

iProfiler

Project report submitted in the partial fulfillment of the requirement for the
degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Isha Gupta (171358)

Under the supervision of

Dr. Kapil Sharma and Mr. Chandra Kalva

to



Department of Computer Science and Engineering and Information
Technology

**Jaypee University of Information and Technology Wahnaghat, Solan-
173234, Himachal Pradesh**

CERTIFICATE

Candidate's Declaration

We hereby declare that the work presented in this report entitled “**iProfiler**” in partial fulfillments of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science and Engineering and Information Technology, Jaypee University of Information and Technology Waknaghat is an authentic record of my own work carried out over a period from March 2021 to April 2021 under the supervision of **Dr. Kapil Sharma (Associate Professor, CSE Department, Jaypee University of Information Technology, Waknaghat)** and **Mr. Chandra Kalva (Senior Engineering Management Specialist, Hashedin by Deloitte)**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Isha Gupta

171358

This is to certify that the above statement made by the candidates is true to the best of my knowledge.



Dr. Kapil Sharma

Associate Professor

Department of CS & IT

Dated:

Chandra Kalva

Senior Engineering Management Specialist

Hashedin by Deloitte

Dated:

PROJECT REPORT UNDERTAKING

I Ms. Isha Gupta, Roll No. 171358, Branch CSE is doing my internship with Hashedin by Deloitte from February 2021 to August 2021.

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/ developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature- Isha Gupta

Name – Isha Gupta

Roll No. - 171358

Date – 23/05/2021

Isha Gupta

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

We are highly indebted to Dr. Kapil Sharma and Mr. Chandra Kalva for their guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

We would like to express our gratitude towards our respective family members for their kind co-operation and encouragement which help me in completion of this project.

Our thanks and appreciation also goes to our friends in developing the project and people who have willingly helped us out with their abilities.

Table of Contents

CHAPTER-1	1
1.1 GENERAL INTRODUCTION	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVES	4
1.4 METHODOLOGY	5
1.5 ORGANIZATION	7
CHAPTER-2 LITERATURE SURVEY	8
2.1 Research Methodology	8
2.2 Literature Collection and Segregation.....	12
2.3 Critical Review of Literature.....	15
CHAPTER-3 SYSTEM DEVELOPMENT.....	17
3.1 REQUIREMENT ANALYSIS	17
3.1.1 Purpose	17
3.1.2 Intended Audience and Reading Suggestions	17
3.1.3 Project Scope	17
3.1.4 Functional Requirements	17
3.1.5 Non-Functional Requirements.....	19
3.1.6 Design Constraints.....	20
3.2 ARCHITECTURAL DESIGN	21
3.2.1 Conceptual Design.....	21
3.2.2 Flow Chart	24
3.3 WORKING.....	26
3.4 BENEFITS	27
3.5 TOOLS AND TECHNOLOGIES	28
3.5.1 Front End	28

3.5.2 Back End.....	29
3.5.3 Integrations	31
3.5.4 Cloud and Deployment.....	31
3.6 MODULES.....	33
3.6.1 Screen 1: Recruiter Login/Signup	33
3.6.2 Screen 2: Landing Page	33
3.6.3 Screen 3: Job Details	33
3.6.4 Screen 4: Apply Page	33
3.6.5 Screen 5: Pricing.....	34
3.6.6 Screen 6: Dashboard.....	34
3.6.7 Screen 7: Job Postings	34
3.6.8 Screen 8: Applicants.....	35
3.6.9 Screen 9: User Profile.....	35
CHAPTER-4 RESULTS AND ANALYSIS	36
4.1 SCREENS	36
4.2 COMPARITIVE STUDY AND DISCUSSIONS.....	49
CHAPTER-5 CONCLUSION	50
5.1 SYSTEM CONCLUSION	50
5.2 FUTURE SCOPE.....	50
REFERENCES	51

LIST OF ABBREVIATIONS

- BI - Business Intelligence
- JSON - JavaScript Object Notation
- API - Application Programming Interface
- CRUD - Create, Read, Update, Delete
- REST - Representational State Transfer
- MVP - Minimum Viable Product
- CI/CD - Continuous Integration, Continuous Delivery/
- ATS - Applicant Tracking System
- SDK - Software Development Kit
- VCS - Version Control System
- XML - Extensible Markup Language
- MVC - Model view Controller

LIST OF FIGURES

Figure 1 Stages of Agile Development	5
Figure 2 Architectural Diagram for iProfiler	21
Figure 3 Flow Chart for iProfiler	24
Figure 4 Landing Page	36
Figure 5 Login Screen.....	37
Figure 6 SignUp Screen	37
Figure 7 Dashboard Screen	38
Figure 8 Current Status of Applicants	38
Figure 9 Current Job Postings.....	39
Figure 10 Add a Job Posting.....	39
Figure 11 Edit a Job posting	40
Figure 12 Send Invitation to a Candidate	40
Figure 13 Company Page.....	41
Figure 14 Available jobs in an Organization	41
Figure 15 Resume Upload	42
Figure 16 Prefilled Form after Resume Upload.....	42
Figure 17 Successful Submission	43
Figure 18 Invitation to Candidates for further rounds	43
Figure 19 Filter Form.....	44
Figure 20 Result of Applied Filters	44
Figure 21 View Profile and Download Resume	45
Figure 22 Downloaded Resume.....	46
Figure 23 Search Candidates by Name	46
Figure 24 Analytics Page	47
Figure 25 Contact Us Page	48

ABSTRACT

Resume filtering and shortlisting the candidates from so many candidates is a very tedious task for the HRs and the recruiting teams. The recruiting teams of different organizations find it difficult to manually filter the resumes of the candidates who they can further call for interviews as per the requirement of their organization.

iProfiler is a profile filtering tool that helps selecting out relevant prospects from a pool of generic applicants. It makes the life of HR personnel and the recruitment teams easier by removing the hassle out of remote recruitment process. It is a very tedious process to manually screen all the resumes and filtering out potential candidates from pool of candidates. Also, it is a time-consuming and counter-productive work of manually screening resumes. In order to solve all these problems, iProfiler comes in rescue. iProfiler is developed using React for the frontend, Flask for the backend and MongoDB is used for the database. It provides various features that are relevant for an organization during recruitments, because, nowadays thousands of applicants apply for a single job posting. iProfiler provides features like gallery view of all the candidates in the dashboard, filtering the candidates based on skills, experience and past designation, search functionality to search the candidates by their names, email service to contact the candidates for further rounds of the interviews, view profile option to view the complete details of a candidate and download resume options if one is interested in the candidate profile. It also provides a section to post a job and send the invitation mail with the apply form link to different candidates to apply for that job role. iProfiler is a tool which suffices all the needs of the recruitment teams and the HR and make their life hassle free.

CHAPTER-1

1.1 GENERAL INTRODUCTION

iProfiler is a profile filtering tool that helps selecting out relevant prospects from a pool of generic applicants. It makes the life of HR personnel and the recruitment teams easier by removing the hassle out of remote recruitment process. It is very tedious process to manually screen all the resumes and filtering out potential candidates from pool of candidates. Also, it is a time consuming and counter-productive work of manually screening resumes.

In order to solve all these problems, iProfiler comes in rescue. iProfiler is developed using React for the frontend, Flask for the backend and MongoDB is used for the database. It provides various features that are relevant for an organization during recruitments, because, nowadays thousands of applicants apply for a single job posting.

iProfiler provides features like gallery view of all the candidates in the dashboard, filtering the candidates based on skills, experience and past designation, search functionality to search the candidates by their names, email service to contact the candidates for further rounds of the interviews, view profile option to view the complete details of a candidate and download resume options if one is interested in the candidate profile. It also provides a section to post a job and send the invitation mail with the apply form link to different candidates to apply for that job role. iProfiler is a tool which suffice all the needs of the recruitment teams and the HR and make their life hassle free.

1.2 PROBLEM STATEMENT

The amount of work opportunities is inversely proportional to the existing workforce as population and technologies grow at an exponential rate. As a result, finding a career becomes more difficult.

There are many applicants for a single career posting. Filtering out potential recruits from a wide pool of applicants becomes very challenging from a company's perspective.

When a large number of candidates apply for a single work posting, the Human Resources Team/Recruiter must go over and screen each resume. They must carefully screen each resume in order to find applicants who are well prepared with the necessary skill sets.

When we look at this on a bigger scale, it takes a lot of time, and there is a risk of losing out on promising applicants in the rush to screen various resumes. In addition, ineffective job hiring can be detrimental to any organization at any stage at any time.

Recruiting candidates that suit a particular job description is a big challenge for most companies. As online recruiting becomes more commonplace, traditional recruiting practices are becoming inefficient. Traditional methods usually include a time-consuming process of manually sorting through all the applicants who have applied, reviewing their resumes, and then creating a shortlist of appropriate candidates to interview.

We can understand the problem with the help of following scenarios: -

- **Scenario 1:** - Evaluating each incoming application is tedious and counterproductive.
- **Scenario 2:** - Recruiting the best candidates for the requirement of a job role can be cumbersome.
- **Scenario 3:** - Difficulty in keeping track of promising applications which weren't processed earlier.

Now let us understand these scenarios with detailed explanations: -

Evaluating each incoming application is tedious and counterproductive. As we all know, manually screening resumes is a time-consuming and inefficient job. It becomes a difficult job, particularly when there are a large number of resumes to go through. More critically, it represents a waste of human resources, time, and capital. Time expended on this task could easily be expended in another meaningful task.

Recruiting the best candidates for the requirement of a job role can be cumbersome. In general, skill sets differ from one career position to the next. To recruit anyone who meets the required skill sets, HR must manually go through a candidate's skills and map them to the desired skill sets for any particular work posting/role. This becomes a source of frustration for the HR professional or its team. Difficulty in keeping track of promising applications which weren't processed earlier.

There may be times where there are a small number of vacancies for a certain job position, but there are only a few applicants who suit the job role. Since only one person can get the position, all qualified applicants are brushed aside and do not have a decent chance to succeed.

Many firms have multiple guidelines and opinions about whether to streamline the resume review process or stick with the traditional method. There may be times where a tool filters and selects a small number of candidates based on their qualifications and skill sets, but the selected candidate may not be capable of doing the job. Alternatively, this could disqualify a few candidates based on their academic credentials, even though they possess the necessary skill sets for the position they have applied for.

Resume-to-job matching is one of the most important stages of the recruitment process. They spend thousands of hours a day at large companies like Tencent, which is tedious and passive, searching for new candidates via internal and external recruitment channels. Furthermore, traditional job descriptions find it difficult to cover a broad variety of recruitment circumstances for increasingly complex and specialized requirements.

1.3 OBJECTIVES

The main objective of this project is to ease the process of resume shortlisting for the HRs and the recruiting teams. It aims to provide efficient filter and search functionalities based on name, skill set of the candidate, experience and job role from a large bunch of resumes. It aims to provide a dashboard after successful login/signup to the recruiters of different companies. The goal of this application is to provide a complete flow of recruitment to the HRs or the recruiters of different companies. In general, recruitments are done when a job role is posted by a company and applicants apply for that role. To achieve this, iProfiler aims to provide a job role section in the dashboard, where a job role can be added, changed and deleted as per the requirement of the organization. For different job roles, one can send the email invites to the applicants to apply for that job role. The objective is that, the email should contain a link which redirects to the form that needs to be submitted for successful application. Once it is done by the applicant side, the HR can view all the candidates in the gallery view of the dashboard. The gallery view aims to provide status of the applicant i.e., whether that candidate is shortlisted for further rounds or not. It aims to provide further rounds information that can be sent by HR through the provided email service. Along with that, the goal is to let HR view individual profiles of the candidates with complete details and download their resume for later reference. The application also aims to provide an analytics section in the dashboard where HR can view the complete statistics of all the candidate through different graphs. Overall, the application aims to provide a great user experience for the recruitments with the features.

1.4 METHODOLOGY

The project is based on the agile methodology for development. Agile model is an incremental and iterative model of development which means that each build of the software would be independent, but the final delivered product will be incremented with that build. In this model, at every iteration different teams of planning, requirement analysis, design, coding, unit testing and acceptance testing works simultaneously. Figure 2 defines the various stages of the Agile Development.

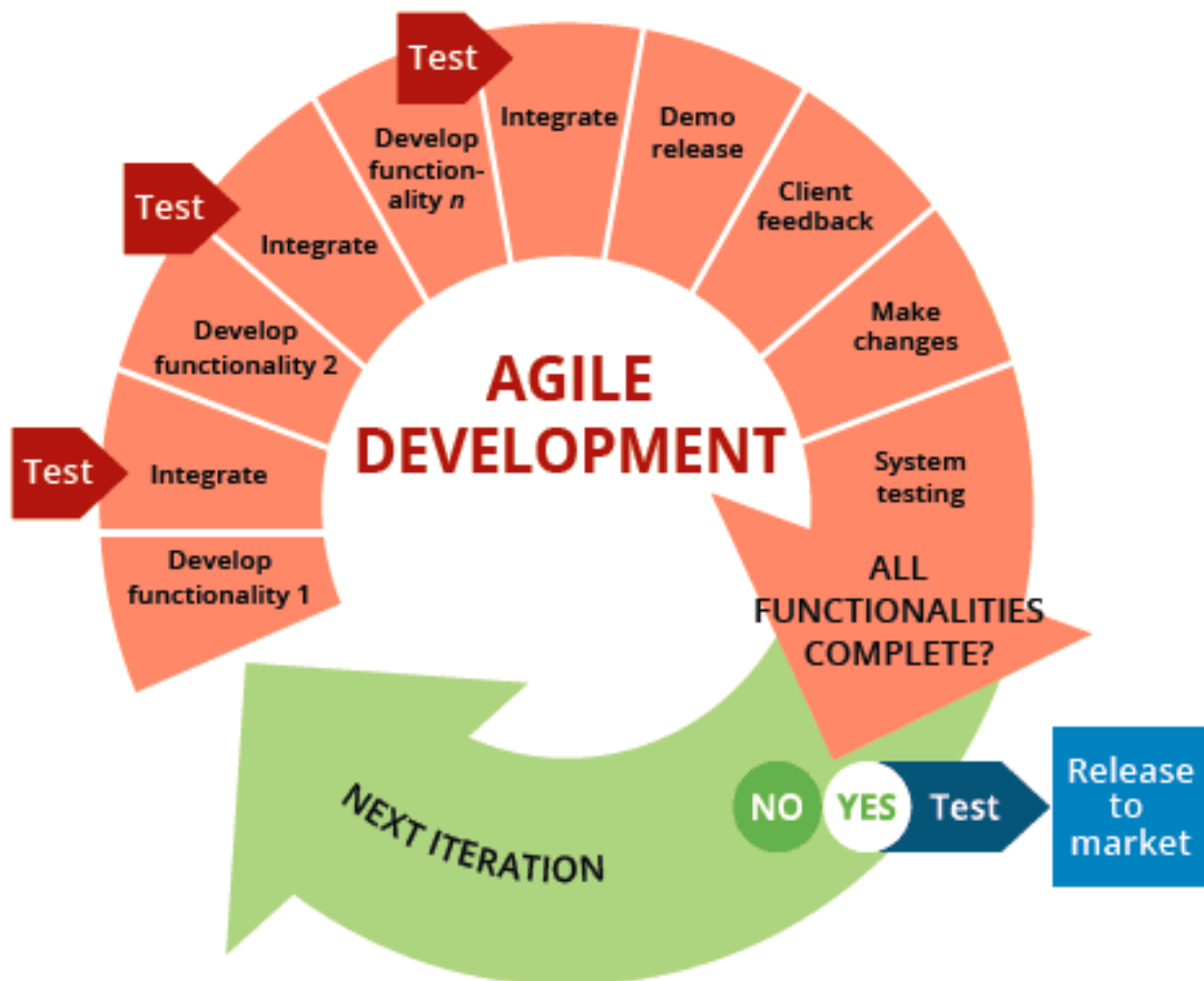


Figure 1 Stages of Agile Development

The motive of developing our project based on this model is that iProfiler Application requires frequent updates due to updates in the features and its new behaviors. Using this model will lead to develop updates and their testing independently and then integrating them with the existing product to test the final product. Along with that it is comparatively easy in agile development than in waterfall approach to fix the bugs or to update the system as per the customer's requirement.

1.5 ORGANIZATION

The Project Report follows the detailed analysis of how the project work is completed. The complete report is organized into the following chapters:

CHAPTER 1:- Chapter 1 of this report gives a brief introduction about what is iProfiler Application and why we need to use it. The chapter describes about the use of Application and the benefit it gives to the users.

CHAPTER 2:-This chapter gives details about the work done previously in this field. It briefs about the technologies and algorithms already used in this field with their significance.

CHAPTER 3:-This chapter gives the details of the process followed to develop the system. It contains the conceptual models of the system and algorithms used in the system.

CHAPTER 4: -In this chapter the result calculated by the proposed algorithm is presented and described. Performance is analyzed with the result.

CHAPTER 5:-The conclusion based on results and performance of the system is made in this chapter. This chapter also defines the future scope of this project. At last the report contains the references that are used in the project.

CHAPTER-2 LITERATURE SURVEY

To create something meaningful for the HR community, we should do analysis around the issues in order to get a 360-degree perspective on them. To address an actual problem, we must conduct rigorous research on the problem and its existing solutions. Also, where do these solutions lag/create problems in some particular use case.

This chapter can be break down into two important parts, which are listed down below:

- Research Methodology
- Literature collection and segregation
- Critical review of selected literature

[1] A CV (Curriculum Vitae) or Resume is a document that is used to create an initial im-pression on a potential employer. It displays important information about an applicant, such as history, qualifications, and schooling, so that a prospective employer can easily see how your past experiences can contribute to a company’s success. A well-written resume shows an applicant’s suitability right away and gets them an interview.

2.1 Research Methodology

[2] Any project must have a well-defined schedule in order to be finished on time. As the saying goes, if you don’t prepare, you plan to fail. There are several methodologies that we might use in the project. When we follow any approach, the difficulty associated with development becomes very straightforward and it allows one to follow in a systematic flow, which eliminates all forms of blockers that we may experience during development.

When we are confronted with a dilemma, we have an inner desire to fix the problem that we and many others face on a regular basis. Most of us can recognize the problem, but it takes an extra step to consider the problems and how they can be handled effectively. As the saying goes, “necessity is the mother of invention”.

The problem statement for this project is that manual resume screening is ineffective and time consuming for any HR professional or recruitment team. So, when we first came across this problem statement, we had to do some research into the potential issues that could arise during the recruiting process. We must identify the issues by putting ourselves in the shoes of the recruiter in order to better appreciate the issues they face.

The software development methodology that we were following in our project is Agile Methodology. Agile is an iterative approach to project management and software creation that enables teams to bring value to clients quicker and with less headaches. Instead of relying solely on a "big boom" launch, an agile team delivers work in tiny, but consumable, increments.

Continuous evaluation of requirements, schedules, and outcomes provides teams with a natural framework for adapting to change rapidly. We choose agile so that we can react rapidly to shifts in the markets or consumer reviews without jeopardizing a year's worth of plans. "Just enough" preparation and delivering in tiny, regular intervals allows us to gain input on each improvement and incorporate it into future plans at a low cost.

Since we were using agile methodology, we divided our work into sprints. Each sprint consisted of five working days a week. We had a scrum meeting with our Tech Lead every day. During scrum sessions, we brainstormed about different problems that HR could face in their professional lives. All was required to present their points, resulting in a large dispersion of problems and, as a result, a wide distribution of coverage of problems.

After everyone pitched their ideas, we then collectively jot down major problems that we can work upon. Also, we did research to identify problems that still exist in tools which is similar to ours.

The problems with the traditional recruitment process are listed as follows: -

- Evaluating each incoming application is tedious and counterproductive.
- Recruiting the best candidates for the requirement of a job role can be cumbersome.
- Difficulty in keeping track of promising applications which weren't processed earlier.

The above problems have been discussed below in detail:-

Evaluating each incoming application is tedious and counterproductive:-

[3] Manually scanning resumes is, as we all know, a time-consuming and cumbersome job. That becomes a challenging task, especially when there are a massive proportion of resumes to review. More importantly, it is a loss of precious energy, time, and money. The time spent on this assignment could easily be spent on some more meaningful task.

Recruiting the best candidates for the requirement of a job role can be cumbersome:-

[4] In general, skill sets vary from job to job. HR must personally go through a student's qualifications and correlate them to the desired skill sets for any given job posting/role in order to hire

someone who fits the requisite skill sets. This becomes a cause of annoyance for the HR professional or their staff.

Difficulty in keeping track of promising applications which weren't processed earlier:-

[4] There may be occasions where there are a limited number of openings for a certain job vacancy, but only a few candidates are qualified for the position. Since the vacancy can only be filled by one person, all eligible candidates are passed over and have no hope of success.

Different companies have different types of forms, which they use for their recruitment process. For some it's small, but for few it's too long. A student/applicant gets tired by manually filling each and every field of the registration / apply form.

In order to address this problem of students or applicants, iProfiler comes in rescue. It solves this problem by automatically filling all the fields in the registration form, by parsing the resume and mapping each value to its corresponding text inputs. Hence, a lot depends upon the structure of resume.

As we all know that for different students use different types of resume format / template. So in order, for this automatically prefilling form feature to work at higher accuracy, the resume which the candidate uploads needs to be ATS Compliant. We could also have asked the candidate to follow a single resume template. But we are not doing so because, we want our web app to cater wide audience and different spectrum of resumes.

Resumes are a great example of unstructured data. Each resume has its unique style of formatting, has its own data blocks, and has many forms of data formatting. This makes reading resumes hard, programmatically. Recruiters spend ample amount of time going through the resumes and selecting the ones that are a good fit for their jobs.

Tech giants like Google and Facebook receive thousands of resumes each day for various job positions and recruiters cannot go through each and every resume. This is why Resume Parsers are a great deal for people like them. Resume Parsers make it easy to select the perfect resume from the bunch of resumes received. For this we are using a library from PyPi i.e. pyresparser. It is a simple resume parser used for extracting information from resumes.

The following is the result which would be shown when a resume is parsed by our library:-

- Extract name

- Extract email
- Extract mobile numbers
- Extract skills
- Extract total experience
- Extract college name
- Extract degree
- Extract designation
- Extract company names

Supported File Formats: -

- PDF and DOCx files are supported on all Operating Systems
- If you want to extract DOC files you can install textract for your OS (Linux,MacOS)
- Note: You just have to install textract (and nothing else) and doc files will getparsed easily

2.2 Literature Collection and Segregation

[5] One of the most critical phases in the procurement process is resume-to-job matching. Every day, they spend thousands of hours looking for possible recruits via internal and external recruitment channels at large corporations like Ten cent, which is boring and passive. Furthermore, conventional job descriptions make it impossible to cover a wide range of recruiting situations for increasingly diverse and customized needs.

Recruiting applicants to meet a specific work description is a critical challenge for most businesses. Traditional recruiting practices are becoming inefficient as on-line recruitment grows in popularity. Traditional techniques typically include a time-consuming process of manually searching through the applied candidates, reviewing their resumes, and then producing a shortlist of suitable candidates to be interviewed.

Work hunting has been smarter and more available in this technological age. Companies generate a large volume of resumes/CVs that are not always arranged. There has been a lot of testing undertaken in preparation for the job hunt. Whereas, the process of selecting a candidate based on their resume has not been entirely automated.

Online recruiting sites are quickly transforming the world of job-market employment practices. On the Web, there are hundreds of millions of registered users with resumes and tens of millions of work listings. It is important to learn successful job-resume matching for recruiting services.

Existing work-resume matching research primarily focuses on learning strong representations of job descriptions and resume texts with robust matching systems. We believe that learning the preferences of both recruiters and work seekers from past interview histories will be beneficial, and that such preferences will help to increase job-resume matching.

[6] The main principle is to investigate the latent preference based on the history of all interviewed applicants for a career posting and the history of all work applications for a specific skill. New innovations are becoming highly important in the continuing competition for top talent. Demographics, alternative job structures, and multiple distance-busting innovations are driving seismic shifts in the population and workplace, rewriting the rules of engagement. Although digital platforms hasten the change, they also include fresh avenues for human resource departments to inspire, empower, and equip top talent.

From compensation and benefits to total rewards, from managing an employee population to managing a workforce ecosystem, from lifelong jobs to a career based on continuous learning, from annual performance reviews to continuous feedback, from

individual achievement to team collaboration, and from personnel administration to workforce engagement, employment practices are changing in every way. HR agencies that are slipping behind should take solace in the fact that the technology revolution is only in its early stages, allowing them to keep up. Although 87 percent of respondents agree that digital would radically transform HR, 75 percent admit that their IT platforms and technologies are yet to reach optimum success or the desired market outcomes. However, the window of opportunity could be shrinking, as 57 percent of respondents expect to raise their IT budgets by 1% to 10% over the next two years, and 25% plan to increase budgets by more than 10%.

[7] Algorithmic decision-making is becoming more popular as a new source of advice in human resource recruitment and development. Although companies use algorithmic decision-making to save money and maximize productivity and objectivity, it can also result in unequal representation of certain classes of individuals, unconscious prejudice, and perceived unfairness. In the field of human resource administration, current awareness about the risks of unfairness and (implicit) sexism posed by algorithmic decision-making remains largely unexplored.

Algorithmic decision-making is characterized as automatic decision-making and remote control, as well as standardization of systematized workplace decisions. Algorithms, instead of people, make choices, and this has significant individual and social consequences in corporate optimization. These shifts in favor of algorithmic decision-making make it possible to find unknown qualified workers in companies and immediately check a vast number of applications.

The primary motivators for algorithmic decision-making are cost and time savings, risk reduction, increased efficiency, and decision-making certainty. Aside from these economic factors, companies aim to reduce human biases (e.g., preferences and personal beliefs) by implementing algorithmic decision-making, thus increasing the objectivity, accuracy, and equity of HR recruiting and production processes. However, relying exclusively on algorithmic decision-making can lead to inequality and inequity.

Discrimination is broadly described as unequal treatment of separate classes based on gender, age, or ethnicity rather than contextual differences such as individual performance. Algorithms that are trained on wrong, biased, or unrepresentative input data produce unequal or biased outcomes. As a

result, regardless of whether their input (or training) data is biased, algorithms are susceptible to replicating biased decisions.

To make matters worse, inequalities and sexism are often discovered after algorithms have decided. As a prominent example from the current debate on transparency, bias, and fairness in algorithmic decision-making, the hiring algorithms used by the American e-commerce specialist Amazon resulted in an extreme disadvantage of female applicants, which eventually led Amazon to shut down the entire algorithmic decision-making for the company.

Thus, possible problems associated with algorithmic decision-making include a lack of clarity and accountability of the input data, the algorithm itself, and the variables affecting algorithmic outcomes. Another concern is whether candidates and/or workers believe algorithmic decision-making is rational. Previous research found that candidates' and employees' approval of algorithmic decision-making in HR recruiting and growth is lower as compared to common human-led procedures.

[8] Every day, large corporations and headhunters obtain hundreds of resumes from work applicants. Extracting structured data automatically to support the automated creation of databases, searching, and resume routing, data from resumes of various styles and formats is required. In various applications, the description of resume knowledge fields varies.

Normally, resume data is represented as a two-layered hierarchical structure. The first layer is made up of general information blocks including Personal Information, Education, and so on. Then, for each general information block, detailed information pieces can be found, such as Name, Address, Email, and so on. For example, in the Personal Information block, detailed information such as Name, Address, Email, and so on can be extracted.

It's not easy to extract details from resumes with high precision and recall. Even though resumes are a restricted area, they can be written in a variety of formats (e.g. standardized tables or plain texts), languages (e.g. Chinese and English), and file formats (e.g. Text, PDF, Word etc.). Furthermore, writing styles can be extremely varied.

2.3 Critical Review of Literature

[9] One point is evident from all of the academic papers that have been referred to: there is a fundamental need to simplify the manual resume scanning procedure that has been used since ancient times. Various papers have proposed their own methodology/idea to address the current problems. Though their tactics vary, their end goal is the same. Finally, we want to reduce the counterproductive and time-consuming task of manually screening applications by HR professionals or the recruiting committee.

Many organizations have differing criteria and their opinion on the either to simplify this resume review procedure or adopt the conventional approach. There may be many occasions where, a tool may filter and choose few applicants based on their qualifications and skill sets they possessed, but often the chosen candidate may not be able to do the job. Alternatively, this may rule out a few applicants because of their academic qualifications, even though they have the requisite skill sets for the work they have applied for.

[10] Algorithmic decision-making is used by businesses to prevent or even transcend individual prejudices. However, our comprehensive analysis of the literature reveals that algorithmic decision-making is not a panacea for removing biases. Algorithms are susceptible to prejudices based on gender, race, sexual preference, or other attributes whether they are built on unreliable, biased, or unrepresentative feedback and training results . Algorithms reproduce biases if the input data were skewed in the first place. As a result, there is a need for transparency; staff and applicants should be able to appreciate what happens during the process.

Furthermore, by using algorithmic decision-making in HR recruiting and growth, organizations must understand the presumed fairness of workers and candidates. It is difficult for businesses to meet both computational justice from computer science, which is determined by rules and formulas, and perceived fairness from management literature, which is sensed subjectively by future and existing workers.

To ensure procedural and distributive justice, organizations must mitigate or eliminate both forms of biases and achieve subjective equity, such as human fairness, community fairness, and equal opportunity. Companies must constantly improve the perceived fairness of their HR recruiting and selection procedures, as well as their HR training and growth processes, to mitigate negative consequences for the organization, such as decreased employer attractiveness, employer image, job success, morale, and satisfaction with the processes.

In terms of fairness perceptions, it tends to be advantageous that humans make the ultimate call when it comes to employee potential or career growth. At first glance, this seems to contradict previous findings that automatic assessment appears to be more valid, since human raters may assess candidates inconsistently or without adequate proof. However, while people agree that an algorithmic machine should execute mechanical tasks (such as job scheduling), they believe that human tasks (such as hiring and work evaluation) should be done by humans.

[11] There are many reasons for the poor acceptance of algorithms in assessing individuals and their ability. The use of this emerging technology in HRM, along with a lack of awareness and clarity about how the algorithms function, raises emotional creepiness and reduces human care and social experiences, as well as honesty expectations and the ability to succeed. To mitigate the negative effects of algorithmic decision-making in HRM, businesses must encourage the use of algorithms to make the systems more accessible.

Online job search through popular websites is extremely useful, and it has long been a popular method for both job seekers and employers. Despite their importance in connecting employers with prospective hires, work search websites' search processes and technologies have not kept up with rapid advances in computing power and artificial intelligence. These websites' information retrieval strategies mainly rely on variations of manually entered search queries combined with advanced similarity metrics for ranking search results.

In conclusion, businesses do not depend entirely on the knowledge generated by algorithms, nor should they adopt automated decision-making without human oversight or auditing. Although certain biases may be more obvious, unconscious prejudice based on less obvious personal traits may be more troublesome since such implicit biases are more difficult to discern.

CHAPTER-3 SYSTEM DEVELOPMENT

3.1 REQUIREMENT ANALYSIS

3.1.1 Purpose

The purpose of this project is to build an application that provides ease in resume filtering to the HRs and the Recruitment Teams. The application aims to provide a personalized dashboard to the HRs where they can view different applicants for different job roles, filter them on the basis of skills, experience and previous designation, search them by their names, add, edit and delete different job roles, view analytics of the applied candidates and can also invite them to apply for different job roles and for further rounds.

3.1.2 Intended Audience and Reading Suggestions

The project mainly targets the recruiting teams and HRs of small to medium level organizations. It mainly targets those companies who have many applicants for a job role. Recruitment teams can segregate the applicants using the features provided by the application and then can invite them for the further rounds of interview.

3.1.3 Project Scope

The project mainly targets the audience who have small to medium number of recruitments. It offers different features to segregate the applicants who fits for a job role on the basis of their resumes and then can invite them for the further rounds of interview.

3.1.4 Functional Requirements

The major high-level requirements for the developed application are as follows-

- **Login and SignUp Authentication** - A user friendly responsive web pages are developed for the Login or Sign Up pages. If a HR from any organization is new to our platform, they can register themselves with us by filling the required fields in the SignUp Screen. If any user has already signed in with us, so they can go to Login Page and enter their user credentials and avail all the services we provide.
- **Gallery view of Applicants in Dashboard** - When a user Logs In, then the first screen that appears is the Dashboard Screen. There is a strong psychology behind the design of this page. We have crafted it in a Gallery View, so that a HR can have a glance at every card in a single shot. This saves a lot of time of HR by viewing each applicant details one by one.

- **Efficient multi-filtering and search functionalities** - In the dashboard screen, we have efficient multi-filtering and search functionalities. When there are 1000's of applicants whose applications have to be processed, it becomes a tough job for the HR to find any suitable person with required skill sets. In order to ease the life of HR, we have provided multi filter and search functionality in the HR Dashboard. HR can filter candidates based on the Skills, Experience and Job position which is been applied for. Also, a highly efficient search functionality is been provided to the user of our platform. The searching mechanism that is involved is as below. Normally search is done based on indexing, which is highly inefficient. So, what we have done here, based on search input, we try to find if sequence of characters is present anywhere, that words would be rendered as response to search request.
- **Applicant Profile View for every candidate** - Whose ever profile is appealing to the HR , they can get to know more about the candidate by clicking on the View Profile button, which navigates back on new screen where we can have comprehensive look about candidate's academic records.
- **Easy resume download service** - If the HR wants to know more about projects of any candidate, they can Download the resume by clicking on the download resume button that page.
- **Tags allocated for Available and Invited candidates** - In order to make the tracking purpose easier for the HR's, we have segregated the candidates based on their availability status. So there are 2 status tags we have allocated, they are Available Invited. Available tag is for those who are available for the recruitment process. Invited tags is for those, who have been invited by HR for the Job Role they have applied for.
- **Applicant communication assistance** - When a candidate has the desired skillsets and competency for a job position, then HR can use our application to communicate with the applicant using in built email service which we are providing, rather than going to Gmail and again sending an invite. This would be a bit of cumbersome task for the HR representative.
- **Add, Edit, and Delete Functionalities for Job Postings** - HR can Add new Job Postings, also if any changes are there, it can also be modified using edit option. If the vacancy for any job role is finished, then we can also delete the job postings.
- **Applicant Analytics incorporated** - Many of times, HR would not be kind a tech geek, so it would be difficult for them to use any BI tool for analytics. iProfiler solves the problem by incorporating the Analytics features in our plat form. So, the HR can have everything on the same plate, rather switching between different tools.

- **Resume parsing and automated Applicant form prefill** - User has 2 options - User needs to upload his/her resume and the form will be prefilled automatically. Or upload the resume and fill relevant details manually. (Note: Resume is kept as a required field even when the user doesn't want the resume parsing option.) For Option 1, Resume gets uploaded to our server and it is being loaded by Backend for resume parsing and past that the user details are saved into a JSON file. Form / Prefill from JSON file is done by using saved JSON file to populate values in the form and the user validates them. User clicks on submit, the information is added to the backend database collection. For option 2, the user fills the form and directly submits the information to the backend database collection.
- **24/7 contact support through email** - If at any point of time, HR face any issues while using our, they can reach out to us by using the email service. Our contact support team will try to resolve the problem as soon as possible.

3.1.5 Non-Functional Requirements

There are many non-functional requirements which are traded off between each another e.g. increased performance often comes at an increased total cost of ownership. Non-functional requirements for this application include-

- **Scalability** –The system should be scalable in terms of the following factors- the number of data feeds that the system can process at a time;in this case the number of trainees and batch owners that the portal can handle at a time.
- **Performance** -Performance is computed by the comparison the amount of work done to the time and resources that are required to do that work. The system should have quick response times i.e., it should response immediately in case of a request.
- **Modifiability** -Modifiability is the ease factor with which the system can be updated with new changes. The application should have easy process for updating new behaviours and data processing because in this field, updates and introduction of new features is quite common.
- **Reliability** – The application should be accurate and dependable so that it could get more correct outputs for the performance details of the trainees, for example, performance graph.
- **Fault tolerance** – The system should be able to tolerate the fault if occurred. This is like reliability, but a system must be reliable to use even after a fault has occurred.

3.1.6 Design Constraints

The software language used to implement our system is Java and TypeScript. Java is slower than C# and C++ but is widely used in Enterprise Application Development because it is a high-level language. Along with that it has several frameworks which make it a better performing and easy to use programming language. API's Development are carried out much more easily due to Java high-performing Spring Boot Framework.

The development tools that we have used in the project are as follows-

- React is used to develop the front-end of the application.
- React Bootstrap is used for the styling of the application.
- Email JS is used as the third-party integration service to send emails.
- Python Flask is a framework and is used to develop the backend of the application.
- MongoDB Atlas is used for the database.
- Google Cloud Platform provides Compute Engine which is used for deploying the application.
- GitLab is used for CI/CD pipeline.

3.2 ARCHITECTURAL DESIGN

3.2.1 Conceptual Design

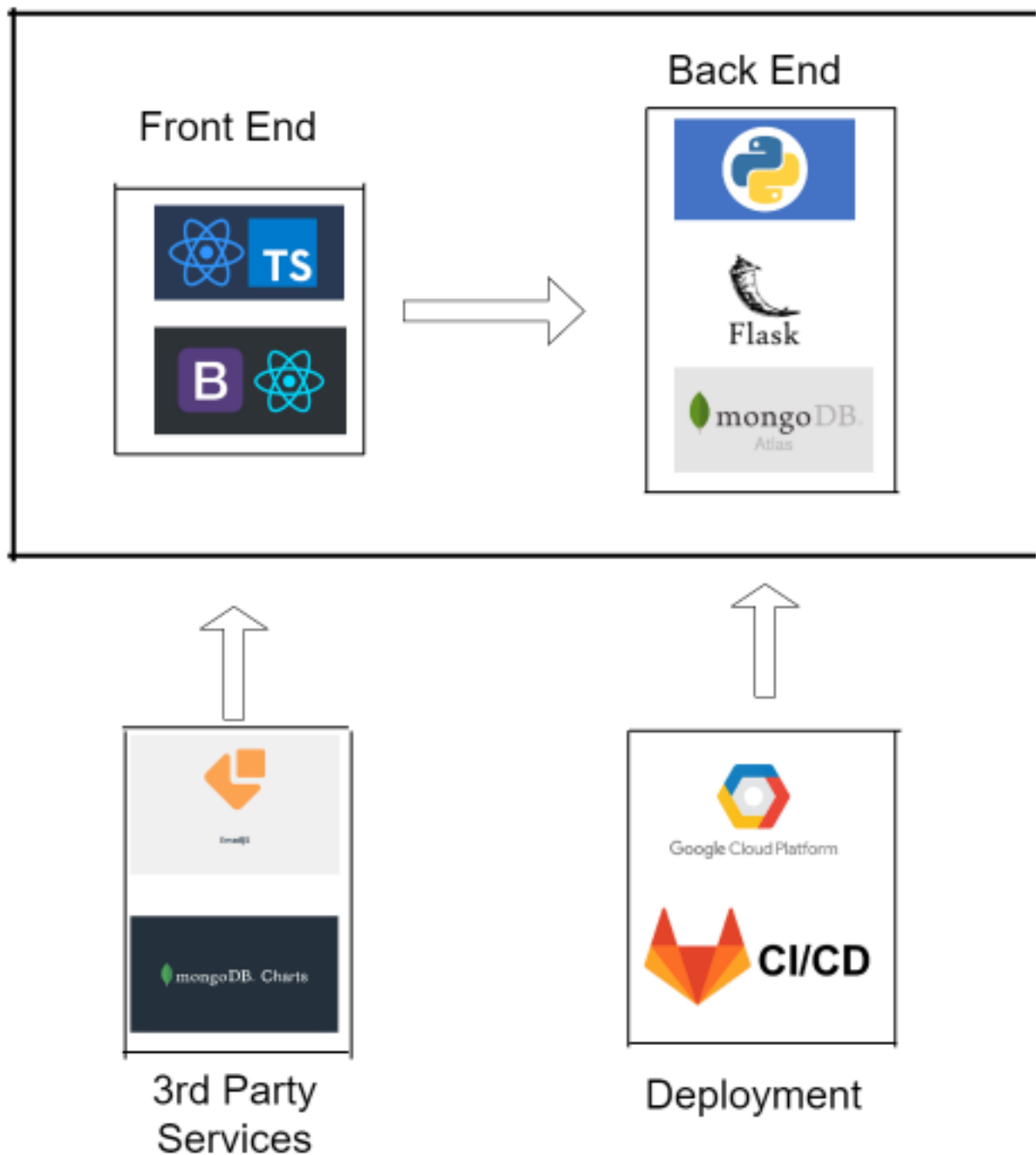


Figure 2 Architectural Diagram for iProfiler

The architecture of iProfiler is depicted in the above figure. It can be modularized into few components. It is composed of Front end , back end, Third party services and deployments.

iProfiler is a responsive web application which is built using the tech stack discussed in the previous section. Now let us understand the architecture-

FrontEnd

The Front end for the app is built on React framework. All operations and communications with backend are performed via various REST API endpoints. The frontend is built with accordance to Responsive Web Design (RWD) and DRY(Don't Repeat Yourself) principle. Depending on whether the app is opened on a pc/laptop or a mobile phone, responsive frontend UI is presented to the user.

Backend

Backend for the app is built as a REST API. Here we are using Flask Web Framework to build light weight applications. It provides various API endpoints to perform various functions like adding users details, store resume data, CRUD operations for Job Postings etc. The REST API is built using Flask. The API endpoints perform all operations of connecting with the DB, Authenticating, Searching etc. CORS is enabled for the API so it only accepts requests from the React front-end.

Data Layer

App Data

For the Database to store all Resumes, Job Postings and user related data, I am using MongoDB Atlas Database. The Flask API backend will be performing all DB operations connecting to the DB and getting requests from the frontend. MongoDB Atlas DB is deployed in Amazon Web Service (AWS) server; hence it gives us lightning fast speed operations.

Auth Data

In order to accommodate the session storage, we are using browser's cookies in order to store the information of the logged in user. To perform this, we leverage the session library from Flask in order to create and get session data. For e.g. if the web-app is opened on a laptop/pc, once user logs in, the session data is stored in the Browser's cookies. Any time user closes and reopens the web-app, the web-app verifies the session with any unique identification using API.

Third Party Services

Email Js

This Email Js is used as a service to enable the Email Functionality in the web application. A general template can be designed in their portal and customized according to their needs or requirements. Most importantly it is a free to use service, but has limited the number of emails that can be made using an access ID.

MongoDB Charts

This yet another Third-party service used for analytics purpose of our HR Dashboard. This would be useful tool for the HR to have a better understanding of various interesting patterns of recruitments. Since we are using MongoDB Atlas DB for storing the important data, it would be easy for us to integrate MongoDB Charts to our web application. It gets refreshed after every 26-28 minutes.

Deployment

Google Compute Engine Deployment

GCE here is used for deployment of our react-flask app in Google Cloud Platform (GCP). Google Compute Engine provides a scalable number of virtual machines (VMs) to serve as large compute clusters for that purpose. GCE can be managed through a RESTful API, command line interface (CLI) or Web console. Compute Engine is a pay-per-usage service with a 10-minute minimum.

GitLab CI/CD

For our project, we have created repository in Gitlab. So, every time we write any code or make any updates or complete any features, we push our code to the repository, so that its version control is saved over there, and we would have a backup of the work we did.

A runner is installed in the Gitlab, which helps Gitlab run its CI/CD pipeline. Git Lab CI/CD is a tool built into GitLab for software development through the continuous methodologies:

- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Deployment (CD)

Continuous Integration works by pushing small code chunks to your application's code base hosted in a Git repository, and to every push, run a pipeline of scripts to build, test, and validate the code changes before merging them into the main branch. Continuous Delivery and Deployment consist of a step further CI, deploying your application to production at every push to the default branch of the repository. These methodologies allow you to catch bugs and errors early in the development cycle, ensuring that all the code deployed to production complies with the code standards you established for your app.

3.2.2 Flow Chart

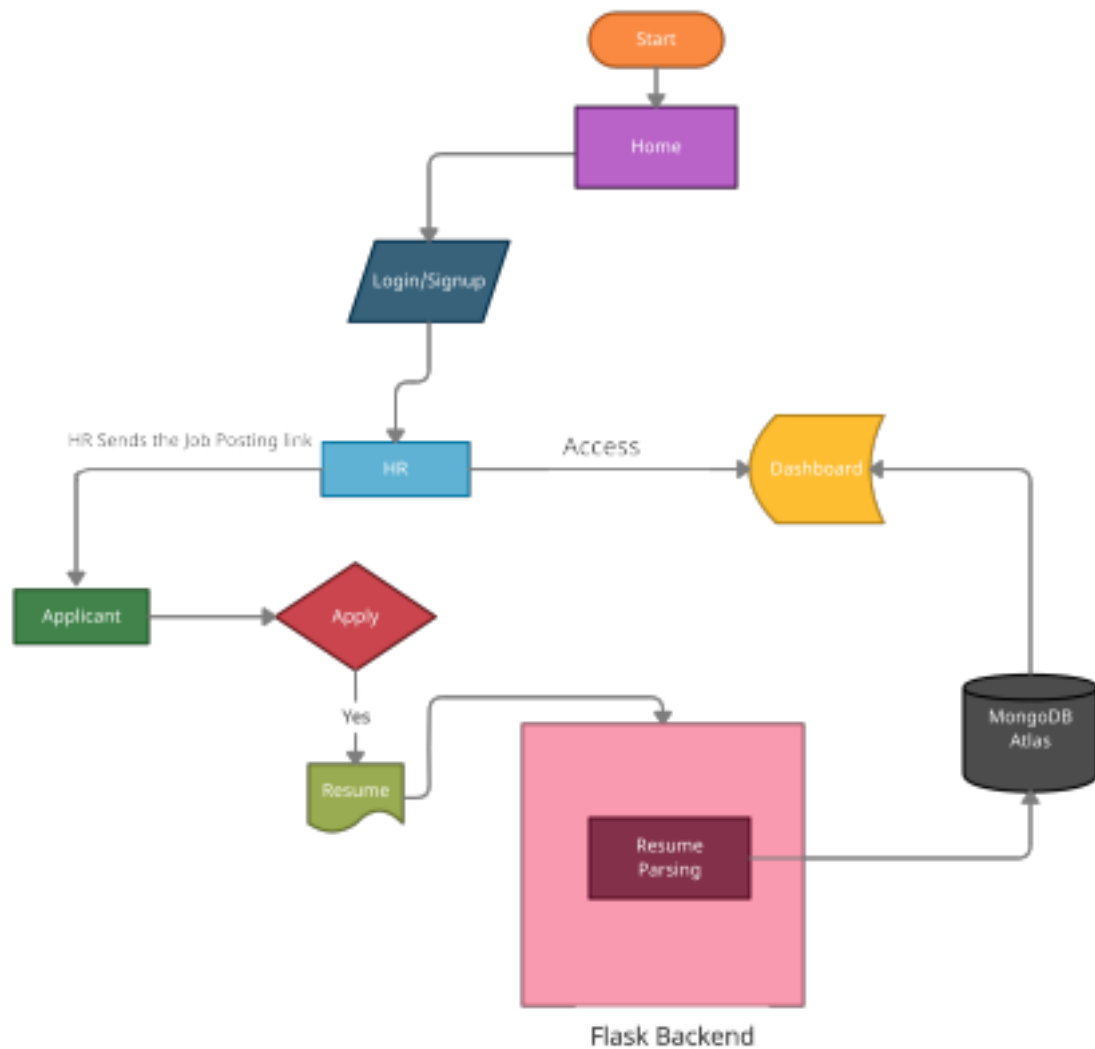


Figure 3 Flow Chart for iProfiler

The above figure 4.2 depicts the flow of the web-app. The flow can be classified into 2 major parts:-

User Flow (HR Flow)

When an HR opens our webpage, they get to see our Landing page, which covers basic idea of why a HR needs to use our tools. Also, it portrays the process by which things are going to happen. When HR clicks on Start button, it goes to Login Page, where the user can login using its credentials or goes to signup form, where they can register themselves.

Once Logged in, a Dashboard is displayed. It portrays all applicant's data who had applied to our web app for various job roles. Each applicant displayed in the form of a card with all the basic details. HR can view details of any applicant by clicking on the concerned card, if his / her profile

sounds interesting to the user, then HR can schedule a meet for next rounds. Also, HR can also invite any external candidate. The HR can also add new Job Postings.

Applicant's Flow

They receive a link to apply in their concerned email. Applicant has to upload its resume, so that it can parse the document, and generate the json and pre-populate it in the Apply Page. If any details are incorrectly filled, applicant can change it before submitting the form.

3.3 WORKING

The working of the application developed can be explained using the proposed flow chart of the system. The working is categorized into two flows- HR flow and Applicant's flow.

When an HR opens our webpage, they get to see our Landing page, which covers basic idea of why a HR needs to use our tools. Also, it portrays the process by which things are going to happen. When HR clicks on Start button, it goes to Login Page, where the user can login using its credentials or goes to signup form, where they can register themselves. Once Logged in, a Dashboard is displayed. It portrays all applicant's data who had applied to our web app for various job roles. Each applicant displayed in the form of a card with all the basic details. HR can view details of any applicant by clicking on the concerned card, if his / her profile sounds interesting to the user, then HR can schedule a meet for next rounds. Also, HR can also invite any external candidate. The HR can also add new Job Postings.

They receive a link to apply in their concerned email. Applicant must upload its resume, so that it can parse the document, and generate the json and pre-populate it in the Apply Page. If any details are incorrectly filled, applicant can change it before submitting the form.

3.4 BENEFITS

iProfiler has the following benefits-

- It provides secure login and signUp for the HR or the recruitment teams for the different organizations.
- The tool is a complete package of resume filtering and communicating with different applicants for the recruitment process.
- It provides highly efficient filter and search functionalities in the dashboard to segregate perfect candidates from a pool of generic applicants.
- It provides integrated email service through which the HR can send apply link invites and the status of further round of interviews.
- It provides 24/7 help support where you can mail your issues and will get a reply with 2 business days.
- It provides a job posting screen where the HR can add, edit and delete a job posting based on the requirement of the organization.
- It provides a analytics screen in the dashboard where the HR can view the status of the candidates in a graphical format.

3.5 TOOLS AND TECHNOLOGIES

3.5.1 Front End

React Typescript

React is a “JavaScript library for building user interfaces”, while Typescript is a “typed super set of JavaScript that compiles to plain JavaScript.” By using them together, we essentially build our UI’s using a typed version of JavaScript.

Why TypeScript? There are many benefits of TypeScript. Let’s discuss a few of those:

- **Easy to read and understand components** -With TypeScript, it’s easy to define Prop types, making the code much easier to read and use. And this will accompany by IntelliSense support plus static type checking. These, in combination, make it a great development experience and reduce the potential for bugs. Besides, adding comments to Prop types also adds more readability when you check a component definition.
- **Better support for JSX** -Another additional benefit of TypeScript + React is that it provides better IntelliSense, code completion for JSX.
- **Benefits comes with Static type checking and IntelliSense in general** - Static Type checking helps to identify errors earlier. The above example correctly identifies that the `getDetails()` function requires a string as an argument rather than a number. When we take TypeScript static Type checking and IntelliSense together, it gives you the confidence that IntelliSense is 100% accurate.

React Bootstrap

React-Bootstrap replaces the Bootstrap JavaScript. Each component has been built from scratch as a true React component, without unneeded dependencies like jQuery. As one of the oldest React libraries, React-Bootstrap has evolved and grown alongside React, making it an excellent choice as your UI foundation.

Why React Bootstrap?

- React-bootstrap creates React components for you. The advantage will be obvious if you consider how to do animations with the help of bootstrap in your React project. Without react-bootstrap, you need something like `CSSTransitionGroup`. You cannot take advantage of

bootstrap's API because it will manipulate the DOM, which makes the React behavior unpredictable.

- Also, bootstrap will hide details of their animation API; that is to say, you cannot use it at all. With bootstrap components, however, you do not need to worry about how the animation is implemented by bootstrap, you just need to specify properties and hooks in a component and the library will do the trick. More specifically, it can add some wrappers which are not visible to the client.

3.5.2 Back End

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.

Why Python?

- **Readable and Maintainable Code** - While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.
- **Multiple Programming Paradigms** - Like other modern programming languages, Python also supports several 18-programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.
- **Compatible with Major Platforms and Systems** - At present, Python supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The

feature makes it easier for you to make changes to the code without increasing development time.

Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Why Flask?

- Integrated support for unit testing
- Built-in development server and fast debugger
- Restful request dispatching
- Flask has a modular design and lightweight so that it can easy to transit into web framework with some extension
- You can plug your favorite ORM
- Highly flexible
- It is easy to deploy the flask in production

MongoDB Atlas

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Google Cloud, and Azure with best-in-class automation and proven practices that guarantee availability, scalability, and compliance with the most demanding data security and privacy standards. It provides various types of services depending upon the client's needs. It is extremely scalable and fast. It is hosted in Amazon Web Service (AWS)

Why MongoDB Atlas?

- One of the biggest reasons to use MongoDB Atlas is that it's in it with you for the long haul; you don't have to worry about finding a new product as you scale. It will keep an eye on production and security for you and has insanely high throughput and low latency, even at a high scale. MongoDB's Atlas team produces with developers in mind and they're with you every step of the way.

- MongoDB Atlas makes it easy to set up, operate, and scale your MongoDB deployments in the cloud because it is intuitive and highly automated. They've taken the guesswork and intensive labor out of building and maintaining infrastructure, but they'll still let you have said where you want to. They'll get you up and running and their replication will help you stay there. From high availability to scalability, security to disaster recovery — the MongoDB Atlas platform has you covered.

3.5.3 Integrations

Email JS for Email functionality

EmailJS helps to send emails using client-side technologies only. No server is required – just connect EmailJS to one of the supported email services, create an email template, and use our Javascript library to trigger an email.

Email templates can optionally contain dynamic variables in almost any field (e.g. subject, content, TO address, FROM name, etc.) which are populated from the Javascript call. For example, the subject can be "Heyname, you have a new message", and using JavaScript the name can be set to "Isha Gupta", for instance.

MongoDB Charts

MongoDB Charts is a modern data visualization tool that is integrated with the MongoDB cloud data platform. It is the best way to create, share and embed visualizations of MongoDB data.

Why use MongoDB Charts?

Charts is the quickest, easiest, and most powerful way to visualize MongoDB data for real-time insights, business intelligence, and embedded analytics.

3.5.4 Cloud and Deployment

Google Compute Engine Deployment

Google Compute Engine provides a scalable number of virtual machines (VMs) to serve as large compute clusters for that purpose. GCE can be managed through a RESTful API, command line interface (CLI) or Web console. Compute Engine is a pay-per-usage service with a 10-minute minimum.

GitLab CI/CD

GitLab CI/CD is a tool built into GitLab for software development through the continuous methodologies:

- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Deployment (CD)

Continuous Integration works by pushing small code chunks to your application's codebase hosted in a Git repository, and to every push, run a pipeline of scripts to build, test, and validate the code changes before merging them into the main branch.

Continuous Delivery and Deployment consist of a step further CI, deploying your application to production at every push to the default branch of the repository. These methodologies allow you to catch bugs and errors early in the development cycle, ensuring that all the code deployed to production complies with the code standards you established for your app.

3.6 MODULES

The web-app is composed of many screens, which is rendered in front end. Each screen has its own purpose and its own defined functionalities. We have followed the Single ton Design Pattern, which indicates that each component/screen should solely serve a single purpose. It should be a complex screen; it should be most decomposed screen as possible.

3.6.1 Screen 1: Recruiter Login/Signup

For Login: HR uses the company login of the website, which will redirect it to the dashboard where the list of all applicants would be visible (only to the admins i.e. Recruiter).

For Signup: Recruiter could use our services by opting any of the pricing plans according to their needs and creating his/her account using the company's mail ID as the primary signup service.

3.6.2 Screen 2: Landing Page

The Landing page of our product having all the information about what our web site is all about. It consists of:

- Home
- Recruiter Login/Signup
- Pricing
- Contact Us

3.6.3 Screen 3: Job Details

Applicants will get a link of this route in their email for letting them know out the current openings. Ones who are interested in the job opening and want to apply, need to fill in the application form. Current Job Postings - As the user clicks on the link, the user is able to see all the job postings and could apply for the same.

3.6.4 Screen 4: Apply Page

After selecting a job role:

User has 2 options -

User needs to upload his/her resume and the form will be prefilled automatically. Or upload the resume and fill relevant details manually.

Note: Resume is kept as a required field even when the user doesn't want the resume parsing option.

For Option 1:

Resume gets uploaded to our server and it is being loaded by Backend for resume parsing and past that the user details are saved into a JSON file. Form / Prefill from JSON file: Saved JSON file is used to populate values in the form and the user validates them.

User clicks on submit; the information is added to the backend database collection.

For option 2:

The user fills the form and directly submits the information to the backend database collection.

3.6.5 Screen 5: Pricing

Our revenue model is a subscription model based on the allowance in the number of Job Postings HR could put forth and/or number of applicants allowed to participate in hiring.

We have 3 Models available for HR/Recruitment Team: -

- Basic
- Standard
- Pro

We are following the Freemium Model where we give the HR/Recruitment Team a 10-day free trial to use this application.

3.6.6 Screen 6: Dashboard

HR can view the job postings as well as the list of applicants. Functionalities:

- Search based on Names
- Filter based on skills, experience and job role.
- Invite External Applicant to apply for a role.

Analytics which helps the HR to understand the trends.

3.6.7 Screen 7: Job Postings

It allows a HR/Recruitment Team to add/delete any job role depending upon the current openings available at a point in time. Also, it provides a feature that can be used to Edit the Job Posting if any changes occur at some point later. Also, the Job Posting is dynamically updated after any CRUD operation occurs.

3.6.8 Screen 8: Applicants

It displays information of an applicant. The details are as follows: -

- First Name Last Name
- Degree(UG/PG)
- Experience
- Skills
- Status of Applicant
- Email icon which allows to invite an applicant for further rounds.
- View Profile button to get more insights of an applicant.

3.6.9 Screen 9: User Profile

This page helps the HR/Recruitment team to know the candidate a bit better. The limited content in the user profile saves a significant amount of time from the recruitment process.

If in case, HR/Recruitment Team is more interested in any applicant, then HR/Recruitment Team can always download their resume for a deep insight on the applicants.

CHAPTER-4 RESULTS AND ANALYSIS

The three main foundations of a project are the results, discussions, and conclusions; without them, the project would not be finished. We would not be able to better ourselves if we do not do these three things on a consistent basis. This chapter contains different screens that is developed for the web app. Also, the analytics of the applicants are displayed as screenshots in this chapter. The scope of future works is also discussed in this chapter.

4.1 SCREENS

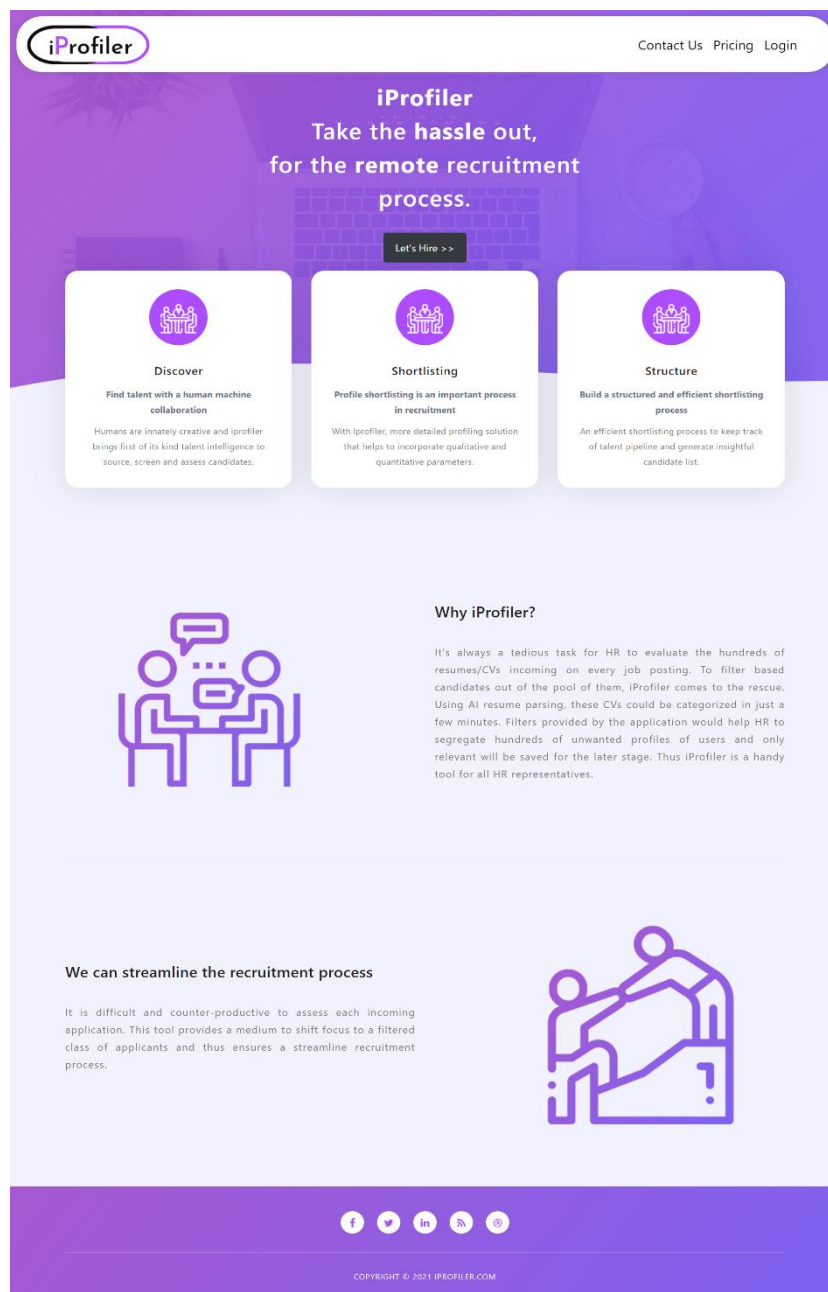


Figure 4 Landing Page

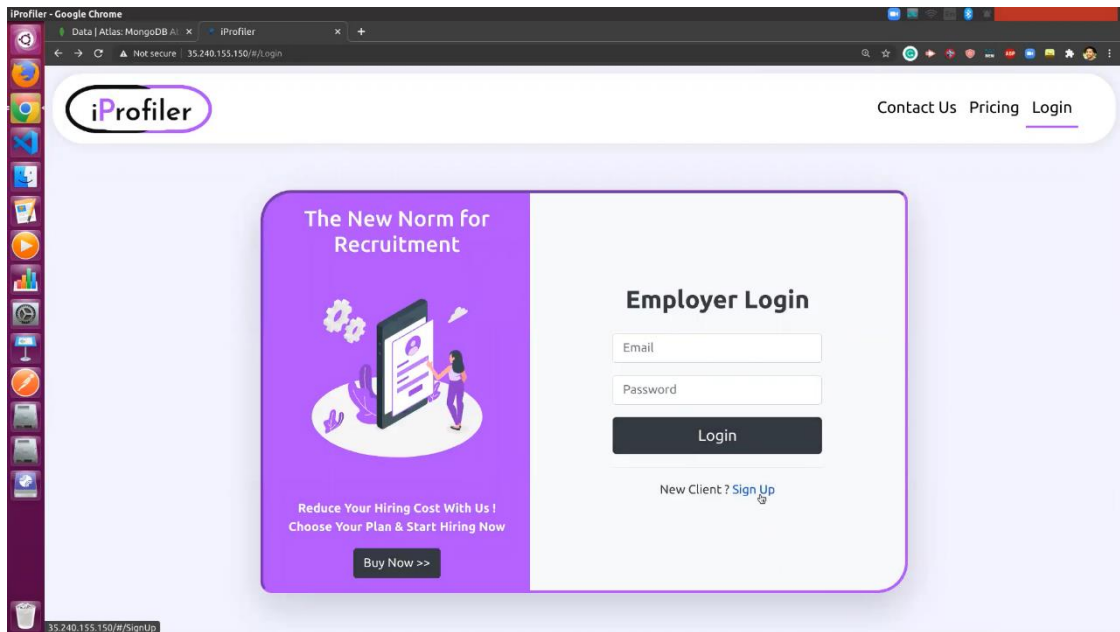


Figure 5 Login Screen

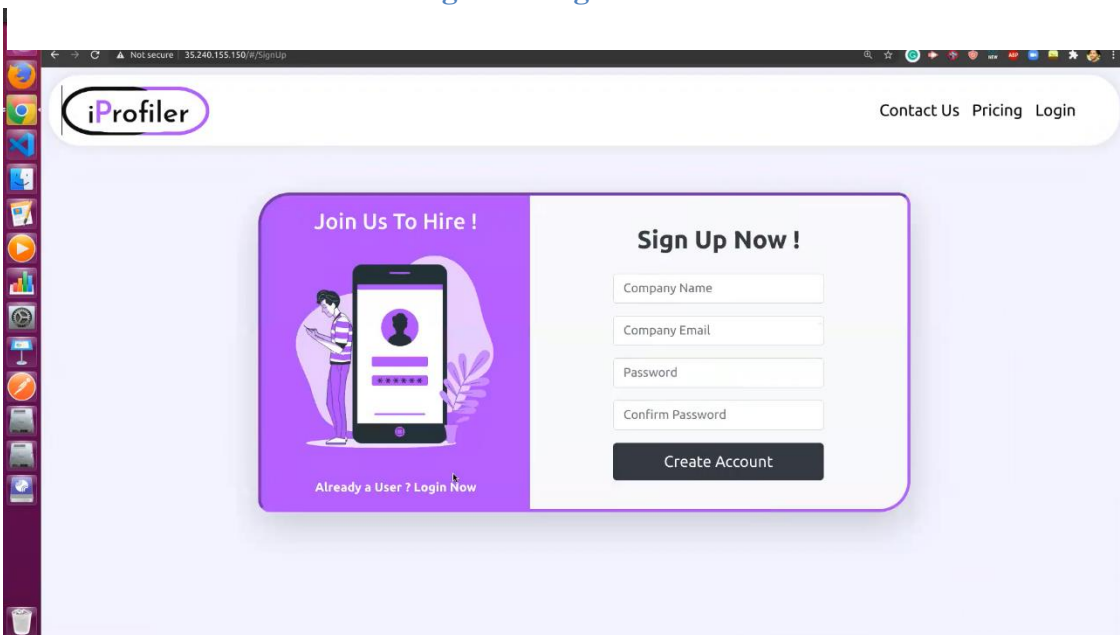


Figure 6 SignUp Screen

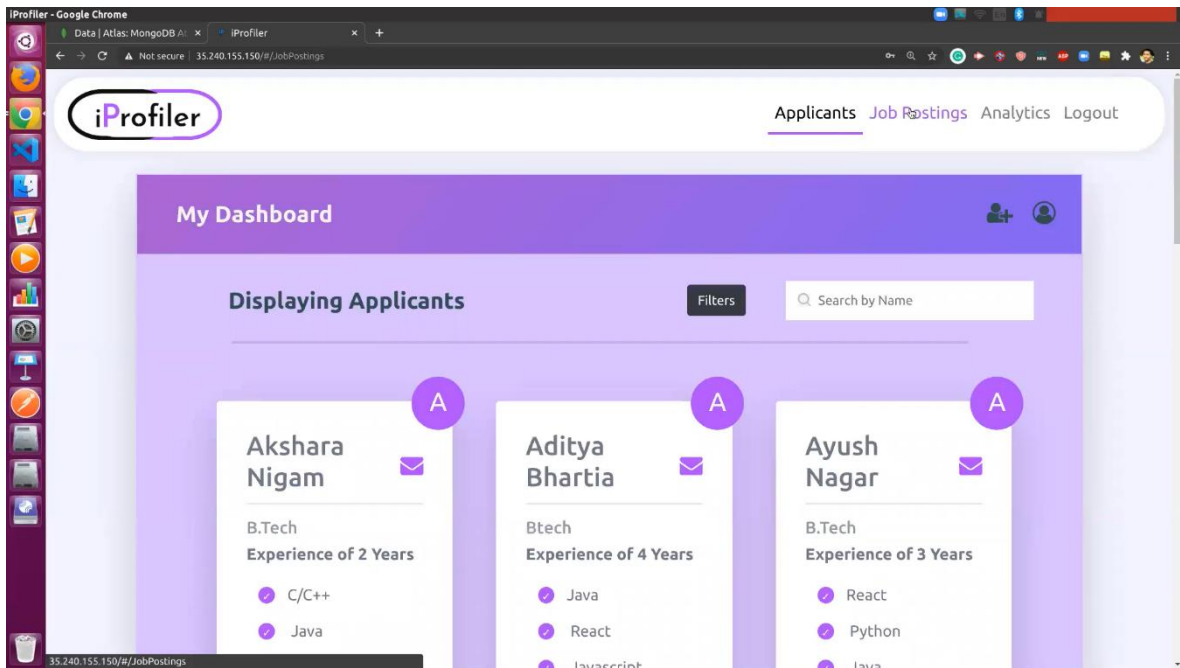


Figure 7 Dashboard Screen

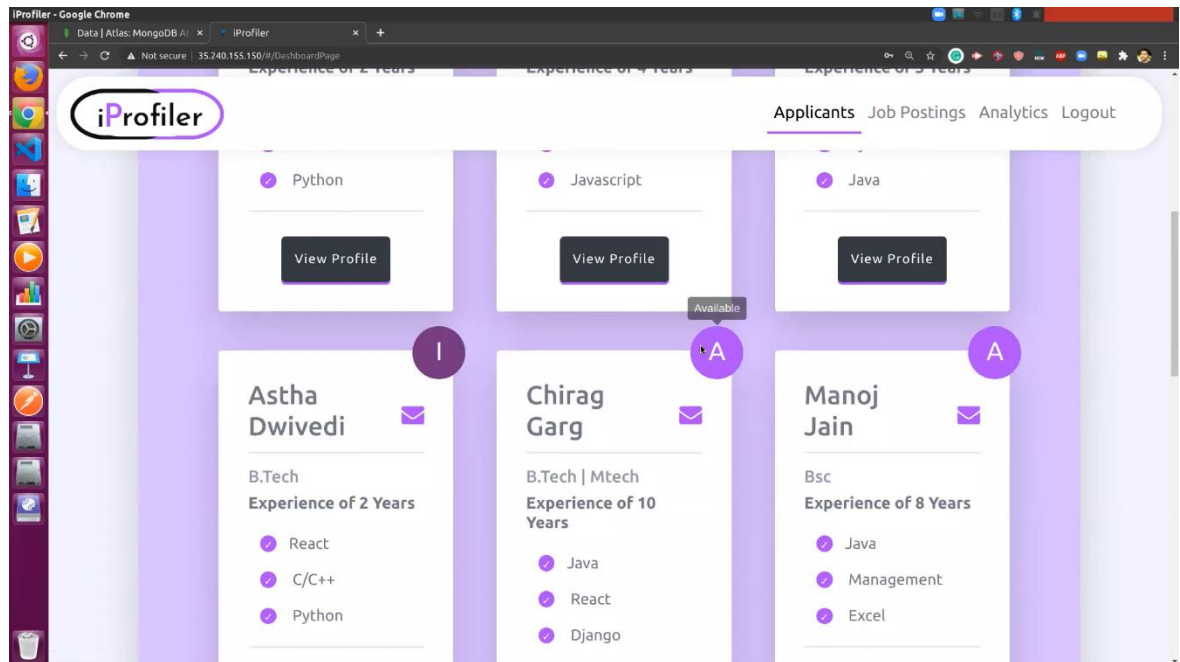


Figure 8 Current Status of Applicants

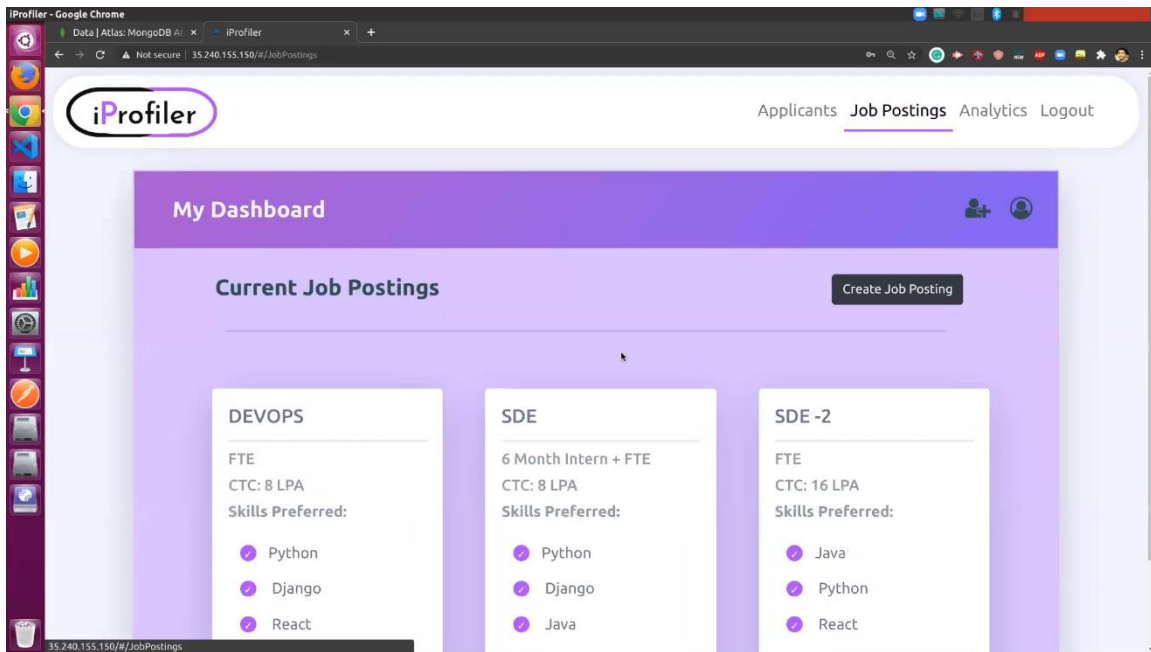


Figure 9 Current Job Postings

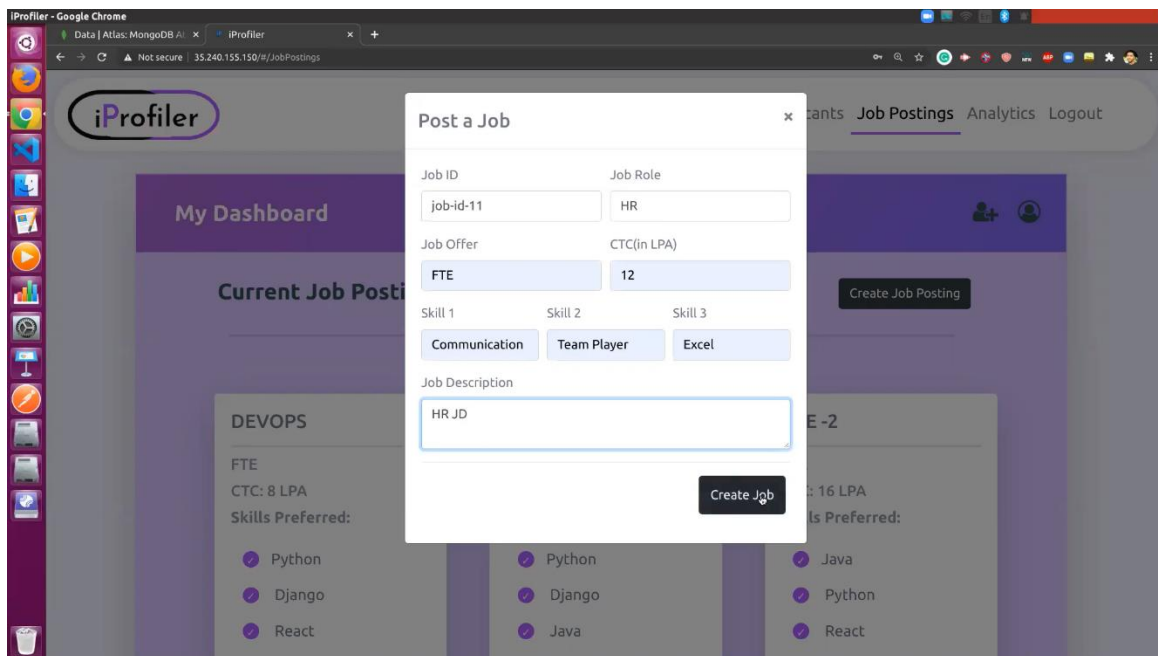


Figure 10 Add a Job Posting

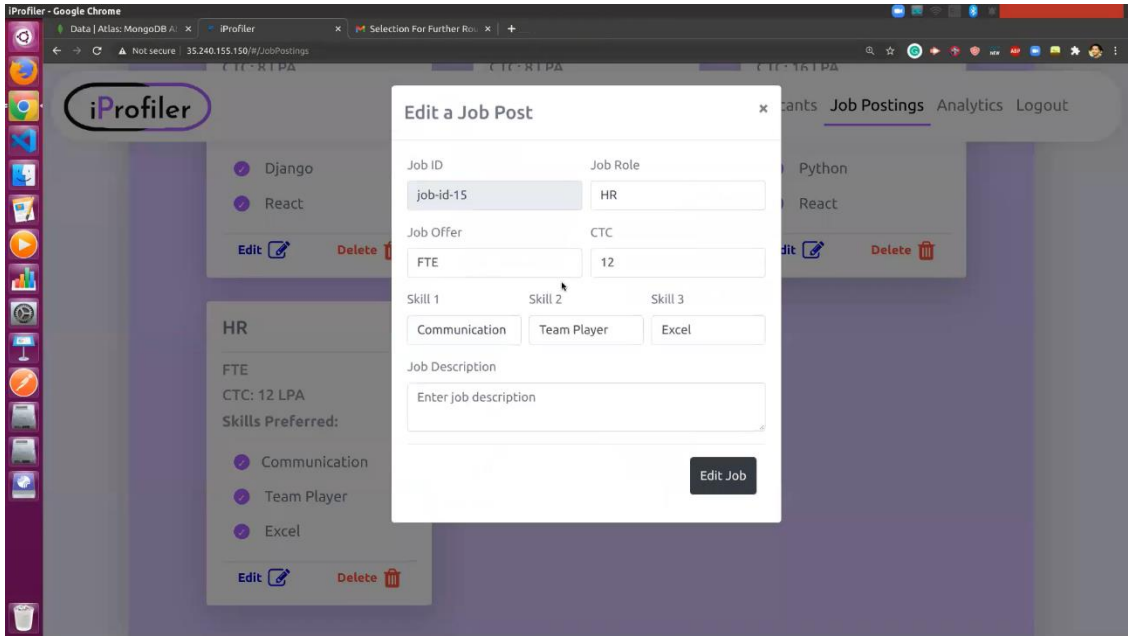


Figure 11 Edit a Job posting

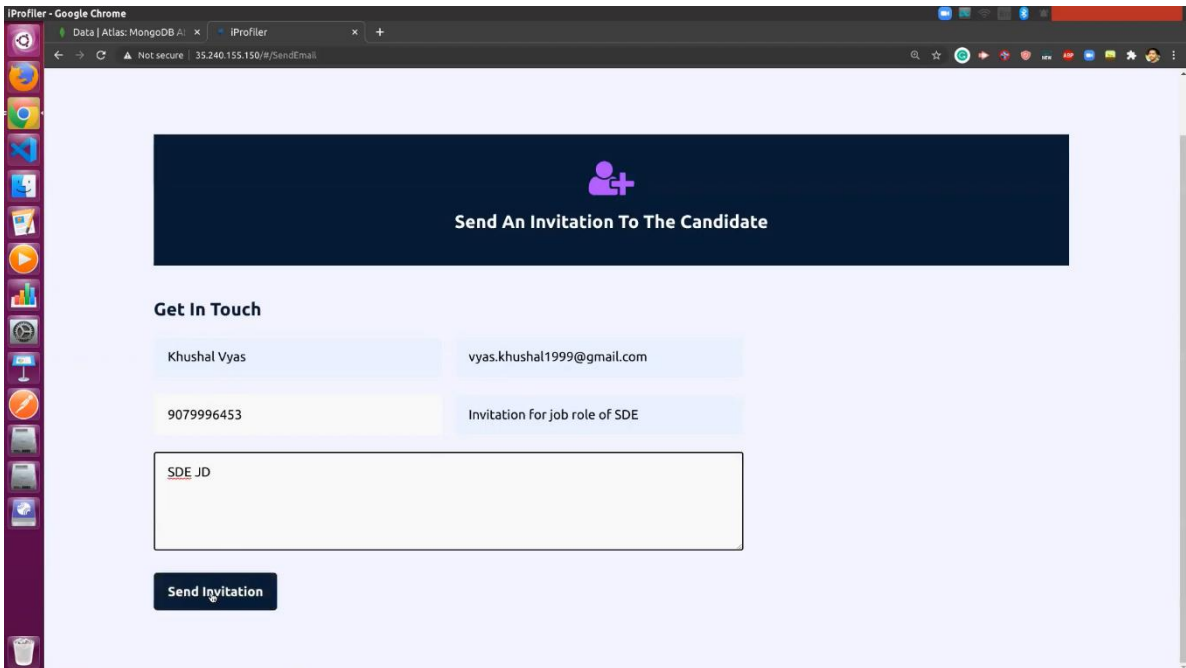


Figure 12 Send Invitation to a Candidate

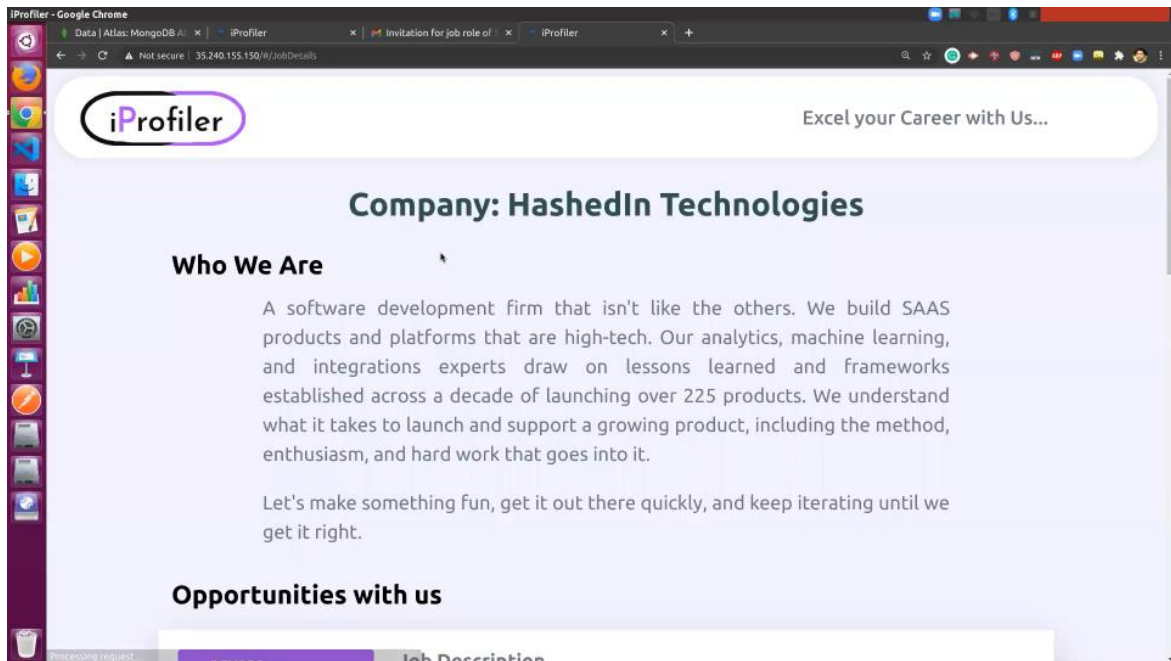


Figure 13 Company Page

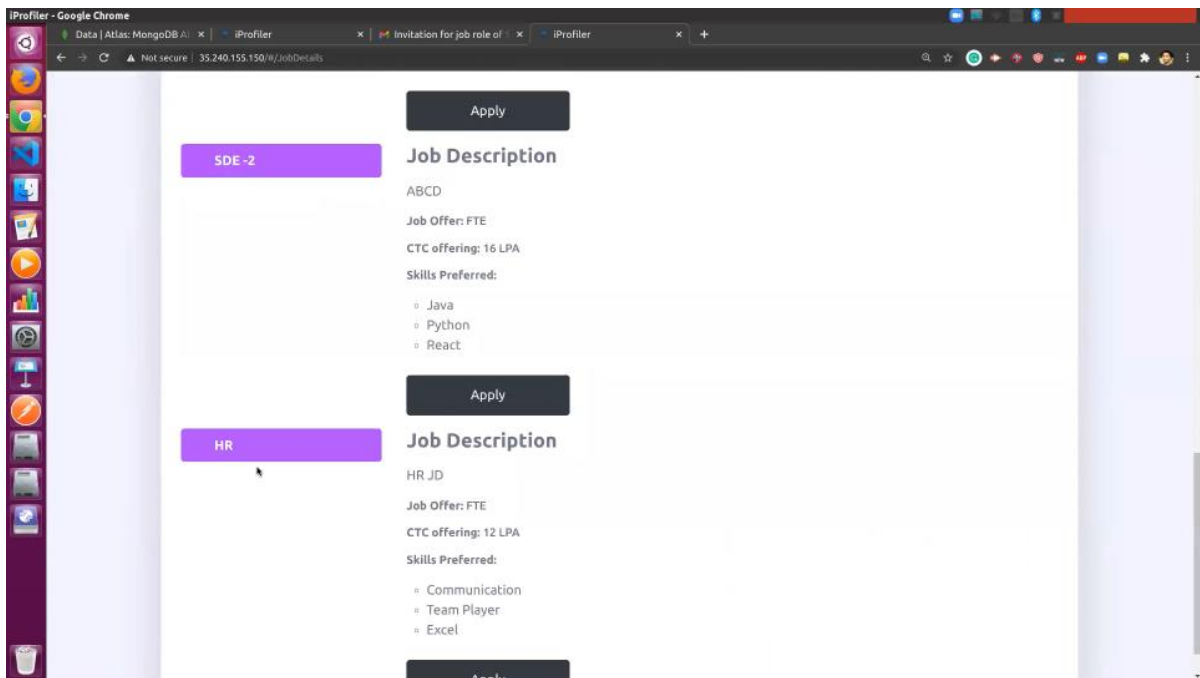


Figure 14 Available jobs in an Organization

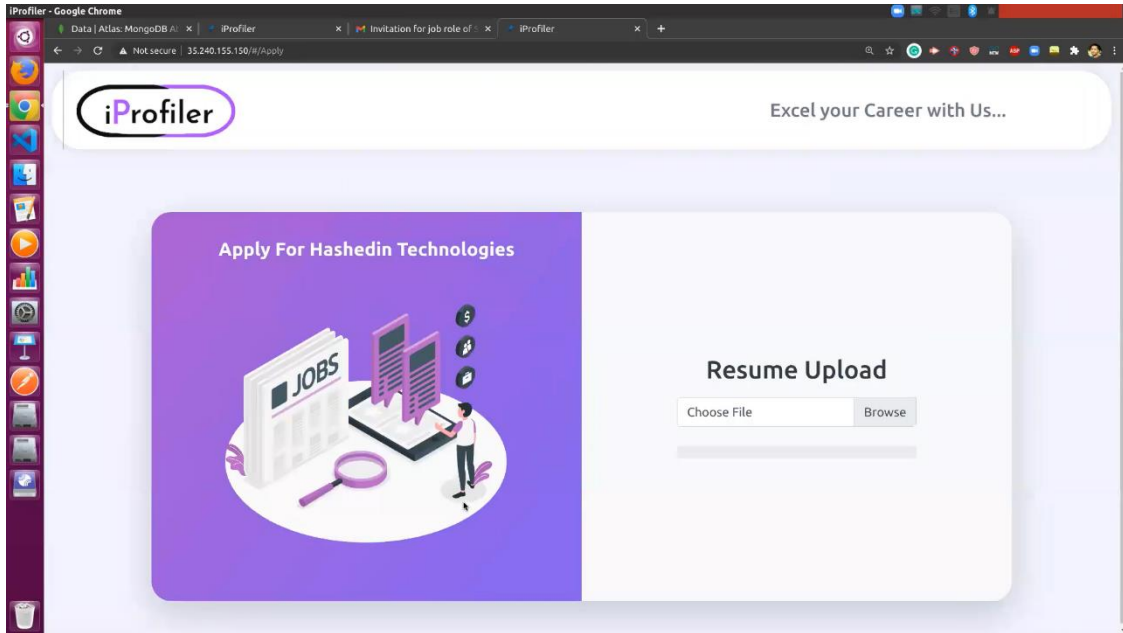


Figure 15 Resume Upload

A screenshot of the iProfiler website showing a form titled "General Details" and "Educational Details". The browser's address bar shows the URL "35.240.155.150/Apply". The "General Details" section includes fields for First Name (Khushal), Last Name (Vyas), Select DOB (dd/mm/yyyy), Email (vyas.khushal1992@gmail.com), Address (1234 Main St), City (Enter City), State (Enter State), Zip (Enter 6 digit pincode), Contact Number (+91-9079996453), and Alternate Contact Number (eg:- 932154XXXX). The "Educational Details" section includes fields for Post Graduation Course (Enter PG Course), College (Enter PG College), Percentage (Enter PG Percentage), Under Graduation Course (Enter UG Course), College (Enter UG College), and Percentage (Enter UG Percentage).

Figure 16 Prefilled Form after Resume Upload

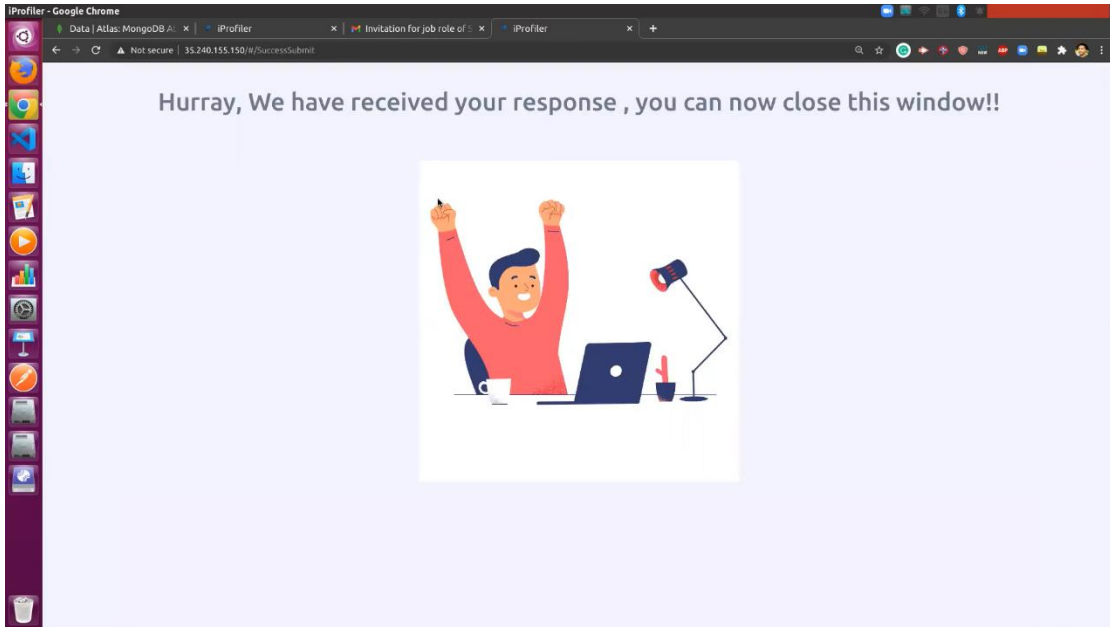


Figure 17 Successful Submission

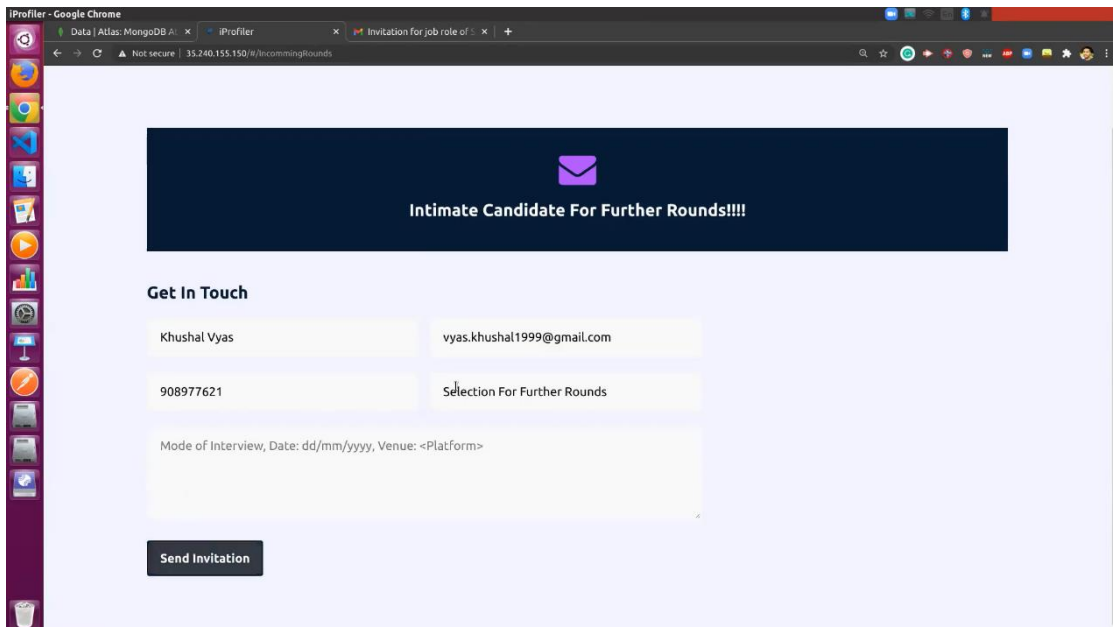


Figure 18 Invitation to Candidates for further rounds

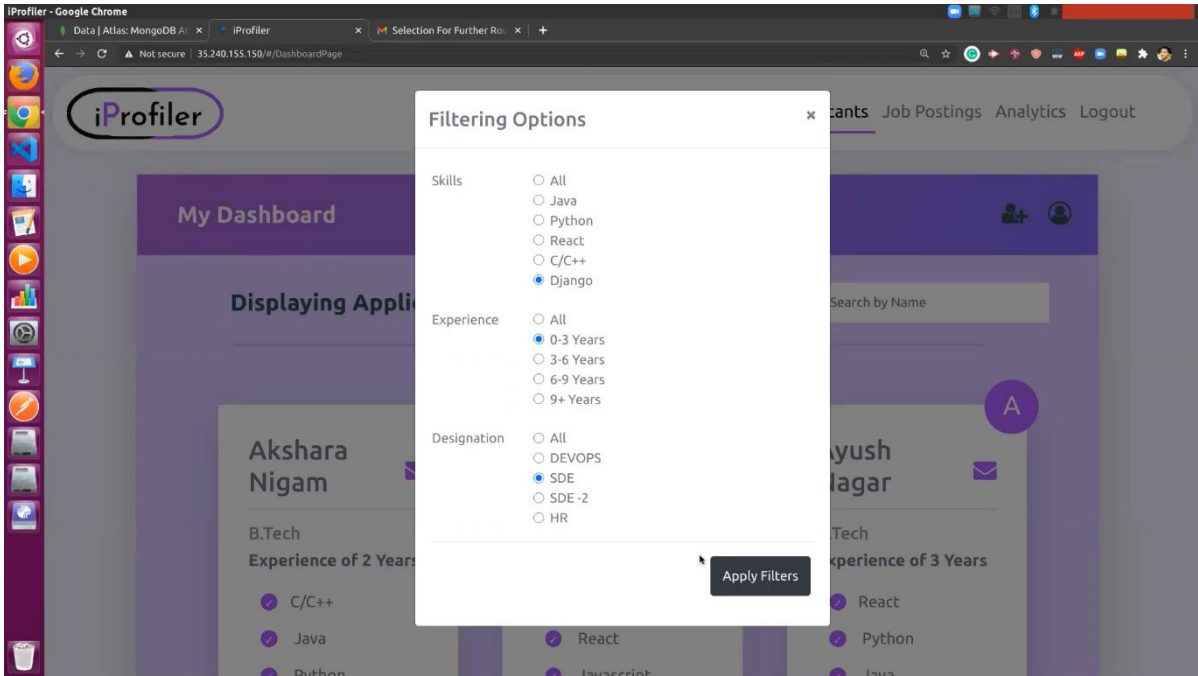


Figure 19 Filter Form

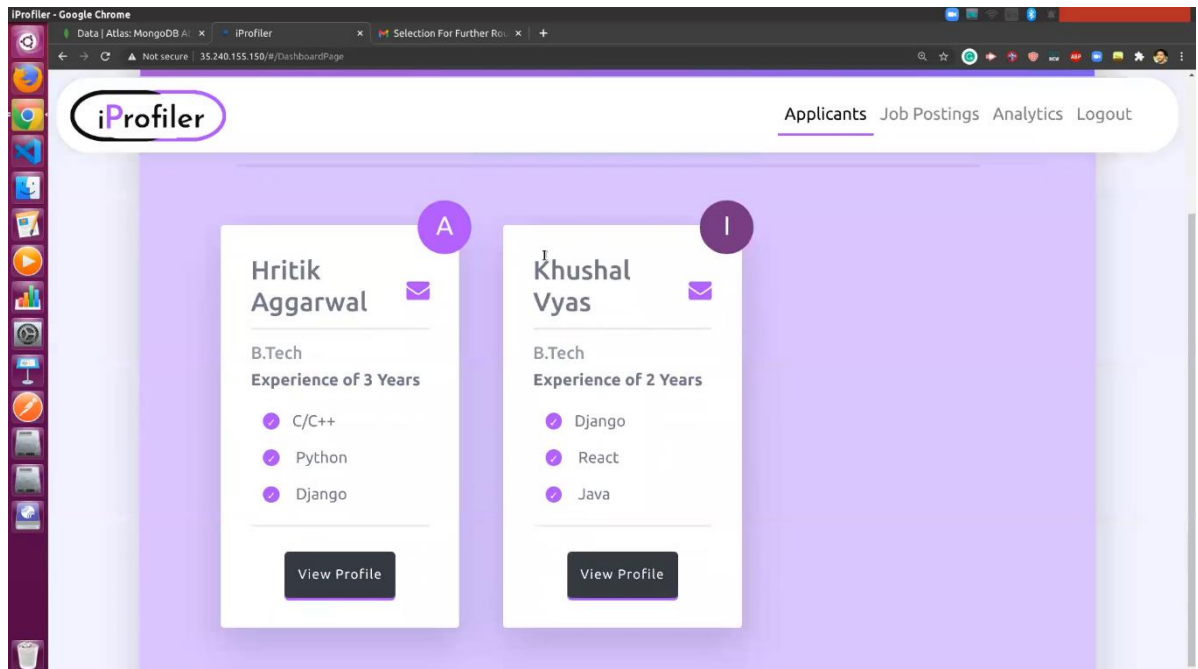


Figure 20 Result of Applied Filters

[← Back to Dashboard](#)



Akshara Nigam

aksharanigam1298@gmail.com

Personal Information

Contact	Address	Date of Birth
9089776212	123,Udaipur,Raj-231300	1999-01-04

Educational Details

Degree	College	%
B.Tech	institute of information technology	89
Degree	College	%

Skills

C/C++	Java	Python
-------	------	--------

Work History

Company	Designation	Experience
	SDE	2 yrs

[Download Resume](#)

Figure 21 View Profile and Download Resume

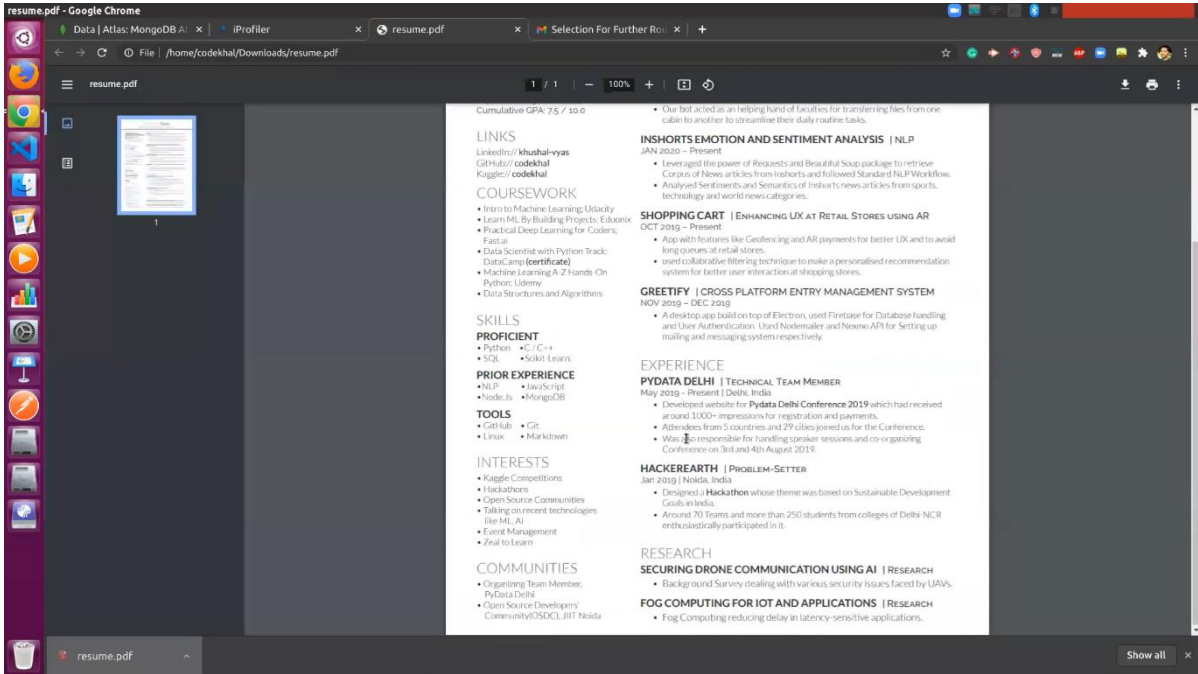


Figure 22 Downloaded Resume

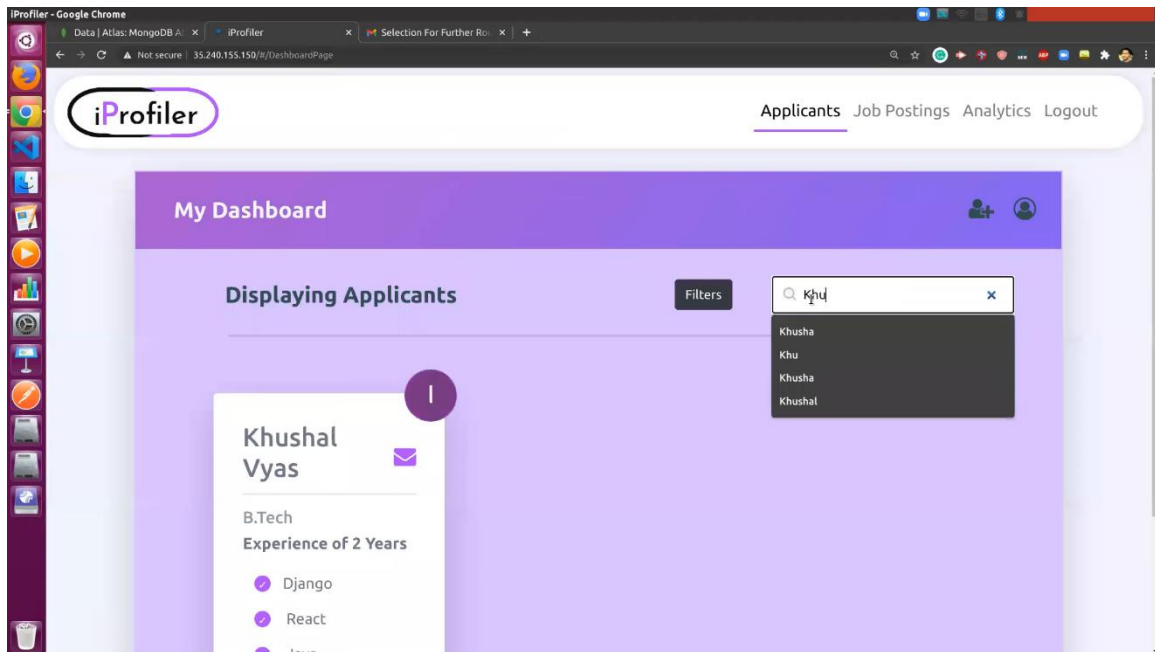
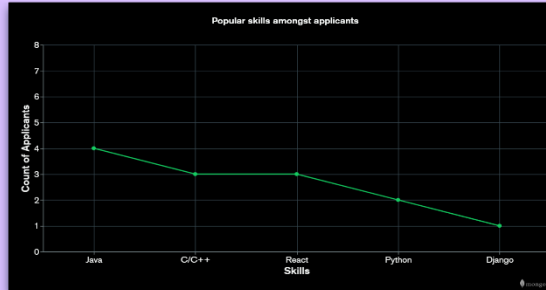


Figure 23 Search Candidates by Name

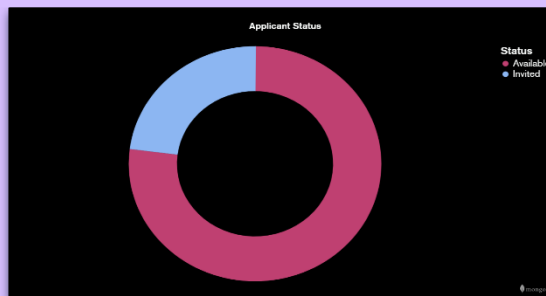
My Dashboard



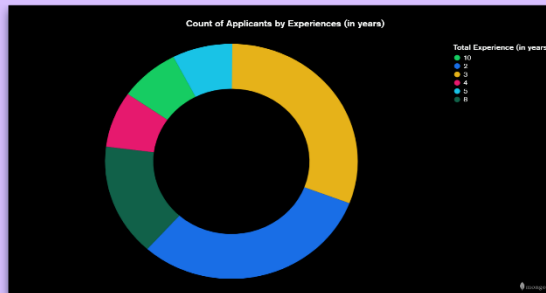
Popular skills amongst applicants



Status of Applicants(in %)



Applicant Count by Experience(in Years)



Applicant Count by Job Role

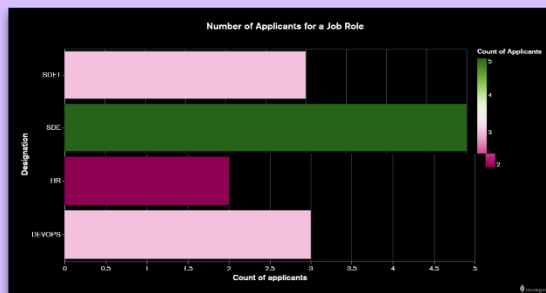


Figure 24 Analytics Page

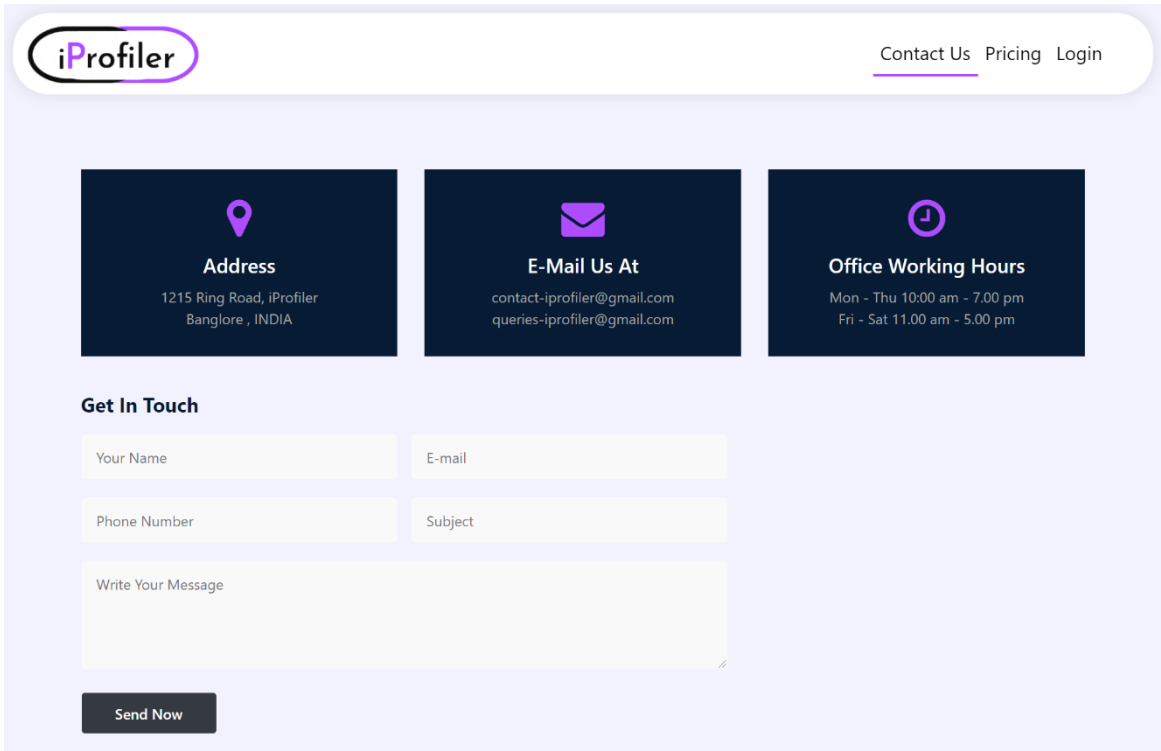


Figure 25 Contact Us Page

4.2 COMPARITIVE STUDY AND DISCUSSIONS

We read a lot of research articles, as discussed in the literature review. Any academic paper that has been studied has given their own insight on how to tackle the time-consuming process of manual resume screening. We realized that each strategy discussed in the literature reviewed above has its own set of advantages and disadvantages. Given that the HR position of many organizations is in charge of existing and future staff, our literature review reveals that HR managers must exercise caution when applying algorithmic decision-making, upholding privacy and fairness considerations, and tracking and auditing the algorithms that are used.

The thesis contributes to a better understanding of the present research field by summarising emerging evidence and potential research directions in the increasingly important area of algorithmic decision-making. Without a doubt, recent research has advanced our understanding of how firms use algorithmic decision-making in HR hiring and development, as well as when and why unfairness or biases emerge in algorithmic decision-making. However, our findings suggest that leading management journals should pay more attention to emerging computer science debates regarding the justice and potential injustice in algorithms. As a result, in order to get the best of all worlds, we must anticipate a collaborative method of automation and human intervention. Having said that, this would undoubtedly increase the efficiency of incoming recruits.

During internship, we had a quality discussion between fellow interns, senior developer, designers and Product Owner during daily scrum meetings. We had to complete the entire web app within the duration of 1 month. After which there was a product Expo, where we had to present our product to the fellow Hashers(Full time Employees in HashedIn are referred as Hashers) and Linkers(Interns in HashedIn are referred as Linkers). We had received appreciations from fellow Hashers Likers. There was a judging panel for this expo, which includes experienced industry experts, We got positive feedback from the judging panel. Overall, it was exciting to have developed something end to end, within tight time constraints.

CHAPTER-5 CONCLUSION

5.1 SYSTEM CONCLUSION

The implementation of the resume filtering application (iProfiler) is done. The system aims for getting the complete flow to the HRs and the recruitment teams of different organization. Over all, the application provides a secure login/signup functionality, a dashboard for the HRs where they can have a gallery view of the applicants, search and filter functionalities, view complete details of interested candidate and download its resume for future reference, view status of applicants, job details section where job roles can be added, altered and deleted, and finally an analytics section to view the complete picture of the recruitment status from the applicants' point of view. The application has successfully developed to provide a great user experience.

5.2 FUTURE SCOPE

Android App

Deploy an android app so that HRs could use it through their phone itself and it will provide convenience to daily basic needs.

Improving Resume Parser

Currently the parser can parse the resume dispensing upon the structure of the resume. We plan to improve the efficiency of resume parsing through different DL/NLP based techniques.

Pricing

We could integrate the payment gateway to our application and provide the customized features according to plan the users choose.

REFERENCES

- [1] Heric, M. (2018). HR new digital mandate. Digital technologies have become essential for HR to engage top talent and add value to the business. Retrieved August, 20, 2019.
- [2] Ma, H., Xu, Y., Ma, W., Lin, Z., Jiang, K. (2020, December). A Muti-Field Feature Interaction Convolutional Neural Network for Resume Recommendation. In 2020 International Symposium on Autonomous Systems (ISAS) (pp. 186-191). IEEE.
- [3] Nimbekar, R., Patil, Y., Prabhu, R., Mulla, S. (2019, December). Automated Resume Evaluation System using NLP. In 2019 International Conference on Advances in Computing, Communication and Control (ICAC3) (pp. 1-4). IEEE.
- [4] Yan, R., Le, R., Song, Y., Zhang, T., Zhang, X., Zhao, D. (2019, July). Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (pp. 914-922).
- [5] Bahdanau, D., Cho, K., Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- [6] Cavaliere, L. P. L., Nath, K., Wisetsri, W., Villalba-Condori, K. O., Arias-Chavez, D., Setiawan, R., ... Regin, R. (2021). The Impact of E-Recruitment and Artificial Intelligence (AI) Tools on HR Effectiveness: The Case of High Schools (Doctoral dissertation, Petra Christian University).
- [7] Kochling, A., Wehner, M. C. (2020). Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development. *Business Research*, 1-54.
- [8] Chou, Y. C., Chao, C. Y., Yu, H. Y. (2019, February). A Resume Evaluation System Based on Text Mining. In 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIC) (pp. 052-057). IEEE. 58
- [9] Yoshua Bengio. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, Vol. 2, 1 (2009), 1–127.
- [10] Yu Cheng, Yusheng Xie, Zhengzhang Chen, Ankit Agrawal, Alok Choudhary, and Songtao Guo. 2013. JobMiner: A Real-time System for Mining Job-related Patterns from Social Media. In

Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13). ACM, 1450–1453.

[11] Wael H. Gomaa and Aly A. Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications*, Vol. 13, 68 (2013), 13–18.