CareerCraft: ATS-Optimized Resume Analyzer using Gemini Model

Project Description

CareerCraft, an ATS-Optimized Resume Analyzer is a cutting-edge project designed to revolutionize the job application process using advanced ATS (Applicant Tracking System) technology. This innovative system empowers job seekers by providing insights into their resumes' compatibility with job descriptions, highlighting missing keywords, and offering tailored profile summaries for optimal presentation to potential employers.

Scenario 1: Resume Optimization
CareerCraft assists job seekers in optimizing their resumes for specific job openings. By analyzing job descriptions and resumes, CareerCraft identifies the percentage match between the two, suggests missing keywords, and offers recommendations to improve resume alignment with the desired job roles. This feature streamlines the application process and increases the chances of securing interviews.

Scenario 2: Skill Enhancement
For individuals looking to enhance their skills and qualifications, CareerCraft serves as a valuable tool for identifying areas of improvement. By comparing resumes to industry-standard job descriptions, CareerCraft identifies skill gaps and provides personalized suggestions for skill development and enhancement. This feature empowers users to tailor their professional profiles to meet the demands of their desired career paths.

Scenario 3: Career Progression Guidance
Professionals seeking career advancement opportunities can rely on CareerCraft for strategic guidance. By analyzing resumes and job descriptions, CareerCraft offers insights into potential career trajectories, identifies relevant skills and experiences, and provides personalized recommendations for achieving career goals. This feature helps users navigate their career paths effectively and capitalize on growth opportunities.

Technical Architecture
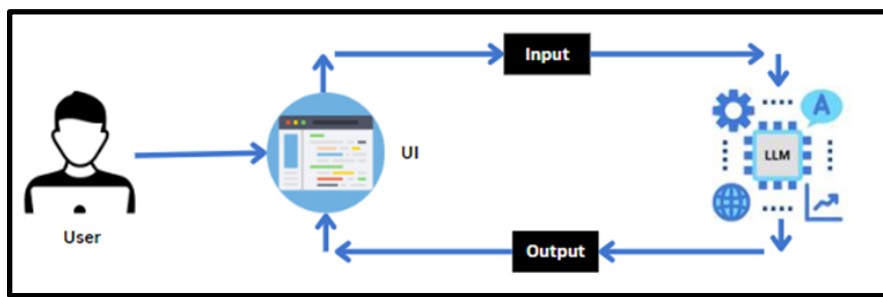
Project Flow:

- User interacts with the UI to enter the input.
- User input is collected from the UI and transmitted to the backend using the Google API key.
- The input is then forwarded to the Gemini Pro pre-trained model via an API call.
- The Gemini Pro pre-trained model processes the input and generates the output.
- The results are returned to the frontend for formatting and display.

To accomplish this, we have to complete all the activities listed below:

- **Requirements Specification**
  - Create a requirements.txt file to list the required libraries.
  - Install the required libraries
- **Initialization of Google API Key**
  - Generate Google API Key
  - Initialize Google API Key
- **Interfacing with Pre-trained Model**
  - Load the Gemini Pro pre-trained model
  - Implement a function to get gemini response
  - Implement a function to read PDF content
  - Write a prompt for gemini model
- **Model Deployment**
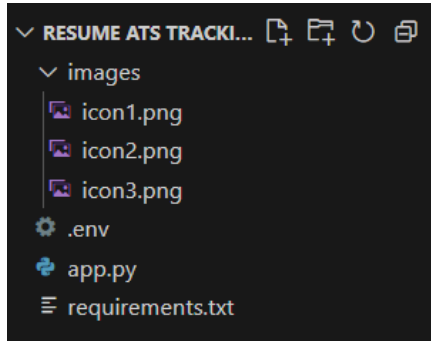 - Integrate with Web Framework
  - Host the Application

Prior Knowledge:

You must have the prior knowledge of the following topics to complete this project.

- Generative AI Concepts
- NLP: https://www.tutorialspoint.com/natural_language_processing/index.htm
- Generative AI: https://en.wikipedia.org/wiki/Generative_artificial_intelligence
- About Gemini: https://deepmind.google/technologies/gemini/#introduction
- Gemini API: https://ai.google.dev/gemini-api/docs/get-started/python
- Gemini Demo: https://colab.research.google.com/github/google/generative-ai-docs/blob/main/site/en/gemini-api/docs/get-started/python.ipynb
- Streamlit: https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/

Project Structure

Create the Project folder which contains files as shown below:

- **images folder**: It is established to store the images utilized in the user interface.
- **.env file**: It securely stores the Google API key.
- **app.py**: It serves as the primary application file housing both the model and Streamlit UI code.
- **requirements.txt**: It enumerates the libraries necessary for installation to ensure proper functioning.
- Additionally, ensure proper file organization and adhere to best practices for version control.

Milestone 1: Requirements Specification

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.
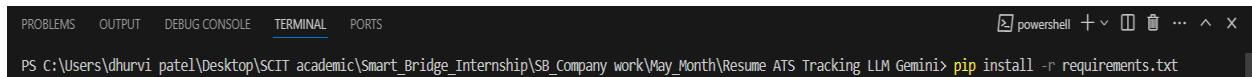
Activity 1: Create a requirements.txt file to list the required libraries.

```
# libraries need to be installed
streamlit
streamlit_extras
google-generativeai
python-dotenv
PyPDF2
Pillow
```

- **streamlit**: Streamlit is a powerful framework for building interactive web applications with Python.
- **streamlit_extras**: Additional utilities and enhancements for Streamlit applications.

- **google-generativeai**: Python client library for accessing the GenerativeAI API, facilitating interactions with pre-trained language models like Gemini Pro.
- **python-dotenv**: Python-dotenv allows you to manage environment variables stored in a .env file for your Python projects.
- **PyPDF2**: It is a Python library for extracting text and manipulating PDF documents.
- **Pillow**: Pillow is a Python Imaging Library (PIL) fork that adds support for opening, manipulating, and saving many different image file formats.

Activity 2: Install the required libraries

PS C:\Users\dhurvi patel\Desktop\SCIT academic\Smart_Bridge_Internship\SB_Company work\May_Month\Resume ATS Tracking LLM Gemini> pip install -r requirements.txt

- Open the terminal.
- Run the command: pip install -r requirements.txt
- This command installs all the libraries listed in the requirements.txt file.

Milestone 2: Initialization of Google API Key

The Google API key is a secure access token provided by Google, enabling developers to authenticate and interact with various Google APIs. It acts as a form of identification, allowing users to access specific Google services and resources. This key plays a crucial role in authorizing and securing API requests, ensuring that only authorized users can access and utilize Google's services.
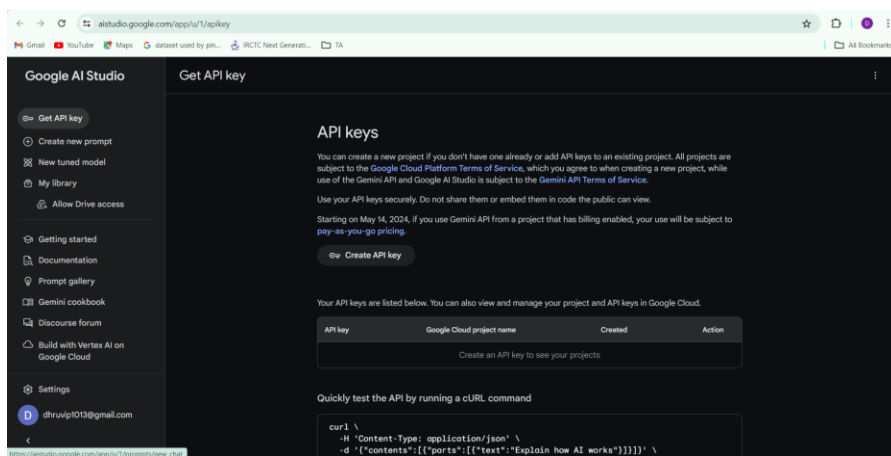
Activity 1: Generate Google API Key

Click the provided link to access the following webpage.
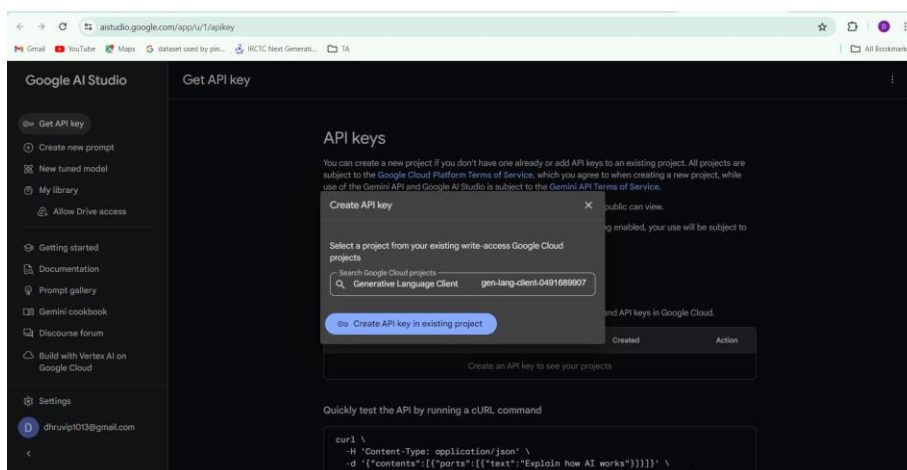
Link: https://ai.google.dev/gemini-api/docs/api-key
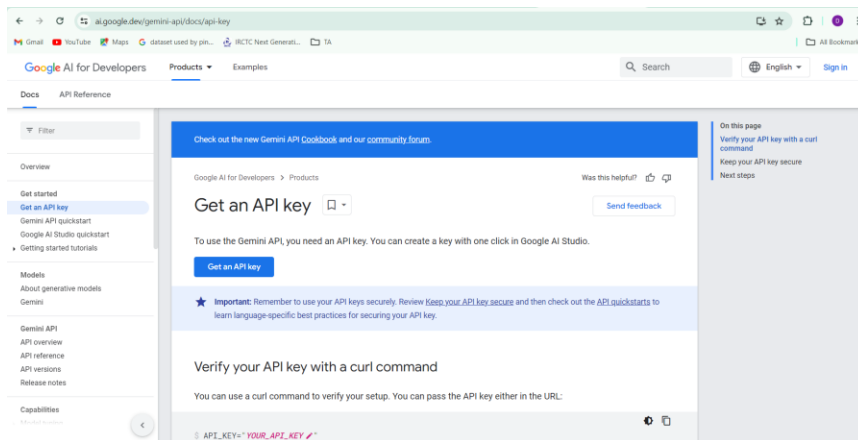
After signing in to your account, navigate to the 'Get an API Key' option. Clicking on this option will redirect you to another webpage as shown below.

Next, click on 'Create API Key' and choose the generative language client as the project. Then, select 'Create API key in existing project'.

Copy the newly generated API key as it is required for loading the Gemini Pro pre-trained model.

Activity 2: Initialize Google API Key

```
GOOGLE_API_KEY = "<Enter the copied Google API Key>"
```

- Create a .env file and define a variable named GOOGLE_API_KEY.
- Assign the copied Google API key to this variable.
- Paste the API key obtained from the previous steps here.

Milestone 3: Interfacing with Pre-trained Model

To interface with the pre-trained model, we'll start by creating an app.py file, which will contain both the model and Streamlit UI code.

Activity 1: Load the Gemini Pro pre-trained model

```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')
```

- The code begins by importing necessary libraries and modules, including dotenv, Streamlit, os, PyPDF2, GenerativeAI from Google, PIL (Python Imaging Library), and a custom module for adding vertical space in Streamlit.
- It loads environment variables from the .env file using the load_dotenv() function.
- The GenerativeAI module is configured with the Google API key stored in the environment variable GOOGLE_API_KEY.
- A GenerativeModel object named "model" is created using the Gemini Pro pre-trained model from Google.
- The code is essentially setting up the environment, configuring the GenerativeAI module with the API key, and loading the Gemini Pro model for generating responses to user inputs in the Streamlit app.

Activity 2: Implement a function to get gemini response

```python
def get_gemini_response(input):
    response = model.generate_content(input)
    return response.text
```

- The function get_gemini_response takes an input text as a parameter.
- It calls the generate_content method of the model object to generate a response.
- The generated response is returned as text.

Activity 3: Implement a function to read PDF content

```python
def input_pdf_text(uploaded_file):
    reader = PyPDF2.PdfReader(uploaded_file)
    text = ''
    for page_num in range(len(reader.pages)):
        page = reader.pages[page_num]
        text += str(page.extract_text())
    return text
```

- The function input_pdf_text takes an uploaded PDF file as input.
- It creates a PdfReader object from the PyPDF2 library to read the uploaded PDF file.
- It initializes an empty string variable text to store the extracted text from the PDF.
- It iterates over each page of the PDF using a loop.
- For each page, it extracts the text using the extract_text() method and appends it to the text variable.
- Finally, it returns the concatenated text extracted from all pages of the PDF.

Activity 4: Write a prompt for gemini model

```
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.
"""
```

- The input_prompt is a multiline string containing instructions for an ATS (Applicant Tracking System).

- It describes the expertise required for the ATS, including proficiency in various technical domains such as Software Engineering, Data Science, etc.
- The objective of the ATS is to assess resumes against provided job descriptions in a competitive job market.
- It requests the response to be structured into three sections: percentage match with the job description, a list of missing keywords, and a profile summary.

Milestone 4: Model Deployment

We deploy our model using the Streamlit framework, a powerful tool for building and sharing data applications quickly and easily. With Streamlit, we can create interactive web applications that allow users to interact with our models in real-time, providing an intuitive and seamless experience.

Activity 1: Integrate with Web Framework

The webpage is organized into four main sections to provide users with a comprehensive experience:

- Introduction:

```python
## 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)
```

- The provided code sets the page configuration for a Streamlit application titled "Resume ATS Tracker" with a wide layout.
- It then creates a layout with two columns, where the first column is three times wider than the second.

- In the first column, it displays the title "CareerCraft" along with a header and a markdown text introducing CareerCraft as an ATS-Optimized Resume Analyzer.
- In the second column, it displays an image sourced from a URL, adjusting its width to fit the column.
- Overall, the code creates a visually appealing interface for the CareerCraft application, combining text and images to convey its features and benefits to users.

- Offering:

```python
col1, col2 = st.columns([3, 2])
with col2:
    st.header("Wide Range of Offerings")
    st.write('ATS-Optimized Resume Analysis')
    st.write('Resume Optimization')
    st.write('Skill Enhancement')
    st.write('Career Progression Guidance')
    st.write('Tailored Profile Summaries')
    st.write('Streamlined Application Process')
    st.write('Personalized Recommendations')
    st.write('Efficient Career Navigation')

with col1:
    img1 = Image.open("images/icon1.png")
    st.image(img1, use_column_width=True)


avs.add_vertical_space(10)
```

- The provided code creates a two-column layout using Streamlit, with the first column containing an image and the second column containing a header followed by a list of offerings.
- The header "Wide Range of Offerings" introduces the list, which includes various features such as ATS-Optimized Resume Analysis, Resume Optimization, Skill Enhancement, and others.
- Overall, this layout effectively presents the range of services provided by the application in a clear and organized manner.

- **Resume ATS Tracking Application:**

```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 uplaod 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)
```

  - This code snippet creates a two-column layout using Streamlit.
  - In the left column (`col1`), it displays a header inviting users to embark on their career adventure, followed by a text area for pasting the job description and a file uploader for uploading the resume.
  - Upon clicking the "Submit" button, it retrieves the text from the uploaded PDF resume, generates a response using the `get_gemini_response` function based on the provided job description, and displays the response as a subheader.
  - In the right column (`col2`), it displays an image representing the career adventure concept.
  - Overall, this layout allows users to interactively submit their job descriptions and resumes while visually engaging with the application's theme.

- **FAQ:**

```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)
```

○ This code snippet divides the page into two columns using Streamlit's `st.columns()` function.
○ In the right column (`col2`), it displays a FAQ section with questions and answers about CareerCraft.
○ Each question and answer pair is presented using `st.write()`.
○ Additionally, vertical space is added between each question and answer pair using `avs.add_vertical_space()`.
○ In the left column (`col1`), an image is displayed using `st.image()`.
○ This layout effectively organizes the FAQ content alongside visual elements, enhancing the user experience.

Activity 2: Host the Application

Launching the Application:

● To host the application, go to the terminal, type - streamlit run app.py
● Here app.py refers to a python script.

```
PS C:\Users\dhurvi patel\Desktop\SCIT academic\Smart_Bridge_Internship\SB_Company work\May_Month\Resume ATS Tracking LLM Gemini> streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.29.80:8501
```

Run the command to get the below results

Deploy

Wide Range of Offerings

ATS-Optimized Resume Analysis

Resume Optimization

Skill Enhancement

Career Progression Guidance

Tailored Profile Summaries

Streamlined Application Process

Personalized Recommendations

Efficient Career Navigation
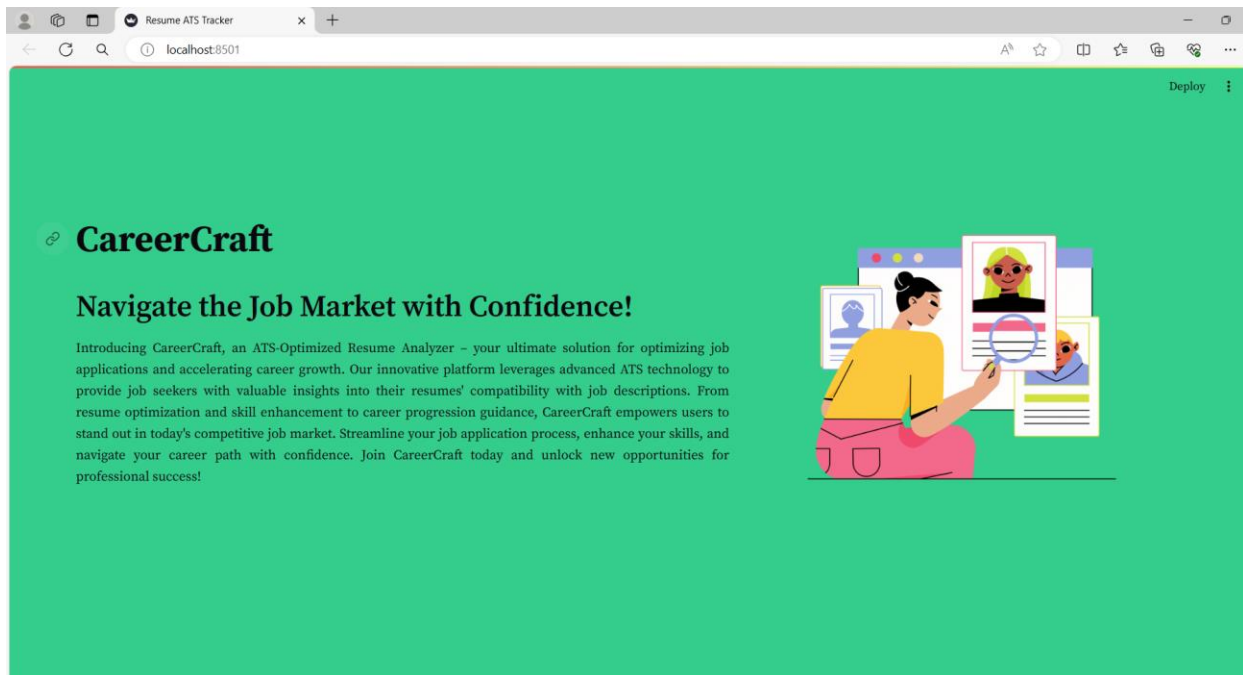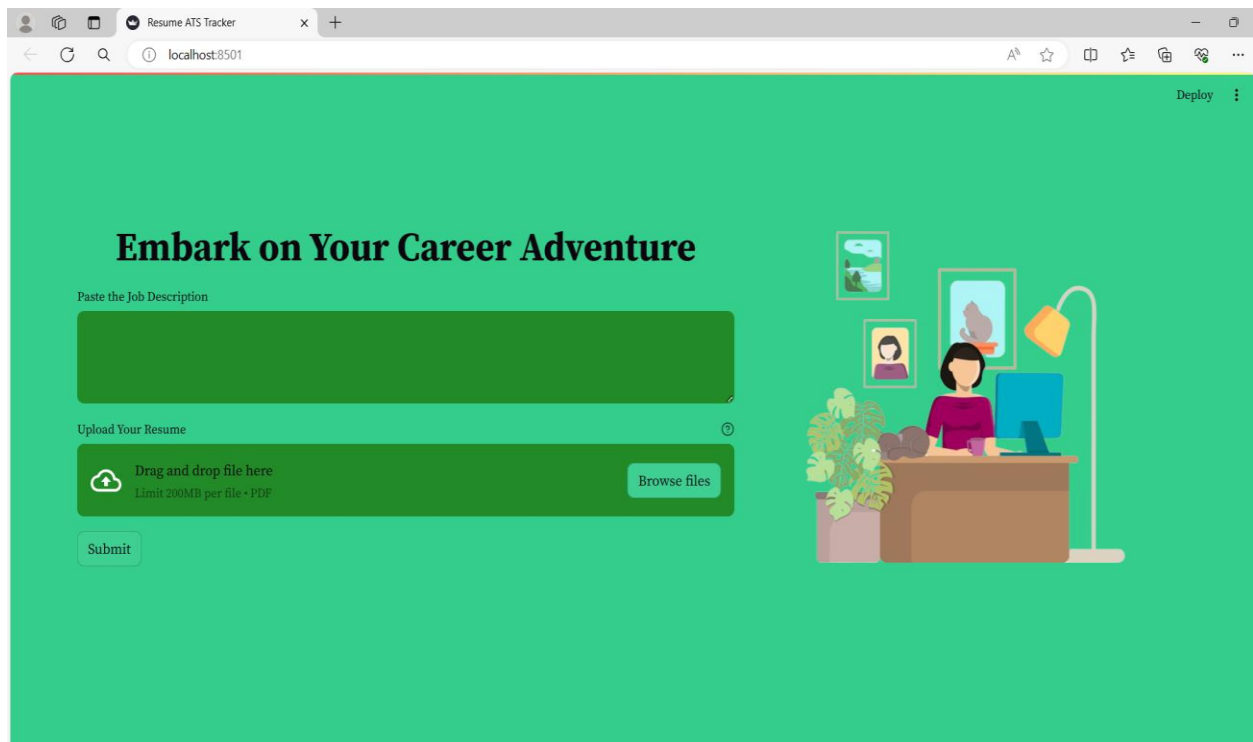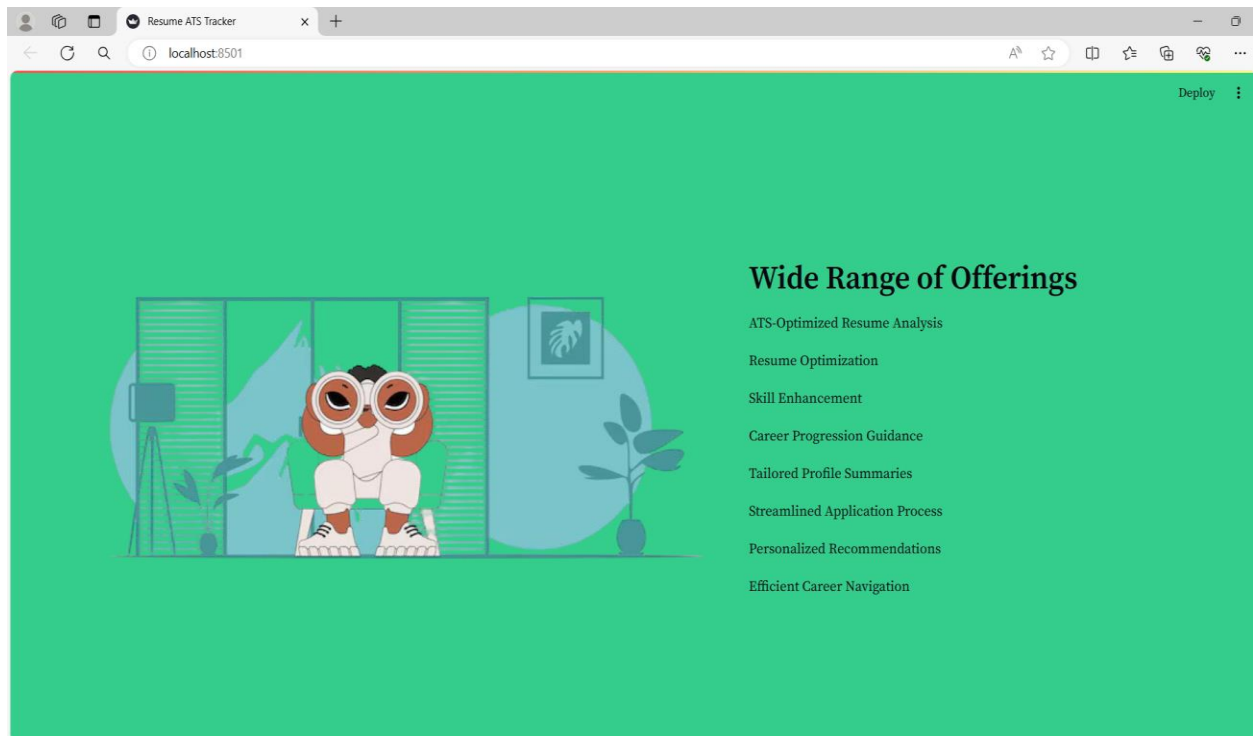
Deploy

Embark on Your Career Adventure

Paste the Job Description

Upload Your Resume

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

Browse files

Submit

INPUT 1

OUTPUT 1

INPUT 2

OUTPUT 2

INPUT 3

OUTPUT 3

FAQ

Question: How does CareerCraft analyze resumes and job descriptions?

Answer: CareerCraft uses advanced algorithms to analyze resumes and job descriptions, identifying key keywords and assessing compatibility between the two.

Question: Can CareerCraft suggest improvements for my resume?

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.

Question: Is CareerCraft suitable for both entry-level and experienced professionals?

Answer: Absolutely! CareerCraft caters to job seekers at all career stages, offering tailored insights and guidance to enhance their resumes and advance their careers.