### RESUME SCREENING AND JOB RECOMMENDATION

A PROJECT REPORT

submitted

*in the partial fulfillment of the requirements for the award of the degree of*

### BACHELOR OF TECHNOLOGY

in

### COMPUTER SCIENCE AND ENGINEERING

by

**SHAIK ABDUL KHADEER – 20B81A05C1**

**P. PRATHIMA – 20B81A05F5**

**L. SIRI CHANDANA – 20B81A05H1**

Under the guidance of Ms P.Sampurnima Sr.Asst. Professor



### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

**(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana- 501 510

**May 2023**

CVR College of Engineering

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, NAAC

(Approved by AICTE & Govt. of Telangana and Affiliated by JNT University)

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

### CERTIFICATE

This is to certify that the project entitled **“Resume Screening and Job Recommendation using Machine Learning”** being submitted by **SHAIK ABDUL KHADEER (20B81A05C1)**,

**P. PRATHIMA (20B81A05F5)** and **L. SIRI CHANDANA (20B81A05H1)** in partial

fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** to the CVR College of Engineering, is a record of bona fide work carried out by them under my guidance and supervision during the year 2022-2023.

The results embodied in this project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of the project guide Signature of the HOD

**Ms. P. Sampurnima Dr. A Vani Vathsala**

**Sr. Assistant Professor Department of CSE**

**Project Coordinator External Examiner**

### DECLARATION

We hereby declare that the project report entitled **“Resume Screening and Job Recommendation using Machine Learning”** is an original work done and submitted to CSE Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in Computer Science and Engineering and it is a record of Bonafide project work carried out by us under the guidance of **Ms. P. Sampurnima** ,Sr. Assistant Professor, Department of Computer Science and Information Technology.

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this Institute or any other Institute or University.

Signature of the Student SHAIK ADBUL KHADEER

Signature of the Student

P. PRATHIMA

Signature of the Student

L. SIRI CHANDANA

### ACKNOWLEDGEMENT

The success and outcome of this project require a lot of guidance and aid from many people, and we are extremely privileged to have this all along with the completion of our project. All that we have done is only due to such supervision and aid and we should not forget to thank them.

We respect and thank our internal guide **Ms. P. Sampurnima**, for supplying us an opportunity to do this project work in CVR College of Engineering and giving us all the support and guidance, which made us complete the project duly. We are extremely thankful to her for supplying such a nice support and guidance, although she had a busy schedule managing the official affairs.

We would like to thank the Head of the Department, Professor **Dr. A Vani Vathsala** for her meticulous care and cooperation throughout the project work. We thank **Ms. A. Soujanya,** the Project Coordinator for providing us an opportunity to do this project and extending good support and guidance.

We are thankful for and fortunate enough to get constant encouragement, support, and guidance from all **Teaching staff of CSE Department** which helped us in successfully completing our project work. Also, we would like to extend our sincere esteem to all staff in the laboratory for their prompt support.

# ABSTRACT

Due to the increasing growth in the recruitment market, companies may receive an enormous number of resumes from applicants with different fields of experience and specializations. The traditional method for classifying resumes is time-consuming and concerned authorities need to go through every resume sent by the candidates. This process is complicated. So, in this project, we are going to propose a model that takes the resume of the candidate in a particular format, extracts the text in the resume by using python module pdfminer which converts the unstructured data into some structured data and using pyresparser, we extracted resume data. With this, we gave some recommendations to the candidate for his improvement and statistics for the recruiter to select best candidates.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents** | | | |
|  | | | Page No. |
|  |  | List of Tables | vii |
|  |  | List of Figures | vii |
|  |  |  |  |
|  |  |  |  |
|  |  | **CHAPTERS** |  |
| 1 |  | **Introduction** | 1 |
|  | 1.1 | Motivation | 2 |
|  | 1.2 | Problem statement | 3 |
|  | 1.3 | Project Objectives | 4 |
|  | 1.4 | Project report Organization | 5 |
| 2 |  | **Literature Survey** |  |
|  | 2.1 | Existing work | 6 |
|  | 2.2 | Limitations of Existing work | 6 |
| 3 |  | **System Requirements** |  |
|  | 3.1 | Software requirements and Hardware requirements | 7 |
|  | 3.2 | System Main module | 8 |
| 4 |  | **Proposed System Design** |  |
|  | 4.1 | System Workflow | 9 |
|  | 4.2 | Use case Diagram | 11 |
|  | 4.3 | Class Diagram | 12 |
|  | 4.4 | Activity Diagram | 13 |
|  | 4.5 | Sequence Diagram | 14 |
|  |  |  |  |
|  |  |  |  |
| 5 |  | **Implementation & Testing** |  |
|  | 5.1 | **Implementation** | 15 |
|  | 5.2 | Sample Code | 16 |
|  | 5.3 | Testing | 18 |
| 6 |  | **Conclusion & Future scope** | 25 |
|  |  | **References:** | 26 |
|  |  |  |  |

### LIST OF TABLES

Table Title Page

3.1 System Requirement 6

### LIST OF FIGURES

Figure Title Page

* 1. System Workflow 6
  2. Working of Resume & Recommendations 8
  3. Sample Codes 15
  4. [Testing 17](#_TOC_250000)

**Chapter 1**

# INTRODUCTION

Resume screening is a process in which recruiters find the right person for a job by narrowing down the list of eligible candidates based on the information in their resumes. Recruiters review candidates' resumes themselves or use software to complete the process.

Manual screening involves recruiters reviewing each resume on their own, while software- based resume screening uses artificial intelligence tools that scan resumes for predetermined criteria.

Resume screening is a straightforward way to remove unqualified candidates from the application process and shortlist candidates to interview.



* 1. **Motivation**

Resume screening and job recommendation technologies are becoming increasingly popular as companies seek more efficient and effective ways to identify the most qualified candidates for open positions. One of the key motivations for these technologies is the high volume of job applications that many companies receive, which can make manual screening a time consuming and inefficient process. By using technology to automate the screening process, companies can quickly and objectively identify the most qualified candidates, reducing the time and resources required for recruitment.

Another motivation for these technologies is the desire for more accurate candidate matching.Traditional resume screening methods often rely on keyword matching or a checklist of required qualifications, which can result in false positives and negatives. By using machine learning algorithms to analyze resumes and job requirements, companies can identify more nuanced patterns and criteria to match candidates with the right job. This can result in better quality hires and improved retention rates, as candidates are more likely to be a good fit for the position and the organization.

Additionally, these technologies can help reduce recruitment costs by streamlining the screening process and reducing the need for expensive recruitment agencies or staffing services. By automating the screening process, companies can free up resources to focus on other aspects of recruitment, such as candidate engagement and onboarding. This can help improve the overall candidate experience and reduce turnover rates, as candidates are more likely to feel valued and supported throughout the recruitment process.

Finally, the need to stay competitive in the job market is another key motivation for these technologies. By providing a more efficient, fair, and accurate recruitment process,companies can position themselves as innovative and desirable employers. This can helpattract top talent and improve the overall reputation of the organization, leading to improved business outcomes and a more engaged workforce.

* 1. **Problem Statement**

The recruitment process is a critical function for organizations of all sizes and industries. It is important for companies to identify the most qualified candidates for open positions to ensure the long-term success of the organization. However, the traditional method of manual screening of resumes and applications is often time-consuming, inefficient, and can lead to biased decisions that negatively impact the recruitment process. Additionally, the high volume of applications received by companies can make it challenging to identify the most qualified candidates quickly and effectively. These challenges have led to the need for an automated system that can efficiently screen resumes and recommend the most qualified candidates for a particular job. The problem addressed by this project is the need for a more efficient and effective way of screening resumes and recommending candidates for open positions, while reducing the time and cost associated with the recruitment process. Additionally, the project aims to address the issue of biased decision-making by providing an objective and consistent screening process that reduces the potential for human error and subjectivity. Overall, the problem statement for this project is to develop a resume screening and job recommendation system that can address the challenges faced by organizations in the recruitment process, and provide a more efficient, accurate, and fair process for identifying the most qualified candidates.

* 1. **Project Objectives**

The objective of this project is to develop a Resume Screening and Job Recommendation Application using resume parsing techniques. The project aims to achieve the following objectives:

* + 1. Develop a resume parser that can extract relevant information such as skills, education, work experience, and achievements from unstructured resume data.
    2. Implement a user-friendly interface that allows job seekers to input their resume data and receive personalized feedback on how to improve their resumes.
    3. Develop a machine learning model that can analyze the parsed resume data and provide feedback on areas such as formatting, keyword optimization, and overall content.
    4. This project also allows the organizations to screen the resumes of the candidates, based on their key constraints.

Overall, the objective of this project is to develop a resume screening and job prediction application that can help job seekers to improve their resumes and increase their chances of

being selected for interviews and also help recruiters find the right person for a job. By leveraging resume parsing techniques and machine learning algorithms, the system can provide personalized feedback that is tailored to the needs of each job seeker. This can lead to a more efficient and effective job search process, while reducing the time and effort required for both job seekers and employers.

**1.4 Project Report Organization**

The project report for "Resume Screening and Job Recommendation” for both Recruiters and Job Seekers using Resume Parser can be organized into the following sections:

Introduction: This section will introduce the project, its objectives, and the motivation behind it. It will also provide an overview of the report's structure.

Literature Review: This section will review the existing literature on resume screening, job recommendation, and resume parsing techniques. It will also discuss related works on NLP and machine learning in recruitment.

Methodology: This section will describe the methods used to develop the resume screening and job recommendation system, including the technologies and tools employed, as well as the design and implementation of the system.

System Architecture: This section will provide a detailed description of the system architecture, including its modules, components, and data flows.

Results and Evaluation: This section will present the results of the evaluation of the resume screening and job recommendation system.

Future Work: This section will discuss potential future directions for the resume screening and job recommendation system, including the integration of additional features, such as increasing scope of industries this application can be used in, video resumes and social media profiles.

Conclusion: This section will summarize the key findings of the project, its contributions to the field of recruitment, and the potential for future work.

References: This section will list the references cited in the literature review and throughout the report.

Appendices: This section will include any additional information or supporting materials, such as sample resumes, code snippets, and data analysis results

# CHAPTER 2 LITERATURE SURVEY

* 1. **EXISTING WORK**

There are several automated resume screening and checking tools available in the market, such as JobScan, Resume Worded etc. These tools use various algorithms and technologies, such as natural language processing (NLP), machine learning, and data analytics, to analyze resumes and provide insights into their content and structure. Some of the features commonly found in these tools include keyword matching, ATS compatibility, grammar and spelling checking, and formatting analysis.

In conclusion, these automated resume screening and checking tools can significantly reduce the time and effort required in recruitment, they should be used with caution and in conjunction with other methods, such as manual screening and interviews, to ensure the selection of the best candidates.

# LIMITATIONS OF EXISTING WORK

Automated resume checkers check if the resume is good according to general standards. It is not always the same general standards used by every company. Users would not know if their resume fits the organization’s standards or style. Similarly, resume screening software used by various organizations, analyze relevant data but fail to make recommendations or suggestions to the applying candidate.

# CHAPTER 3

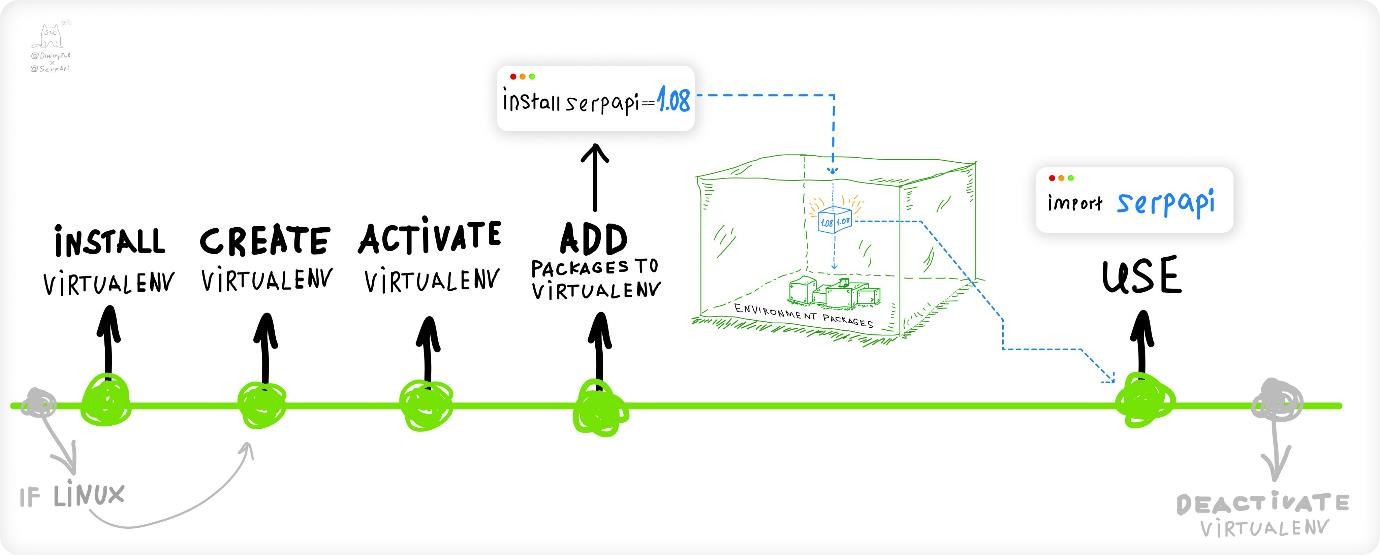
**SYSTEM REQUIREMENTS**

## Hardware and Software Requirements

|  |  |  |
| --- | --- | --- |
|  | **Hardware** | **Software** |
| **Developers** | 1. 4 GB RAM 2. 256 GB Storage 3. Intel i5 5th Gen + Processor | 1. Anaconda or Python 2. Visual Studio Code 3. Streamlit, PDFMiner 4. Pyparser, NLTK |
| **Users** | 1. Mobile 2. Tablet/PC 3. Laptop | Any Browser with internet conntivity |

*Table.1 System Requirement*

*Python Virtual Environment - Work flow*



## System Main Module

The system main module for the "Resume Screening and Job Recommendation" project includes the following components:

Resume Storing: This module allows users to upload their resumes into the system and store them on our local server for future preprocessing.

PDF Extracting: Since users can only upload resumes in PDF format, this module is responsible for processing the PDF into several images. This allows for easier text extraction in the next module.

Text Extracting: After the PDF pages are converted into images, this module extracts all the text from the images. This is necessary for natural language processing (NLP) techniques that require text input.

Smart Recommendation: Using the extracted text from the resumes, this module generates smart recommendations based on the user's skills and experience. For example, if the user has skills in data science, the system will recommend relevant courses and skills to improve their data science knowledge.

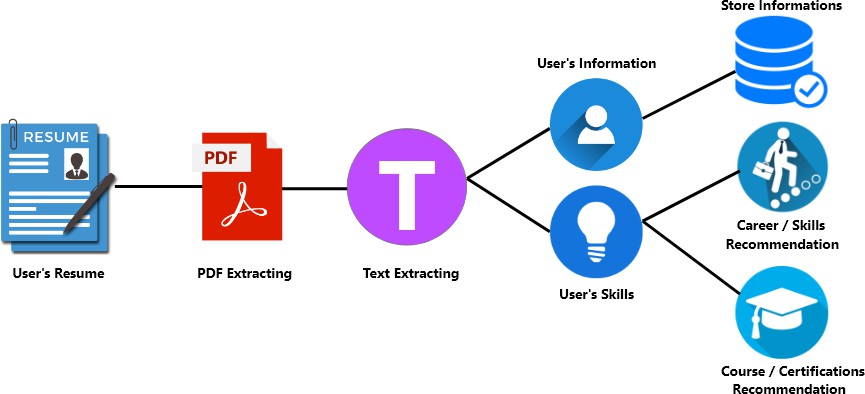
Data Store: Once recommendations are provided to the user, the system stores their data securely for future use. This allows the system to continuously improve and provide better recommendations based on the user's previous interactions.

Together, these modules create a comprehensive system that allows users to upload their resumes, receive personalized job recommendations, and have their data stored securely for future use.

# CHAPTER 4

**PROPOSED SYSTEM DESIGN**

## 4.1 System Workflow



*Working of Resume and Job Recommendation*

We are going to see how our system is working behind, we have divided our work in separate tasks, let us understand each of the steps:

## PDF Extracting

PDF Extracting is a module in which the condition is that user’s resume should be in PDF format. This module will automatically extract the user’s data from the resume.

## Text Extracting

Text Extracting is the module in which text information is fetched from the resume. This text data will be used for language processing to further tasks like recommendations and fetching the user’s personal information.

## 4.1.3 User’s Data

After the text extraction, next module is fetching the user information. For instance, to find an X person’s full name, contact, email, mobile, and skills from the text extraction data.

## Career/Skills Recommendations

Now we have fetched the information of the user. Now based on the user’s current skills we recommend career paths and skills, for example, if user has Machine Learning skills, then it will give you the career, tools and technology recommendations accordingly.

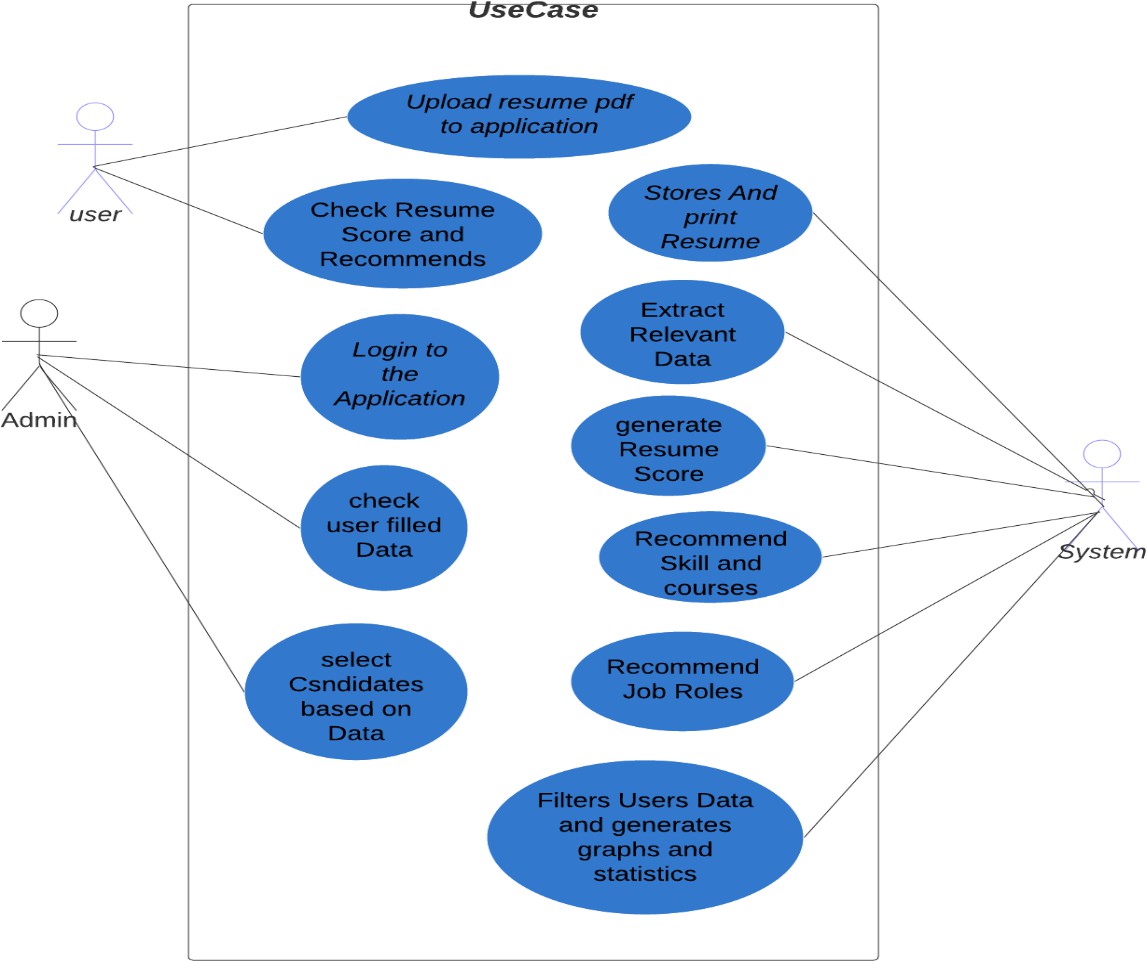
## Course/Certifications Recommendations

Now we have fetched the information of the user. Based on user’s current skills, we recommend courses and certifications, for example, if user has Machine Learning skills, then it will recommend various types (free and paid) of courses and certifications.

## Data Analytics

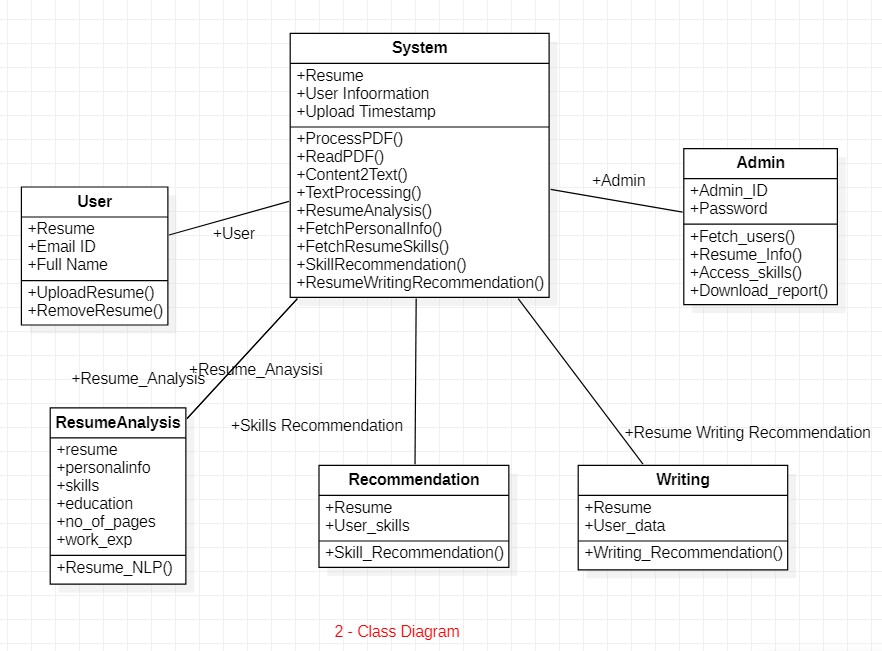
Now we have attached a new module called data analytics. We have a good amount of user data. Here, visualization is the most effective technique to understand the data’s pattern. Now we are creating the visualizations (Pie Charts) for the Admin side so data can be easily understood.

**4.2 UseCase Diagram**



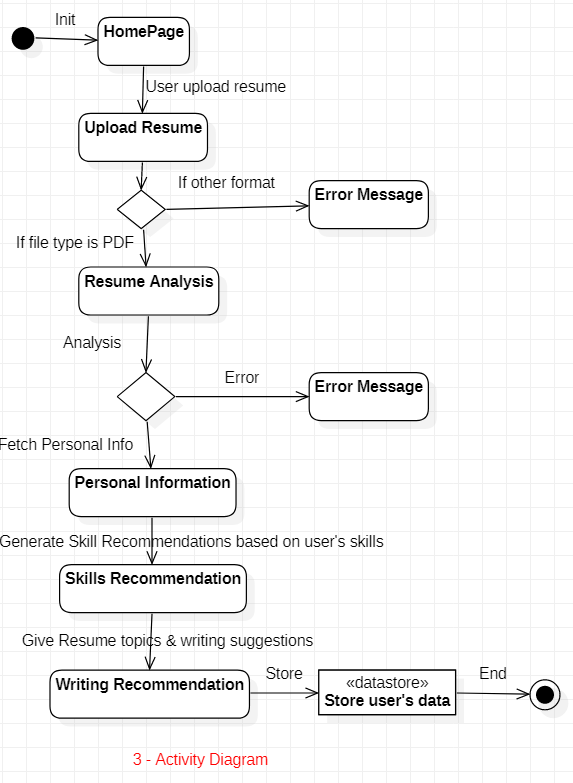
In this diagram, we can see what the functionalities user & admin use. We can see user & admin have the different access to the functional.

## 4.3 Class Diagram

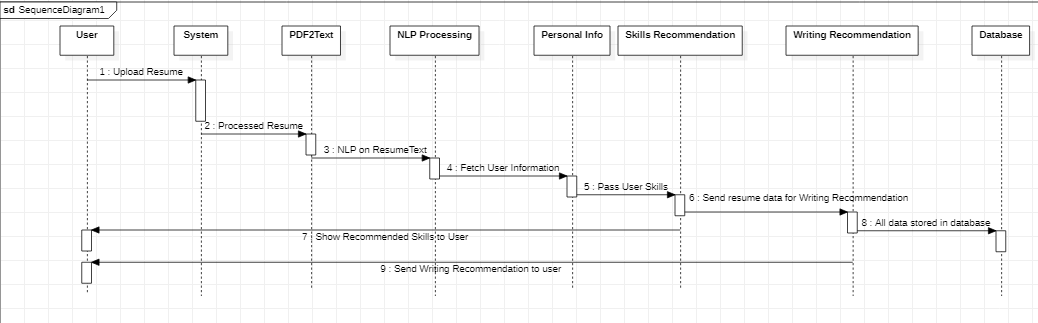


In this diagram, we see different classes and functions according to the functionality. We can see how the system is associated with the user & admin functionality.

* 1. **Activity Diagram**



* 1. **Sequence Diagram**



**Chapter 5**

**Implementation and Testing**

* 1. **Implementation**

To implement a resume screening and checker system, the following steps can be taken:

Resume storage: The system should allow users to upload their resumes, which are stored in the system for future processing. The resumes can be stored in a database or a file system.

Resume parsing: Once the resumes are stored, the system should parse the resumes to extract relevant information such as work experience, education, skills, etc. This can be done using NLP libraries or third-party parsing tools.

Keyword matching: The system should match the extracted information against a set of predefined keywords to determine the relevance of the resume to a particular job. This can be done using regular expressions or fuzzy matching algorithms.

Ranking and scoring: The system should rank the resumes based on their relevance to the job description and score them accordingly. This can be done using machine learning algorithms or heuristics.

Resume checking: The system should also check the resumes for common errors such as spelling mistakes, grammatical errors, and formatting issues. This can be done using pre- trained models or third-party libraries.

Recommendation engine: Finally, the system should recommend the most suitable candidates for a particular job based on the ranking and scoring of the resumes. This can be done using collaborative filtering or content-based recommendation algorithms.

Throughout the implementation process, it is important to document all the steps taken, the tools and libraries used, and the results obtained. This documentation will help in the future maintenance and improvement of the system, as well as in the sharing of the system with others.

## Sample Code

The code which is mentioned below is the main logic of giving the recommendations to the user. This code is only just functional part, not the full part with the backend.

## PDF Extracting

This code is used for the PDF extraction; it will extract the Pdf of resume which is uploaded by the user.

*def* pdf\_reader(*file*):

resource\_manager = PDFResourceManager() fake\_file\_handle = io.StringIO()

converter = TextConverter(resource\_manager, fake\_file\_handle, *laparams*=LAParams())

page\_interpreter = PDFPageInterpreter(resource\_manager, converter)

with open(file, 'rb') as fh:

for page in PDFPage.get\_pages(fh,

*caching*=True, *check\_extractable*=True):

page\_interpreter.process\_page(page) print(page)

text = fake\_file\_handle.getvalue()

# close open handles converter.close() fake\_file\_handle.close()

return text

## Recommender Code

This code is used for the course recommender, it will fetch the user’s skills, based on skills it will give the recommendations.

*def* course\_recommender(*course\_list*): st.subheader("\*\*Courses & Certificates➵

Recommendations\*\*") rec\_course = []

no\_of\_reco = st.slider('Choose Number of Course Recommendations:', 1, 10, 4)

random.shuffle(*course\_list*)

for c\_name, c\_link in *course\_list*: c += 1

st.markdown(*f*"({c}) [{c\_name}]({c\_link})") rec\_course.append(c\_name)

if c == no\_of\_reco: break

return rec\_course

## PDF Showing Code

This code is used to show the uploaded resume in the user interface, it simply allows to display the uploaded resume into the system.

*def* show\_pdf(*file\_path*):

with open(*file\_path*, "rb") as f:

base64\_pdf = base64.b64encode(f.read()).decode('utf-8') # pdf\_display = f'<embed

src="data:application/pdf;base64,{base64\_pdf}" width="700" height="1000" type="application/pdf">'

pdf\_display = *F*'<iframe src="data:application/pdf;base64,{base64\_pdf}" width="700" height="1000" type="application/pdf"></iframe>'

st.markdown(pdf\_display, *unsafe\_allow\_html*=True)

## Insert user’s Data Code

This code is used to insert the user data into our database.

*Def*insert\_data(*name*,*email*,*mobile*,*timestamp*,*no\_of\_pages*,*reco\_fie ld*,*cand\_level*,*skills*,*recommended\_skills*,*courses*):

DB\_table\_name = 'user\_data'

insert\_sql = "insert into " + DB\_table\_name + """ values (0,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)""" rec\_values = (*name*, *email*, str(*mobile*),

*timestamp*,str(*no\_of\_pages*), *reco\_field*, *cand\_level*, *skills*,*recommended\_skills*,*courses*)

cursor.execute(insert\_sql, rec\_values)

connection.commit()

## Visualization Code (Pie Chart)

This code is used display the Pie Chart for data analytics.

labels = plot\_data.Predicted\_Field.unique() values = plot\_data.Predicted\_Field.value\_counts() st.subheader("\*\*Pie-Chart📈 for Predicted Field Recommendations\*\*")

fig = px.pie(df, *values*=values, *names*=labels, *title*='Predicted Field according to the Skills')

st.plotly\_chart(fig)

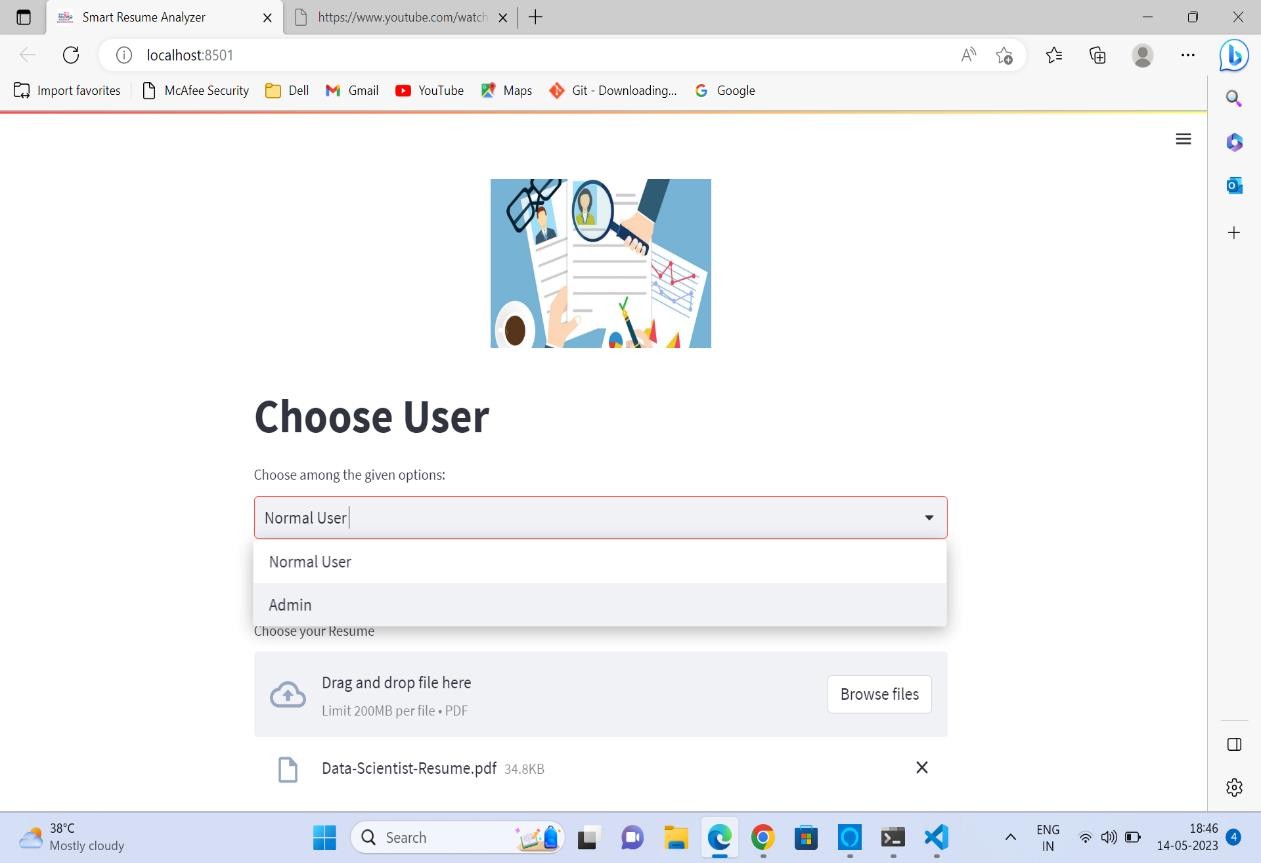
There are different implementations for different task, we have just seen the basic functions that are working for our system.

**5.3 Testing**

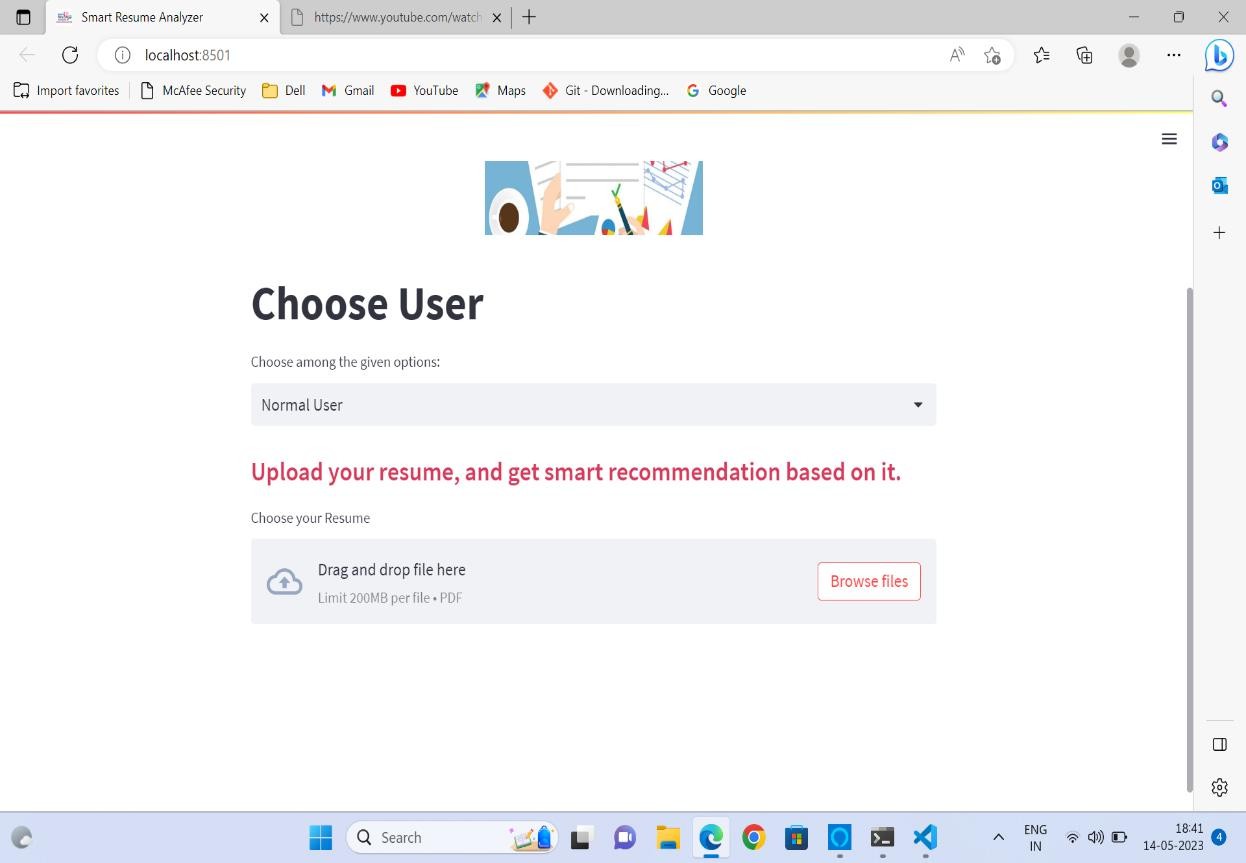
Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

## Home page of System

This will be the main page of our web application; user just need to visit the URL. It is just one-page application, user doesn’t need to do any Registration process. Everyone can use it directly.

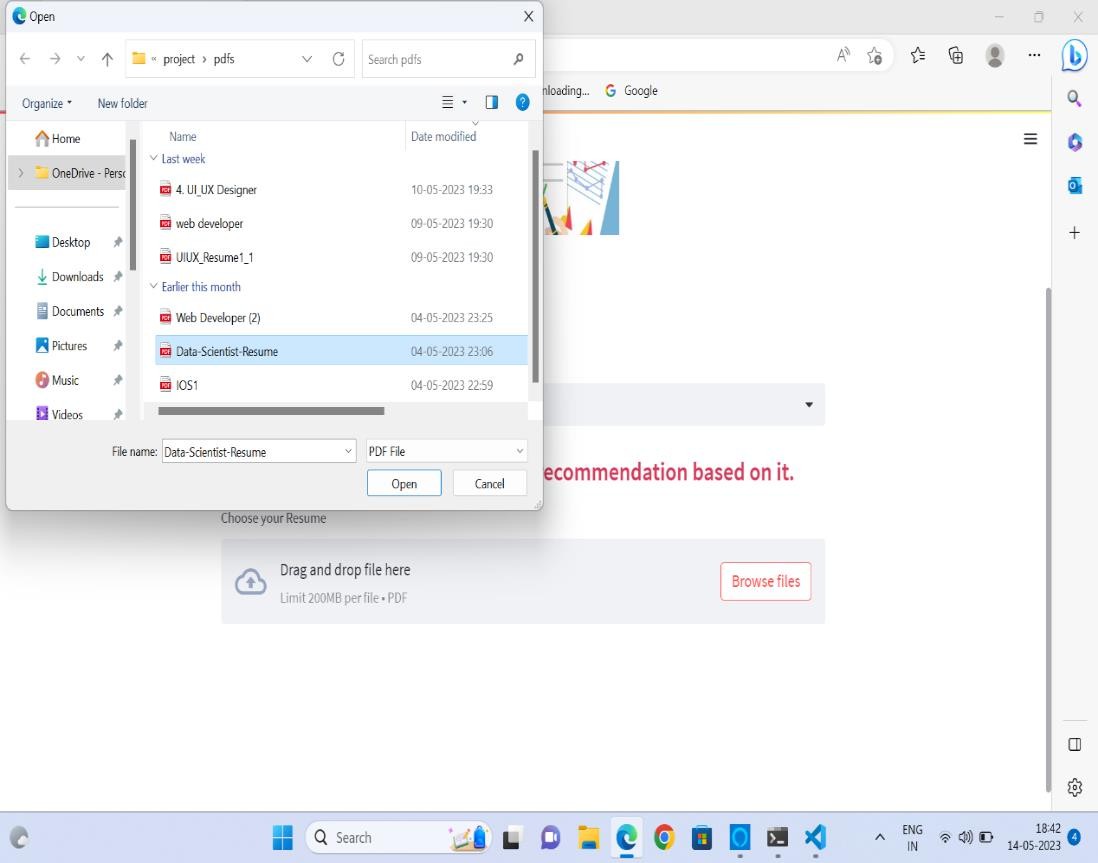


## User’s Role

User can select the role from the left panel. There are 2 roles given. One is normal user another one is Admin role. In admin role, credentials are required.

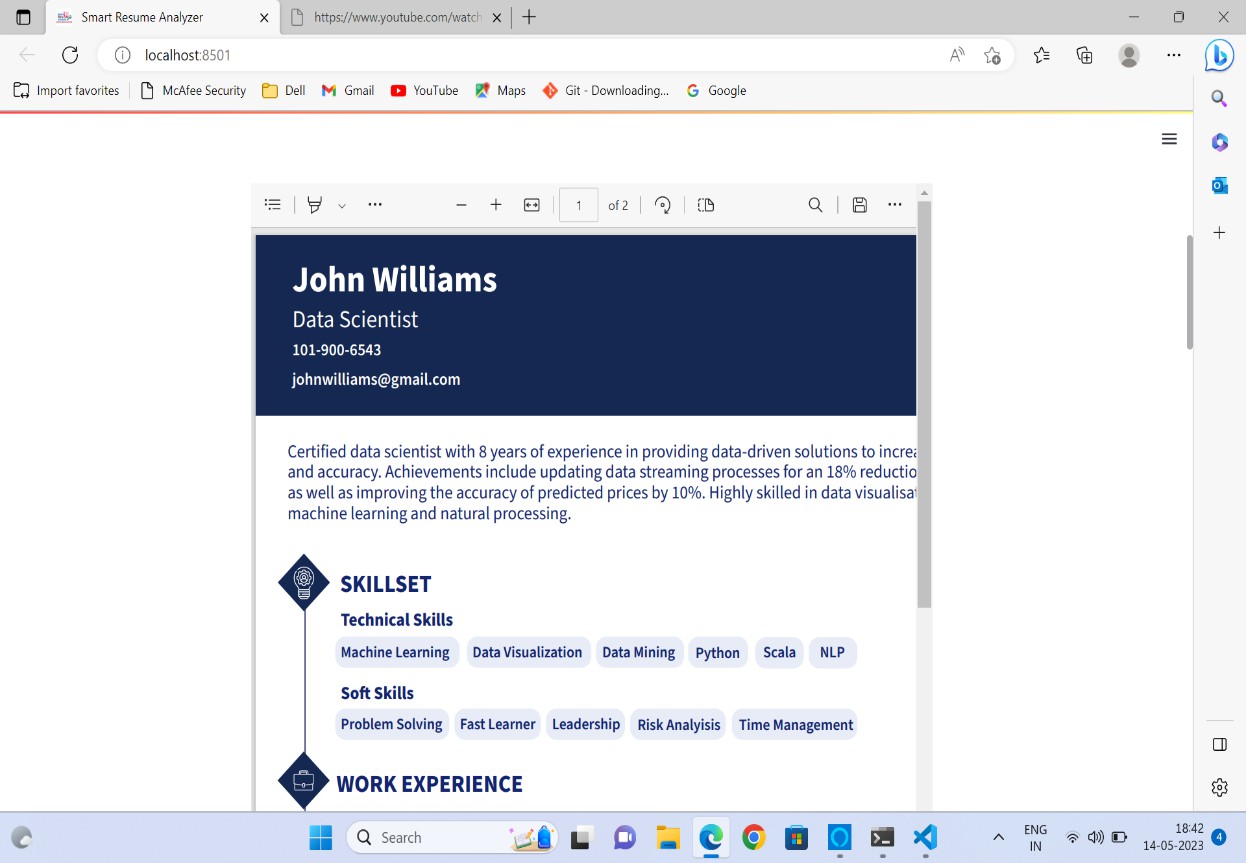
## Resume Uploader

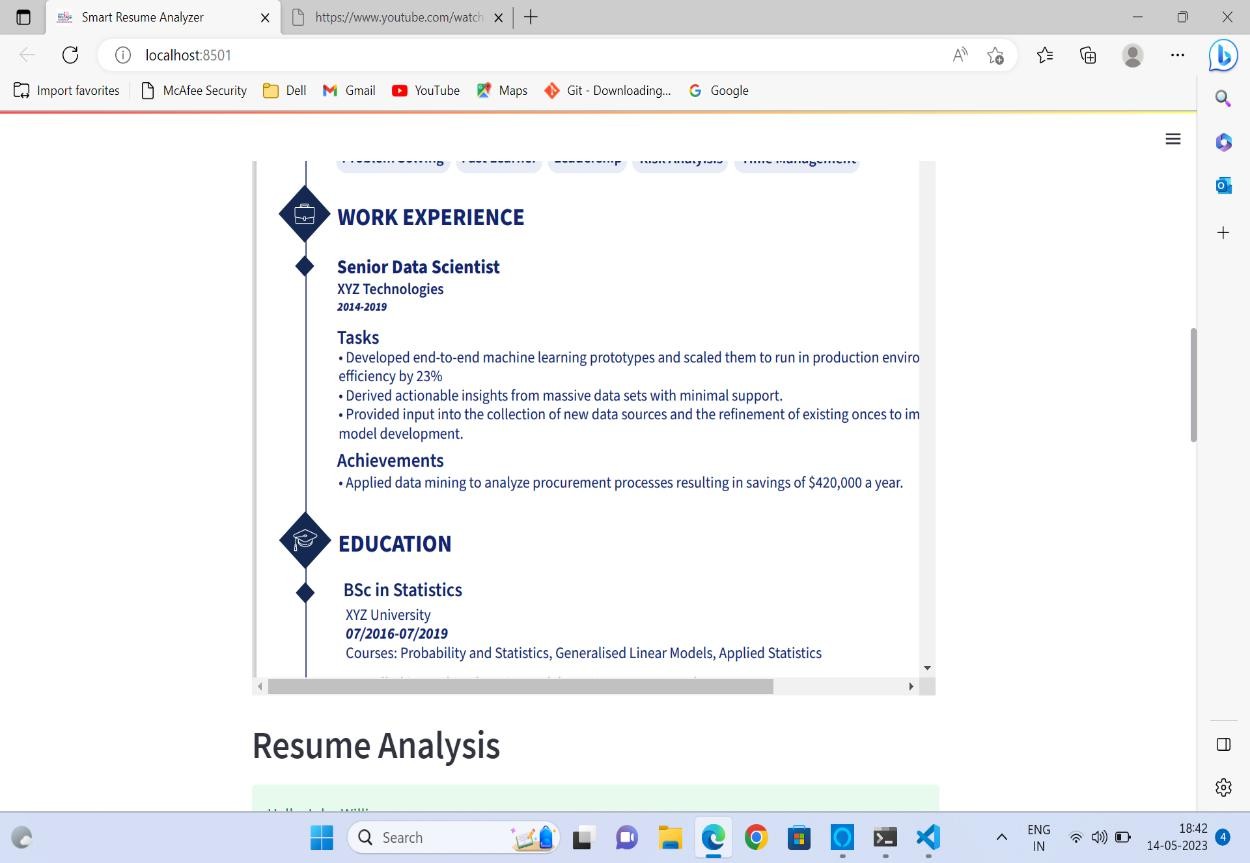
In normal user, User uploads the resume in PDF format by click on browse button or by just drag & drop. Currently PDF format is only supported.



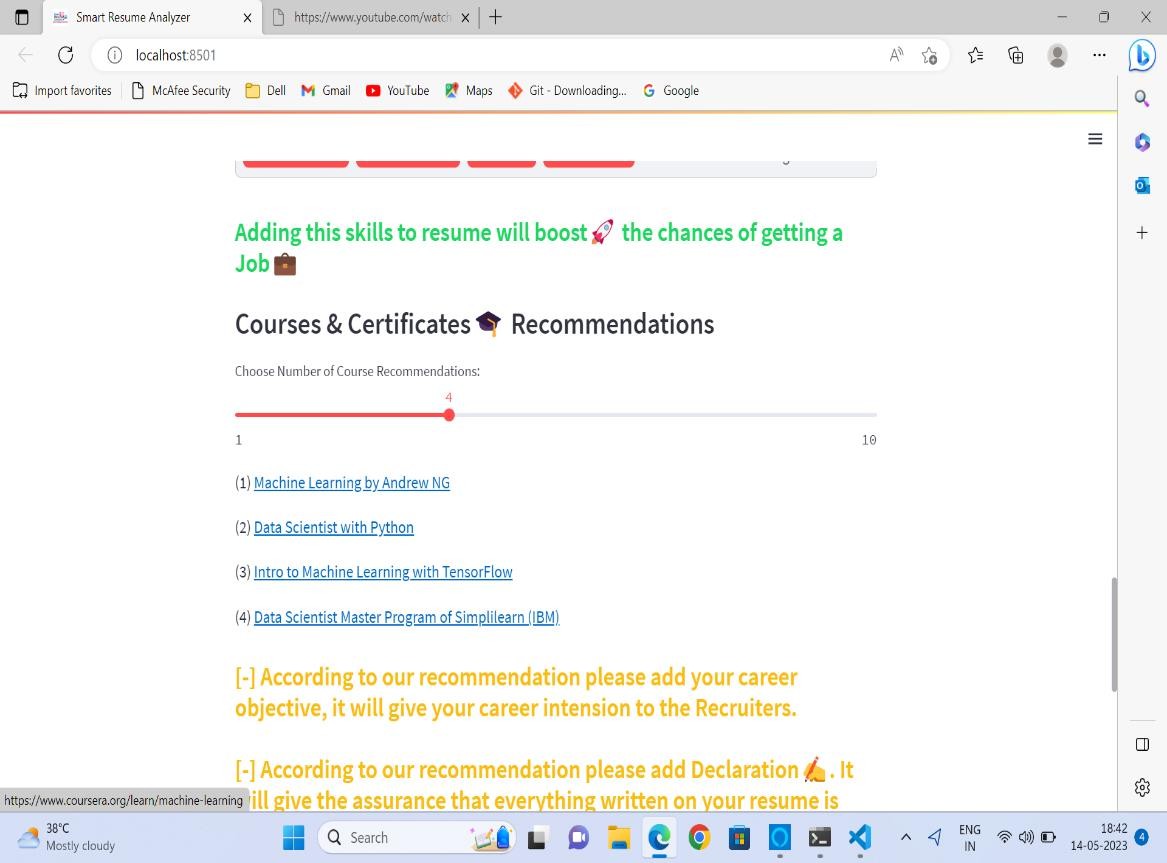
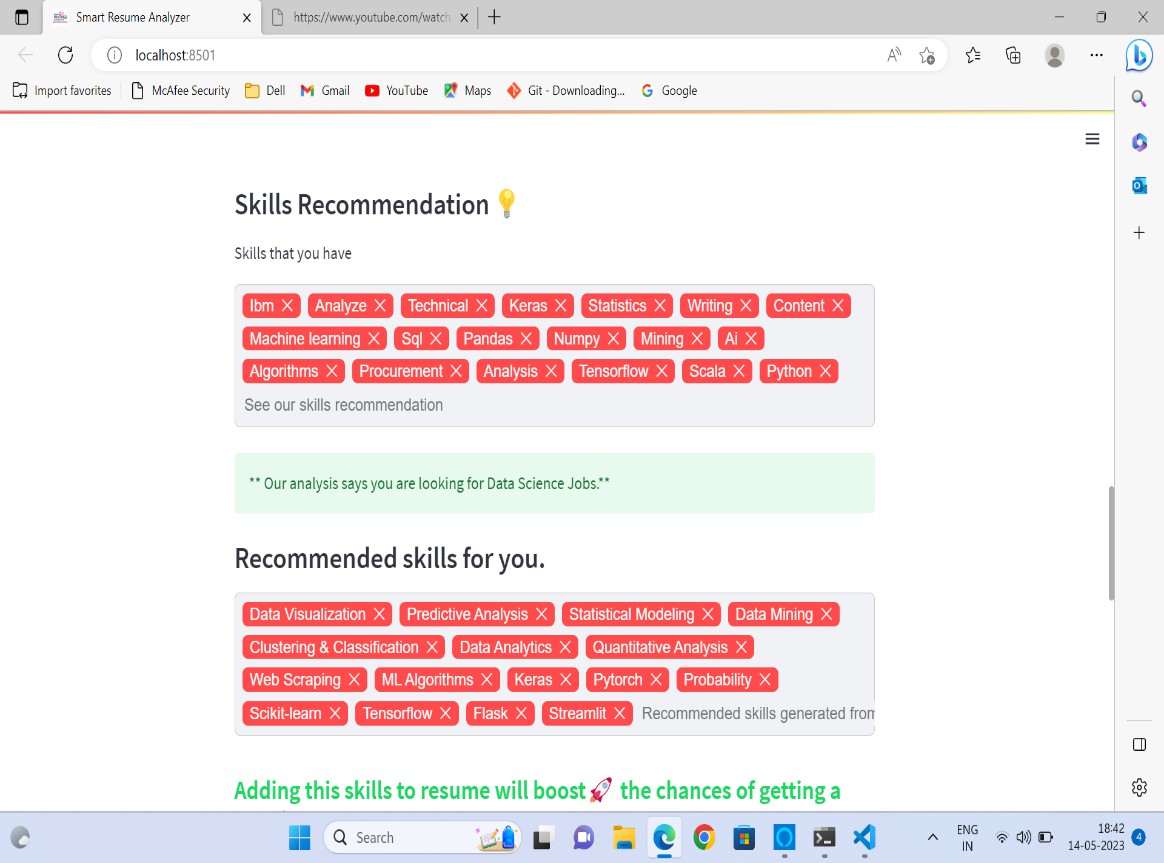
## Resume Display

After uploading the resume, it will display the resume in the web application.



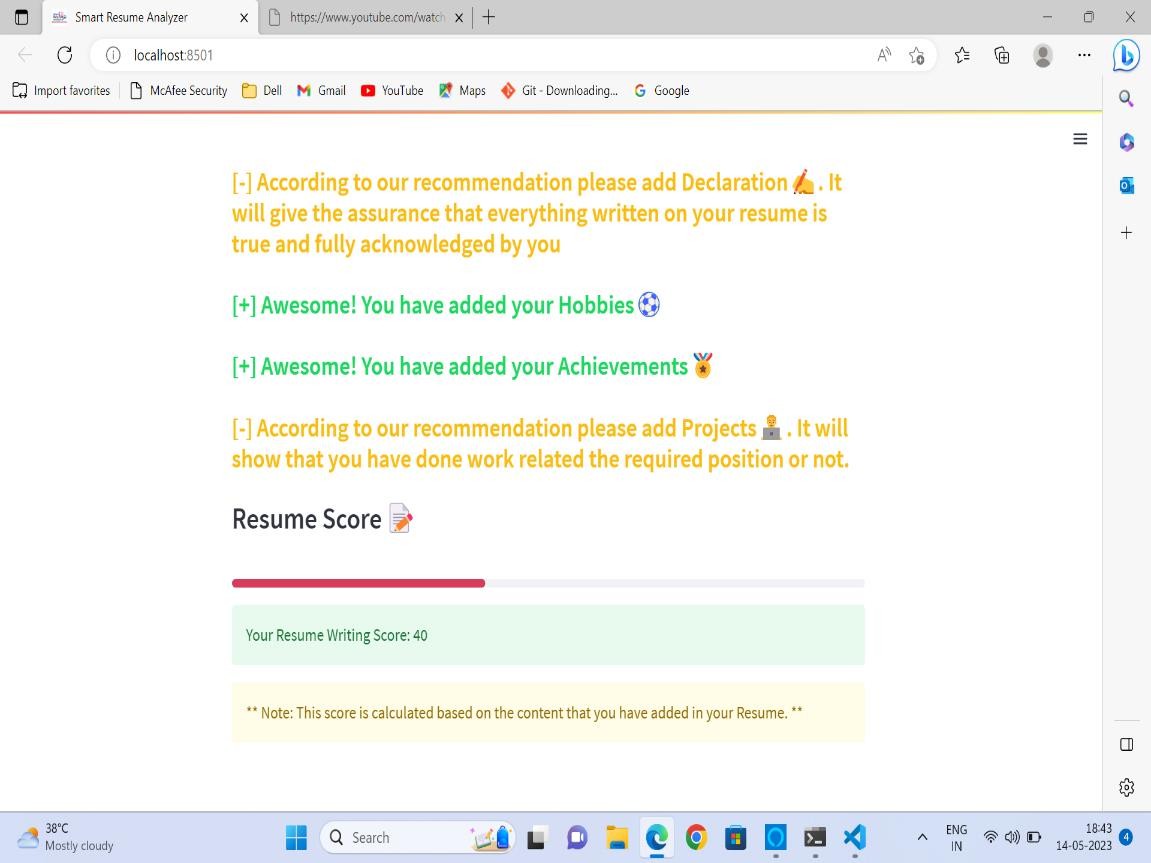


## Recommendations



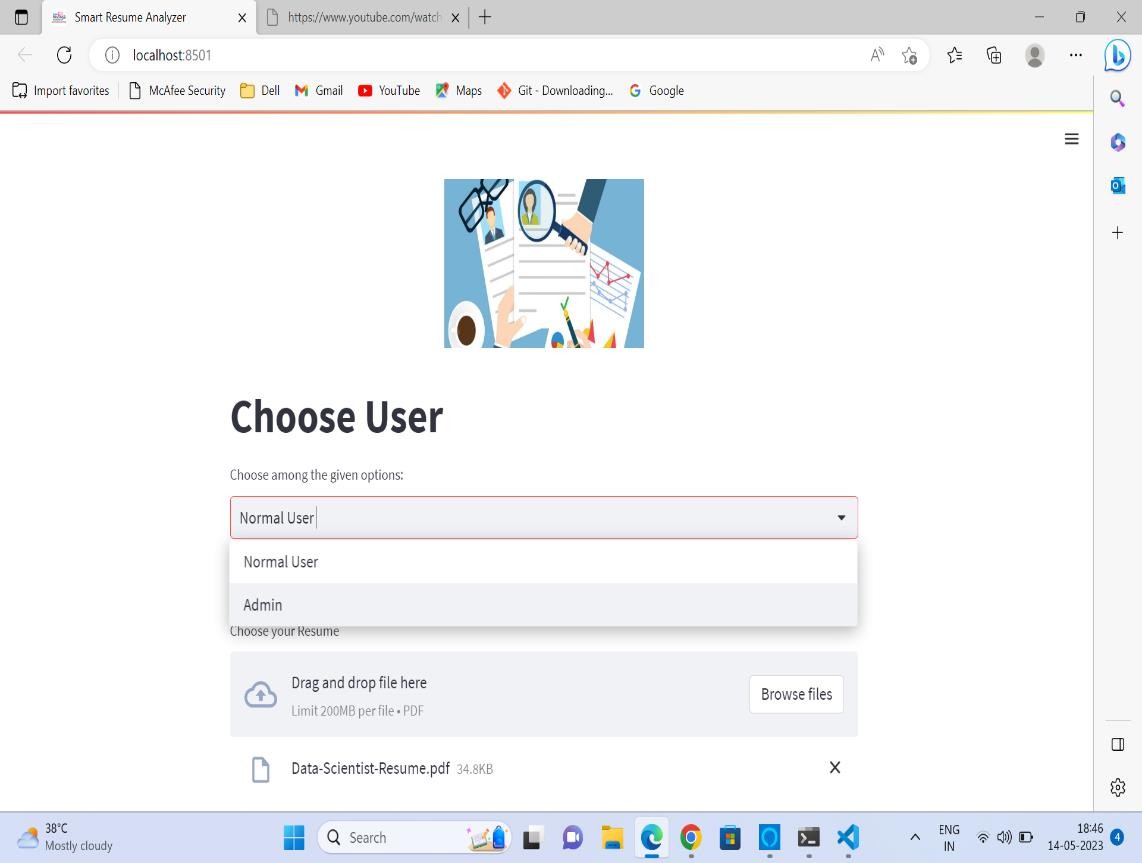
* + 1. **Resume Score: -**

Our system will generate the score for user’s uploaded resume.



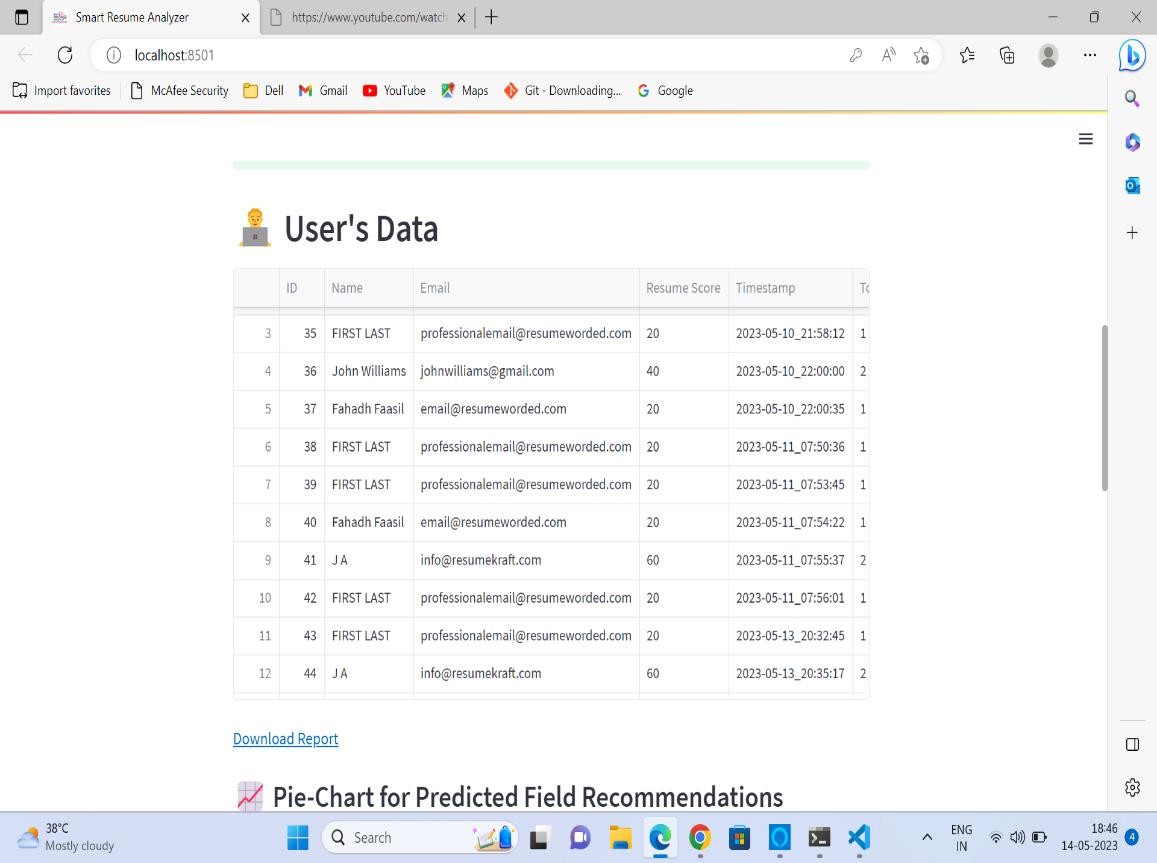
## Admin side

In selection panel, if you will select admin side, then it will show you the Admin Login panel. With default credentials admin login.



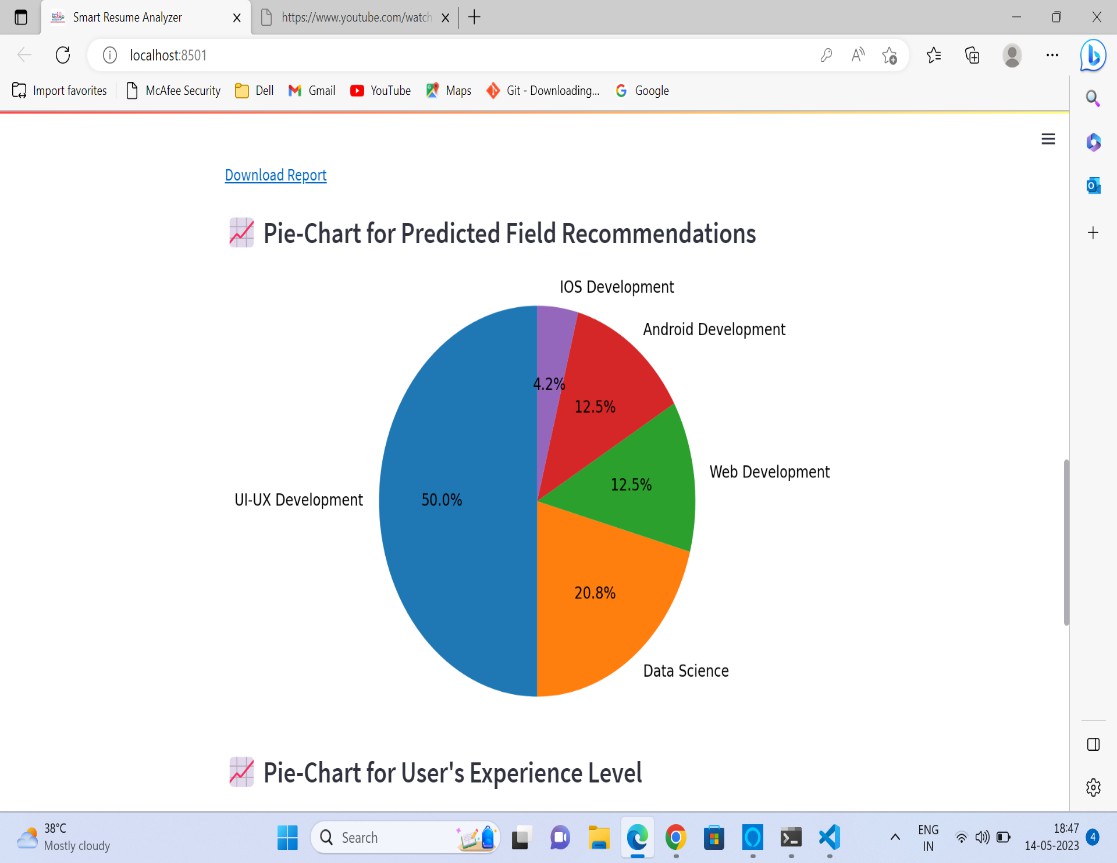
## Admin Side User report: -

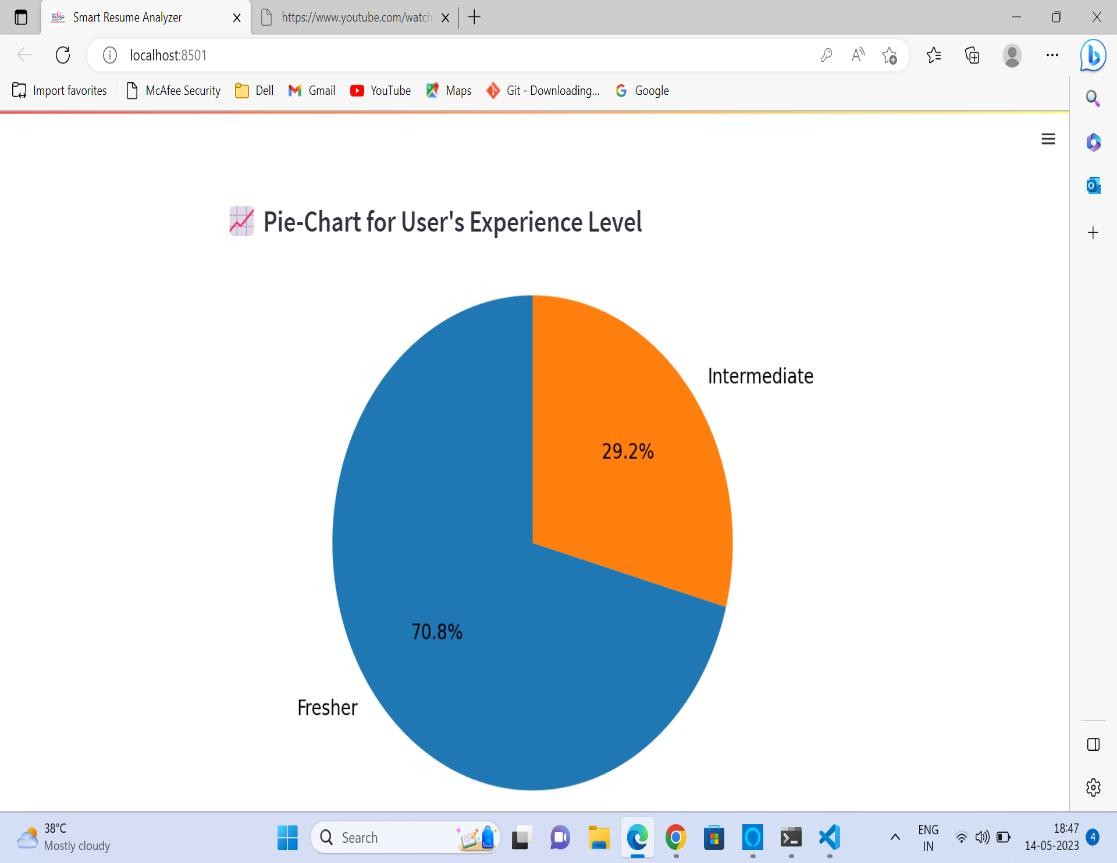
Admin can see the user’s data which is stored in the database.



## Admin Side Visualizations

Our system will display the visualizations based on the data. Representation is done using Pie-Chart for job role recommendations.





# CHAPTER 6 CONCLUSION

The purpose of this project is to learn Natural Language Processing. Using NLP, here we tried to recommend skills, courses and job roles based on extracted data. We learnt to create a web application using Streamlit python. We have gathered the knowledge of Plotly for the visualization and Data analytics which is used on the Admin side. In this project, we have learnt many Technical and Non-Technical skills including time management, documentations, NLP, Database handling, Visualization, Web development and much more.

# FUTURE SCOPE

* We can add more formats of resume, currently it is supporting only PDF format for uploading the resume.
* This system is currently working with limited types of fields and recommendations, we can say it is only working for IT professionals as of now. We can add more fields in future for all kinds of resume recommendations.

## References

* Streamlit Documentation | Documentation from Streamlit Developer Community
* OpenCV Documentation | Documentation from OpenCV Developer Community
* NLTK Documentation || Official documentation from NLTK Developer Community
* PyResParser Documentation || Official documentation from PyResParser Developer Community
* PDFMiner Documentation || Official documentation from PDFMiner Developer Community