

Resume Reviewer

Project Group Name: Achievers



Team members:

1. Achyuth Kumar Valeti
2. Anusha Nelluri
3. Venu Babu Linga

GitHub Link:

Presentation Link:

YouTube Link:



Introduction:

Resume Reviewer is an application used to review the candidates resumes and grade them using the NLP (Natural Language Processing) techniques. An individual can preview their resumes and the mandatory information like name, contact details will be highlighted or not. It will also highlight the candidate's level of experience and focus on the skills that the user has provided. It assists users with recommended courses, as well as the certifications and qualifications required to have a resume selected, using the above-mentioned skills.

Background:

Now a days, it has become difficult for individuals to build their professional resume to the appropriate job openings. Especially the skills and certifications needed from professionals to the clients. Various skill sets are needed for the distinct roles of work. It became a very hectic task for the individuals to fulfill the skills needed for applying a new job.

With the help of this software individuals can know prior regarding the needed skills, resume layouts, courses needed before applying for the job.

Many companies are using resume analyzer software to shortlist the candidates for job openings based on the level of experience and the resume rating. It helps the Human resources department to select candidates for the job interview.

Features:

Basic Details: This feature helps user to fetch basic details from resume.

★★★★ **Resume Analysis** ★★★★★

Hello Data Scientist

Your Basic Details

Name: Data Scientist

Email: info@qwikresume.com

Great!! You are at intermediate level..

Skills Recommendations: Candidates that are multi-skilled will have a bright career in any sector. However, many of the applicants are unaware of the necessary qualifications for the post. With a set of abilities based on the field based, we used the skill suggestions functionality.

Skills Recommendation💡

Skills that you have

Design ×

Algorithms ×

Cloud ×

Documentation ×

Litigation ×

Aws ×

Conversion ×

Analytics ×

Process ×

Machine learning ×

Etl ×

Data analysis ×

Scala ×

Mining ×

Analytical ×

Research ×

Email ×

R ×

Sphinx ×

Website ×

Automation ×

Analysis ×

Programming ×

Matlab ×

Python ×

Physics ×

Spark ×

Reporting ×

See our skills recommendation

** Our analysis says you are looking for Data Science Jobs.**

Career Recommendations: Reviewing the job description is the most important part of creating a resume since it displays what the company is looking for in a candidate. After reviewing the curriculum vitae, candidates can preview the jobs that are available for them to apply for, which will be displayed in their career recommendations.

Recommended skills for you.

Data Visualization ×

Predictive Analysis ×

Statistical Modeling ×

Data Mining ×

Clustering & Classification ×

Data Analytics ×

Quantitative Analysis ×

Web Scraping ×

ML Algorithms ×

Keras ×

Pytorch ×

Probability ×

Scikit-learn ×

Tensorflow ×

Flask ×

Streamlit ×

Recommended skills generated from

Adding these skills to resume will boost🚀 the chances of getting a Job👛

Course Recommendations: Certifications and courses are becoming increasingly required for job vacancies in recent years. Because of their existing understanding of technology, certified individuals are more likely to receive job offers from organizations. As a result, resumes with certified candidates are in high demand. We used the feature to display the skill's suggested credentials in addition to the possibility of being hired.

Courses & Certificates 🎓 Recommendations

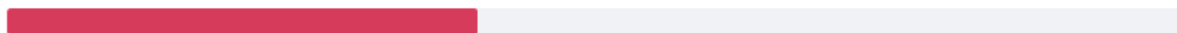
Choose Number of Course Recommendations:



- (1) [Intro to Machine Learning with TensorFlow](#)
- (2) [Data Science Foundations: Fundamentals by LinkedIn](#)
- (3) [Introduction to Data Science](#)
- (4) [Programming for Data Science with Python](#)

Resume Score: Resume Score compares each resume to important criteria that hiring managers and applicant tracking systems (ATS) look for when considering a job application. Applicant tracking systems are reported to reject 50% to 75% of job applications automatically. Resume Score is a brief explanation of which aspects of your resume would be used for certain jobs.

Resume Score 📄✍️



**** Your Resume Writing Score: 40****

**** Note: This score is calculated based on the content that you have added in your Resume. ****

Resume writing Tips & suggestions: We do have a section called resume tips & ideas, which will assist the user in preparing their resume with some standard information such as achievements, hobbies, and the number of projects they have worked on.

Resume Tips & Ideas💡

[+] Awesome! You have added Objective

[-] According to our recommendation please add Declaration📝. It will give the assurance that everything written on your resume is true and fully acknowledged by you

[+] Awesome! You have added your Hobbies⚽

[-] According to our recommendation please add Achievements🏆. It will show that you are capable for the required position.

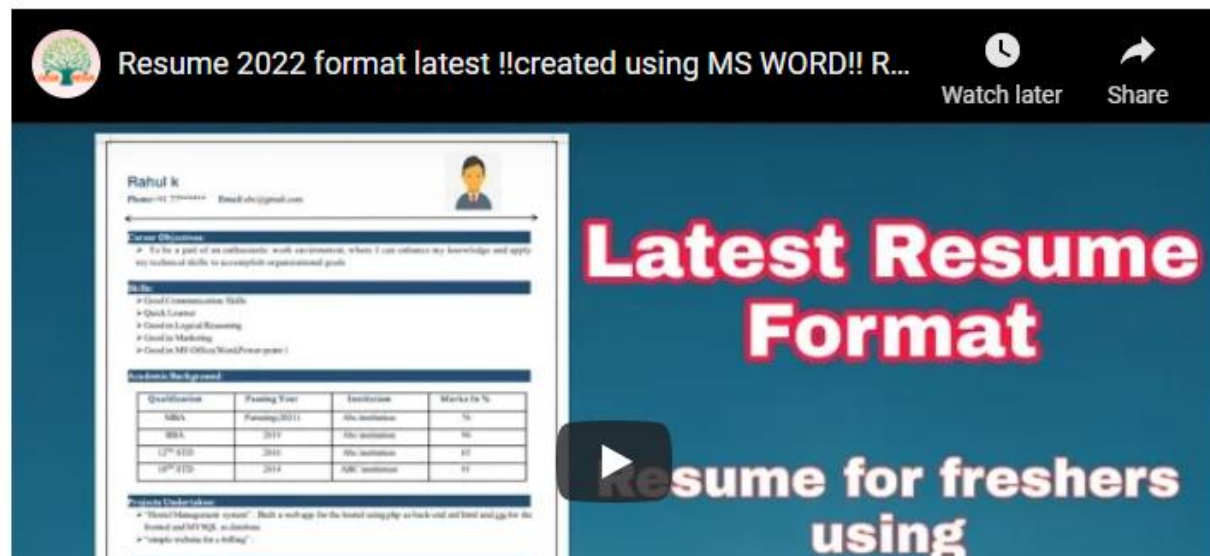
[-] According to our recommendation please add Projects💻. It will show that you have done work related the required position or not.

YouTube video recommendations to build resumes: We've added video player links to help you prepare a better CV by correcting the layouts and content words.

Bonus Video for Resume Writing Tips💡

✅ **Resume 2022 format latest !!created using MS WORD!!**

Resume for freshers of BE, BCOM, BSC,BCA,MCA,MBA



Design Implementation:

1. Firstly, we created a new project using “WebStorm” and used “streamlit” framework as it is open-source and compatible with major python libraries that are helpful to our project like pyreparser, pdfminer3 and pandas etc.

2. We have installed the below listed packages using “pip install ‘package-name’” command which are required for our project.

- a. pdfminer3
- b. pyreparser
- c. streamlit
- d. pandas
- e. pafy
- f. plotly
- g. pymysql
- h. streamlit-tags
- i. Pillow

3. Before starting the implementation process, we have divided the project into different modules as below:

- a. User's & Admin Section
- b. Resume Score
- c. Career Recommendations
- d. Resume writing Tips suggestions
- e. Courses Recommendations
- f. Skills Recommendations
- g. YouTube video recommendations

4. We are going to use the libraries listed below, each of which has its own set of benefits.

- a. **Streamlit**: It is compatible with popular Python libraries like scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, and Matplotlib etc. As widgets are considered as variables in Streamlit, no call backs are required. Streamlit monitors changes to the connected Git repository for updates, and the application is automatically deployed in the shared link.

'Streamlit' can be imported as below.

```
import streamlit as st
```

- b. **pandas**: It includes a variety of tools, from parsing different file formats to transforming an entire data table into a NumPy matrix array.

'pandas' can be imported as below.

```
import pandas as pd
```

- c. **base64, random**:

Base64 encoding is a type of byte-to-ASCII character conversion.

Python Random module is a built-in module for generating random numbers.

```
import base64, random
```

- d. **datetime**: Date and time manipulation classes are provided by the datetime module.

```
import time, datetime
```

- e. **pyreparser**: A resume parser that extracts information from resumes in a simple manner. All the user details can be extracted from resume like name, mobile number, E-mail etc. It will return an output as list of dictionary objects.

```
from pyresparser import ResumeParser
```

- f. **pdfminer3:** pdfminer3 is a tool that extracts data from PDF files. It is distinct among PDF-related tools in that it focuses completely on extracting and analysing text data. It has different objects. 'Pdfparser' reads data from a file and 'PDFDocument' saves the data. To handle the page contents, use the 'PDFPageInterpreter'. The PDFResourceManager stores shared resources like fonts and graphics.

```
from pdfminer3.layout import LAParams, LTTextBox
from pdfminer3.pdfpage import PDFPage
from pdfminer3.pdfinterp import PDFResourceManager
from pdfminer3.pdfinterp import PDFPageInterpreter
from pdfminer3.converter import TextConverter
```

- g. **io:** File-related input and output activities can be controlled using io module.
- h. **Streamlit tags:** In Streamlit, we can use this custom component to add Tags.
- i. **PIL (Python Imaging Library):** PIL is a free and open-source Python library that makes it simple to create and modify digital images.

```
from PIL import Image
```

- j. **pymysql:** It is a library that is used to connect to database (MySQL) from python and it is a pure python library. It is an interface and used to connect to a MySQL database server from Python.

```
import pymysql
```

- k. **Pafy:** It is a Python library that allows you to download YouTube videos and extract metadata. All the metadata can be retrieved such as view count, duration, rating, author, thumbnail, and keywords.

```
import pafy
```

- l. **Plotly.express:** The plotly.express is an open-source library that can generate the complete Figure at once.

```
import plotly.express as px
```


Execution:

In this project, there are two modules: Candidate and Employer.

Firstly, we need to connect to the database by supplying the host, username, password, and database name. As we used the 'Streamlit' framework to develop this application, it allows to build front end for the project by simply using the python code. It allows us to quickly develop attractive online applications for data science and machine learning.

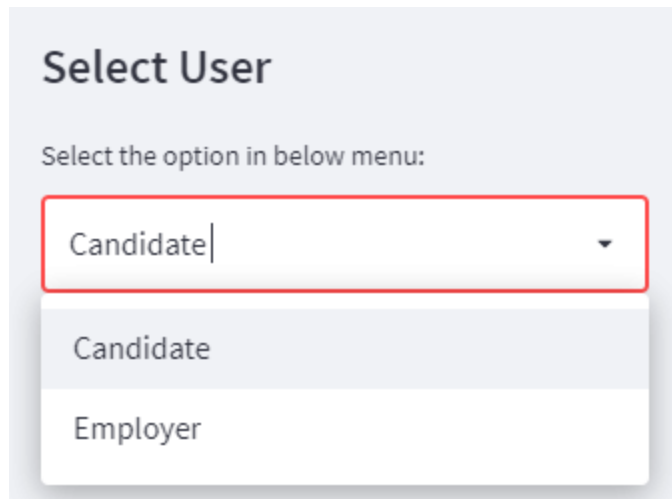
Streamlit allows us to configure our web page effectively use `set_page_config()`. It must be the first Streamlit command in the script and can only be called once per app. We used the 'page_title' and 'page_icon' attributes to set the page title and icon as below.

```
st.set_page_config(
    page_title="Resume Reviewer",
    page_icon='./Logo/resume_logo.ico',
)
```

Candidate Execution:

In run () method, we used the following components to build our webpage.

1. **st.title():** The title of our web page was set to "Smart Resume Analyzer" using this component.
2. **st.sidebar():** It is used to build the sidebar.
3. **st.markdown():** Used to display the text to user as "Select User" in our web page.
4. **st.selectbox():** It is used to make a drop-down list and it contains "Candidate" & "Employer" options.



5. st.image(): It is used to display the image that has been loaded into the application.

Here is the code that we used:

```
st.title("Smart Resume Analyser")
st.sidebar.markdown("# Select User")
selected_users = ["Candidate", "Employer"]
choice = st.sidebar.selectbox("Select the option in below menu:", selected_users)
logo_img = Image.open('./Logo/SRA_Logo.jpg')
logo_img = logo_img.resize((245,245))
st.image(logo_img)
```

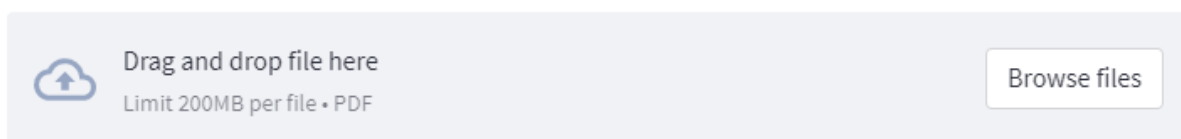
Image.open() helped us in opening and displaying an image on our web page using the PIL library.

Database creation & Connection: Using SQL queries, we firstly created the database using "CREATE DATABASE" command and executing the command by passing it to the cursor.

Then, we created the table using "CREATE TABLE" command and declared all the column names and types as ID, NAME, Email_ID, resume_score, Timestamp etc.

If the user is set to "Candidate," the first step is to upload their résumé.

6. st.file_uploader: It assists us whenever we need to upload a file. Currently, we were supporting only ".pdf" files.



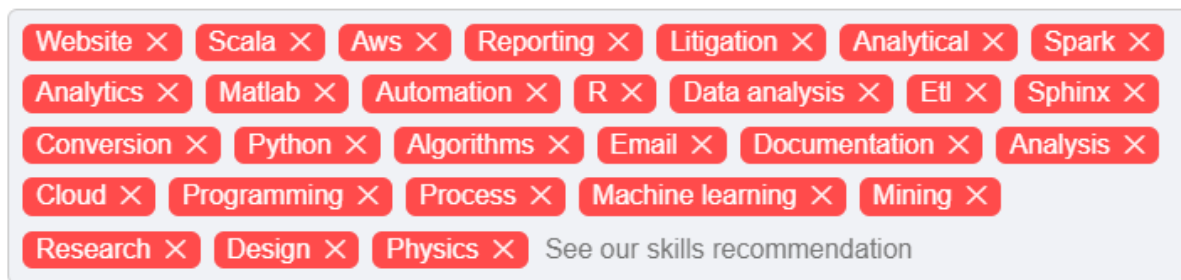
Once the file is uploaded, we were displaying it on our web page for reference. We were also saving it in the "Uploaded Resumes" folder on our machines.

Using 'ResumeParser', we were extracting candidate information by using the submitted resume as an input and saving it in a variable. This variable holds the all the details of candidate like name, phone number, email, number of pages etc.

We categorized candidate's level of experience as "Fresher," "Intermediate," or "Experienced" based on the number of pages in their resume and showing the same information to the candidate.

Skills that candidate have:

7. st_tags: It is used a custom component to have the tags in streamlit. Using st_tags, we printed all the candidate skills that is extracted from the resume in the format below.



Recommended Skills: We then displayed the user's "Recommended skills" in the next step. We have saved all the candidate's skills. If any skill matches one of the skills included in our list, then we were displaying recommendations that are most appropriate for that job. We have added skill recommendations for courses like datascience, web development, android development, ios development & uiux development.

Courses, Certificates & Recommendations: We were also giving the reference links to the courses that they might use when upgrading their resumes. Course recommendations can be chosen from 1 to 10.

Resume Tips & Ideas: We provided suggestions and thoughts to the user in this part. The resume score will be improved if the resume contains general things like Objective, Declaration, Hobbies, Achievements & Projects. If the candidate's resume does not include these things, it will be suggested to include it. Resume score will be shown to the user based on the content that we added to the resume.

8. st.balloons(): It helps us to show the celebratory balloons.

Inserting data into a database table:

After completing all these steps, we have inserted all the data into a table using insert query command "insert into database-table"

Video Tips: We were giving reference YouTube videos along with their names using the "pafy" library. "pafy" library helps us to extract the title from the video and display it to the user. We have chosen a few YouTube videos and displaying them to the candidates as a reference.

Employer Execution:

In the employer section we fetched all the user data which is extracted and stored in table "user_data" in the candidate section from the candidate's resumes. The data is stored in the "data frame (df)". And we displayed using `st.dataframe()` with the respective column names. We can also download the data in .csv file using the download link.

we have plotted the columns user_level and predicted_Field counts with their values from the user_data table in the pie chart using the `st.plotly_chart()`.

9. st.plotly_chart(): Plotly is a Python graphing package. This function's inputs are like those for Plotly's plot () function.

Conclusion:

We have implemented most of the concepts learned in class and from ICP's. By doing this project we gained hands-on experience about implementation of NLP in real-time projects from scratch.

References:

1. <https://docs.github.com/en/repositories>
2. <https://www.youtube.com>
3. <https://www.google.com>
4. <https://en.wikipedia.org/>
5. <https://stackoverflow.com/>
6. <https://geeksforgeeks.org/>
7. <https://streamlit.io/>
8. <https://docs.streamlit.io/>
9. <https://pypi.org/project/PyMySQL/>
10. <https://pymysql.readthedocs.io/en/latest/>
11. <https://www.apache.org/>