International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Charting Your Path Comprehensive Career Guidance.

***Muneer Unnisa*[1], *Muhammadi Azmath*[2].**

[1,2] *Department of IT, Nawab Shah Alam Khan College of Engineering and Technology, Hyderabad, India*

Email: www.muneerunnisa07@gmail.com

ABSTRACT

The rapid evolution of technology and the diversification of engineering disciplines present significant challenges for engineering graduates in identifying suitable career paths. This paper proposes a comprehensive career guidance system designed to assist engineering students and recent graduates in navigating the job market. The system leverages advanced data analytics, machine learning algorithms, and industry insights to provide personalized career recommendations. Key features include skill assessment tools, industry trend analysis, and a matching algorithm that aligns users' profiles with job opportunities. By integrating academic performance, personal interests, and market demand, the proposed system aims to bridge the gap between education and employment, ensuring that graduates embark on successful and fulfilling career journeys.

Keywords: Chatbots, Web predicted, career path, Skills.

1. INTRODUCTION:

Students are often seemed to be confused regarding career choices. because they have no one to guide them. Sometimes, they are unaware of various career opportunities available for them, hence they end up choosing the stream in which they are least interested. Career Guidance system helps them to choose their career which is most suitable for them by providing them with proper career details. Students can even consult the counsellors if they are still confused about career choices. This system not only helps the students but also the counsellors are benefitted from this, as they get clients through this system. This system ultimately, assist the students making appropriate, satisfying, and interesting educational occupational choices regarding the selection of a career. Hence, it has huge future scope amongst Students, Counsellors, etc.

Career Counselling is a process wherein we gain knowledge about the world of work and know and understand yourself to make a career for future life. This Online Career Guidance System proves to be a decisive point for the career choosing phase. There are immense tools and data made available for students who keep interest in variety of fields therefore our system helps students to make a choice over wide variety without research or any resource, the software takes care of all that.

2. Literature Surveys:

Survey -1:

Title: A Career Guidance Mobile Application Based on Personality

Year: 2017

Author: Faith Too-091673, Faculty of Information Technology Strathmore. University

Methodology: Agile methodology was adopted as the software methodology for developing the application.

Key Findings:

A comprehensive study of the career guidance situation in the country and globally. Analysis of the data collected was done using Google analysis tool and the results used to come up with system. requirements.

Research Gaps:

The mobile application is only usable by smartphone owners with Android operating system. Some information on mobile application and the entire web application. need Internet for one to access them. The mobile application should be developed for other mobile platforms to allow users who do not have Android to access the mobile application's functionality.

3.Requirements:

3.1.1 Hardware Requirements:

Laptop or PC

1. Servers and Hosting

★ Web Servers: For hosting the application (e.g., Apache, Nginx).

* Application Servers: For running backend processes (e.g., AWS EC2, Google Cloud Compute Engine).

Database Servers: For managing databases (e.g., AWS RDS, Azure SQL Database).

*Backup Servers: For storing backup data.

3.1.2Software Requirements:

• Laptop or PC

★ Web Development Frameworks: For creating web- based applications (e.g., React, Angular, Vue.js).

Backend Technologies: Server-side languages and frameworks (e.g., Node.js, Django, Ruby on Rails).

* Database Systems: For storing user data and project information (e.g., MySQL, PostgreSQL, MongoDB).

* Content Management Systems (CMS): For managing and updating content (e.g., WordPress,

Drupal).

Integrated Development Environment (IDE): For code development (e.g., Visual Studio Code, IntelliJ IDEA).

4. System Analysis and modules:

4.1 modules.

The framework totally consists of three modules where the whole process takes place.

Skill set assessment module:

In this module, the candidate takes up an assessment which will be having a combination of psychological and core skills-oriented questions

At the end of the assessment, the candidate can know their scores in each and every skill set separately.

Prediction module:

In this module, with the help of the scores obtained by the candidate in the first module, the prediction takes place with the help of a machine learning algorithm running at the back-end of the web application. The final result in the second module will be the prediction of the suitable department for the candidate.

In this module, a detailed analysis of the candidate's performance will be represented in various

4.2 Architecture:

Creating a system architecture for career guidance after engineering involves multiple layers and components to effectively support engineering graduates in making informed career decisions. Here’s a high-level overview:

1. User Interface Layer

Web/Mobile Application: A platform where users can interact with the system.

Chatbot Integration: For instant responses and guidance.

2. Business Logic Layer

Career Assessment Tools: Aptitude tests, personality assessments, and interest inventories.

Recommendation Engine: Algorithms to suggest career paths based on user profiles, market trends, and skills.

Content Management System: To manage and update career-related content, articles, and resources.

Scheduling System: For booking one-on-one career counseling sessions.

3. Data Layer

User Profile Database: Stores user information, assessment results, and career preferences.

Career Information Database: Contains data about various career paths, required skills, salary expectations, job market trends, etc.

Feedback and Analytics Database: Collects user feedback and system usage data to improve recommendations and services.

4. Integration Layer

Educational Institutions: Integration with colleges and universities for real-time data on courses and programs.

Industry Partners: Partnerships with companies for internships, job openings, and industry insights.

APIs: For integrating external resources such as job boards, online courses, and certification programs.

5. Security and Compliance Layer

Authentication and Authorization: Ensures secure access to the system.

Data Privacy and Compliance: Adherence to data protection laws and regulations.

Regular Audits and Monitoring: To ensure system integrity and security.

5.Conclusion:

A career guidance system for engineering graduates is essential for navigating the myriad of career paths available post-graduation.

Overall, this system plays a pivotal role in aligning students skills, interests, and career aspirations with the dynamic job market, facilitating a smoother transition from education to employment.

6. Source Code:

```
# **1. Importing Necessary Libraries** 🗄

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import pickle

import time

import streamlit as st

from db import *

pickleFile=open("weights.pkl","rb")

regressor=pickle.load(pickleFile) # our model

# **2. Loading Dataset**  C:\Users\syeda\OneDrive\Desktop\Career-Prediction-System-main/Career-Prediction-System-main/data

#C:/Users/syeda/OneDrive/Desktop/Career-Prediction-System-main/Career-Prediction-System-main/Career-Prediction-System-main/data

df = pd.read_csv('mldata.csv')

#df = pd.read_csv('./data/mldata.csv')

df.head()

df['workshops'] = df['workshops'].replace(['testing'],'Testing')

df.head()

print(df.columns.unique)
```

```
n = df['Suggested Job Role'].unique()

print(len(n))

print('The shape of our training set: %s professionals and %s features'%(df.shape[0],df.shape[1]))

# **5. Feature Engineering**

## (a) Binary Encoding for Categorical Variables

newdf = df

newdf.head(10)

cols = df[["self-learning capability?", "Extra-courses did","Taken inputs from seniors or elders", "worked in teams ever?", "Introvert"]]

for i in cols:

    print(i)

    cleanup_nums = {i: {"yes": 1, "no": 0}}

    df = df.replace(cleanup_nums)

print("\n\nList of Categorical features: \n" , df.select_dtypes(include=['object']).columns.tolist())

## (b) Number Encoding for Categorical

mycol = df[["reading and writing skills", "memory capability score"]]

for i in mycol:

    print(i)

    cleanup_nums = {i: {"poor": 0, "medium": 1, "excellent": 2}}

    df = df.replace(cleanup_nums)

category_cols = df[['certifications', 'workshops', 'Interested subjects', 'interested career area ', 'Type of company want to settle in?',

                'Interested Type of Books']]

for i in category_cols:

    df[i] = df[i].astype('category')

    df[i + "_code"] = df[i].cat.codes

print("\n\nList of Categorical features: \n" , df.select_dtypes(include=['object']).columns.tolist())

## (c) Dummy Variable Encoding

print(df['Management or Technical'].unique())

print(df['hard/smart worker'].unique())

df = pd.get_dummies(df, columns=["Management or Technical", "hard/smart worker"], prefix=["A", "B"])

df.head()

df.sort_values(by=['certifications'])

print("List of Numerical features: \n" , df.select_dtypes(include=np.number).columns.tolist())

category_cols = df[['certifications', 'workshops', 'Interested subjects', 'interested career area ', 'Type of company want to settle in?', 'Interested Type of Books']]

for i in category_cols:

  print(i)

Certifi = list(df['certifications'].unique())

print(Certifi)

certi_code = list(df['certifications_code'].unique())
```

```
print(certi_code)
Workshops = list(df['workshops'].unique())
print(Workshops)
Workshops_code = list(df['workshops_code'].unique())
print(Workshops_code)
Certi_l = list(df['certifications'].unique())
certi_code = list(df['certifications_code'].unique())
C = dict(zip(Certi_l,certi_code))
Workshops = list(df['workshops'].unique())
print(Workshops)
Workshops_code = list(df['workshops_code'].unique())
print(Workshops_code)
W = dict(zip(Workshops,Workshops_code))
Interested_subjects = list(df['Interested subjects'].unique())
print(Interested_subjects)
Interested_subjects_code = list(df['Interested subjects_code'].unique())
ISC = dict(zip(Interested_subjects,Interested_subjects_code))
interested_career_area = list(df['interested career area '].unique())
print(interested_career_area)
interested_career_area_code = list(df['interested career area _code'].unique())
ICA = dict(zip(interested_career_area,interested_career_area_code))
Typeofcompany = list(df['Type of company want to settle in?'].unique())
print(Typeofcompany)
Typeofcompany_code = list(df['Type of company want to settle in?_code'].unique())
TOCO = dict(zip(Typeofcompany,Typeofcompany_code))
Interested_Books = list(df['Interested Type of Books'].unique())
print(Interested_subjects)
Interested_Books_code = list(df['Interested Type of Books_code'].unique())
IB = dict(zip(Interested_Books,Interested_Books_code))
Range_dict = {"poor": 0, "medium": 1, "excellent": 2}
print(Range_dict)
A = 'yes'
B = 'No'
col = [A,B]
for i in col:
  if(i=='yes'):
    i = 1
  print(i)
f =[]
```

```
A = 'r programming'
clms = ['r programming',0]
for i in clms:
 for key in C:
  if(i==key):
   i = C[key]
   f.append(i)
print(f)
C = dict(zip(Certifi,certi_code))
print(C)
import numpy as np
array = np.array([1,2,3,4])
array.reshape(-1,1)
def inputlist(Name,Contact_Number,Email_address,
   Logical_quotient_rating, coding_skills_rating, hackathons,
   public_speaking_points, self_learning_capability,
   Extra_courses_did, Taken_inputs_from_seniors_or_elders,
   worked_in_teams_ever,Introvert, reading_and_writing_skills,
   memory_capability_score, smart_or_hard_work, Management_or_Techinical,
   Interested_subjects, Interested_Type_of_Books,certifications, workshops,
   Type_of_company_want_to_settle_in, interested_career_area):
 #1,1,1,1,'Yes','Yes''Yes''Yes''Yes',"poor","poor","Smart worker", "Management","programming","Series","information security"."testing","BPA","testing"
 Afeed = [Logical_quotient_rating, coding_skills_rating, hackathons, public_speaking_points]
 input_list_col =
[self_learning_capability,Extra_courses_did,Taken_inputs_from_seniors_or_elders,worked_in_teams_ever,Introvert,reading_and_writing_skills,memory_capability_score,smart_or_hard_work,Management_or_Techinical,Interested_subjects,Interested_Type_of_Books,certifications,workshops,Type_of_company_want_to_settle_in,interested_career_area]
 feed = []
 K=0
 j=0
 for i in input_list_col:
  if(i=='Yes'):
   j=2
   feed.append(j)
   print("feed 1",i)
  elif(i=="No"):
   j=3
   feed.append(j)
   print("feed 2",j)
```

```
elif(i=='Management'):
  j=1
  k=0
  feed.append(j)
  feed.append(K)
  print("feed 10,11",i,j,k)
elif(i=='Technical'):
  j=0
  k=1
  feed.append(j)
  feed.append(K)
  print("feed 12,13",i,j,k)
elif(i=='Smart worker'):
  j=1
  k=0
  feed.append(j)
  feed.append(K)
  print("feed 14,15",i,j,k)
elif(i=='Hard Worker'):
  j=0
  k=1
  feed.append(j)
  feed.append(K)
  print("feed 16,17",i,j,k)
else:
  for key in Range_dict:
    if(i==key):
      j = Range_dict[key]
      feed.append(j)


      print("feed 3",i,j)


  for key in C:
    if(i==key):
      j = C[key]
      feed.append(j)
      print("feed 4",i,j)
  for key in W:
    if(i==key):
          j = W[key]
          feed.append(j)
          print("feed 5",i,j)
      for key in ISC:
        if(i==key):
          j = ISC[key]
          feed.append(j)
          print("feed 6",i,j)
      for key in ICA:
        if(i==key):
          j = ICA[key]
          feed.append(j)
          print("feed 7",i,j)
      for key in TOCO:
        if(i==key):
          j = TOCO[key]
          feed.append(j)
          print("feed 8",i,j)
      for key in IB:
        if(i==key):
          j = IB[key]
          feed.append(j)
          print("feed 9",i,j)
  t = Afeed+feed
  output = regressor.predict([t])
  return(output)
def main():
  # with st.spinner('Wait for it...'):
  #     time.sleep(5)
  # st.success('Done!')

  html1="""
    <div style="text-align:center; text-shadow: 3px 1px 2px purple;">
      <h1>🧑‍💻 Career Path Prediction app 🧑‍💻</h1>
    </div>
      """
  st.markdown(html1,unsafe_allow_html=True) #simple html
  # Images
  col1, col2, col3 = st.columns(3)
```

```
with col1:
    st.image("./assets/Career _Isometric.png")
with col2:
    st.image("./assets/career.png")
with col3:
    st.image("./assets/Career _Outline.png")
html2="""
  <div style="text-align:center; text-shadow: 3px 1px 2px purple;">
    <h2>Your Friendly Career Advisor<h2>
  </div>
    """
st.markdown(html2,unsafe_allow_html=True) #simple html
st.sidebar.title("Your Information")
Name = st.sidebar.text_input("Full Name")
Contact_Number = st.sidebar.text_input("Contact Number")
Email_address = st.sidebar.text_input("Email address")
if not Name and Email_address:
  st.sidebar.warning("Please fill out your name and EmailID")
if Name and Contact_Number and Email_address:
  st.sidebar.success("Thanks!")
Logical_quotient_rating = st.slider(
  'Rate your Logical quotient Skills', 0,10,1)
st.write(Logical_quotient_rating)
coding_skills_rating = st.slider(
  'Rate your Coding Skills', 0,10,1)
st.write(coding_skills_rating)
hackathons = st.slider(
  'Enter number of Hackathons participated',0,10,1)
st.write(hackathons)
public_speaking_points = st.slider(
  'Rate Your Public Speaking', 0,10,1)
st.write(public_speaking_points)
self_learning_capability = st.selectbox(
  'Self Learning Capability',
  ('Yes', 'No')
  )
# st.write('You selected:', self_learning_capability)
Extra_courses_did = st.selectbox(
  'Extra courses',
('Yes', 'No')
)
# st.write('You selected:', Extra_courses_did)
Taken_inputs_from_seniors_or_elders = st.selectbox(
  'Took advice from seniors or elders',
  ('Yes', 'No')
  )
# st.write('You selected:', Taken_inputs_from_seniors_or_elders)
worked_in_teams_ever = st.selectbox(
  'Team Co-ordination Skill',
  ('Yes', 'No')
  )
# st.write('You selected:', worked_in_teams_ever)
Introvert = st.selectbox(
  'Introvert',
  ('Yes', 'No')
  )
# st.write('You selected:', Introvert)
reading_and_writing_skills = st.selectbox(
  'Reading and writing skills',
  ('poor','medium','excellent')
  )
st.write('You selected: **{}**' .format(reading_and_writing_skills))
memory_capability_score = st.selectbox(
  'Memory capability score',
  ('poor','medium','excellent')
  )
st.write('You selected: **{}**' .format(memory_capability_score))

smart_or_hard_work = st.selectbox(
  'Smart or Hard Work',
  ('Smart worker', 'Hard Worker')
  )
st.write('You selected: **{}**' .format(smart_or_hard_work))
Management_or_Techinical = st.selectbox(
  'Management or Techinical',
  ('Management', 'Technical')
  )
st.write('You selected: **{}**' .format(Management_or_Techinical))
```

```
  Interested_subjects = st.selectbox(
    'Interested Subjects',
    ('programming', 'Management', 'data engineering', 'networks', 'Software Engineering', 'cloud computing', 'parallel computing', 'IOT', 'Computer Architecture', 'hacking')
    )
  st.write('You selected: **{}**' .format(Interested_subjects))
  Interested_Type_of_Books = st.selectbox(
    'Interested Books Category',
    ('Series', 'Autobiographies', 'Travel', 'Guide', 'Health', 'Journals', 'Anthology', 'Dictionaries', 'Prayer books', 'Art', 'Encyclopedias', 'Religion-Spirituality', 'Action and Adventure', 'Comics', 'Horror', 'Satire', 'Self help', 'History', 'Cookbooks', 'Math', 'Biographies', 'Drama', 'Diaries', 'Science fiction', 'Poetry', 'Romance', 'Science', 'Trilogy', 'Fantasy', 'Childrens', 'Mystery')
    )
  st.write('You selected: **{}**' .format(Interested_Type_of_Books))
  certifications = st.selectbox(
    'Interested_Type_of_Books',
    ('information security', 'shell programming', 'r programming', 'distro making', 'machine learning', 'full stack', 'hadoop', 'app development', 'python')
    )
  st.write('You selected: **{}**' .format(certifications))
  workshops = st.selectbox(
    'Workshops Attended',
    ('Testing', 'database security', 'game development', 'data science', 'system designing', 'hacking', 'cloud computing', 'web technologies')
    )
  st.write('You selected: **{}**' .format(workshops))

  Type_of_company_want_to_settle_in = st.selectbox(
    'Type of Company You Want to Settle In ',
    ('BPA', 'Cloud Services', 'product development', 'Testing and Maintainance Services', 'SAaS services', 'Web Services', 'Finance', 'Sales and Marketing', 'Product based', 'Service Based')
    )
  st.write('You selected: **{}**' .format(Type_of_company_want_to_settle_in))
  interested_career_area = st.selectbox(
    'Interested Career Area',
    ('testing', 'system developer', 'Business process analyst', 'security', 'developer', 'cloud computing')
    )
  st.write('You selected: **{}**' .format(interested_career_area))
  result=""
   if st.button("Predict"):
    result=inputlist(Name,Contact_Number,Email_address,Logical_quotient_rating, coding_skills_rating, hackathons,
                public_speaking_points, self_learning_capability,Extra_courses_did,
          Taken_inputs_from_seniors_or_elders,worked_in_teams_ever, Introvert,
          reading_and_writing_skills,memory_capability_score, smart_or_hard_work,
          Management_or_Techinical,Interested_subjects, Interested_Type_of_Books,
          certifications, workshops, Type_of_company_want_to_settle_in, interested_career_area)
 # Progress bar
 my_bar = st.progress(0)
 for percent_complete in range(100):
   time.sleep(0.05)
   my_bar.progress(percent_complete + 1)
 # Balloons
 st.balloons()
 #result will be displayed if button is pressed
 st.success("Predicted Career Option : "
       "{}".format(result))
 # Plot
 corr = df[['Logical quotient rating', 'hackathons',
     'coding skills rating', 'public speaking points']].corr()
 f,axes = plt.subplots(1,1,figsize = (10,10))
 sns.heatmap(corr,square=True,annot = True,linewidth = .4,center = 2,ax = axes)
 st.subheader("Here are some nerdy analytics 😁")
 st.text("Correlation Between Numerical Features")
 st.pyplot(f)
 # Expander
 with st.expander("See explanation"):
  st.write("""
    The plot above shows the correlation of the features.
    As we can see, no highly correlated pair is found!
  """)
 create_table()
 add_data(Name,Contact_Number,Email_address,Logical_quotient_rating, coding_skills_rating, hackathons,
      public_speaking_points, self_learning_capability,Extra_courses_did,
      Taken_inputs_from_seniors_or_elders,worked_in_teams_ever, Introvert,
      reading_and_writing_skills,memory_capability_score, smart_or_hard_work,
      Management_or_Techinical,Interested_subjects, Interested_Type_of_Books,
      certifications, workshops, Type_of_company_want_to_settle_in, interested_career_area)
# if choice == "Add Post":
#    st.subheader("Add Your Article")
#    create_table()
#    blog_title = st.text_input('Enter Post Title')
```

```
 #    blog_author = st.text_input("Enter Author Name",max_chars=50)
 #    blog_article = st.text_area("Enter Your Message",height=200)
 #    blog_post_date = st.date_input("Post Date")
 #    if st.button("Add"):
 #      add_data(blog_author,blog_title,blog_article,blog_post_date)
 #      st.success("Post::'{}' Saved".format(blog_title))
 html3="""
   <div style="color:yellow; margin:80px; text-align:center;">
     Developed with ♡ by <a href=https://hrugved06.github.io/Portfolio-Hrugved-Kolhe/> Hrugved Kolhe</a>
   </div>
     """
 st.markdown(html3,unsafe_allow_html=True)
if __name__=='__main__':
   main()
```

7. Reference:

Faith Too-091673, Faculty of Information Technology Strathmore University.

Nairobi, Kenya, A Career Guidance Mobile Application Based on Personality" 2017.

Abisoye O peyemi A., Alabi 1. O., Ganiyu Shefiu O., Abisoye Blessing O, Omokore Josiah

Crystal D'Mello, RiniAranha, Boni Gregory, Varsha Shrivastava, "Online Career

Guidance System", 2016. Saad Ahmed, Suraj Prakash Maurya, Vaibhav Khalane, "Online Career Guidance System,2018.

Prof. Seema K. Yadav, Mr. Sujit J. Singh, Mr. Akshay M. Bora, Mr. Shashankit N. Thakur, "Educational Recommendation and Tracking System", 2016.

Vishal Bende, Prachi Erande, Priyanka Jore, Ankita Bhor, Ms.Dumbre T.M, "ONLINE CAREER GUIDANCE SYSTEM" 2016.

G Supriyanto, I Widiaty, A G Abdullahand YR Yustiana, "Application expert system career guidance for students", 2019.

Muhammad Dawood, Amna Arshad, "Career counseling for better future studies. using machine learning", 2018.