# CareerCraft ATS-Friendly Resume Analyzer

## I.Project Overview

CareerCraft ATS-Friendly Resume Analyzer is a tool designed to help users analyze resumes and make them Applicant Tracking System (ATS) friendly. By utilizing the Gemini Pro pre-trained model, this project processes user-inputted resumes and provides suggestions and analysis based on ATS best practices. This web-based application provides an easy-to-use interface that allows users to upload resumes in PDF format, analyze them, and receive feedback in real-time.

## II.Project Structure

```
CareerCraft-ATS-Friendly-Resume-Analyzer/
  ├── images/              # Contains UI images used in the application
  ├── app.py               # Main application file
  ├── requirements.txt     # Lists the required libraries for the project
  └── .env                 # Contains environment variables like Google API Key
```

## III.Project Flow

1. User Interaction: Users interact with a simple web interface where they can upload their resumes in PDF format.
2. Data Transmission: The uploaded PDF is sent to the backend, where the content is processed using a Google API key.
3. Model Integration: The PDF content is sent via an API call to the Gemini Pro pre-trained model for analysis.
4. Result Generation: The Gemini Pro model analyzes the resume and returns the results, which are processed for ATS compliance.
5. Frontend Display: The analyzed results are returned to the frontend, where they are formatted and displayed in an easy-to-read format for the user.

We will be focusing on the following aspects:

- Frontend: A Streamlit-based UI that collects user input.
- Backend: Integrates Google API for authentication and calls the Gemini Pro pre-trained model for processing the resume.
- Pre-trained Model: Gemini Pro, which processes the resume content and provides an ATS-compliant analysis.
- Deployment: The application is hosted using a web framework.

# 1 Requirements Specification

## 1.1 Create a requirements.txt file to list the required libraries.

We need the following Python libraries to run the project:

- streamlit: For creating the web interface.
- streamlit_extras: Likely for additional Streamlit functionalities.
- google-generativeai: To interface with Google's generative AI services.
- python-dotenv: For loading environment variables like API keys from a `.env` file.
- PyPDF2: For reading and extracting text from PDF files.
- Pillow: For image processing.

Create a 'requirements.txt' file with the necessary dependencies:

requirements.txt
```
streamlit
streamlit_extras
google-generativeai
python-dotenv
PyPDF2
Pillow
```

## 1.2 Install the required libraries

Install these libraries by running:

```
pip install -r requirements.txt
```

# 2 Initialization of Google API Key

## 2.1 Generate Google API Key

To authenticate the application with Google's services, you will need to generate a Google API key.

Steps to Generate Google API Key:
1. Go to the [Google Cloud Console](https://console.cloud.google.com/).
2. Create a new project (or use an existing one).
3. Enable the Google APIs and create an API key.

## 2.2 Initialize Google API Key

Download the API key and save it in a `.env` file for security

.env
```
GOOGLE_API_KEY="AIzaSyASLanzRSJ2Bzil1XkASthCJVb8dl3l66w"
```

# 3 Interfacing with the Pre-trained Model

## 3.1 Load the Gemini Pro Pre-trained Model:

We import various libraries and configure Google's Generative AI with an API key stored in an environment variable.

1. Imports:
  - dotenv: Used to load environment variables from a `.env` file.
  - streamlit: Used for creating web applications.
  - streamlit_extras.add_vertical_space: Adds vertical space in Streamlit apps.
  - google.generativeai: Likely an SDK for interacting with Google's Generative AI.
  - os: Allows access to environment variables.
  - PyPDF2: A library for PDF manipulation.
  - PIL.Image: For image processing.

2. Configuration:
  - load_dotenv(): Loads environment variables from a .env file.
  - genai.configure(api_key=os.getenv("GOOGLE_API_KEY")): Configures Google's Generative AI with an API key.
  - model = genai.GenerativeModel('gemini-pro'): Initializes a generative model named "gemini-pro".

```python
from dotenv import load_dotenv
import streamlit as st
from streamlit_extras import add_vertical_space as avs
import google.generativeai as genai
import os
import PyPDF2
from PIL import Image

load_dotenv()

genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

model=genai.GenerativeModel('gemini-pro')
```

## 3.2 Implement A Function To Get Gemini Response

The function, `get_gemini_response`, takes an input, generates content using the "gemini-pro" model, and returns the response text.

1. **Function Definition**: `get_gemini_response(input)`: A function that accepts `input` as a parameter.
2. **Generate Content**: `response = model.generate_content(input)`: Calls the `generate_content` method on the `model` object (which is an instance of `genai.GenerativeModel`), passing in the `input`. This generates a response based on the input prompt.
3. **Return Response**: `return response.text`: Returns the `text` attribute of the `response`, which likely contains the generated output.

This function essentially serves as a wrapper around the generative model's `generate_content` method, making it easy to generate responses by simply calling `get_gemini_response(input)`.

```
15  def get_gemini_response(input):
16      response=model.generate_content(input)
17      return response.text
```

### 3.3 Implement A Function To Read PDF Content

We define a function, `input_pdf_text`, that extracts text from each page of an uploaded PDF file using the `PyPDF2` library.

1. **Function Definition**: `input_pdf_text(uploaded_file)` - takes `uploaded_file` as an argument.
2. **PDF Reader Setup**: `reader = PyPDF2.PdfReader(uploaded_file)` initializes a PDF reader for the uploaded file.
3. **Text Extraction Loop**:
   - Initializes an empty string `text`.
   - Loops through each page in the PDF using `range(len(reader.pages))`.
   - Extracts text from each page using `page.extract_text()` and appends it to the `text` variable.
4. **Return**: Finally, returns the complete text extracted from the PDF.

```
19  def input_pdf_text(uploaded_file):
20      reader=PyPDF2.PdfReader(uploaded_file)
21      text=''
22      for page_num in range(len(reader.pages)):
23          page=reader.pages[page_num]
24          text+=str(page.extract_text())
25      return text
```

### 3.4 Write A Prompt For Gemini Model

We create a prompt template designed to guide an Applicant Tracking System (ATS) or AI model in analyzing and comparing a resume against a job description (JD) for optimal job fit assessment.

1. **Purpose**:
   - The system is expected to act as an expert in ATS-related analysis across various technical roles.
   - It evaluates resumes meticulously against job descriptions to provide resume enhancement guidance.
2. **Parameters**:
   - **Resume**: Provided as `{text}` in the prompt.
   - **Job Description (JD)**: Provided as `{jd}` in the prompt.
3. **Expected Response Structure**:
   - **First Line**: Percentage match with the JD.
   - **Second Line**: List of missing keywords.
   - **Third Line**: Profile summary.
4. **Instructions**:
   - Each section should be labeled accordingly.
   - Space should be added between sections to improve readability.

This setup will help an AI model provide precise feedback for job applicants looking to tailor their resumes for specific job descriptions.

```
input_prompt="""
As an experienced ATS (Applicant Tracking System), proficient in the technical domain encompassing
Software Engineering, Data Science, Data Analysis, Big Data Engineering, Web Developer, Mobile App
Developer, DevOps Engineer, Machine Learning Engineer, Cybersecurity Analyst, Cloud Solutions Architect,
Database Administrator, Network Engineer, AI Engineer, Systems Analyst, Full Stack Developer, UI/UX
Designer, IT Project Manager, and additional specialized areas, your objective is to meticulously assess
resumes against provided job descriptions. In a fiercely competitive job market, your expertise is crucial
in offering top-notch guidance for resume enhancement. Assign precise matching percentages based on the JD
(Job Description) and meticulously identify any missing keywords with utmost accuracy.

resume: {text}

description: {jd}

I want the response in the following structure:
The first line indicates the percentage match with the job description (JD).
The second line presents a list of missing keywords.

The third section provides a profile summary.

Mention the title for all the three sections.
While generating the response put some space to separate all the three sections.
"""
```

# 4 Model Deployment

## 4.1 Integrate Web Framework

Create a simple Streamlit interface in the `app.py` file that allows users to upload their resumes and display the processed results.

### 4.1.1 Introduction

The code begins by configuring the Streamlit app page, setting the title to "Resume ATS Tracker" with a wide layout for an expansive user view. It introduces a two-column layout, where the first column is set to be three times wider than the second.

- **First Column (`col1`)**: This column houses the app's title "CareerCraft," a header, and a description introducing CareerCraft as an ATS-Optimized Resume Analyzer. This introductory section is styled with a markdown text to emphasize CareerCraft's purpose: optimizing resumes for job applications. The text is formatted in a justified alignment to enhance readability.
- **Second Column (`col2`)**: An image sourced from an external URL is displayed, adjusting automatically to fit the column width, complementing the introductory text by adding visual appeal.

```
##Streamlit UI
st.set_page_config(page_title="Resume ATS Tracker",layout="wide")

avs.add_vertical_space(4)

col1,col2=st.columns([3,2])

with col1:
    st.title("CareerCraft")
    st.header("Navigate the Job Market with Confidence!")
    st.markdown("""<p style="text-align: justify;">
    Introducing CareerCraft, an ATS-Optimized Resume Analyzer [] your ultimate solution for optimizing
    job applications and accelerating career growth. Our innovative platform leverages advanced ATS
    technology to provide job seekers with valuable insights into their resumes' compatibility with
    job descriptions. From resume optimization and skill enhancement to career progression guidance,
    CareerCraft empowers users to stand out in today's competitive job market. Streamline your job
    application process, enhance your skills, and navigate your career path with confidence. Join
    CareerCraft today and unlock new opportunities for professional success!
    </p>""", unsafe_allow_html=True)

with col2:
    st.image('https://cdn.dribbble.com/userupload/12500996/file/original-b458fe398a6d7f4e9999ce66ec856ff9.gif', use_column_width=True)

avs.add_vertical_space(10)
```

### 4.1.2 Offerings

Following the introduction, the code organizes a new two-column layout to showcase CareerCraft's offerings:

- **First Column (`col1`)**: Here, an image representing CareerCraft's services is displayed to add a visual context.
- **Second Column (`col2`)**: This column displays the header "Wide Range of Offerings," followed by a list of services such as ATS-Optimized Resume Analysis, Resume Optimization, and Skill Enhancement. These offerings provide a quick overview of the platform's key features, allowing users to understand CareerCraft's benefits at a glance.

```
76  col1, col2 = st.columns([3, 2])
77
78  with col2:
79      st.header("Wide Range of Offerings")
80      st.write('ATS-Optimized Resume Analysis')
81      st.write('Resume Optimization')
82      st.write('Skill Enhancement')
83      st.write('Career Progression Guidance')
84      st.write('Tailored Profile Summaries')
85      st.write('Streamlined Application Process')
86      st.write('Personalized Recommendations')
87      st.write('Efficient Career Navigation')
88
89  with col1:
90      img1 = Image.open("images/icon1.png")
91      st.image(img1, use_column_width=True)
92
93  avs.add_vertical_space(10)
```

### 4.1.3 Resume ATS Analysis Section

This section provides an interactive layout that allows users to submit job descriptions and resumes:

- **Left Column (`col1`)**: This column includes a header that encourages users to start their "career adventure." Here, users can paste a job description into a text area and upload their resume in PDF format via a file uploader. Upon clicking the "Submit" button, the uploaded PDF resume is processed. The `get_gemini_response` function then generates a response based on the provided job description, displaying it as a subheader.
- **Right Column (`col2`)**: An image representing a career adventure is displayed, further engaging users visually.

```python
col1, col2 = st.columns([3, 2])
with col1:
    st.markdown("<h1 style='text-align: center;'>Embark on Your Career Adventure</h1>", unsafe_allow_html=True)
    jd=st.text_area("Paste the Job Description")
    uploaded_file=st.file_uploader("Upload Your Resume",type="pdf",help="Please upload the pdf")

    submit = st.button("Submit")

    if submit:
        if uploaded_file is not None:
            text=input_pdf_text(uploaded_file)
            response=get_gemini_response(input_prompt)
            st.subheader(response)

with col2:
    img2 = Image.open("images/icon2.png")
    st.image(img2, use_column_width=True)

avs.add_vertical_space(10)
```

### 4.1.4 FAQ

The FAQ section is designed to address common questions, adding to the user experience with organized information and visuals:

- **Right Column (`col2`)**: This column contains a header labeled "FAQ," followed by frequently asked questions and answers about CareerCraft. Questions are presented with `st.write()` statements, while vertical space between each question-answer pair is added using `avs.add_vertical_space()` to ensure clarity and readability.
- **Left Column (`col1`)**: This column includes an image that aligns with the FAQ content, further enhancing the visual organization.

```python
col1, col2 = st.columns([2, 3])
with col2:
    st.markdown("<h1 style='text-align: center;'>FAQ</h1>", unsafe_allow_html=True)
    st.write("Question: How does CareerCraft analyze resumes and job descriptions?")
    st.write("""Answer: CareerCraft uses advanced algorithms to analyze resumes and job descriptions,
                identifying key keywords and assessing compatibility between the two.""")
    avs.add_vertical_space(3)
    st.write("Question: Can CareerCraft suggest improvements for my resume?")
    st.write("""Answer: Yes, CareerCraft provides personalized recommendations to optimize your resume
                for specific job openings, including suggestions for missing keywords and alignment with
                desired job roles.""")
    avs.add_vertical_space(3)
    st.write("Question: Is CareerCraft suitable for both entry-level and experienced professionals?")
    st.write("""Answer: Absolutely! CareerCraft caters to job seekers at all career stages, offering
                tailored insights and guidance to enhance their resumes and advance their careers.""")

with col1:
    img3 = Image.open("images/icon3.png")
    st.image(img3, use_column_width=True)
```

### 4.2 Hosting the Application

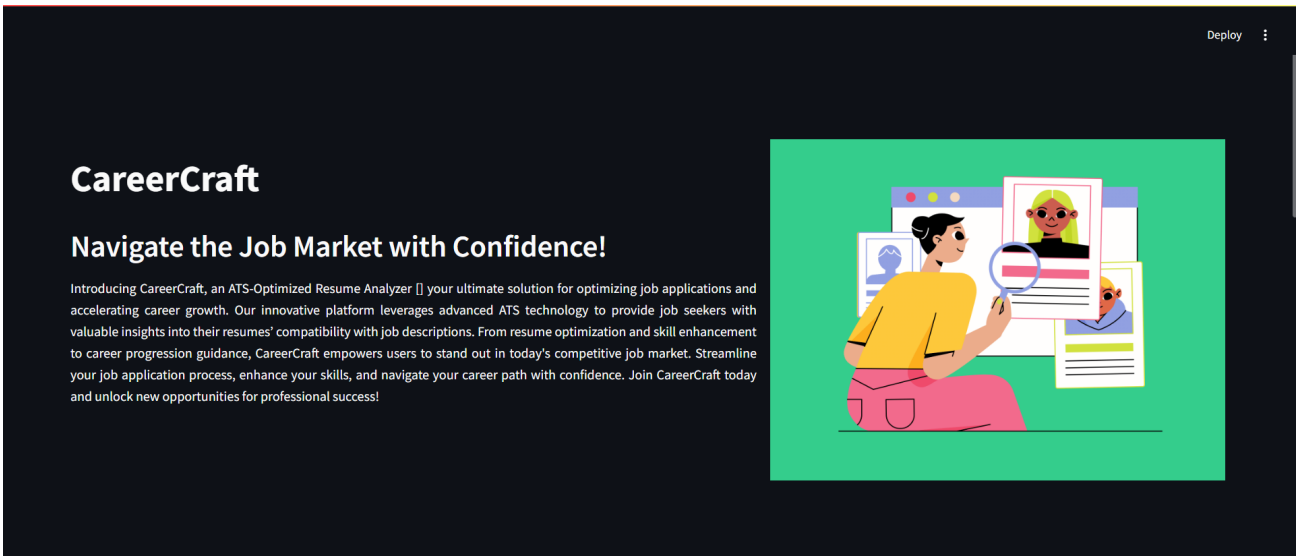
To run the app, open a new terminal and run the following command :

`streamlit run app.py`

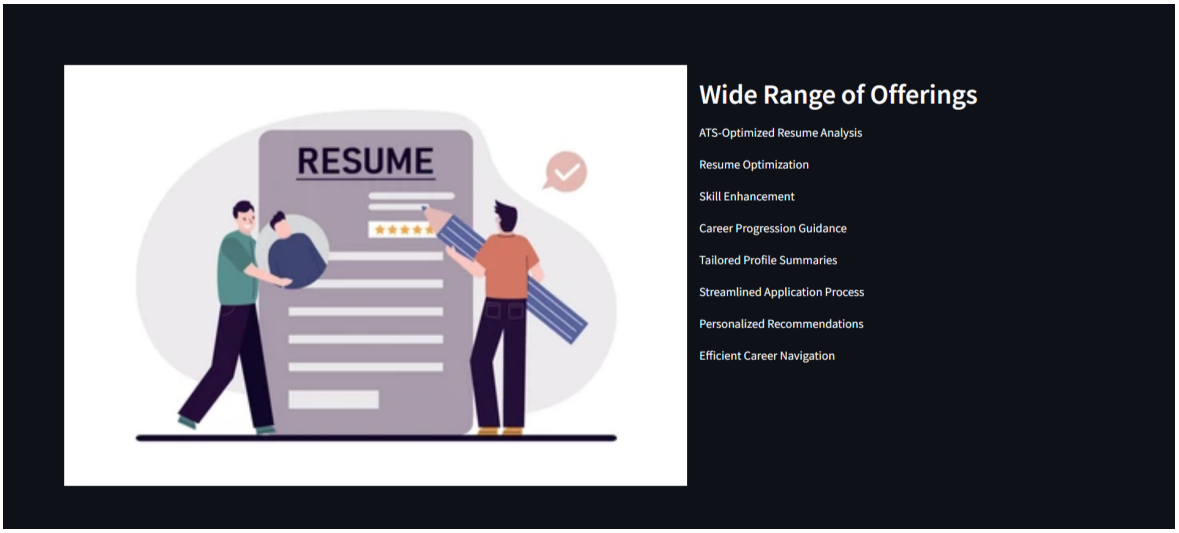
```
PS C:\Users\navul\OneDrive\Desktop\GEN-AI\smartinternz\career-craft\CareerCraft> streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.43:8501
```

## 5 Results/Outputs

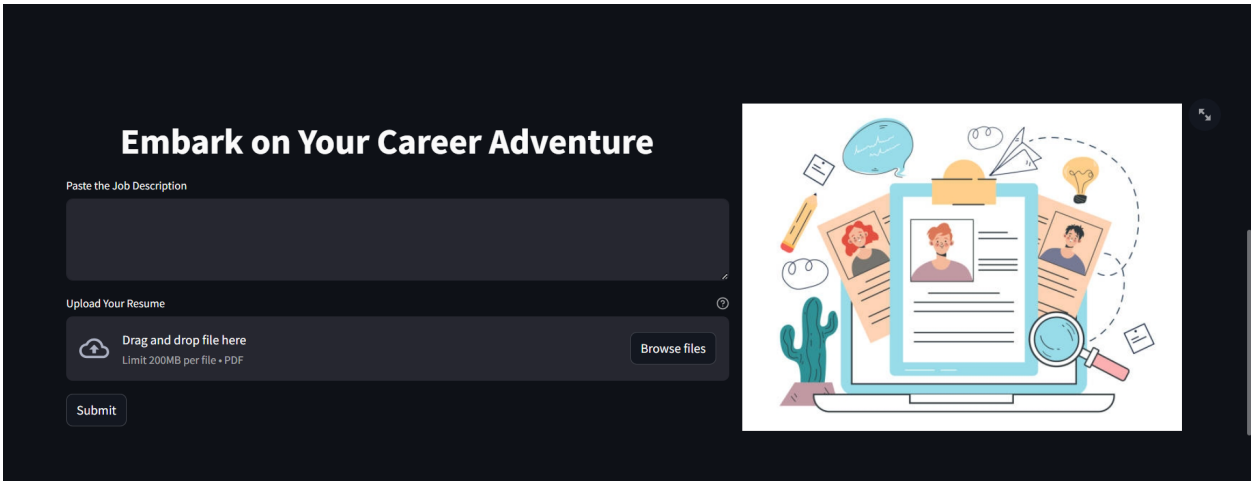
### 5.1 Introduction Section




### 5.2 Offerings Section



### 5.3 Resume ATS Analysis Section




## Sample JD1:

**Responsibilities Include But Are Not Limited To**

- Develop, and maintain robust and scalable software using modern languages.
- Use AWS services for deploying and maintaining applications.
- Collaborate with team leads and managers to complete tasks and projects.
- Participate in code reviews, team meetings, and sprint planning.
- Review, update, and maintain documentation such as design diagrams, architectural diagrams, and test procedures.
- Provide insights and contribute to process improvements.
- Work onsite at a company office.

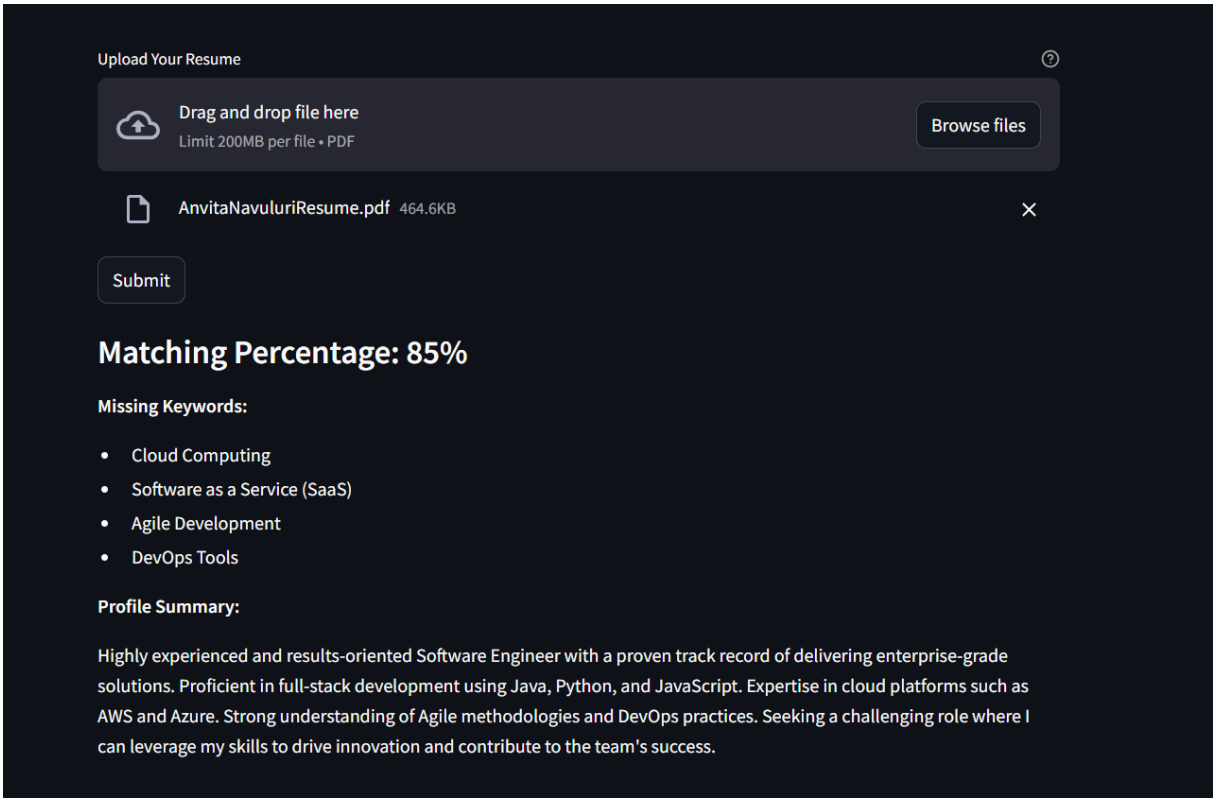
Minimum Qualifications

- Bachelor’s degree in Computer Science, Engineering, or a related field.
- Knowledge of React frontend framework.
- Experience in software development, including internships and practical experience in various projects.
- Skilled in Python and TypeScript.
- Knowledge of cloud computing with AWS.
- Experience building REST and/or GraphQL APIs.
- Ability to collaborate effectively in a team environment.
- Excellent problem-solving skills and attention to detail.
- Strong communication abilities.
- Ability to earn trust, maintain positive and professional relationships, and contribute to a culture of inclusion.
- Must be a U.S. citizen or national, U.S. permanent resident (current Green Card holder), or lawfully admitted into the U.S. as a refugee or granted asylum.

Preferred Qualifications

- Understanding of standard cloud architecture patterns, including microservice architectures.
- Familiarity with Agile methodologies.
- Knowledge of Git, CI/CD, and other DevOps practices.

**Output:**




**Sample JD 2:**

Objectives of this role

Maintain essential IT infrastructure, including operating systems, security tools, applications, servers, email systems, laptops, desktops, software, and hardware

Take responsibility for projects and solutions within the larger business initiative

Handle business-critical IT tasks and systems administration

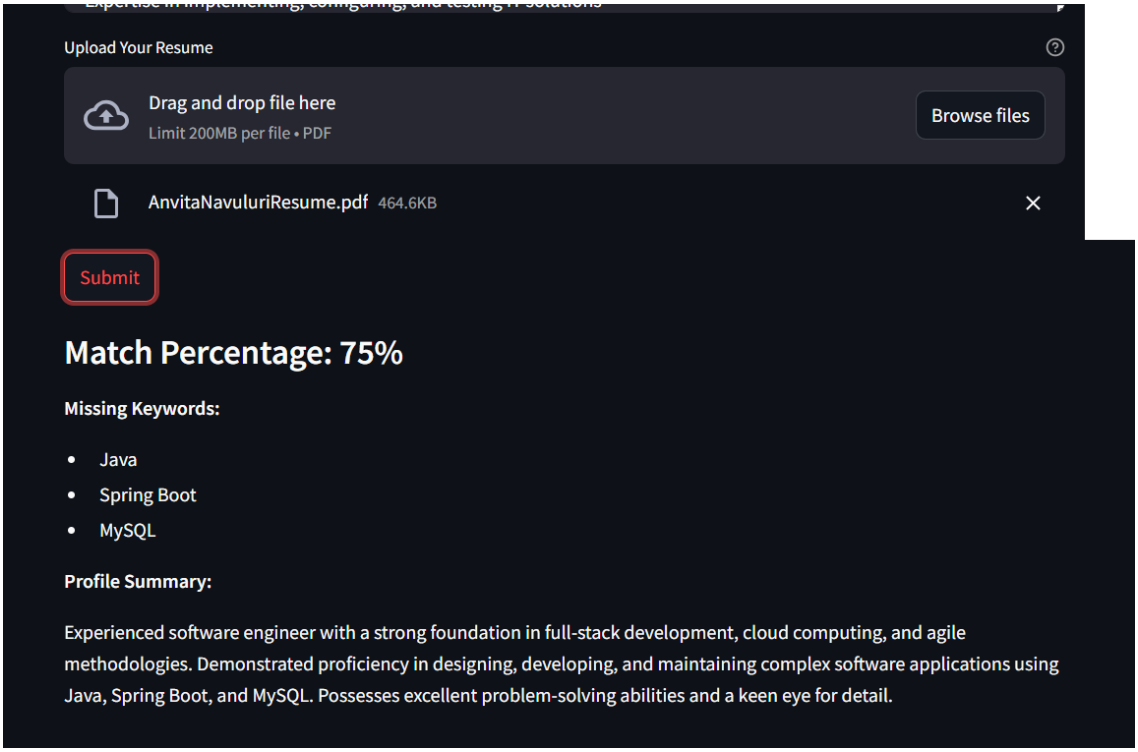
Research and evaluate emerging technologies, hardware, and software

Track and maintain hardware and software inventory

Responsibilities

Analyze departmental needs, identify vulnerabilities, and boost productivity, efficiency,

and accuracy to inform business decisions
Ensure network components work together seamlessly to meet business needs, using their full range of capabilities, and stay informed about new features and competitor solutions
Analyze processes, technologies, and vendors continually to find areas for improvement
Prepare cost–benefit analyses when upgrades are necessary, and monitor vendors to ensure that they're offering the best-possible service and value for business needs
Train employees to use software and hardware; troubleshoot issues and provide technical support when needed
Develop and execute disaster planning and maintain data backups
Required skills and qualifications
Five or more years of experience in IT management
Deep knowledge of programming languages and operating systems (ex: Microsoft Exchange, Active Directory), current equipment and technologies, enterprise backup and recovery procedures, and systems performance monitoring
Expertise in implementing, configuring, and testing IT solutions
Strong creative and analytical thinking
Preferred skills and qualifications
Bachelor's degree (or equivalent) in information technology or computer science
Professional certification
Output:



**Sample JD 3:**

Objectives of this role

Work in tandem with our engineering team to identify and implement the most optimal cloud-based solutions for the company

Define and document best practices and strategies regarding application deployment and infrastructure maintenance

Provide guidance, thought leadership, and mentorship to developer teams to build their cloud competencies

Ensure application performance, uptime, and scale, maintaining high standards for code quality and thoughtful design

Manage cloud environments in accordance with company security guidelines

Responsibilities

At the direction of lead architects, develop and implement technical efforts to design, build, and deploy AWS applications, including large-scale data processing, computationally intensive statistical modeling, and advanced analytics

Participate in all aspects of the software development lifecycle for AWS solutions, including planning, requirements, development, testing, and quality assurance

Troubleshoot incidents, identify root causes, fix and document problems, and implement preventive measures

Educate teams on the implementation of new cloud-based initiatives, providing associated training when necessary

Demonstrate exceptional problem-solving skills, with an ability to see and solve issues before they affect business productivity

Required skills and qualifications

Three or more years of experience in architecting, designing, developing, and implementing cloud solutions on AWS platforms

Understanding of and experience with the five pillars of a well-architected framework

Experience in several of the following areas: database architecture, ETL, business intelligence, big data, machine learning, advanced analytics

Proven ability to collaborate with multidisciplinary teams of business analysts**,** developers, data scientists, and subject-matter experts

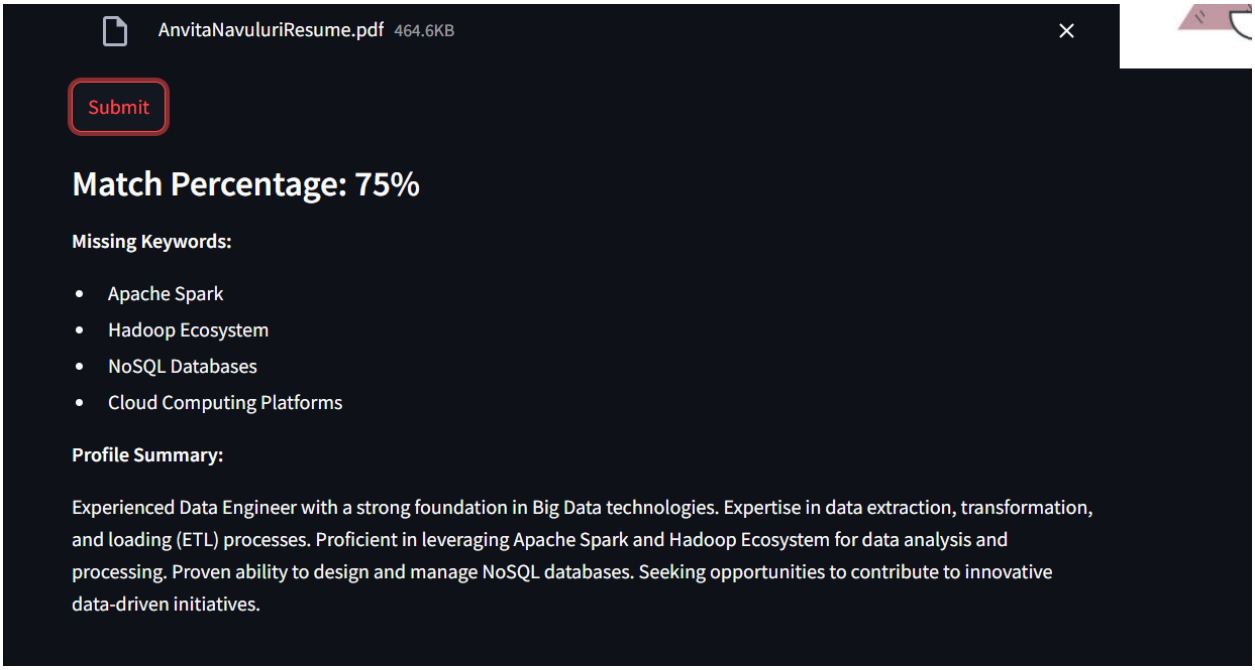
Preferred skills and qualifications

Bachelor’s degree (or equivalent) in computer science, information technology, or mathematics

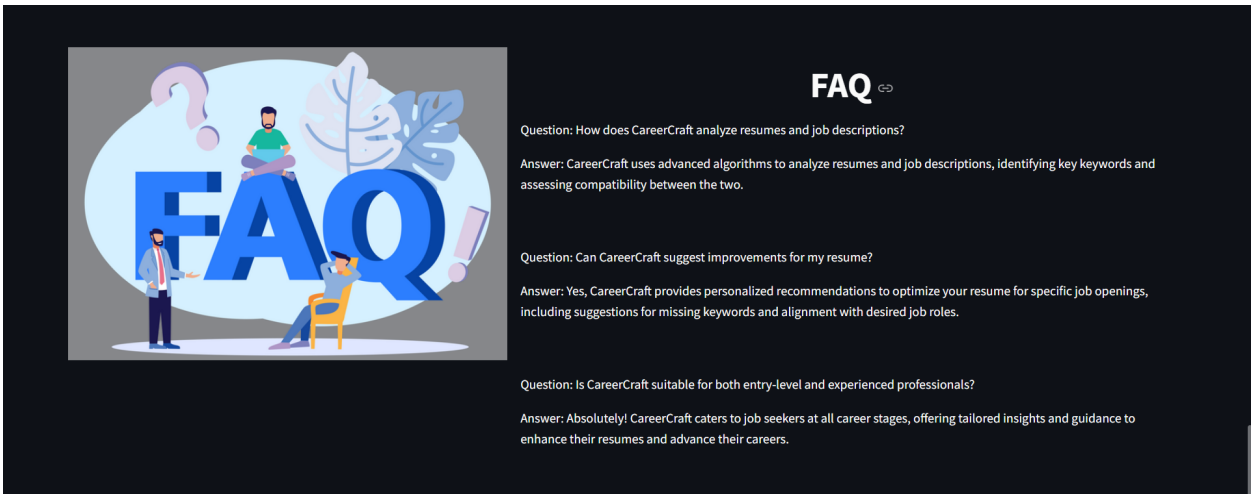
Knowledge of web services, API, REST, and RPC

AWS certification

Output:




### 5.4 FAQS




## 6 Conclusion

The CareerCraft ATS-Friendly Resume Analyzer is a powerful tool for analyzing resumes to ensure they are compliant with ATS standards. By integrating with the Gemini Pro model and deploying a user-friendly UI, this project simplifies the process of optimizing resumes for job applications.