UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+
Accredited University

Project Report

MASTER OF COMPUTER APPLICATIONS IN ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Chandigarh University, GHARUAN
AUGUST 2024

Submitted by

Pratham Bansal(23MCI10131)

Submitted to:

Dr. Gagandeep Kaur

SIGNATURE

HireSense: Application Tracking System using Transfer Learning

Table of Contents

1. Introduction to the Project

In this project, we aim to develop an Application Tracking System (ATS) using Transfer Learning, specifically leveraging Gemini Flash, a state-of-the-art natural language processing model designed for high-speed and high-efficiency performance. The primary objective is to build a system that automates the evaluation and management of job applications, focusing on features such as resume parsing, job matching, and candidate scoring.

Key Components of the Project

- **Application Tracking System (ATS):** An ATS is used by companies to streamline the hiring process, enabling efficient management of job applications, sorting resumes, and assessing candidate fit for open roles. Our ATS will integrate various functions, including:
- **Resume Parsing:** Extracting structured information from resumes, such as contact details, skills, education, and experience.
- **Job Matching:** Identifying the best candidates for specific roles based on their qualifications and experience.
- **Candidate Scoring:** Ranking applicants based on relevancy and compatibility with job requirements.

Transfer Learning:

Transfer learning allows us to use a pre-trained model to improve the performance of our system with minimal labeled data. By using a model that has been pre-trained on large datasets, we can adapt it to our specific ATS needs without training from scratch. This approach is both time-efficient and cost-effective, as it reduces the need for extensive datasets and computational resources.

Gemini Flash:

Gemini Flash is a transfer learning model that specializes in rapid and efficient processing of textual data. It is well-suited for applications where real-time performance is crucial, such as ATS platforms where speed and accuracy in matching candidates to roles are essential. Gemini Flash is pre-trained on vast amounts of language data, making it capable of understanding complex textual information, a key requirement for parsing resumes and job descriptions.

Project Objectives:

- **Efficient Resume Parsing:** Enable the ATS to accurately parse resumes to extract meaningful, structured data points.
- **Intelligent Candidate Matching:** Use transfer learning to match candidates with open roles based on factors such as skills, experience, and education.

- **Automated Scoring System:** Develop a scoring mechanism to rank candidates, helping recruiters prioritize the most qualified applicants.
- **Real-Time Processing:** Leverage Gemini Flash's speed to ensure the ATS can handle large volumes of applications without lag, supporting a seamless experience for both recruiters and applicants.

2. Problem Definition:

The primary problem that this project seeks to address is the inefficiency and inaccuracy in the recruitment process, specifically in the tasks of resume screening, candidate matching, and application management. Traditional Application Tracking Systems (ATS) are often limited in their ability to accurately parse resumes and match candidates to jobs, resulting in a tedious and time-consuming process for recruiters who have to manually sift through large volumes of applications.

Problem Aspects

1. **High Volume of Applications**:
   - Recruiters often receive hundreds to thousands of applications for each job posting. This makes it challenging to quickly identify the most qualified candidates, leading to long hiring times and, potentially, missed opportunities to secure top talent.
2. **Inaccurate Resume Parsing**:
   - Existing ATS platforms may not accurately parse resumes, often struggling with different formats and unstructured data. This can result in missing or incorrect candidate information, which negatively affects the matching and evaluation processes.
3. **Poor Candidate Matching**:
   - Traditional ATS platforms rely on keyword matching, which can be superficial and lead to poor candidate-job fits. This method may overlook highly qualified candidates who do not use exact keywords, while sometimes surfacing less qualified applicants who happen to match certain terms.
4. **Limited Scalability and Speed**:
   - Many ATS systems lack the speed and scalability required to process large amounts of text data in real-time, which can slow down hiring processes and lead to a poor experience for both recruiters and applicants.
5. **Lack of Intelligent Scoring Mechanisms**:
   - Without an effective scoring system, recruiters may find it challenging to prioritize candidates efficiently. This often means that highly qualified candidates might not be identified early, and recruiter time is spent on manually screening applicants rather than focusing on high-potential candidates.

Solution overview

To address these challenges, the ATS developed in this project will focus on:

- **Automating Resume Parsing with High Accuracy**: Extract key information (e.g., skills, experience, education) from various resume formats to create structured, reliable data.
- **Enhancing Candidate Matching through NLP**: Use transfer learning via Gemini Flash to intelligently match candidates to job roles based on their skills and experience, beyond basic keyword matching.
- **Improving Processing Speed and Scalability**: Leverage the high-speed processing capabilities of Gemini Flash to enable real-time parsing and matching, supporting seamless handling of high application volumes.
- **Developing a Robust Scoring System**: Implement a scoring algorithm to rank candidates based on their compatibility with job roles, allowing recruiters to quickly prioritize high-potential applicants.

The project focuses on creating an advanced **Application Tracking System (ATS)** by leveraging **Gemini Flash** through **Transfer Learning** to address key challenges in the recruitment process. The core issues identified include handling high application volumes, improving accuracy in resume parsing, achieving better candidate-job matching, and developing an efficient scoring mechanism for ranking candidates.

3. Review Summary

The proposed solution offers several enhancements over traditional ATS platforms, aiming to make recruitment faster, more accurate, and scalable:

1. **Automated and Accurate Resume Parsing**: By utilizing transfer learning, the system aims to parse resumes with high precision, extracting structured data such as skills, experience, and qualifications, regardless of format.
2. **Intelligent Candidate Matching**: Moving beyond basic keyword matching, the ATS will leverage Gemini Flash's natural language understanding to assess candidate profiles more contextually, resulting in better alignment with job requirements.
3. **Enhanced Processing Speed and Scalability**: With the high-speed processing capabilities of Gemini Flash, the ATS can handle large volumes of applications in real-time, improving recruiter and candidate experience.
4. **Prioritization through Candidate Scoring**: The system will implement a scoring mechanism to rank candidates based on job fit, helping recruiters prioritize high-potential applicants and streamline the hiring process.

Key Benefits

This project stands out for its focus on efficiency, accuracy, and real-time processing. By addressing the limitations of traditional ATS systems, it has the potential to significantly reduce the time and resources required for hiring, while improving the quality of candidate selection.

Overall, this project presents a comprehensive approach to revolutionizing recruitment by incorporating advanced NLP techniques and leveraging the speed of Gemini Flash for real-time, intelligent application tracking and candidate evaluation.

4. Goals and Objectives

The primary goal of this project is to develop an Application Tracking System (ATS) that leverages Gemini Flash's transfer learning capabilities to streamline the recruitment process. By enhancing resume parsing, candidate matching, and scoring accuracy, this ATS aims to save recruiters time, improve candidate-job fit, and optimize application management.

1. Goal: Achieve Accurate and Automated Resume Parsing

Objectives:

- Design an ATS that can parse various resume formats with high accuracy.
- Use Gemini Flash to extract structured data points, including skills, experience, education, and certifications.
- Ensure reliable data extraction across diverse resume styles, reducing manual data correction efforts for recruiters.

2. Goal: Improve Candidate Matching with NLP and Transfer Learning

Objectives:

- Implement NLP-driven candidate matching that goes beyond keyword matching to consider context, synonyms, and nuanced job requirements.
- Train the system using Gemini Flash's transfer learning capabilities to understand candidate profiles and job descriptions more deeply.
- Ensure the ATS can recommend candidates who truly align with the role requirements, helping recruiters identify strong matches more effectively.

3. Goal: Develop a Real-Time and Scalable ATS

Objectives:

- Leverage Gemini Flash's high-speed processing to handle large volumes of applications without lag, making the ATS suitable for organizations of various sizes.
- Optimize the system's infrastructure to maintain processing speed and efficiency even as application volumes increase.

- Enable seamless handling of real-time data updates to provide recruiters with immediate, up-to-date insights on candidate pools.

4. Goal: Implement an Effective Candidate Scoring System

Objectives:

- Develop a scoring algorithm that assesses candidates based on compatibility with the job role, experience, and skills.
- Provide a ranked list of candidates, making it easier for recruiters to prioritize applicants with the highest potential.
- Allow customizable scoring criteria so that recruiters can adjust the system's scoring emphasis based on specific role requirements.

5. Goal: Enhance Recruiter and Candidate Experience

Objectives:

- Design a user-friendly interface for recruiters to navigate applications, view candidate details, and utilize the scoring and matching features.
- Minimize manual tasks for recruiters by automating as much of the screening, matching, and ranking process as possible.
- Ensure data security and privacy to protect candidate information, a critical factor in building trust with applicants and complying with data protection regulations.

Success Metrics

To measure success, the following metrics will be tracked:

- **Resume Parsing Accuracy:** Percentage accuracy in parsing information across various formats.
- **Candidate Matching Precision:** Improvement in candidate-to-job matching quality, assessed through recruiter feedback and successful hires.
- **Processing Speed:** Average time to process a batch of applications under different loads.
- **Recruiter Satisfaction:** Survey-based feedback from recruiters on ATS usability and performance.
- **Candidate Scoring Reliability:** Effectiveness of scoring algorithm in helping recruiters identify top candidates quickly.

By achieving these objectives, this ATS will create a faster, more accurate, and scalable solution that transforms recruitment processes, supporting companies in efficiently finding the best talent.

5. Methodologies

This project will follow a structured approach, incorporating machine learning, natural language processing, and software engineering principles to develop the Application Tracking System (ATS). The methodologies can be broken down into several stages, each focusing on specific functionalities of the ATS to achieve accurate parsing, matching, and scoring of candidates using Gemini Flash's transfer learning capabilities.

1. Data Collection and Preprocessing

- **Data Gathering**: Collect a diverse set of resumes and job descriptions from various sources to cover a range of formats, industries, and skill levels. This data will serve as the basis for training and testing the ATS.
- **Data Cleaning**: Preprocess resumes and job descriptions to remove inconsistencies, redundancies, and irrelevant content (e.g., formatting characters). This ensures the model receives clean input for accurate parsing.
- **Data Annotation**: Label key resume components (such as skills, experience, education) and job description attributes (such as required skills, experience levels) to prepare a high-quality dataset. This labeled data will be essential for training and fine-tuning the model.

2. Transfer Learning with Gemini Flash

- **Model Selection**: Use Gemini Flash, a highly efficient NLP model pre-trained on extensive language data, as the foundation for parsing and matching text. Its transfer learning capabilities allow adaptation with minimal additional data.
- **Fine-Tuning on Domain-Specific Data**: Fine-tune Gemini Flash on domain-specific resume and job description data to optimize the model for ATS functionalities. Fine-tuning will involve training the model on labeled data to improve its ability to accurately parse resumes and match candidates to job roles.
- **Transfer Learning Techniques**:
  - **Feature Extraction**: Leverage pre-trained features from Gemini Flash to process resumes and job descriptions, thus reducing the need for extensive labeled data.
  - **Layer-Freezing**: Freeze the earlier layers of Gemini Flash (which capture general language patterns) and fine-tune only the later layers on our specific dataset. This allows the model to retain its general language understanding while adapting to our ATS needs.

3. Natural Language Processing for Resume Parsing

- **Entity Recognition and Classification**: Utilize NLP techniques like Named Entity Recognition (NER) to extract structured data such as names, skills, experiences, and education from resumes. NER models, fine-tuned on Gemini Flash, will identify specific segments of text as relevant entities.

- **Skill Extraction and Matching**: Use phrase embedding and semantic similarity algorithms to identify and categorize relevant skills. This includes accounting for synonyms and related terms, enabling more effective candidate-job matching.
- **Experience and Qualification Extraction**: Build custom classifiers to identify and rank qualifications (e.g., years of experience, education levels), which will be used to prioritize candidates based on job requirements.

4. Intelligent Candidate Matching

- **Contextual Matching**: Use Gemini Flash's contextual understanding to go beyond simple keyword matching by evaluating the overall relevance of a candidate's experience and skills to the job description.
- **Semantic Similarity Scoring**: Apply semantic similarity algorithms to compare resumes and job descriptions, focusing on key qualifications and experience. This will ensure that candidates who meet role requirements—even if they don't use exact keywords—are matched appropriately.
- **Candidate Ranking Algorithm**: Develop a matching and ranking algorithm that scores candidates based on their fit for the job role. This will integrate semantic similarity scores with other factors (e.g., experience and education levels) to provide a comprehensive assessment.

5. Candidate Scoring System Development

- **Scoring Model**: Create a scoring model that uses weighted criteria (skills, experience, education) to rank candidates. Each candidate receives a score that reflects their alignment with the job requirements.
- **Customizable Weighting**: Allow recruiters to adjust the scoring weights based on specific job needs. For example, some roles may prioritize specific skills over years of experience.
- **Scoring Calibration**: Use feedback loops from recruiters to continually refine and calibrate the scoring model, ensuring that it meets practical hiring requirements and adapts to evolving job needs.

6. System Scalability and Real-Time Processing

- **System Architecture**: Design the system with a scalable architecture that supports real-time processing and integration with existing ATS platforms. This may involve a microservices architecture that can handle independent functions like parsing, matching, and scoring in parallel.
- **Batch and Real-Time Processing**: Enable both batch processing (for handling large volumes of applications) and real-time processing (for individual applications), ensuring the system can scale to meet different recruiter needs.

- **Optimization of Processing Pipeline**: Implement load balancing, caching, and efficient data handling techniques to ensure the ATS maintains high processing speeds even with high application volumes.

7. User Interface (UI) and User Experience (UX) Design

- **Dashboard for Recruiters**: Develop a user-friendly dashboard where recruiters can view parsed resume data, candidate scores, and match recommendations. The UI will be designed for easy navigation, enabling quick access to key functions.
- **Search and Filtering**: Provide recruiters with filtering and search options to quickly find candidates who meet specific criteria, based on parsed resume data and scoring.
- **Candidate Comparison**: Implement a side-by-side comparison feature to allow recruiters to compare top candidates and make informed decisions based on their scores and matching attributes.

8. Testing and Evaluation

- **Performance Testing**: Test the system's processing speed, scalability, and load-handling capabilities to ensure it can efficiently manage high volumes of applications.
- **Accuracy Evaluation**: Measure parsing accuracy, candidate matching precision, and scoring reliability against a labeled test set to validate the system's effectiveness.
- **User Feedback Collection**: Conduct user testing with recruiters to gather feedback on ATS usability, parsing accuracy, and match quality. This feedback will inform further refinements and updates.

9. Continuous Improvement and Model Retraining

- **Iterative Model Improvements**: Regularly retrain and fine-tune the model with updated data to ensure it adapts to new resume formats, job requirements, and industry-specific terminology.
- **Feedback Loop**: Integrate a feedback mechanism where recruiters can rate the relevance of candidate matches and scoring accuracy, allowing the system to continually refine its performance.
- **System Monitoring and Updates**: Monitor system performance and model effectiveness, ensuring timely updates and optimizations based on evolving industry needs.

By combining transfer learning, NLP, and a scalable architectural approach, this ATS methodology ensures high accuracy, speed, and user-friendly functionality, providing recruiters with a powerful tool for effective and efficient candidate selection.

6. Tools and Technologies

To build this Application Tracking System (ATS) with transfer learning capabilities using Gemini Flash, a robust set of tools and technologies is required. These tools will facilitate data handling, model development, and deployment, as well as provide a user-friendly interface for recruiters.

1. Natural Language Processing (NLP)

- **Gemini Flash:** The central NLP model, chosen for its speed and efficiency, particularly well-suited for text-heavy tasks such as resume parsing and candidate-job matching.
- **spaCy:** A popular NLP library that provides pre-trained models, Named Entity Recognition (NER), and tokenization functionalities. Useful for supplementary NLP tasks like text preprocessing and entity extraction.
- **NLTK (Natural Language Toolkit):** A toolkit with various text processing libraries, ideal for tokenization, stemming, and basic NLP tasks that support Gemini Flash's functionalities.

2. Machine Learning Frameworks

- **PyTorch or TensorFlow:** Frameworks for fine-tuning Gemini Flash, implementing the transfer learning process, and building custom scoring algorithms.
- **Hugging Face Transformers:** Offers a variety of pre-trained models, including Gemini Flash, and provides a user-friendly interface for model fine-tuning, adaptation, and deployment.
- **Scikit-learn:** Useful for additional machine learning tasks like classification, clustering, and evaluation metrics for candidate scoring and matching algorithms.

3. Data Processing and Storage

- **Pandas:** For data manipulation and processing, especially useful when handling large datasets of resumes and job descriptions.
- **NumPy:** For numerical operations that facilitate efficient data handling and matrix operations within machine learning algorithms.

4. Deployment and Hosting:

- **Streamlit Sharing**:
  Role: Hosting the final web application.
  Use Cases: Quickly deploy the ATS to the cloud for user access and interaction.

By using **Streamlit** for the interface, **Gemini Flash** for NLP and transfer learning, and a range of additional tools and libraries for machine learning, backend development, and deployment, this ATS system will be capable of automating resume parsing, candidate-job matching, and ranking, all within an intuitive and responsive web application. This technology stack ensures both scalability and efficiency in handling large recruitment data while delivering an easy-to-use interface for recruiters.

7. Hardware and Software Specifications

For the successful development, deployment, and operation of the **Application Tracking System (ATS)** using **Gemini Flash** and **Streamlit**, we need to define both hardware and software specifications. These requirements ensure that the system can handle the heavy processing needs of natural language processing (NLP), machine learning model training, real-time resume parsing, and candidate matching, while providing a smooth experience for users (recruiters).

Hardware Specifications

The hardware specifications depend on the scale of the application, the size of the data, and the deployment requirements. Below are general recommendations for both development and production environments.

- **Processor (CPU)**:
  - **Recommended**: Intel Core i7 or AMD Ryzen 7 (or higher) with at least 8 cores.
  - **Minimum**: Intel Core i5 or AMD Ryzen 5 with 4 cores.
  - **Reasoning**: NLP and machine learning tasks (like fine-tuning models) require significant processing power, especially when handling large datasets and training models.
- **Memory (RAM)**:
  - **Recommended**: 16 GB or more.
  - **Minimum**: 8 GB.
  - **Reasoning**: Machine learning model training and large-scale data processing can be memory-intensive, particularly when fine-tuning Gemini Flash or loading large datasets like resumes and job descriptions.
- **Graphics Processing Unit (GPU)**:
  - **Recommended**: NVIDIA RTX 3060 or higher for machine learning tasks.
  - **Minimum**: NVIDIA GTX 1650 or equivalent.
  - **Reasoning**: Training deep learning models like Gemini Flash requires GPU acceleration. GPUs are especially useful for faster model training and fine-tuning, reducing processing times significantly.
- **Storage**:
  - **Recommended**: 512 GB SSD (Solid-State Drive) or larger.
  - **Minimum**: 256 GB SSD.
  - **Reasoning**: SSDs provide fast read/write speeds, essential for loading large datasets and machine learning models, enabling faster development iterations and data processing.

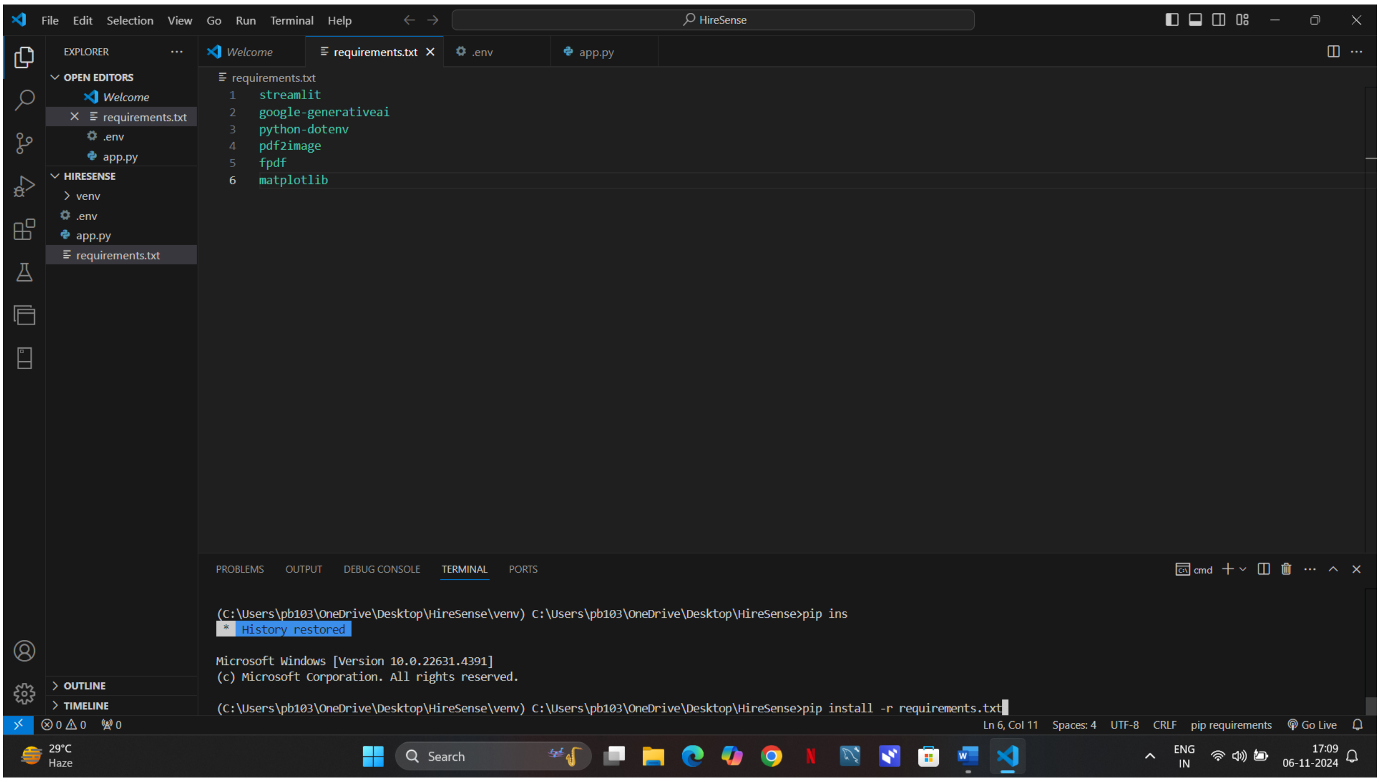
- **Networking**:
  - **Recommended**: High-speed Internet (1 Gbps or higher) for real-time data processing and smooth user interaction.
  - **Minimum**: 100 Mbps (for lower-scale applications).
  - **Reasoning**: Fast and reliable networking ensures minimal latency, especially for cloud-hosted ATS systems where real-time processing and updates are key.

Software Specifications
Below is a breakdown of the software stack required to build and deploy the ATS.

- **Operating System**:
  - **Recommended**: Linux (Ubuntu 20.04 or newer), macOS, or Windows 10/11.
  - **Reasoning**: Linux and macOS are widely preferred for machine learning and data science development due to better support for open-source libraries. Windows is also compatible but may require additional configurations for certain tools.
- **Python**:
  - **Recommended**: Python 3.8 or higher.
  - **Reasoning**: Python is the primary programming language for NLP and machine learning tasks, as well as for building the Streamlit application. Streamlit and most NLP libraries (e.g., Hugging Face, SpaCy) are fully compatible with Python 3.8+.
- **Libraries and Frameworks**:
  - **Machine Learning and NLP**:
    - **TensorFlow** or **PyTorch** (for model training and fine-tuning)
    - **Hugging Face Transformers** (for integrating pre-trained models like Gemini Flash)
    - **SpaCy** (for NLP tasks such as tokenization, NER, etc.)
    - **Scikit-learn** (for simple machine learning models, feature extraction, and evaluation)
  - **Web Development**:
    - **Streamlit** (for building the user interface and handling interactions)
    - **Flask** or **FastAPI** (optional, for serving the model through APIs)
  - **Data Storage and Management**:
    - **SQLite** (for lightweight local databases)
    - **PostgreSQL** or **MySQL** (for more robust cloud-based databases)
  - **Visualization**:
    - **Plotly**, **Matplotlib**, or **Seaborn** (for visualizing results like candidate rankings)

- **Integrated Development Environment (IDE)**:
  - **Recommended**: VS Code, PyCharm, or Jupyter Notebooks.
  - **Reasoning**: These IDEs provide excellent support for Python development and are compatible with data science libraries.
- **Version Control**:
  - **Git** for version control to manage codebase changes and collaborate with teams.
- **Containerization and Virtual Environments**:
  - **Docker** for containerizing the application and ensuring consistency across development and production environments.
  - **Virtual env** or **Conda** for managing Python dependencies and isolating environments.

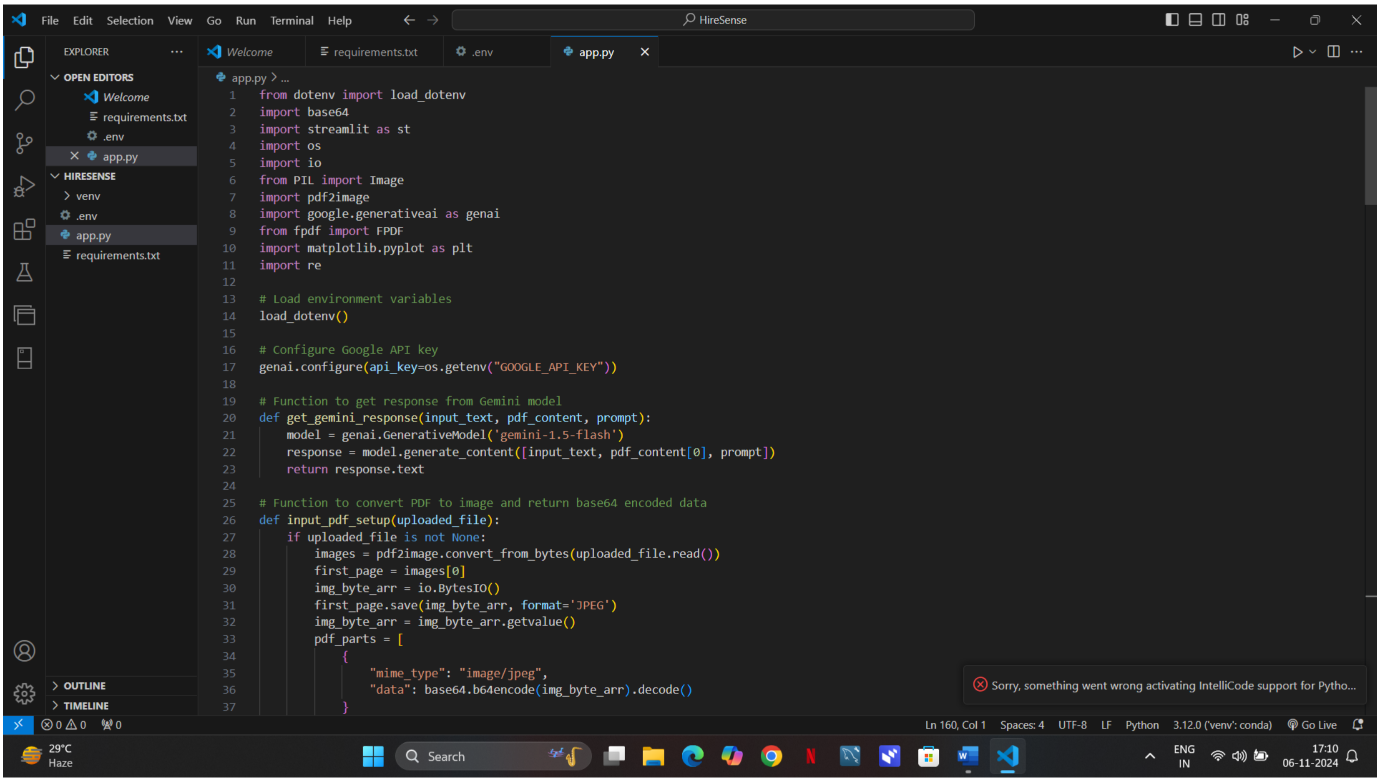
8. Implementation :



requirements.txt

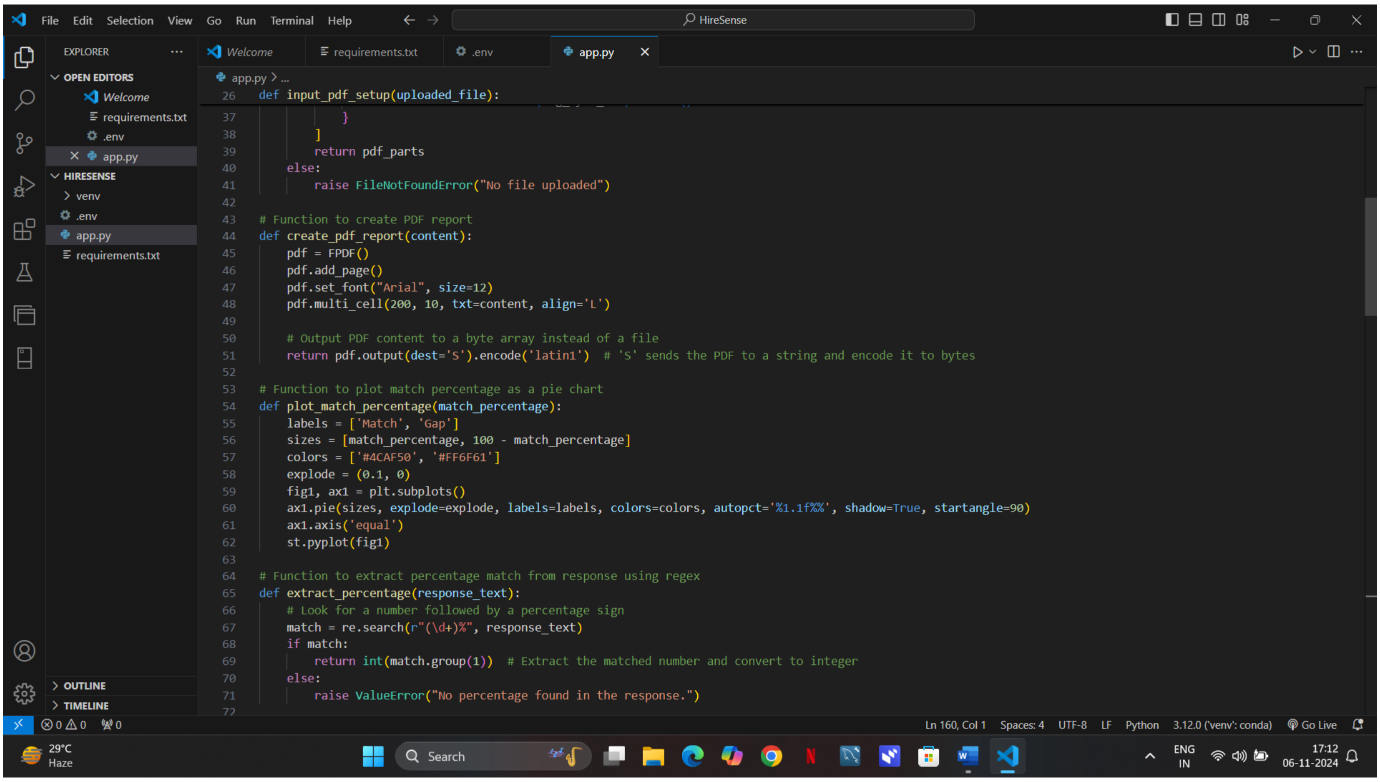
```
streamlit
google-generativeai
python-dotenv
pdf2image
fpdf
matplotlib
```

```
(C:\Users\pb103\OneDrive\Desktop\HireSense\venv) C:\Users\pb103\OneDrive\Desktop\HireSense>pip ins
 *  History restored

Microsoft Windows [Version 10.0.22631.4391]
(c) Microsoft Corporation. All rights reserved.

(C:\Users\pb103\OneDrive\Desktop\HireSense\venv) C:\Users\pb103\OneDrive\Desktop\HireSense>pip install -r requirements.txt
```

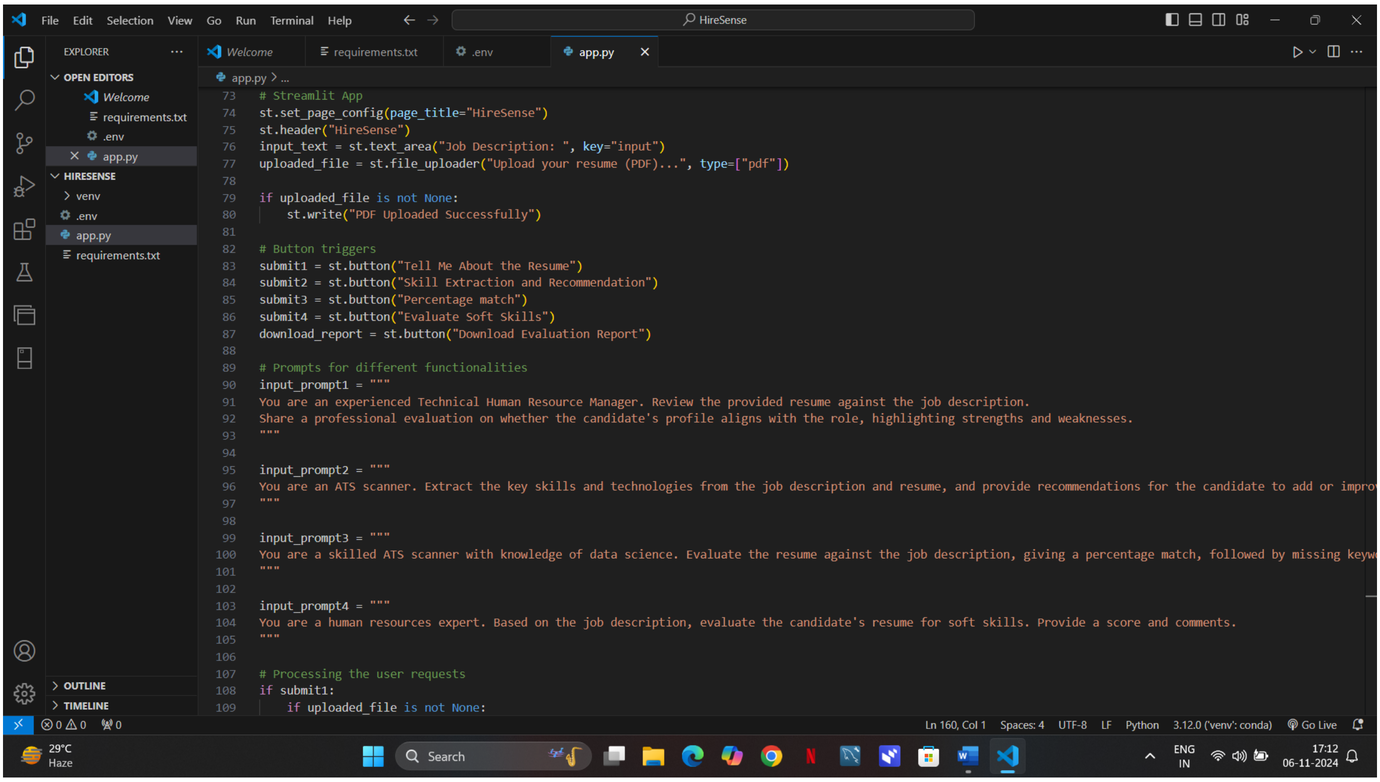


```python
from dotenv import load_dotenv
import base64
import streamlit as st
import os
import io
from PIL import Image
import pdf2image
import google.generativeai as genai
from fpdf import FPDF
import matplotlib.pyplot as plt
import re

# Load environment variables
load_dotenv()

# Configure Google API key
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

# Function to get response from Gemini model
def get_gemini_response(input_text, pdf_content, prompt):
    model = genai.GenerativeModel('gemini-1.5-flash')
    response = model.generate_content([input_text, pdf_content[0], prompt])
    return response.text

# Function to convert PDF to image and return base64 encoded data
def input_pdf_setup(uploaded_file):
    if uploaded_file is not None:
        images = pdf2image.convert_from_bytes(uploaded_file.read())
        first_page = images[0]
        img_byte_arr = io.BytesIO()
        first_page.save(img_byte_arr, format='JPEG')
        img_byte_arr = img_byte_arr.getvalue()
        pdf_parts = [
            {
                "mime_type": "image/jpeg",
                "data": base64.b64encode(img_byte_arr).decode()
            }
```

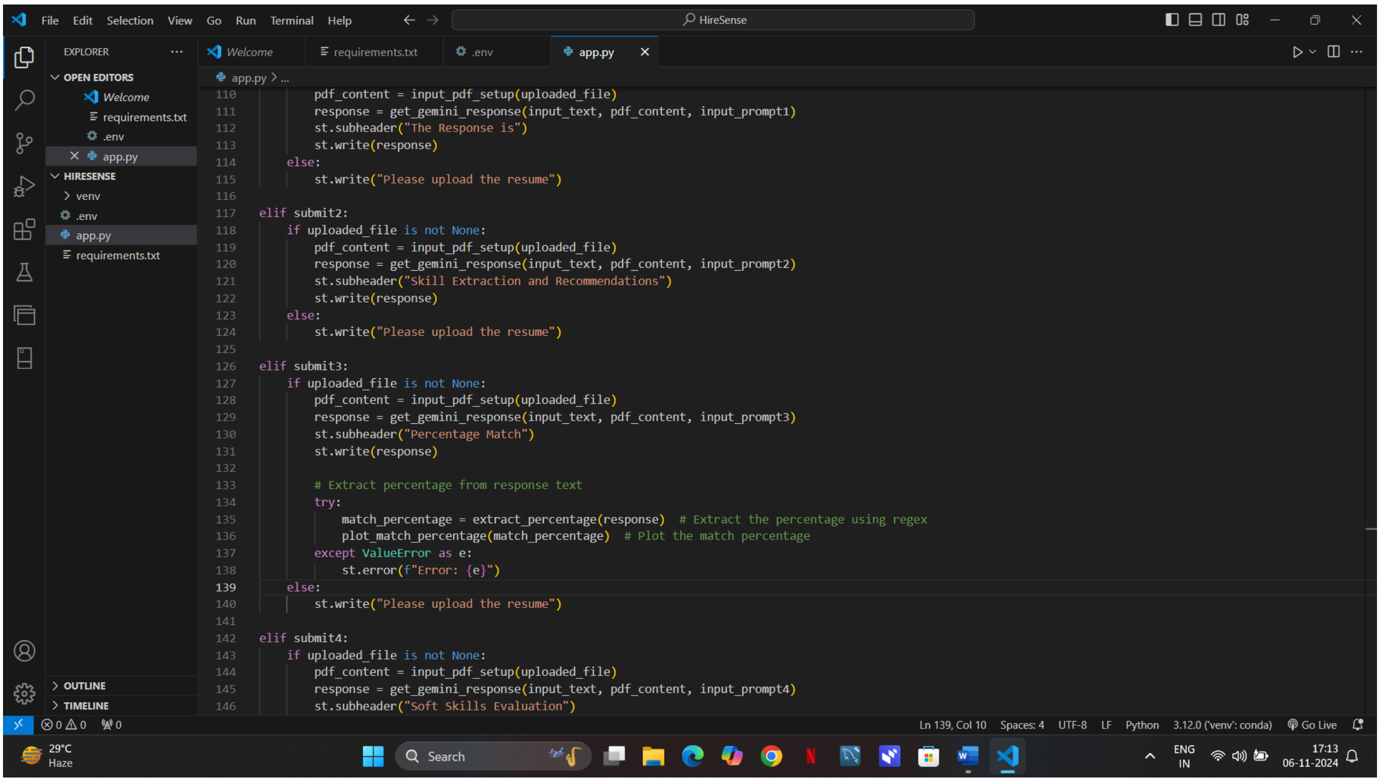
Sorry, something went wrong activating IntelliCode support for Pytho...



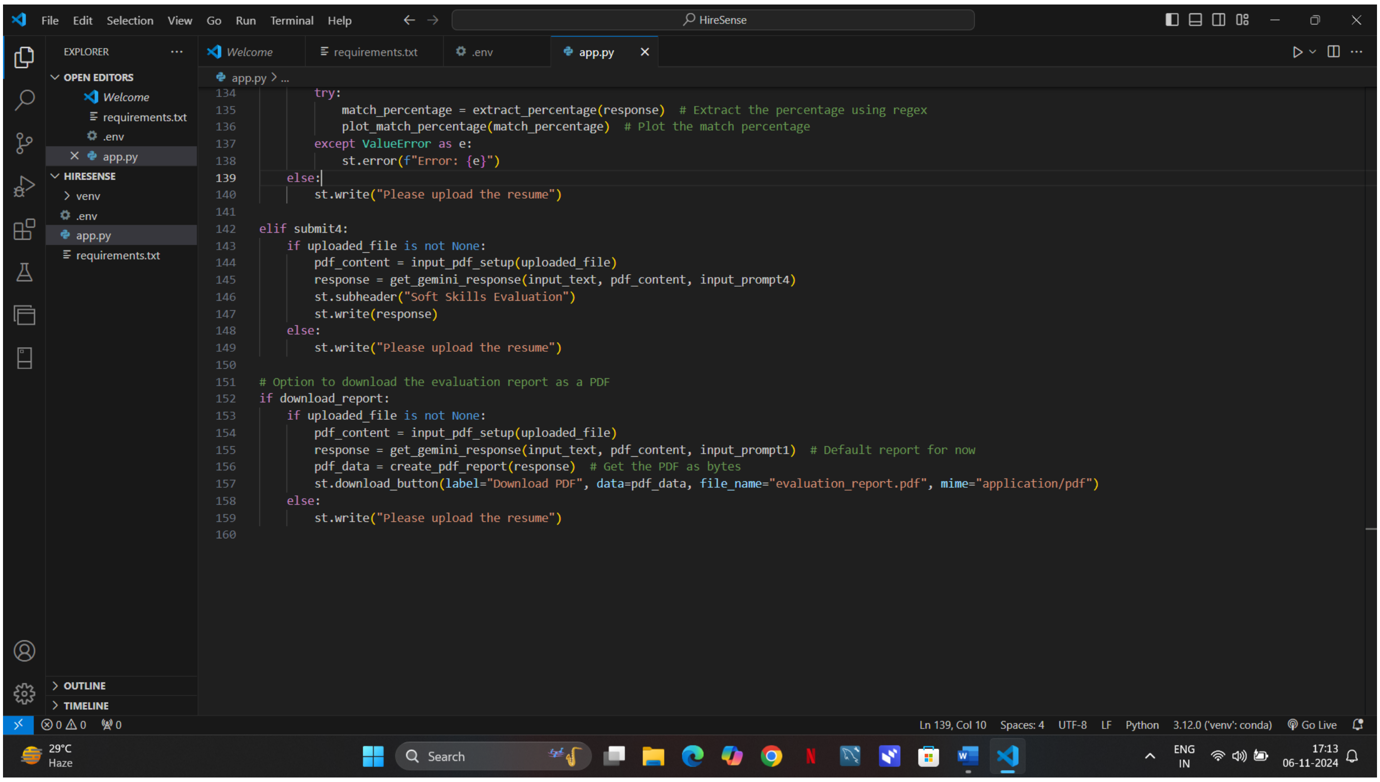
```python
def input_pdf_setup(uploaded_file):
            }
        ]
        return pdf_parts
    else:
        raise FileNotFoundError("No file uploaded")

# Function to create PDF report
def create_pdf_report(content):
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=12)
    pdf.multi_cell(200, 10, txt=content, align='L')

    # Output PDF content to a byte array instead of a file
    return pdf.output(dest='S').encode('latin1')  # 'S' sends the PDF to a string and encode it to bytes

# Function to plot match percentage as a pie chart
def plot_match_percentage(match_percentage):
    labels = ['Match', 'Gap']
    sizes = [match_percentage, 100 - match_percentage]
    colors = ['#4CAF50', '#FF6F61']
    explode = (0.1, 0)
    fig1, ax1 = plt.subplots()
    ax1.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=90)
    ax1.axis('equal')
    st.pyplot(fig1)

# Function to extract percentage match from response using regex
def extract_percentage(response_text):
    # Look for a number followed by a percentage sign
    match = re.search(r"(\d+)%", response_text)
    if match:
        return int(match.group(1))  # Extract the matched number and convert to integer
    else:
        raise ValueError("No percentage found in the response.")
```



```python
# Streamlit App
st.set_page_config(page_title="HireSense")
st.header("HireSense")
input_text = st.text_area("Job Description: ", key="input")
uploaded_file = st.file_uploader("Upload your resume (PDF)...", type=["pdf"])

if uploaded_file is not None:
    st.write("PDF Uploaded Successfully")

# Button triggers
submit1 = st.button("Tell Me About the Resume")
submit2 = st.button("Skill Extraction and Recommendation")
submit3 = st.button("Percentage match")
submit4 = st.button("Evaluate Soft Skills")
download_report = st.button("Download Evaluation Report")

# Prompts for different functionalities
input_prompt1 = """
You are an experienced Technical Human Resource Manager. Review the provided resume against the job description.
Share a professional evaluation on whether the candidate's profile aligns with the role, highlighting strengths and weaknesses.
"""

input_prompt2 = """
You are an ATS scanner. Extract the key skills and technologies from the job description and resume, and provide recommendations for the candidate to add or impro
"""

input_prompt3 = """
You are a skilled ATS scanner with knowledge of data science. Evaluate the resume against the job description, giving a percentage match, followed by missing keyw
"""

input_prompt4 = """
You are a human resources expert. Based on the job description, evaluate the candidate's resume for soft skills. Provide a score and comments.
"""

# Processing the user requests
if submit1:
    if uploaded_file is not None:
```

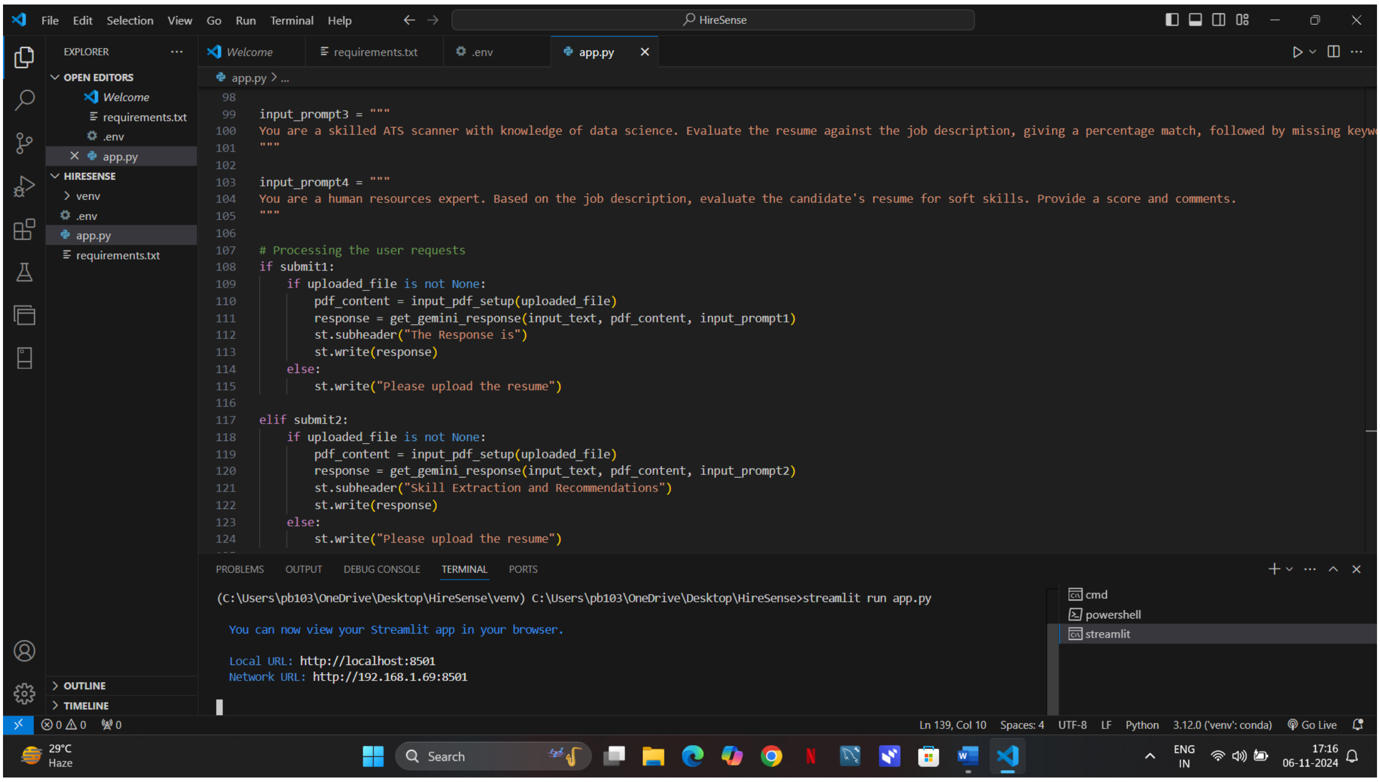


```python
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt1)
        st.subheader("The Response is")
        st.write(response)
    else:
        st.write("Please upload the resume")

elif submit2:
    if uploaded_file is not None:
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt2)
        st.subheader("Skill Extraction and Recommendations")
        st.write(response)
    else:
        st.write("Please upload the resume")

elif submit3:
    if uploaded_file is not None:
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt3)
        st.subheader("Percentage Match")
        st.write(response)

        # Extract percentage from response text
        try:
            match_percentage = extract_percentage(response)  # Extract the percentage using regex
            plot_match_percentage(match_percentage)  # Plot the match percentage
        except ValueError as e:
            st.error(f"Error: {e}")
    else:
        st.write("Please upload the resume")

elif submit4:
    if uploaded_file is not None:
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt4)
        st.subheader("Soft Skills Evaluation")
```



```python
            try:
                match_percentage = extract_percentage(response)  # Extract the percentage using regex
                plot_match_percentage(match_percentage)  # Plot the match percentage
            except ValueError as e:
                st.error(f"Error: {e}")
        else:
            st.write("Please upload the resume")

    elif submit4:
        if uploaded_file is not None:
            pdf_content = input_pdf_setup(uploaded_file)
            response = get_gemini_response(input_text, pdf_content, input_prompt4)
            st.subheader("Soft Skills Evaluation")
            st.write(response)
        else:
            st.write("Please upload the resume")

    # Option to download the evaluation report as a PDF
    if download_report:
        if uploaded_file is not None:
            pdf_content = input_pdf_setup(uploaded_file)
            response = get_gemini_response(input_text, pdf_content, input_prompt1)  # Default report for now
            pdf_data = create_pdf_report(response)  # Get the PDF as bytes
            st.download_button(label="Download PDF", data=pdf_data, file_name="evaluation_report.pdf", mime="application/pdf")
        else:
            st.write("Please upload the resume")
```

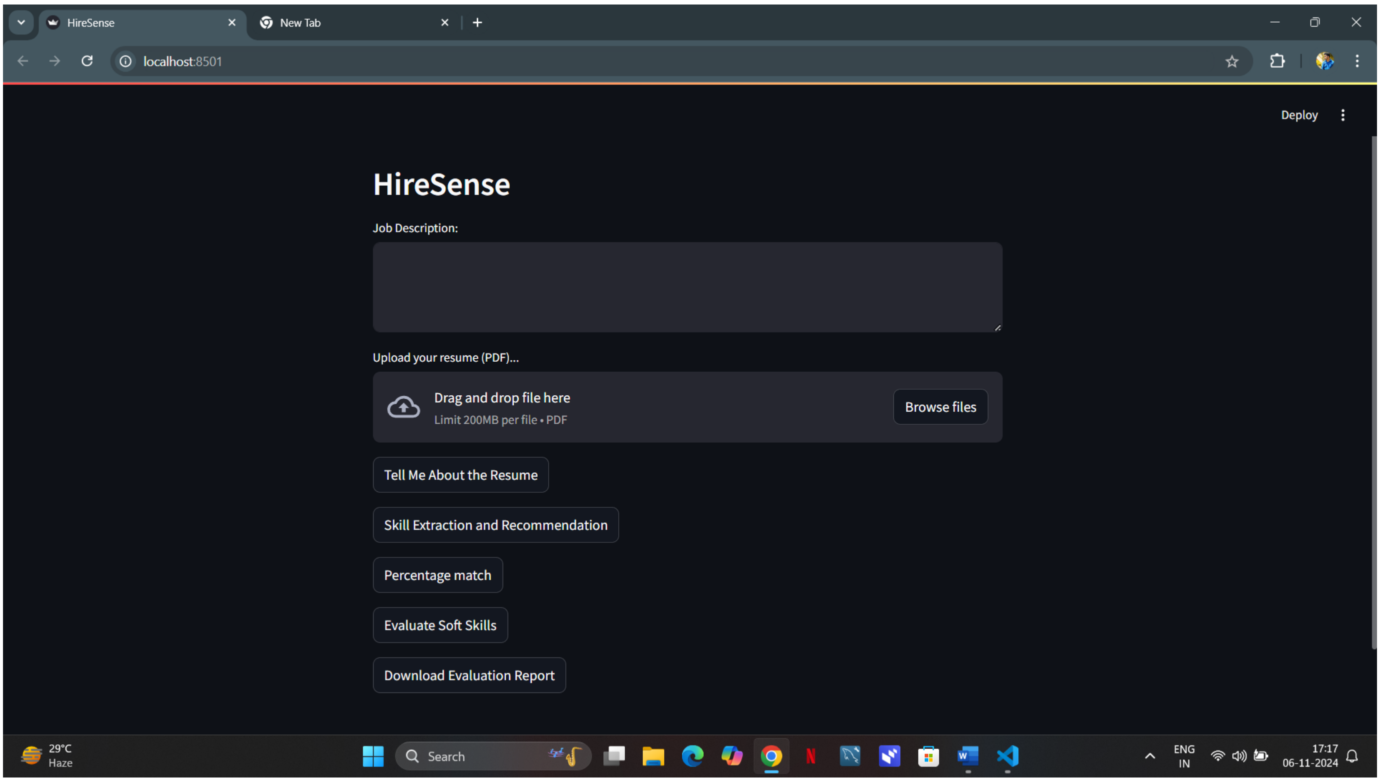
9. Results:



```python
input_prompt3 = """
You are a skilled ATS scanner with knowledge of data science. Evaluate the resume against the job description, giving a percentage match, followed by missing keyw
"""

input_prompt4 = """
You are a human resources expert. Based on the job description, evaluate the candidate's resume for soft skills. Provide a score and comments.
"""

# Processing the user requests
if submit1:
    if uploaded_file is not None:
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt1)
        st.subheader("The Response is")
        st.write(response)
    else:
        st.write("Please upload the resume")

elif submit2:
    if uploaded_file is not None:
        pdf_content = input_pdf_setup(uploaded_file)
        response = get_gemini_response(input_text, pdf_content, input_prompt2)
        st.subheader("Skill Extraction and Recommendations")
        st.write(response)
    else:
        st.write("Please upload the resume")
```

```
(C:\Users\pb103\OneDrive\Desktop\HireSense\venv) C:\Users\pb103\OneDrive\Desktop\HireSense>streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.69:8501
```

10. Output:

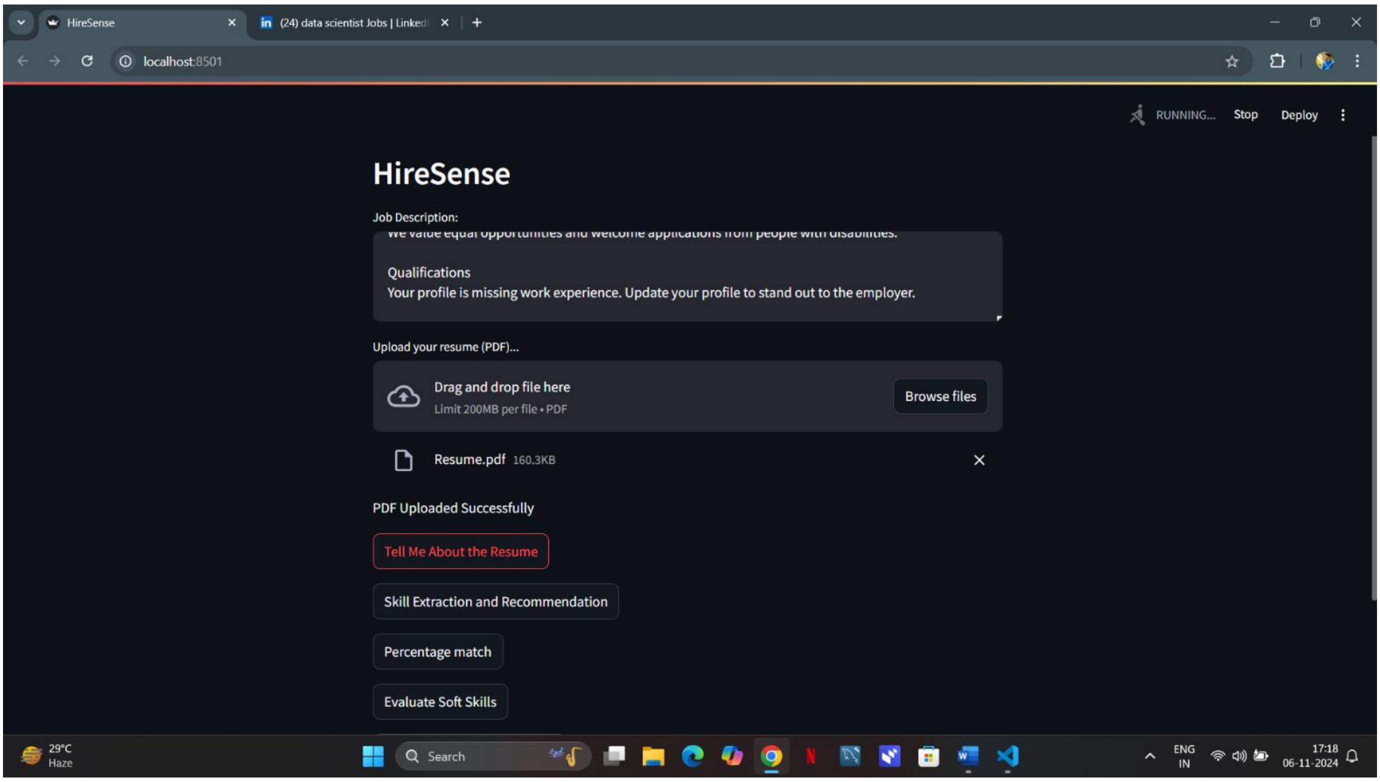


HireSense

Job Description:

Upload your resume (PDF)...

Drag and drop file here
Limit 200MB per file • PDF



HireSense

Job Description:

We value equal opportunities and welcome applications from people with disabilities.

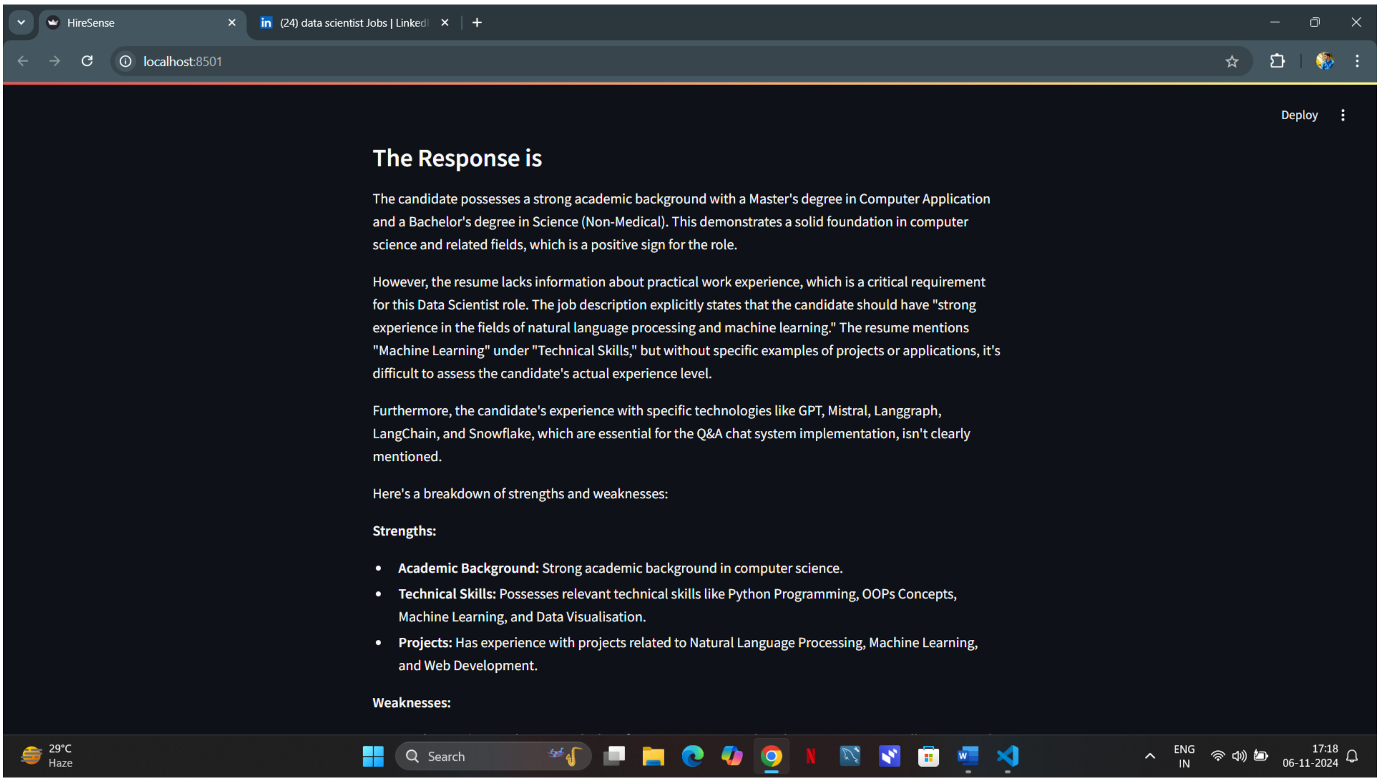
Qualifications
Your profile is missing work experience. Update your profile to stand out to the employer.

Upload your resume (PDF)...

Drag and drop file here
Limit 200MB per file • PDF

Resume.pdf 160.3KB

PDF Uploaded Successfully



The Response is

The candidate possesses a strong academic background with a Master's degree in Computer Application and a Bachelor's degree in Science (Non-Medical). This demonstrates a solid foundation in computer science and related fields, which is a positive sign for the role.

However, the resume lacks information about practical work experience, which is a critical requirement for this Data Scientist role. The job description explicitly states that the candidate should have "strong experience in the fields of natural language processing and machine learning." The resume mentions "Machine Learning" under "Technical Skills," but without specific examples of projects or applications, it's difficult to assess the candidate's actual experience level.

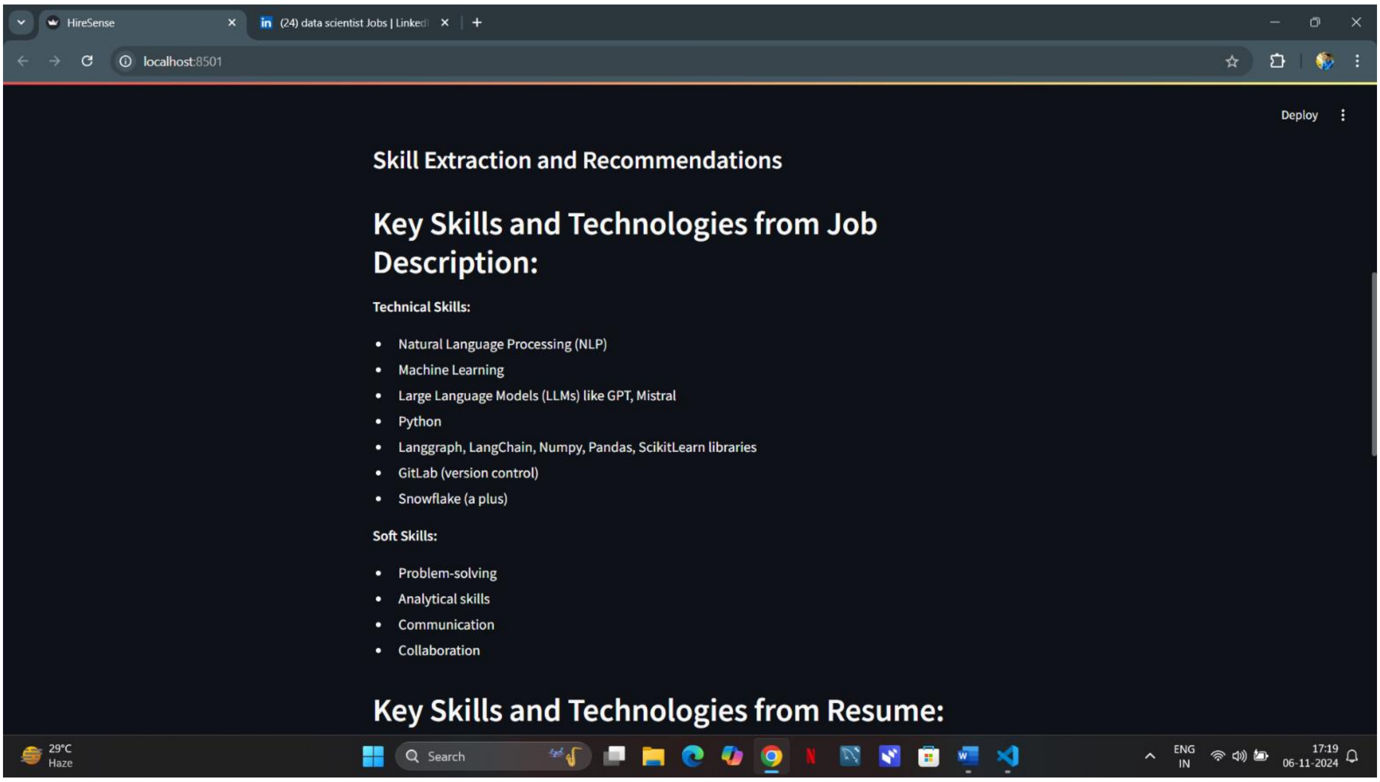
Furthermore, the candidate's experience with specific technologies like GPT, Mistral, Langgraph, LangChain, and Snowflake, which are essential for the Q&A chat system implementation, isn't clearly mentioned.

Here's a breakdown of strengths and weaknesses:

Strengths:

- **Academic Background:** Strong academic background in computer science.
- **Technical Skills:** Possesses relevant technical skills like Python Programming, OOPs Concepts, Machine Learning, and Data Visualisation.
- **Projects:** Has experience with projects related to Natural Language Processing, Machine Learning, and Web Development.

Weaknesses:



Skill Extraction and Recommendations

Key Skills and Technologies from Job Description:

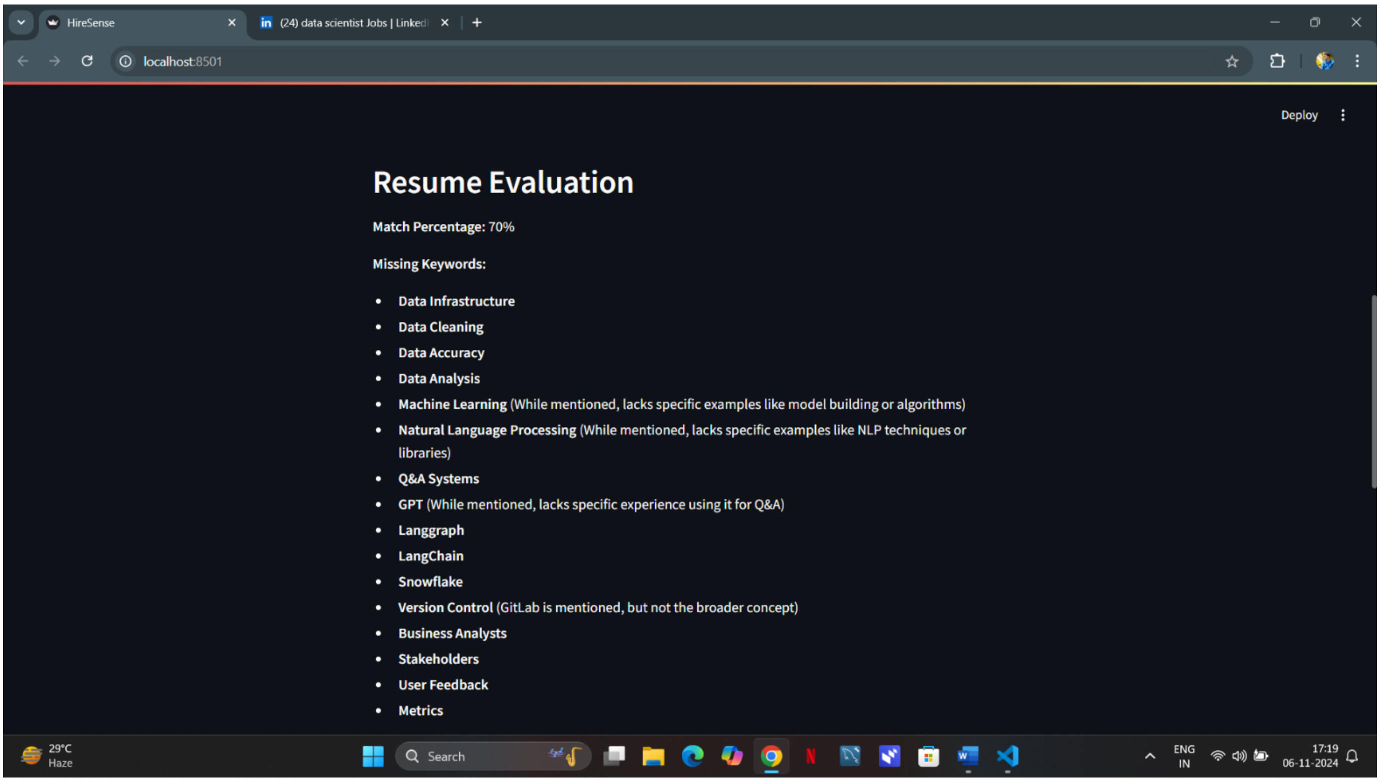
Technical Skills:

- Natural Language Processing (NLP)
- Machine Learning
- Large Language Models (LLMs) like GPT, Mistral
- Python
- Langgraph, LangChain, Numpy, Pandas, ScikitLearn libraries
- GitLab (version control)
- Snowflake (a plus)

Soft Skills:

- Problem-solving
- Analytical skills
- Communication
- Collaboration

Key Skills and Technologies from Resume:

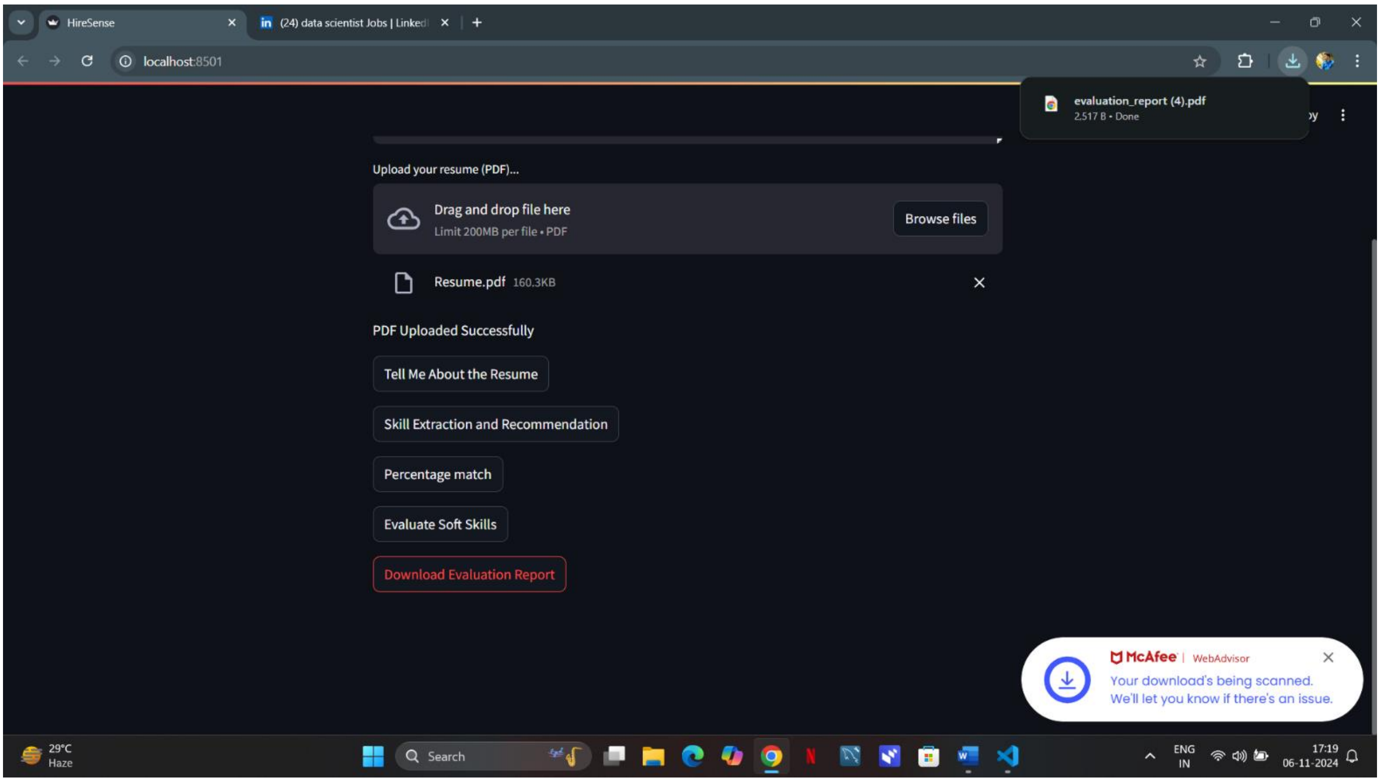


Resume Evaluation

Match Percentage: 70%

Missing Keywords:

- **Data Infrastructure**
- **Data Cleaning**
- **Data Accuracy**
- **Data Analysis**
- **Machine Learning** (While mentioned, lacks specific examples like model building or algorithms)
- **Natural Language Processing** (While mentioned, lacks specific examples like NLP techniques or libraries)
- **Q&A Systems**
- **GPT** (While mentioned, lacks specific experience using it for Q&A)
- **Langgraph**
- **LangChain**
- **Snowflake**
- **Version Control** (GitLab is mentioned, but not the broader concept)
- **Business Analysts**
- **Stakeholders**
- **User Feedback**
- **Metrics**



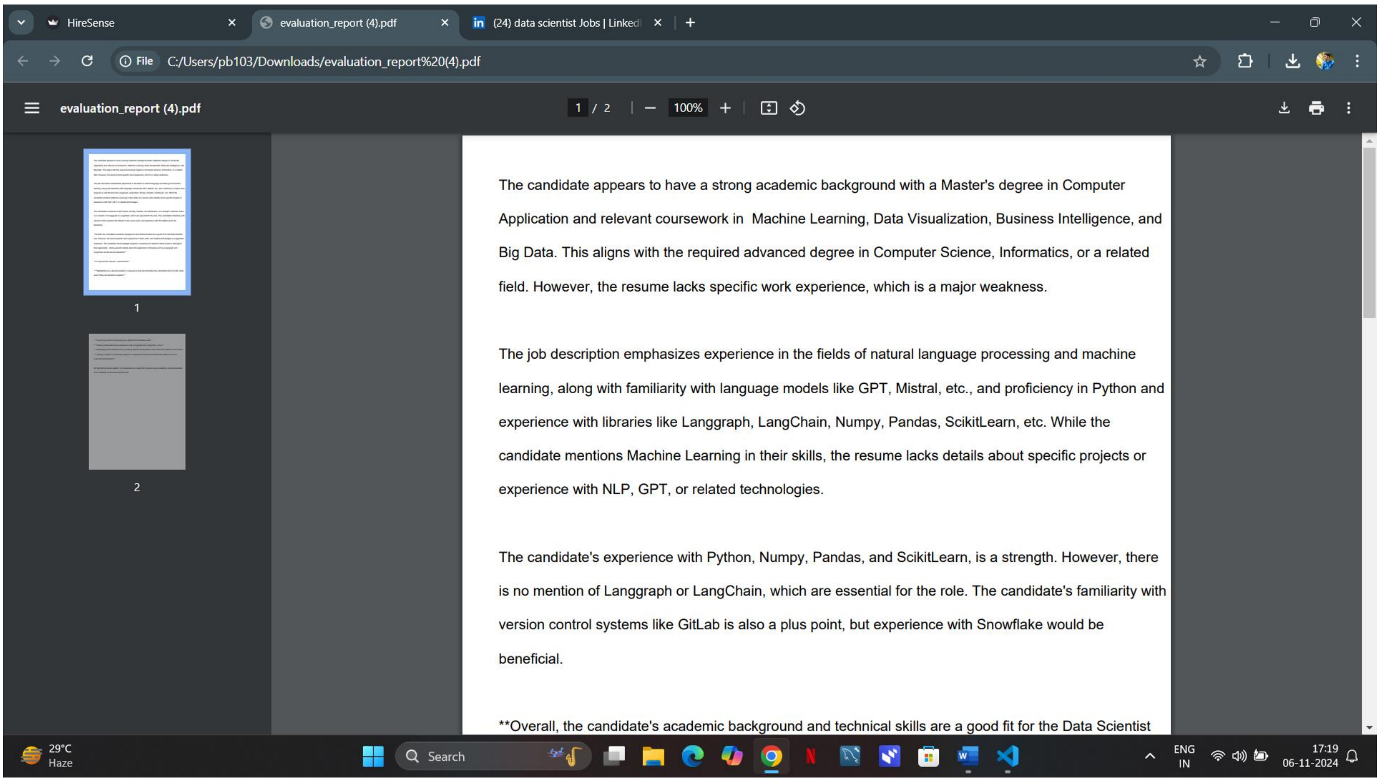
Upload your resume (PDF)...

Drag and drop file here
Limit 200MB per file • PDF

Resume.pdf 160.3KB

PDF Uploaded Successfully

Your download's being scanned.
We'll let you know if there's an issue.


HireSense
evaluation_report (4).pdf
(24) data scientist Jobs | LinkedI
File C:/Users/pb103/Downloads/evaluation_report%20(4).pdf
evaluation_report (4).pdf
1 / 2
100%
1
2
29°C
Haze
Search
ENG
IN
17:19
06-11-2024

The candidate appears to have a strong academic background with a Master's degree in Computer Application and relevant coursework in Machine Learning, Data Visualization, Business Intelligence, and Big Data. This aligns with the required advanced degree in Computer Science, Informatics, or a related field. However, the resume lacks specific work experience, which is a major weakness.

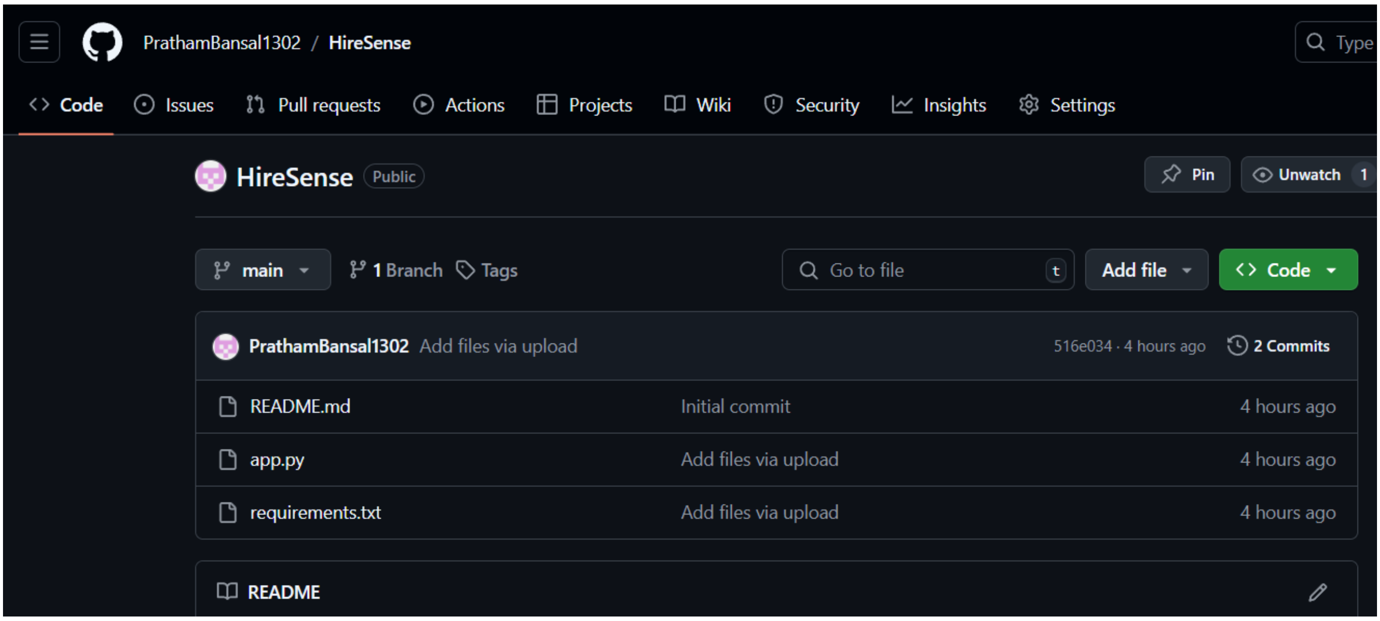
The job description emphasizes experience in the fields of natural language processing and machine learning, along with familiarity with language models like GPT, Mistral, etc., and proficiency in Python and experience with libraries like Langgraph, LangChain, Numpy, Pandas, ScikitLearn, etc. While the candidate mentions Machine Learning in their skills, the resume lacks details about specific projects or experience with NLP, GPT, or related technologies.

The candidate's experience with Python, Numpy, Pandas, and ScikitLearn, is a strength. However, there is no mention of Langgraph or LangChain, which are essential for the role. The candidate's familiarity with version control systems like GitLab is also a plus point, but experience with Snowflake would be beneficial.

**Overall, the candidate's academic background and technical skills are a good fit for the Data Scientist

11. Github Deployment:

Link: https://github.com/PrathamBansal1302/HireSense



PrathamBansal1302 Add files via upload 516e034 · 4 hours ago 2 Commits

README.md Initial commit 4 hours ago

app.py Add files via upload 4 hours ago

requirements.txt Add files via upload 4 hours ago

README

12. Future Scope:

The Application Tracking System (ATS), enhanced with Gemini Flash for transfer learning and an interactive Streamlit interface, holds significant potential for future development and refinement. As recruitment needs evolve and technology advances, there are several promising areas for enhancing the ATS's functionality, accuracy, and user experience.

1. Improved Candidate Matching with Enhanced NLP Models

- **Integration of Advanced NLP Models**: As larger and more sophisticated NLP models (e.g., GPT-4, BERT variants) are released, the ATS could leverage these models to improve the accuracy and depth of candidate matching.
- **Domain-Specific Model Training**: Creating separate models trained on industry-specific resumes and job descriptions (e.g., tech, finance, healthcare) could yield more tailored and relevant matches.
- **Continual Learning**: Implement continual learning so the model can update itself periodically with new industry trends, keywords, and role requirements, keeping up with evolving job markets.

2. Real-Time Collaboration and Multi-User Access

- **Multi-User Support**: Extend the system to allow multiple recruiters to collaborate on the same platform. This would be especially useful for larger organizations or teams working remotely.
- **Role-Based Access Control**: Implement different access levels, where recruiters, hiring managers, and administrators have distinct permissions, improving security and role-specific functionality.
- **Comments and Feedback**: Add features for recruiters to leave comments, feedback, or notes on candidates directly within the ATS, facilitating collaborative decision-making.

3. Integration with External Platforms and APIs

- **LinkedIn and Job Portals**: Integrate with LinkedIn, Indeed, and other job portals to import candidate profiles directly and automatically populate the ATS with new applications, improving data richness and ease of use.
- **Human Resources Management Systems (HRMS)**: Seamlessly integrate with HR systems (e.g., Workday, SAP SuccessFactors) to streamline the onboarding process for selected candidates.
- **Email and Calendar Integration**: Link with email platforms (e.g., Gmail, Outlook) and calendar tools for easy interview scheduling, follow-up reminders, and communication tracking, improving recruiter efficiency.

4. Enhanced User Interface and Visualization Tools

- **Customizable Dashboards**: Enable users to personalize their dashboard to see data most relevant to them, such as candidate sources, diversity metrics, or hiring trends.

- **Advanced Data Visualization**: Include additional visualization options like heatmaps, charts showing candidate experience distributions, and skill gap analyses, giving recruiters a deeper understanding of their talent pipeline.
- **Mobile-Friendly UI**: Optimize the ATS for mobile use, allowing recruiters to manage candidates and job postings on the go.

5. AI-Driven Interview Assistance and Evaluation

- **Automated Interview Scheduling**: Develop an AI-powered scheduling assistant that coordinates interview timings between candidates and recruiters, optimizing for convenience and minimizing back-and-forth emails.
- **Candidate Assessment**: Integrate tools for AI-driven video interview assessments, where the ATS can analyze video interviews for insights into candidate personality, communication skills, and cultural fit.
- **Smart Question Recommendations**: Use AI to recommend interview questions based on the candidate's profile and job description, allowing recruiters to focus on key areas during interviews.

6. Enhanced Candidate Scoring with Behavioral and Cultural Fit

- **Behavioral Analysis**: Integrate with behavioral assessment tools (e.g., Myers-Briggs, DISC) to evaluate candidates' personality traits and work preferences, helping recruiters assess cultural fit.
- **Employee Retention Predictions**: Use historical hiring data and machine learning to predict a candidate's likelihood of staying with the company, improving long-term hiring outcomes.
- **Diversity and Inclusion Metrics**: Develop scoring mechanisms that factor in diversity and inclusion goals, helping organizations build balanced and diverse teams.

7. AI-Powered Candidate Recommendations and Predictive Analytics

- **Candidate Recommendations**: Implement a recommendation system that suggests additional candidates who may be a good fit for similar roles, based on past hiring patterns and candidate profiles.
- **Predictive Hiring Analytics**: Use predictive analytics to forecast future hiring needs based on company growth, industry trends, and turnover rates, allowing organizations to plan their recruitment more strategically.
- **Talent Pool Management**: Track and analyze candidates who were a "good fit" but not selected, creating a talent pool that can be automatically re-evaluated for future positions.

8. Advanced Security and Compliance Features

- **GDPR and Data Privacy Compliance**: Ensure compliance with data privacy regulations like GDPR by implementing features such as data anonymization, candidate consent tracking, and secure data storage practices.

- **Data Encryption**: Encrypt all sensitive data, including candidate information and company job postings, to enhance security and protect against breaches.
- **Audit Logs and Transparency**: Introduce audit logs to track recruiter actions within the ATS, providing transparency and accountability, especially important in industries with high regulatory requirements.

9. Multilingual Support and Global Reach

- **Multilingual Processing**: Train the ATS to parse and evaluate resumes and job descriptions in multiple languages, making it adaptable to international recruiting needs.
- **Localization**: Allow recruiters to customize the platform's language and job market preferences based on their region, increasing usability across different markets.

10. Enhanced Reporting and Insights for Decision-Making

- **Recruitment Analytics Reports**: Provide detailed reports on recruiting metrics (e.g., time-to-hire, cost-per-hire, candidate conversion rates) to help companies optimize their recruitment strategy.
- **Hiring Success Metrics**: Track hiring success by analyzing the performance and retention of selected candidates, using this feedback to improve the ATS's candidate matching and scoring algorithms.
- **Diversity, Equity, and Inclusion (DEI) Insights**: Offer insights on diversity within the talent pipeline, helping companies work toward their DEI goals by identifying gaps and implementing targeted hiring strategies.

13. Conclusion

The Application Tracking System (ATS) utilizing **Gemini Flash** and **Streamlit** presents a modern, AI-enhanced approach to recruitment, designed to streamline the candidate evaluation and matching process. By leveraging transfer learning through Gemini Flash, the ATS effectively understands and processes complex resume and job description data, achieving high accuracy in matching candidates with job requirements.

The project successfully integrates core functionalities, including resume parsing, candidate ranking, and a responsive Streamlit interface, creating a seamless user experience for recruiters. This ATS not only improves the efficiency of candidate selection but also enables recruiters to make data-driven, objective hiring decisions, ultimately helping organizations attract and retain the most suitable talent.

Looking forward, this ATS holds substantial potential for future enhancements, from advanced NLP models and predictive analytics to integration with third-party HR and job portal systems. The future scope also includes the possibility of adding collaborative features, enhancing security and compliance, and supporting multilingual processing to expand its adaptability for global us.