Contents

[INTRODUCTION 3](#_Toc153752973)

[PROPOSED SYSTEM 4](#_Toc153752974)

[ARCHITECTURE DIAGRAM 5](#_Toc153752975)

[Technology Used 6](#_Toc153752976)

[ER DIAGRAM 7](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752977)

[Software Designing Architecture: MVT 8](#_Toc153752978)

[LOGIN 8](#_Toc153752979)

[REGISTRATION 10](#_Toc153752980)

[Recruiter 11](#_Toc153752981)

[ADMINISTRATOR 12](#_Toc153752982)

[JOB SEEKER 14](#_Toc153752983)

[CV generator 15](#_Toc153752984)

[Future Enhancement 16](#_Toc153752985)

[CONCLUSION 17](#_Toc153752986)

[REFERENCES 17](#_Toc153752987)

# Table of figure

[Figure 1: Architecture Diagram 5](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752052)

[Figure 2: ER diagram 7](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752053)

[Figure 3: Admin Login 8](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752054)

[Figure 4: Recruiter Login 8](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752055)

[Figure 5: User Login 8](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752056)

[Figure 6: Home page 9](#_Toc153752057)

[Figure 7: Registration page for User 10](#_Toc153752058)

[Figure 8: Registration page for Recruiter **Error! Bookmark not defined.**](file:///H:\Study%20Materials\4.1\SAProject\project%20report%20new.docx#_Toc153752059)

[Figure 9: Recruiter Dashboard 11](#_Toc153752060)

[Figure 10: Add Job post 12](#_Toc153752061)

[Figure 11: View applied candidates 12](#_Toc153752062)

[Figure 12: Admin Dashboard 13](#_Toc153752063)

[Figure 13: Admin manages recruiters 13](#_Toc153752064)

[Figure 14: Admin manages users 13](#_Toc153752065)

[Figure 15: Job seekers dashboard 14](#_Toc153752066)

[Figure 16: Job list shown by the job seeker 15](#_Toc153752067)

[Figure 17: CV generator form 15](#_Toc153752068)

[Figure 18: Output of the CV 16](#_Toc153752069)

Abstract

Our current generation uses the internet for everything right from shopping to getting hired. In this project, I attempt to address and minimize the gap between the job seeker and the recruiter through this ONLINE JOB PORTAL (Web Application) using python Django. This is done by considering details of both the job seeker and the recruiter and by applying a variety of filters to satisfy their requirements. Job seekers can search for vacancies according to their location, preferred job as well as a recruiter can easily find suitable and eligible candidates using the job seeker's resume and profile details. In addition to this, there will be an administrator to manage and authenticate the system services. I am using MVT, a software designing architecture, to design this system. This application provide updates on all job availabilities posted within the past 15 days along with their opening and closing dates.

# INTRODUCTION

Nowadays various problems can be solved using portals of different applications or services.

One such is a portal of requirements for the person who is seeking a job. Nowadays, online recruitment has become the standard method for employers.

As of today, an infinite number of online job portals are available dividing the labor market into information rooms, making it a path to find a job. Yet the current system needs to be upgraded to make the portals work easier for the recipients.

Hence, we have added the feature of detecting the search error using A.I and we also provided informative videos to uplift their career. This Project is done by using Python Django, HTML, CSS, JavaScript, and Bootstrap 5. We have used MVT (Models, Views, Template) to design this project.

In this application there are mainly three types of actor, one is the job seekers, one is the employers or recruiters and the admin panel.

By this application system the jobseekers can register themselves and find jobs of their choice and apply for them by the requirements.

The recruiters can post Job vacancies and manage the applied jobseekers.

The admin panel mainly manages all the users.

# PROPOSED SYSTEM

In the proposed system, I am intended to make the portal user-friendly. Providing updates on all job availabilities posted in the past along with their opening and closing dates on the home page. In this project, there are searching job option by their location and job title. I am using Models-Views-Template (MVT), software designing architecture, to design the system where ‘Models’ connect the frontend and backend, “Views” provides business logic for manipulating data, and “Template” acts as an interface (GUI) that facilitates the user and system interaction. MVT architecture is provided by Django which is scalable and reusable

# ARCHITECTURE DIAGRAM

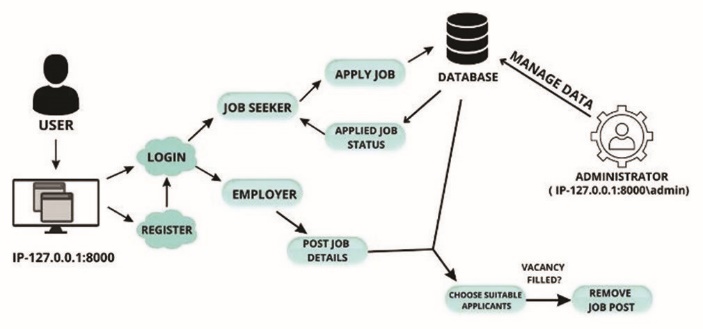


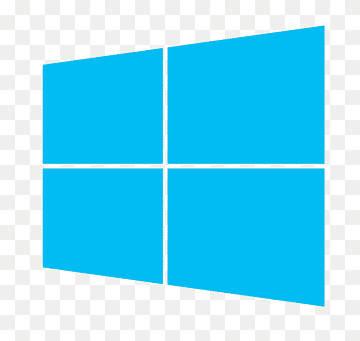
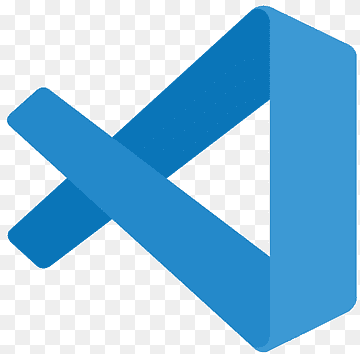
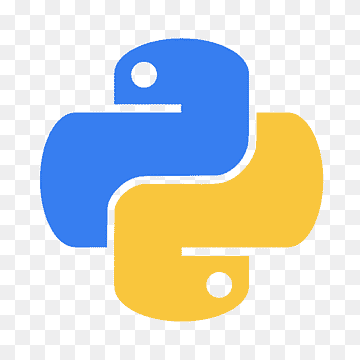
Figure 1: Architecture Diagram

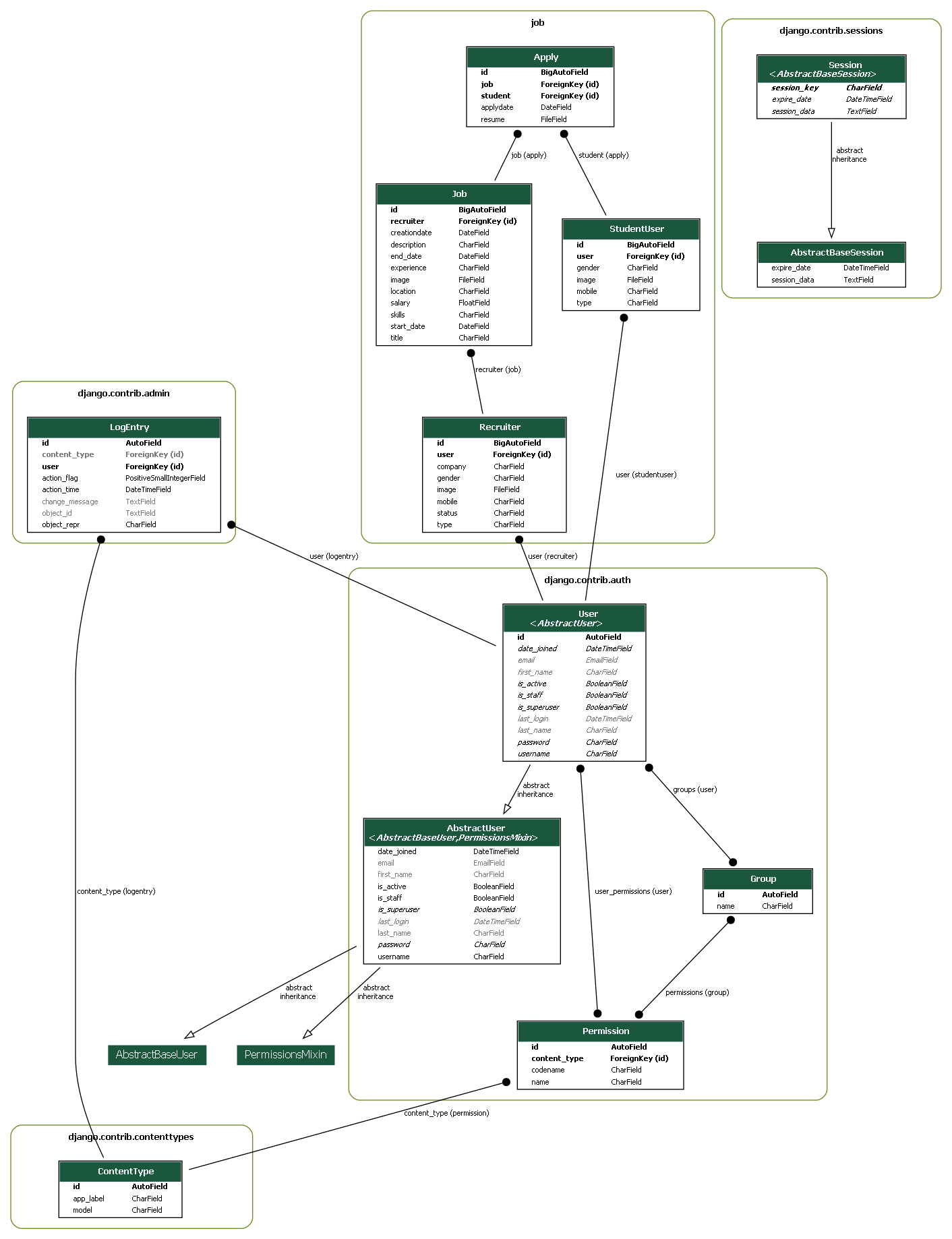
Explanation:

The user logs in to the main page of the web application. A registered user can log in to his/her profile through their login credentials. If not, then he needs to register first and then log in. The user of this application is either a job seeker or an employer. The roles of the job seeker are applying for the job, viewing the status of the applied job as well seek guidance from the available video resources. The role of the employer is to post available jobs and access the information from the database that contains the information about the job seeker and choose the eligible applicants. Once the employer has chosen the employee then he can change the status of the job from “vacant” to “filled” and remove the job post. The administrator accesses his dashboard and be able to view and manage the details of both the job seeker and employer from the database.

## Technology Used

1. For the backend there used **python , Django framework**.
2. For the database there used **sqlite3.**
3. For frontend and styling there used **HTML,CSS,Bootstrap5**.
4. The IDE I used is **Visual Studio Code.**
5. This project is made in **Windows10 Operating System.**





# ER DIAGRAM

Figure 2: ER diagram

# Software Designing Architecture: MVT

The model acts as an interface for the data. Its responsibility is to maintain the data. It makes up the logical data structure of the entire application and is represented by a database. The view is your web browser’s user interface a website’s rendering produces the view. It is represented by HTML/CSS/JavaScript. The static part of the desired HTML output is the Template. However, it has some special syntax that defines how to insert dynamic content.

Modules

i. Login

ii. Registration

iii. Recruiter

iv. Jobseeker

v. Administrator

# LOGIN

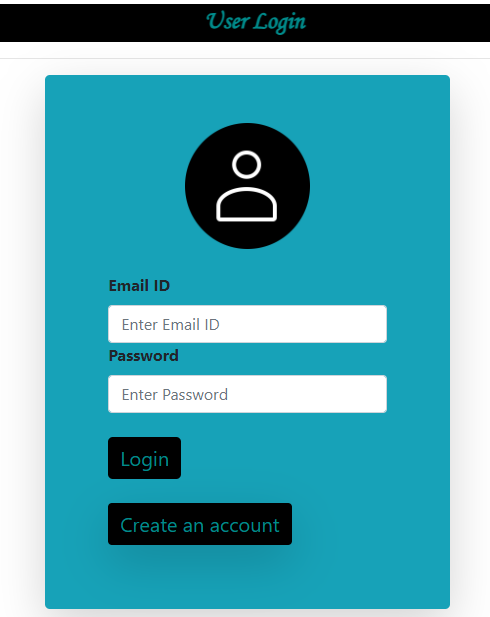
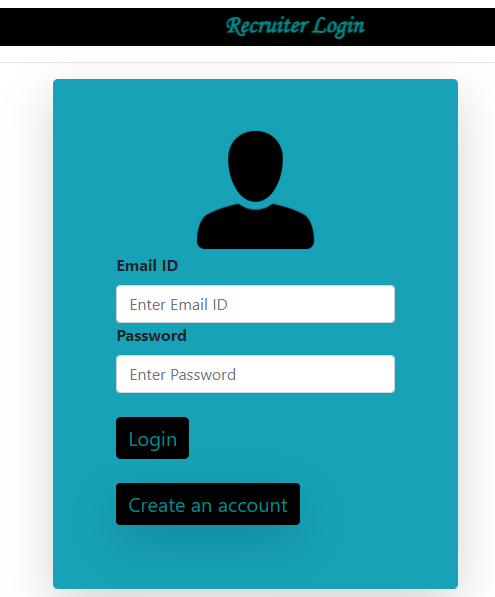
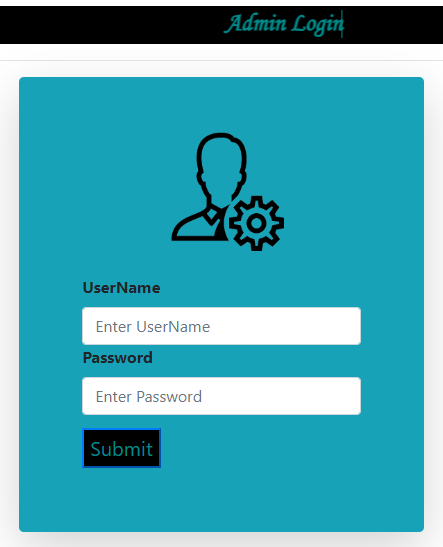
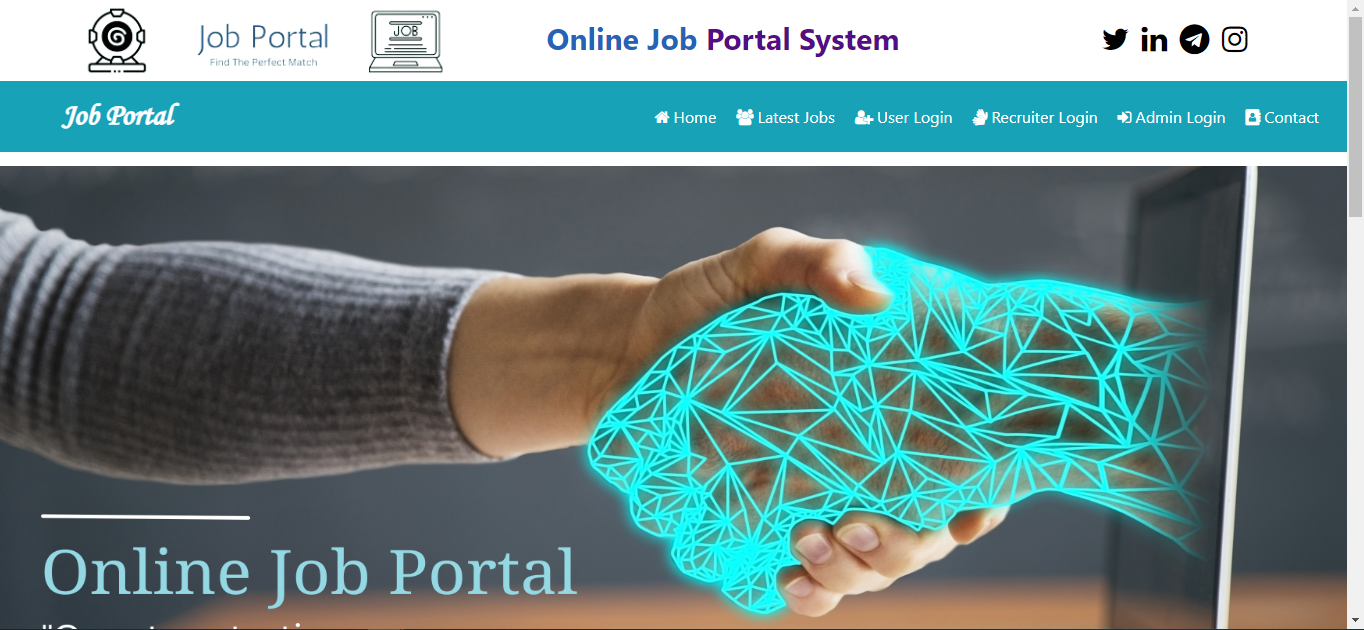
The login module gets the username and password of the user. If the user is already a registered user, then it leads to the user’s dashboard. In the case of a new user, he/she needs to sign up by giving his/her details

Figure 3: Admin Login

Figure 4: Recruiter Login

Figure 5: User Login



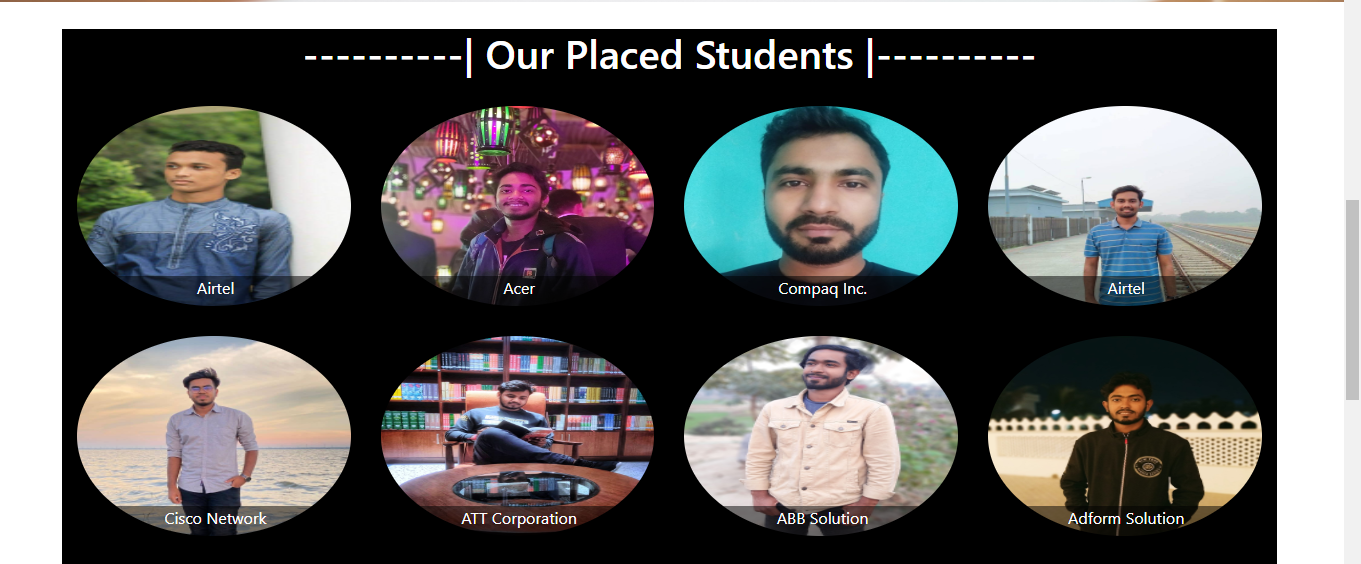
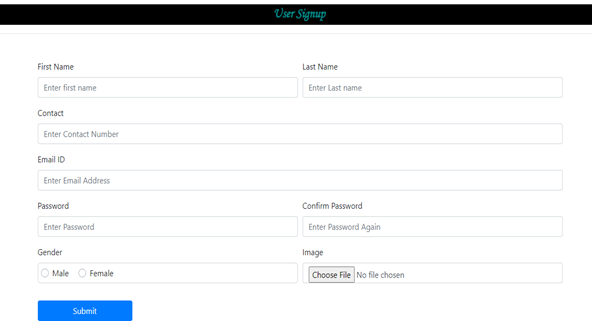


Figure 6: Home page

REGISTRATION

This module gets the required details of the user (name, mail, password, etc.,) to register the account to make the user an authenticated one to access the web application.

Figure 7: Registration page for User

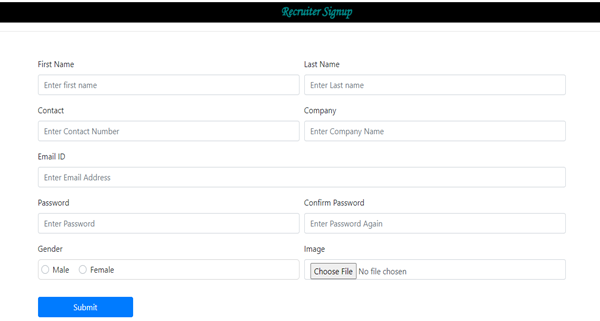


Figure 8: Registration page for Recruiter

# Recruiter

1. The recruiter first logs into his/her account.

2. He/she can post the details of the job such as skill requirements, specifications, working hours, salary details, and mode of job (online or offline).

3. They choose the suitable applicants till the vacancies are full and remove the posted job details.

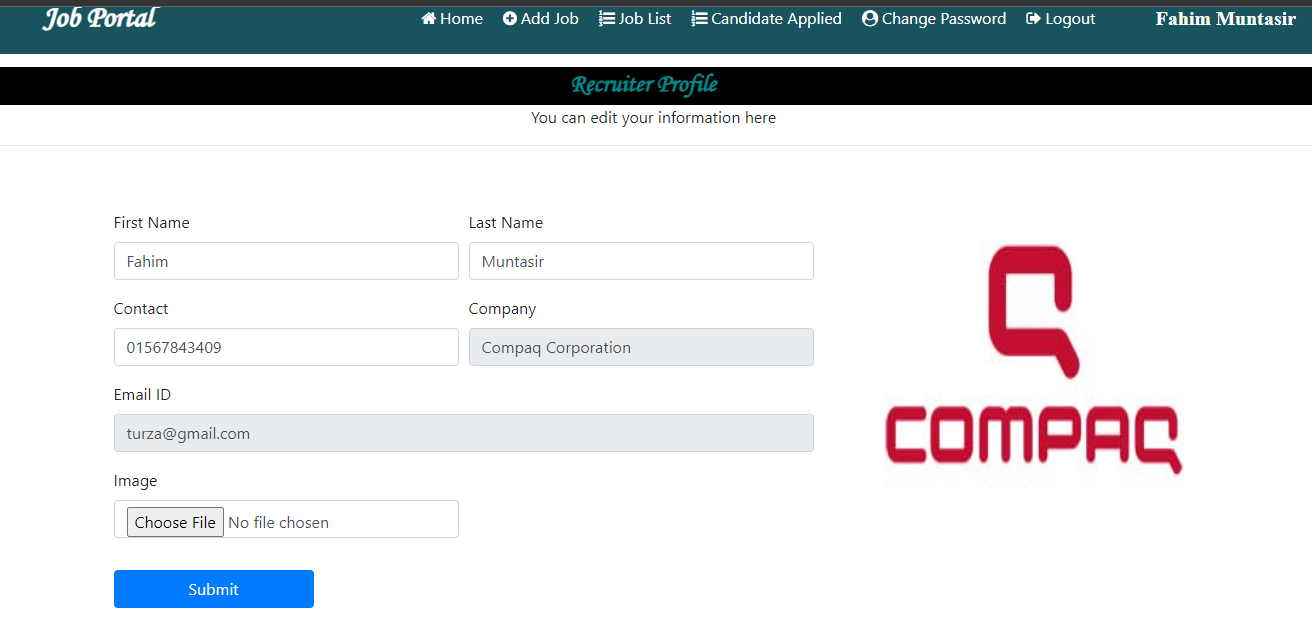


Figure 9: Recruiter Dashboard

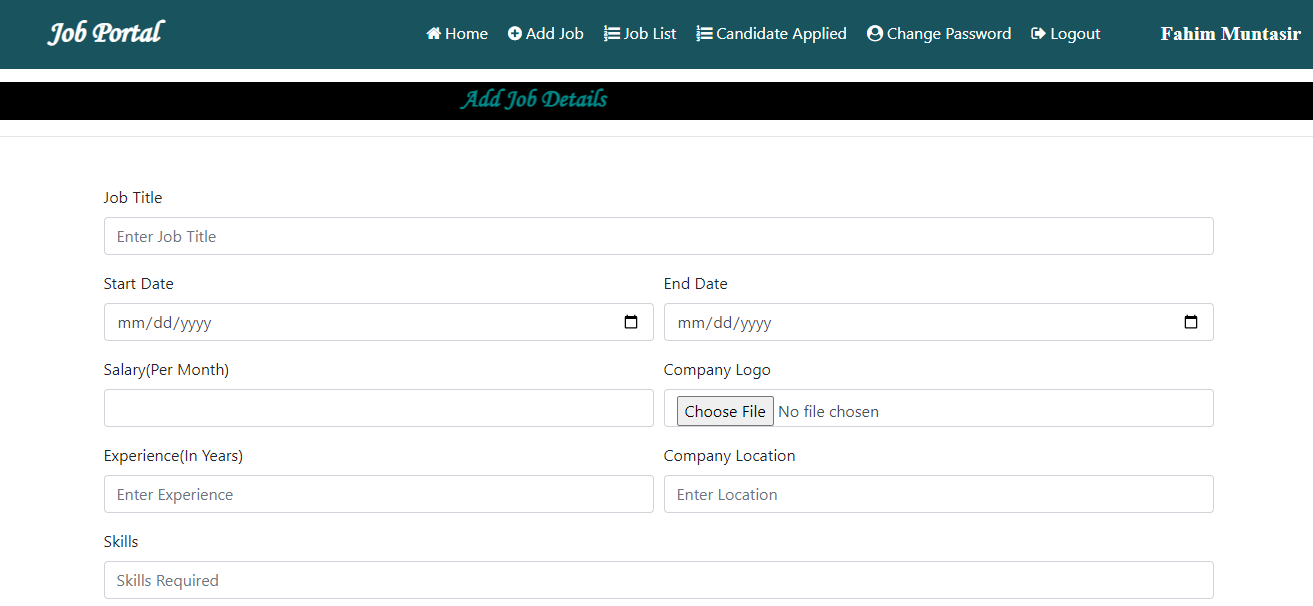


Figure 10: Add Job post



Figure 11: View applied candidates

# ADMINISTRATOR

1. The administrator collects the information and maintains the databases containing the details of the employer and the job seeker.

2. The administrator can view the data, manage them, and alter the data in correspondence with the varying demands of the employer and job seeker.

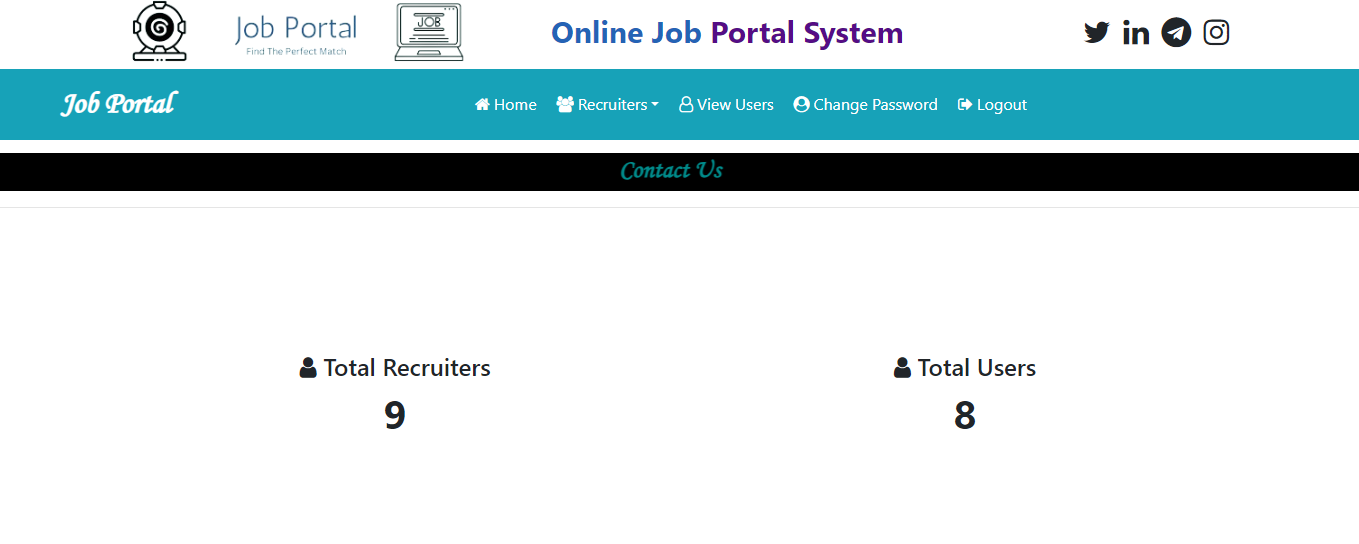


Figure 12: Admin Dashboard

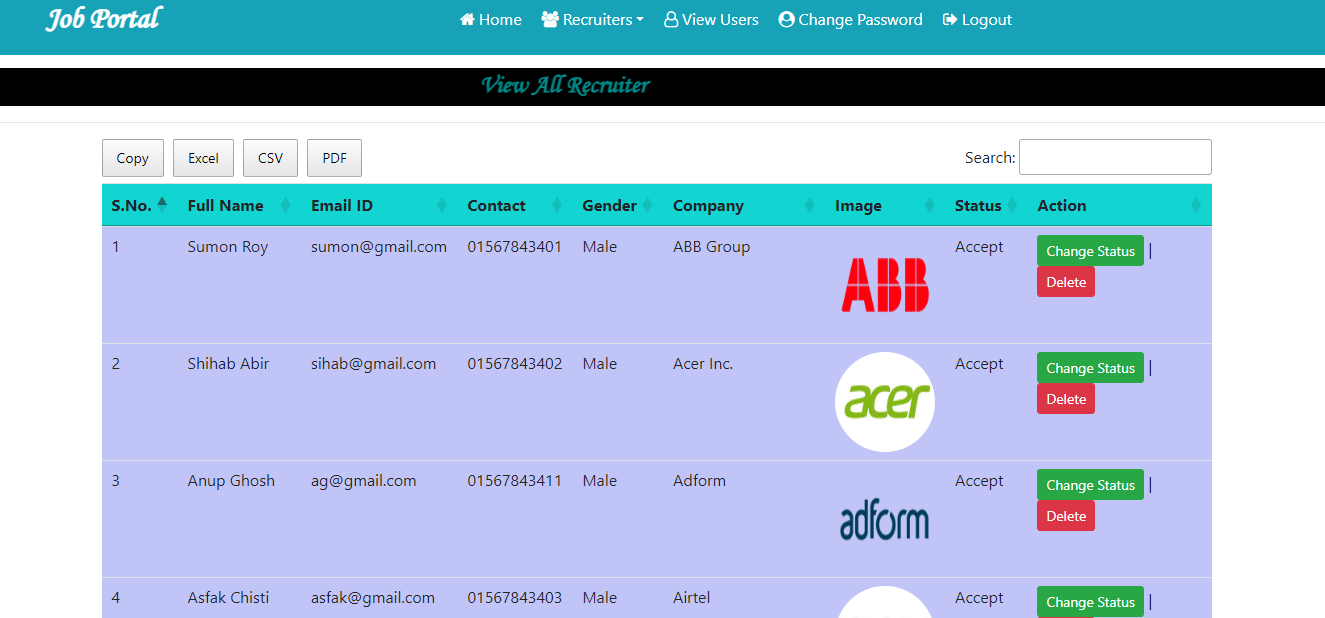


Figure 13: Admin manages recruiters

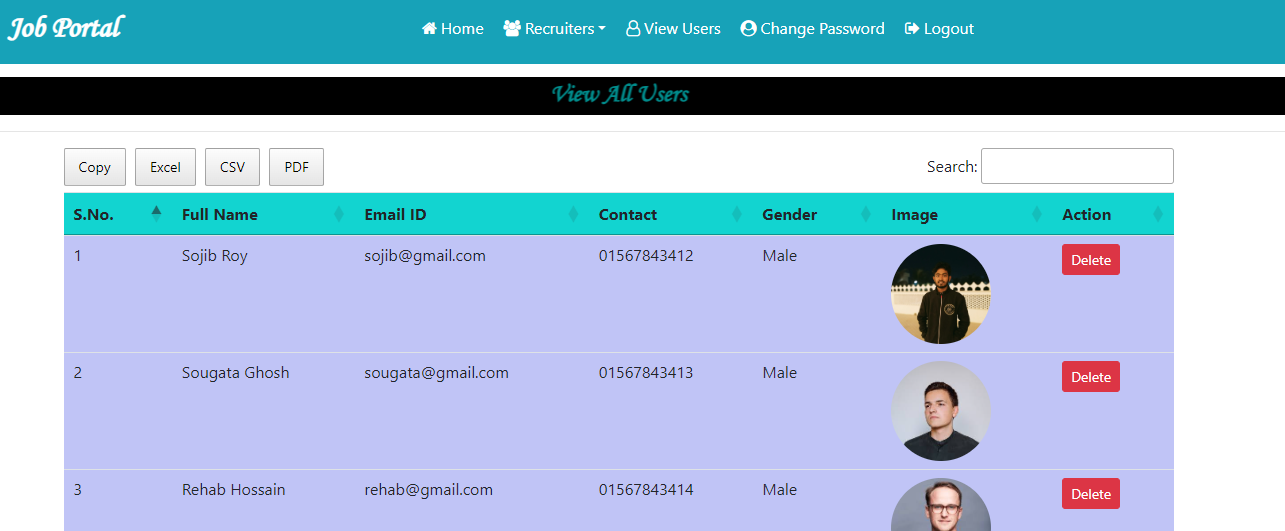


Figure 14: Admin manages users

# JOB SEEKER

1. The job seeker enters the home page using the login credentials.

2. He/she searches through the list of available job openings.

3. They can apply for a job that fulfills the requirements and also check the status of the applied job.

4. They can make CV from this portal.

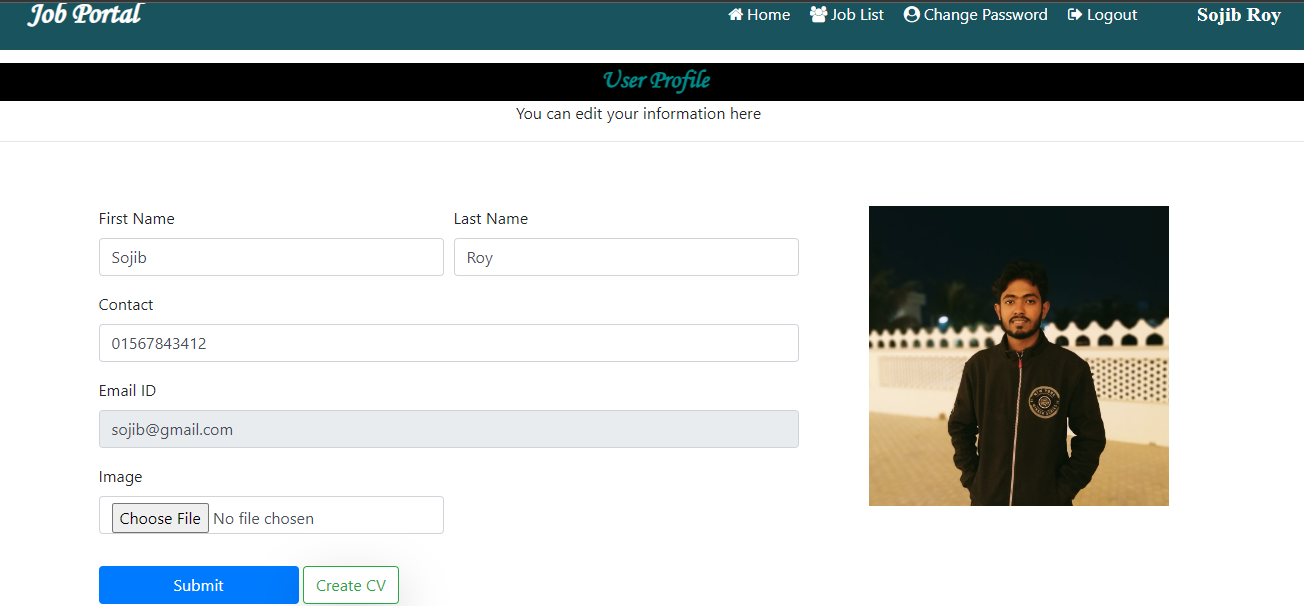


Figure 15: Job seekers dashboard

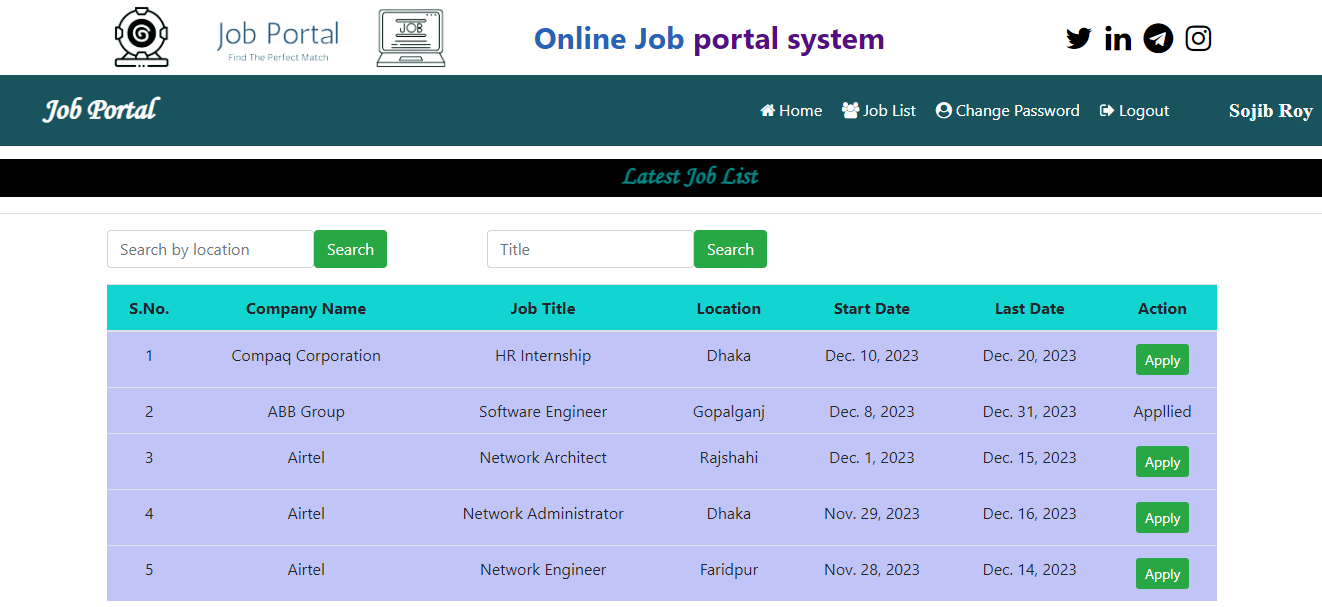


Figure 16: Job list shown by the job seeker

## CV generator

Job seekers can make CV by filling up this form.

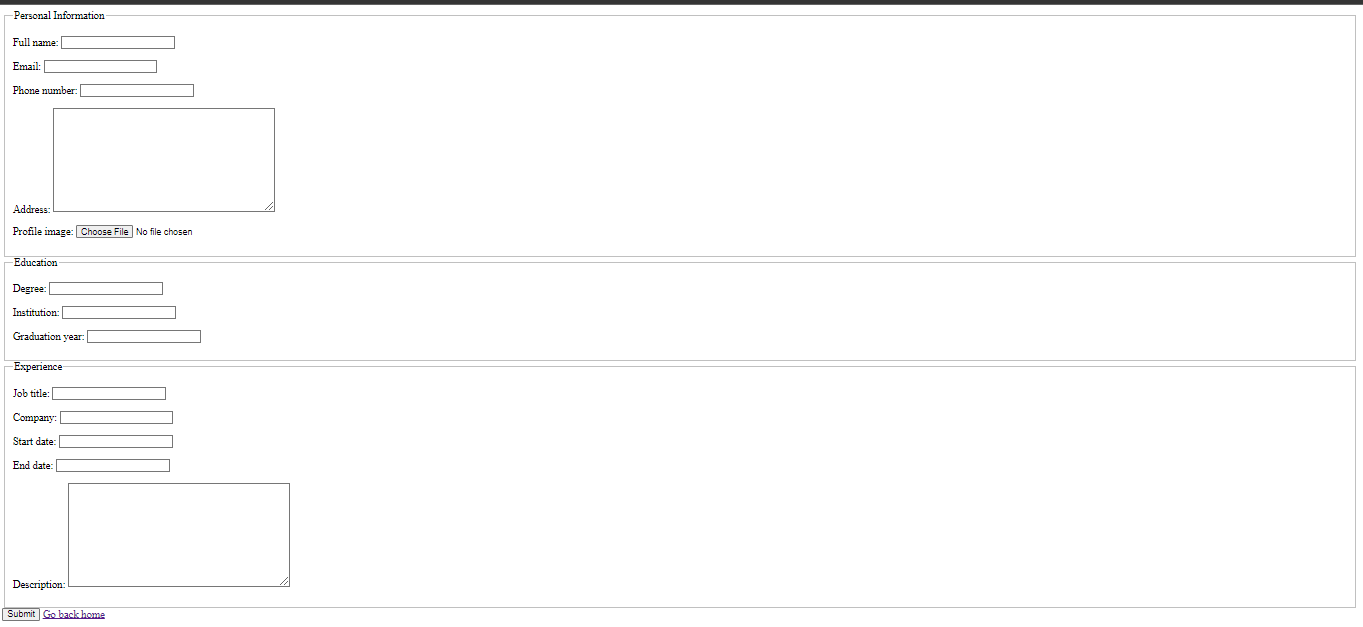


Figure 17: CV generator form

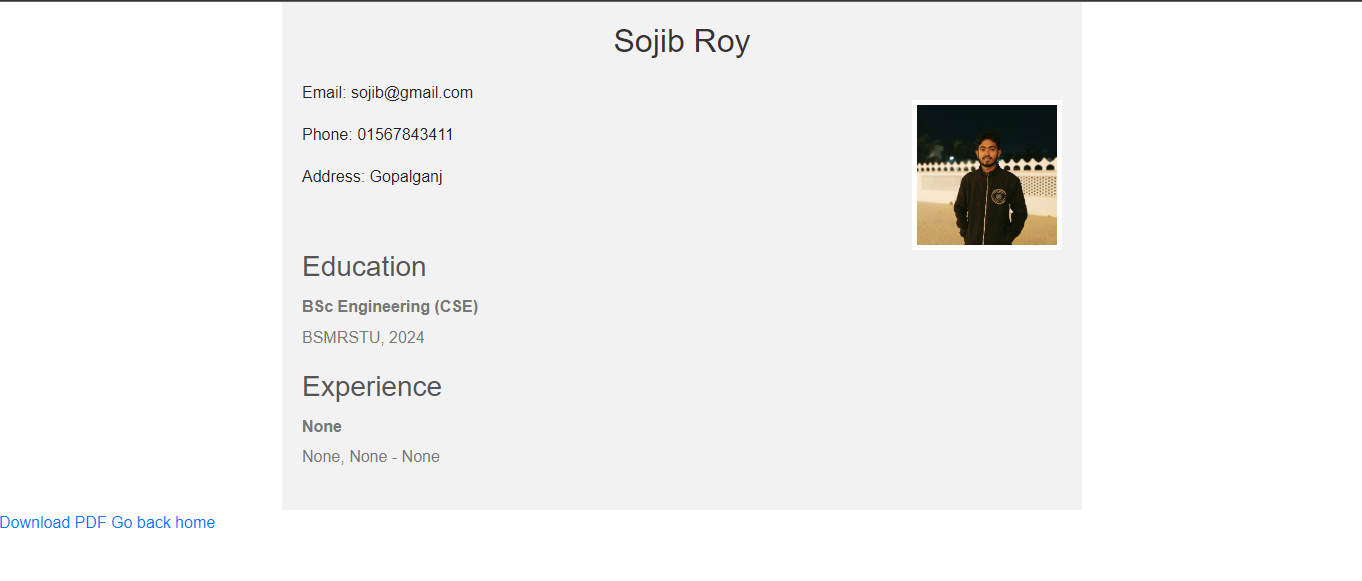


Figure 18: Output of the CV

## Future Enhancement

The future implementation will be online help for the users and chatting with website administrator. Add job preparation section.

## CONCLUSION

To sum it up, the online job portal we talked about makes finding jobs and hiring people much easier. It's like a super user-friendly website where job seekers and employers can easily connect. With lots of jobs and smart technology, it's set to change the way people find work and companies find employees. It's not just a website; it's a cool way to make the job market work better for everyone.

## REFERENCES

Bootstrap *Containers · Bootstrap v5.3*. Bootstrap. Retrieved December 16, 2023, from https://getbootstrap.com/docs/5.3/layout/containers/

Django Software Foundation *Django*. Django documentation | Django documentation. Retrieved December 16, 2023, from https://docs.djangoproject.com/en/5.0/

Materialize *Getting Started*. Materialize. Retrieved December 16, 2023, from https://materializecss.com/getting-started.html