQUIREMENT AND ANALYSIS | 4 |
| 3.1 Problem Definition..... | 4 |
| 3.2 Requirements Specification | 4 |
| 3.3 Planning and Scheduling: | 6 |
| 3.4 Software and Hardware Requirements: | 6 |
| 3.5 Preliminary Product Description: | 8 |
| Chapter 4: SYSTEM DESIGN | 10 |
| 4.1 Basic Modules..... | 10 |
| 4.2 Schema Design..... | 12 |
| 4.3 UML Diagrams | 16 |
| 4.4 User Interface Design..... | 23 |
| 4.5 Security Issues: | 28 |
| Chapter 5: IMPLEMENTATION AND TESTING..... | 29 |
| 5.1 Code (Place Core Segments)..... | 29 |
| 5.2 Testing Approach and Test Cases..... | 32 |
| Chapter 6: RESULTS AND DISCUSSION | 52 |
| 6.1 Test Reports | 52 |
| 6.2 User Documentation | 57 |
| Chapter 7: CONCLUSIONS | 61 |
| 7.1 Conclusion | 61 |
| 7.1.1 Significance of the System..... | 61 |
| 7.2 Limitation..... | 62 |
| 7.3 Future Scope of the Project | 62 |

| | |
|------------------|----|
| REFERENCES | 63 |
| APPENDIX..... | 64 |

List of Tables

| | |
|---|----|
| Table 1 MANUAL TESTING OF REGISTRATION FORM (UNIT TESTING)..... | 35 |
| Table 2 MANUAL TESTING OF LOGIN FORM (UNIT TESTING) | 36 |
| Table 3 SYSTEM TESTING | 51 |
| Table 4 UNIT TESTING REPORT | 52 |
| Table 5 SYSTEM TESTING REPORT..... | 56 |

List of Figures

| | |
|---|----|
| Figure 1 GANNT CHART..... | 6 |
| Figure 2 BASIC MODULE JOB SEEKER | 11 |
| Figure 3 BASIC MODULE EMPLOYER..... | 12 |
| Figure 4 USER DATABASE TABLE..... | 12 |
| Figure 5 JOB SEEKER DATABASE TABLE..... | 13 |
| Figure 6 EMPLOYER DATABASE TABLE | 14 |
| Figure 7 JOB POST DATABASE TABLE | 14 |
| Figure 8 APPLIED JOBS DATABASE TABLE | 15 |
| Figure 9 QUESTIONNAIRE DATABASE TABLE..... | 15 |
| Figure 10 ENTITY- RELATIONSHIP DIAGRAM..... | 16 |
| Figure 11 CLASS DIAGRAM..... | 17 |
| Figure 12 SEQUENCE DIAGRAM | 18 |
| Figure 13 USER ACTIVITY DIAGRAM | 19 |
| Figure 14 COMPANY ACTIVITY DIAGRAM | 19 |
| Figure 15 COLLABORATION DIAGRAM | 20 |
| Figure 16 USE CASE DIAGRAM | 21 |
| Figure 17 USER STATE TRANSITION DIAGRAM | 22 |
| Figure 18 EMPLOYER STATE TRANSITION DIAGRAM | 22 |
| Figure 19 HOME PAGE..... | 23 |
| Figure 20 REGISTRATION FORM..... | 23 |
| Figure 21 SERVICES PAGE OF EMPLOYER | 24 |
| Figure 22 SERVICES PAGE OF JOB SEEKER | 24 |
| Figure 23 JOB POST FORM | 25 |
| Figure 24 LOGIN FORM | 26 |
| Figure 25 JOBS SEARCHED PAGE | 26 |
| Figure 26 APPLIED CANDIDATES PAGES | 27 |
| Figure 27 DETAILS OF CANDIDATE PAGE..... | 27 |
| Figure 28 DASHBOARD | 28 |
| Figure 29 SIGN UP CODE..... | 29 |
| Figure 30 SIGN IN CODE..... | 30 |

| | |
|---|----|
| Figure 31 HOME PAGE CODE | 30 |
| Figure 32 JOB POST FORM CODE | 31 |
| Figure 33 RESUME BUILDING AND JOBS APPLIED CODE | 31 |
| Figure 34 JOBS APPLIED BY CANDIDATES | 32 |
| Figure 35 TEST CASE OF JOB POST FORM | 36 |
| Figure 36 TEST CASE OF JOB POST FORM | 36 |
| Figure 37 TEST CASES OF JOB POST FORM AND USER FORM | 37 |
| Figure 38 TEST CASES OF USER FORM | 38 |
| Figure 39 TEST CASE OF USER FORM..... | 39 |
| Figure 40 TEST CASE OF EMPLOYER DETAILS FORM..... | 40 |
| Figure 41 JOB SEEKER's DETAILS FORM..... | 41 |
| Figure 42 JOB SEEKER's DETAILS FORM..... | 42 |
| Figure 43 JOB SEEKER's DETAILS FORM..... | 42 |
| Figure 44 RESULTS OF TEST CASES..... | 43 |
| Figure 45 URL TESTING TEST CASES | 48 |
| Figure 46 RESULTS OF URL TESTING | 48 |
| Figure 47 UNIT TESTING REPORT- PIE CHART | 53 |
| Figure 48 SYSTEM TESTING REPORT - PIE CHART | 57 |
| Figure 49 SWOT ANALYSIS | 68 |

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND:

Job Search Portal is a web application, which allows Job Seekers to find available jobs and Employers to find eligible job seekers for their company. Online Job Search Portal act as a bridge of communication between organizations and applicants. Here employers will be able to post their jobs and review applications. Job Seekers can search for jobs in any field. Can upload their resumes.

“**JOB PRAISE**” will be an online job searching portal. It will have basic features of the job portal like job seekers will be able to find jobs which will be posted by the registered companies. These companies will also be verified to some extent. It will be a platform where job seekers and employers connect. It aims to be a platform that can be used from fresher to an experienced person. Job Seekers can upload their resumes or make one if they don’t have one and apply to the jobs posted by the employers. Also, at a time they can apply to multiple companies. Employers have to register their company on the portal and then they have to post a job for candidates to apply for. They can then select the candidates based on their skills of the job seeker they can shortlist or select candidates for offline interviews or any other process. An employer can post multiple job posts.

1.2 OBJECTIVES:

1. To build a base for the job portal
2. To build a system where users can search for jobs and apply for them.
3. To build a system where the employer can register their company and post several job vacancies.
4. To build a system that helps employers to select the best candidates through less effort.i.e. by doing Skill Testing
5. To build a system that allows job seekers to filter job postings according to their requirements. Location, Salary Range, Job Type, Education Level, etc.

6. To build a system that allows the recruiters to shortlist candidates and also send a notification to the candidates through the mail.
7. To provide a resume based on the user's profile.

1.3APPLICABILITY:

1. Fresher or Job Seeker can Search for Jobs Online.
2. Employers can post their vacancies and get an eligible candidate for their company.
3. Job Seekers can get to know on-demand skills and industrial requirements.
4. Job Seekers are exposed to the large corporate world.

CHAPTER 2: GAP ANALYSIS/ DRAWBACK OF EXISTING SYSTEM

Through gap analysis, we get to know what is wrong with the existing systems, and then we can further think of how we can rectify those mistakes in our system. The gap analysis also helps us to set the objectives of the system. We get a clear vision of what and how we are supposed to build our new system. Through gap analysis, we can make a better version of the system from the existing system. Gap analysis can be done by either reading research papers or it can also be done by actually interacting with people and questioning them about the problems faced by them while using the existing system or what else do they feel should be included in the new system.

I have done my gap analysis by reading research papers as well by analyzing the problem faced by people.

Gap Analysis:

1. A Large number of applications for a job post makes it a bit difficult for the recruiter to choose the most eligible candidate.
 - When a job post is posted a large number of applicants start applying to the post which makes the employer's work a bit tedious. So to overcome this problem we can set a limit on the number of applicants that can apply to the post the limit will be determined by the employer
2. Unqualified applicants applying for the post.
 - When a job post is posted any user i.e. candidate can apply to the post which is not right, the employer should get applications of the right candidate i.e. a candidate that is suitable for the job role. We can overcome this problem by making the candidates answer certain questionnaires and based on their answers the employers can decide whether to select or reject the candidate.
3. Fake Job Postings or Duplication of Jobs
 - Proper verification should be conducted on the company's existence
4. No proper feedback through the portal after the recruiter views a profile.
 - After applying to a job post no proper feedback from the employer whether a candidate is selected or not.

CHAPTER 3: REQUIREMENT AND ANALYSIS

3.1 Problem Definition

Online Job Search Portal is a commonly available app in today's times. The development of the "JOB PRAISE" job portal is basically for self-assessment and self-development. However, through some research, I have figured out certain gap analysis in the existing portals.

The basic functionality of the job portal is to make allow the job seeker to search and get information about the job and then based on its skills allow the job seeker to apply for the job. One job seeker can apply to as many job posts as possible. The Job Portal should allow the employer to post job details and select the candidates and call for an interview.

The Existing systems do not check for the qualification of the candidate before allowing him/her to apply for the job post which eventually makes it difficult for the employer to shortlist the candidates. The second problem that employers encounter is a large number of candidates applying to the job post. So, imagine if one employer is posting 3 job posts then how difficult would it be for the employer to shortlist the candidate.

Also, no proper feedback from the employer if the candidate is been shortlisted or rejected after applying for the job post.

3.2 Requirements Specification

3.2.1 Purpose:

The purpose of this document is to build an "Online Job Search Portal" as a bridge of communication between organizations and applicants. Build a portal that allows Job Seekers to find available jobs and Employers to find eligible job seekers for their company.

3.2.2 Intended Audience:

The targeted audience for Online Job Search Portal would be people who are searching for jobs, who are willing to change a job, also it would be a good platform for fresher who is willing to search for a job.

3.2.3 Project Scope:

The purpose of this project is to ease the job search for fresher, and for those who need jobs also for employers who are searching for candidates and have vacancies in their company. The project aims at creating an easy-to-use application for all users who are trying to search for jobs and to find the right candidates.

3.2.4 Product Perspective:

The system first asks the user whether he/she is a job seeker or employer. Then if the user is a job seeker they are allowed to search for jobs and they want to apply they have to create their account and log in to the system. If the user is an employer then the user has to first register their company and login into the system. Then the employer can create a job post and further shortlist the candidates them and select for an interview. The employer can further delete the job post if required.

The Job Praise database system will store the following details:

Job Seeker details:

It includes user name, contact number, email id qualification, CV. This information is used for verification, for contacting the candidates in case of the selection, and for providing qualification details to the employer.

Company Details:

It includes company name, contact number, official website, email id. This information is used for verification and for providing contact details to candidates when they are selected.

Job Application Details:

It includes the candidate id applied to which job id and the date of application.

3.3 Planning and Scheduling:

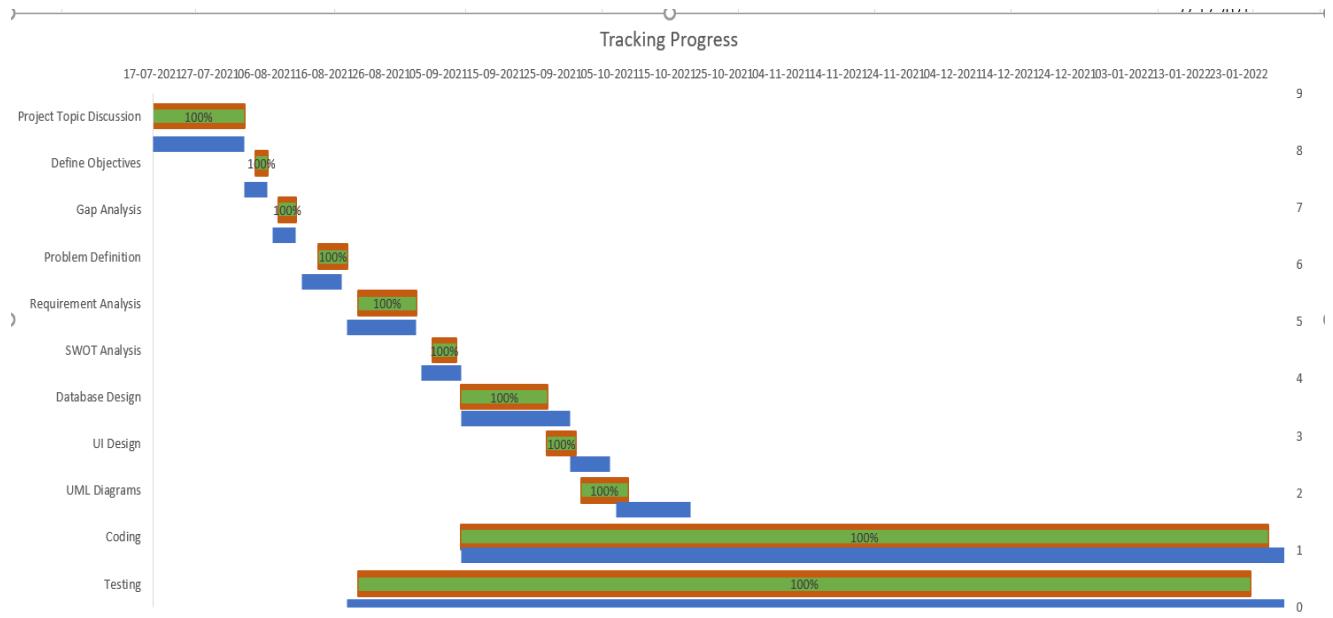


Figure 1 GANNT CHART

3.4 Software and Hardware Requirements:

3.4.1 Software Requirement:

- Programming Languages: Front-end - HTML, CSS, Bootstrap, Back-end – Django
- Code Platform: IDE: PyCharm
- Database: MySQL

3.4.2 Hardware Requirements:

- Keyboard
- Any mouse
- Windows

3.4.3 Justification of selection of technology:

HTML:

It stands for HyperText Markup Language. It is used for documents to be displayed in a web browser. It can be further assisted by CSS and javascript. Web browsers receive HTML documents from a web server or local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS:

CSS stands for cascading style sheet. It is used to design the HTML document that is displayed on the web browser. Using CSS we can make our web page attractive and user-friendly which eventually makes the user feel to stay on our site for a little longer. Through we can add colors, style, change font size, style, etc. Also through CSS, we can make our page responsive that is based on the different screen sizes we can adapt our web page.

Django:

Django is a high-level Python web framework that enables the rapid development of secure and maintainable websites. Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. Django is used for the development of different types of websites, in particular, a highly customizable app, such as a social media website. Django is a free, open-source, high-end framework written in the famously simple, flexible, and relatively easy-to-learn Python programming language.

MySQL:

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services.

3.5 Preliminary Product Description:

Online Job Search Portal act as a bridge of communication between organizations and applicants. Here employers will be able to post their jobs and review applications. Job Seekers can search for jobs in any field. Can upload their resumes. “**JOB PRAISE**” will be an online job searching portal. It will have basic features of the job portal like job seekers will be able to find jobs which will be posted by the registered companies. These companies will also be verified to some extent.

3.5.1 Modules

1. Registration:

The job seeker has to register if wants to apply for any of the job posts. The system will store information such as name, contact number, email id, qualification, skills, CV.

The employer must register its company with the system. The system will store information such as name, contact number, official email id, official website.

2. Login:

After registration, the user has to log in to the system to apply for the jobs. Without logging in to the system the job seeker can just search for a job post but won't be able to apply for it. The employer has to also log in to the system to get access to various functions.

3. Search for Job Post:

This feature is to be used by the job seeker. Here a Jobseeker and search for available jobs and then they can apply for it. Also, the job seeker can search for a job by applying various filters such as job location, job title, job salary, job skills, and qualifications required.

4. Post Job Vacancies:

This feature is to be used by the employer. Here the employer has to give a job title, job description, skills and qualifications required for the post, salary, job type, job location, etc. According to which the candidate will apply to the post.

5. Resume Building:

Here candidates can get their resumes if they don't have one. This feature is basically for fresher.

3.5.2 System Functionalities:

User Login:

1. The system will check for credentials if the user exists the login will be granted.
2. The system will allow the user to apply for job posts if and only if the login is done.
3. The system will allow the job seeker to search even if the user is not logged in.
4. The system will allow the candidate to build a resume only after login.

Employer Login:

1. The system will allow the employer to post job vacancies only after login.
2. The system will provide the candidate's details to the employer when the job seeker applies for the job post.
3. The system will allow employers to shortlist the candidates.
4. The system will allow employers to search for a candidate based on their qualifications.

Chapter 4: SYSTEM DESIGN

4.1 Basic Modules

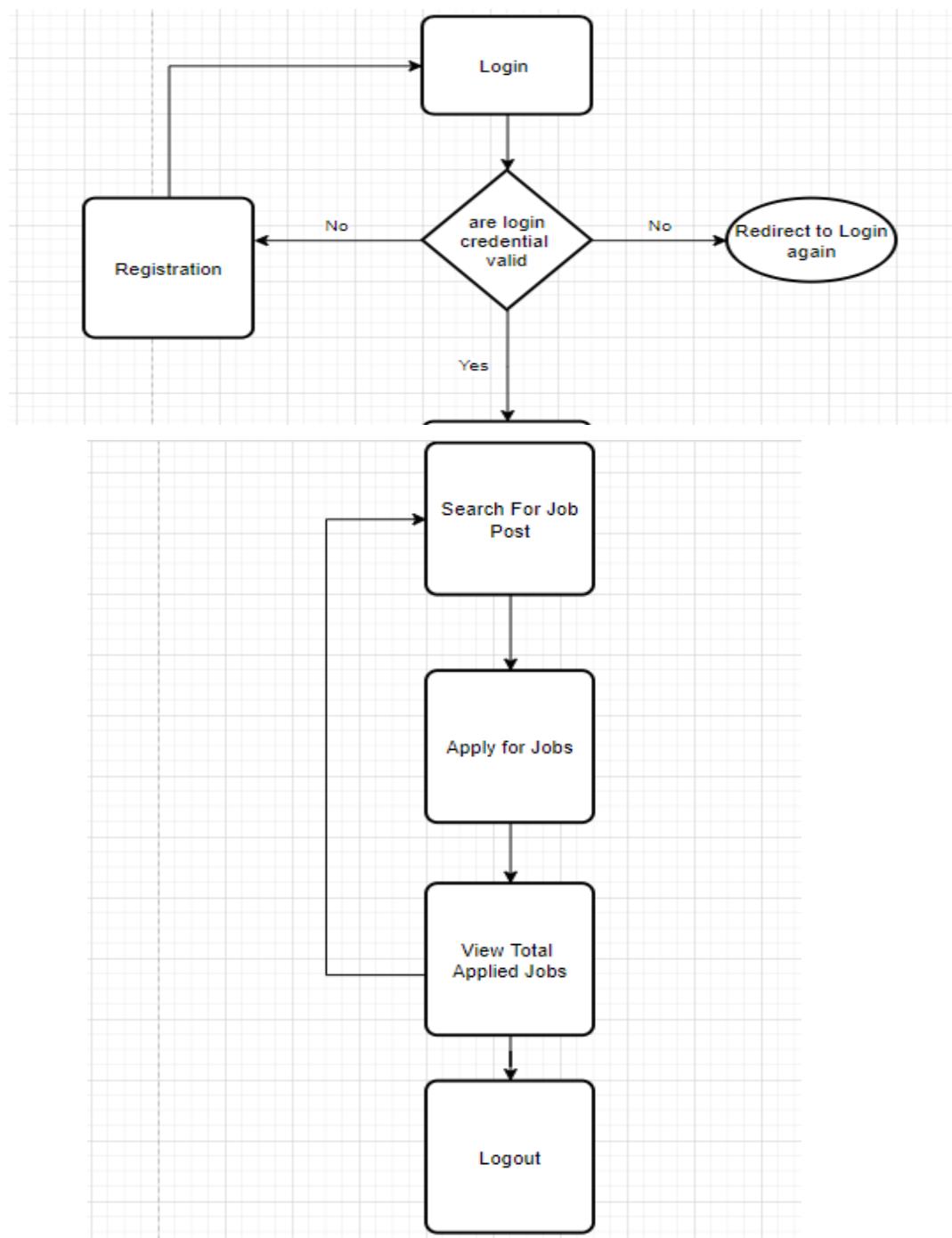
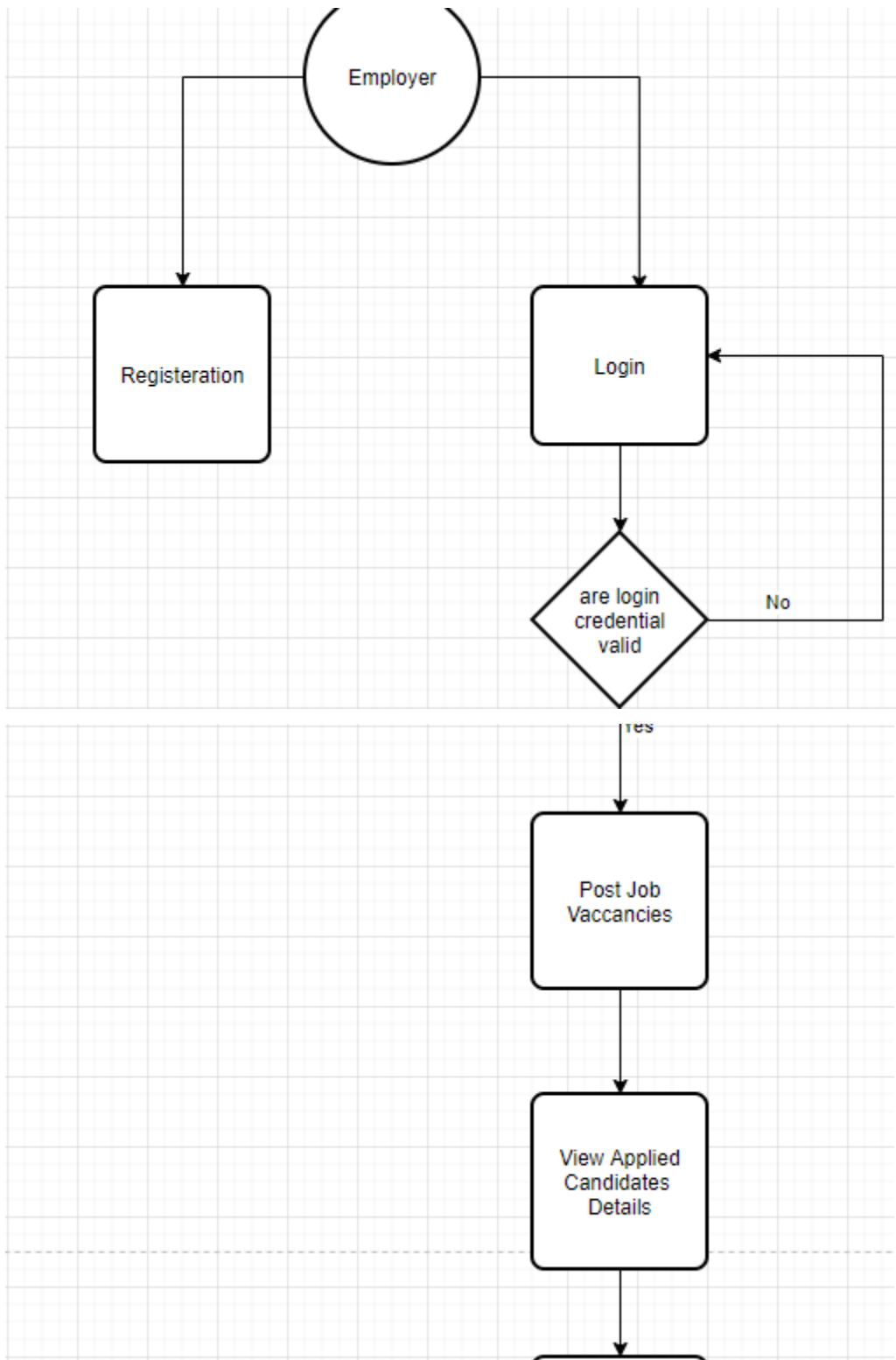


Figure 2 BASIC MODULE JOB SEEKER

Employer



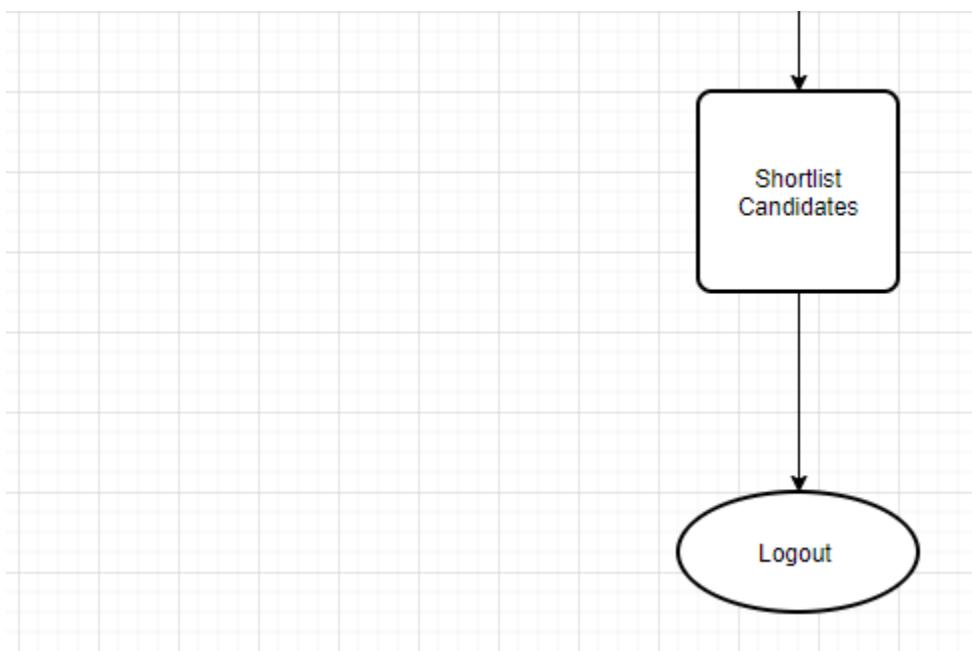


Figure 3 BASIC MODULE EMPLOYER

4.2 Schema Design

USER TABLE

| 22 DESCRIBE Proxy.create_proxy_user | | | | | | |
|-------------------------------------|--------------|---------------|------|-----|---------|----------------|
| | Field | Type | Null | Key | Default | Extra |
| ▶ | id | bigint | NO | PRI | NULL | auto_increment |
| | password | varchar(128) | NO | | NULL | |
| | last_login | datetime(6) | YES | | NULL | |
| | is_superuser | tinyint(1) | NO | | NULL | |
| | username | varchar(250) | NO | | NULL | |
| | first_name | varchar(150) | NO | | NULL | |
| | last_name | varchar(150) | NO | | NULL | |
| | email | varchar(1000) | NO | | NULL | |
| | is_staff | tinyint(1) | NO | | NULL | |
| | is_active | tinyint(1) | NO | | NULL | |
| | date_joined | datetime(6) | NO | | NULL | |
| | type | varchar(50) | NO | | NULL | |
| | name | varchar(255) | NO | | NULL | |

Figure 4 USER DATABASE TABLE

Candidate Details Table:

23 DESCRIBE Proxy.create_proxy_candidatemore

The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. A query has been run to describe the 'Proxy.create_proxy_candidatemore' table. The results are displayed in a grid with the following columns: Field, Type, Null, Key, Default, and Extra. The table structure is as follows:

| Field | Type | Null | Key | Default | Extra |
|----------------|--------------|------|-----|---------|----------------|
| id | bigint | NO | PRI | NULL | auto_increment |
| birth_date | date | NO | | NULL | |
| gender | varchar(30) | NO | | NULL | |
| marital_status | varchar(30) | NO | | NULL | |
| qualification | varchar(76) | NO | | NULL | |
| contact_number | varchar(10) | NO | | NULL | |
| upload_cv | varchar(100) | NO | | NULL | |
| user_id | bigint | NO | UNI | NULL | |
| hobbies | varchar(100) | NO | | NULL | |
| skills | varchar(100) | NO | | NULL | |
| objective | varchar(500) | NO | | NULL | |
| full_name | varchar(100) | NO | | NULL | |
| upload_photo | varchar(100) | NO | | NULL | |

Figure 5 JOB SEEKER DATABASE TABLE

Employer Details Table:

24 **DESCRIBE Proxy.create_proxy_employermore**

The screenshot shows the 'Result Grid' of a DESCRIBE command for a table named 'Proxy.create_proxy_employermore'. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data includes fields like id (bigint, auto_increment), name_of_company (varchar(30)), website (varchar(500)), contact_number (varchar(10)), company_email_id (varchar(254)), user_id (bigint), company_address (varchar(200)), and upload_photo (varchar(100)).

| Field | Type | Null | Key | Default | Extra |
|------------------|--------------|------|-----|---------|----------------|
| id | bigint | NO | PRI | NULL | auto_increment |
| name_of_company | varchar(30) | NO | | NULL | |
| website | varchar(500) | NO | | NULL | |
| contact_number | varchar(10) | NO | | NULL | |
| company_email_id | varchar(254) | NO | | NULL | |
| user_id | bigint | NO | UNI | NULL | |
| company_address | varchar(200) | NO | | NULL | |
| upload_photo | varchar(100) | NO | | NULL | |

Figure 6 EMPLOYER DATABASE TABLE

Job Post Table

25 **DESCRIBE Proxy.create_proxy_jobpost**

The screenshot shows the 'Result Grid' of a DESCRIBE command for a table named 'Proxy.create_proxy_jobpost'. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data includes fields like id (bigint, auto_increment), job_title (varchar(100)), location (varchar(100)), skills_required (varchar(255)), salary (int unsigned), last_date_of_applying (date), user_id (bigint), job_description (varchar(200)), monthly_annually (varchar(20)), job_type (varchar(50)), and total_working_hours (varchar(200)).

| Field | Type | Null | Key | Default | Extra |
|-----------------------|--------------|------|-----|---------|----------------|
| id | bigint | NO | PRI | NULL | auto_increment |
| job_title | varchar(100) | NO | | NULL | |
| location | varchar(100) | NO | | NULL | |
| skills_required | varchar(255) | NO | | NULL | |
| salary | int unsigned | NO | | NULL | |
| last_date_of_applying | date | NO | | NULL | |
| user_id | bigint | NO | MUL | NULL | |
| job_description | varchar(200) | NO | | NULL | |
| monthly_annually | varchar(20) | NO | | NULL | |
| job_type | varchar(50) | NO | | NULL | |
| total_working_hours | varchar(200) | NO | | NULL | |

Figure 7 JOB POST DATABASE TABLE

Applied Jobs Table:

26 DESCRIBE Proxy.create_proxy_applied_jobs

The screenshot shows the MySQL Workbench interface with the SQL tab active. The query 'DESCRIBE Proxy.create_proxy_applied_jobs' has been run, and the results are displayed in a grid. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data for the applied_jobs table is as follows:

| Field | Type | Null | Key | Default | Extra |
|--------------|--------------|------|-----|---------|----------------|
| id | bigint | NO | PRI | NULL | auto_increment |
| candidate_id | varchar(100) | NO | MUL | NULL | |
| post_id | bigint | NO | MUL | NULL | |
| employer_id | bigint | NO | MUL | NULL | |
| status | varchar(50) | YES | | | in-touch |
| applied_date | date | NO | | NULL | |

Figure 8 APPLIED JOBS DATABASE TABLE

Questionnaire Table:

27 DESCRIBE Proxy.create_proxy_questions

The screenshot shows the MySQL Workbench interface with the SQL tab active. The query 'DESCRIBE Proxy.create_proxy_questions' has been run, and the results are displayed in a grid. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data for the questions table is as follows:

| Field | Type | Null | Key | Default | Extra |
|------------|----------|------|-----|---------|----------------|
| id | bigint | NO | PRI | NULL | auto_increment |
| q1 | longtext | NO | | NULL | |
| q2 | longtext | NO | | NULL | |
| q3 | longtext | NO | | NULL | |
| user_id | bigint | NO | MUL | NULL | |
| jobpost_id | bigint | NO | MUL | NULL | |

Figure 9 QUESTIONNAIRE DATABASE TABLE

4.3 UML Diagrams

Entity Relationship Diagram

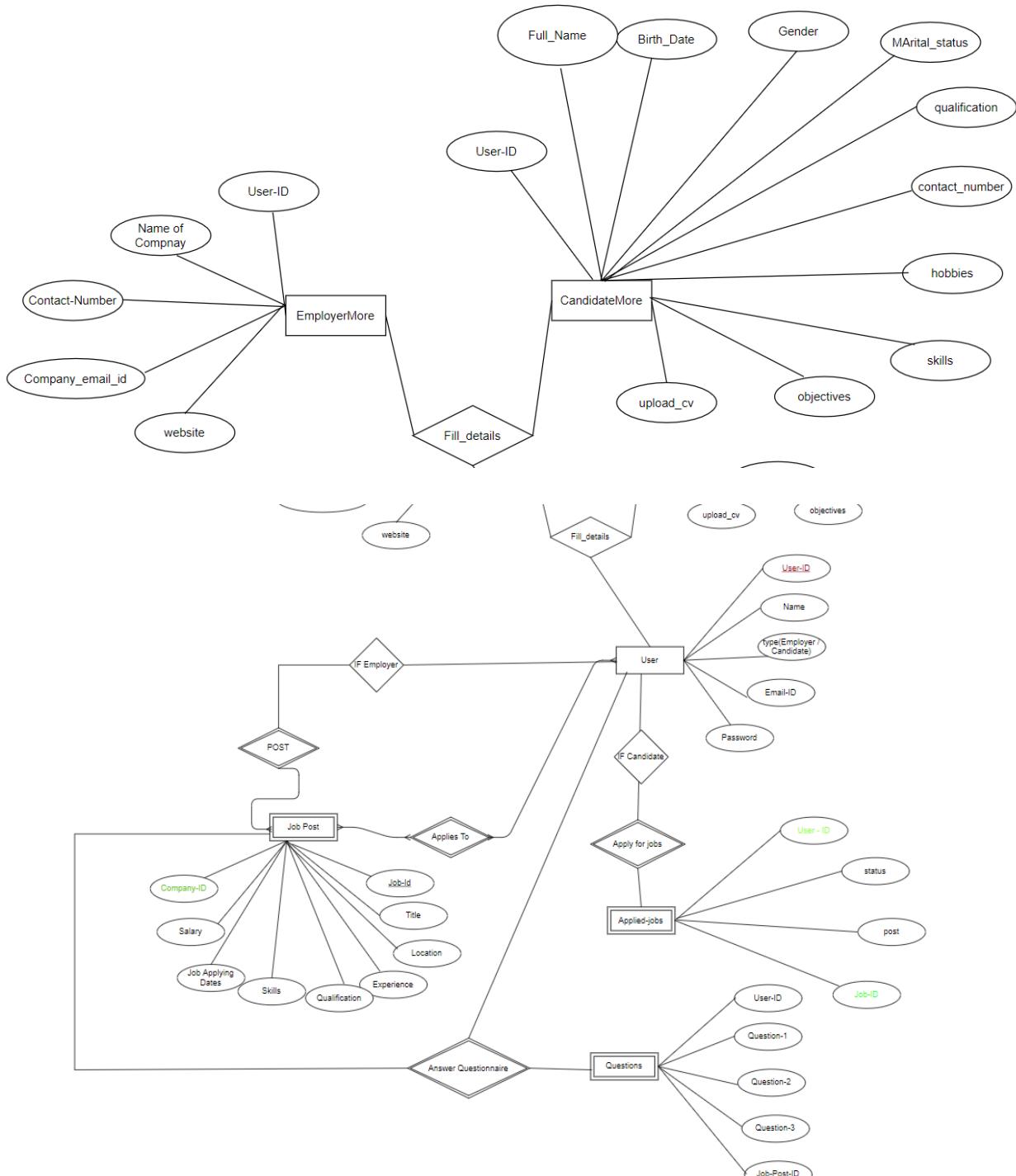


Figure 10 ENTITY- RELATIONSHIP DIAGRAM

Class Diagram:

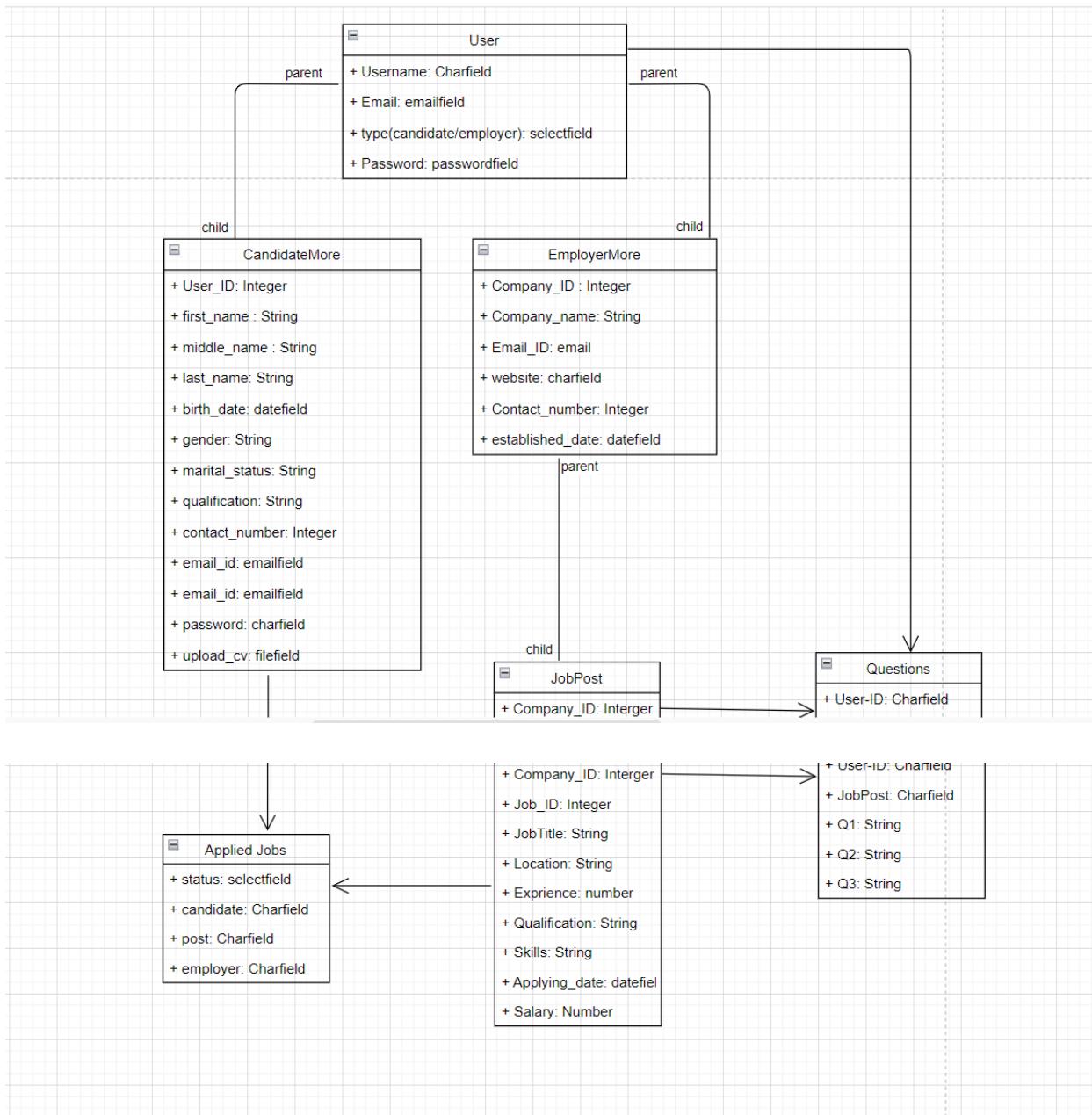


Figure 11 CLASS DIAGRAM

Sequence Diagram:

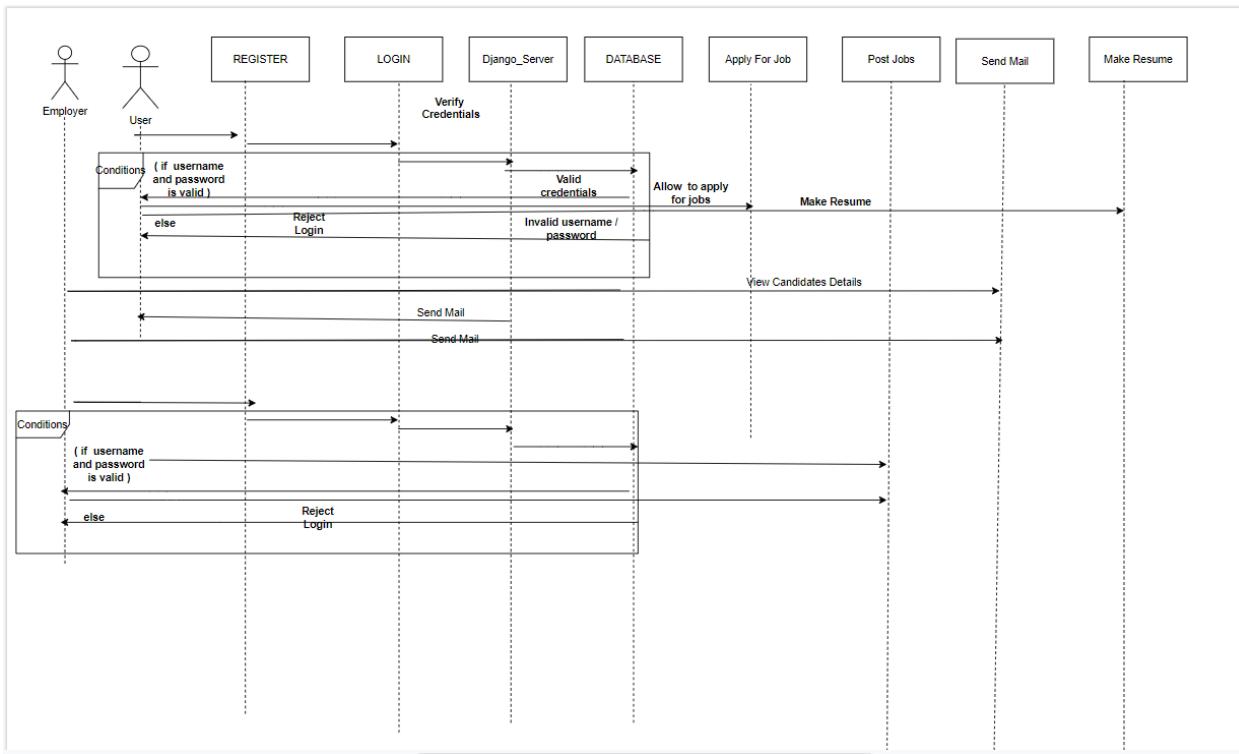


Figure 12 SEQUENCE DIAGRAM

Activity Diagram:

User Activity Diagram

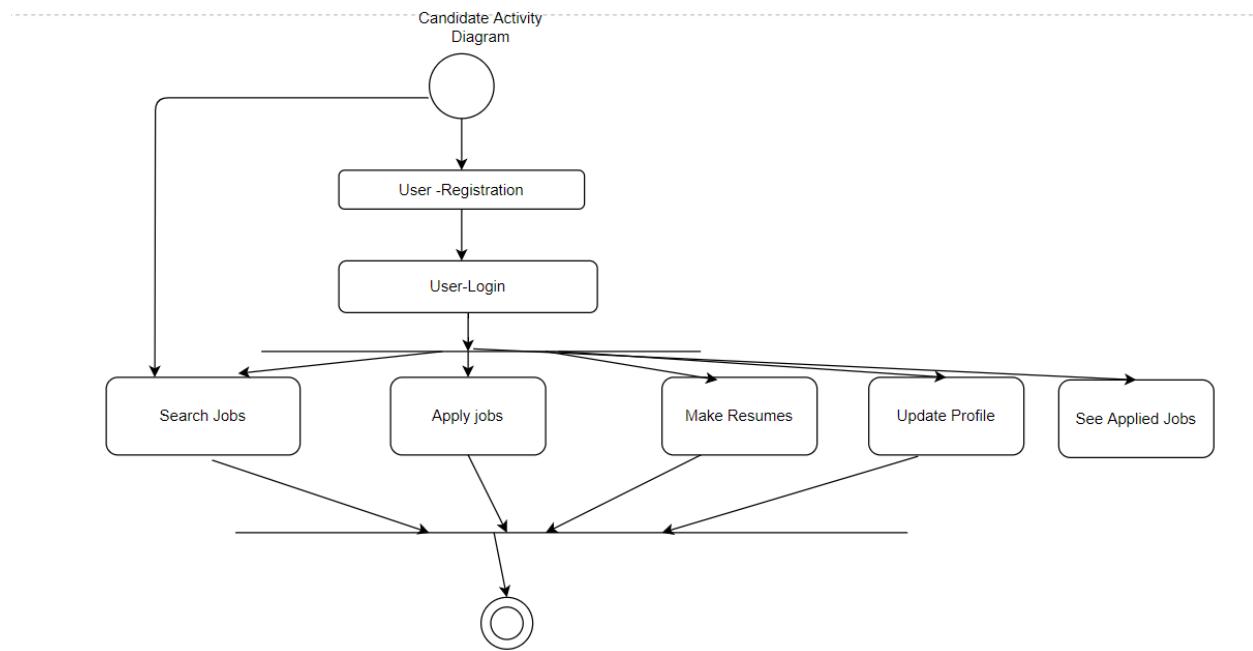


Figure 13 USER ACTIVITY DIAGRAM

Company Activity Diagram

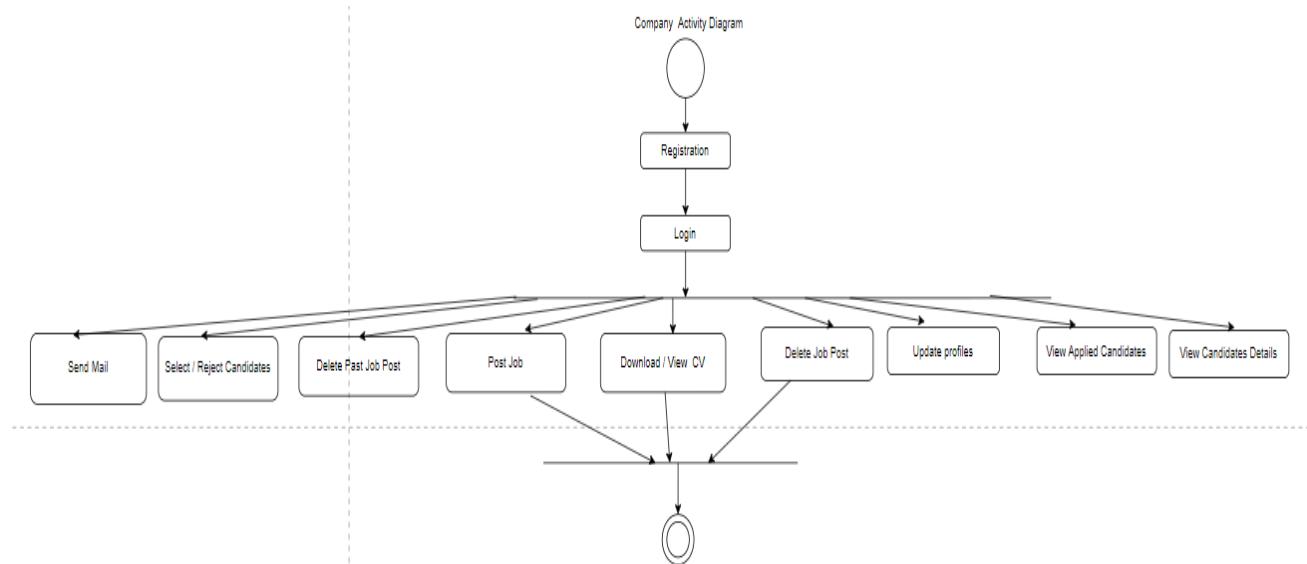


Figure 14 COMPANY ACTIVITY DIAGRAM

Collaboration Diagram

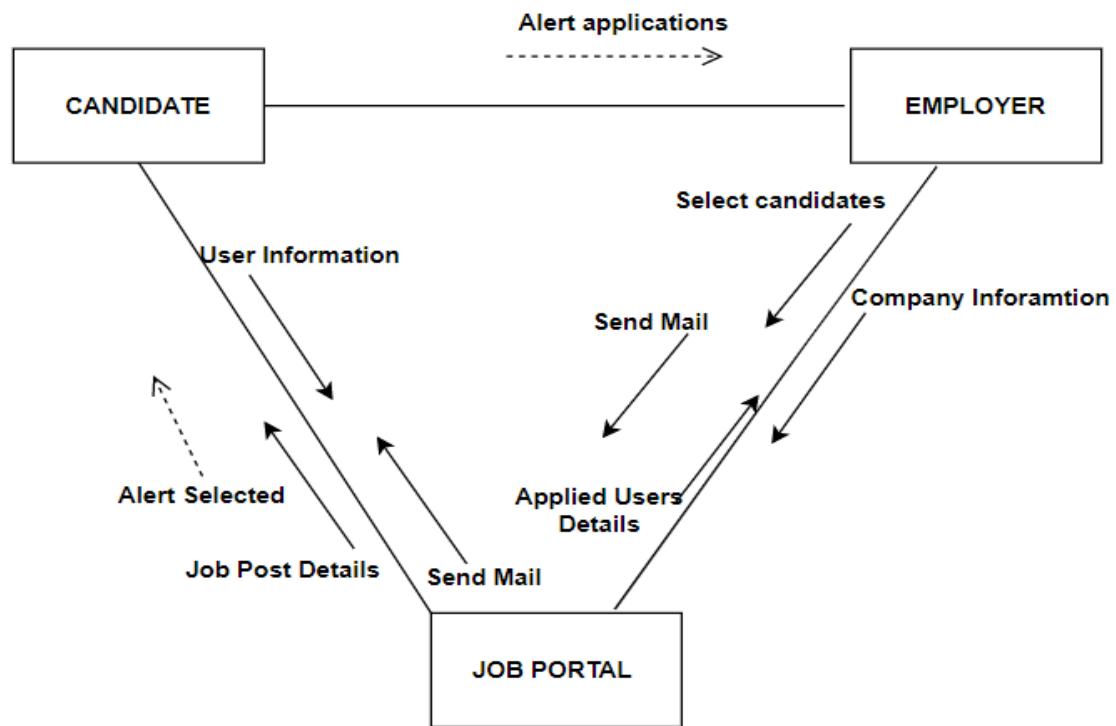


Figure 15 COLLABORATION DIAGRAM

Use Case Diagram

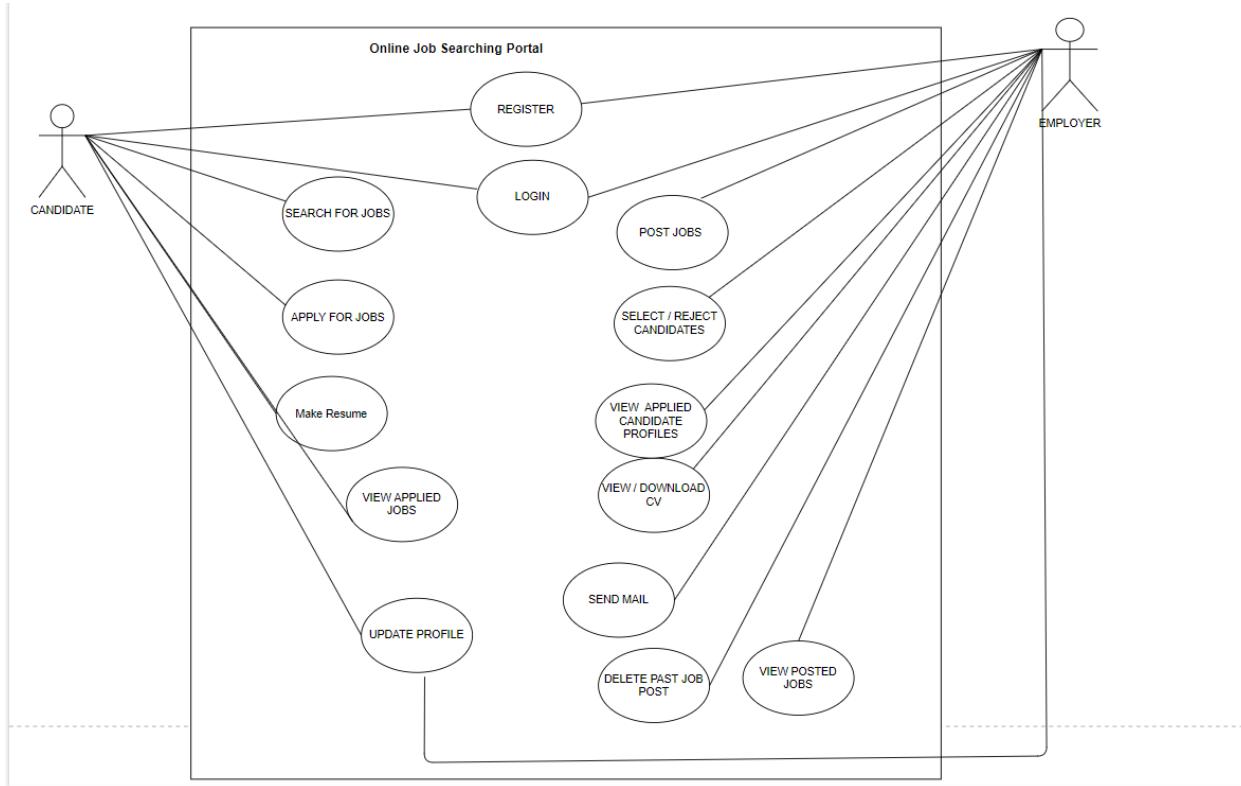


Figure 16 USE CASE DIAGRAM

State Transition Diagram

Job Seeker:

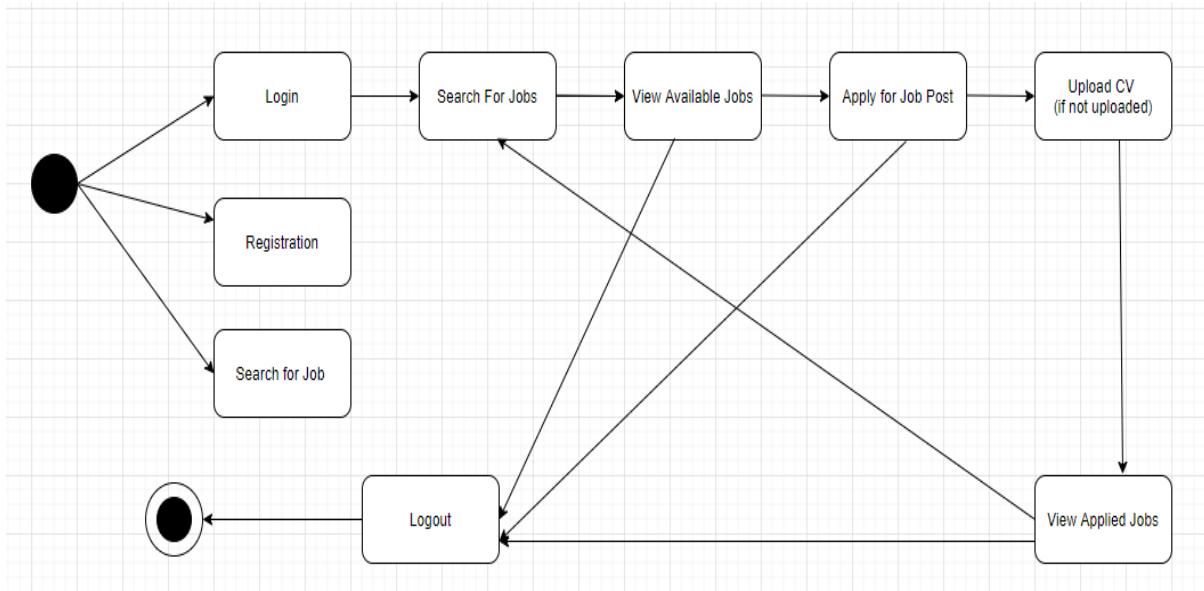


Figure 17 USER STATE TRANSITION DIAGRAM

Employer:

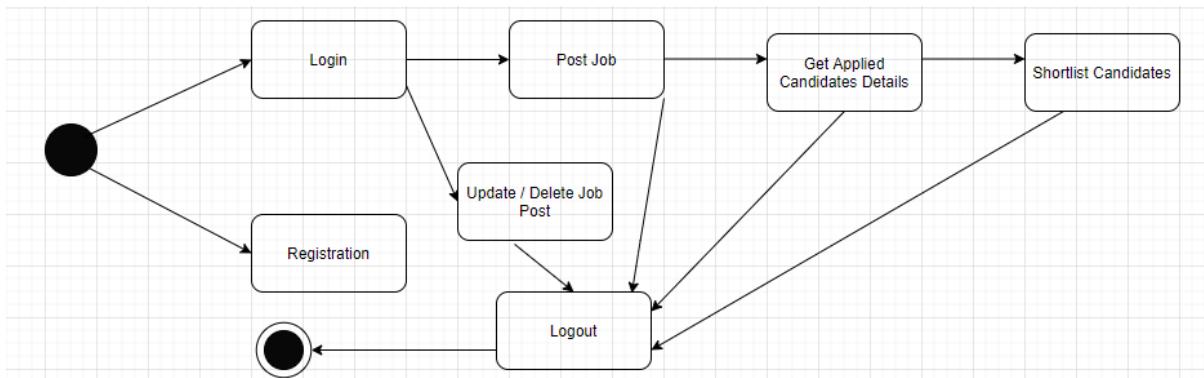


Figure 18 EMPLOYER STATE TRANSITION DIAGRAM

4.4 User Interface Design

Home Page

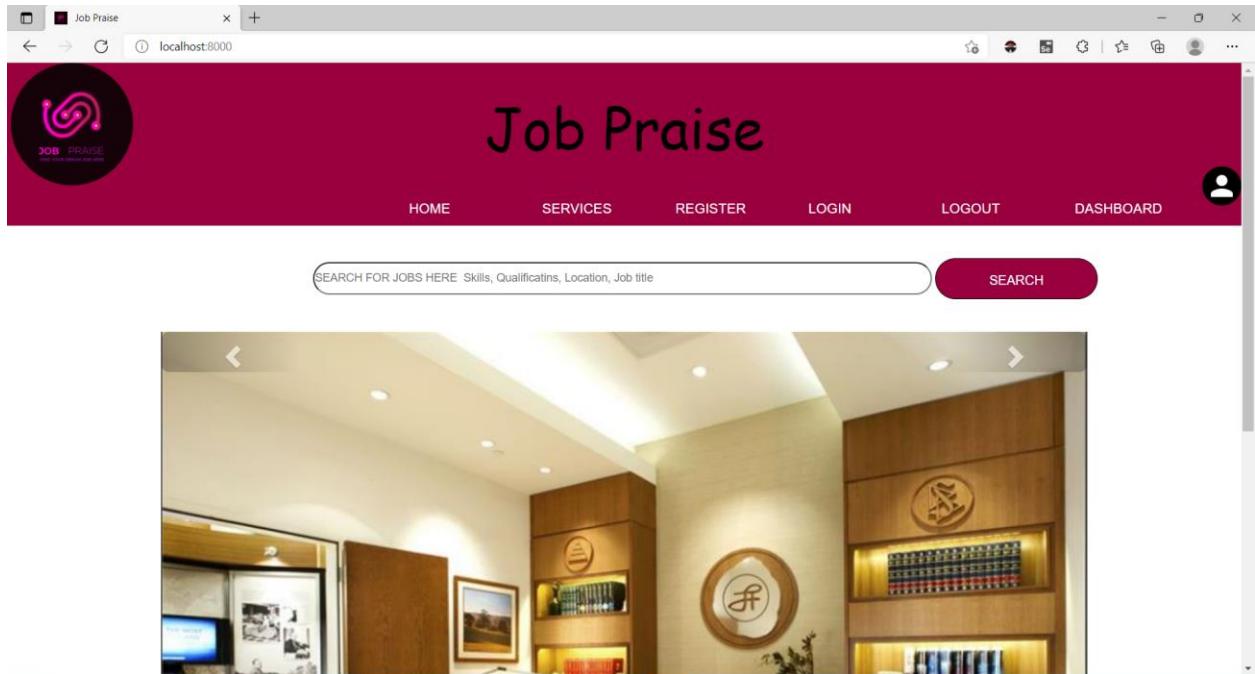


Figure 19 HOME PAGE

Registration Form:

A screenshot of a web browser displaying the 'Job Praise' website's registration form. The header is identical to the home page, with the 'Job Praise' logo and navigation links. The main content area is a white form with the following fields:

- Username:**
- Enter Your Email Id:**
- Type:**
- Password:**
- Confirm password:**

A purple 'Create Account' button is located at the bottom of the form.

Figure 20 REGISTRATION FORM

Service (When Logged-in as Employer)



Figure 21 SERVICES PAGE OF EMPLOYER

Service (When Logged-in Candidate)

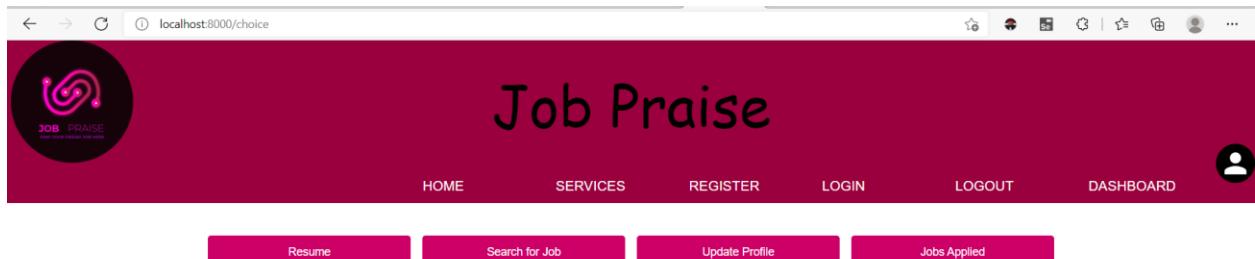


Figure 22 SERVICES PAGE OF JOB SEEKER

Job Post Form:

The screenshot shows a web browser window for 'Job Praise' at 'localhost:8000/jobpost'. The header features a logo with a stylized ear icon and the text 'JOB PRAISE'. The navigation bar includes links for HOME, SERVICES, REGISTER, LOGIN, LOGOUT, and DASHBOARD. A user profile icon is also present. The main content area is titled 'Job Post Form' and contains several input fields:

- Job title:**
- Job description:**
- Location:**
- Skills required:**
- Salary:**
- Monthly or Annual:**
- Last date of applying:**

A large red 'POST' button is located at the bottom of the form.

Figure 23 JOB POST FORM

Login Form:

The screenshot shows a web browser window for 'Job Praise' at the URL 'localhost:8000/signin'. The page has a dark red header with the 'Job Praise' logo on the left and navigation links for HOME, SERVICES, REGISTER, LOGIN, LOGOUT, and DASHBOARD on the right. Below the header is a white login form area. It contains fields for 'Enter Your Email Id:' and 'Password:', both with placeholder text (''). A 'Login' button is located below the password field.

Figure 24 LOGIN FORM

Search Items Lists:

The screenshot shows a web browser window for 'Job Praise' at the URL 'localhost:8000'. The layout is identical to Figure 24, with the 'Job Praise' logo, navigation links, and a white search results area. The search results area displays two job listings, each with a summary of requirements and an 'Apply' button.

| Job Title | Location | Skills Required | Salary | Last Date | Action |
|-----------------|----------|---------------------------------------|------------|----------------|--------|
| General Manager | Mumbai | Computer Knowledge, Accounting, Excel | 35000 p.m. | March 17, 2022 | Apply |
| General Manager | Mumbai | Computer Knowledge, Accounting, Excel | 10000 p.m. | April 22, 2022 | Apply |

Figure 25 JOBS SEARCHED PAGE

Applied Candidates Pages:

The screenshot shows a web browser window titled "Job Praise" with the URL "localhost:8000/showdel". The page has a dark red header with the "Job Praise" logo on the left and navigation links for HOME, SERVICES, REGISTER, LOGIN, LOGOUT, and DASHBOARD on the right. Below the header is a table listing candidate details:

| Candidate Email | See Details | Job Title | Send Email |
|------------------|-----------------|--------------|------------|
| shirke@gmail.com | Web DDeveloper | Show Details | Send Mail |
| shirke@gmail.com | General Manager | Show Details | Send Mail |
| shirke@gmail.com | DATA ANALYST | Show Details | Send Mail |
| monica@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | DATA ANALYST | Show Details | Send Mail |
| malise@gmail.com | DATA ANALYST | Show Details | Send Mail |

This Is Candidate Details page

| Candidate Email | See Details | Job Title | Send Email |
|------------------|-----------------|--------------|------------|
| shirke@gmail.com | Web DDeveloper | Show Details | Send Mail |
| shirke@gmail.com | General Manager | Show Details | Send Mail |
| shirke@gmail.com | DATA ANALYST | Show Details | Send Mail |
| monica@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | General Manager | Show Details | Send Mail |
| malise@gmail.com | DATA ANALYST | Show Details | Send Mail |
| malise@gmail.com | DATA ANALYST | Show Details | Send Mail |

Figure 26 APPLIED CANDIDATES PAGES

Candidate's Profile who has applied to the job

The screenshot shows a web browser window titled "Job Praise" with the URL "localhost:8000/del/shirke@gmail.com/7". The page has a dark red header with the "Job Praise" logo on the left and navigation links for HOME, SERVICES, REGISTER, LOGIN, LOGOUT, and DASHBOARD on the right. Below the header is a table showing user details:

| SIDDHI SANTOSH SHIRKE DETAILS | |
|--|---|
| Email-ID | shirke@gmail.com |
| Contact Number | 8547120369 |
| Qualification | Higher Secondary(Science 12th Pass), Bachelor of Computer Applications |
| Birth -Date | Dec. 12, 2001 |
| What are your strengths and weakness? | Good in Communication |
| Describe your past experience in detail or any project that you have done related to our job profile if not then type N.A. | 2 years of experience in python programming |
| When could you start? | Immediately |
| Download CV | DOWNLOAD |
| View CV | View |
| Status | in-touch |
| SAVE | Save |

This is user details

| SIDDHI SANTOSH SHIRKE DETAILS | |
|--|---|
| Email-ID | shirke@gmail.com |
| Contact Number | 8547120369 |
| Qualification | Higher Secondary(Science 12th Pass), Bachelor of Computer Applications |
| Birth -Date | Dec. 12, 2001 |
| What are your strengths and weakness? | Good in Communication |
| Describe your past experience in detail or any project that you have done related to our job profile if not then type N.A. | 2 years of experience in python programming |
| When could you start? | Immediately |
| Download CV | DOWNLOAD |
| View CV | View |
| Status | in-touch |
| SAVE | Save |

Figure 27 DETAILS OF CANDIDATE PAGE

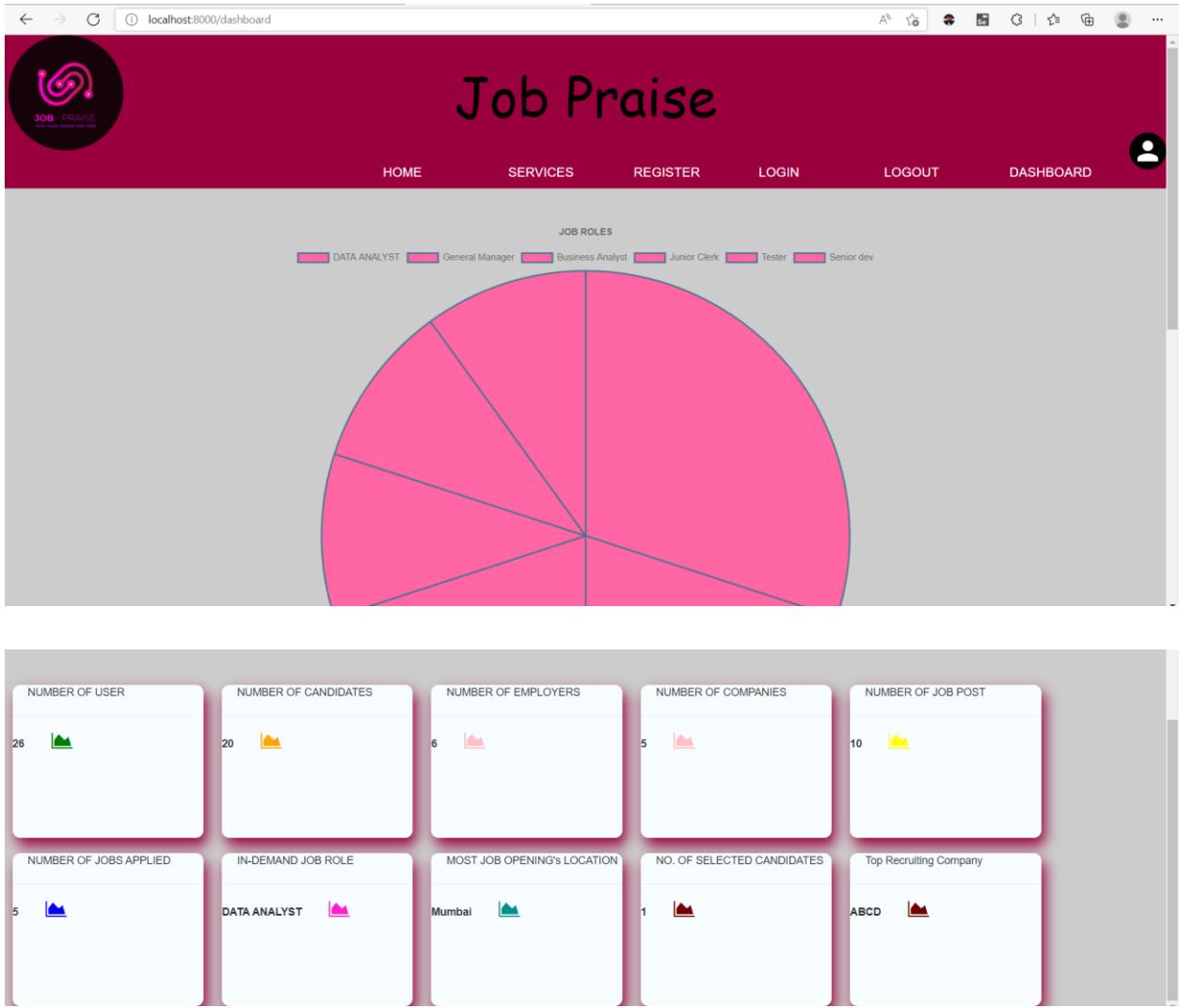


Figure 28 DASHBOARD

4.5 Security Issues:

1. Since this is a job portal information needs to be passed from the candidate to the employer and visa-versa. So to some extent data privacy can be leaked.
2. One can create a fake job post and then use the candidate's data for other reasons such as job consultancy, etc.
3. A candidate can fake their document and identity but this could be resolved at the time of the interview where a candidate has to carry all of this identity proofs.

Chapter 5: IMPLEMENTATION AND TESTING

5.1 Code (Place Core Segments)

```
def signup(request):
    form = UserForm
    if request.method == 'POST':
        form = UserForm(request.POST)
        if form.is_valid():
            print('valid')

            user = form.save()
            user.refresh_from_db()
            raw_password = form.cleaned_data.get('password')
            user.set_password(raw_password)
            user.save()
            login(request, user, backend='django.contrib.auth.backends.ModelBackend')

            if user.type == 'EMPLOYER':
                messages.success(request, 'Registered Successfully')
                return redirect('/empdel')
            else:
                messages.success(request, 'Registered Successfully')
                return redirect('/canddel')
        else:
            print('INVALID')
    return render(request, 'signup.html', {'form': form})
```

Figure 29 SIGN UP CODE

```

63     def signin(request):
64         form = AuthenticationForm()
65         if request.method == "POST":
66             email = request.POST['username']
67             password = request.POST['password']
68             # email = request.POST['email']
69             user = authenticate(request, email=email, password=password)
70             if user is not None:
71                 login(request, user)
72
73             if user.type == 'EMPLOYER':
74                 messages.info(request, 'You logged in.')
75                 return redirect('/choice')
76             else:
77                 messages.info(request, 'You logged in.')
78                 return redirect('/')
79
80         else:
81             messages.error(request, 'Invalid Credentials')
82             #return render(request, 'signin.html', {'form': form})
83             return redirect('/signin')
84     else:
85         print("Something went wrong")
86     return render(request, 'signin.html', {'form': form})

```

Figure 30 SIGN IN CODE

```

251     def home(request):
252         listopt = []
253         if request.method == "POST":
254             search = request.POST['searched']
255             print(search)
256             obj = (JobPost.objects.filter(skills_required__icontains = search) | JobPost.objects.filter(location__icontains = search) | JobPost.objects.filter(job_title__icontains = search))
257             print(obj)
258             if obj:
259                 for i in obj:
260                     if i.last_date_of_applying > date.today():
261                         listopt.append(i)
262             # else:
263             #     obj1 = JobPost.objects.all()
264             #     print(obj1)
265             #     for i in obj1:
266             #         if i.last_date_of_applying > date.today():
267             #             listopt.append(i)
268             #     print(listopt,"Else")
269
270         return render(request, 'search.html', {'search': search, 'obj': listopt})
271     else:
272         return render(request, 'home.html')
273

```

Figure 31HOME PAGE CODE

```

@login_required(login_url='/signin')
def jobpost(request):
    name = request.user.type
    email = request.user
    if name == 'EMPLOYER':
        cname1 = EmployerMore.objects.get(user_id=request.user) #initial={"name_of_comp":}
        cname = cname1.name_of_company
        print(cname1.name_of_company)
        form = JobPostForm()
        if request.method == 'POST':
            form = JobPostForm(request.POST)
            if form.is_valid():
                post = form.save(commit=False)
                email = request.user
                print(email)
                post.user = email
                post.save()
                messages.success(request, 'Job Posted Successfully')
                return redirect('/choice')
            else:
                messages.error(request, 'Invalid Details')
        else:
            return HttpResponseRedirect('<h1>You not an employer</h1> <a href="/signin">Go Back</a>')
    return render(request, 'employer/post.html', {'form': form})

```

Figure 32 JOB POST FORM CODE

```

@login_required(login_url='/signin')
def resume(request):
    candidate = User.objects.get(id=request.user.id)
    moredetail = CandidateMore.objects.get(user_id=request.user.id)
    pdf = render_to_pdf('resume.html', {'candidate': candidate, 'more': moredetail})
    return HttpResponseRedirect('resume.pdf')

@login_required(login_url='/signin')
def jobsapplied(request):
    user1 = CandidateMore.objects.get(user_id=request.user)
    applied = applied_jobs.objects.filter(candidate=user1)
    print(applied)
    return render(request, 'jobsapplied.html', {'applied': applied})

```

Figure 33 RESUME BUILDING AND JOBS APPLIED CODE

```

@login_required(login_url='/signin')
def applied(request,id):
    name1 = request.user.username
    if request.user.type == 'CANDIDATE':
        print(name1)
        post = JobPost.objects.get(id=id)
        user = User.objects.get(id=post.user_id)
        user1 = CandidateMore.objects.get(user_id = request.user)
        print(post, 'POST')
        print(user1,'user1')
        name = request.user.type
        name1 = request.user.email
        print(name1, 'I')
        apply = applied_jobs.objects.create(candidate=user1, post=post, employer=user)
        apply.save()
        messages.success(request,'Applied Successfully')
        # return redirect('/quest')
        return render(request, 'fillQ.html', {'post': post})
    else:
        return HttpResponse('<h1>You are not a candidate</h1> <a href="/signin">Go Back</a>')

```

Figure 34 JOBS APPLIED BY CANDIDATES

5.2 Testing Approach and Test Cases

Testing Approach:

A test approach is the test strategy implementation of a project which defines how testing would be carried out, defines the strategy which needs to be implemented and executed, to carry out a particular task.

Unit Testing:

Unit testing is testing a small unit of the system against its functionality. Here basically we check whether all the units are designed as per the requirements or not.

Integrated Testing:

Here all the units are integrated that are combined logically and then it is tested. Here we test the interaction between the modules.

System Testing:

The whole system is integrated and then testing is done against the functionality. It is concerned with the behavior of the whole system as defined by the scope of a development project.

Acceptance Testing:

The purpose of acceptance testing is to evaluate whether a system is acceptable for delivery and evaluate the system's working with the user and business requirements. This testing can also be performed by the client.

In this project, I have performed unit testing by testing all modules manually with a test case. I have built test-case in tabular formats as well I have written some code where the code checks the validation of the form is correctly implemented or not.

V-Model:

The methodology used in this project is V-Model. It is also known as the Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. That is after designing one module of the system testing is to be conducted and only then next phase will begin.

Test Cases for Registration Form:

| TEST CASE ID | DESCRIPTION | EXPECTED RESULT | ACTUAL RESULT | REMARK |
|--------------|---|---|---------------------|----------|
| TC-01 | Username: siddhi type: Candidate Email: siddhi@gmail.com Password: sid@1234 Confirm Password: sid@1234 | Account Created | Account Created | Pass |
| TC – 02 | Username: Shreya type: Employer Email: siddhi@gmail.com Password: shreya@1201 Confirm Password:shreya@1201 | Email ID already exists Account Not Created | Account Not Created | Not Pass |
| TC – 03 | Username: Shreya type: Employer Email: shreya@gmail.com Password: shre@1201 Confirm Password:shreya@1201 | Password and Confirm Password Not Matching Account Not Created | Account Not Created | Not Pass |

| | | | | | |
|---------|---|--|-----------------|-----|------|
| TC – 04 | Username:AA type: Candidate Email: anita@gmail.com Password:anita@1201 | The username should be a minimum of 3 characters | Account Created | Not | Pass |
|---------|---|--|-----------------|-----|------|

| | | | | |
|---------|---|---------------------|-----------------|------|
| | Confirm Password :anita@1201 | Account Not Created | | |
| TC – 05 | Username: Anita type: Candidate Email: anita@gmail.com Password: anita@1201 Confirm Password :anita@1201 | Account Created | Account Created | Pass |

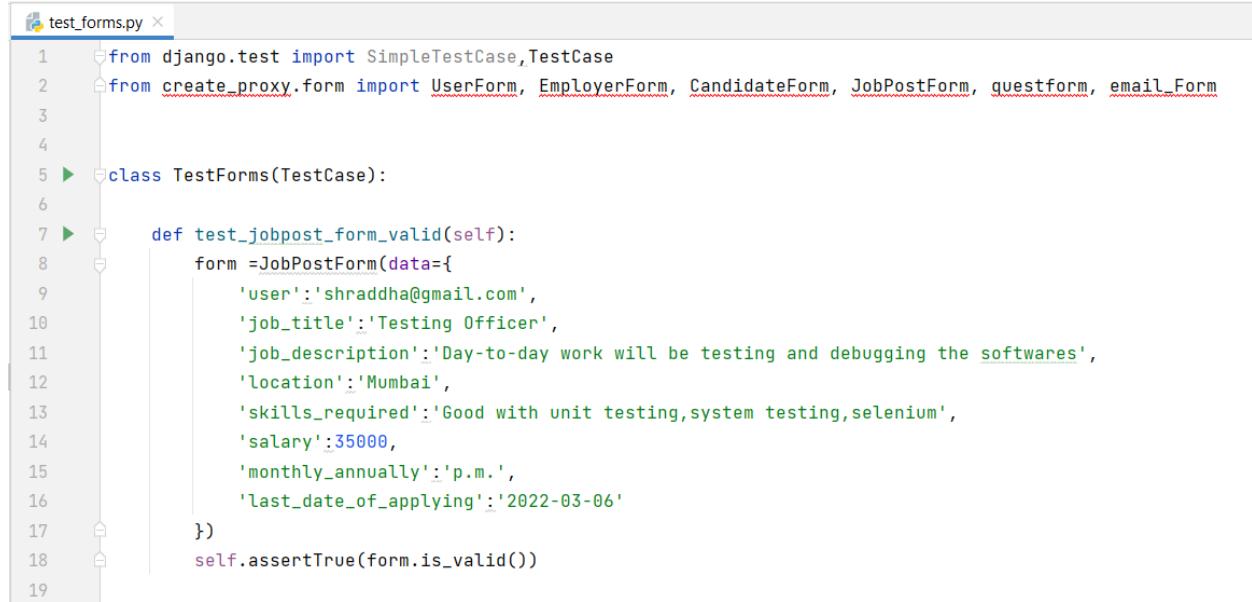
Table 1 MANUAL TESTING OF REGISTRATION FORM (UNIT TESTING)

Login Form:

| TEST CASE ID | DESCRIPTION | EXPECTED RESULT | ACTUAL RESULT | REMARK |
|--------------|---|-------------------------------------|---------------|--------|
| TC – 01 | Email: siddhi@gmail.com Password: sid@1234 | Grant Login | Login Granted | Pass |
| TC – 02 | Email: anita@gmail.com Password: anita@1201 | Grant Login | Login Granted | Pass |
| TC -03 | Email: nehagmail.com Password: neha@0911 | Deny Login | Deny Login | Pass |
| TC-04 | Email: siddhi@gmail.com Password: | Deny Login {Enter Your Password} | Deny Login | Pass |
| | | | | |

| | | | | |
|---------|-----------------------------|------------|------------|------|
| TC – 05 | Email: Password: sid@123 | Deny Login | Deny Login | Pass |
|---------|-----------------------------|------------|------------|------|

Table 2 MANUAL TESTING OF LOGIN FORM (UNIT TESTING)



```

1  from django.test import SimpleTestCase, TestCase
2  from create_proxy.form import UserForm, EmployerForm, CandidateForm, JobPostForm, questform, email_Form
3
4
5  class TestForms(TestCase):
6
7      def test_jobpost_form_valid(self):
8          form = JobPostForm(data={
9              'user': 'shradha@gmail.com',
10             'job_title': 'Testing Officer',
11             'job_description': 'Day-to-day work will be testing and debugging the softwares',
12             'location': 'Mumbai',
13             'skills_required': 'Good with unit testing, system testing, selenium',
14             'salary': 35000,
15             'monthly_annually': 'p.m.',
16             'last_date_of_applying': '2022-03-06'
17         })
18         self.assertTrue(form.is_valid())
19

```

Figure 35 TEST CASE OF JOB POST FORM



```

20  def test_jobpost_form_no_data(self):
21      form = JobPostForm(data={})
22      self.assertFalse(form.is_valid(), 8)
23
24      #INVALID DATE
25  def test_jobpost_form_invalid_date(self):
26      form = JobPostForm(data={
27          'user': 'shradha@gmail.com',
28          'job_title': 'Testing Officer',
29          'job_description': 'Day-to-day work will be testing and debugging the softwares',
30          'location': 'Mumbai',
31          'skills_required': 'Good with unit testing, system testing, selenium',
32          'salary': 35000,
33          'monthly_annually': 'p.m.',
34          'last_date_of_applying': '2021-03-06'
35      })
36      self.assertFalse(form.is_valid())
37

```

Figure 36 TEST CASE OF JOB POST FORM

```
test_forms.py ×
38 #INVALID SALARY
39 def test_jobpost_form_invalid_salary(self):
40     form = JobPostForm(data={
41         'user': 'shradhha@gmail.com',
42         'job_title': 'Testing Officer',
43         'job_description': 'Day-to-day work will be testing and debugging the softwares',
44         'location': 'Mumbai',
45         'skills_required': 'Good with unit testing,system testing,selenium',
46         'salary': -35000,
47         'monthly_annually': 'p.m.',
48         'last_date_of_applying': '2022-03-06'
49     })
50     self.assertFalse(form.is_valid())
51
52
53 def test_user_form_valid(self):
54     form = UserForm(data={
55         'type':'EMPLOYER',
56         'email': 'shradhha@gmail.com',
57         'username':'shradhha01',
58         'password':'Shrad@1234',
59         'confirm_password':'Shrad@1234'
60     })
61     self.assertTrue(form.is_valid())
62
63
```

Figure 37 TEST CASES OF JOB POST FORM AND USER FORM

```
64 #Password Not containing a Capital Letter
65 def test_user_form_password1test_invalid(self):
66     form = UserForm(data={
67         'type': 'EMPLOYER',
68         'email': 'shraddha@gmail.com',
69         'username': 'shraddha01',
70         'password': 'shrad@1234',
71         'confirm_password': 'shrad@1234'
72     })
73     self.assertFalse(form.is_valid())
74
75
76 #Confirm Password and Password not same
77 def test_user_form_password2test_invalid(self):
78     form = UserForm(data={
79         'type': 'EMPLOYER',
80         'email': 'shraddha@gmail.com',
81         'username': 'shraddha01',
82         'password': 'Shir@1234',
83         'confirm_password': 'Shrad@1234'
84     })
85     self.assertFalse(form.is_valid())
86
87
```

Figure 38 TEST CASES OF USER FORM

```
test_forms.py x
88     #Missing Special Characters
89     def test_user_form_password3test_invalid(self):
90         form = UserForm(data={
91             'type': 'EMPLOYER',
92             'email': 'shraddha@gmail.com',
93             'username': 'shraddha01',
94             'password': 'Shrad1234',
95             'confirm_password': 'Shrad1234'
96         })
97         self.assertFalse(form.is_valid())
98
99
100
101    # Invalid Email missing @@
102    def test_user_form_password4test_invalid(self):
103        form = UserForm(data={
104            'type': 'EMPLOYER',
105            'email': 'shraddhagmail.com',
106            'username': 'shraddha01',
107            'password': 'Shrad@1234',
108            'confirm_password': 'Shrad@1234'
109        })
110        self.assertFalse(form.is_valid())
111
112
```

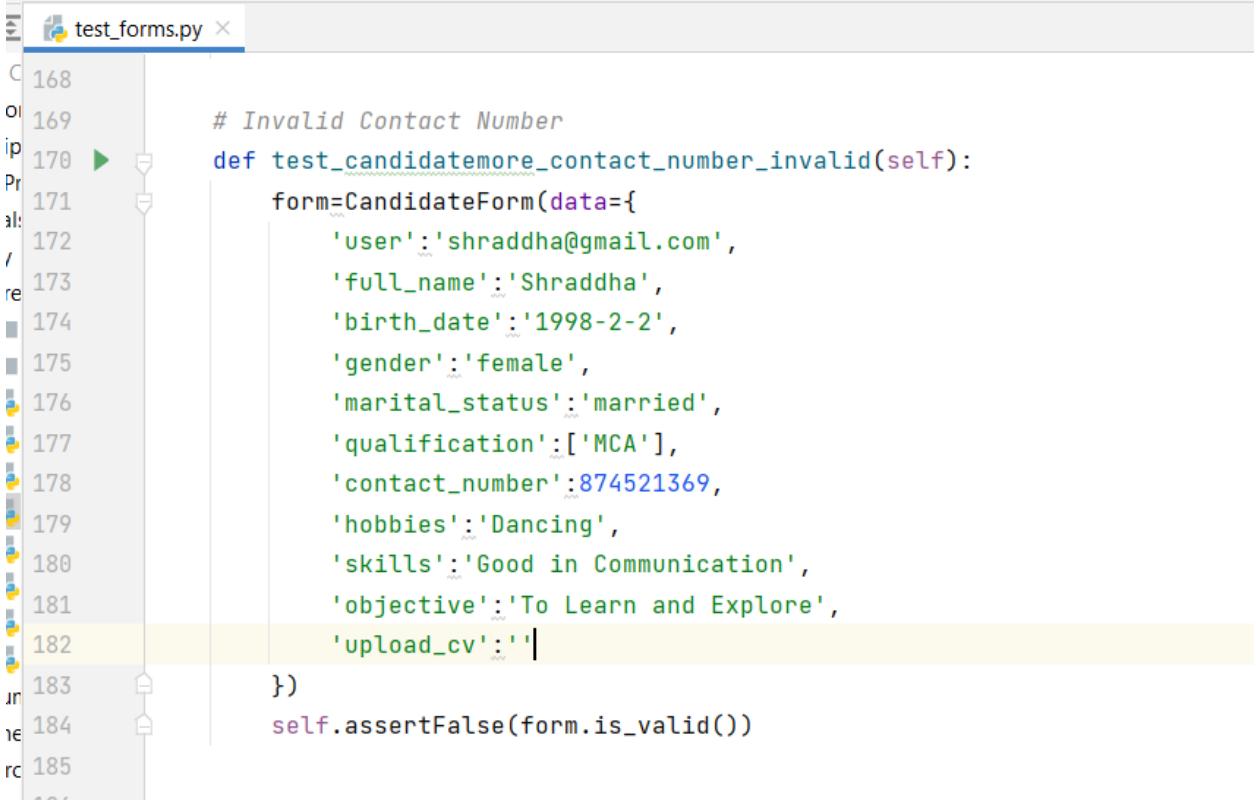
Figure 39 TEST CASE OF USER FORM

```
test_forms.py
116 def test_employermore_valid(self):
117     form =EmployerForm(data={
118         'user':'santosh@gmail.com',
119         'name_of_company':'ABC',
120         'website':'www.abc.com',
121         'contact_number':9874521360,
122         'company_email_id':'abc@gmail.com'
123     })
124     self.assertTrue(form.is_valid())
125
126 #Invalid company Email
127 def test_employermore_email_invalid(self):
128     form =EmployerForm(data={
129         'user':'santosh@gmail.com',
130         'name_of_company':'ABC',
131         'website':'www.abc.com',
132         'contact_number':9874521360,
133         'company_email_id':'abcmail.com'
134     })
135     self.assertFalse(form.is_valid())
136
```

Figure 40 TEST CASE OF EMPLOYER DETAILS FORM

```
test_forms.py
142     form = EmployerForm(data={
143         'user': 'santosh@gmail.com',
144         'name_of_company': 'ABC',
145         'website': 'www.abc.com',
146         'contact_number': 987452136,
147         'company_email_id': 'abc@gmail.com'
148     })
149     self.assertFalse(form.is_valid())
150
151
152     # Valid CandidateMore
153     def test_candidateform_valid(self):
154         form = CandidateForm(data={
155             'user': 'shraddha@gmail.com',
156             'full_name': 'Shraddha',
157             'birth_date': '1998-2-2',
158             'gender': 'female',
159             'marital_status': 'married',
160             'qualification': ['MCA'],
161             'contact_number': 8745213690,
162             'hobbies': 'Dancing',
163             'skills': 'Good in Communication',
164             'objective': 'To Learn and Explore',
165             'upload_cv': ''
166         })
167         self.assertTrue(form.is_valid())
168
```

Figure 41 JOB SEEKER's DETAILS FORM



```
168
169     # Invalid Contact Number
170 def test_candidatemore_contact_number_invalid(self):
171     form=CandidateForm(data={
172         'user':'shraddha@gmail.com',
173         'full_name':'Shraddha',
174         'birth_date':'1998-2-2',
175         'gender':'female',
176         'marital_status':'married',
177         'qualification':['MCA'],
178         'contact_number':874521369,
179         'hobbies':'Dancing',
180         'skills':'Good in Communication',
181         'objective':'To Learn and Explore',
182         'upload_cv':''}
183     )
184     self.assertFalse(form.is_valid())
185
186
```

Figure 42 JOB SEEKER's DETAILS FORM



```
187 def test_candidatemore_present_future_date_invalid(self):
188     form=CandidateForm(data={
189         'user':'shraddha@gmail.com',
190         'full_name':'Shraddha',
191         'birth_date':'2022-5-2',
192         'gender':'female',
193         'marital_status':'married',
194         'qualification':['MCA'],
195         'contact_number':8745213690,
196         'hobbies':'Dancing',
197         'skills':'Good in Communication',
198         'objective':'To Learn and Explore',
199         'upload_cv':''}
200     )
201
202     self.assertFalse(form.is_valid())
203
```

Figure 43 JOB SEEKER's DETAILS FORM

```

PS C:\Users\Shirke\PycharmProjects\Project\Proxy> python manage.py test create_proxy
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.True
.....2021-03-06 From Forms
2022-02-11
.2022-03-06 From Forms
2022-02-11
.None From Forms
2022-02-11
.2022-03-06 From Forms
2022-02-11

```

```

.ResolverMatch(func=create_proxy.views.choose, args=(), kwargs={}, url_name=choice, app_names=[], namespaces=[], route=choice)
.ResolverMatch(func=create_proxy.views.dashboard, args=(), kwargs={}, url_name=dashboard, app_names=[], namespaces=[], route=dashboard)
.ResolverMatch(func=create_proxy.views.displaydel, args=(), kwargs={'user_id': 'user', 'jobid': 12}, url_name=del, app_names=[], namespaces=[], route=del/<str:user_id>/<int:jobid>)
.ResolverMatch(func=create_proxy.views.delete_jobpost, args=(), kwargs={}, url_name=delete_jobpost, app_names=[], namespaces=[], route=delete_jobpost)
.ResolverMatch(func=create_proxy.views.delete_job, args=(), kwargs={'id': 12}, url_name=deletejob, app_names=[], namespaces=[], route=deletejob/<int:id>)
.ResolverMatch(func=create_proxy.views.employermore, args=(), kwargs={}, url_name=empdel, app_names=[], namespaces=[], route=empdel)
.ResolverMatch(func=create_proxy.views.home, args=(), kwargs={}, url_name=home, app_names=[], namespaces=[], route=)
.ResolverMatch(func=create_proxy.views.jobpost, args=(), kwargs={}, url_name=jobpost, app_names=[], namespaces=[], route=jobpost)
.ResolverMatch(func=create_proxy.views.jobsapplied, args=(), kwargs={}, url_name=jobsapplied, app_names=[], namespaces=[], route=jobsapplied)
.ResolverMatch(func=create_proxy.views.logout, args=(), kwargs={}, url_name=logout, app_names=[], namespaces=[], route=logout)
.ResolverMatch(func=create_proxy.views.candidate_status_mail, args=(), kwargs={'jobid': 12, 'user_id': 'user', 'id': 12}, url_name=mail, app_names=[], namespaces=[], route=mail/<int:jobid>/<str:user_id>/<int:id>)
.ResolverMatch(func=create_proxy.views.seeposted, args=(), kwargs={}, url_name=posted, app_names=[], namespaces=[], route=posted)
.ResolverMatch(func=create_proxy.views.questions, args=(), kwargs={'id': 12}, url_name=quest, app_names=[], namespaces=[], route=quest/<int:id>)
.ResolverMatch(func=create_proxy.views.resp, args=(), kwargs={}, url_name=res, app_names=[], namespaces=[], route=res)
.ResolverMatch(func=create_proxy.views.resume, args=(), kwargs={}, url_name=resume, app_names=[], namespaces=[], route=pdf/)
.ResolverMatch(func=create_proxy.views.show_user, args=(), kwargs={}, url_name=show, app_names=[], namespaces=[], route=show)
.ResolverMatch(func=create_proxy.views.show_applied, args=(), kwargs={}, url_name=showapply, app_names=[], namespaces=[], route=showapply)
.ResolverMatch(func=create_proxy.views.showdetail, args=(), kwargs={}, url_name=showdel, app_names=[], namespaces=[], route=showdel)
.ResolverMatch(func=create_proxy.views.signin, args=(), kwargs={}, url_name=signin, app_names=[], namespaces=[], route=signin)
.ResolverMatch(func=create_proxy.views.signup, args=(), kwargs={}, url_name=signup, app_names=[], namespaces=[], route=signup)
.ResolverMatch(func=create_proxy.views.update_candidate, args=(), kwargs={}, url_name=update, app_names=[], namespaces=[], route=update)
.ResolverMatch(func=create_proxy.views.update_employer, args=(), kwargs={}, url_name=update_emp, app_names=[], namespaces=[], route=update_emp)
.

-----
Ran 40 tests in 0.067s

OK
Destroying test database for alias 'default'...
PS C:\Users\Shirke\PycharmProjects\Project\Proxy> 
```

Figure 44 RESULTS OF TEST CASES

Testing All links against their views:

```
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from create_proxy.views import signup, signin, show_user, jobpost, candidatemore, employermore, home, applied, show_applied, showdetail, displaydel, questions, choose, resume

class TestUrls(SimpleTestCase):

    def test_home_url(self):
        url = reverse('home')
        print(resolve(url))
        self.assertEqual(resolve(url).func, home)

    def test_signup_url(self):
        url = reverse('signup')
        print(resolve(url))
        self.assertEqual(resolve(url).func, signup)

    def test_show_url(self):
        url = reverse('show')
        print(resolve(url))
        self.assertEqual(resolve(url).func, show_user)

    def test_signin_url(self):
        url = reverse('signin')
        print(resolve(url))
        self.assertEqual(resolve(url).func, signin)

    def test_jobpost_url(self):
        url = reverse('jobpost')
        print(resolve(url))
        self.assertEqual(resolve(url).func, jobpost)

    def test_candidatemore_url(self):
        url = reverse('candidatemore')
        print(resolve(url))
        self.assertEqual(resolve(url).func, candidatemore)

    def test_employermore_url(self):
        url = reverse('employermore')
        print(resolve(url))
        self.assertEqual(resolve(url).func, employermore)

    def test_homepage_url(self):
        url = reverse('home')
        print(resolve(url))
        self.assertEqual(resolve(url).func, home)

    def test_applied_url(self):
        url = reverse('applied')
        print(resolve(url))
        self.assertEqual(resolve(url).func, applied)

    def test_show_applied_url(self):
        url = reverse('show_applied')
        print(resolve(url))
        self.assertEqual(resolve(url).func, show_applied)

    def test_showdetail_url(self):
        url = reverse('showdetail')
        print(resolve(url))
        self.assertEqual(resolve(url).func, showdetail)

    def test_displaydel_url(self):
        url = reverse('displaydel')
        print(resolve(url))
        self.assertEqual(resolve(url).func, displaydel)

    def test_questions_url(self):
        url = reverse('questions')
        print(resolve(url))
        self.assertEqual(resolve(url).func, questions)

    def test_choose_url(self):
        url = reverse('choose')
        print(resolve(url))
        self.assertEqual(resolve(url).func, choose)

    def test_resume_url(self):
        url = reverse('resume')
        print(resolve(url))
        self.assertEqual(resolve(url).func, resume)
```

```
test_urls.py x
29 ► def test_jobpost_url(self):
30     url = reverse('jobpost')
31     print(resolve(url))
32     self.assertEquals(resolve(url).func, jobpost)
33
34 ► def test_canddel_url(self):
35     url = reverse('canddel')
36     print(resolve(url))
37     self.assertEquals(resolve(url).func, candidate)
38
39 ► def test_empdel_url(self):
40     url = reverse('empdel')
41     print(resolve(url))
42     self.assertEquals(resolve(url).func, employer)
43
44 ► def test_applied_url(self):
45     url = reverse('applied', args=[12])
46     print(resolve(url))
47     self.assertEquals(resolve(url).func, applied)
48
49 ► def test_showapply_url(self):
50     url = reverse('showapply')
51     print(resolve(url))
52     self.assertEquals(resolve(url).func, show_applied)
53
```

```
test_urls.py x
54 ► def test_showdel_url(self):
55     url = reverse('showdel')
56     print(resolve(url))
57     self.assertEqual(resolve(url).func, showdetail)
58
59 ► def test_del_url(self):
60     url = reverse('del', args=['user', 12])
61     print(resolve(url))
62     self.assertEqual(resolve(url).func, displaydel)
63
64 ► def test_quest_url(self):
65     url = reverse('quest', args=[12])
66     print(resolve(url))
67     self.assertEqual(resolve(url).func, questions)
68
69 ► def test_choice_url(self):
70     url = reverse('choice')
71     print(resolve(url))
72     self.assertEqual(resolve(url).func, choose)
73
74 ► def test_resume_url(self):
75     url = reverse('resume')
76     print(resolve(url))
77     self.assertEqual(resolve(url).func, resume)
78
```

```
test_urls.py x
79 ► def test_updatecandidate_url(self):
80     url = reverse('update')
81     print(resolve(url))
82     self.assertEquals(resolve(url).func, update_candidate)
83
84 ► def test_updateemp_url(self):
85     url = reverse('update_emp')
86     print(resolve(url))
87     self.assertEquals(resolve(url).func, update_employer)
88
89 ► def test_delete_jobpost_url(self):
90     url = reverse('delete_jobpost')
91     print(resolve(url))
92     self.assertEquals(resolve(url).func, delete_jobpost)
93
94 ► def test_logout_url(self):
95     url = reverse('logout')
96     print(resolve(url))
97     self.assertEquals(resolve(url).func, logout)
98
99 ► def test_dashboard_url(self):
100    url = reverse('dashboard')
101    print(resolve(url))
102    self.assertEquals(resolve(url).func, dashboard)
```

```

103
104 > def test_jobsapplied_url(self):
105     url = reverse('jobsapplied')
106     print(resolve(url))
107     self.assertEqual(resolve(url).func, jobsapplied)
108
109 > def test_posted_url(self):
110     url = reverse('posted')
111     print(resolve(url))
112     self.assertEqual(resolve(url).func, seeposted)
113
114 > def test_res_url(self):
115     url = reverse('res')
116     print(resolve(url))
117     self.assertEqual(resolve(url).func, resp)
118
119 > def test_delete_url(self):
120     url = reverse('deletejob', args=[12])
121     print(resolve(url))
122     self.assertEqual(resolve(url).func, delete_job)
123
124 > def test_mail_url(self):
125     url = reverse('mail', args=[12, 'user', 12])
126     print(resolve(url))
127     self.assertEqual(resolve(url).func, candidate_status_mail)

```

Figure 45 URL TESTING TEST CASES

Results

```

.....
.ResolverMatch(func=create_proxy.views.applied, args=(), kwargs={'id': 12}, url_name=applied, app_names=[], namespaces=[], route=applied/<int:id>
.ResolverMatch(func=create_proxy.views.candidatemore, args=(), kwargs= {}, url_name=canddel, app_names=[], namespaces=[], route=canddel)
.ResolverMatch(func=create_proxy.views.choose, args=(), kwargs= {}, url_name=choice, app_names=[], namespaces=[], route=choice)
.ResolverMatch(func=create_proxy.views.dashboard, args=(), kwargs= {}, url_name=dashboard, app_names=[], namespaces=[], route=dashboard)
.ResolverMatch(func=create_proxy.views.displaydel, args=(), kwargs={'user_id': 'user', 'jobid': 12}, url_name=del, app_names=[], namespaces=[], route=del/<str:user_id>/<int:jobid>
.ResolverMatch(func=create_proxy.views.deletejobpost, args=(), kwargs= {}, url_name=delete_jobpost, app_names=[], namespaces[], route=delete_jobpost)
.ResolverMatch(func=create_proxy.views.delete_job, args=(), kwargs={'id': 12}, url_name=deletejob, app_names=[], namespaces[], route=deletejob/<int:id>
.ResolverMatch(func=create_proxy.views.empmore, args=(), kwargs= {}, url_name=empdel, app_names=[], namespaces[], route=empdel)
.ResolverMatch(func=create_proxy.views.home, args=(), kwargs= {}, url_name=home, app_names=[], namespaces[], route=home)
.ResolverMatch(func=create_proxy.views.jobpost, args=(), kwargs= {}, url_name=jobpost, app_names=[], namespaces[], route=jobpost)
.ResolverMatch(func=create_proxy.views.jobsapplied, args=(), kwargs= {}, url_name=jobsapplied, app_names=[], namespaces[], route=jobsapplied)
.ResolverMatch(func=create_proxy.views.logout, args=(), kwargs= {}, url_name=logout, app_names=[], namespaces[], route=logout)
.ResolverMatch(func=create_proxy.views.candidate_status_mail, args=(), kwargs={'jobid': 12, 'user_id': 'user', 'id': 12}, url_name=mail, app_names=[], namespaces[], route=mail/<int:jobid>
>/<str:user_id>/<int:id>
.ResolverMatch(func=create_proxy.views.seeposted, args=(), kwargs= {}, url_name=posted, app_names=[], namespaces[], route=posted)
.ResolverMatch(func=create_proxy.views.questions, args=(), kwargs={'id': 12}, url_name=question, app_names=[], namespaces[], route=question/<int:id>
.ResolverMatch(func=create_proxy.views.resp, args=(), kwargs= {}, url_name=resp, app_names=[], namespaces[], route=resp)
.ResolverMatch(func=create_proxy.views.resume, args=(), kwargs= {}, url_name=resume, app_names=[], namespaces[], route=resume)
.ResolverMatch(func=create_proxy.views.show_user, args=(), kwargs= {}, url_name=show, app_names=[], namespaces[], route=show)
.ResolverMatch(func=create_proxy.views.show_applied, args=(), kwargs= {}, url_name=showapply, app_names=[], namespaces[], route=showapply)
.ResolverMatch(func=create_proxy.views.showdetail, args=(), kwargs= {}, url_name=showdel, app_names[], namespaces[], route=showdel)
.ResolverMatch(func=create_proxy.views.signup, args=(), kwargs= {}, url_name=signin, app_names[], namespaces[], route=signin)
.ResolverMatch(func=create_proxy.views.signup, args=(), kwargs= {}, url_name=signup, app_names[], namespaces[], route=signup)
.ResolverMatch(func=create_proxy.views.update_candidate, args=(), kwargs= {}, url_name=update, app_names[], namespaces[], route=update)
.ResolverMatch(func=create_proxy.views.update_employer, args=(), kwargs= {}, url_name=update_emp, app_names[], namespaces[], route=update_emp)
.
Ran 40 tests in 0.158s

```

Figure 46 RESULTS OF URL TESTING

System Testing

| 1 | TEST ID | Functions | Test Description | Steps | Test Data | Expected Result | Actual Result | Pass / Fail | Remark |
|---|---------|----------------------|--|---|--|------------------------------|--|-------------|--|
| 2 | TC-01 | Search and Apply | The system should allow the user to only view jobs and should not allow to apply for the job | Steps: 1. Start the application 2.Type "Python" in the search box. 3. Click Enter. 4. Jobs Will be displayed based on the search. 5. Click on "Apply" | User is not authenticated | Redirect to login page | User is Redirected to Login Page | Pass | Since user was not logged-in, he/she cannot apply for job |
| 3 | TC-02 | Search and Apply | The system should allow the user to only view jobs and apply for the job | Steps: 1. Start the application 2.Type "Python" in the search box. 3. Click Enter. 4. Jobs Will be displayed based on the search. 5. Click on "Apply" | User is authenticated | Applied Successfully | User is redirected to a page where he/she can answer the questionnaire | Pass | Since user was logged-in, he/she can apply for job and also answer the questionnaire |
| 4 | TC-03 | Resume | The system should allow the user to view their resume pre-built by the system | Steps: 1. Start the application 2.Click on "Services" and then click on "Resume". | User is authenticated | Resume is displayed | Resume is displayed | Pass | Since user is signed-in based on its profile resume is created and displayed |
| 5 | TC-04 | Updating Profile | The system should allow the user to update their profiles. | Steps: 1. Start the application 2.Click on "Services" and then click on "Update Profile". 3. Form is displayed to update details., User can now make changes to the profile. 4. Click on "Update" . | User is authenticated | Details Updated Successfully | Details Updated Successfully | Pass | Since user is signed-in based profile can be updated |
| 6 | TC-05 | Applied Jobs Display | The system should show the list of jobs to which the user has applied to | Steps: 1. Start the application 2. Click on "Services" and then click on "Jobs Applied". 3. List is displayed of jobs to which has applied along with the date of applications. | User is authenticated | Applied Jobs are displayed | Applied Jobs are displayed | Pass | Since user is signed-in so it easy to track which jobs user has applied to. |
| 7 | TC-06 | Registration | The system should display a registration form. The system should only accept a username which is of 3 or more character. Password should be atleast of 8 characters consisting of uppercase,lowercase, special character, atleast one digit. The password should match to the confirm password | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhi@gmail.com Type: CANDIDATE Password: Passwd@66 Confirm Password: Passwd@66 | Registered Successfully | Registered Successfully | Pass | Since all the details were valid then account is created |
| 8 | TC-07 | Registration | The system should display a registration form. The system should only accept a username which is of 3 or more character. Password should be atleast of 8 characters consisting of uppercase,lowercase, special character, atleast one digit. The password should match to the confirm password | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhi@gmail.com Type: CANDIDATE Password: Passwd@66 Confirm Password: Passwd@66 | Email-ID already exists | Error in the form."Email-ID already exists" | Pass | Since email-id used already exists in the database account is not created |

| | | | | | | | | | |
|----|-------|------------------------------|---|---|--|---|--|---------------------|--|
| | | | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhigmail.com Type: CANDIDATE Password: Passwd@66 Confirm Password: Passwd@66 | Not allowed to create account | Error is displayed for invalid email-id in the registration form. "Missing @" | Pass | Account not created | |
| 9 | TC-08 | Registration | The system should display a registration form. The system should only accept a username which is of 3 or more character. Password should be atleast of 8 characters consisting of uppercase,lowercase, special character, atleast one digit. The password shoul | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhi@gmail.com Type: CANDIDATE Password: Passww66 Confirm Password: Passww66 | Not allowed to create account | Error is displayed for invalid password in the registration form. "Password should contain atleast 1 of these @,#,\$,& special characters" | Pass | Account not created |
| 10 | TC-09 | Registration | The system should display a registration form. The system should only accept a username which is of 3 or more character. Password should be atleast of 8 characters consisting of uppercase,lowercase, special character, atleast one digit. The password shoul | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhi@gmail.com Type: CANDIDATE Password: Passww66 Confirm Password: Passww66 | Not allowed to create account | Error is displayed for invalid password in the registration form. "Password should contain atleast 1 of these @,#,\$,& special characters" | Pass | Account not created |
| 11 | TC-10 | Registration | The system should display a registration form. The system should only accept a username which is of 3 or more character. Password should be atleast of 8 characters consisting of uppercase,lowercase, special character, atleast one digit. The password shoul | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Registration Form: Username: Siddhi Email: siddhi@gmail.com Type: CANDIDATE Password: Passw@66 Confirm Password: Passw@66 | Not allowed to create account | Error is displayed for password not equal to confirm password in the registration."Password and Confirm Password are not equal" | Pass | Account not created |
| 12 | TC-11 | User Details Form(Candidate) | The system should allow the user to fill the detail only after registration. The system should accept past dates as birth dates, also it should be a date a which calculate the age to be 18 or above. Contact Number should be of 10 digits only | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Birth-Date: 12/12/2001 Contact Number:9632580147 | Details Entered Successfully | Details Entered Successfully | Pass | Since all the details were then profile details were entered |
| 13 | TC-12 | User Details Form(Candidate) | The system should allow the user to fill the detail only after registration. The system should accept past dates as birth dates, also it should be a date a which calculate the age to be 18 or above. Contact Number should be of 10 digits only | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Birth-Date: 12/12/2021 Contact Number:932580147 | Details Not Entered | Error displayed in the form "You seem to be young or you have entered wrong birth date" and "Contact Number should be of 10 digits" | Pass | Since birth date and contact number is invalid. |
| 14 | TC-13 | User Details Form(Employer) | The system should allow the user to fill in the detail only after registration. Email-id should be valid and contact number should be of 10 digits only | Steps: 1. Start the application 2. Click on "Register". 3. Registration form is displayed. 4. Click "Create Account". 5. If details are valid, redirect to a page to add more details of the candidate. 6. If details are valid then profile details are added. | Company-Email-ID: comp@gmail.com Contact Number: 9874520136 | Details Entered Successfully | Details Entered Successfully | Pass | Since all the details were then profile details were entered |

| | A | B | C | D | E | F | G | H | I |
|----|-------|-------------|--|---|--|----------------|---|------|---|
| 15 | TC-14 | Job Posting | The system should allow the user to post job if he/she is logged in to the system. The system should not accept past dates for last date for applying. | Steps: 1. Start the application. 2. Login (Employer) 3. Click on "Services",click on "Post Jobs". Form will be displayed to post job. 4. If all details are valid then job will be posted | Job type: Full-Time Job title: Business Analyst Job description: Python Location: Mumbai Skills required: Good Communication, Team Leader Salary: 100000 Total working hours: 5-7hrs Monthly or Annual:Monthly Last date of applying:02/18/2022 Job type: Full-Time Job title: Business Analyst Job description: Python Location: Mumbai Skills required: Good Communication, Team Leader Salary: 100000 Total working hours: 5-7hrs Monthly or Annual:Monthly Last date of applying:02/18/2022 | Job Posted | Job Posted | Pass | Since all details were valid job was posted successfully |
| 16 | TC-15 | Job Posting | The system should allow the user to post job if he/she is logged in to the system. The system should not accept past dates for last date for applying. | Steps: 1. Start the application. 2. Login (Employer) 3. Click on "Services",click on "Post Jobs". Form will be displayed to post job. 4. If all details are valid then job will be posted | Job type: Full-Time Job title: Business Analyst Job description: Python Location: Mumbai Skills required: Good Communication, Team Leader Salary: 100000 Total working hours: 5-7hrs Monthly or Annual:Monthly Last date of applying:02/18/2021 | Job Not Posted | Error in the form."Cannot Enter Past Dates" | Pass | Since all last date of applying was invalid. Job was not posted |

| | A | B | C | D | E | F | G | H | I |
|----|-------|--------------|--|---|---|----------------|---|------|---|
| 16 | TC-15 | Job Posting | The system should allow the user to post job if he/she is logged in to the system. The system should not accept past dates for last date for applying. | Steps: 1. Start the application. 2. Login (Employer) 3. Click on "Services",click on "Post Jobs". Form will be displayed to post job. 4. If all details are valid then job will be posted | Job type: Full-Time Job title: Business Analyst Job description: Python Location: Mumbai Skills required: Good Communication, Team Leader Salary: 100000 Total working hours: 5-7hrs Monthly or Annual:Monthly Last date of applying:02/18/2021 | Job Not Posted | Error in the form."Cannot Enter Past Dates" | Pass | Since all last date of applying was invalid. Job was not posted |
| 17 | TC-16 | Sending Mail | The system should allow the employer to send mail to the candidate regarding their applications status | The candidate will receive an mail regarding their status | Check if employer is logged -in | Mail Sent | Mail Sent | Pass | Mail is sent to the candidate to inform its application status along with the employers email-id. |

Table 3 SYSTEM TESTING

Chapter 6: RESULTS AND DISCUSSION

6.1 Test Reports

Job Praise is a web-based application. It has several functions, these functions are tested against various inputs. There are various modules such as registration, login, job post, profile, applied jobs candidate, etc. So, the functionality of each module is tested during unit testing. Also, these modules are further tested during system testing.

Unit testing and System testing are manually done i.e. it is done using test cases. Models are tested

Unit Testing Report

| FUNCTION | DESCRIPTION | NO. OF TEST CASES EXECUTED | % TEST CASES PASSED | RESULTS |
|----------------------|--|----------------------------|---------------------|---------|
| NEW USER | Register new user as an employer or candidate | 10 | 100% | Passed |
| LOGIN USER | Registered can login to the portal | 5 | 100% | Passed |
| POST JOB | Employers can post job | 4 | 100% | Passed |
| USER PROFILES | Candidates and Employers can fill in their details | 6 | 100% | Passed |
| URL MAPPING | Requested URLs Maps to the correct view | 24 | 100% | Passed |
| Total No. Test Cases | | 49 | | |

Table 4 UNIT TESTING REPORT

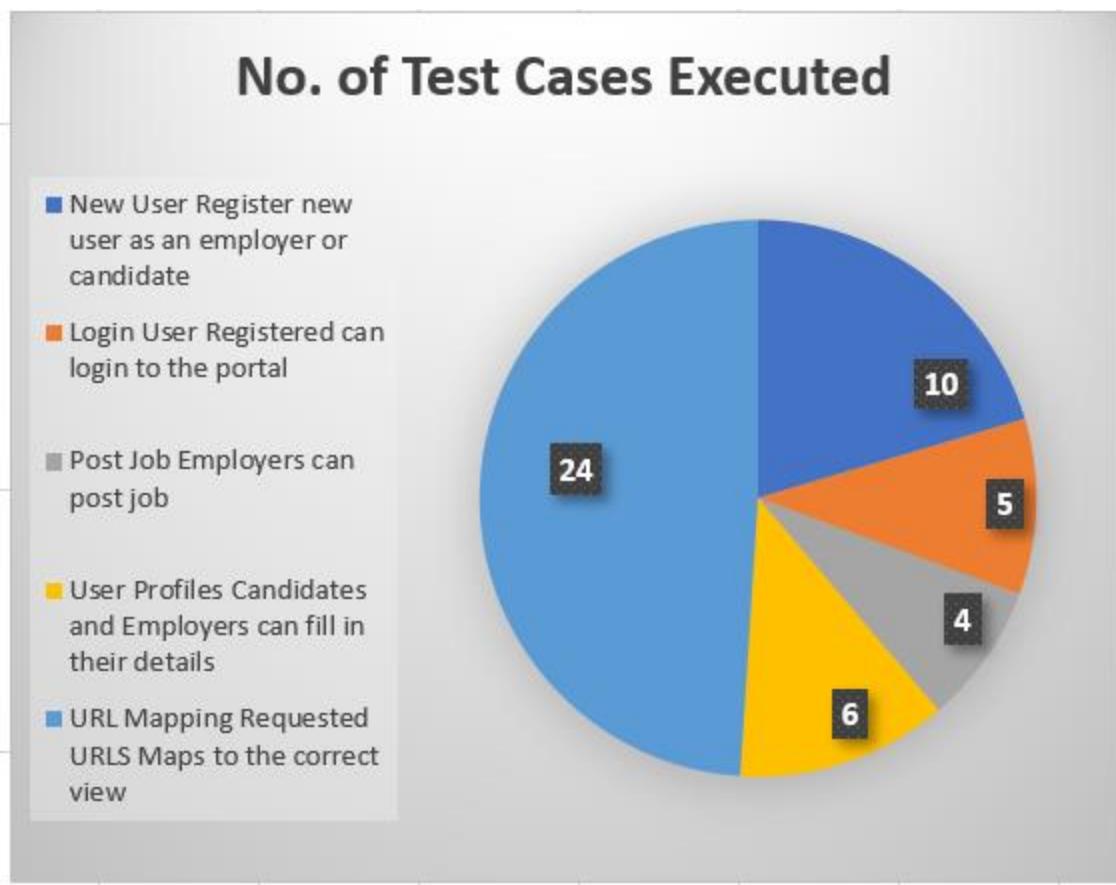


Figure 47 UNIT TESTING REPORT- PIE CHART

System Testing Report

| FUNCTION | DESCRIPTION | NO. OF TEST CASES EXECUTED | % TEST CASES PASSED | RESULTS |
|----------------------|--|----------------------------|---------------------|---------|
| Search and Apply | The system should allow the user to only view jobs and should not allow applying for the job | 2 | 100% | Passed |
| Resume | The system should allow the user to view their resume pre-built by the system | 1 | 100% | Passed |
| Updating Profile | The system should allow user to update their profiles. | 1 | 100% | Passed |
| Applied Jobs Display | The system should show the list of jobs to which the user has applied to | 1 | 100% | Passed |
| Registration | The system should display a registration form. The system should only accept a | 4 | 100% | Passed |

| | | | | |
|------------------------------|---|---|------|--------|
| | <p>username that is of 3 or more characters.</p> <p>Password should be at least 8 characters consisting of uppercase, lowercase, special character, at least one digit. The password should match the confirmed password</p> | | | |
| User Details Form(Candidate) | <p>The system should allow the user to fill in the detail only after registration.</p> <p>The system should accept past dates as birth dates, also it should be a date that calculates the age to be 18 or above.</p> <p>Contact Number</p> | 3 | 100% | Passed |

| | | | | |
|--------------|--|---|------|--------|
| | should be of 10 digits only | | | |
| Job Posting | The system should allow the user to post a job if he/she is logged in to the system. The system should not accept past dates for the last date for applying. | 2 | 100% | Passed |
| Sending Mail | The system should allow the employer to send mail to the candidate regarding the status of their application | 1 | 100% | Passed |

Table 5 SYSTEM TESTING REPORT

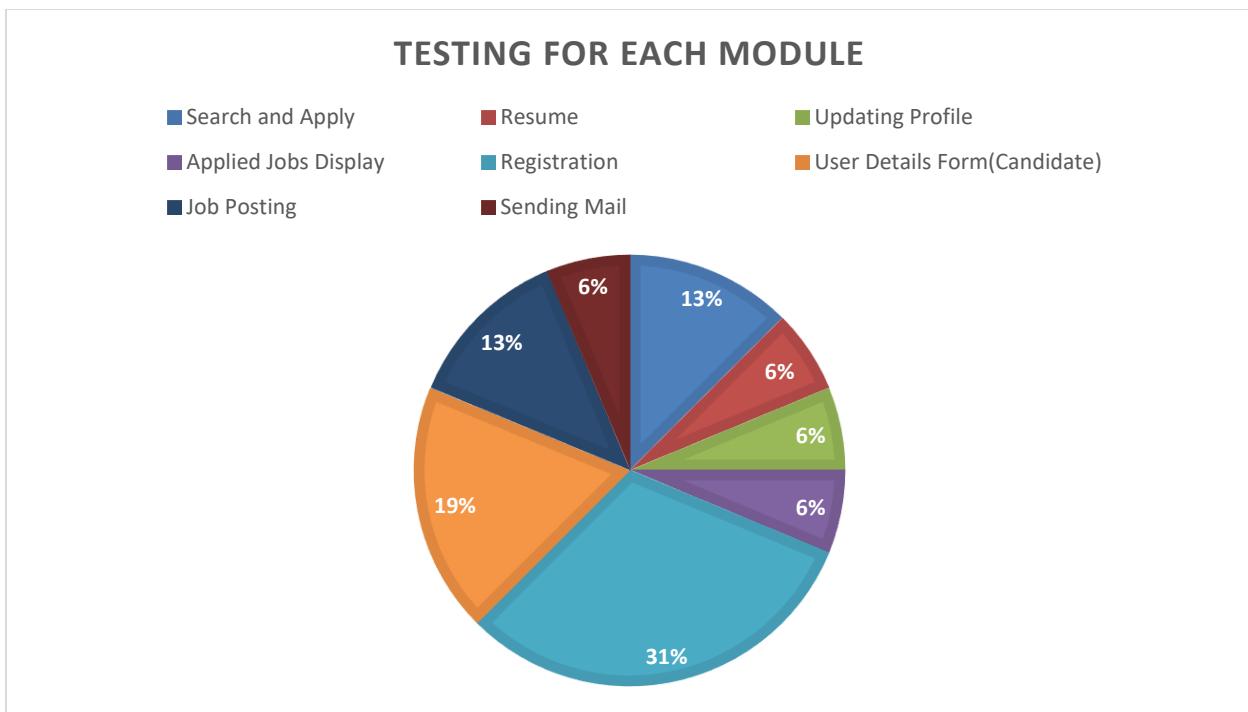


Figure 48 SYSTEM TESTING REPORT - PIE CHART

All Links are tested against their views and models are tested with their forms:

6.2 User Documentation

One has to register into the job portal to avail of its services. There are limited features available if you are not logged in to the system. There are 3 types of users in this portal namely admin, job seeker (candidate), employer. The features and functionality of all types of users will be mentioned. Each user type has different features to access.

ADMIN:

The admin of the site will have all the access. Admin can perform all the activities such as creating users, deleting users, modifying user details, etc. Firstly, the admin is to be created normal users have no access to create an admin. Admin is created internal commands this process is generally hidden and only known to the owner of the sites.

Features Of Admin:

1. Create, Delete, or Modify Users, Change User types.
2. Create Job Post on behalf of employers.
3. Apply to particular jobs on behalf of candidates.
4. View jobs applied, modify them, or delete them on the behalf of a candidate.

Job Seeker:

The job seekers are the ones who are looking for jobs on the portal. A job seeker can only search for jobs without registering to the portal. To avail of all the features, a job seeker needs to register himself or herself with the portal.

1. The user needs to first click on the Register on the home page.
2. Then the user just needs to enter his or her username, email, password, and in the “type of user” field select ‘Candidate’, which implies the user is a job seeker.
3. Then the user will be taken to the next page where he/she has to enter their details such as Full Name, Birth Date, Qualifications, Contact Number, Marital Status, Gender, Skills, CV, Objectives, Hobbies.
4. Now the user needs to login into the system, to apply for desired jobs and get selected in their desired companies.
5. The user can search for jobs according to their preferences of location, job title, based on their skills, salary required, and even by company name.
6. Then, if the user is logged in they can apply for jobs by just clicking on the apply button next to the job description.
7. After clicking on apply the user will be taken to a page where the job seeker has to answer basic questions which are set by the portal itself. The questions are,
 1. What are your strengths and weakness?
 2. Describe your experience in detail or any project that you have done related to our job profile if not then type N.A.
 3. When could you start?
 4. How will you handle a stressful situation. Let's say the deadline is near and teammates are not working properly?

5. What all things do you consider when you have to be quick while making a decision?
6. What are your short terms and long terms goals?
7. Describe honesty in your own words. And tell me a situation where you chose to be honest instead of a lie

Answering these questions is optional, but answering these questions will help the recruiter to make a better decision.

8. The job seeker can see the list of jobs that they have applied to. One can see the list by clicking on Services→Jobs Applied
9. A Resume is automatically generated based on the details entered by the job seeker. The Candidate can download and upload the resume on the portal. One can see their resume by clicking on Services→ Resume.
10. Candidate can even update their profile details, also one can update their resume by clicking on the profile icon on the top right corner.
11. One can also see the dashboard by clicking on “Dashboard”. By looking at the dashboard one can get an idea of the number of users selected, most in-demand job roles, most job-opening locations, etc.

Employer:

Employers are the ones who are in search of candidates for their organization. An employer needs to register to post jobs and receive applications from the candidates.

1. The Employer first needs to register by clicking on the register button. He / She has to enter the email id which is provided by the company or an email id that is used for business purposes, a password is to be set.
2. Then after clicking on submit, the employer will be redirected to a page where he/she has to enter the company details such as the name of the company, company's official email-id, website, and its contact number.
3. Then the employer needs to log in to the portal. After this, the employer can post any number of jobs by clicking on Post Jobs under Services.

4. After posting the jobs the employer can also view the applied candidates' details by clicking on Applied Candidates under Services.
5. Also, the employer can see the jobs posted him/her by clicking on See Job Posted under Services.
6. After viewing the jobs posted if there are any previous dates jobs whose deadlines are been crossed and when job post is of no use then it can be deleted by clicking on Delete Past Job Post under Services.
7. One can also see the dashboard by clicking on "Dashboard". By looking at the dashboard one can get an idea of the number of users selected, most in-demand job roles, most job-opening locations, etc.

Chapter 7: CONCLUSIONS

7.1 Conclusion

Nowadays since due to this pandemic and even before the pandemic people always used to prefer online over offline in many cases and so in while searching for jobs. Instead of using the traditional old methods like going newspapers in search of job etc. people are using online mediums to search for jobs. There are already many sites where job seekers can make a good profile and get a job in their desired company. I have worked on this project to showcase my skills and knowledge to learn something new from it.

I have overcome many drawbacks which existed in the previous system to some extent such as giving proper feedback from the employer to the candidate with no extra where there is no need for an employer to go out of their way to send the candidate personal feedback. The user interface is user-friendly. Moreover, job portals are now in demand as there are a lot number of job seekers and employers who are in search of candidates.

Also, nowadays many job portals charge a lot to employers to post their jobs, the famous the application is higher are the charges. So as a result, employers are always in search of new and upcoming job portals which would charge less and is affordable. To conclude, the project was aimed at self-development and all the necessary features were included in it.

7.1.1 Significance of the System

Online Job Portals are always in demand in freshers and employers this is the plus point of it. It aims to provide a platform where job seekers can search for jobs and employers can post job openings.

1. Ease To Use: The Job Portal is very user-friendly everyone can understand its features. Also, there are no complex services or methods of accessing the portals.
2. Time and Cost can be reduced: The employers can find the right candidate for their organization, there is a feature in this portal where the candidates have to answer certain questions based on the answers of the candidate the employer can shortlist them view their profile in one go also send an email of rejection or selection from the portal itself the employer doesn't have to go out of their way to send an

acknowledgment message to the candidate. Cost can be reduced as this will be a new job portal in the market so the cost of posting the jobs will be less compared to other job portals. So, no need to pay higher charges for the organization to find the right candidates.

7.2 Limitation

1. The application is a web-based application so it will require an internet connection.

7.3 Future Scope of the Project

The Job Portal ‘Job Praise’ fulfils all the requirements of the job seeker as well as of the employers. The current portal is web-based so the user will require to go on the site and access it, but in the future, a mobile application can be created for it, as mobile applications are more in-demand and find it easy to use. Further, a voice search can be created for the job post

REFERENCES

[1] Keethana Kopuri et al. (2017), the paper title “An Online Job portal management system”,

[\(PDF\) A Online Job portal management system \(researchgate.net\)](#)

[2] Malgorzata Mochol et al. (2007), the paper title “Improving the Accuracy of Job Search with Semantic Techniques.

[\(PDF\) Improving the Accuracy of Job Search with Semantic Techniques \(researchgate.net\)](#)

[3] Marjan Mansourvar and Norizan Binti Mohd Yasin, (2014), paper title “Development of a Job Web Portal to Improve Education

[\(PDF\) Development of a Job Web Portal to Improve Education Quality \(researchgate.net\)](#)

[4] Mustafa Pinjari et al. 2019 “Online Job Portal International Research Journal of Engineering and Technology (IRJET)

[IRJET-V6I43320190615-44144-cgrnqq-with-cover-page-v2.pdf \(d1wqtxts1xzle7.cloudfront.net\)](#)

[5] Salathiel Bogle et al. paper title “JOB SEARCH SYSTEM IN ANDROID ENVIRONMENT-APPLICATION OF INTELLIGENT AGENTS” International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.3, May 2012

[Intelligent agent based job search system in Android environment | Request PDF \(researchgate.net\)](#)

[6] Zarrin Tasnim et al. (2020), paper title “Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms”

[Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms - with-cover-page-v2.pdf \(d1wqtxts1xzle7.cloudfront.net\)](#)

APPENDIX

Literature Review

Mustafa Panjabi et al (2019), paper titled “Online Job Portal” the authors of this research paper explain the need to use the Online Job Portal. It is a web-based app. An Online Job Portal is a portal where a recruiter and job seeker can satisfy their need of searching for the right candidates and searching for the desired job respectively. This Job Portal is a Web-based Application making it free from any additional software installation for the users. Online Job Search Portal act as a bridge of communication between organizations and applicants. Basically, here employers will be able to post their jobs and review applications. Job Seekers can search for jobs in any fields. The author has managed to keep the user interface as friendly as possible so that it would be easy to use. The author further adds that many freshers who find it difficult to search for a job through the traditional methods so this portal will be helpful for them. The Existing System does not give us an idea of what the companies need from Job Seeker. Just mentioning the skills required for the job is not enough, in-depth specifications will help the candidate to self-examine whether they are fit to apply for the job.

As well as many candidates using the job portal will be freshers so since they don't have any practical knowledge about the work they should also be well addressed. So, to conclude all my findings and understandings will be used in my current project. According to the findings, we can say that the system is very basic and it requires lots of advanced features to stand out from the existing system. Working on the drawbacks that are found in the proposed system will make the system a bit better. So, to conclude the overall system will be helpful for the job seeker and the employers and will be more efficient if advanced features get added to the system.

Zarrin Tasnim et al, (2020), paper title “Implementation of an Intelligent Online Job Portal Using Machine Learning Algorithms”, the paper is related to Online Job Portal which is implemented using machine learning algorithms to make an efficient system that will help job seekers efficiently search for jobs. So, this system involves three phases namely EJC phase – Extract Job Circular, CSK phase – Cluster Similar Kind of job search, ENS phase – Email

Notification Send. Through these 3 phases, an efficient online job portal is built. EJC Phase: In this phase, the job circular post is extracted from the career section of the company's website. For this task, a web crawler is implemented. The web crawler extracts the data and then stores it in the database. What is a Web Crawler? The web crawler is a computer program or automated script that crawls through the World Wide Web in a predefined and methodical manner to collect data. ENS Phase: In this phase, a decision tree algorithm is implemented that generates a list of candidates who are right for the job circular this decision is based on three attributes namely job position, experience, and expected salary, these data are compared from the job seekers and job circular data. Here the email is sent to those candidates whose email is received as the output of the decision tree. CSK Phase: In this phase, the K-means algorithm is implemented, this gives the cluster of similar kinds of jobs in the dataset, where the clustering is done on basis of job titles. Through this process, the market demand is analyzed as a higher density of clusters indicates higher demand. The implementation of machine learning algorithms in online job portals improves the efficiency of the job portals. In the EJC phase, the updated job posts are directly extracted and stored in the system's database which gives the candidates the right information of job post. In the ENS phase, an email notification is sent to only those candidates whose profiles match the job circulars requirements so no unnecessary or irrelevant notifications are sent to the job seekers. In the CSK phase, by analyzing the density of the cluster furthermore, training programs can be implemented to help the job seeker to get better jobs.

Keethana Kopuri et al, (2017), paper title “A Online Job portal management system”, the author of this paper explains that there is a huge competition in every professional field. A Job Portal is a platform where recruiters as well as job seekers interact with each other and get their information. A recruiter will be able to find the right candidate for his/her organization and a candidate will be able to find the best-suited job role based on his/her educational qualifications, experience, and job preferences. The main objective of this application is to enable interactions between employers and job seekers. A Job Seeker can search for a jobs view and start applying for the desired post. After applying they can apply online to the employer. A registered user will also have access to the information regarding placement papers and sample resumes which in turn will help them to create their resume according to industry standards. The job seeker can edit or modify the profile and can also change their password. An Employer can advertise their

vacancies by registering and also posting the job information along with the eligibility criteria. An Employer can search resumes. An Employer can select the best-suited candidate for their organization. The author explains that this job portal will be very helpful as it allows the users of different profiles to upload CVs, search for jobs based on their qualifications. Job Seeker can track their job applications. Employers can also search for required people. This job portal is very fast and efficient, it highly secure and portable, also it provides an efficient search mechanism. The system does need any extra sophisticated training.

Malgorzata Mochol et al, (2007), paper title “Improving the Accuracy of Job Search with Semantic Techniques”, the author of this paper introduces a prototype job portal that uses semantically annotated job offers and applicants. Also, the author states that the internet is an effective communication medium. Further the author’s research that about 90% of human resource managers in Germany have rated the Internet as an important recruitment channel. The reason that the internet has a high rate is that the internet is reaching applicants who are young and qualified. The companies generally use more than one channel for posting their job opening, they assume that applicants go through every site. Further, the old and new ways of job procurement are stated as Since a large number of online job portals have developed a job seeker can’t check out every site and get the right information about the job post. On the other hand, the most in-demand portals charge employers high fees for publishing open positions which increases the expenses of the employers and therefore employers start posting their job openings on smaller platforms. Here the hiring organizations assume that the job seeker will visit multiple sites. Using Semantic Web technologies in the domain of online recruitment will reduce the transaction cost for employers, increase market transparency, and speed up the procurement process. In Semantic Web-based recruitment the data is exchanged between employers, applicants, and the job portal is based on a set of vocabularies that provide shared terms to describe occupations and industrial sectors and job skills. Semantic Matching is a technique that combines annotations using controlled vocabularies with background knowledge about a certain application domain.

Salathiel Bogle et al, (2012), the paper title “JOB SEARCH SYSTEM IN ANDROID ENVIRONMENT-APPLICATION OF INTELLIGENT AGENTS”, the author in this paper have addressed the problem in the job selection process which is faced by prospective employees no matter how experienced they are. The traditional method involves searching newspapers, job websites, human agents, etc. Nowadays various sites are available where the employers and job seekers have to register and post and apply for jobs respectively, but the validity of data submitted by the employees has not been checked as well any proper feedback from employers on the applications submitted by the employers. Considering all these problems the proposed system has an intelligent agent instead of a human agent to perform the same search operations by interacting with the employer and job search co-ordinator agent. The system intelligently anticipates the needs of the user and makes intelligent decisions based on fuzzy preference rules and dynamically makes location, salary markup, markdown, and allowance choices that are perceived benefits to the user. The application agent performs the activities of a human agent for job search and is a key entity in the process. The Applicant Agent’s primary responsibilities are to submit a search request, allow configuration of job search importance preference matrix, submit employer rating data, present search details in a user-friendly format. Job Search Agent is the brain behind the job search process and equivalent in role to online job search site, local newspaper, printed media, etc. it interacts with employer agents to acquire job listings and also uses fuzzy preference rules. The Employer Agent responsibilities are posting job vacancies listed by employers, allowing configuration of job importance, interacting with job search agents to ensure only valid jobs appear in the listings. So basically, the proposed system intelligently predicts the needs of the user and makes decisions accordingly.

Marjan Mansourvar and Norizan Binti Mohd Yasin, (2014), paper title “Development of a Job Web Portal to Improve Education Quality”, the author of this research paper aims to develop a web portal for the student of IT and CS, of Malaya University (UM), so that they can connect to industries interact with them as well as find a right IT job after their graduation. Also, a survey was conducted to know the problem faced by the student with the existing system. Unemployment is one of the serious problems in developing as well as in developed countries. One of the reasons for this problem is the lack of information among people about various opportunities which means there are certain jobs but people are unaware of them. Online job portals are efficient as they provide all the information about the available vacancies. Also, a job

seeker can submit their application on it, as well as search for jobs based on their skills, qualifications. The research has used the questionnaire to conduct the survey. The data collected indicates that students are not satisfied with the current portal, they say that they don't have proper access to information when they are looking for a job. All these problems are considered during the development of the new web portal. The research attempts to provide a new link between students, unemployed graduates, and the IT industry. This relationship illustrates how knowledge-sharing is done through an online system to help students to find jobs. The author points out that the job web portals can never fulfill all the problems of jobless graduates.

SWOT ANALYSIS

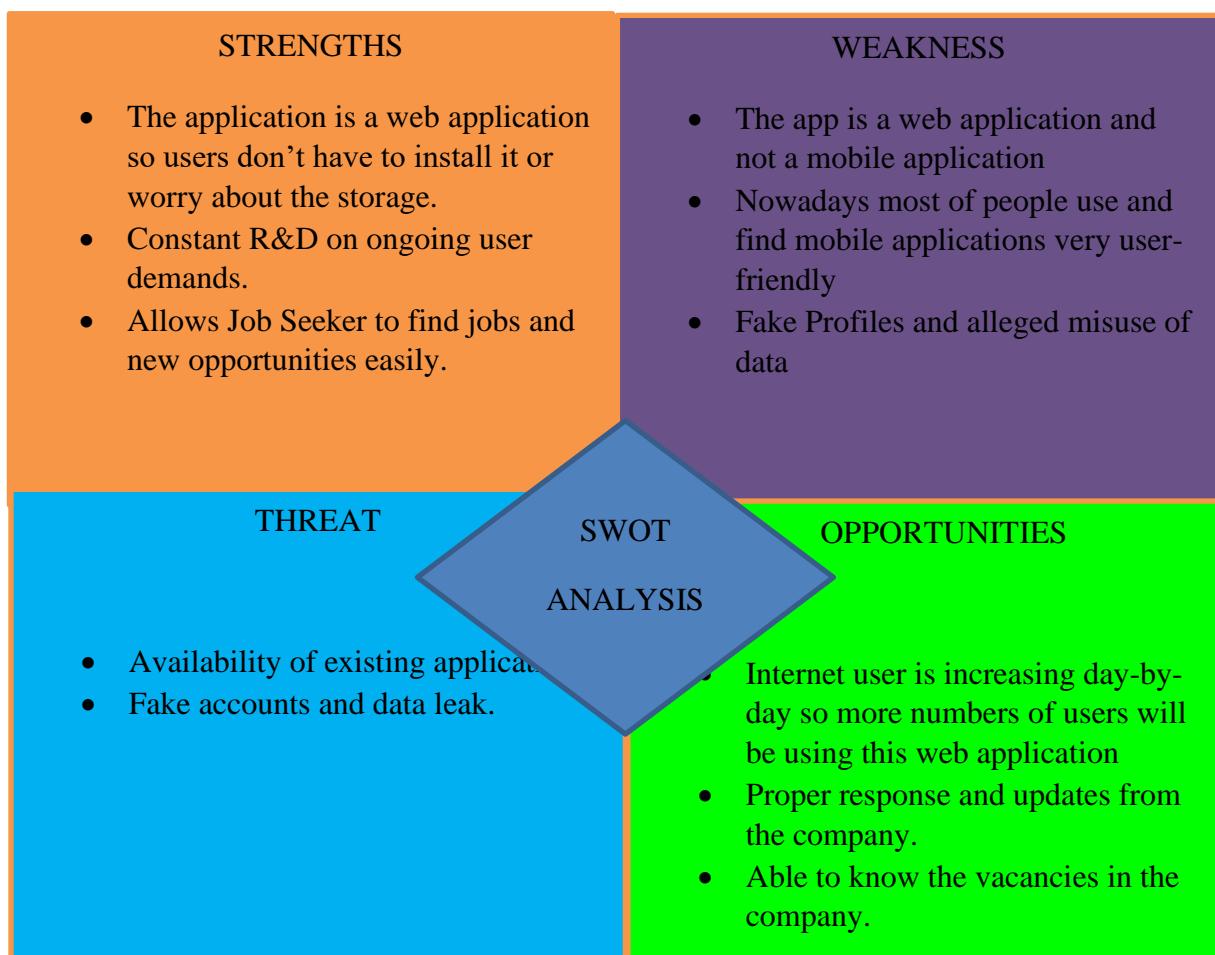


Figure 49 SWOT ANALYSIS

Plagiarism

Part 1

| | |
|---|---|
| Report Title: | Plagiarism Part 1 |
| Report Link: (Use this link to send report to anyone) | https://www.check-plagiarism.com/plag-report/6841459339cd227fe6da5b0b9ef9cd195f9171648357233 |
| Report Generated Date: | 27 March, 2022 |
| Total Words: | 2442 |
| Total Characters: | 14879 |
| Keywords/Total Words Ratio: | 0% |
| Excluded URL: | No |
| Unique: | 96% |
| Matched: | 4% |

Sentence wise detail:

Job Search Portal is a web application, which allows Job Seekers to find available jobs and Employers to find eligible job seekers for their company.

Online Job Search Portal is a medium using which organizations and applicants can communicate. Here employers will be able to post their jobs and review applications.

Job Seekers can search for jobs in any field. Can upload their resumes. (0)

"JOB PRAISE" will be an online job searching portal.

It will have basic features of the job portal like job seekers will be able to find jobs which will be posted by the registered companies.

These companies will also be verified to some extent.

It will be a platform where job seekers and employers connect.

It aims to be a platform that can be used from fresher to an experienced person.

Job Seekers can upload their resumes or make one if they don't have one and apply to the jobs posted by the employers.

Also, at a time they can apply to multiple companies.

Employers have to register their company on the portal and then they have to post a job for candidates to apply for.

They can then select the candidates based on their skills of the job seeker they can shortlist or select candidates for offline interviews or any other process.

An employer can post multiple job posts. 1.

To build a base for the job portal 2.

To build a system where users can search for jobs and apply for them. 3.

To build a system where the employer can register their company and post several job vacancies. 4.

To build a system that helps employers to select the best candidates through less effort. i. e.
by doing Skill Testing 5.

To build a system that allows job seekers to filter job postings according to their requirements.

Location, Salary Range, Job Type, Education Level, etc. 6.

To build a system that allows the recruiters to shortlist candidates and also send a notification to the candidates through the mail. 7.

To provide a resume based on the user's profile. 1.

Fresher or Job Seeker can Search for Jobs Online. 2.

Employers can post their vacancies and get an eligible candidate for their company. 3.

Job Seekers can get to know on-demand skills and industrial requirements. 4.

Job Seekers are exposed to the large corporate world.

CHAPTER 2: GAP ANALYSIS/ DRAWBACK OF EXISTING SYSTEM Through gap analysis, we get to know what is wrong with the existing systems, and then we can further think of how we can rectify those mistakes in our system.

The gap analysis also helps us to set the objectives of the system.

We get a clear vision of what and how we are supposed to build our new system.

We can overcome this problem by making the candidates answer certain questionnaires and based on their answers the employers can decide whether to select or reject the candidate. 3. (1)

Fake Job Postings or Duplication of Jobs □ Proper verification should be conducted on the company's existence 4.

No proper feedback through the portal after the recruiter views a profile.

□ After applying to a job post no proper feedback from the employer whether a candidate is selected or not. **CHAPTER 3: REQUIREMENT AND ANALYSIS 3. (2)**

1 Problem Definition Online Job Search Portal is a commonly available app in today's times.

The development of the "JOB PRAISE" job portal is basically for self-assessment and self-development.

However, through some research, I have figured out certain gap analysis in the existing portals.

The basic functionality of the job portal is to make allow the job seeker to search and get information about the job and then based on its skills allow the job seeker to apply for the job.

One job seeker and apply to as many job posts as possible.

The Job Portal should allow the employer to post job details and select the candidates and call for an interview.

The Existing systems do not check for the qualification of the candidate before allowing him/her to apply for the job post which eventually makes it the employer difficult to shortlist the candidates.

The second problem that employers encounter is a large number of candidates applying to the job post.

So, imagine if one employer is posting 3 job posts then how difficult would it be for the employer to shortlist the candidate.

Also, no proper feedback from the employer if the candidate is been shortlisted or rejected after applying for the job post. 3.

2 Requirements Specification 3.2. 1 Purpose: The purpose of this document is to build an "Online Job Search Portal" as a bridge of communication between organizations and applicants. (3)

This information is used for verification, for contacting the candidates in case of the selection, and for providing qualification details to the employer.

Company Details: It includes company name, contact number, official website, email id.

This information is used for verification and for providing contact details to candidates when they are selected.

Part 2

| | |
|---|---|
| Report Title: | Part 2 |
| Report Link: (Use this link to send report to anyone) | https://www.check-plagiarism.com/plag-report/68414323d32d12eabb46166eeaf162a7c2b451648358502 |
| Report Generated Date: | 27 March, 2022 |
| Total Words: | 2035 |
| Total Characters: | 12127 |

| | |
|------------------------------------|-----|
| Keywords/Total Words Ratio: | 0% |
| Excluded URL: | No |
| Unique: | 95% |
| Matched: | 5% |

Sentence wise detail:

One has to register into the job portal to avail of its services.

There are limited features available if you are not logged in to the system.

There are 3 types of users in this portal namely admin, job seeker (candidate), employer.

The features and functionality of all types of users will be mentioned.

Each user type has different features to access.

The admin of the site will have all the access.

Admin can perform all the activities such as creating users, deleting users, modifying user details, etc.

Firstly, the admin is to be created normal users have no access to create an admin.

Admin is created internal commands this process is generally hidden and only known to the owner of the sites. 1.

Create, Delete, or Modify Users, Change User types. 2.

Create Job Post on behalf of employers. 3.

Apply to particular jobs on behalf of candidates.

The job seekers are the ones who are looking for jobs on the portal.

A job seeker can only search for jobs without registering to the portal.

To avail of all the features, a job seeker needs to register himself or herself with the portal. 1.

The user needs to first click on the Register on the home page. 2.

Then the user just needs to enter his or her username, email, password, and

in the "type of user" field select 'Candidate', which implies the user is a job seeker. 3. (0)

Then the user will be taken to the next page where he/she has to enter their

details such as Full Name, Birth Date, Qualifications, Contact Number, Marital Status, Gender, Skills, CV, Objectives, Hobbies. 7. (1)

After clicking on apply the user will be taken to a page where the job seeker has to answer basic questions which are set by the portal itself. 8.

The job seeker can see the list of jobs that they have applied to.

One can see the list by clicking on Services||Jobs Applied 9.

A Resume is automatically generated based on the details entered by the job seeker.

The Candidate can download and upload the resume on the portal.

One can see their resume by clicking on Services|| Resume. 10.

Candidate can even update their profile details, also one can update their resume by clicking on the profile icon on the top right corner. 11.

One can also see the dashboard by clicking on "Dashboard".

By looking at the dashboard one can get an idea of the number of users selected, most in-demand job roles, most job-opening locations, etc.

A Job Seeker can search for a job view and start applying for the desired post. After applying they can apply online to the employer. The author explains that this job portal will be very helpful as it allows the users of different profiles to upload CVs, search for jobs based on their qualifications. (3)

Job Seeker can track their job applications.

Employers can also search for required people.

This job portal is very fast and efficient, it is highly secure and portable, also it provides an efficient search mechanism.

The system does not need any extra sophisticated training.

Also, the author states that the internet is an effective communication medium. It increases market transparency, and speeds up the procurement process.

In Semantic Web-based recruitment the data is exchanged between employers, applicants, and the job portal is based on a set of vocabularies that provide shared terms to describe occupations and industrial sectors and job skills.

Semantic Matching is a technique that combines annotations using controlled vocabularies with background knowledge about a certain application domain.

The paper has addressed the problem in the job selection process which is faced by prospective employees no matter how experienced they are.

The traditional method involves searching newspapers, job websites, human agents, etc.

Nowadays various sites are available where the employers and job seekers have to register and post and apply for jobs respectively, but the validity of data submitted by the employees has not been checked as well as any proper feedback from employers on the applications submitted by the employers.

Considering all these problems the proposed system has an intelligent agent instead of a human agent to perform the same search operations by interacting with the employer and job search co-ordinator agent.

The system intelligently anticipates the needs of the user and makes intelligent decisions based on fuzzy preference rules and dynamically makes location, salary markup, markdown, and allowance choices that are perceived benefits to the user.

The application agent performs the activities of a human agent for job search and is a key entity in the process.

The Employer Agent responsibilities are posting job vacancies listed by employers, allowing configuration of job importance, interacting with job search agents to ensure only valid jobs appear in the listings. (4)