PLACEMENT BOT

MINI PROJECT REPORT

Submitted by

PRISCILLA RACHEL G 210701196

RACHEL SHERIN J 210701200

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY CHENNAI 600 025

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**Placement Bot**" is the bonafide work of **"Priscilla Rachel G (210701196), Rachel Sherin J (210701200)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Rakesh Kumar M , M.E.,
Assistant Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai – 602015

Submitted to Mini Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

This project introduces a Placement Prediction Bot aimed at improving the efficiency and accuracy of job placement predictions for university students. Utilizing advanced machine learning techniques, the bot analyzes a wide array of student data, including academic records, internship histories, skill sets, and extracurricular involvement, to predict employment outcomes. The predictive model is built using ensemble learning methods, combining decision trees, support vector machines, and neural networks to enhance accuracy and robustness. An intuitive user interface allows students to input their data and receive detailed predictions and recommendations for improving their employability. The system's effectiveness is validated through cross-validation with historical placement data, demonstrating significant predictive power and reliability. The Placement Prediction Bot not only serves as a strategic tool for students to optimize their job search efforts but also aids career counselors in providing targeted advice. This technology represents a significant advancement in educational data science, offering actionable insights to bridge the gap between education and employment.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide,

Dr.RAKESH KUMAR M , M.E., Assistant Professor, Department of Computer Science Engineering. Rajalakshmi Engineering College for his encouragement and guiding us throughout the project to build the project.

Rachel Sherin J-210701200
Priscilla Rachel G-210701196

TABLE OF CONTENTS

CHAPTER

INTRODUCTION

A placement bot is an automated system designed to streamline and enhance the recruitment and placement process for educational institutions and job seekers. Leveraging artificial intelligence and machine learning, the bot efficiently matches candidates with suitable job opportunities based on their skills, qualifications, and preferences. It can handle various tasks such as resume screening, initial interviews, and scheduling further discussions, significantly reducing the workload of HR professionals. By providing real-time updates, personalized job recommendations, and actionable insights, placement bots improve the overall candidate experience and ensure that employers find the best talent quickly and accurately. These bots are particularly beneficial in campus recruitment scenarios, where they can manage large volumes of applications, ensuring a smoother and more effective placement process for students and companies alike. Placement bots automate recruitment by screening resumes, conducting initial interviews, and managing schedules. They enhance efficiency, reduce recruiter workload, and improve candidate matching. Ideal for campus recruitment, they provide data analytics, enhance candidate experience, and integrate with HR systems to streamline hiring and improve overall recruitment outcomes. Placement bots revolutionize the recruitment process by automating tasks like resume screening, initial interviews, and scheduling. Using AI and machine learning, they match candidates to job openings based on skills and qualifications, reducing recruiter workload and improving efficiency. In campus recruitment, they manage large volumes of applications, organize job fairs, and provide personalized job recommendations. These bots also enhance the candidate experience with real-time updates and feedback, ensuring engagement and transparency. By integrating seamlessly with HR systems and providing valuable data analytics, placement bots streamline hiring processes, reduce bias, and improve overall recruitment outcomes for both employers and job seekers.Placement bots streamline recruitment by automating resume screening, initial interviews, and scheduling. They enhance efficiency, reduce recruiter workload, and improve candidate-job matching using AI. Ideal for campus recruitment, they handle large volumes of applications and provide valuable insights, ensuring a better candidate experience and more effective hiring processes.

1.1 OBJECTIVE

The primary objective of this project is to develop and implement a Placement Prediction Bot that accurately forecasts job placement outcomes for university students. This encompasses several specific goals. First, we aim to collect and integrate diverse data points, including academic records, extracurricular activities, internship experiences, and personal skills, and perform comprehensive data preprocessing to ensure quality and consistency. Second, we plan to develop a robust ensemble of machine learning algorithms, such as decision trees, support vector machines, and neural networks, tailored to predict job placement probabilities. These algorithms will be optimized through training on historical placement data to enhance accuracy and reliability. Third, we will create an intuitive and interactive user interface that allows students to input their profiles and receive real-time placement predictions, ensuring the interface is user friendly and provides clear, actionable feedback. Fourth, we intend to conduct rigorous testing and cross-validation of the predictive models against historical data to assess performance and iteratively refine the models based on validation results.

1.2 EXISTING SYSTEM

In the current landscape of career services within higher education institutions, several systems and methods are employed to assist students with job placement. These systems primarily rely on a combination of manual efforts by career counselors, career fairs, online job portals, and alumni networks. Career counseling services offer personalized guidance through one-on-one sessions, workshops, and resume reviews, helping students prepare for the job market. Additionally, career fairs and networking events provide platforms for students to connect with potential employers directly.

Online job portals and career management systems, such as Handshake, LinkedIn, and university-specific platforms, serve as digital intermediaries where students can search for job opportunities, submit applications, and access career resources. These platforms often include features like job matching algorithms, which suggest job listings based on user profiles and preferences. However, these suggestions are typically based on basic filtering and keyword matching rather than sophisticated predictive analytics.

1.4 PROPOSED SYSTEM

The proposed Placement Prediction Bot aims to revolutionize the job placement process for university students by leveraging advanced machine learning techniques and integrating diverse data sources. This system is structured into three main components: Data Collection and Integration, Machine Learning and Prediction, and User Interfaces and Experience.

Data Collection and Integration:

The system gathers data from multiple sources, including academic records (GPA, coursework, academic achievements), extracurricular activities (club participation, sports), internship experiences (roles, responsibilities, performance evaluations), personal skills (soft skills, technical skills, certifications), and job application history (previous applications, interviews, job offers). To ensure the data's quality and consistency, preprocessing steps such as data cleaning, normalization, and feature extraction are employed**.** These steps are crucial for preparing the data for accurate predictive modeling.

Machine Learning and Prediction:

At the core of the system are advanced machine learning algorithms designed to predict job placement outcomes. The bot utilizes ensemble learning models, combining decision trees, support vector machines, and neural networks to enhance predictive accuracy and robustness. These models are trained on historical placement data and validated through rigorous cross-validation techniques to ensure their reliability. The prediction engine generates personalized placement probabilities for students based on the analyzed data and provides tailored recommendations for improving employability, such as suggested courses, skill development opportunities, and networking events.

User Interfaces and Experience:

The system features intuitive user interfaces tailored for both students and career counselors. The student portal allows users to input and update their profiles, view their placement predictions, and receive personalized recommendations. This empowers students with actionable insights to enhance their career prospects. The counselor dashboard enables career counselors to monitor student progress, access predictive insights, and provide more effective, data-driven guidance. Additionally, the system includes robust analytics and reporting features that track key performance metrics such as prediction accuracy, student engagement, and job placement rates. This feedback loop allows for continuous improvement of the system, incorporating new data and user feedback to refine the models and recommendations.

By achieving these objectives, the Placement Prediction Bot aims to empower students to proactively manage their career trajectories, enhance the guidance provided by career counselors, and contribute to more strategic resource allocation within educational institutions. This innovative approach ultimately bridges the gap between education and employment, fostering better outcomes for students and institutions.

CHAPTER 2

LITERATURE SURVEY

Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. International Journal of Computer Science, 1(2), This project emphasizes the importance of data preprocessing in improving the performance of machine learning algorithms. It discusses various techniques such as data cleaning, normalization, and feature extraction, which are essential for preparing raw data for analysis. These processes help in handling issues like missing values, noise, and inconsistencies in the dataset, ensuring that the data fed into the predictive models is of high quality.Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), This comprehensive review explores the field of educational data mining (EDM), focusing on the integration of diverse educational data sources. The authors highlight the challenges and strategies for collecting and preprocessing data from academic records, extracurricular activities, and personal skills to build robust predictive models. The study underscores the necessity of integrating multiple data points to provide a holistic view of student performance and potential.Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. Proceedings of the 23rd International Conference on Machine Learning This project provides an empirical comparison of several supervised learning algorithms, including decision trees, support vector machines, and neural networks. The findings indicate that ensemble learning models, which combine multiple algorithms, often outperform individual models in terms of prediction accuracy. This insight is crucial for developing robust models for the Placement Prediction Bot.Dietterich, T. G. (2000). Ensemble methods in machine learning. International Workshop on Multiple Classifier Systems, Dietterich's work discusses the theory and practice of ensemble learning methods. These methods improve the robustness and accuracy of predictive models by combining the predictions of multiple classifiers. Techniques such as bagging, boosting, and stacking are explored, demonstrating their effectiveness in reducing prediction errors and enhancing model performance. This approach is particularly relevant for developing accurate job placement predictions.LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, This seminal paper provides an overview of deep learning and its applications. The authors discuss how neural networks, particularly deep learning models, can handle complex patterns in large datasets. The ability of these models to capture intricate relationships in data makes them highly effective for predictive tasks, including job placement predictions. The study's insights into neural network architectures and training techniques are invaluable for developing sophisticated predictive

models.Garrett, J. J. (2010). The Elements of User Experience: User-Centered Design for the Web and Beyond. New Riders.Garrett's book outlines the principles of user-centered design, emphasizing the need for interfaces that are functional, accessible, and intuitive. The elements of user experience discussed in this work, such as usability, information architecture, and interaction design, are critical for developing interfaces that enable users to effectively interact with predictive systems. By focusing on the needs and preferences of users, the Placement Prediction Bot can provide an engaging and user-friendly experience, enhancing its over all effectiveness.Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques. Elsevier.This comprehensive text delves into various data mining concepts and techniques, including data preprocessing. It covers essential steps such as data cleaning, transformation, reduction, and discretization, which are crucial for preparing raw data for machine learning algorithms. These preprocessing steps help improve the quality and utility of the data, ensuring more accurate predictive modeling.Kaur, P., Kaur, H., & Singh, G. (2015). Mining educational data to predict student's academic performance using ensemble methods. International Journal of Computer Applications, 107(12), 14-19.This study demonstrates the application of educational data mining (EDM) techniques to integrate data from various sources and predict student performance. By using ensemble methods, the authors highlight how combining multiple models can enhance prediction accuracy. The integration of diverse data points, such as academic records, extracurricular activities, and personal skills, provides a holistic view of student capabilities, which is vital for accurate job placement

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

System design involves the formulation and creation of systems that meet the specific needs of users. Fundamentally, the essence of studying system design lies in comprehending the individual elements and how they interact with each other.

3.2 DEVELOPMENT ENVIRONMENT

3.2.1 HARDWARE SPECIFICATIONS

This document offers a comprehensive overview of the hardware and its implementation, detailing the key components, their interactions, and the necessary requirements for seamless connectivity to utilities and installation.

Table 3.2.1 Hardware Specifications

PROCESSOR	Intel Core i5
RAM	4GB or above (DDR4 RAM)
GPU	Intel Integrated Graphics
HARD DISK	6GB
PROCESSOR FREQUENCY	1.5 GHz or above

3.2.2 SOFTWARE SPECIFICATIONS

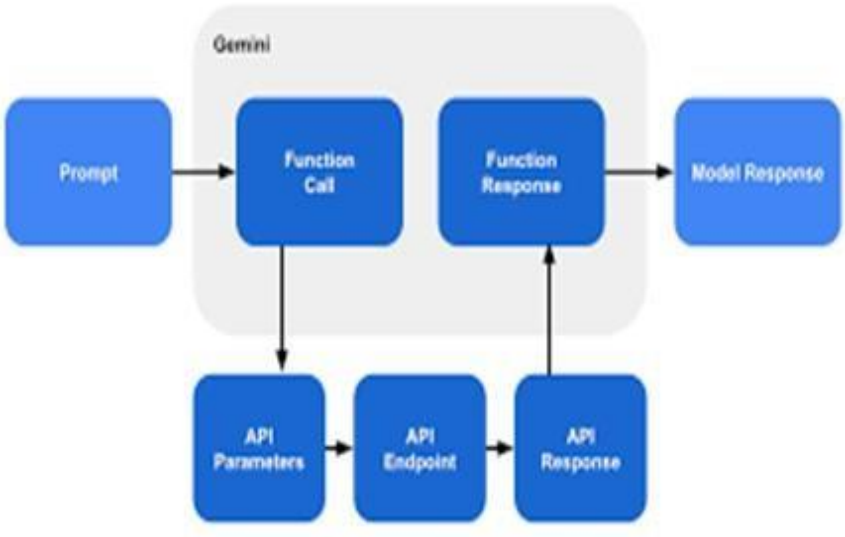
The below table constitutes a thorough evaluation of requirements that precedes the more detailed phases of system design, aiming to minimize the need for subsequent revisions. Furthermore, it should offer a practical foundation for estimating product expenses, potential risks, and project timelines.

Table 3.2.2 Software Specifications

FRONT END	React Js
BACK END	Flask
CODE EDITOR	Visual Studio Code

3.3 SYSTEM DESIGN

3.3.1 ARCHITECTURE DIAGRAM



CHAPTER 4
PROJECT DESCRIPTION

4.1 PROJECT DESCRIPTION:

The Placement Prediction Bot is an innovative tool designed to enhance the job placement process for university students by leveraging advanced machine learning algorithms. This system integrates diverse data sources, including academic records, extracurricular activities, internship experiences, personal skills, and job application history, to provide accurate predictions of job placement outcomes. The bot processes and normalizes this data to ensure high quality and relevance, subsequently using ensemble learning models such as decision trees, support vector machines, and neural networks to generate personalized predictions. Students access the system through a user-friendly portal, where they can input their data and receive tailored recommendations to improve their employability. Career counselors and academic administrators benefit from a comprehensive dashboard that offers data-driven insights, enabling them to better guide students and optimize resource allocation. By delivering precise job placement probabilities and customized advice, the Placement Prediction Bot aims to bridge the gap between education and employment, thereby enhancing career planning and decision-making processes for students and supporting universities in improving overall student outcomes.

The Placement Prediction Bot is an advanced predictive tool designed to assist university students in navigating their career paths by utilizing cutting-edge machine learning techniques. This system aggregates and processes a wide array of data sources, including academic performance records, extracurricular involvement, internship experiences, skill assessments, and job application histories. By employing robust data preprocessing techniques such as cleaning, normalization, and feature extraction, the bot ensures the input data is of the highest quality. It then applies sophisticated machine learning algorithms, particularly ensemble models like decision trees, support vector machines, and neural networks, to predict job placement outcomes with high accuracy. Students interact with the system through an intuitive user interface, which not only displays their personalized job placement probabilities but also provides tailored recommendations to enhance their employability. These recommendations might include suggested courses, skill development opportunities, and networking events. Additionally, career counselors and academic administrators gain access to a comprehensive dashboard, offering detailed insights into student progress and predictive analytics, which aids in more effective career advising and strategic planning. The ultimate goal of the Placement Prediction Bot is to bridge the educational and professional worlds, facilitating better career decisions for students and contributing to the enhancement of university placement programming.

CHATBOT:

This chatbot is built using LangChain and Google Generative AI. The backend is designed using Flask. We will process the text file and convert it into manageable chunks. We will embed it using Google's Generative AI. We will store the embeddings in the FAISS vector database. We will handle the queries of the user by generating responses using conversational AI model. We have to secure the API key so we are using dotenv to load the environment variables.

The function ‘get_text_from_files’ is used to read the text from the file. Next we use the Recursive Character Text Splitter library from LangChain. The text is then taken from the file and, using the ‘get_text_chunks’ from the Recursive Character Text Splitter, it is split into usable parts. This splitter function breaks down the text in to portions that are equivalent to 10,000 characters only. It has an overlap of 1000 characters when the end of text in the source language is aligned with the beginning or within the first 100 characters of the target language. This is used to ensure continuity from one chunk to another so that chunks make sense as smaller parts of a whole text. Splitter library from LangChain. The text from the file is processed using the ‘get_text_chunks’ from the Recursive Character Text Splitter. This splitter function divides the text into chunks which contain 10,000 characters.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

server.py

```
from flask import Flask, request, jsonify
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain_google_genai import GoogleGenerativeAIEmbeddings
import google.generativeai as genai
from langchain_community.vectorstores import FAISS
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain.chains.question_answering import load_qa_chain
from langchain.prompts import PromptTemplate
from dotenv import load_dotenv
import os

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



def get_text_from_files(txt_files):
    text = ""
    for txt_file in txt_files:
        with open(txt_file, 'r', encoding='utf-8') as file:
```

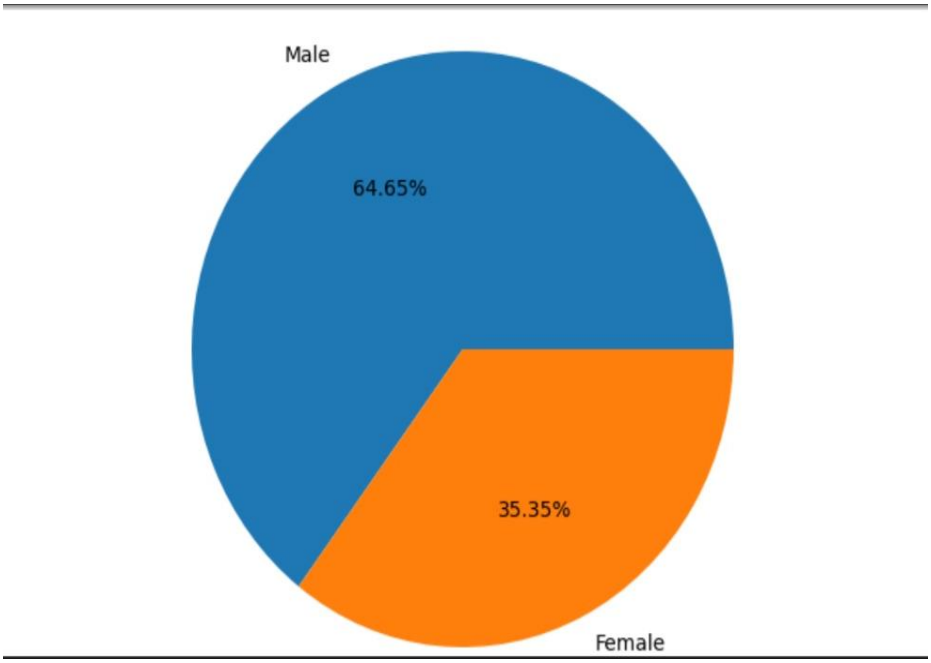
```
            text += file.read()
    return text

def get_text_chunks(text):
    text_splitter = RecursiveCharacterTextSplitter(chunk_size=10000,
chunk_overlap=1000)
    chunks = text_splitter.split_text(text)
    return chunks


def get_vector_store(text_chunks):
    embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-
001")
    vector_store = FAISS.from_texts(text_chunks, embedding=embeddings)
    vector_store.save_local("faiss_index")


def get_conversational_chain():
    prompt_template = """
    Answer the question as detailed as possible from the provided context, make sure
to provide all the details, if the answer is not in
    provided context just say, "answer is not available in the context", don't provide the
wrong answer\n\n
    Context:\n {context}?\n
    Question: \n{question}\n

    Answer:
    """

    model = ChatGoogleGenerativeAI(model="gemini-pro",
                                   temperature=0.3)
```

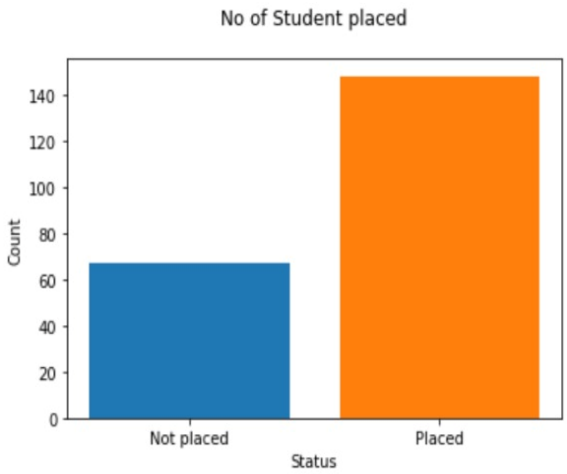
```
    prompt = PromptTemplate(template = prompt_template, input_variables = ["context", "question"])
    chain = load_qa_chain(model, chain_type="stuff", prompt=prompt)

    return chain



def user_input(user_question):
    embeddings = GoogleGenerativeAIEmbeddings(model = "models/embedding-001")

    new_db = FAISS.load_local("faiss_index", embeddings,allow_dangerous_deserialization=True)
    docs = new_db.similarity_search(user_question)

    chain = get_conversational_chain()



    response = chain(
        {"input_documents":docs, "question": user_question}
        , return_only_outputs=True)
    return response["output_text"]



app = Flask(__name__)



@app.route('/api/append',methods=['POST'])
def append_to_file():
    company = request.json.get('company')
    rounds = request.json.get('rounds')
```

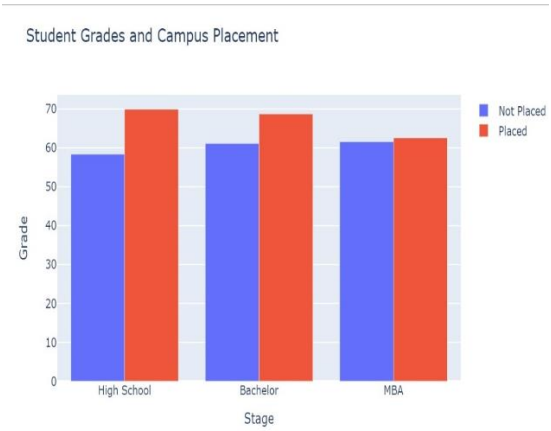
```
    description = request.json.get('description')
    others = request.json.get('others')
    print(company,rounds,description,others)
    if not company or not rounds:
        return jsonify({'error':'Content is required. '}),400
    print(company,rounds)
    file_path = "./data/file.txt"
    try:
        with open(file_path,'a') as file:
            file.write("\n")
            file.write("The company name is "+company+"\n")
            file.write(company+" "+"has "+str(rounds)+" rounds"+"\n")
            file.write("The following points give the desciption of the rounds of the company.\n")
            for i in range(len(description)):
                file.write("In Round "+str(i+1)+" topics from which questions asked: "+description[i]['inputField1']+"\n")
                file.write("In Round "+str(i+1)+" list of questions asked: "+description[i]['inputField2']+"\n")
                file.write("For Round "+str(i+1)+" the preparation strategies are: "+description[i]['inputField3']+"\n")
            file.write(others)
            file.write("\n")
        return jsonify({'message': 'Content appended successfully.'}),200
    except Exception as e:
        return jsonify({'error':str(e)}),500


@app.route('/api/getAns',methods=['POST'])
def getAns():
    directory_path = "./data"
    text_files = [os.path.join(directory_path, file) for file in os.listdir(directory_path) if file.endswith('.txt')]
```

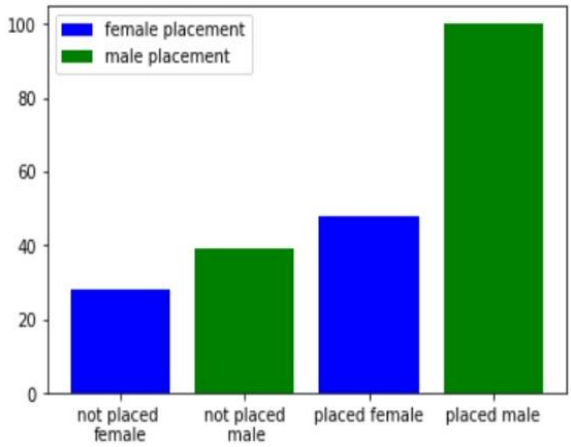
```
    combined_text = get_text_from_files(text_files)
    print("DVFGDFGEGFDGDFSG>>>>>>>>>>>>>>>"+combined_text)
    query = request.json.get("query")
    text_chunks = get_text_chunks(combined_text)
    print(text_chunks)
    get_vector_store(text_chunks)
    if query:
        text=user_input(query)
        print(text)
        return jsonify({'message':text}),200
    print(query)

    return jsonify({'message': 'Content appended successfully.'}),200



if __name___== '__main__':
    app.run(debug=True)
```

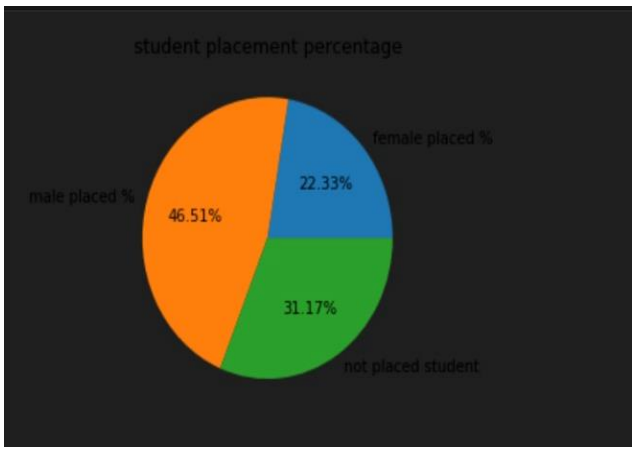
5.2 OUTPUT SCREENSHOTS

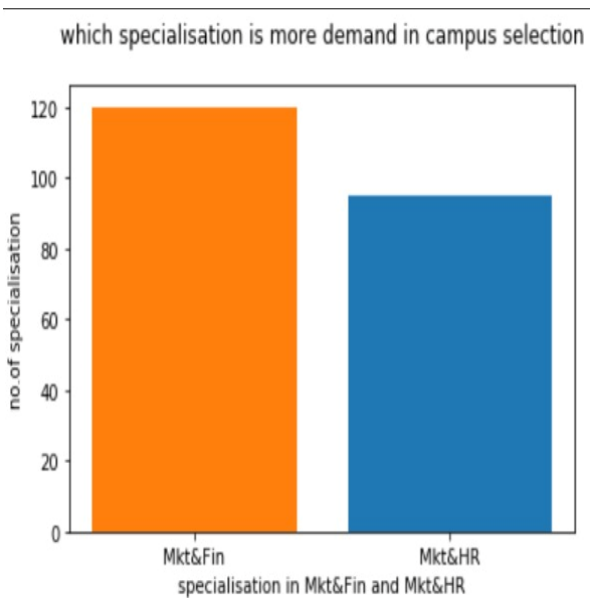


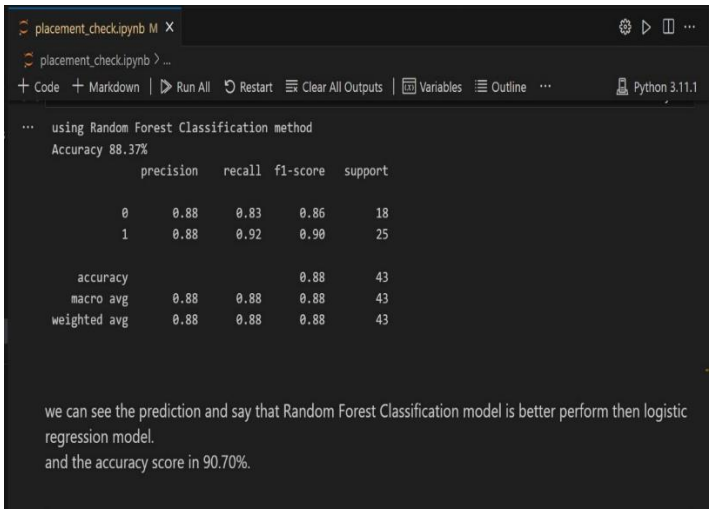

No of Student placed
Count
Not placed
Placed
Status


Student Grades and Campus Placement
Not Placed
Placed
Grade
High School
Bachelor
MBA
Stage


female placement
male placement
not placed female
not placed male
placed female
placed male


student placement percentage
female placed %
22.33%
male placed %
46.51%
31.17%
not placed student


which specialisation is more demand in campus selection
no.of specialisation
Mkt&Fin
Mkt&HR
specialisation in Mkt&Fin and Mkt&HR


placement_check.ipynb M
Code
Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
Python 3.11.1
using Random Forest Classification method
Accuracy 88.37%
precision recall f1-score support
0 0.88 0.83 0.86 18
1 0.88 0.92 0.90 25
accuracy 0.88 43
macro avg 0.88 0.88 0.88 43
weighted avg 0.88 0.88 0.88 43
we can see the prediction and say that Random Forest Classification model is better perform then logistic regression model.
and the accuracy score in 90.70%.

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

The Placement Prediction Bot represents a significant advancement in the realm of career planning and job placement for university students. By harnessing the power of machine learning and integrating diverse data sources, the bot provides highly accurate predictions and personalized recommendations, thereby empowering students to make informed decisions about their career paths. It also equips career counselors and academic administrators with valuable insights, enabling them to offer more effective guidance and optimize resource allocation. Ultimately, the Placement Prediction Bot aims to bridge the gap between education and employment, fostering improved career outcomes for students and enhancing the overall effectiveness of university placement programs.

6.2 FUTURE ENHANCEMENTS

Future enhancements for the Placement Prediction Bot could include the integration of real-time labor market data to provide dynamic and up-to-date job placement predictions, the incorporation of more sophisticated natural language processing (NLP) techniques to analyze students' resumes and cover letters for better skill matching, and the development of advanced recommendation algorithms that factor in industry trends and emerging job roles. Additionally, expanding the bot's capabilities to support global data sets and multilingual interfaces would make it accessible to a broader range of users. Another potential enhancement is the inclusion of feedback loops where students and employers can provide input on the accuracy and utility of the predictions and recommendations, thereby continuously refining and improving the system's performance.

REFERENCES

[1] G. T. P. Lauw, A. Lim, and K. Wang, "Text mining for the integration of information resources in e-learning systems," 2009 IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong, 2009, pp. 684-688, doi: 10.1109/IEEM.2009.5372950.

[2] J. Cross, J. Sheard, and M. A. Hamilton, "The role of online communities in student learning networks," 2012 IEEE Frontiers in Education Conference (FIE), Seattle, WA, 2012, pp. 1-6, doi: 10.1109/FIE.2012.6462410.

[3] S. Braun, A. Schmidt, and M. Walter, "A review on online peer support for students in higher education," 2016 IEEE Global Engineering Education Conference (EDUCON), Abu Dhabi, 2016, pp. 496- 505, doi: 10.1109/EDUCON.2016.7474581.

[4] R. Sohail, T. Ali, and M. A. Shah, "A Chatbot for Facilitating Student Communication in University," 2018 14th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 2018, pp. 1-4, doi: 10.1109/ICET.2018.8603556.

[5] A. Singh, R. Jain, and A. K. Das, "Development of a Campus Recruitment Training System Using Chatbot," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2018, pp. 912-917, doi: 10.1109/IEMCON.2018.8614881.

[6] K. P. S. Bagade and V. H. Khachane, "AIbased Chatbot for Job Application Processing," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 173-178, doi: 10.1109/ICECA49313.2020.9297392.

[7] H. F. Harb and M. M. Abdel-Aal, "Design and implementation of a smart chatbot for e-learning system," 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2020, pp. 1-6, doi: 10.1109/ICCES51560.2020.9334704.

[8] J. S. Wu, C. M. Wu, and C. Y. Chen, "Using AI Chatbot to Assist the Career Counseling for College Students," 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Macao, Macao, 2019, pp. 1242-1246, doi: 10.1109/IEEM44572.2019.8978731.

[9] T. K. R. Nanda, S. K. Kumar, and V. R. Kumar, "An Intelligent Placement Assistance System Using Chatbot," 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing

[10] S. B. Sathish and M. V. Kumar, "A study on artificial intelligence based chatbot for student counseling in universities," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 793-796, doi: 10.1109/CESYS.2018.8624555

[11] A. K. Reddy, P. Y. Reddy, and A. R. P. Kumar, "An intelligent chatbot for academic queries resolution in universities," 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2017, pp. 926-931, doi: 10.1109/ISS1.2017.8388823.

[12] S. Kumar and A. Verma, "Design and development of chatbot using natural language processing for higher education," 2020 International Conference on Inventive Computation Technologies.

[13] N. Sharma, S. Jain, and S. Singh, "A comparative study of chatbot technologies in educational domain," 2020 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 261-266, doi: 10.1109/ICIRCA49075.2020.9209937

[14] S. K. Raj and A. Kumar, "Development of an AI-based chatbot for educational counseling and support," 2019 IEEE Calcutta Conference (CALCON), Kolkata, India, 2019, pp. 1-5, doi: 10.1109/CALCON47608.2019.9051204

[15] R. Singh, P. Sharma, and P. Sinha, "Implementation of AI-based chatbot for educational institute," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2021, pp. 1131- 1136, doi: 10.1109/ICICCS50460.2021.9443092.

[16] S. K. Saravanan and P. Sriram, "Smart chatbot for educational institution using natural language processing," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225502.

[17] R. Bhattacharya and S. Chattopadhyay, "A comparative study of AI-based chatbots for educational assistance," 2021 International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2021, pp. 399-403, doi: 10.1109/ICECA52998.2021.9451704.

[18] A. K. Sharma and A. S. Thakur, "Development of a chatbot for student engagement in educational institutes," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 274-279, doi: 10.1109/Confluence47617.2020.905443

[19] S. Verma and R. R. Mishra, "An AI-based chatbot for student support services in higher education," 2021 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 382-387.

[20] H. F. Harb and M. M. Abdel-Aal, "Design and implementation of a smart chatbot for e-learning system," 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 2020, pp. 1-6, doi: 10.1109/ICCES51560.2020.9334704.