

STREAMLINING JOB SEARCH

**DOMAIN: JOB SEARCH AND RECRUITMENT
OPTIMIZATION**

By-
Tejaas Mukunda Reddy
CWID: 886702844
CPSC-597 (Spring, 2024)
30th April 2024



Department of Computer Science

CPSC 597 / 598 PROJECT / THESIS DEFINITION

To the graduate student:

1. Complete a project proposal, following the department guidelines.
2. Have this form signed by your advisor and reviewer / committee.
3. Submit it with the proposal attached to the Department of Computer Science.

 Project Thesis

Please print or type.

Student Name: TEJAAS MUKUNDA REDDY Student ID: 886702844

Address: Street City Zip Code

Home Phone: 6575565985 Work Phone: _____

E-Mail: tejaas@csu.fullerton.edu Units: 03 Semester: Spring 2024

Are you a Classified graduate student?

 Yes

Is this a group project?

 Yes No No

Proposal Date: _____

Tentative Date for Demonstration _____

/Presentation/Oral Defense: _____

Completion Deadline: _____

Tentative Title: _____

We recommend that this proposal be approved:

Faculty Advisor

Printed name	Signature	Date
--------------	-----------	------

Faculty Reviewer

Printed name	Signature	Date
--------------	-----------	------

Faculty Reviewer

Printed name	Signature	Date
--------------	-----------	------

Abstract

In the dynamic landscape of the IT industry and computer science education, individuals invest considerable time, typically 1-2 hours daily, in the laborious process of job searching. This involves researching companies, crafting cover letters, and networking, with each job application taking 10-20 minutes. This inefficiency, coupled with challenges such as fraudulent job posts and skill misalignment, hinders candidate satisfaction, and widens the gap between candidate skills and industry demands.

This project proposes a solution centered on skill development, leveraging a global ranking system based on resumes and certifications to guide candidates' improvement. Real-time trend analysis minimizes skill acquisition time, enhancing competitiveness. The platform streamlines job-seeking by enabling candidates to express interest, receive global rankings, and connect directly with recruiters, thereby bypassing intermediaries and ensuring a secure recruitment process. Advanced algorithms facilitate efficient connections between recruiters and candidates with specific skill sets.

The incorporation of real-time trend analysis ensures candidates stay abreast of new technologies. Simultaneously, the project simplifies the application process by allowing candidates to indicate company preferences, generating a global ranking based on their domain accessible to recruiters. Recruiters can then filter based on domains or candidates interested in their hiring company, enabling direct outreach via email and eliminating the need for formal applications.

This project aims to revolutionize IT and software recruiting by prioritizing skill development, immediate trend analysis, and overcoming challenges associated with traditional job boards. By alleviating candidate burdens, reducing the risks of fake job postings, and aiding recruiters in optimizing their hiring strategy by finding the most suitable candidates for their domain, as well as filtering candidates based on those who have shown interest in working for their company, the proposed system seeks to provide a flexible and responsive platform meeting current needs while anticipating future dynamics through innovative features.

Table of Contents

<i>1. Introduction</i>	5
1.1 Problem Domain and Background.....	5
1.2 Current Solution	6
1.3 Proposed Solution	6
<i>2. Project Objectives</i>	7
<i>3. Project Activities</i>	8
3.1 Use Case Diagram.....	8
3.2 Sequence Diagram	10
3.3 Application Architecture.....	11
3.4 Hadoop Architecture	12
3.5 MongoDB Architecture	16
3.6 Flask API	17
3.7 Web Page Design.....	19
3.8 Activities	30
<i>4. Software Requirement Specifications (SRS)</i>	31
<i>5. Environment</i>	32
<i>6. Project Results</i>	35
<i>7. Project Schedule</i>	38
<i>8. Bibliography</i>	39
<i>9. Glossary</i>	40

1. Introduction

People in the IT industry and graduating computer science students are confronted with various problems when seeking jobs opportunities as the software and IT industry dedicated to recruiting is evolving at an accelerated rate as more diverse jobs are available to an increasing number of applicants. Job hunting requires numerous hours of commitment, including cases of fraudulent job posts and misaligned skills. This inefficiency impedes the candidate's satisfaction and increases the gap between the skills of potential candidates and the demand of the industry. The context of these issues clearly shows that these gaps need to be closed to improve hiring for both recruiters and job seekers.

1.1 Problem Domain and Background

The current recruitment process takes significant time and is stressful for candidates as they apply for many jobs and receive fewer responses as compared to 2022. Only 36% of candidates are satisfied with the hiring process, according to Indeed. In August 2022, Indeed surveyed 100 IT hiring managers, revealing the top 10 lacking skills in IT interview candidates: self-assurance, enthusiasm for IT, questioning ability, problem-solving, soft skills, professionalism, business acumen, and interview question responses, emphasizing the need for skill development.

The IT industry is the most targeted industry for job scams, according to the Federal Trade Commission. (2024, February). Job scams in the IT industry continue to proliferate, deceiving aspiring professionals and costing them financially, highlighting the need for effective measures to detect and prevent these fraudulent schemes. Many students lack awareness of the latest technology trends, hindering their ability to adapt and thrive in our rapidly evolving digital world. According to National Skills Coalition and partners (Bergson-Shilcock et al., 2023) only 23% of US adults have the basic digital skills needed for most jobs in today's economy.

Newly graduated students spend 10-20 hours a week finding a job, according to the Indeed (2023). However, students face many constraints, such as schoolwork, part-time jobs, and extra-curricular activities that limit their time. Only some of those hours are available for committing. This could affect an individual's interview skills. Finding a job is hard, and more time does not always translate to more interviews. Remarkably, despite a clear perception of fake job advertisements among many applicants, almost half of them admit that they have not heard a word about this deceit during their job search on the internet. Many job hunters who lack awareness put themselves in danger of having their identity stolen for nefarious reasons, necessitating more awareness of the situation.

In summary, the modern recruitment procedures impose major difficulties on candidates. Student's poor awareness of changing technology trends coupled with time constraints of a typical job search create additional worries concerning their ability to adjust to the emerging digital realm. The search for a job itself usually requires much time so it is important to manage it properly and know about fraud advertisements for jobs which makes a more transparent and effective hiring market. Moreover, there is a job scam that occurs often in the IT industry which makes it even more necessary to implement appropriate techniques of screening and detection.

1.2 Current Solution

These days, job websites such as LinkedIn, Indeed, and Glassdoor provide an efficient way for companies to advertise vacancies and for people to search for and apply for jobs. It further allows a direct connection between job hunters and hiring teams, thereby making hiring easier.

But there are still problems in the existing system. People looking for jobs mostly spend much time examining long job descriptions, aiming to pinpoint what skills businesses need. This requires them to work hard to gain the necessary skills before applying for roles. Moreover, problems noted in the problem statement, such as the time-consuming nature of each application and the prevalence of job scams on these boards, continue to hinder the efficiency of the job-seeking process.

Surprisingly, some of the job boards can present outdated or even non-existent job openings. Therefore, employers who post positions can receive applications from people searching for jobs when the position does not even exist anymore. This complicates their search for jobs and makes it more annoying to them. Additionally, there is no all-inclusive platform for monitoring the current trending technological in the present job market. This is where individuals may find it difficult in trying to match the evolving skills needed for jobs with their own skills development.

I aim to introduce a new approach to the recruitment process. This approach prioritizes candidates and students working on enhancing their skills, while recruiters focus on finding and hiring suitable candidates.

1.3 Proposed Solution

In comparing our proposed solution with current advances in job boards, it becomes evident that traditional platforms have inherent limitations. The conventional model relies on job seekers to proactively identify and acquire the skills demanded by employers and apply for each job individually, resulting in a time-consuming and often inefficient process. The persistence of fraudulent job postings and the inability to ensure real-time accuracy in the job listings further amplify the challenges faced by both candidates and recruiters.

Our proposed solution acknowledges these limitations and aims to usher in a paradigm shift by addressing the root causes of inefficiencies in the current system.

Key Features of Our Proposal

- Skill Development Focus: Our system adopts a progressive approach towards skill enhancement by giving the candidates a global ranking based on their resumes and certifications. The candidates will be ranked based on their skills. This ranking will show the candidates which skill they need to improve on and for which job. Additionally, by utilizing real-time trend analysis of the current “market” skills, we will help candidates direct their attention to upcoming skills.

- Real-time Trend Analysis: Our platform constantly analyzes trends in the IT and software industry. This form of real-time analysis prepares the candidates for the latest skills required by several employers and, consequently reduces the acquisition of the skill while increasing competitiveness in the job market.
- Time Efficiency: Firstly, our solution will significantly reduce the amount of time needed to be invested by job seekers as they do not have to send an application for each job manually now. They can express interest in a few companies and work on the current trending technologies, and once they upload the resume, we will give them a global ranking using which the recruiters can contact candidates directly via email.
- Risk Mitigation: Our platform connects employers directly with job seekers as part of the efforts to address issues such as fraudulent job ads. Our service is different from traditional methods where jobs are normally posted, and candidates apply because it bypasses this middleman, guarding against any possible scams. We greatly narrow down the chances of fraud by making the recruitment process lean. This ensures a safe and sound work atmosphere for all parties involved.
- Efficient Recruiter-Candidate Connections: We have also made it worthwhile for recruiters by giving them an efficient way of linking up with the right candidates. With advanced algorithms and data-driven insights, recruiters would be able to target specific individuals who exactly fit the required skill sets for some positions in an organization.

Revolutionizing IT and Software Industry Recruitment

Our proposition will try to revolutionize the current recruiting scenario in the IT and software domain. With an emphasis on skill development, immediate trend analysis, and overcoming the pitfalls related to conventional job boards, our system seeks to spare candidates unnecessary hassle, decrease the hazards of fake employment posts, and assist recruiters in optimizing their hiring strategy. The aim is to offer a flexible platform responsive to today's needs while staying ahead of tomorrow's dynamics through such innovations.

2. Project Objectives

This project is aimed at addressing the main purpose that includes skills development, efficient recruitment, and ensuring transparency in candidate—recruiter relations in the IT/software industry. This has resulted in the current dissatisfaction amongst the jobseekers, mismatch in the skill set between what the industry demands and the actual one and an increased number of fraudulent jobs (scams).

Real-time trend analysis will be our project's pathway to keep candidates updated with new technologies. At the same time, we want to simplify the application procedure and let applicants indicate which companies or domains excite them and then rank the candidates based on their resume, generating a global ranking to which the recruiter shall have access, enabling them to filter based on domains or

candidates who have expressed interest in the recruiter's current hiring company. This feature empowers recruiters to reach out to candidates directly via email, eliminating the need to submit formal applications and saving much time.

The significance of this project lies in its challenging task of building a complex web-based platform incorporating various components and frameworks:

- Frontend: Angular framework for user interface development.
- Backend API: Flask framework for building APIs to integrate the frontend and backend.
- Big Data Technologies: Hadoop and Apache Kafka for data storage and processing.
- Neural Network Models: OneVsRest Classifier and KNN models to analyze resumes and rank candidates.
- Web Scraping: Beautiful Soup to extract data from job boards like LinkedIn, identifying current industry trends.

The complexity of the system architecture and the potential impact of the solution makes this project a suitable and significant endeavor for a Master's project.

3. Project Activities

The following sections provides an overview of the project activities conducted to develop the project.

3.1 Use Case Diagram

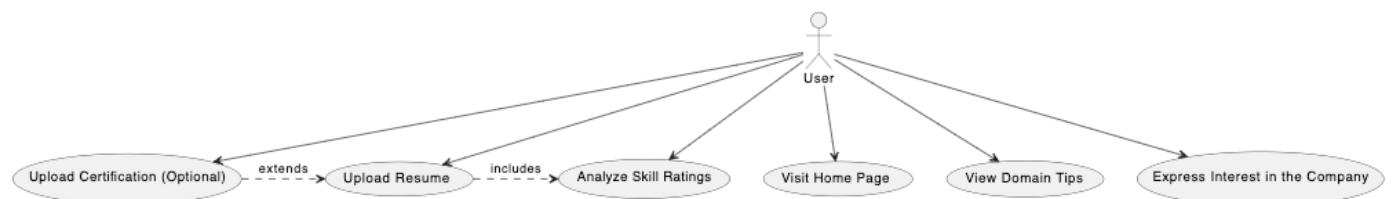


Fig 3.1.1 Use Case Diagram For User/Candidate

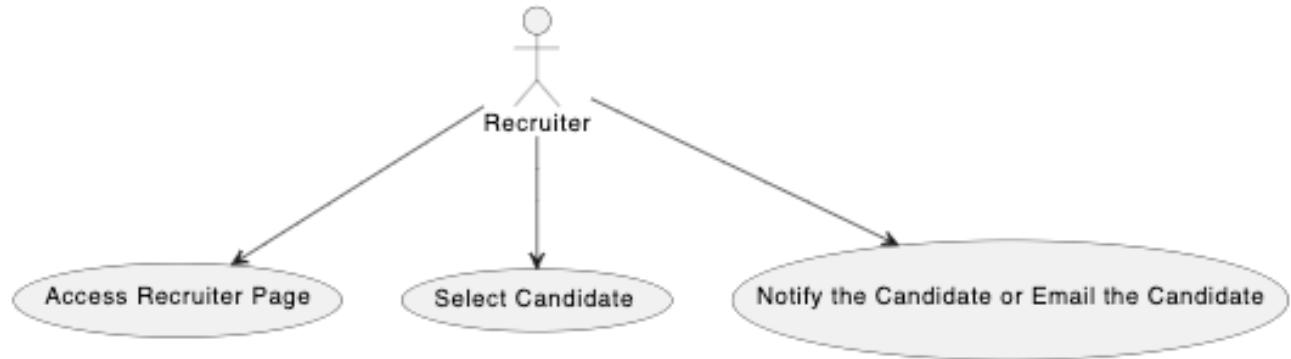


Fig 3.1.2 Use Case Diagram For Recruiter

Note: I am considering naming my website "**Tech Trend.**"

Figure 3.1.1 and 3.1.2 represents the use case diagram which shows how the users will interact with the system to accomplish specific tasks and goals.

From Candidate/User Perspective

- The User visits the Tech Trend Home Page and logs in.
- Tech Trend fetches the current trending technologies data from the job board using web scraping.
- Tech Trend returns the current trending technologies data to the User.
- The Users upload their resumes to Tech Trend application.
- Tech Trend analyzes the User's resume and displays Domain and ranking.
- The Users express interest in a company.
- Tech Trend saves the User's company selection.
- Tech Trend notifies the company's recruiters.
- The Users upload their certification to Tech Trend.
- Tech Trend validates the certification.
- Tech Trend saves the certification.
- The User emails a recruiter.

From Recruiter Perspective

- The Recruiter visits their Recruiter Page and logs in.
- Tech Trend displays candidate profiles to the Recruiter.
- The Recruiter filters candidates by domain.
- The Recruiter filters the candidates based on ranking.
- Tech Trend applies the filter and displays the candidates to the Recruiter.
- The Recruiter selects a candidate and TechTrend displays the email of the candidate.
- The Recruiter emails the candidate.

3.2 Sequence Diagram

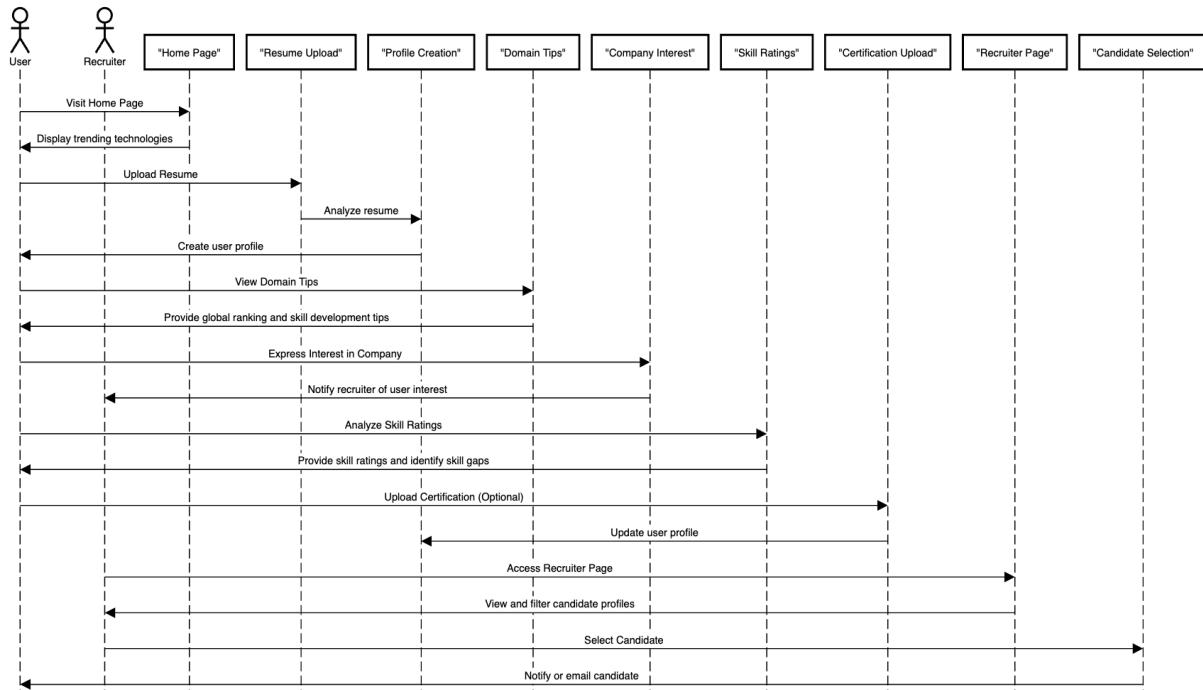


Fig 3.2.1 Sequence Diagram

Figure 3.2.1 represents the sequence diagram which shows the interaction between users and system components in chronological order.

- Step 1: Visit Home Page
 - Description: The User begins by accessing the website's home page. Prominently displaying current trending technologies to capture attention and highlight in-demand skills.
- Step 2: Upload Resume
 - Description: The User uploads their resume for analysis and profile creation.
- Step 3: View Domain Tips
 - Description: Analyzing the resume, the website offers global ranking and score to enhance the user's job search strategy and skill development.
- Step 4: Express Interest in the Company
 - Description: The User shows interest in a particular company, signaling to initiate potential connections with the recruiter.

- Step 5: Analyze Skill Ratings
 - Description: Tech trend analyzes the User's skills based on their resume and provides ratings for clarity and the identification of potential skill gaps.
- Step 6: Upload Certification (Optional)
 - Description: The User has the option to upload certifications to strengthen their profile further and demonstrate expertise.
- Step 7: Recruiter Page (Recruiter View)
 - Description: Recruiters access their dedicated page to view and filter candidate profiles based on ranking and specific criteria.
- Step 8: Select Candidate
 - Description: The Recruiter selects a candidate they're interested in to explore their full profile and qualifications.
- Step 9: Notify the Candidate or Email the Candidate
 - Description: The Recruiter either notifies or directly emails the candidate to initiate the hiring process.

3.3 Application Architecture

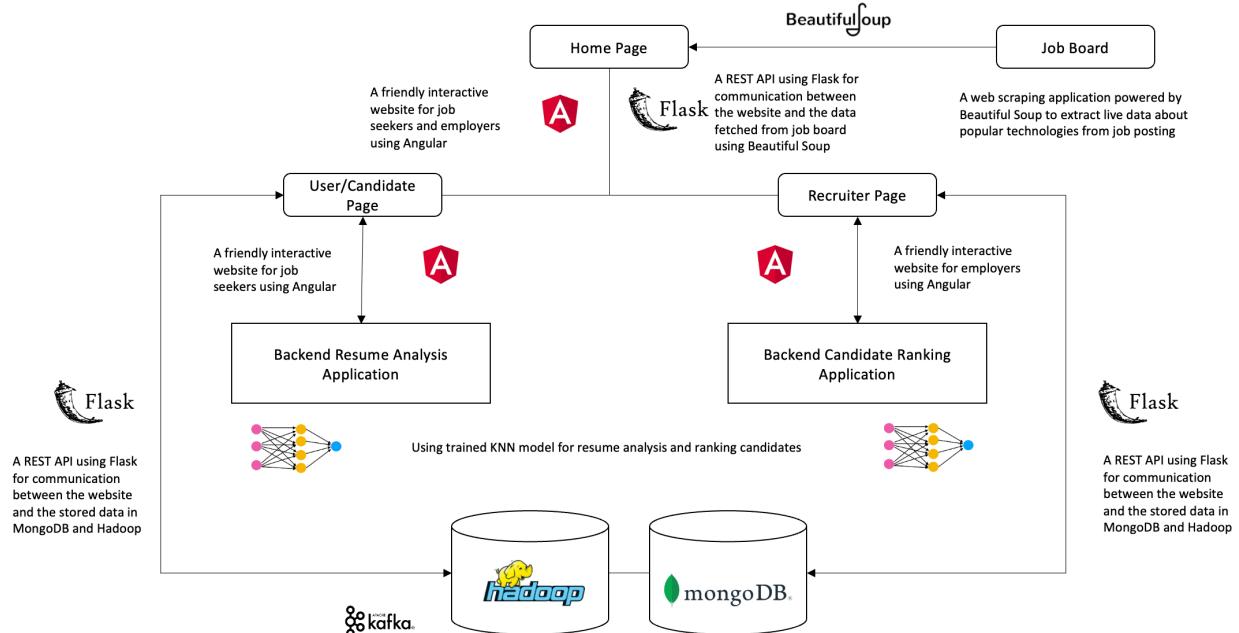


Fig 3.3.1 Application Architecture

Figure 3.3.1 represents the application architecture, which showcases the visual representation of the deployment and utilized technologies.

Architecture Flow:

- The homepage of the platform will feature the latest trending technologies sourced from data collected through the Job Board using Web Scraping technologies like Beautiful Soup.
- Two distinct links will be available on the homepage—one for candidate/user to log in and another for recruiters.
- Users will have access to various functionalities, including the ability to upload their resumes.
- The back-end resume analysis application will then assess the resumes, providing feedback and assigning rankings based on the candidates' domain expertise using a OneVsRest Classifier and KNN Model.
- Recruiters, on the other hand, will be able to utilize the back-end candidate ranking application. This application ranks candidates and presents them to recruiters based on their respective rankings.
- All data, including user profiles, resumes, and rankings, will be stored, and retrieved from the Hadoop database, ensuring robust and scalable data management.
- The frontend application will be built using Angular and will be linked to the backend using REST API built using Flask.

3.4 Hadoop Architecture

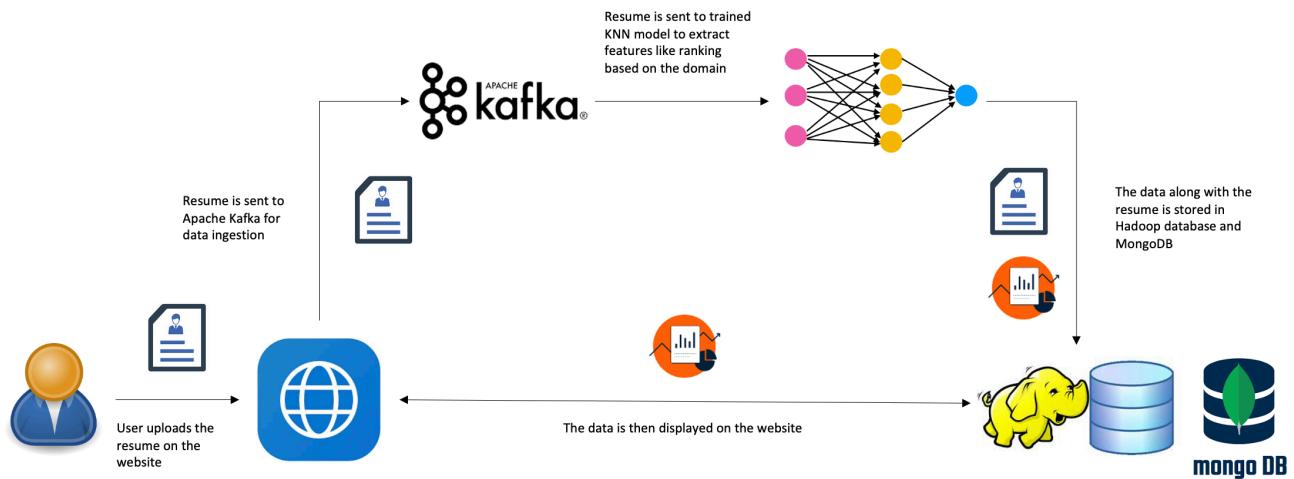


Fig 3.4.1 Hadoop Architecture For Resume Analysis

Figure 3.4.1 showcases the flow of architecture when a user uploads a resume, proceeds through the ML model, and then gets stored in the Hadoop database.

Hadoop is currently configured on my local machine, comprising of one Name Node and one Data Node. Apache Kafka is set up to ingest resumes or certifications uploaded by users into Hadoop. Zookeeper is essential for Kafka to coordinate distributed processes.

Architecture Flow:

- The user initiates the process by uploading a resume from the user page.
- The resume is sent to Apache Kafka for data ingestion.
- The resume undergoes parallel data processing across multiple systems using Apache Spark, using the trained OneVsRest Classifier and KNN models.
- This process extracts features such as candidate ranking based on the domain.
- The processed data, along with the original resume, is stored in the Hadoop database.
- Finally, the data and the resume are presented and displayed on the website for the users.

Datanode Information

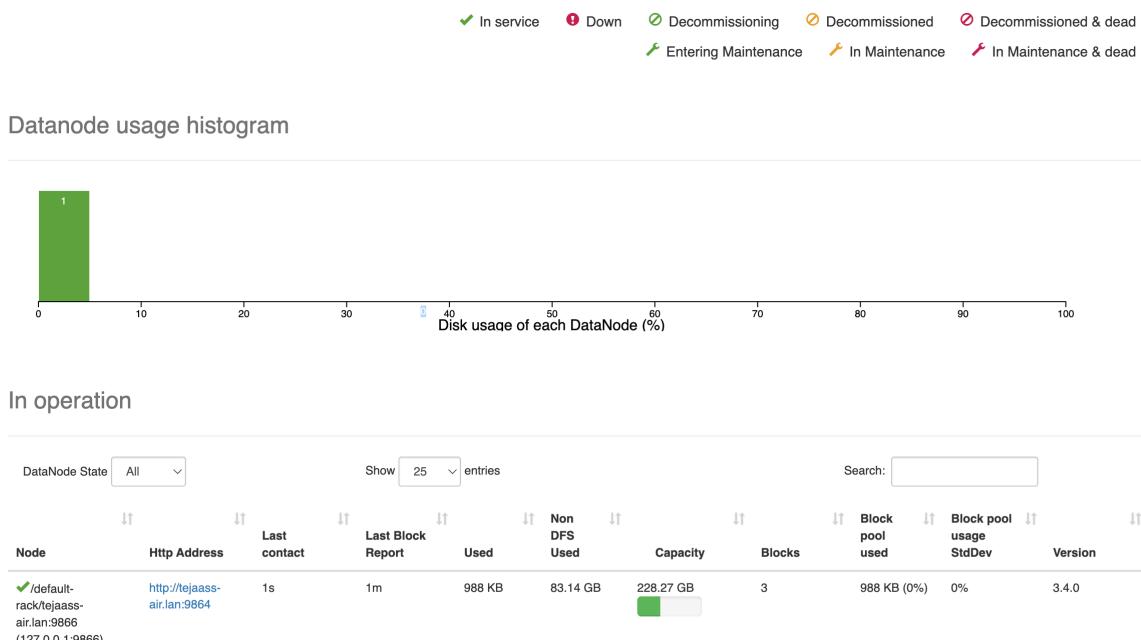


Fig 3.4.2 Hadoop Data Node Architecture

Figure 3.4.2 represents the storage structure within the Hadoop database's Data Node. The Hadoop Data Node usage screen displays a histogram visualizing disk usage distribution across cluster nodes. Additionally, it indicates the current status of the each node, which currently indicates that no Data Nodes are unavailable or undergoing maintenance.

This allows administrators to monitor storage capacity and identify potential imbalances or hotspots in the cluster. Moreover, the screen provides real-time status updates, indicating that all nodes are currently operational.

Hadoop	Overview	Datanodes	Datanode Volume Failures	Snapshot	Startup Progress	Utilities ▾
--------	----------	-----------	--------------------------	----------	------------------	-------------

Overview 'localhost:9000' (✓active)

Started:	Thu Apr 25 19:06:12 -0700 2024
Version:	3.4.0, rdb8b77f398f626bb7791783192ee7a5dfaeecc760
Compiled:	Sun Mar 03 22:35:00 -0800 2024 by root from (HEAD detached at release-3.4.0-RC3)
Cluster ID:	CID-7d3aab26-edc2-42e7-a777-2f637082bf2a
Block Pool ID:	BP-887500391-127.0.0.1-1713731885435

Summary

Security is off.
 Safemode is off.
 7 files and directories, 3 blocks (3 replicated blocks, 0 erasure coded block groups) = 10 total filesystem object(s).
 Heap Memory used 98.13 MB of 154 MB Heap Memory. Max Heap Memory is 2 GB.
 Non Heap Memory used 63.41 MB of 66.75 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	228.27 GB
Configured Remote Capacity:	0 B
DFS Used:	988 KB (0%)
Non DFS Used:	83.15 GB
DFS Remaining:	145.13 GB (63.58%)
Block Pool Used:	988 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Thu Apr 25 19:06:12 -0700 2024

Fig 3.4.3 Hadoop Database Architecture

Figure 3.4.3 presents an overview of the Hadoop database currently hosted on my local machine, accessible via port 9000. It furnishes details such as:

- Version: 3.4.0
- Cluster ID: CID-7d3aab26-edc2-42e7-a777-21637082bf2a
- Block Pool ID: BP-887500391-127.0.0.1-1713731885435
- Configured Capacity: 228.27 GB
- DFS Used: 988 KB (0%)
- Live Nodes: 1
- Dead Nodes: 0

To establish the Hadoop cluster and activate Apache Kafka for data ingestion, as well as Apache Zookeeper for resource management, follow these commands.

1. # start-all.sh
 - a. The following services are initiated:
 - Data Node: Responsible for data storage within a Hadoop cluster node.
 - JPS: Java Virtual Machine Process Status Tool to facilitate the listing of Java processes.
 - Secondary Name Node: Executes periodic checkpoints for the Name Node in Hadoop.
 - Resource Manager: Oversees resource management and application scheduling.
 - Name Node: Manages the file system namespace and metadata for HDFS within Hadoop.
 - Node Manager: Manages resources and containers on a node.
2. # zookeeper-server-start.sh \$KAFKA_HOME/config/zookeeper.properties
 - a. Initiates the Zookeeper server crucial for Kafka to synchronize distributed processes.
3. # kafka-server-start.sh \$KAFKA_HOME/config/server.properties
 - a. Launches the Kafka server, following the designated properties of configuration.
4. # kafka-topics.sh --create --topic resume_upload_topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1
 - a. Establishes a "resume_upload_topic" Kafka topic with specified replication factor and configuration parameters.
5. # kafka-console-producer.sh --topic resume_upload_topic --bootstrap-server localhost:9092
 - a. Initiates a console-based producer, allowing messages to be published to the Kafka topic "resume_upload_topic".
6. # kafka-console-consumer.sh --topic resume_upload_topic --bootstrap-server localhost:9092 --from-beginning
 - a. Starts a consumer that makes it easier to retrieve messages from the "resume_upload_topic" Kafka topic right from the start.
7. # stop-all.sh
 - a. To stop all the services.

You can terminate other services such as Apache Kafka and Apache Zookeeper by using the command + c or ctrl + c.

These steps will ensure a robust infrastructure for data handling and processing using Big Data Technologies like Apache Hadoop, Apache Kafka, Apache Zookeeper.

3.5 MongoDB Architecture

The screenshot shows the MongoDB Atlas web interface. On the left, there's a sidebar with sections for Project 0, Deployment, Database (selected), Services, Security, and Goto. The main content area is titled 'Cluster0' and shows the 'techtrend' database with the 'techtrend_user' collection. It provides details like storage size (572KB), logical data size (997.39KB), total documents (15), and index size (0B). Below this, there's a preview of 15 documents and a pipeline editor. The top right shows the version (7.0.8) and region (AZURE Virginia-East2 (eastus2)).

Fig 3.5.1 MongoDB Architecture

Figure 3.5.1 represents the architecture of MongoDB database which is currently hosted on the Microsoft Azure cloud platform. It is a NoSQL database product that utilizes JSON-like documents with optional schemas to store the data. Steps to connect to the database using Python and perform operations using Flask API.

The screenshot shows the MongoDB Charts interface. On the left, there's a sidebar with a 'Data Source' section set to 'techtrend_user' and a 'FIELDS' section listing various document fields. The main area is titled 'Stacked Bar' and shows a preview of the chart. A large number '15' is displayed prominently. To the right, there are three 'Suggested Charts' cards: 'Grouped Combo' (X-axis: # feature_vectors, Y-axis: A_id Count), 'Grouped Bar' (X-axis: A_id, Y-axis: # feature_vectors), and another 'Grouped Bar' card below it. At the bottom right is a circular icon with a gear and a play button.

Fig 3.5.2 MongoDB Architecture Graph

Figure 3.5.2 provides a visual representation of the data stored within the MongoDB database. From the image, it's evident that there are 15 records. Furthermore, supplementary information including fields and the name of the collection is also provided. Fetch the database connection URL by entering your cluster home, database name, password, and username which you can find in your MongoDB database profile.

URL for connecting to database.

- `atlas_uri = "mongodb+srv://username:password@cluster-hostname/database-name?retryWrites=true&w=majority&ssl_ca_certs=/path/to/cafile.pem"`

Connect to the MongoDB Atlas Cluster

- `client = MongoClient(atlas_uri)`

Connect to the 'techtrend' database and 'techtrend_user' collection.

- `db = client.techtrend`
- `collection = db.techtrend_user`

There are two users: one is the candidate, and one is the Recruiter.

For User/Candidate you can find the below fields:

- `_id`: A unique ID randomly generated by the MongoDB database.
- `user_name`: Stores the username of the user/candidate.
- `password`: Stores the password.
- `contact_info`: Stores the email of the candidate.
- `category`: Categorizes the resume.
- `rank`: Indicates the rank of the user/candidate.
- `feature_vectors`: Stores the feature vectors of the resume for candidate ranking.
- `user`: Identifies the type of user, either 'user' or 'recruiter'.
- `interest`: Captures the candidate's expressed interest in working for the company.

For Recruiter you can find the below fields:

- `_id`: A unique ID randomly generated by the MongoDB database.
- `contact_info`: Stores the email of the recruiter.
- `rank`: A random value assigned to the recruiter as they don't require a rank.
- `feature_vectors`: A null value as recruiters do not need feature vectors.
- `user_name`: Stores the username of the recruiter.
- `password`: Stores the password.
- `category`: Default value as recruiters haven't uploaded any resumes.
- `user`: Identifies the type of user, either 'user' or 'recruiter'.
- `company`: Stores the company the recruiter is hiring for.

3.6 Flask API

To create my API and facilitate communication between my Python-developed frontend and database, I used Flask. Postman has also been helpful in testing and improving the functionality of the API.

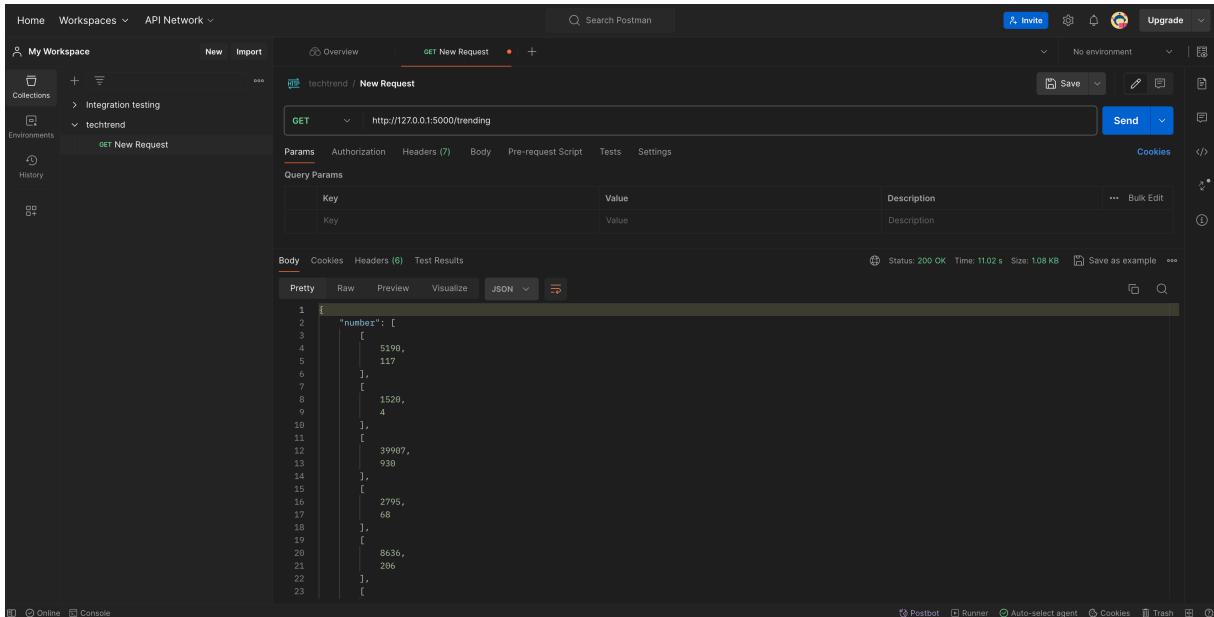
A screenshot of the Postman application interface. The left sidebar shows 'My Workspace' with a collection named 'Integration testing' containing a folder 'techtrend' and a 'GET New Request'. The main workspace shows a 'GET' request to 'http://127.0.0.1:5000/trending'. The 'Body' tab displays a JSON response with a single key 'numbers' containing an array of integers. The array elements are: 5198, 117, 1520, 4, 39987, 930, 2795, 68, 8636, 206. The status bar at the bottom indicates 'Status: 200 OK'.

Fig 3.6.1 Postman API Testing

You can see the result of the GET operation for the URL in the attached image. Real-time trending technologies are retrieved by this procedure from job platforms such as LinkedIn. The following functions have been implemented.

- trending: Flask route for fetching current trending technologies.
- fetch_rank: Flask route for fetching rank.
- fetch_company: Flask route for fetching company details.
- fetch_interest: Flask route for fetching candidate interest.
- fetch_user_details: Flask route for fetching user details for recruiter page.
- company_interest: Flask route for updating company interest.
- update_company: Flask route for updating company details.
- filter_candidates: Flask route for filtering candidates based on domain.
- upload_resume: Flask route for uploading a resume and fetching rank for the user.
- upload_certificate: Flask route for uploading a certificate.
- create_user: Flask route for creating a new user.
- login_user: Flask route for user/candidate login.
- recruiter_login: Flask route for recruiter login.
- get_user: Flask route for getting user data.
- update_user: Flask route for updating user details.
- delete_user: Flask route for deleting a user.

3.7 Web Page Design

This section will offer an elaborate overview of the features accessible on the homepage, recruiter page, and candidate page. Through the images of the webpage and explanations of functions and workflow, you'll gain a comprehensive understanding of the website's structure.

3.7.1 Home Page

This section will provide an overview of features available on the website's homepage.

The Home page will have the following sections:

- User/Candidate Login
- Recruiter Login
- List of top current trending technologies
- Graph showing current technology trends based on data collected from job boards like LinkedIn.

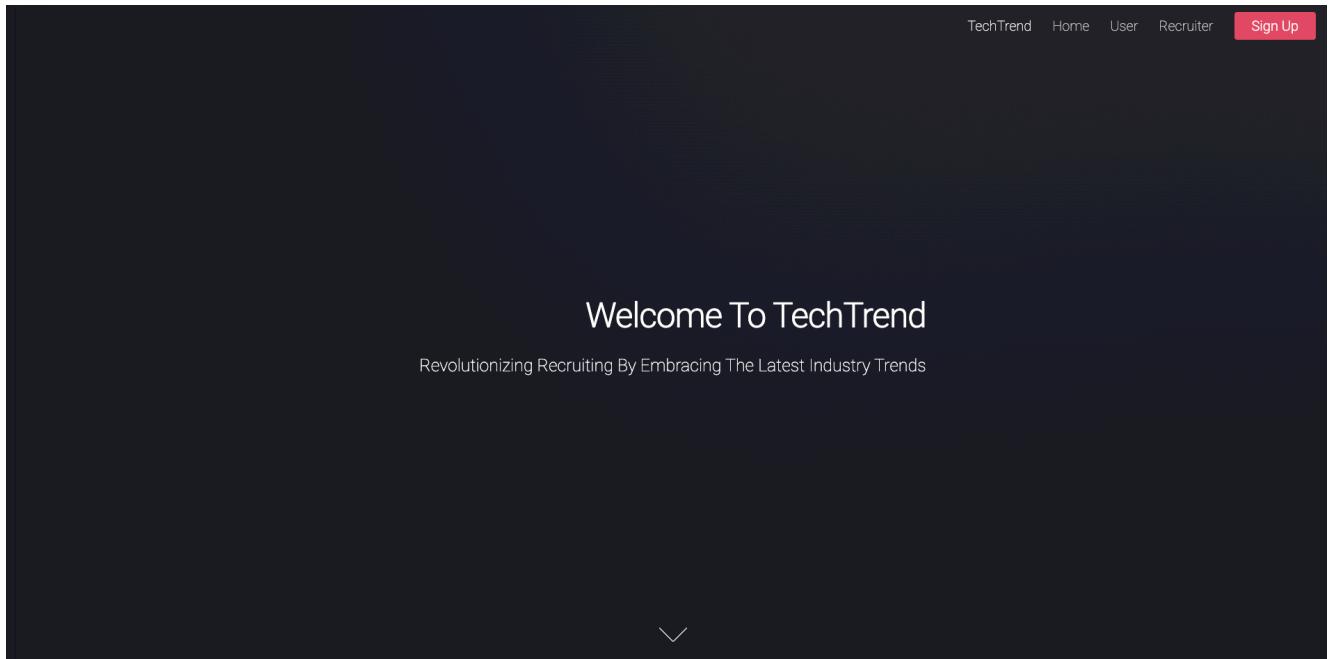


Fig 3.7.1.1 Cover Page Section

- Figure 3.7.1.1 represents the cover page which welcomes the user, showing the website name “TechTrend” and what it aims to achieve.
- There are three links on the navigation bar: one for navigating back to the homepage, one for the user login page, and one for the recruiter page.
- There is a signup link for new users or recruiters.

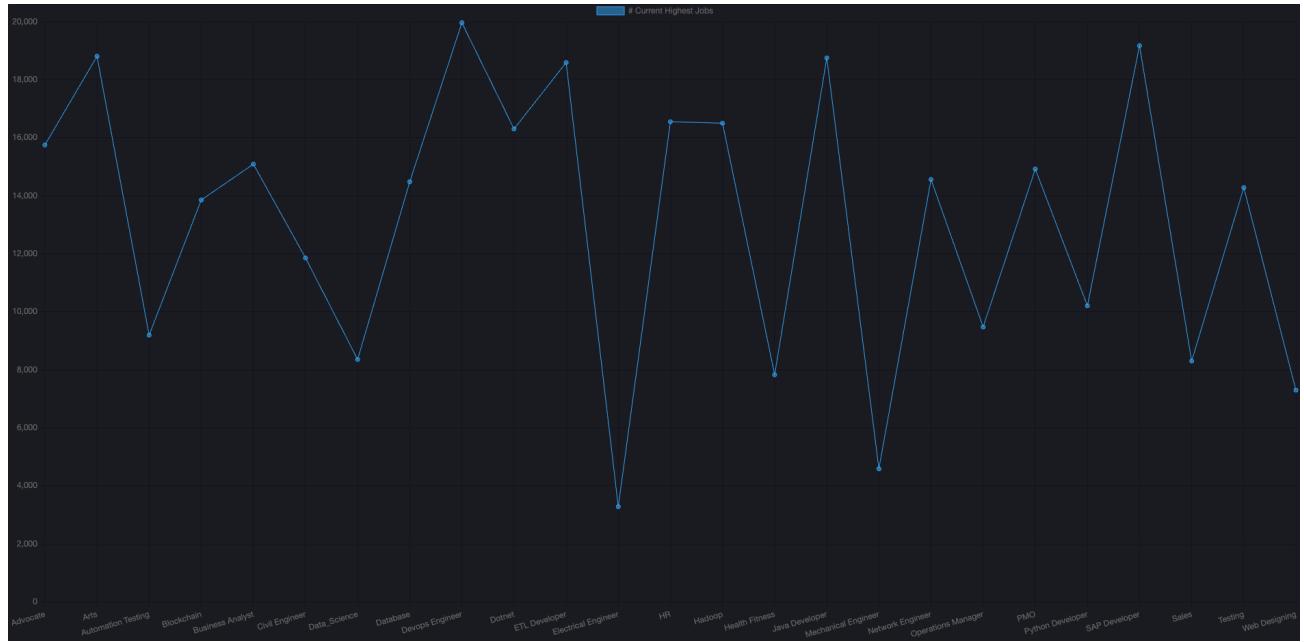


Fig 3.7.1.2 Current Highest Jobs Section

- Figure 3.7.1.2 represents the graph section which shows the current number of jobs in the United States based on domains fetched in real-time from job boards like LinkedIn.

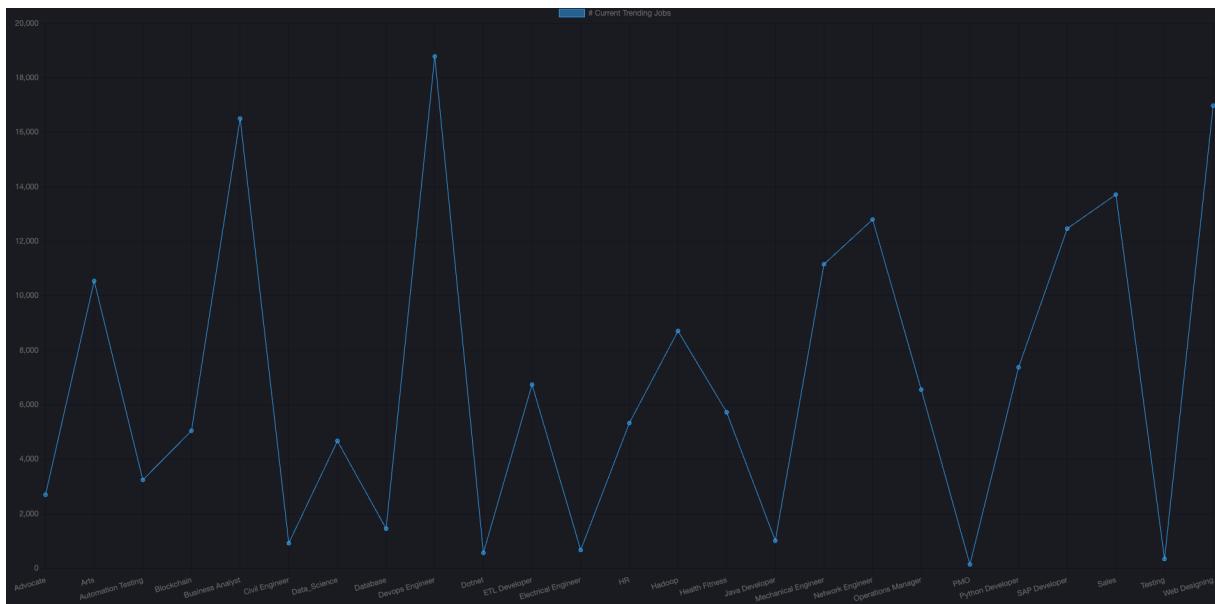


Fig 3.7.1.3 Current Trending Jobs Section

- Figure 3.7.1.3 represents the graph section which shows the latest jobs in the United States on that day based on domains fetched in real-time from job boards like LinkedIn.

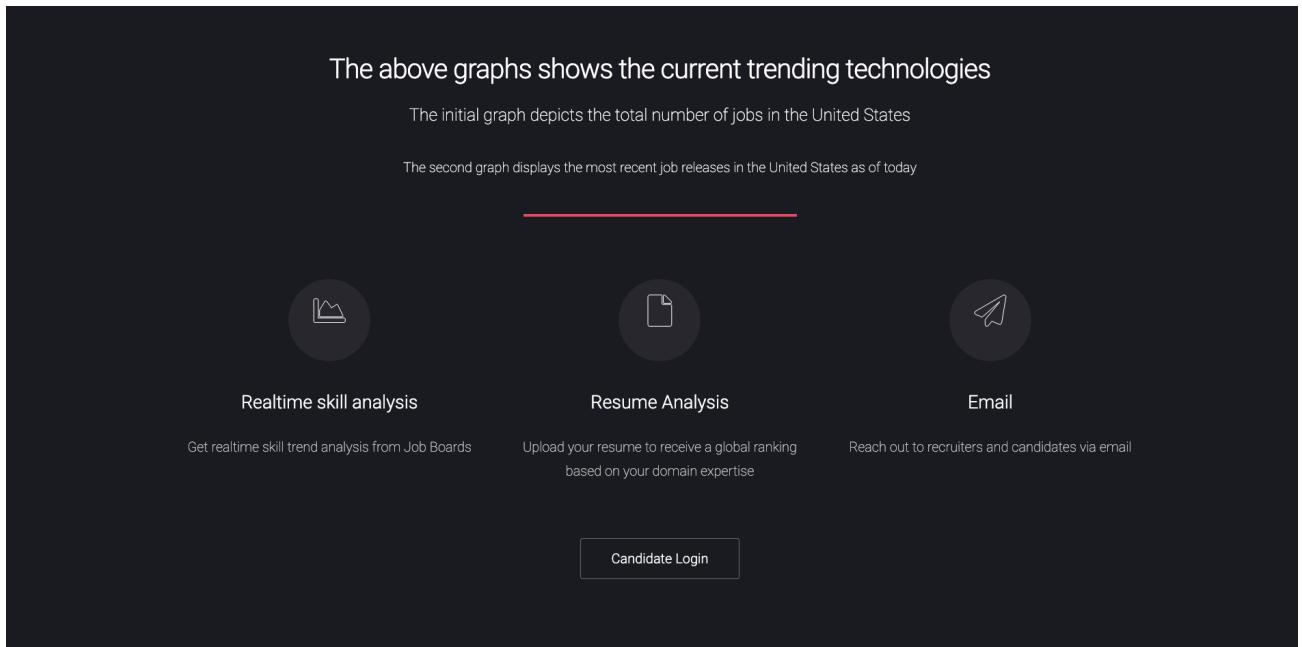


Fig 3.7.1.4 Features Section

- Figure 3.7.1.4 illustrates the website's offerings, such as real-time skill analysis, resume analysis, and email functionality.

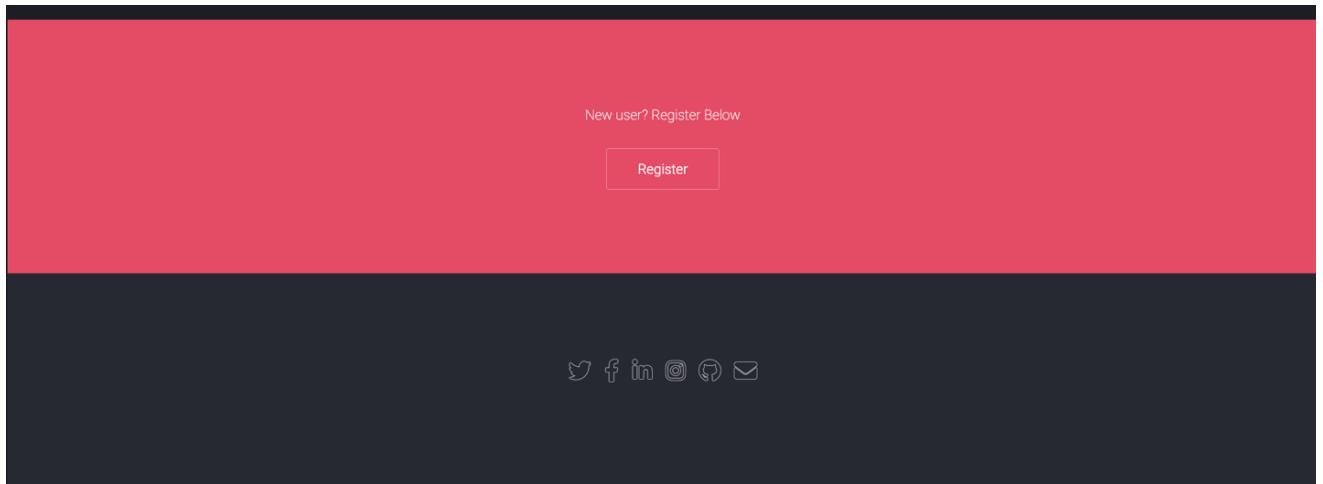


Fig 3.7.1.5 Registration Link Section

- Figure 3.7.1.5 showcases the registration link for both recruiters and candidates offering a convenient access point for individuals from both groups to sign up.

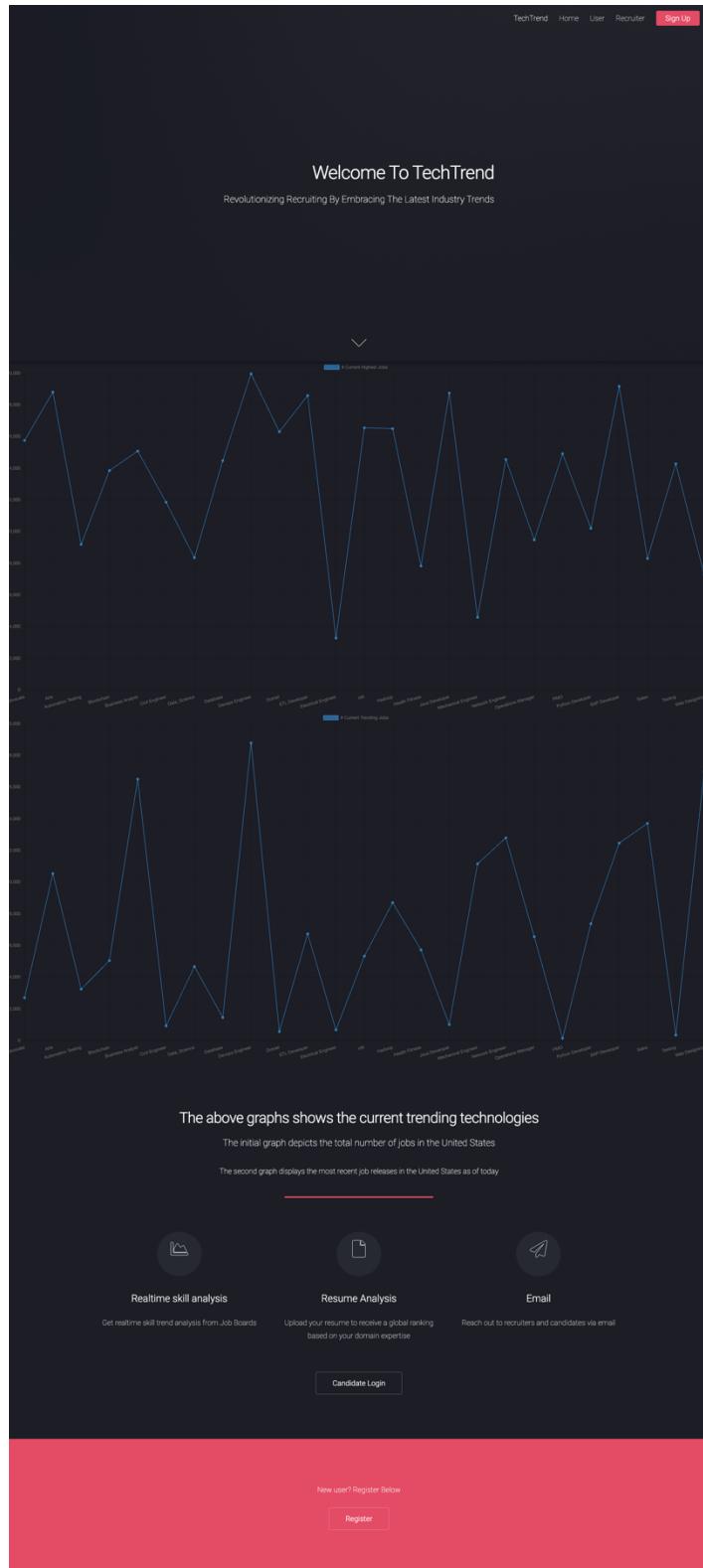


Fig 3.7.1.6 Home Page

- Figure 3.7.1.6 showcases the home page of my website.

3.7.2 Candidate Page

This section will provide an overview of features available on the candidates webpage, detailing the functionalities and tools available for use.

The Candidate page will have the following sections:

- Ranking section - to view their global ranking based on their skill rating.
- About me section - where they can mention their skill set along with their brief introduction
- Resume Review section - to upload a resume, the system will provide suitable domain tips.
- Certifications section - to upload their certifications.
- Company Interest section – to express interest in companies.

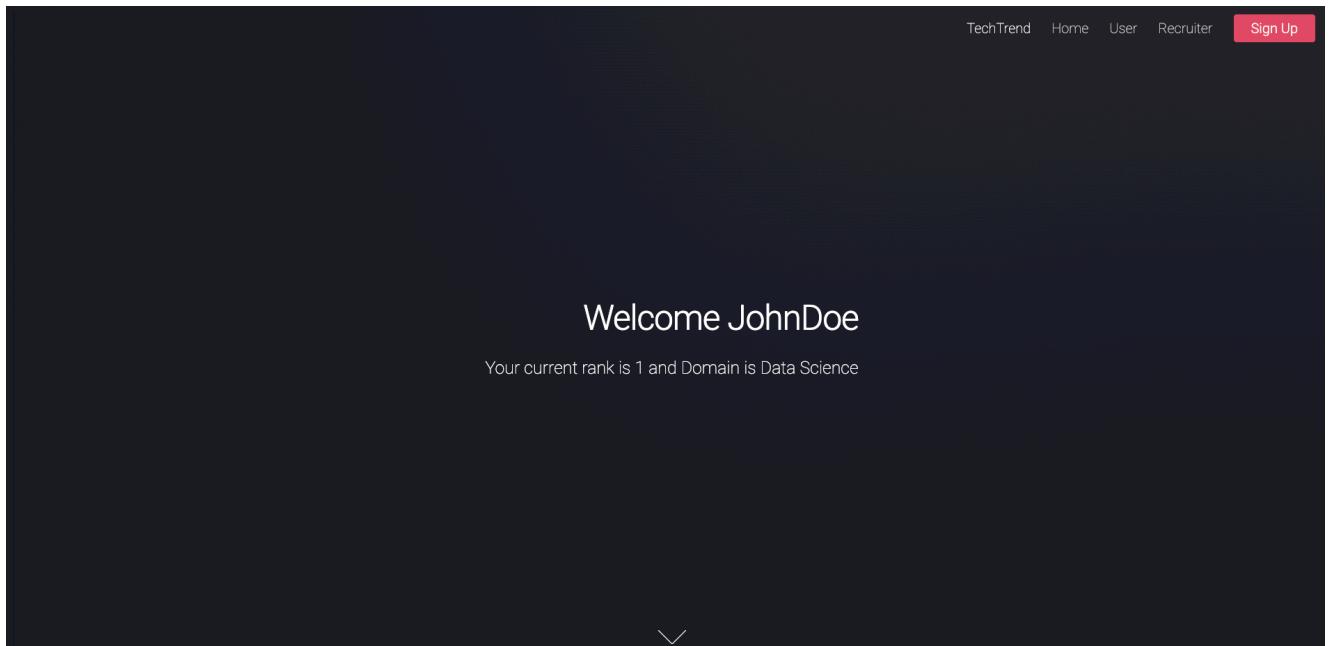


Fig 3.7.2.1 Cover Page Section

- In Figure 3.7.2.1, the candidate's name and current global rank are displayed, categorized by domain and domain name.
- The information is retrieved from the attached resume provided by the user.
- New users will be prompted to upload a resume.
- This action allows them to obtain their global ranking and determine the domain category to which their resume belongs.

Figure 3.7.2.2 showcases the candidate's name and options to upload the resume and certifications. It also showcases the companies the candidate is interested in.

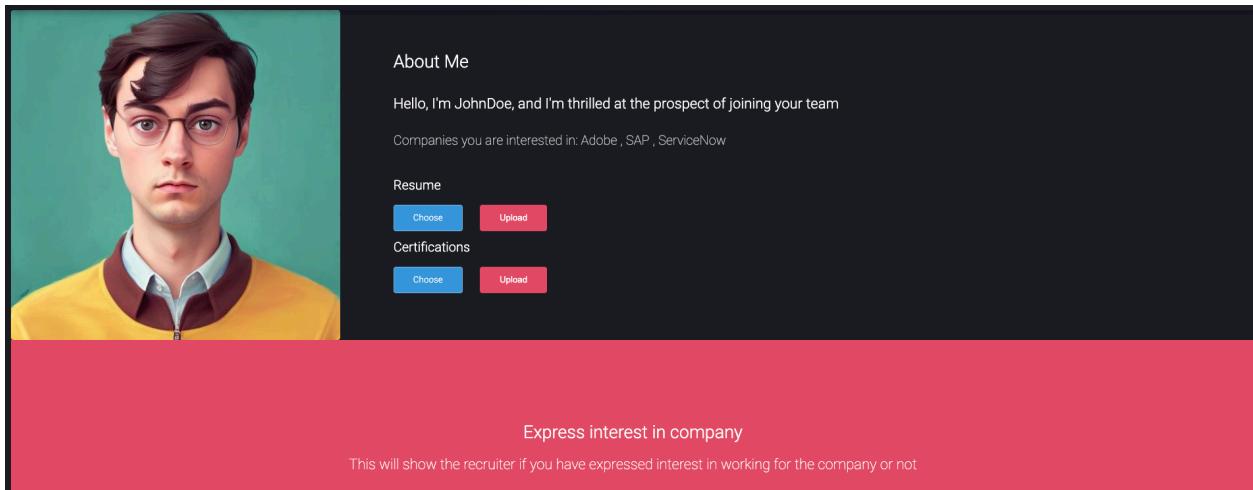


Fig 3.7.2.2 About Me Section

- Figure 3.7.2.3 showcases a section where the candidate can express interest in companies from the dropdown.

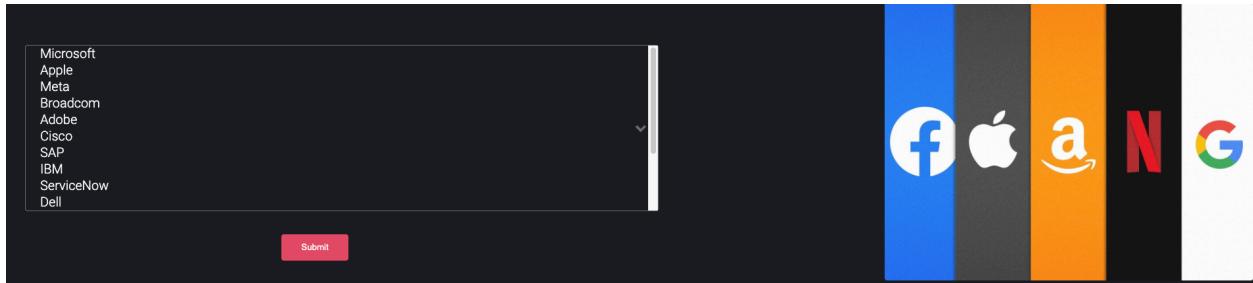


Fig 3.7.2.3 Company Interest Section

- Figure 3.7.2.4 showcases features the page offers.

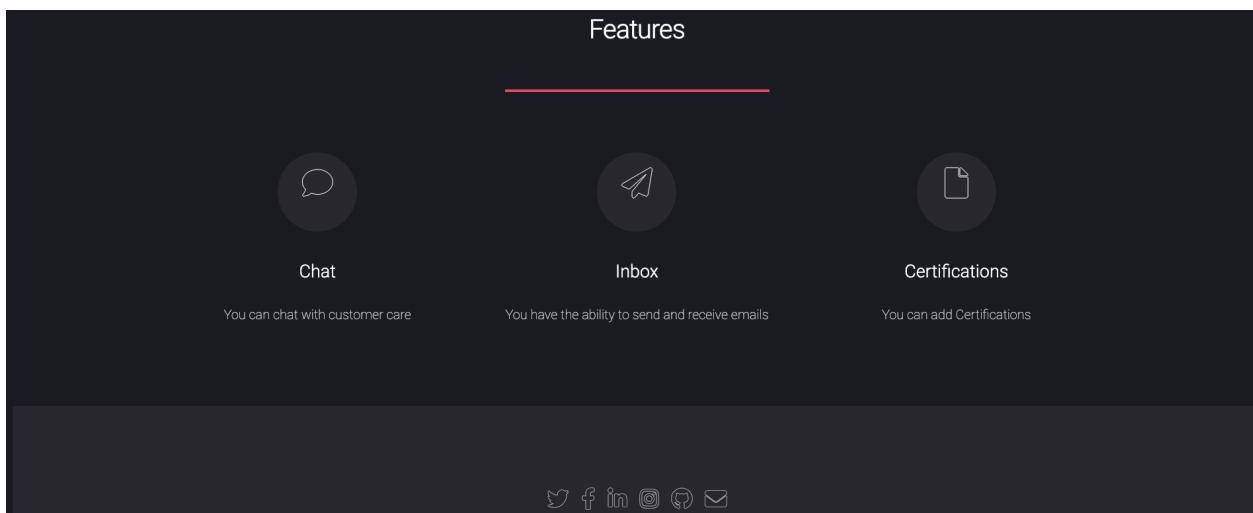


Fig 3.7.2.4 Features Section

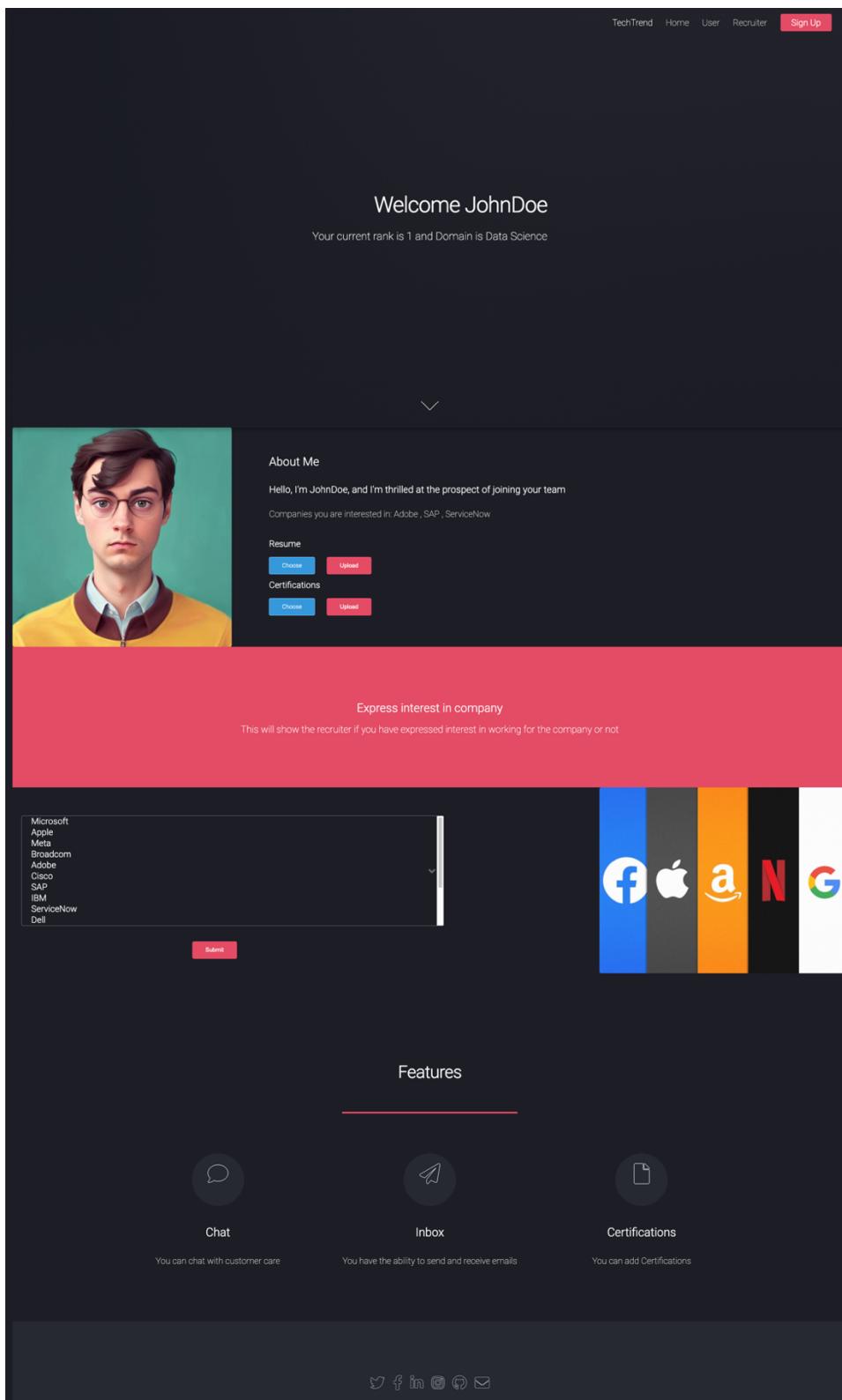


Fig 3.7.2.5 Candidate page

- Figure 3.7.2.5 showcases the candidate page of my website.

3.7.3 Recruiter Page

This section will provide an overview of features available on the recruiter's webpage, detailing the functionalities and tools available for use.

The Recruiter page will have the following sections:

- About me section - where they can provide information about their company along with a brief introduction
- Find section - Identify and explore potential candidates for the position by filtering candidates based on global ranking, domain and company interest.
- Domain section - To specify the domain in which they are seeking candidates.
- Filter section - The "Filter" section enables users to refine candidate selections by ranking and domain expertise, providing a more targeted hiring process.

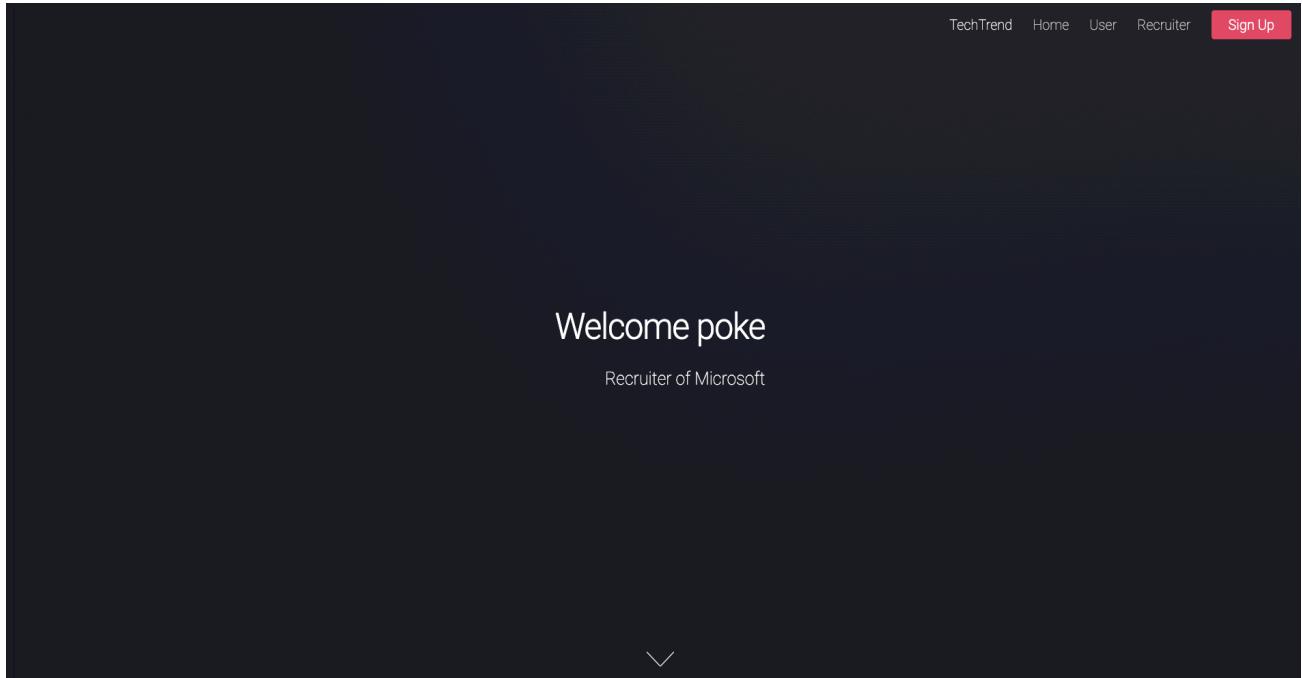


Fig 3.7.3.1 Cover Page Section

- Figure 3.7.3.1 showcases the recruiter's name and the company the recruiter is hiring for.
- If the recruiter hasn't specified the company they are hiring for, the system will prompt them to provide the company name.
- To register as a recruiter, individuals must select the recruiter checkbox on the sign-up page, thereby completing their registration as recruiters.

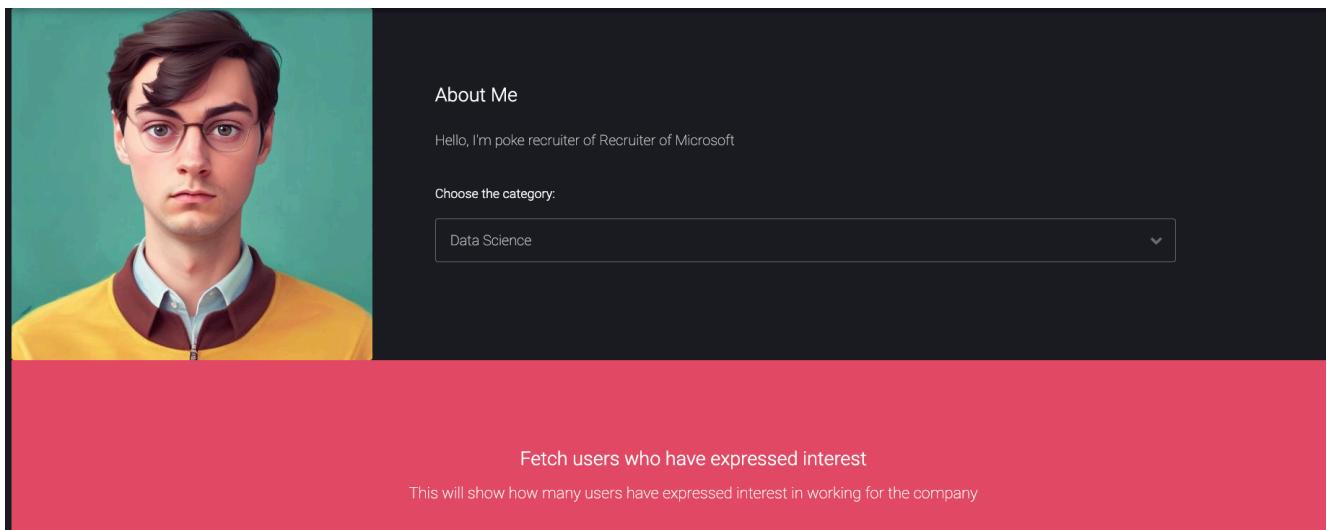


Fig 3.7.3.2 About Me Section

- Figure 3.7.3.2 showcases the name of the recruiter along with a short introduction.
- The recruiter can choose to filter candidates by domain.
- After selecting the category, a graph will appear below showing candidate names ranked accordingly.
- Recruiters can focus on assessing candidates who meet specific job requirements, improving decision-making.

Recruiter of Microsoft				
Fetch User Details				
User Name	Contact Info	Rank	Interests	Category
pirate	pirate@gmail.com	1	• Microsoft	Devops Engineer

Fig 3.7.3.3 Candidate Details Section

- Figure 3.7.3.3 showcases a list of users who have expressed interest in working in the company the recruiter is hiring for.
- In every row, you'll find the user's username, along with their contact details like email address.
- Additionally, the current global rank of the user is displayed.
- The list also includes the companies that the user has shown interest in, providing insight into their professional preferences.
- Additionally, it also shows the category name to which each user's resume is classified.

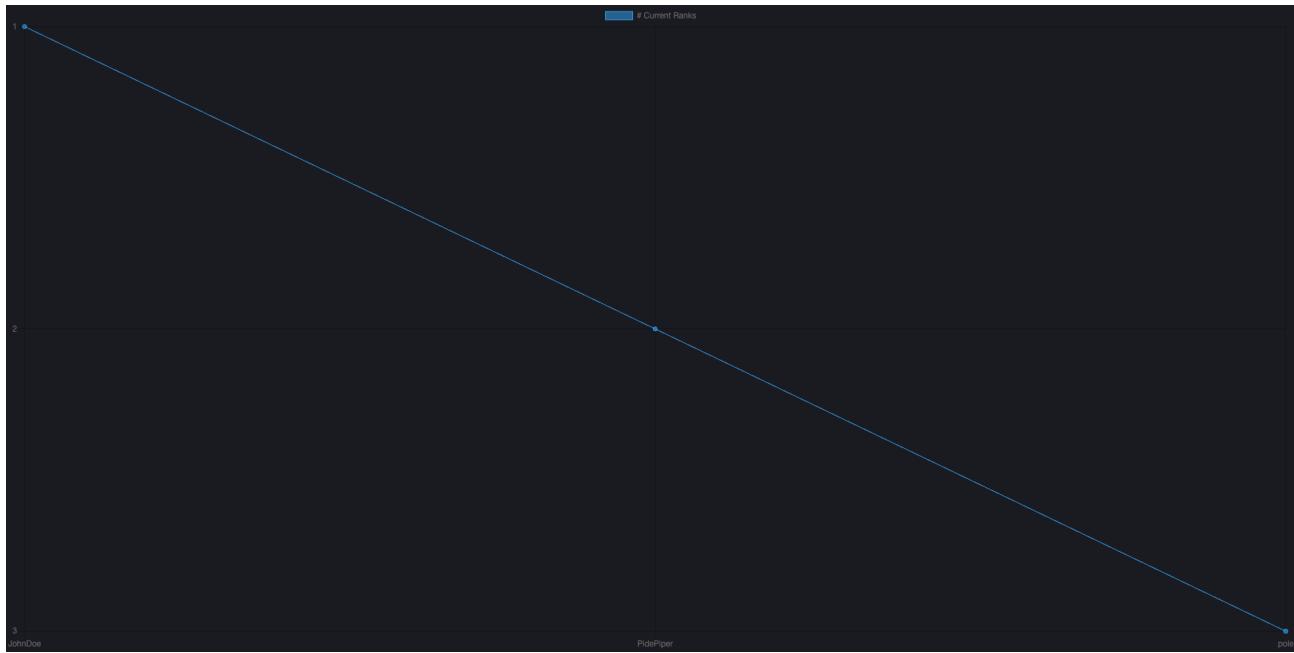


Fig 3.7.3.4 Filtered Candidate Section

- Figure 3.7.3.4 showcases a graph showing the filtered candidates based on the domain for which recruiters have a filter.

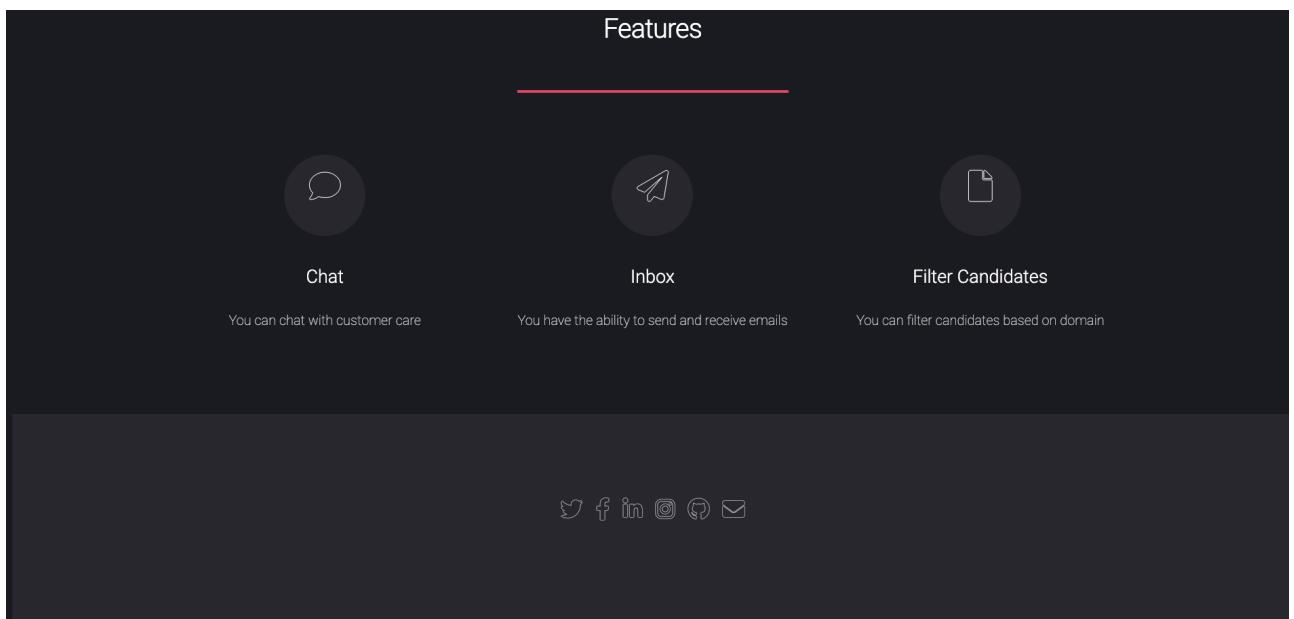


Fig 3.7.3.5 Features Section

- Figure 3.7.3.5 showcases features the page offers.

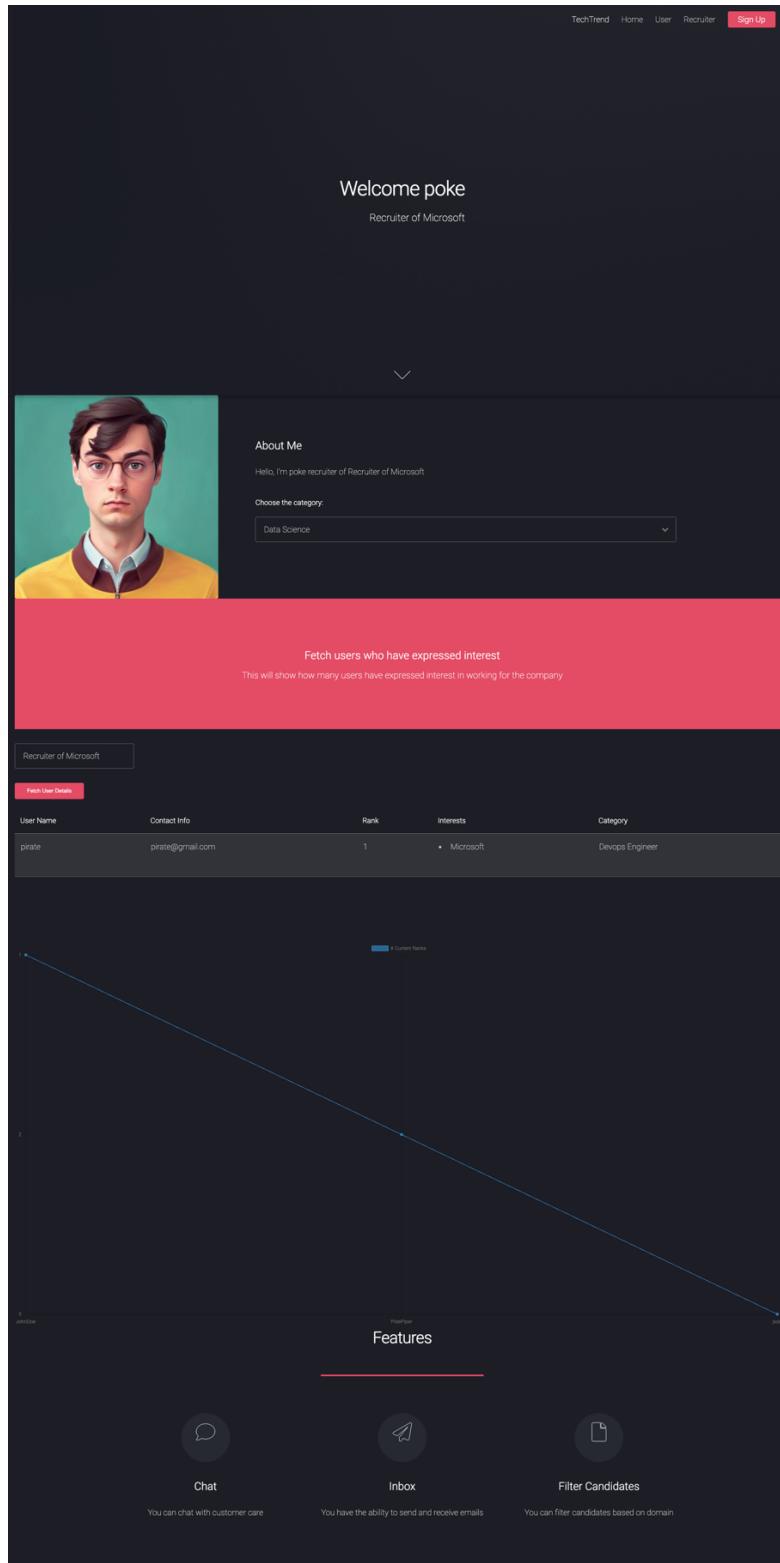


Fig 3.7.3.6 Recruiter Page

- Figure 3.7.3.6 showcases the recruiter page of my website.

3.8 Activities

Table 3.8.1 offers a summary of the tasks executed during the project.

Activities	Tasks	Deliverables
Neural Network Model Development	Develop and train OneVsRest Classifier and KNN models for resume analysis and candidate ranking. Fine-tune models based on iterative testing and feedback.	Trained models with documented accuracy rates.
Web Scraping Application	Develop a web scraping application using BeautifulSoup to extract real-time data from job boards. Implement error handling and data validation mechanisms.	Functional web scraping application providing timely technology trend data.
Website Development	Develop a user-friendly interface using Angular with pages for candidates and recruiters. Integrate the web scraping application, Neural Network models and Hadoop database.	A fully functional website with integrated applications.
Big Data Implementation	Set up a Hadoop cluster for storing Users, recruiters, and additional data. Implement Apache Kafka for data ingestion and Apache Spark for data processing.	Configured and operational Big Data infrastructure
REST API Integration	Develop a REST API using Flask to establish a connection between the website and Hadoop-stored data. Implement secure data transfer protocols.	Functional REST API ensures seamless data flow
Testing	Conduct accuracy testing for the resume analysis and candidate ranking models. Perform UI/UX testing for website usability. Conduct API testing to ensure data retrieval from both the database and job boards.	Testing reports highlighting accuracy rates and system reliability.

Tab 3.8.1 Activities

4. Software Requirement Specifications (SRS)

This platform aims to streamline the recruitment process, prioritize skill development, and enhance the connection between candidates and recruiters.

Functional Requirements

- User Authentication
 - The system shall provide a secure authentication for candidate/recruiter.
 - The Candidate and Recruiter will be assigned a login password.
- Resume Analysis
 - The system shall have a resume analysis module which involves Machine Learning models like OneVsRest Classifier and KNN.
 - The system shall let candidates post their CVs or Resumes for assessment.
 - The system will give out domain tips and scores based on the resume analysis.
- Candidate Ranking
 - The system shall use the Machine Learning models to rank candidates according to their skill, certification, and resume content.
 - The recruiters shall be able to see the ranked list.
 - The candidates shall be able to see his/her current global ranking.
- Trend Analysis
 - The system shall use a web scraping application built using Beautiful Soup to gather real-time information about the latest technological trends from job portals such as LinkedIn.
 - The system shall have a graph showing current technology.
- User Interaction
 - The candidates shall have an opportunity to choose up to twenty firms that they would like to work for.
 - The candidates shall be able to email the recruiters through the system.
 - The recruiters shall be able to use emails, filters, and express areas of interest.
- Certifications and Skill Ratings
 - The candidates shall have an opportunity to upload certificates.
 - The candidates can assess their respective skill ratings.
- Recruitment Process Management
 - The recruiter shall have access to the global ranking of all candidates, enabling them to filter based on domains or candidates who have expressed interest in the Recruiter's current hiring company. This feature empowers recruiters to reach out to candidates directly via email, eliminating the need for candidates to submit formal applications.

Non-functional Requirements

- Performance
 - The system shall have no deterioration in performance when multiple users are executing simultaneous operations.
- Security
 - The system shall provide encryption and secure storage of user data, such as resumes and personal data.
 - The system shall limit unauthorized access to users.
- Usability
 - The system shall have an intuitive user interface centered around smooth experience for a user.
 - The system shall give tooltips and directives to users.
- Scalability
 - The system shall have architecture which will allow scalability, such as accommodating more users.
- System Interfaces
 - The system shall connect to external job boards via web scraping.
 - The system shall use Hadoop and Apache Kafka for data storage and processing in integration with big data technologies.

5. Environment

The environment section describes the several technical contexts that were used during the project's development. These environments include programming languages, hardware specifications, IDEAs, and platforms that are specifically designed for a given use.

Development Environment

- Computer
 - The development of the project has been performed on a Mac laptop with an Apple M2 processor with 8 CPU cores (4 performance cores and 4 efficiency cores and 10 GPU cores) along with 256GB SSD and 8GB RAM.
- Programming Languages
 - I have used Angular for the development of the front end.
 - For back-end API development, I have used Flask.
 - I have built web scraping applications with Python and Beautiful Soup.
 - Neural Network Models has been built using Python.
 - Hadoop Integration using Apache Kafka has been done using Python.

- Integrated Development Environment (IDE)
 - Visual Studio Code has been used to integrate development environment for both back-end and front-end application.
 - I have used Visual Studio Code for integrating Hadoop using Apache Kafka.
- Google Colab
 - I have utilized Google Colab for the development and training of OneVsRest Classifier and KNN models.
 - It provides GPU support which will help speed up the process during training phase.

Big Data Environment

- Hadoop Cluster
 - Local Hosting
 - Hadoop clusters has been developed locally for storage as well as processing purposes during development.
 - To replicate the distributed environment, the cluster is installed in the development machine.
- Apache Kafka
 - I have setup Apache Kafka for data streaming and real-time data ingestion to Hadoop database.
 - The data is to be fed to Apache Kafka for storage in the Hadoop database.
 - With Apache Kafka, our data ingestion process into the Hadoop database is streamlined, ensuring scalability, reliability, and optimal performance.

Web Development Environment

- Web Server
 - The website is locally hosted for testing and debugging purposes.
 - Because I'm utilizing Angular, the application employs NodeJS as the web server.
 - The REST API has been developed using Flask.
- Frameworks
 - Angular powers the front-end development, ensuring a robust and interactive user interface.
 - Flask serves as the backbone for API development, providing flexibility and scalability for back-end operations.

Machine Learning Environment

- Neural Network Libraries
 - Model development and training process has been done on Google Colab platform.
 - TensorFlow and Keras libraries has been used for model development.

Quality Assurance and Testing Environment

- Testing Platforms
 - Entire application was tested on multiple browsers like Chrome browser, Brave browser, and Safari browser.
 - The application underwent testing on both macOS and Windows operating systems.
 - This process was performed manually.
- Issue Tracking
 - Issues were tracked using Jira and Excel Sheet.

Deployment Environment

- Local Environment
 - The project is currently hosted on my base machine.
 - This includes the complete website comprising both its frontend and backend, along with the machine learning model and the Hadoop database.
- Cloud Environment
 - The MongoDB database is utilized for storing user and recruiter information.
 - The database is hosted on the Microsoft Azure Cloud platform.

Security Environment

- Encryption Mechanisms
 - Local security measures, such as encrypted connections, were implemented during development.
 - Encryption for MongoDB database was handled by the Azure cloud provider.
- Firewalls and Access Controls
 - User authentication was implemented for all users and recruiters to prevent unauthorized access.
 - Unauthorized access was prevented utilizing passwords.
 - Accessing the MongoDB database involves utilizing my user credentials for data protection.

6. Project Results

The following tangible outcomes and deliverables will be presented for demonstration and evaluation:

- Functional Platform
 - Description: A complete recruitment platform “Tech Trend” for resume analysis, ranking the candidate, and analyzing the current trending technologies.
 - Demonstration: The demonstration of the platform in real-time shows the overall user interaction with the system, beginning with the display of current trending technologies on the home page. The demonstration will then walk through the user journey, such as logging into the user account, uploading the resume for analysis, and viewing the skill-based ranking. Additionally, we will showcase how the candidates can express interest in companies. For the recruiters, the demonstration will highlight how they can filter the candidates based on domain and ranking and reach out to them directly via email.
- Neural Network Models
 - Description: Using trained OneVsRest Classifier and KNN models for resume analysis and ranking candidates.
 - Demonstration: The demonstration will showcase the precision and accuracy of neural network models in ranking the candidates based on their skills and domain.
- Web Scraping Application
 - Description: A web scraping application powered by Beautiful Soup to extract live data about popular technologies from job postings.
 - Demonstration: The demonstration will showcase a live scenario where the application can fetch and present the latest technology trends from job boards like LinkedIn.
- Website Interface
 - Description: A friendly interactive website for job seekers and employers using Angular
 - Demonstration: The demonstration will showcase the seamless user interaction with the system in real-time.
- REST API
 - Description: A REST API written in Flask for communication between the website and the stored data in Hadoop.
 - Demonstration: The demonstration will showcase operations involving accessing and refreshing data from the database through the website via the API.
- Big Data Integration
 - Description: Hadoop and Apache Kafka integration for effective data storage and retrieval for users and recruiters.
 - Demonstration: We will be explaining the implementation of Hadoop for data storage, Kafka for data transfer.

- Testing Reports
 - Description: Reports detailing the testing stage, such as model accuracy tests, user interface/user experience testing, and application protocol testing.
 - Demonstration: We will be summarizing main results and metrics obtained for testing to establish platform reliability and functionality.
- Documentation
 - Description: A complete record and a concluding project report.
 - Demonstration: We will be summarizing the documentation to highlight the transparency and comprehensiveness of the project and documentation.
- Final Presentation
 - Description: A final summary outlining all key aspects, including project goals, methods applied, results achieved, and future implications.
 - Demonstration: We will be presenting the entire project, showcasing all the key aspects, live demonstration, results achieved, and future implications

Machine Learning Model Results

The OneVsRest Classifier along with the KNN algorithm is used to build the Machine Learning model for resume filtering. Figure 6.1 illustrates that the model is trained on a total of 1657 entries obtained from sources such as Kaggle and web scraping from platforms like LiveCareer.

```
RangeIndex: 1657 entries, 0 to 1656
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   Category    1657 non-null   object 
 1   Resume      1657 non-null   object 
 2   Final_Resume 1657 non-null   object 
dtypes: object(3)
memory usage: 39.0+ KB
Displaying the distinct categories of resume:

['Data Science' 'HR' 'Advocate' 'Arts' 'Web Designing'
 'Mechanical Engineer' 'Sales' 'Health and fitness' 'Civil Engineer'
 'Java Developer' 'Business Analyst' 'SAP Developer' 'Automation Testing'
 'Electrical Engineering' 'Operations Manager' 'Python Developer'
 'DevOps Engineer' 'Network Security Engineer' 'PMO' 'Database' 'Hadoop'
 'ETL Developer' 'DotNet Developer' 'Blockchain' 'Testing']
```

Fig 6.1 Recruiter Page

The resume filtering model is trained on these 1657 entries, which will predict the domain to which the resume categorizes. I have trained the model on 25 categories: Data Science, HR, Advocate, Arts, Web Designing, Mechanical Engineer, Sales, Health and fitness, Civil Engineer, Java Developer, Business Analyst, SAP Developer, Automation Testing, Electrical Engineering, Operations Manager, Python Developer, DevOps Engineer, Network Security Engineer, PMO, Database, Hadoop, ETL Developer, DotNet Developer, Blockchain and Testing

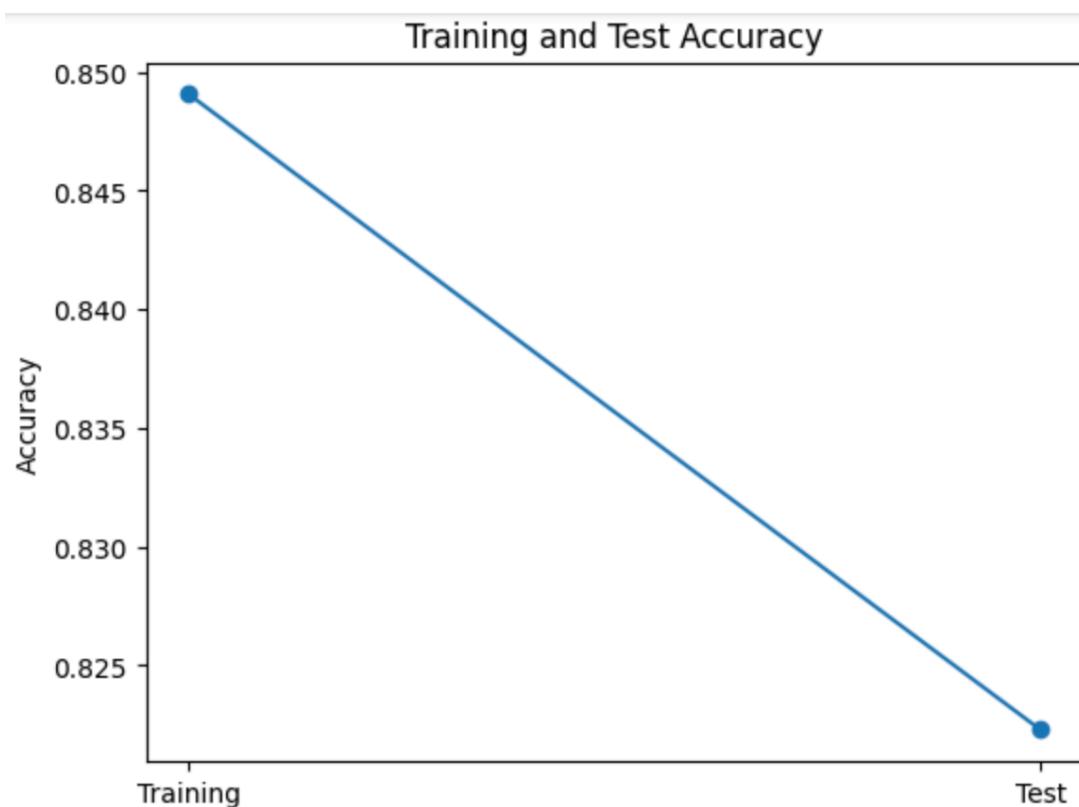


Fig 6.2 Recruiter Page

Figure 6.2 represents the model accuracy, for 1657 records the model achieves a training accuracy of 85% and a test accuracy of 82%.

The candidate ranking model uses a cosine similarity function to map the reference feature vector. This vector is formed by aggregating all resumes within the domain, serving as the reference feature vector. The candidate whose resume bears the closest resemblance to the reference feature vector receives a higher ranking. This approach ensures that candidates who closely align with the desired attributes and qualifications for the given position are prioritized in the ranking process, facilitating more efficient and effective candidate selection.

7. Project Schedule

Table 7.1 provides an overview of the tasks, including their schedule, current status indicating completion progress, and brief comments detailing the executed tasks throughout the project.

Status: completed  ongoing  not completed 

Item #	Tasks	Status	Comments
1	<p>Week 1-2: Project Planning and Technology Exploration Tasks</p> <ul style="list-style-type: none"> • Get acquainted with Angular, Flask, and everything related to them. • Cloud Services Integration – identify and plan (AWS, Microsoft Azure). • Task: Research Angular and Flask, explore integration with cloud services, and high-level project documentation. 		The task has been completed. I have familiarized myself with both Angular and Flask frameworks, The application is set to run on my local machine, with the MongoDB database hosted on the Microsoft Azure cloud network. The Flask API will be developed using Python, while the Angular framework will be employed for the frontend of the website. The backend database will be Hadoop HDFS, hosted on my local machine, and the MongoDB database. Additionally, the KNN model for resume analysis and candidate ranking has been built using Python and trained on GoogleColab.
2	<p>Week 3-4: Development Environment Setup Tasks</p> <ul style="list-style-type: none"> • Create a local development environment with Visual Studio Code. • Set up dependencies • Install and create a primary cloud platform environment. • Task: Setting up the environment, installing dependencies, project layout. 		The task has been successfully completed. The setup process is now finalized. I've chosen Visual Studio Code as my main text editor. Angular, Flask, and Hadoop have all been successfully configured. Furthermore, MongoDB is currently hosted on the Microsoft Azure cloud platform. The neural network models has been trained on GoogleColab on Python.
3	<p>Week 5-8: Development Phase å Tasks</p> <ul style="list-style-type: none"> • Front-end Development (25 hours): Begin by mastering Angular basics and building the home, candidate, and recruiter pages. • API Development (25 hours): Basic endpoints for user interactions, learning Flask basics, and setting up API for cloud. • Task: Cloud configuration, front-end, API development, and integration. 		This task has been completed. The home page, candidate page, and recruiter page have been implemented using Angular, and the backend API has been implemented using Flask API in Python. The MongoDB database, currently hosted on the Microsoft Azure cloud platform, has been integrated using Flask API.
4	<p>Week 9-10: Creating a neural network model. Tasks</p> <ul style="list-style-type: none"> • Refresh knowledge of neural networks. 		The task has been successfully completed. A KNN model for resume analysis has been developed on GoogleColab initially using the dataset from Kaggle, but since the dataset was fetched using

	<ul style="list-style-type: none"> • Create an OneVsRest Classifier and KNN resume analysis model with a ranking system. • Train models with sample data set up Google Colab for model training. • Task: Neural network modeling, API Integration, and Google Colab for training. 		web scraping and sourced from livecareer.com and indeed.com I couldn't find the dataset on any other website. The original dataset consisted of approximately 900 records, and after web scraping from my end on livecareer.com, the current dataset has expanded to nearly 1600 records. The candidate ranking has been achieved through the utilization of a cosine similarity function and feature vectors.
5	<p>Week 11-12: Web Scraping Application Tasks:</p> <ul style="list-style-type: none"> • Develop a simple web scraping application using Beautiful Soup. • Discover and display the latest technological trends on homepage. • Task: Develop the web crawler program, integrate with homepage 	●	The task has been completed using beautiful soup in python. The web scraper is designed to retrieve the current number of jobs and count of jobs released today in the United States based on domains, from job portal (LinkedIn). This real-time data will be displayed on the homepage, offering up-to-date insights into job trends.
6	<p>Week 13-14: Data Processing with Kafka and Apache Spark</p> <p>Tasks:</p> <ul style="list-style-type: none"> • Set up Apache Kafka for data streaming. • Integrate Apache Kafka with the platform for real-time data ingestion. • Task: Integration of Kafka 	●	Successfully implemented the task of saving user-uploaded resumes onto Hadoop HDFS without relying on Apache Spark. Utilized Apache Kafka for data ingestion and this approach ensured a seamless integration of user resumes into the Hadoop ecosystem, facilitating further data processing and analysis tasks.
7	<p>Week 15-16: Testing and Debugging</p> <p>Tasks:</p> <ul style="list-style-type: none"> • Conduct testing for major functionalities. • Task: Testing and debugging, refinement of functionalities. 	●	The task has been successfully completed, with primary unit test cases written and the webpage thoroughly tested. Any identified issues have been debugged to ensure smooth functionality.
8	<p>Week 17-18: Deployment and Basic User Training</p> <p>Tasks:</p> <ul style="list-style-type: none"> • Deploy the minimal platform on a local server or a cloud environment. • Gather initial feedback for adjustments. • Task: Deployment and initial feedback collection. 	●	This task has been completed, the project has been deployed on my local base machine and the MongoDB database is hosted on Microsoft Azure cloud platform. I have also considered feedback from my project guide and advisor while deploying my application.
9	<p>Week 19-20: Documentation and Final Presentation</p> <p>Tasks:</p> <ul style="list-style-type: none"> • Document the essential development process. • Integrate cloud-related documentation. • Practice and refine the presentation. • Task: Documentation preparation, final presentation, and submission. 	●	This task has been completed, and a final project report has been compiled. The project is now prepared for demonstration, accompanied by a presentation.

Tab 7.1 Project Schedule

8. Bibliography

Song, L., Shi, J., Luo, P., Wei, W., Fang, Y., & Wang, Y. (2020). More Time Spent, More Job Search Success?: The Moderating Roles of Metacognitive Activities and Perceived Job Search Progress. *Journal of Career Assessment*, 28(1), 147–164. <https://doi.org/10.1177/1069072719841575>

Sokratis Vidros, Constantinos Kolias, Georgios Kambourakis. (2016). Online recruitment services: another playground for fraudsters. *Computer Fraud & Security*, Volume 2016, Issue 3, 8-13. ISSN 1361-3723. [https://doi.org/10.1016/S1361-3723\(16\)30025-2](https://doi.org/10.1016/S1361-3723(16)30025-2)

Albassam, W. A. (2023). The Power of Artificial Intelligence in Recruitment: An Analytical Review of Current AI-Based Recruitment Strategies. *International Journal of Professional Business Review*, 8(6), e02089. <https://doi.org/10.26668/businessreview/2023.v8i6.2089>

Kumar, R. S., Prakash, N., & Anbuchelian, S. (2020). Prediction of Job Openings in the IT Sector using a Long Short-Term Memory Model. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud) (I-SMAC), 49–953. <https://doi.org/10.1109/I-SMAC49090.2020.9243483>

Kumar, N., Gupta, M., Sharma, D., Ofori, I., & Bhardwaj, A. (2022). Technical Job Recommendation System Using APIs and Web Crawling. *Computational Intelligence and Neuroscience*, 2022, 1–11. <https://doi.org/10.1155/2022/7797548>

Indeed. (2023, March 10). How many job applications per day? <https://www.indeed.com/career-advice/finding-a-job/how-many-job-applications-per-day>

Bergson-Shilcock, A., Taylor, R., Nye Hodge, National Skills Coalition, Federal Reserve Bank of Atlanta, LaPrad, J., Unruh, R., & Miller, S. (2023). CLOSING THE DIGITAL SKILL DIVIDE: THE PAYOFF FOR WORKERS, BUSINESS, AND THE ECONOMY. https://nationalskillscoalition.org/wp-content/uploads/2023/02/NSC-DigitalDivide_report_Feb2023.pdf

Federal Trade Commission. (2024, February). Facts about fraud from the FTC – and what it means for your business [Blog post]. Retrieved from <https://www.ftc.gov/business-guidance/blog/2024/02/facts-about-fraud-ftc-what-it-means-your-business>

Gaurav Dutta (2020). Resume dataset [Data set]. Kaggle. <https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset>

9. Glossary

- **REST API** – Representational State Transfer Application Programming Interface
- **AWS** – Amazon Web Services
- **IT** – Information Technology
- **API** – Application Programming Interface
- **KNN** – K-Nearest Neighbour
- **UI/UX** – User Experience and User Interface
- **EC2** – Amazon Elastic Computer Cloud