PROJECT REPORT

ON

AI RECRUITER – SHORTLIST A SUITABLE CANDIDATE FOR SPECIFIC JOB ROLE

By Team Enigma

Team Members:

1. **Shivam Yadav**
2. **Srivatsa Prakki**
3. **Siddhant Kesarkar**
4. **Mandar Patil**

Under the guidance of **Mrs. Lalitha Gayatri Lolla** and **Mr. Hemant Kumar Gahlot**

1. INTRODUCTION

I. Overview

An AI Recruiter Chatbot which assess the candidate on his/her technical knowledge about the job post, Strength of Video resume and his/her personality via a quick conversation.

II. Purpose

In today's world, opportunity is less and competition is more. So, there are lots of resume for a single available job post and shortlisting suitable candidates is tedious and requires lots of man hours.

2. LITERATURE SURVEY

I. Existing Problem

52% of talent acquisition leaders say that the most difficult part of their job is to shortlist the right candidate and 3% of candidates never hear back from a company after one touchpoint. On the flip side, it's a challenge for employers to communicate well with all their candidates. For high volume recruiting, this would require communicating with thousands of candidates, in addition to a recruiter's normal screening functions and other duties.

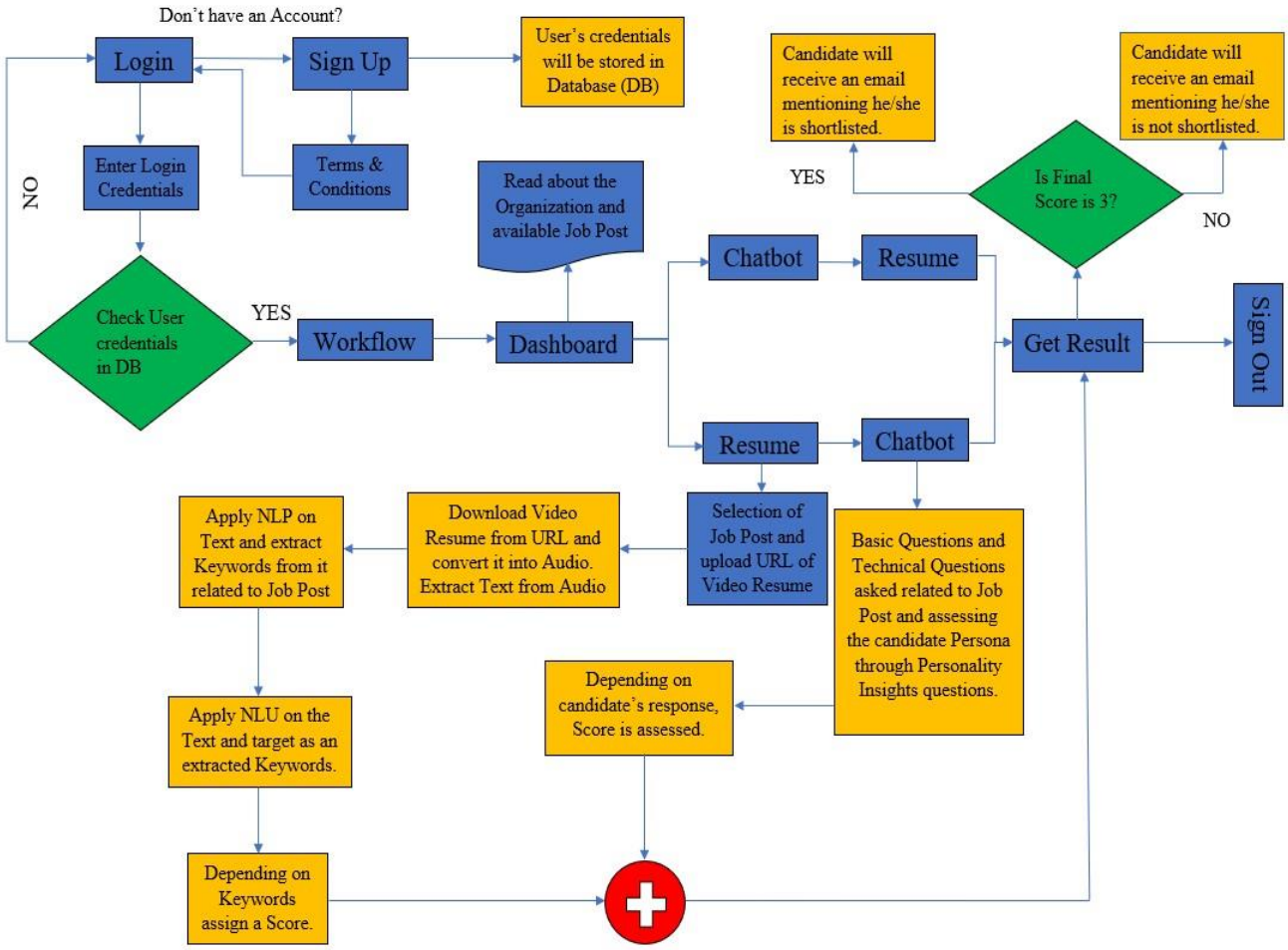
II. Proposed Solution

In this project, we will be making an Artificial Intelligence enabled Chatbot to screen through the candidate's application, conduct technical and personality related assessment and rate the candidate accordingly. Finally sending the confirmation e-mail to the shortlisted candidates.

To take it a step further we will be using video resume instead of traditional old text resume.

3. THEORETICAL ANALYSIS

I. Block Diagram



II. Software Designing

a. Front End templates were designed in HTML and CSS.
b. Chatbot was designed using IBM Watson Assistant tools.
c. Cloud function, NLU, NLP and server-side coding were done using Python.

4. EXPERIMENTAL INVESTIGATION

During our investigation for this project, we saw working of already deployed Chatbots where we found that video of the candidate is recorded during conversation to evaluate micro-expressions which were still under development i.e. the accuracy was very low.

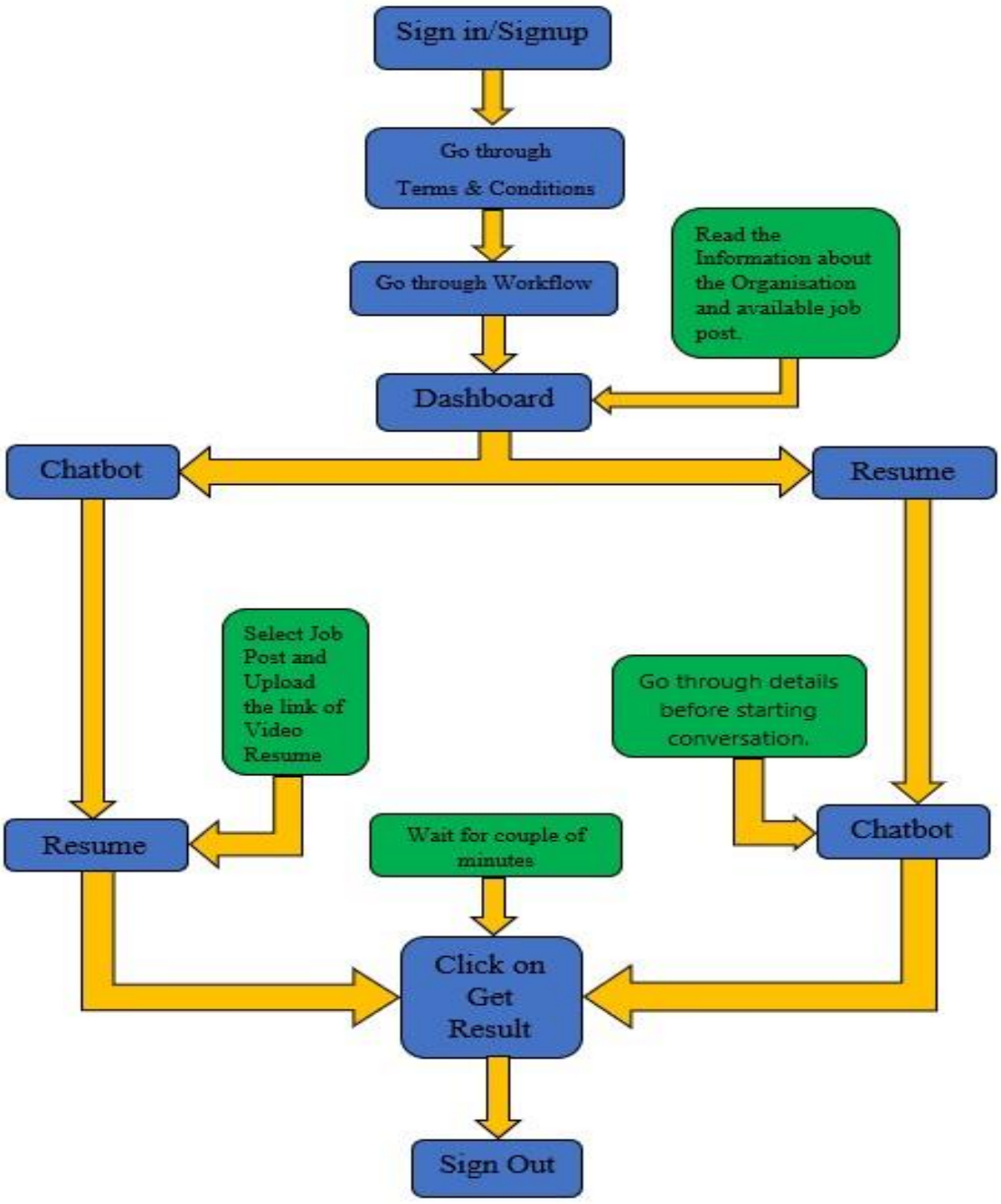
Therefore, to make our Chatbot capable of analysing micro-expressions in the future we are taking, Video Resume.

In many online Technical quizzes, we found that there is a possibility of copying the code and executing it in the local PC. To avoid this, we went with the idea of code snippets (showing images of parts of code) and asking the user to answer.

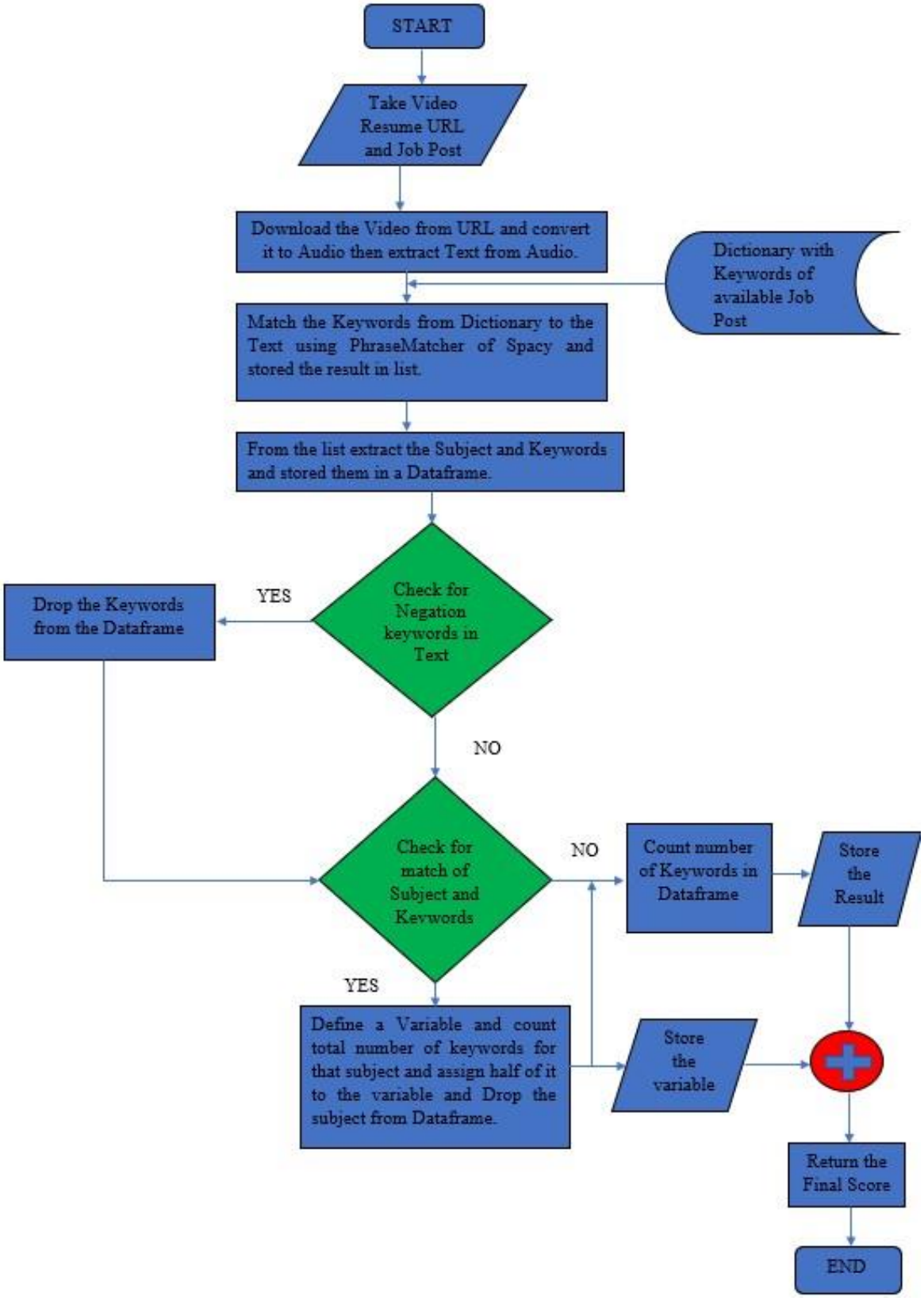
There is a very low yet a slight possibility where the candidate may use negation words to describe his/her knowledge about the tool/technology, which we don't want to use for the assessment of the resume. Therefore, NLU is used to remove all such tool/technology from the assessment.

5. FLOWCHART

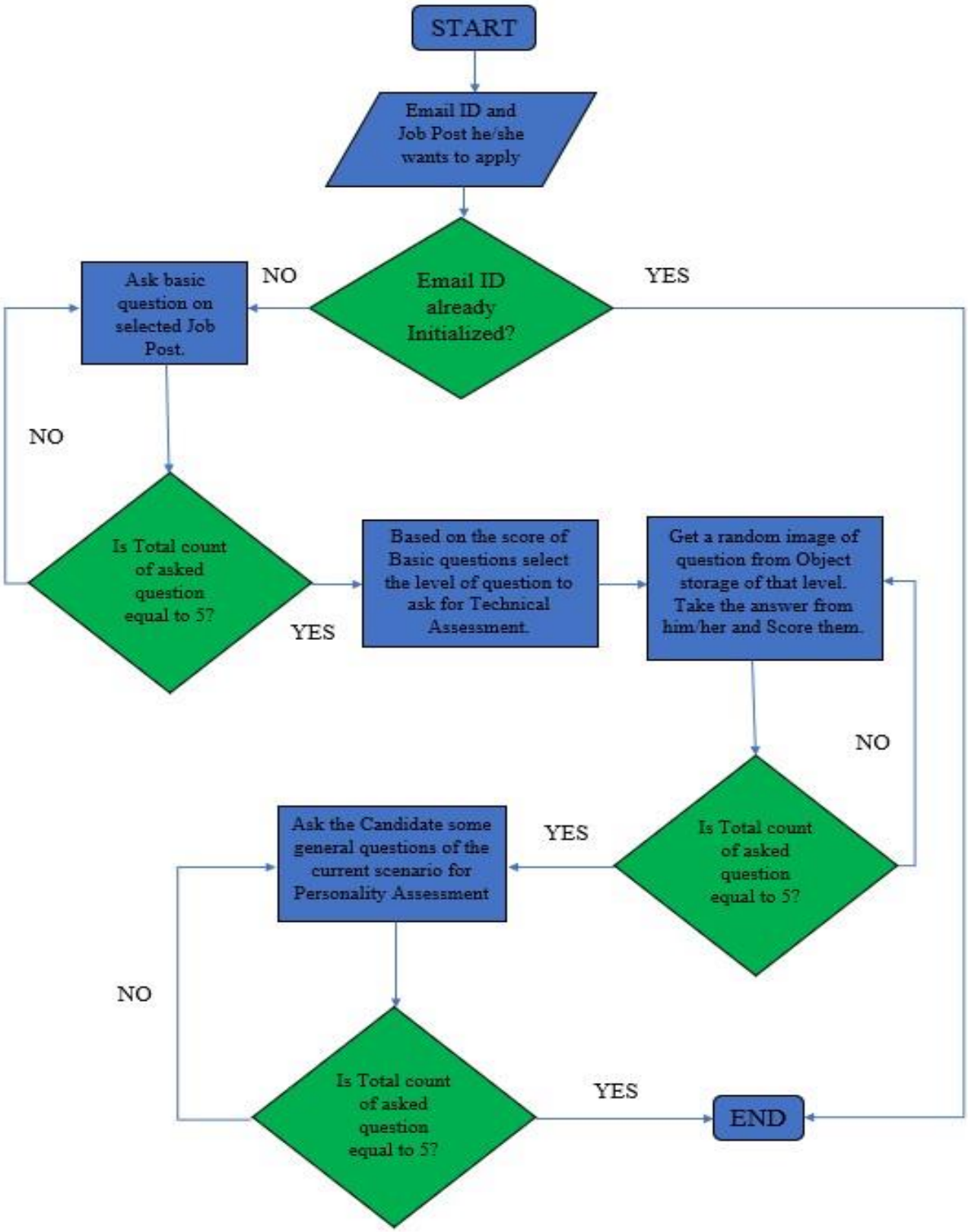
I. Front End



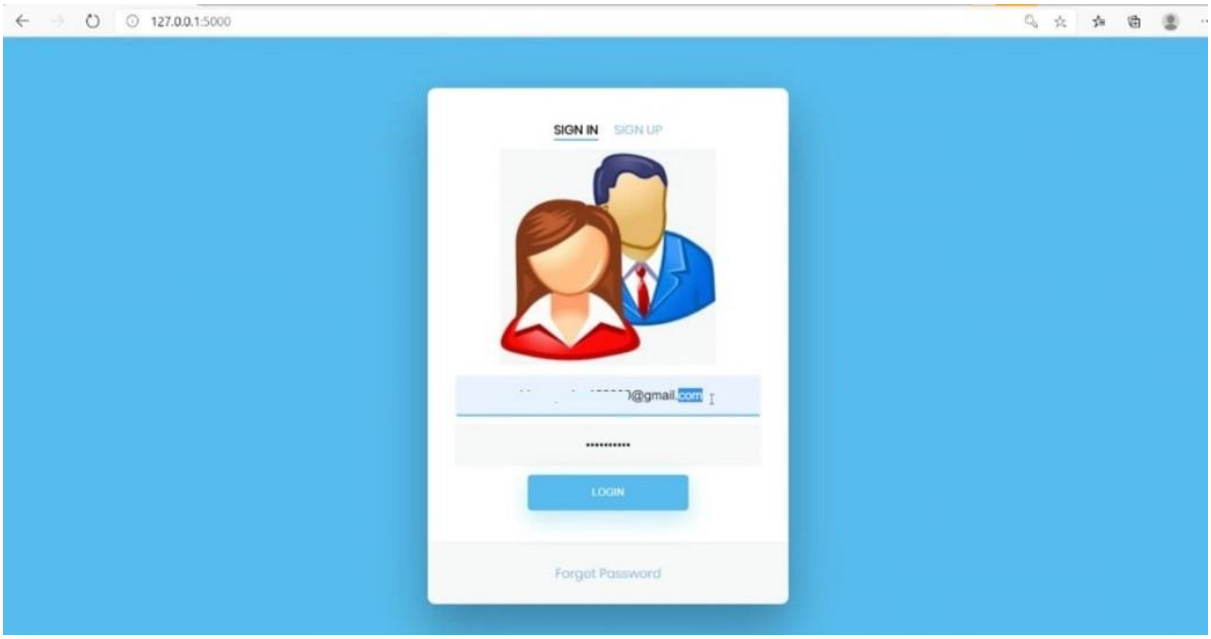
II. Video Resume

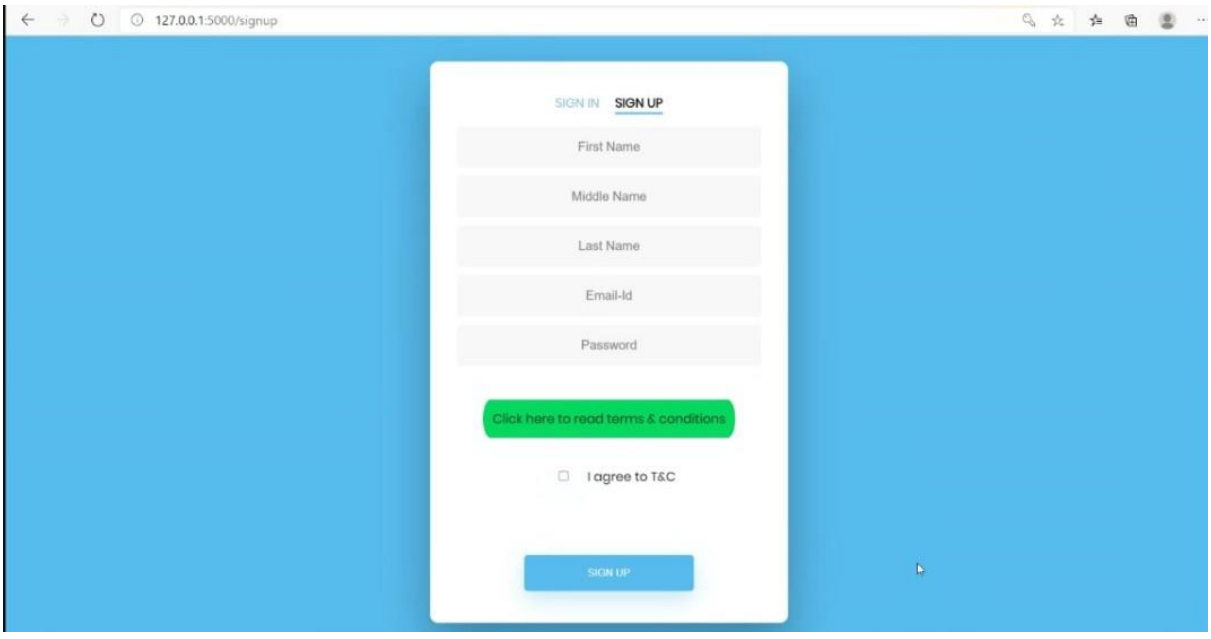


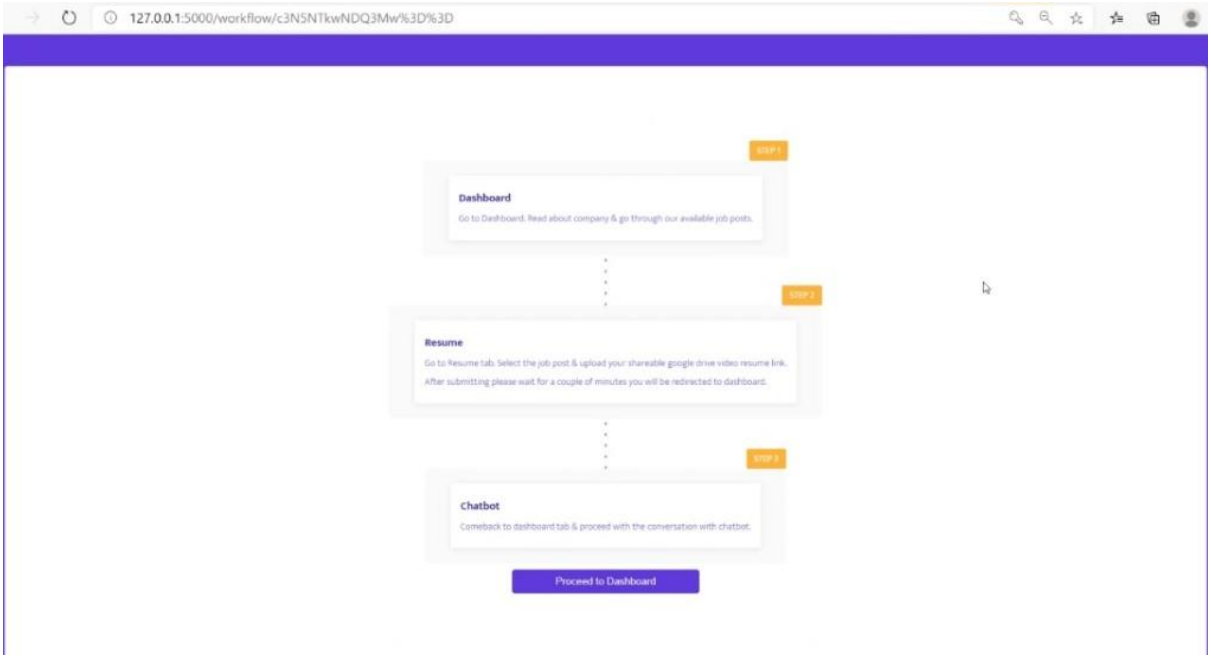
III. Watson Assistant

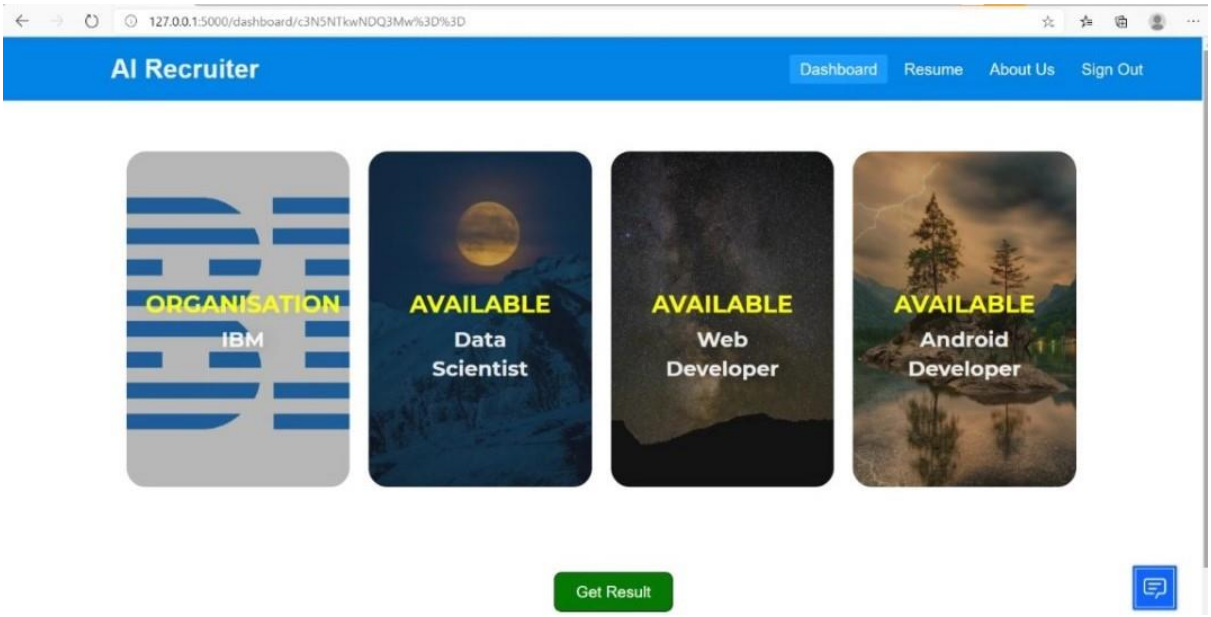


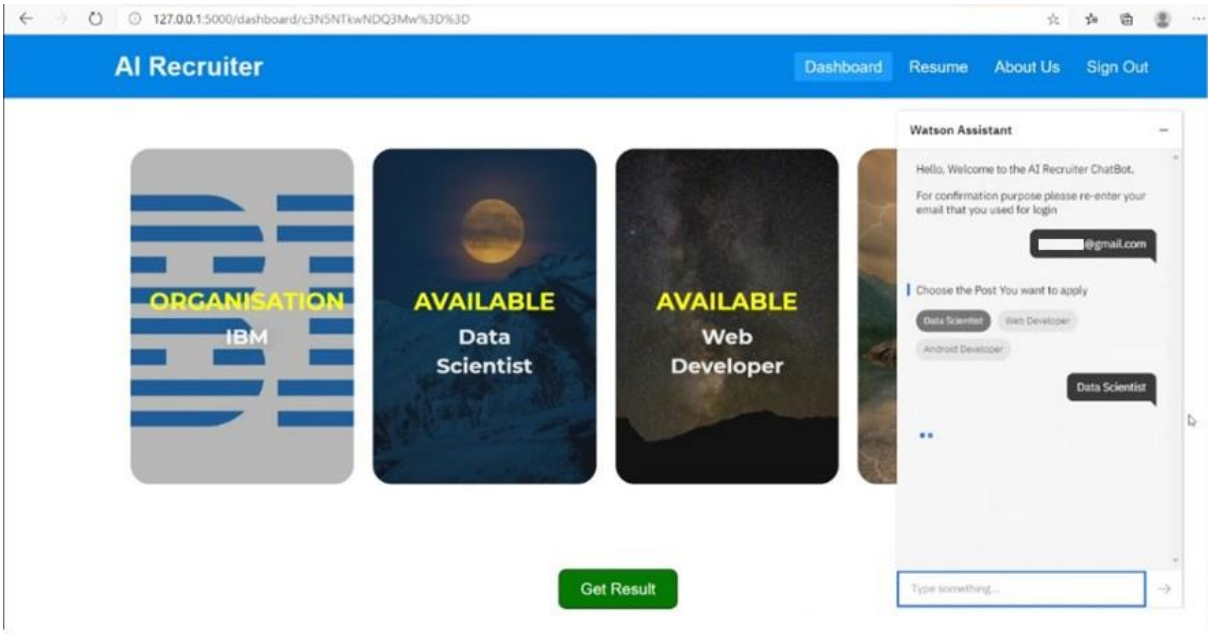
6. RESULT

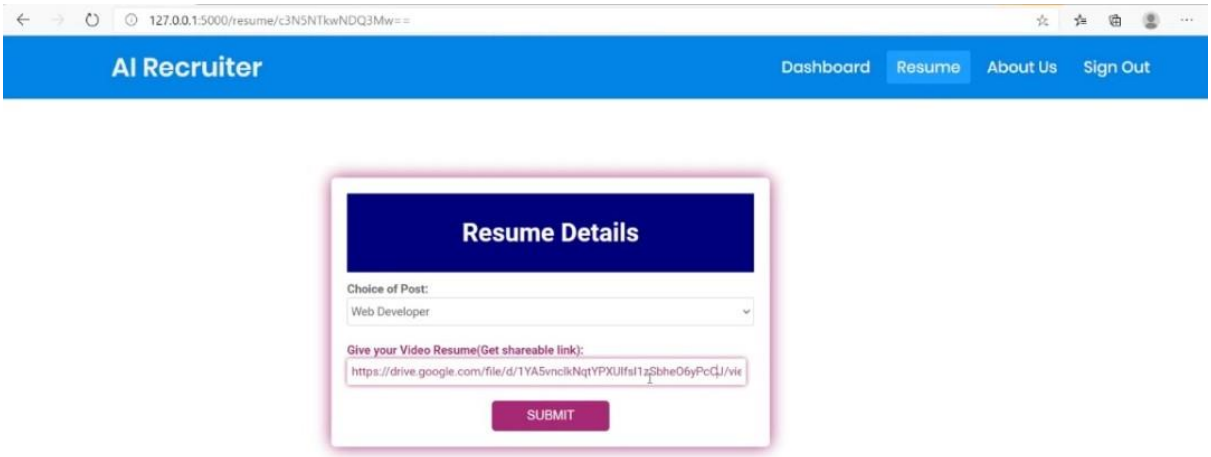

127.0.0.1:5000
SIGN IN
SIGN UP
LOGIN
Forget Password


127.0.0.1:5000/signup
SIGN IN
SIGN UP
First Name
Middle Name
Last Name
Email-Id
Password
Click here to read terms & conditions
I agree to T&C
SIGN UP

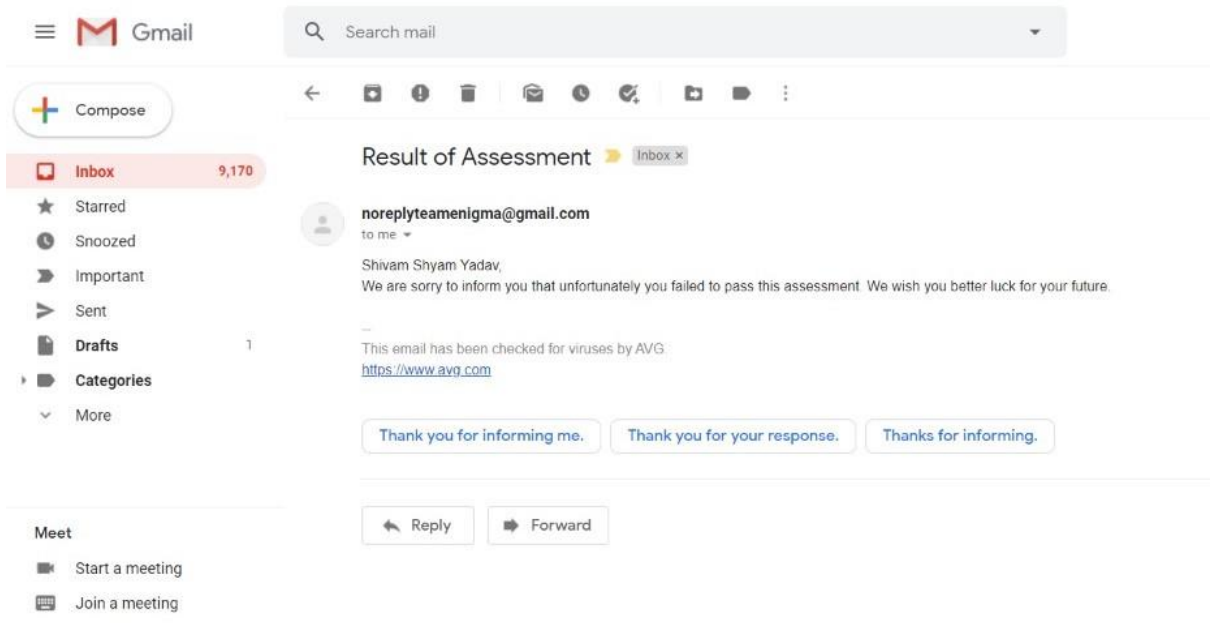

127.0.0.1:5000/workflow/c3N5NTkwNDQ3Mw%3D%3D
Dashboard
Go to Dashboard. Read about company & go through our available job posts.
Resume
Go to Resume tab. Select the job post & upload your shareable google drive video resume link.
After submitting please wait for a couple of minutes you will be redirected to dashboard.
Chatbot
Comeback to dashboard tab & proceed with the conversation with chatbot.
Proceed to Dashboard


127.0.0.1:5000/dashboard/c3N5NTkwNDQ3Mw%3D%3D
AI Recruiter
Dashboard
Resume
About Us
Sign Out
ORGANISATION
IBM
AVAILABLE
Data
Scientist
AVAILABLE
Web
Developer
AVAILABLE
Android
Developer
Get Result





After clicking on the submit button wait for a couple of minutes, you will be redirected to dashboard tab.
Please do not refresh the page.



7. ADVANTAGES & DISADVANTAGES

I. Advantages:

- Rediscovery of candidate is possible when same position is vacant again.
- Depending on the answer of technical question the subsequent questions of varying difficulty level will be asked.
- It is more useful in the Pandemic and Post-Pandemic period for hiring of suitable candidates.
- Unbiased playing field for all the candidates.
- Improve candidate experience and engagement.

II. Disadvantages:

- Issues with accuracy and reliability
- Too much dependency on certain keywords
- Lacks nuance of human judgement

8. APPLICATIONS

- Chatbot is an important application of AI in Recruitment
- AI powered Video Interviews in Recruitment
- Automated Talent Sourcing through AI in Recruitment
- Talent rediscovery can be accomplished through AI in Recruitment
- Sentiment Analysis through AI in Recruitment

9. CONCLUSION

- In any organization the most vital organ is human resources.
- In the today's business atmosphere, the main challenge of the manager's is the selection of the right candidate, then development of the candidate and the retention of the candidate in the organization.
- The recruitment of the effective, efficient and deserving candidates is very difficult task for the human resource department of the company.
- The recruitment process is based on the requirement of the organization.
- The decision of the human resource department to recruit one wrong candidate can prove fatal for the organization, in terms of money, time and reputation of the company.
- The selection process ensures that the best suitable candidates in terms of skills, education and experience should be chosen for the particular job.
- The selection's process main goal is to match the job requirements with the person's technical knowledge and individual characteristics.

10. FUTURE SCOPE

- Since video resume is being taken, the concept of micro expression can be implemented in the future. Skills matching of candidate to be recruited with previously hired or already working employee to get most suitable candidate for org. (Candidate matching)
- Follow-up activity with organization can be planned for the selected candidate in which the company can rate the employee work – Can be used in bigger organization.
- Company oriented UI can be created which will take the required details form the company and Chatbot will be created accordingly, further it will be deployed on candidate-oriented UI and most suitable candidates will get shortlisted.

11. BIBLIOGRAPHY

[1] Aslam, F.A., Mohammed, H.N., Mohd, J.M., Gulamgaus, M.A. and Lok, P.S., 2015. Efficient Way of Web Development Using Python and Flask. *International Journal of Advanced Research in Computer Science*, *6*(2).

[2] Lokhande, P. S., Fankar Aslam, Nabeel Hawa, Jumal Munir, and Murade Gulamgaus. "Efficient way of web development using python and flask." (2015).

[3] https://www.w3schools.com/w3css/

[4] https://www.w3schools.com/html/

[5] Memeti, S., & Pllana, S. (2018). PAPA: A parallel programming assistant powered by IBM Watson cognitive computing technology. *Journal of computational science*, *26*, 275-284.

[6] https://cloud.ibm.com/docs

[7] Manning, Christopher, and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[8] Dahlgren, Kathleen, and Edward Stabler. "Natural language understanding system." U.S. Patent 5,794,050, issued August 11, 1998.

[9] https://towardsdatascience.com/how-to-extract-keywords-from-pdfs-and-arrange-in-order-of-their-weights-using-python-841556083341

APPENDIX:

Source Code: Video Resume Analysis

1. Importing Important Libraries

```
from google_drive_downloader import GoogleDriveDownloader as gdd
import moviepy.editor
from ibm_watson import SpeechToTextV1
from ibm_watson import NaturalLanguageUnderstandingV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibm_watson.natural_language_understanding_v1 import Features, SentimentOptions
```

2. Extracting Text from Video

```
gdd.download_file_from_google_drive(file_id=self.file_id, dest_path=self.destinatio
n_path)
self.video = moviepy.editor.VideoFileClip(self.destination_path)
self.audio = self.video.audio
self.api = IAMAuthenticator('FB-BtLEbKl4axbacSnO4x6DoDXq_VgRSvhTKdVKIfFfb')
self.speech_to_text = SpeechToTextV1(authenticator=self.api)
self.speech_to_text.set_service_url('https://api.eu-gb.speech-to-
text.watson.cloud.ibm.com/instances/af924047-1418-4683-955f-4376ca2b7707')
with open(self.audio_path, 'rb') as self.audio_file:
    self.result = self.speech_to_text.recognize(audio=self.audio_file, content_typ
e='audio/mp3').get_result()
```

3. Creating Keywords

```
self.job_dict = {'Data Scientist':self.data_scientist, 'Web Developer':self.web_dev
eloper, 'Android Developer':self.android_developer}
```

4. Natural Language Understanding

```
self.authenticator = IAMAuthenticator('LSSpBBaLUWaNu2X35ptYDGDmfoVTV5_seU2MXsiJs6yF'
)
self.natural_language_understanding = NaturalLanguageUnderstandingV1(version='2019-
07-12', authenticator=self.authenticator)
self.natural_language_understanding.set_service_url('https://api.eu-gb.natural-
language-understanding.watson.cloud.ibm.com/instances/ad96f725-ca09-4e44-a5b9-
55a64883b953')
self.response = self.natural_language_understanding.analyze(text = text,
      features=Features(sentiment=SentimentOptions(targets=keyword))).get_result()
self.dict_ = eval(json.dumps(self.response, indent=2))
self.negative_keyword = []
for lst in self.dict_['sentiment']['targets']:
    if lst['score'] <= 0.75:
        self.negative_keyword.append(lst['text'])
return self.negative_keyword
```

5. Creating Profile of Candidate

```
self.keyword, self.total_keywords = self.keyword_count(job_post)
for column in self.keyword.columns:
      self.columns_keyword[column] = [nlp(text) for text in self.keyword[column].drop
na(axis=0)]
self.matcher = PhraseMatcher(nlp.vocab)
for column in self.keyword.columns:
     self.matcher.add(column, None, *self.columns_keyword[column])
self.doc = nlp(self.text)
self.keywords = "\n".join(f'{i[0]} {i[1]} ({j})' for i,j in Counter(self.d).items())
self.df = pd.read_csv(StringIO(self.keywords),names = ['Keywords_List'])
self.df1 = pd.DataFrame(self.df.Keywords_List.str.split(' ',1).tolist(),columns = ['S
ubject','Keyword'])
self.df2 = pd.DataFrame(self.df1.Keyword.str.split('(',1).tolist(),columns = ['Keywor
d', 'Count'])
self.df3 = pd.concat([self.df1['Subject'],self.df2['Keyword'], self.df2['Count']], ax
is =1)
self.df3['Count'] = self.df3['Count'].apply(lambda x: x.rstrip(")"))
self.dataf = pd.concat([self.df3['Subject'], self.df3['Keyword'], self.df3['Count']],
 axis = 1)

self.negative_keyword = self.nlu(self.text, list(self.dataf['Keyword']))

self.column_score, self.matched_keyword = self.scoring_column(self.keyword_match, sel
f.col_name, self.col_count)

self.score = self.dataf['Keyword'].count()
self.profile_score = self.column_score + self.score
self.final_score = self.profile_score / self.total_keywords
```