

## – Smart Resume Analyzer

### Project Overview

HirEase AI is an AI-powered resume analysis tool designed to help job seekers optimize their resumes for specific job descriptions. It uses **Google's Gemini AI model** to analyze resumes and provide insights on how well a candidate's profile aligns with the job requirements. The system also provides feedback on strengths, weaknesses, and areas for improvement.

---

### Project Features

#### **Resume Parsing & Analysis**

- Uploads and extracts content from a **PDF resume**
- Converts **first page of PDF into an image** for processing
- Uses **Google Gemini AI** to analyze resume content

#### **Job Description Matching**

- Allows users to paste a **job description**
- Compares resume content with job requirements
- Highlights **key skills, experience, and qualification gaps**

#### **AI-Powered ATS Scanner**

- Acts as a **simulated Applicant Tracking System (ATS)**
- Calculates a **resume-job match percentage**
- Identifies **missing keywords** affecting ranking

#### **User-Friendly Interface**

- Built with **Streamlit** for a **clean and responsive UI**
- Features **file upload, text input, and AI-generated feedback**
- Uses **CSS-based styling** for enhanced UI elements

### Technologies Used

#### Frontend & UI

- **Streamlit**: For creating a responsive and interactive web UI.
- **HTML & CSS**: For styling and enhancing UI elements.

#### Backend & AI Processing

- **Google Gemini API**: Used for Natural Language Processing (NLP) to analyze resume and job description.
- **Python**: The core programming language for implementation.

- **pdf2image & PIL:** To extract and convert PDF resume into an image for AI processing.
- **base64 Encoding:** To encode images for API requests.
- **dotenv:** For managing API keys securely.

## How It Works

### Step 1: Upload a Resume (PDF Format)

- The user uploads their resume via the **Streamlit sidebar**
- The system extracts **the first page of the PDF**
- Converts it to an **image** for AI processing

### Step 2: Enter the Job Description

- The user pastes the **job description** into a text box

### Step 3: Select an Analysis Option

- **"Get Resume Details"** → AI analyzes the resume and provides feedback on skills, strengths, and weaknesses
- **"Get Resume Match %"** → AI calculates a compatibility percentage based on the job description

### Step 4: AI Processing & Output

- The **Google Gemini AI model** processes the input
- It returns a **detailed evaluation** of the resume's match to the job description

## Use Cases

- Job Seekers:** Optimize resumes for better job opportunities
- HR Recruiters:** Quickly analyze and filter resumes
- Career Coaches:** Provide AI-powered feedback to clients
- Students & Freshers:** Improve resume alignment with industry expectations

## Code Breakdown & Explanation

### 1. Importing Required Libraries

```
import streamlit as st
from dotenv import load_dotenv
import base64
import os
import io
from PIL import Image
import pdf2image
import google.generativeai as genai
```

- **streamlit** → For building the web UI.
- **dotenv** → Loads API keys securely.
- **base64** → Encodes images for AI processing.
- **PIL (Pillow)** → Handles image processing.
- **pdf2image** → Converts PDF resumes to images.

- `google.generativeai` → Connects with Google Gemini AI.

## 2. Loading API Key & Configuring AI

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

- Loads the **Google Gemini API Key** from a `.env` file.

## 3. AI Processing Function (Gemini API)

```
def get_gemini_response(input_text, pdf_content, prompt):
    model = genai.GenerativeModel('gemini-1.5-flash')
    response = model.generate_content([input_text, pdf_content, prompt])
    return response.text if response else "Error in generating response"
```

- Calls Gemini AI to analyze the resume and job description.
- Uses "**gemini-1.5-flash**" model for fast processing.
- Returns the AI-generated text response.

## 2. PDF Processing & Conversion

```
import pdf2image
import base64
import io
from PIL import Image

def input_pdf_setup(uploaded_file):
    POPPLER_PATH = r"C:\Program Files\poppler\Library\bin"
    if uploaded_file is not None:
        images = pdf2image.convert_from_bytes(uploaded_file.read(),
poppler_path=POPPLER_PATH)
        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()}]
    return pdf_parts[0]
```

- Converts a **PDF** into an **image** using **pdf2image**
- Uses **Poppler** for PDF processing
- Encodes the image in **Base64** for AI input

## 4. Streamlit UI for User Interaction

```
st.set_page_config(page_title="HirEase AI", layout="centered")
st.title("🤖 HirEase AI - Smart Resume Analyzer")
st.write("Upload your resume and paste the job description to analyze how well your profile matches the job role.")

uploaded_file = st.sidebar.file_uploader("Upload your resume (PDF format only)", type=["pdf"])
input_text = st.sidebar.text_area("Paste the job description here", height=150)
submit1 = st.button("📄 Get Resume Details")
submit3 = st.button("📊 Get Resume Match %")
```

- Creates the web interface using **Streamlit**
- Provides **file upload** and **text input** options

- Includes **buttons** to trigger analysis
- 

## 5. AI-Powered Resume Evaluation & Match Calculation

```

input_prompt1 = """
As an experienced Technical HR Manager, analyze the resume against the job description.
Provide a detailed evaluation on how well the candidate meets the required
qualifications, skills, and experience.
Highlight key strengths, gaps, and suggestions for improvement.
"""

input_prompt3 = """
As a trained ATS scanner, evaluate the resume's alignment with the job description.
Calculate an accurate match percentage, list missing keywords, and provide final
thoughts.
"""

if (submit1 or submit3) and uploaded_file:
    pdf_content = input_pdf_setup(uploaded_file)
    if pdf_content:
        prompt = input_prompt1 if submit1 else input_prompt3
        response = get_gemini_response(input_text, pdf_content, prompt)
        st.subheader("💡 AI Analysis Result:")
        st.write(response)

```

- Triggers **AI evaluation** based on **user selection**
- Uses **predefined prompts** for **HR-style analysis** and **ATS ranking**
- Displays the **AI-generated response**

## Conclusion

HirEase AI is an **intelligent resume analyzer** that leverages **AI & ATS methodologies** to help users optimize their resumes for better job prospects. Its **user-friendly design, AI-powered insights, and real-time analysis** make it a valuable tool for job seekers and recruiters.