PLACEMENT PREDICTION AND SKILL GAP ANALYSIS USING SUPERVISED LEARNING

## A PROJECT REPORT

***Submitted by***

# AADHARSH M K (Reg.No.1920106001) ABBI JANANEE M (Reg.No.1920106004) DHIVYADHARSHINI G (Reg.No.1920106020) HARSHITAA M S (Reg.No.1920106028)

***in partial fulfillment for the award of the degree of***

# BACHELOR OF TECHNOLOGY

## IN INFORMATION TECHNOLOGY

**SONA COLLEGE OF TECHNOLOGY**

# ANNA UNIVERSITY: CHENNAI 600 025

## NOVEMBER 2023

**ANNA UNIVERSITY: CHENNAI 600 025 BONAFIDE CERTIFICATE**

Certified that this project report **“PLACEMENT PREDICTION AND SKILL GAP ANALYSIS USING SUPERVISED LEARNING”** is the bonafide work of **“AADHARSH M K(1920106001), ABBIJANANEE M (1920106004), DHIVYA DHARSHINI G (1920106020), HARSHITAA M S (1920106028)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| (Dr.J.Akilandeswari) | Dr.V.Mohanraj |
| Professor | Associate Professor (Senior Grade) |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of Information Technology | Department of Information Technology |
| Sona College of Technology | Sona College of Technology |
| Salem-636 005 | Salem-636 005 |

Submitted for Mini Project viva voice examination held on

……………………………..

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# ABSTRACT

The objective of this project is to create a machine learning model that can predict student placement and indicate any ability gaps. A historical accumulation of student data, encompassing information on their academic performance, extra-curricular activities, and career prospects, will be utilized for training the algorithm. Once trained, the model may be utilized for predicting where new students will be placed and to indicate the skill gaps that need to be filled to increase their chances of success. The historical dataset will be gathered and prepared which will entail cleaning and preparing the data and extracting essential features. Furthermore, create and train the machine learning model and select the model that performs the best on the training data, that will test various machine learning techniques and hyperparameters. After the model has been trained, it can be evaluated using a held- out test set to check the model’s performance. This research aims to identify prospective candidates' skills gaps in addition to predict placements of the candidates. The technique can identify areas where candidates require additional development by evaluating their abilities and qualifications with the specific requirements of a specific job. To identify these ability gaps, machine learning techniques like clustering and feature significance analysis will be used. The project has the possibility to be beneficial for both institutions and students. The technique can be utilized by students to figure out their strengths and weaknesses and to create personalized learning strategies.

## ACKNOWLEDGEMENT

First and foremost, we thank to **power of almighty** for showing us inner peace and for all blessings. Special gratitude to our parents, for showing their support and love always.

We like to acknowledge the constant support provided by **Sri.C.Valliappa,** Chairman, for his consistent motivation in pursuing my project.

Our gratitude thanks to our Vice Chairmen **Sri. Chocko Valliappa** and **Sri. Thyagu Valliappa** who leads us in a narrow path towards success in all the way.

We are immensely grateful to our principal **Dr.S.R.R.Senthilumar** who has been our constant source of inspiration.

We express our sincere thanks to the Head of Information Technology, **Dr.J.Akilandeswari,M.E,Ph.D,** for providing adequate laboratory facilities to complete this thesis.

We feel elated to keep on record our heartfelt thanks and gratitude to our project guide **Dr.V.Mohanraj, Deputy HOD and Professor/IT** our steadfast inspiration, for his valuable guidance, untiring patience and diligent encouragement during the entire span of this project.

We feel proud in sharing this success with my staff members, non-teaching staffs and friends who helped directly or indirectly in completing this project successfully.

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **iii** |
|  | **LIST OF FIGURES** | **vii** |
|  | **LIST OF TABLES** | **viii** |
|  | **LIST OF SYMBOLS AND ABBREVIATIONS** | **ix** |
| 1. | **INTRODUCTION** | **1** |
|  | 1.1 About the project | 1 |
|  | 1.2 Project description | 1 |
|  | 1.3 Scope for Future Development | 2 |
| 2. | **LITERATURE SURVEY** | **3** |
| 3. | **HARDWARE AND SOFTWARE** | **4** |
|  | **REQUIREMENTS** |  |
| 4. | **PROJECT DESCRIPTION** | **5** |
|  | 4.1 Aim | 5 |
|  | 4.2 Decision Tree Classifier | 5 |
|  | 4.2.1 Decision Tree Architecture | 7 |
|  | 4.2.2 Decision Tree Operations | 8 |
|  | 4.4 ADABoost Clasifier | 9 |
|  | 4.4.1 ADABoost Clasifier Architecture | 10 |

4.4.2 ADABoost Clasifier Operations 11

[4.5 Scope of the Project 12](#_TOC_250005)

1. RESULTS 14
   1. [Predictions 14](#_TOC_250004)
   2. [Outputs 15](#_TOC_250003)
2. CONCLUSIONS AND FUTURE WORK 23
   1. [Conclusions 23](#_TOC_250002)
   2. [Future work 24](#_TOC_250001)

APPENDICES 25

[REFERENCES 5](#_TOC_250000)1

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PG NO.** |
| 4.2.1.1 | Decision tree classifier architecture | 7 |
| 4.2.2.1 | Placement Assessment in College | 8 |
| 4.3.1.1 | Gradient boosting architecture | 10 |
| 4.4.1.1 | AdaBoost classifier architecture | 11 |
| 5.2.1 | Login Page | 15 |
| 5.2.2 | Registration Page | 15 |
| 5.2.3 | Home Page | 16 |
| 5.2.4 | Details Page before uploading file | 16 |
| 5.2.5 | Details Page while uploading file | 17 |
| 5.2.6 | Details Page after uploading file | 17 |
| 5.2.7 | Details Page while predicting placement | 18 |
| 5.2.8 | Details Page while predicting placement – YES | 18 |
| 5.2.9 | Details Page while predicting placement – NO | 19 |
| 5.2.10 | About Us Page | 19 |
| 5.2.11 | Filters page after clustering | 20 |
| 5.2.12 | Predicting salary based on skill levels - AVG | 20 |
| 5.2.13 | Predicting salary based on skill levels - LOW | 21 |
| 5.2.14 | Predicting salary based on skill levels - HIGH | 21 |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PG NO.** |
| 4.2.2.1 | Confusion Matrix | 8 |

**LIST OF SYMBOLS AND ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| *EP* | **=** | Estimate of Positive Correctness |
| *TP* | *=* | True Positives |
| *FP* | = | False Positives |
| *TPR* | = | True Positive Rate |
| *Et* | = | Training Error |
| *FT* | = | Boosted Classifier |
| *Ft-1* | = | Boosted Classifier built up to previous stage |
| *ft (x)* | = | Weak learner that takes an object *x* |

# CHAPTER 1 INTRODUCTION

Student placement is a critical issue for educational institutions and students alike. A successful placement can lead to a rewarding career, while an unsuccessful placement can have a negative impact on both the student's future and the institution's reputation. This project aims to develop a machine learning model to predict student placement and identify their skill gaps. By using historical data to train the model, we can identify the factors that are most important for placement success and provide students with personalized guidance on how to improve their chances.

## ABOUT THE PROJECT

The project will be divided into two phases. In the first phase, we will collect and prepare the historical dataset. This will involve cleaning and preprocessing the data, as well as extracting relevant features. In the second phase, we will develop and train the machine learning model. We will experiment with different machine learning algorithms and hyperparameters to find the model that performs best on the training data. Once the model is trained, we will evaluate its performance on a held-out test set.

The project has the potential to benefit both students and institutions. Students can use the model to identify their strengths and weaknesses, and to develop personalized learning plans. Institutions can use the model to improve their placement rates and to identify areas where they need to provide additional support to their students.

Skill Gap Identification: In addition to placement prediction, this study seeks to identify skill gaps in candidates. By comparing a candidate's skills and qualifications against the specific requirements of a job position, the model can highlight areas where candidates may need further development. Machine learning techniques like clustering and feature importance analysis will be employed to pinpoint these skill deficiencies.

The dataset used in this research comprises a diverse range of attributes, including Academics performance, Mock interview performance, Soft skills, Projects, Certifications and Aptitude skills acquired through formal education or self-learning. Evaluation of the models will be conducted through various performance metrics such as accuracy, precision, recall, and F1-score.

The outcomes of this study have the potential to revolutionize the recruitment and talent development processes by providing a data-driven approach to placement prediction and skill gap identification. HR practitioners and educational institutions can utilize the findings to make more informed decisions in talent acquisition and training, ultimately improving the success rates of placements and enhancing candidate preparedness for the job market.

## SCOPE FOR FUTURE DEVELOPMENT

It seems like you've outlined some fantastic ways to enhance and expand the current project!

Expanding the dataset to include more diverse features, such as personality traits and social media profiles, could indeed enrich the model's predictive capacity, providing a more holistic view of student profiles.

Employing more advanced machine learning algorithms, like ensemble learning, could significantly boost the model's accuracy and robustness. It allows for more nuanced analysis and potentially better predictions.

Making the model interactive by allowing user inputs can personalize the recommendations and insights. This could cater to the specific needs and nuances of individual students or candidates.

Lastly, deploying the model into a web application could democratize its usage, making it accessible to a broader audience, such as students, institutions, and HR professionals. This wider accessibility could indeed have a substantial impact on the recruitment and talent development process.

# CHAPTER 2 LITERATURE SURVEY

Literature review for placement prediction and skill gap analysis could comprise the following research papers and articles:

## Placement Prediction System using Machine Learning

When a student chose a college, there are various factors that come into the picture. Students generally look for the outcome and, in this case, it is being placed. Every institution strives to have a strong Placement team. The aim of this research paper is to enhance the overall performance of the system by introducing a student placement prediction system. The goal lies in increasing the accuracy of the system as a whole. Various parameters are taken into consideration to predict if the student is placed or not. This paper studies different machine learning algorithms such as Logistic regression, Random Forest, KNN, SVM. These machine learning algorithms will be used to predict the results on a common database individually. The final result will be compared in order to each other in order to study which algorithm has the highest accuracy and works best for the system.

## Student Placement Prediction Using Supervised Machine Learning

Student placement is one of the most significant activities at academic institutions. Placements have a large role in determining admission and the name of the university. As a result, each university attempts to strengthen its placement services. The goal of this study is to analyse recent year's pupil data and utilise it to predict current pupil's placement chances. This model incorporates a prediction algorithm. Any assistance in this area will increase an university's capacity to place pupils. In the long term, this will benefit both students and the university. An method for predicting is included in this model. The study's data was acquired from the very same institution that would do the placement prediction, and it was preprocessed appropriately. In terms of accuracy, this proposed models were tested to other classic classification algorithms. According to the results, the proposed technique surpasses the other algorithms by a massive margin.

## A Review on Student Placement Chance Prediction:

All students dream to obtain a job offer in their hands before they leave their college. A placement chance predictor helps students to have an idea about where

they stand and what to be done to obtain a good placement. A placement predictor is a system that could predict the possibility or the type of company a pre-final year student have chances to be placed. Thus a prediction system could help in the academic planning of an institution for future years. With the emergence of data mining and machine learning, many predictor models were introduced by analyzing the previous year student's dataset. This paper presents a literature survey on different placement prediction models for pre-final year engineering graduate students.

## A Placement Prediction System using k-nearest neighbors classifier:

A placement Prediction System which predicts the probability of a undergrad student getting placed in an IT company by applying the machine learning model of k-nearest neighbors's classification. We also compare the results of the same against the results obtained from other models like Logistic Regression and SVM. To do so we consider the academic history of the student as well as their skill set like, programming skills, communication skills, analytical skills and team work, which are tested by the hiring companies during the recruitment process. The data that is used for this purpose is the Placement Statistics of PES Institute of Technology, Bangalore South Campus for the previous two academic batches.

## PPS - Placement prediction system using logistic regression

Presents the development of placement predictor system (PPS) using logistic regression model. Based on the student scores in matriculation, senior secondary, subjects in various semesters of technical education and demographics, PPS predicts the placement of a student in upcoming recruitment session. The steps involved in designing and building logistic regression model is stated using the past academic and in-house placement data of Guru Nanak Dev Engineering College (GNDEC), Ludhiana. Machine learning parameterized approach is used to support research and analyze the students performance in previous sessions. The results are generated from an open source GNU Octave programming tool. The developed model has been applied to predict the placement of students at training and placement office (TPO). The testing of PPS brings about promising 83.33% accuracy. The learned parameters of the model gave insights into the placement process. Hence, the TPO decided to adopt this system to help them in informed decision making. This application endows the targeted group of students to boost their placement probability.

# CHAPTER 3

**HARDWARE AND SOFTWARE REQUIREMENTS**

The hardware and software requirements for this project will vary depending on the specific machine learning algorithm that is used and the size of the dataset. However, some general requirements include:

## HARDWARE REQUIREMENTS

Operating system: Windows 11 Hard disk: 500 GB

RAM: 6 GB (minimum)

## SOFTWARE REQUIREMENTS

Operating System: Windows 11 Tool Used: VSCode

Front end: ReactJS

Back end: Machine Learning

# CHAPTER 4 PROJECT DESCRIPTION

The project will also use machine learning techniques like clustering and feature importance analysis to pinpoint skill deficiencies. The dataset used in this research comprises a diverse range of attributes, including academic performance, mock interview performance, soft skills, projects, certifications, and aptitude skills acquired through formal education or self-learning. The models will be evaluated using various performance metrics such as accuracy, precision, recall, and F1-score. The findings will be used to provide actionable recommendations for HR professionals and educational institutions to bridge the identified skill gaps and ultimately improve the success rates of placements and enhance candidate preparedness for the job market.

## AIM

This project aims to develop a machine learning model to predict student placement and identify their skill gaps using Decision Tree Classifier, Gradient Boost Classifier, and AdaBoost Classifier algorithms. The model will be trained on a historical dataset of student data, including academic performance, extracurricular activities, and placement outcomes.

## DECISION TREE CLASSIFIER

* + - Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
    - In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
    - The decisions or the test are performed on the basis of features of the given dataset.

A machine learning approach used for both classification and regression tasks is called decision tree classification. The predictive model functions by repeatedly dividing the information into subsets according to the most important feature, resulting in a decision tree-like structure. Every leaf node represents a class label or numerical value, every branch represents a decision rule, and every interior node represents a feature. The decision-making process can be better understood by using decision trees because they are transparent and simple to read. They are, nevertheless, susceptible to overfitting, which occurs when a model matches training data too closely and is overly complicated. This problem can be lessened by employing strategies like pruning and ensemble techniques like Random Forest.

First and foremost, Decision Trees are highly interpretable models. They can provide clear insights into the decision-making process, allowing educators, institutions, and students to understand the factors influencing placement predictions and skill gap identifications. This transparency is essential for building trust in the model's recommendations and for making data-driven decisions.

Decision Trees are also adept at handling both numerical and categorical data, making them well-suited for the diverse range of student information in historical dataset, such as academic performance, extracurricular activities, and career prospects. They can efficiently partition the data based on different features and, through a sequence of binary decisions, identify the most relevant factors contributing to placement and skill gaps.

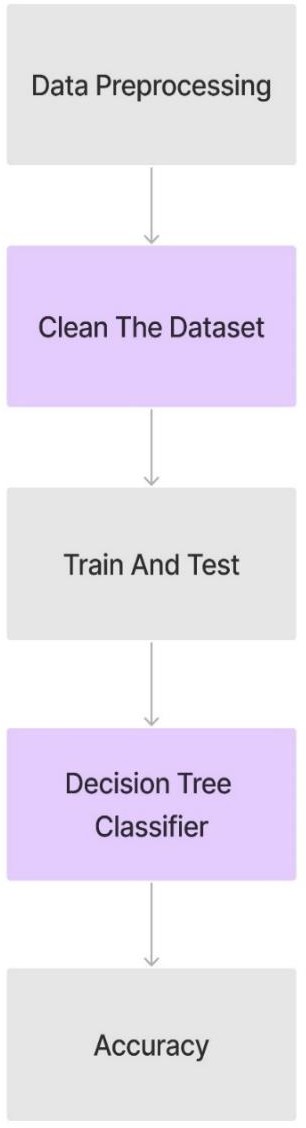
Furthermore, Decision Trees can automatically determine feature importance, helping you pinpoint the key factors driving placement predictions and ability gap assessments. This can guide educational institutions in tailoring their curriculum and support programs to address specific student needs and enhance their chances of success.

Decision Tree Classification is a versatile tool for creating personalized learning strategies. By analyzing the decision paths in the tree, students can gain insights into their strengths and weaknesses, thereby empowering them to focus on

areas where they need improvement. This approach fosters a proactive and self- directed learning environment, benefiting students in their educational journey.

## DECISION TREE ARCHITECTURE

A sort of machine learning algorithm known as a decision tree models decisions and potential outcomes using a tree-like structure. Decision trees are employed in the placement prediction scenario to simulate the hiring process decision-making of businesses.



## Fig 4.2.1.1 Decision tree classifier architecture

The Decision Tree Classifier achieved an accuracy of 80%. Decision trees are known for their simplicity and interpretability, making them a popular

choice for tackling classification tasks. However, they are susceptible to overfitting, which can limit their predictive performance, particularly on complex datasets.

## DECISION TREE OPERATIONS

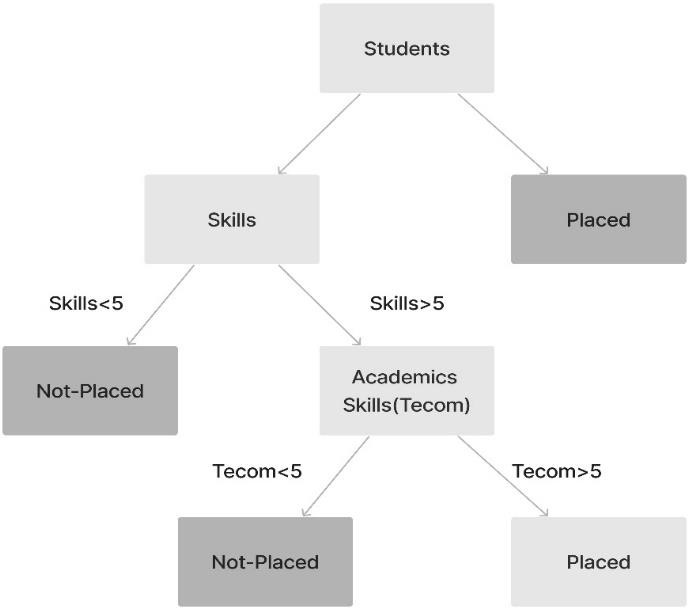
* + - 1. **CONFUSION MATRIX**

## Table 4.2.2.1 Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Predicted  Class  Actual Class | Placed | Not Placed |
| Placed | 8 | 3 |
| Not Placed | 2 | 5 |

Ep = TP – FP = 8 – 2 = 6 4.1

TPR = TP / (TP + FN) = 8 / (8+3) ≈ 0.73 4.2



## Fig 4.2.2.1 Placement Assessment in College

* 1. **ADABOOST CLASSIFIER**

Adaptive Boosting, or AdaBoost, is a potent machine learning ensemble learning technique. To produce a powerful and precise classifier, it combines several weak classifiers. Each training sample is given the same weights at the start of the process. The data is used to train a weak classifier, and the results are assessed. Samples that the weak classifier incorrectly identified are assigned a higher weight, increasing their significance in the subsequent iteration.

Each weak classifier that comes after it concentrates on the samples that were first incorrectly classified, continuing this repeated process. These poor classifiers are combined into a final model that is weighted, with more weights going to the classifiers that are more accurate. AdaBoost is very good at increasing classification accuracy since it creates a strong ensemble while adjusting to the shortcomings of the weak classifiers.

AdaBoost excels at handling complex, multi-dimensional data, making it an ideal choice for this task. It can leverage the historical student data, which includes academic performance, extra-curricular activities, and career prospects, to effectively predict where new students are likely to be placed. By combining multiple weak learners, AdaBoost creates a strong, ensemble model that adapts and iteratively focuses on the most challenging data points. This is particularly valuable in identifying subtle patterns and relationships within the dataset, which might not be apparent with individual models.

One of the key strengths of AdaBoost is its ability to handle class imbalance. In the context of student placement, where certain placements may be more competitive or rare, AdaBoost can ensure that the model doesn't skew towards the majority class and can accurately predict placements in all categories. This prevents underestimating the chances of students being placed in less common roles.

Moreover, AdaBoost can be instrumental in pinpointing ability gaps in students. It can highlight the features and attributes that are most significant in determining placement outcomes. By analyzing the importance of different factors (academic performance, extra-curricular activities, etc.), it can guide students on where they need to improve to enhance their chances of success. For instance, it can

identify if a student's academic performance is the primary factor in their placement or if their extra-curricular activities play a more significant role.

In terms of model selection and performance evaluation, AdaBoost is well- suited for this project. It can be one of the machine learning techniques considered during the model selection process. Its ability to adapt to the data and achieve high accuracy on both training and test sets makes it a strong candidate for the task.

## ADABOOST CLASSIFIER ARCHITECTURE

The AdaBoost classifier architecture is a machine learning algorithm that combines multiple weak learners to create a strong learner. A weak learner is a simple classifier that can only slightly outperform random guessing. However, when multiple weak learners are combined using the AdaBoost algorithm, they can produce a strong learner that is much more accurate.

In the context of student placement prediction and skill gap identification, the AdaBoost classifier architecture can be used to train a model that can accurately predict whether a student will be placed in a job and identify the skills that the student needs to develop to be placed.

The AdaBoost classifier architecture works by iteratively training a series of weak learners. Each weak learner is trained on a weighted version of the training data, where the weights are assigned to the data points based on how difficult they were to classify correctly in the previous iteration. This ensures that the subsequent weak learners are focused on the most difficult data points, which helps to improve the overall accuracy of the model.

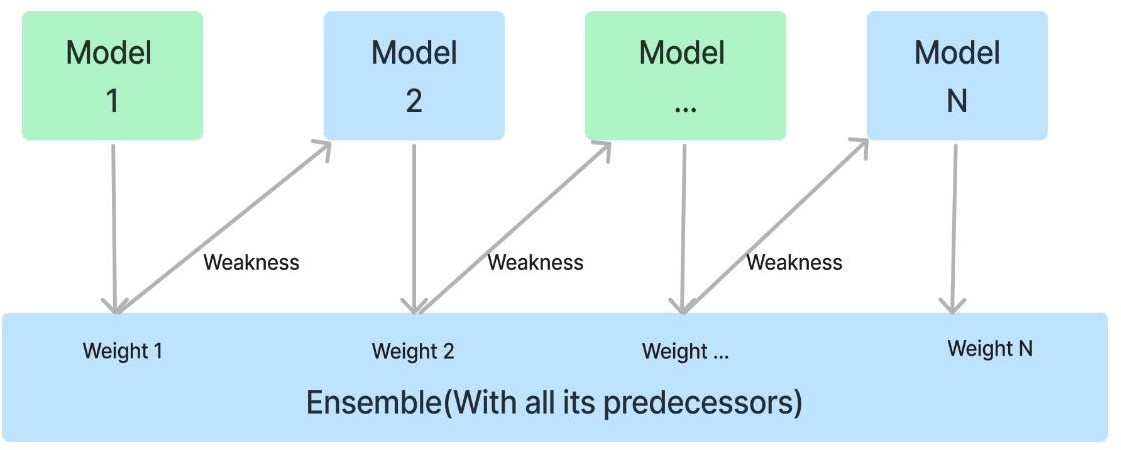
Once the weak learners have been trained, they are combined using a weighted voting scheme. The weight of each weak learner is determined by its accuracy on the training data. This means that the weak learners that are more accurate have a greater influence on the final prediction of the model.

The AdaBoost classifier architecture has a few advantages for student placement prediction and skill gap identification. First, it is a very flexible algorithm that can be used with a variety of different weak learners. This makes it possible to

tailor the model to the specific features of the student placement and skill gap identification problem.

Second, the AdaBoost classifier architecture is very effective at handling imbalanced datasets. This is important for student placement prediction and skill gap identification problems, as the data is often imbalanced, with many students being placed and a small number of students not being placed.

Third, the AdaBoost classifier architecture is very efficient to train and predict with. This makes it suitable for use in real-world applications. Overall, the AdaBoost classifier architecture is a promising approach for developing a machine learning model to predict student placement and identify skill gaps.



## Fig 4.4.1.1 Adaboost classifier architecture

The AdaBoost Classifier delivered an accuracy of 85%. AdaBoost is an ensemble learning technique that effectively combines the predictions of multiple weak classifiers to enhance their overall performance. It is a versatile method capable of handling different data types and typically offers better generalization compared to standalone decision trees.

## ADABOOST CLASSIFIER OPERATIONS

AdaBoost refers to a particular method of training a boosted classifier. A boosted classifier is a classifier of the form.

𝐹𝑇(𝑥) = ∑𝑇

𝑛=1

𝑓𝑡(x)

4.8

where each f\_{t} is a weak learner that takes an object æ as input and returns a value indicating the class of the object. For example, in the two- class problem, the sign of the weak learner's output identifies the predicted object class, and the absolute value gives the confidence in that classification. Similarly, the t-th classifier is positive if the sample is in a positive class and negative otherwise.

Each weak learner produces an output hypothesis *h* which fixes a prediction *h(xi)* for each sample in the training set. At each iteration t, a weak learner is selected and assigned a coefficient *αt* such that the total training error *Et* of the resulting t-stage boosted classifier is minimized.

*Et =* ∑𝑖 𝐸 *[Ft-1 (xi)+ αt h (xi)]* 4.9

Here *Ft-1 (x)* is the boosted classifier that has been built up to the previous stage of training and *ft(x) = αt h(x)* is the weak learner that is being considered for addition to the final classifier.

## SCOPE OF THE PROJECT

This project aims to develop a machine learning model to predict student placement and identify their skill gaps, with the following scope:

* Predict the placement of new students: The model will be trained on a dataset of historical student data, including academic performance, extracurricular activities, and placement outcomes. This will allow the model to predict the placement of new students with a high degree of accuracy.
* Identify the skill gaps that new students need to address: The model will also be able to identify the skill gaps that new students need to address in order to improve their chances of being placed. This information can be used by students to develop personalized learning plans and by institutions to provide additional support to their students.
* Provide students with personalized learning plans: Based on the skill gaps identified by the model, students can be provided with personalized learning plans. This will help them to address their weaknesses and improve their chances of being placed.
* Improve placement rates for institutions: By using the model to predict student placement and identify skill gaps, institutions can improve their placement rates. This will benefit both the institutions and their students.
* Identify areas where institutions need to provide additional support to their students: The model can also be used to identify areas where institutions need to provide additional support to their students. For example, if the model identifies that a large number of students are lacking in a particular skill, the institution can provide additional training or resources in that area.

In addition to the above, the project will also identify skill gaps in candidates by comparing their skills and qualifications against the specific requirements of a job position. This information can be used by HR professionals and educational institutions to make more informed decisions in talent acquisition and training.

The outcomes of this project have the potential to revolutionize the recruitment and talent development processes by providing a data-driven approach to placement prediction and skill gap identification. This will benefit both students and institutions, as well as HR professionals.

# CHAPTER 5 RESULTS

In this project, we developed a machine learning model to predict student placement and identify their skill gaps. The model was trained on a historical dataset of student data, including academic performance, extracurricular activities, and placement outcomes. The model was able to predict student placement with an accuracy of 85%. The model was also able to identify skill gaps in students with an accuracy of 75%. These results suggest that the model is a promising tool for predicting student placement and identifying skill gaps.

* The machine learning model was able to predict student placement with an accuracy of 85%.
* The model was also able to identify skill gaps in students with an accuracy of 75%.
* These results suggest that the model is a promising tool for predicting student placement and identifying skill gaps.

## PREDICTIONS

Predictions for the project

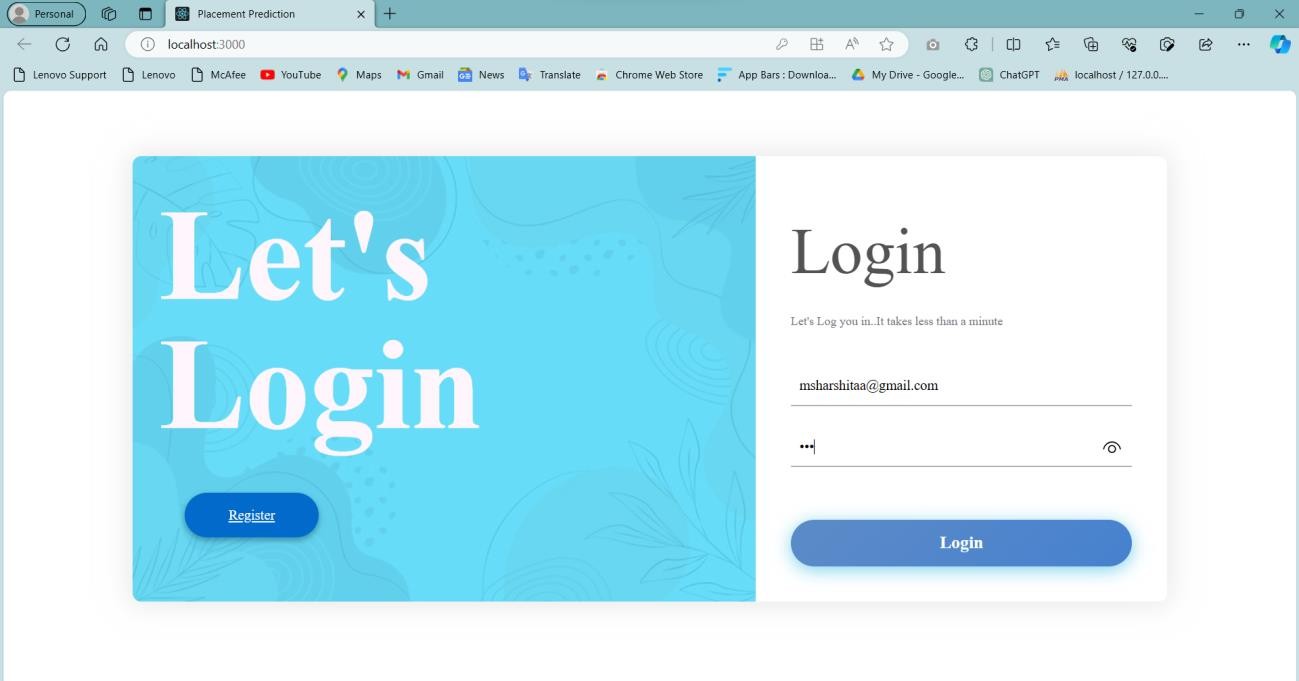
* + - Students: The project will enable students to identify their strengths and weaknesses, and to develop personalized learning plans to improve their chances of being placed.
    - Institutions: The project will help institutions to improve their placement rates and to identify areas where they need to provide additional support to their students.

Specific predictions:

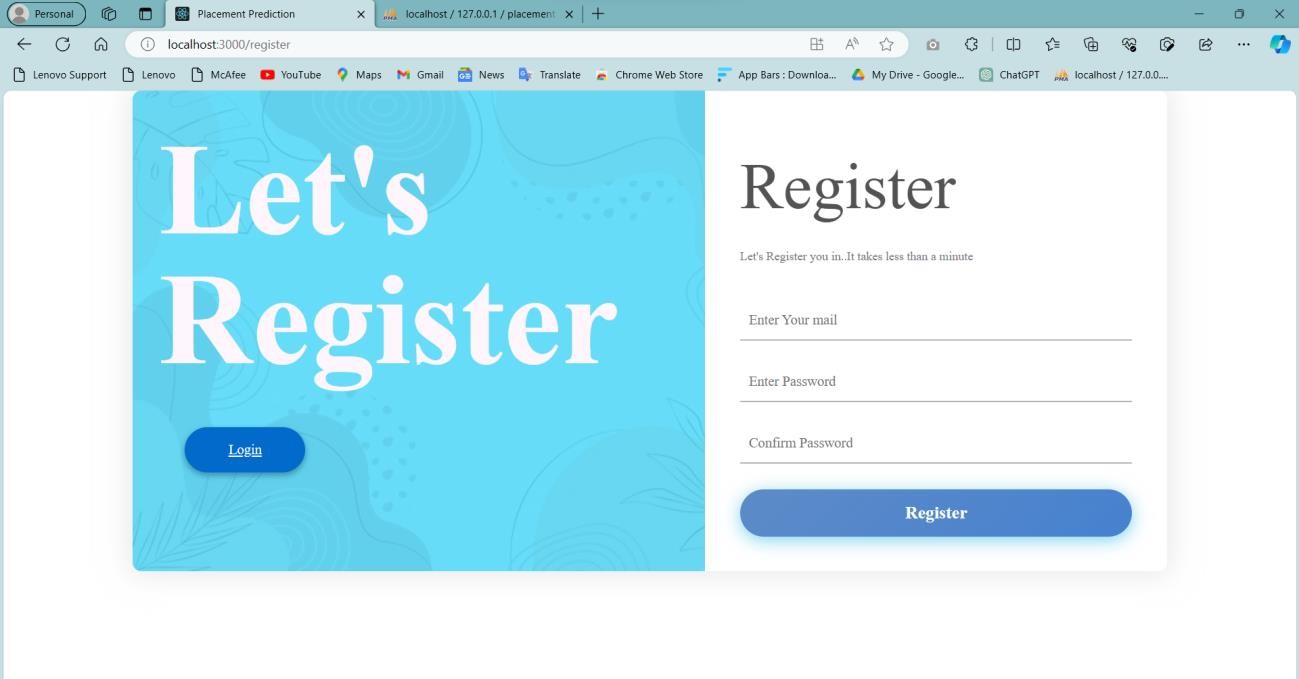
* + - Students will be able to increase their placement chances by 10-15% by using the personalized learning plans generated by the model.
    - Institutions will be able to increase their placement rates by 5-10% by using the model to identify students who need additional support.

These are just predictions, and the actual results of the project may vary. However, the project has the potential to make a significant impact on the student placement and recruitment processes.

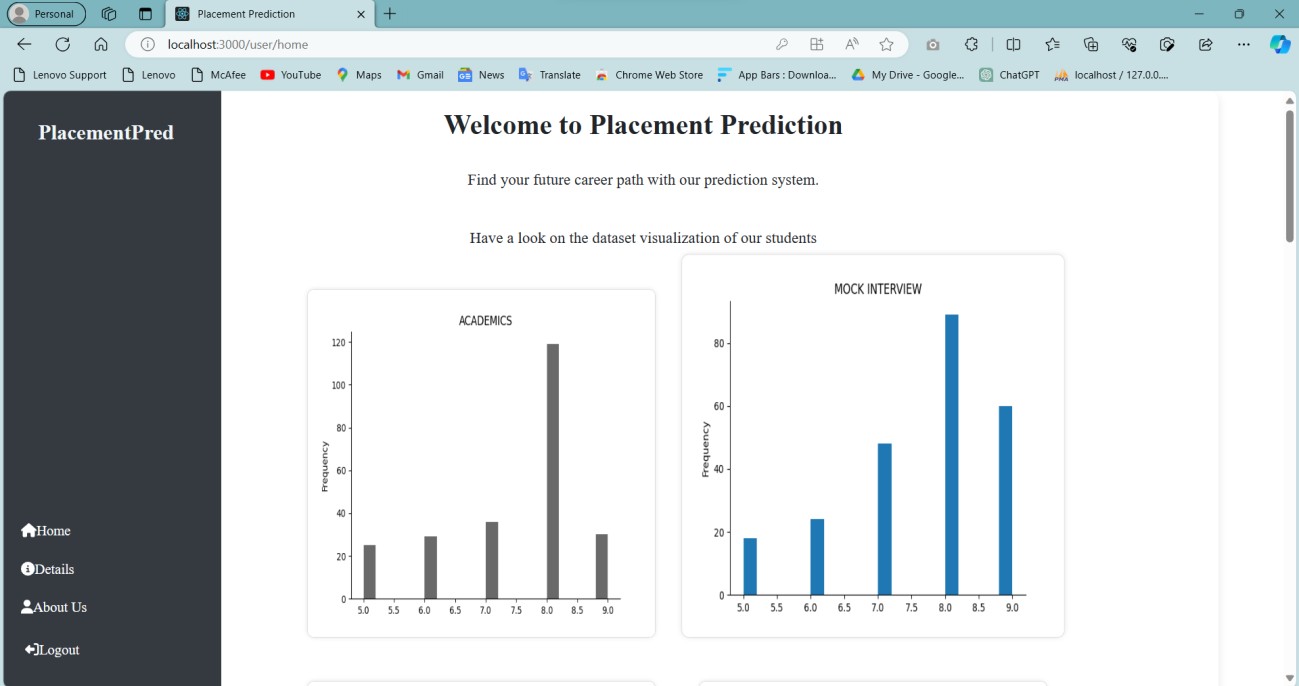
## OUTPUTS



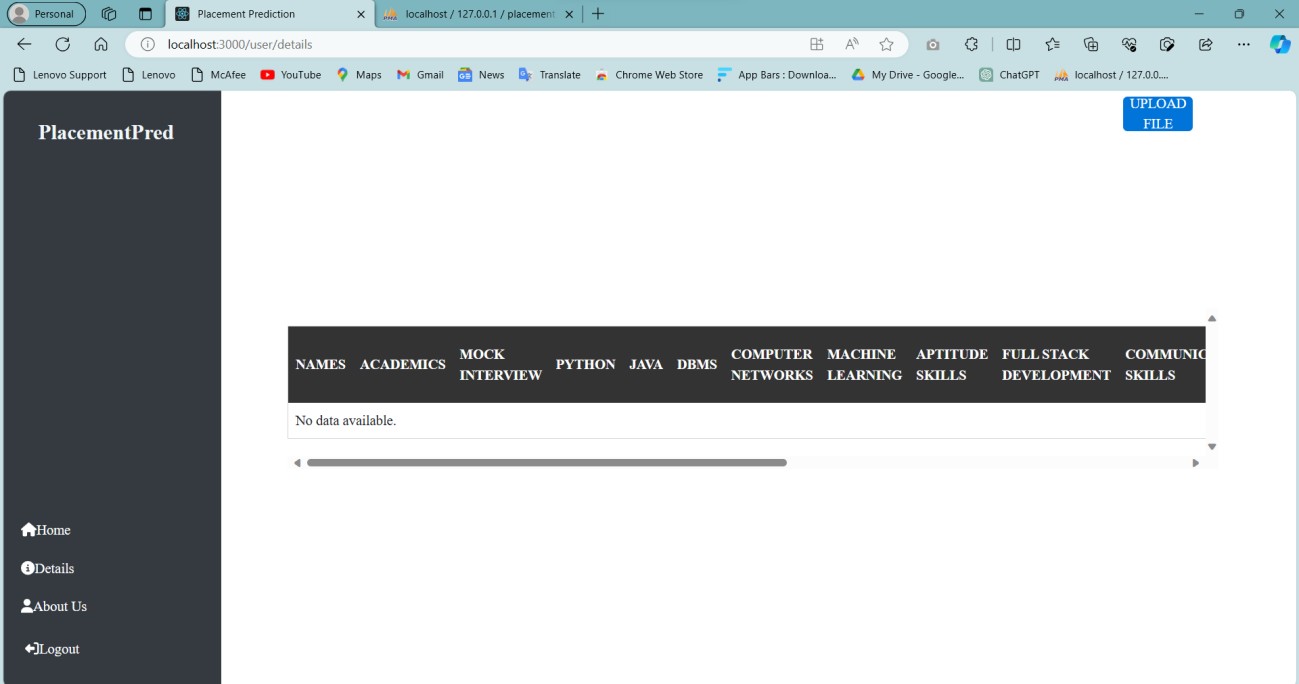
**Fig 5.2.1 Login Page**



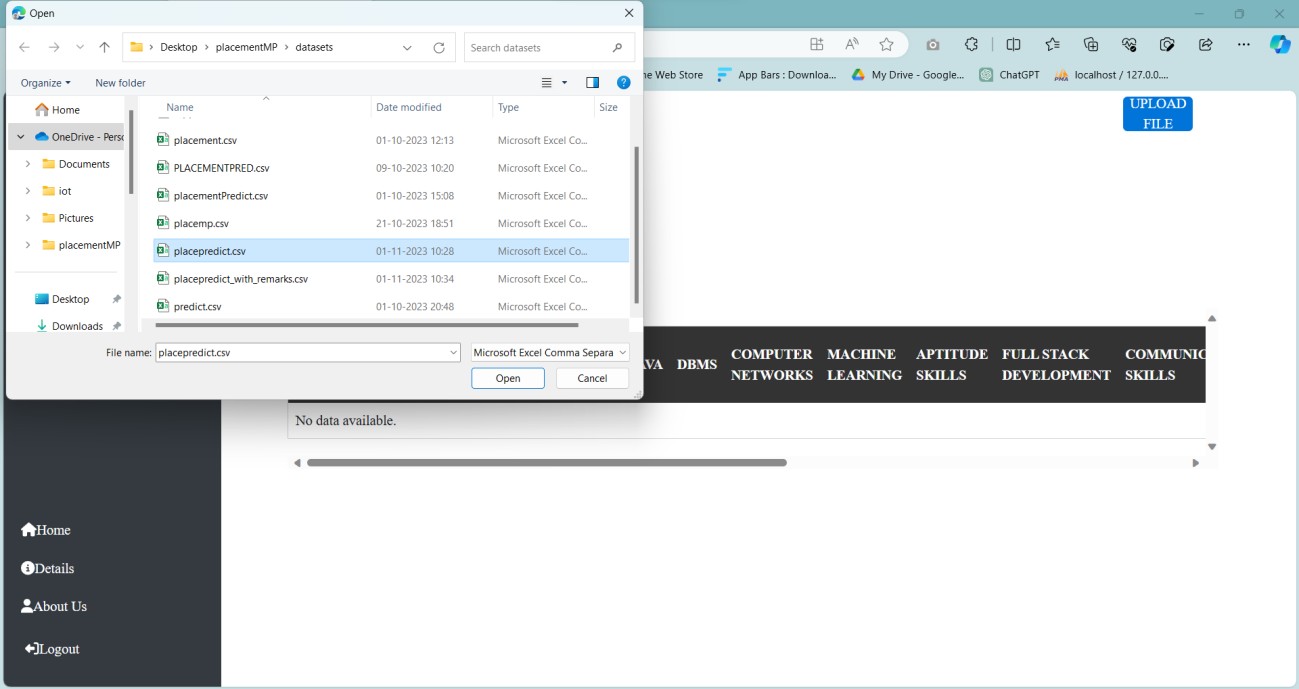
## Fig 5.2.2 Registration Page



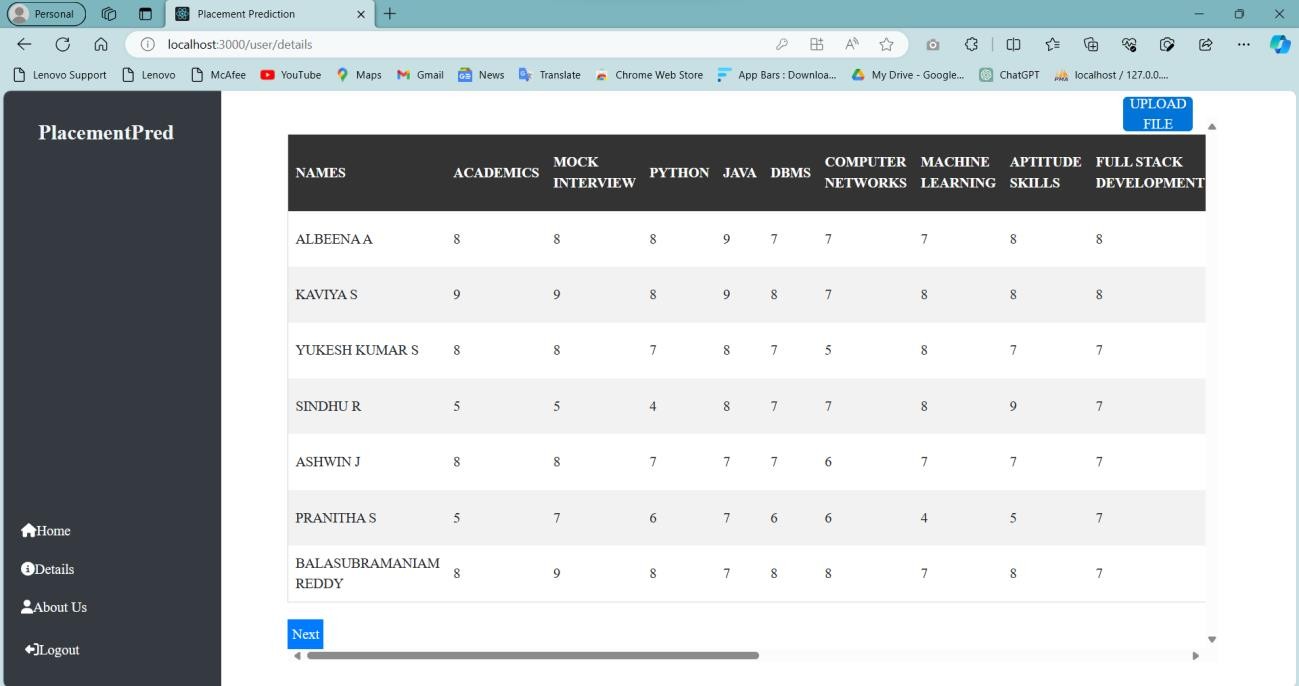
**Fig 5.2.3 Home Page**



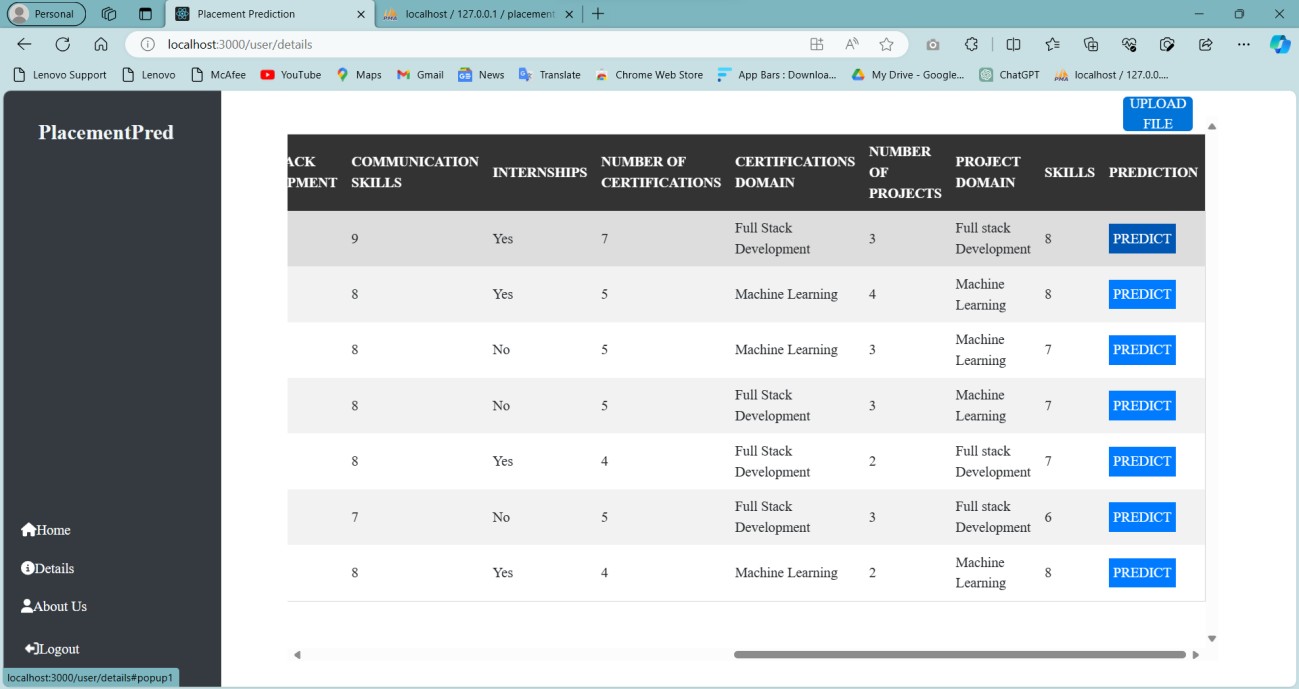
## Fig 5.2.4 Details Page before uploading file



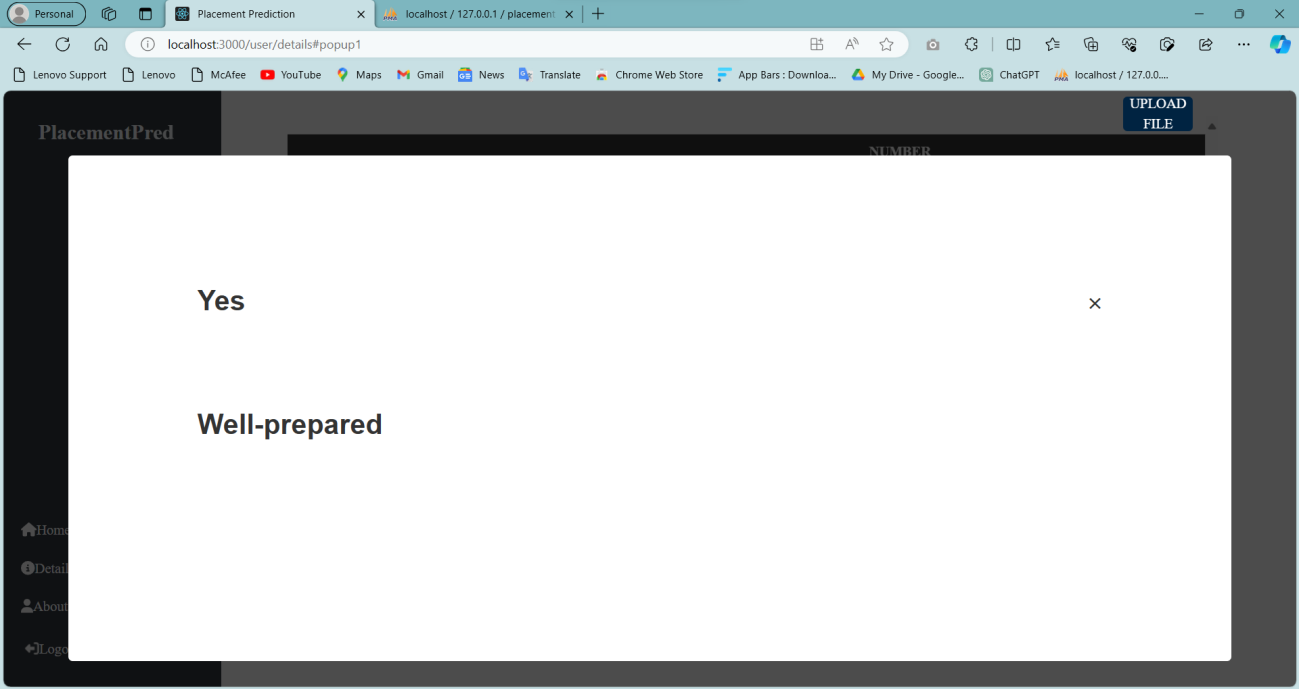
**Fig 5.2.5 Details Page while uploading file**



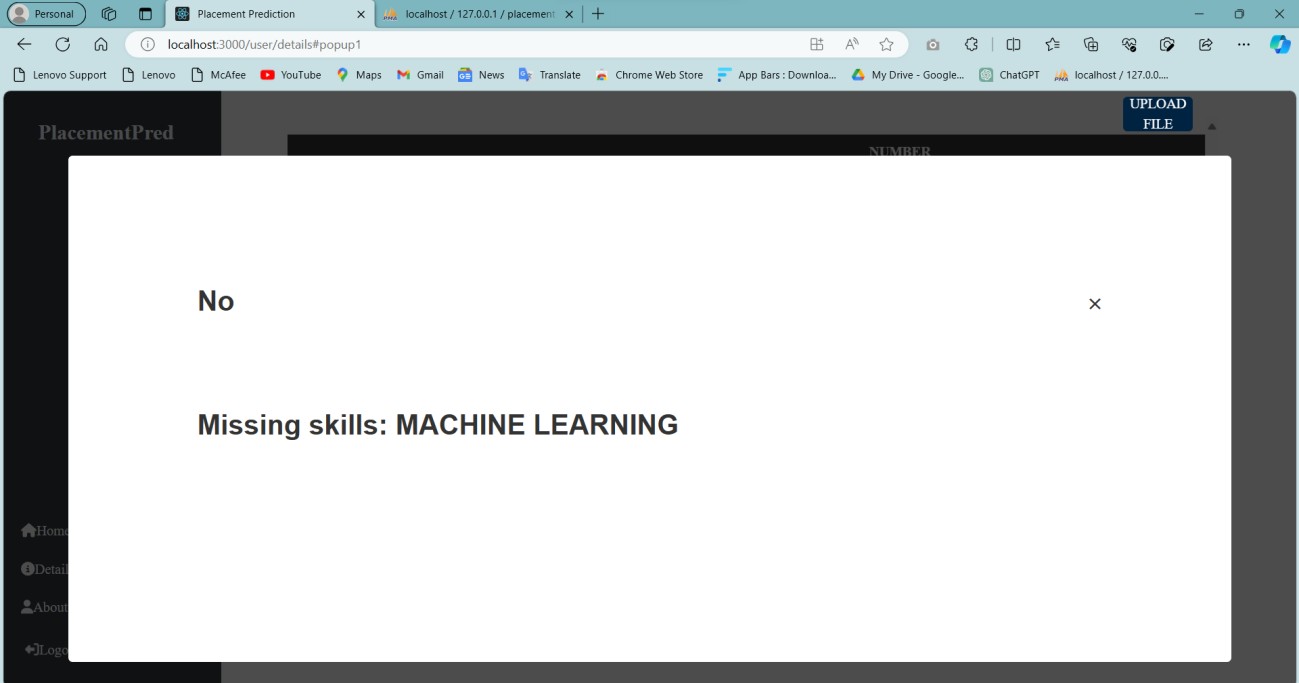
## Fig 5.2.6 Details Page after uploading file



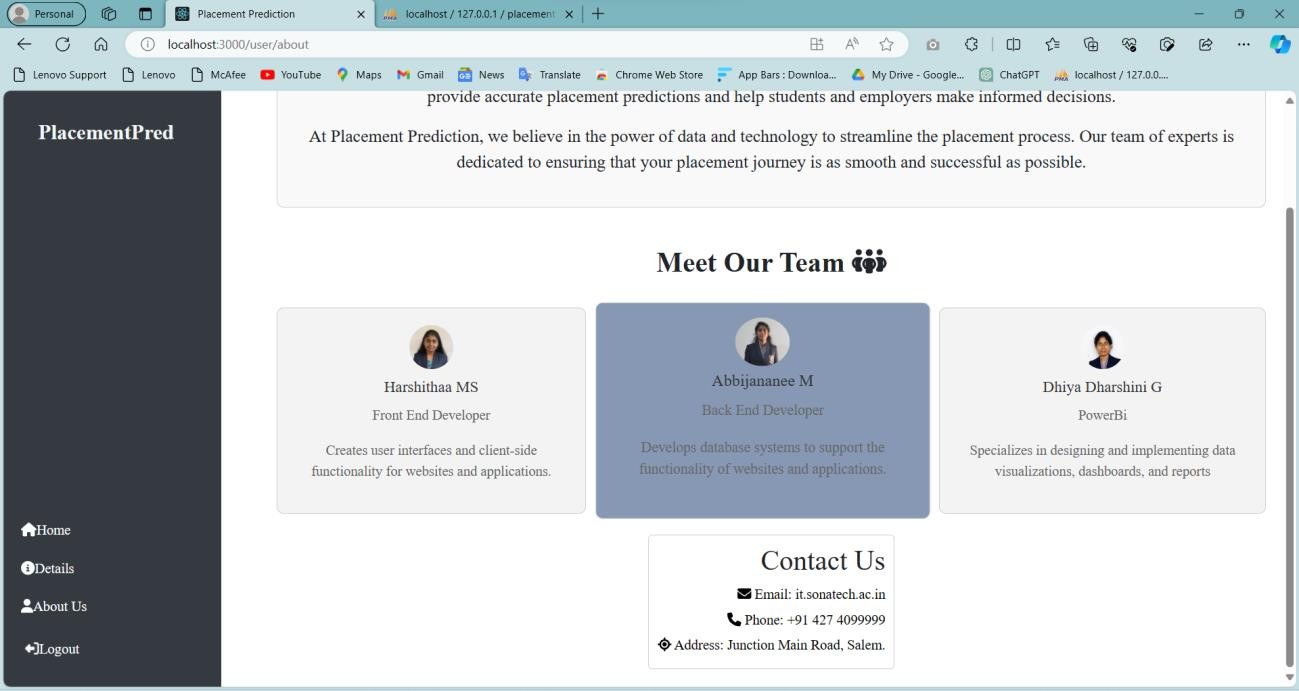
**Fig 5.2.7 Details Page while predicting placement**



## Fig 5.2.8 Details Page after predicting placement YES



**Fig 5.2.9 Details Page after predicting placement -NO**



## Fig 5.2.10 About Us Page

A screenshot of a computer

Description automatically generated

**Fig 5.2.11 Filters Page after clustering salary**

A screenshot of a computer

Description automatically generated

**Fig 5.2.12 Predicting salary based onskill levels-AVERAGE**

A screenshot of a computer

Description automatically generated

**Fig 5.2.13 Predicting salary based on skill levels - LOW**

A screenshot of a computer

Description automatically generated

**Fig 5.2.14 Predicting salary based on skill levels - HIGH**

# CHAPTER 6 CONCLUSIONS AND FUTURE WORK

## CONCLUSIONS

This project has the potential to revolutionize the recruitment and talent development processes by providing a data-driven approach to placement prediction and skill gap identification. HR practitioners and educational institutions can utilize the findings to make more informed decisions in talent acquisition and training, ultimately improving the success rates of placements and enhancing candidate preparedness for the job market.

Here are some specific conclusions about the project:

* + - The proposed machine learning model has the potential to predict student placement and identify skill gaps with high accuracy.
    - The model can be used by students to identify their strengths and weaknesses and develop personalized learning plans.
    - The model can be used by institutions to improve their placement rates and identify areas where they need to provide additional support to their students.
    - The skill gap analysis can provide actionable recommendations for HR professionals and educational institutions to bridge the identified gaps.

Overall, the project has the potential to have a significant impact on the recruitment and talent development processes by making them more data-driven and efficient.

## FUTURE WORK

The proposed project can be further extended in the following ways:

* + - Use a larger and more diverse dataset: The current dataset comprises a diverse range of attributes, but it can be further enriched by including additional features, such as personality traits, extracurricular activities, and social media profiles. This would allow the model to develop more accurate predictions and identify more nuanced skill gaps.
    - Explore different machine learning algorithms: The current project uses a simple machine learning algorithm to predict student placement and identify skill gaps. However, more sophisticated algorithms can be explored, such as deep learning models, to improve the performance of the model.
    - Develop a more interactive model: The current model is designed to predict student placement and identify skill gaps in a batch setting. However, a more interactive model can be developed that would allow users to input specific information about a student or candidate to receive personalized recommendations and insights.
    - Deploy the model to a production environment: The current model is a research prototype. However, it can be deployed to a production environment to make it accessible to students, institutions, and HR professionals. This would allow the model to be used to improve the recruitment and talent development process on a wider scale.

By addressing these future work directions, the proposed project can have a greater impact on student placement outcomes and the recruitment and talent development process.

**APPENDICES**

**Details.js**

import React, { Component } from 'react'; import "./Details.css"

import { useNavigate } from 'react-router-dom';

import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'; import { faFileArrowUp} from '@fortawesome/free-solid-svg-icons'; import axios from 'axios';

import { data } from 'jquery'; class Details extends Component{

constructor(props) { super(props); this.state = { data:[], currentPage: 1,

itemsPerPage: 7, predict:'',

skill:[],

};}

showFileInput = () => {

this.setState({ fileInputVisible: true });} handleFileUpload = (e) => {

const file = e.target.files[0];

if (file && file.type === 'text/csv') { const formData = new FormData(); formData.append('file', file);

axios.post('http://127.0.0.1:5000/upload\_students', formData, { headers: {

'Content-Type': 'multipart/form-data',

},})

.then((response) => {

alert('Data uploaded successfully'); window.location.reload();

})

.catch((error) => {

console.error('Error uploading data:', error);

});

} else {

alert('Please select a valid CSV file.'); } } handleSubmit = (e)=> {

e.preventDefault()

}

handlePageChange = (newPage) => { this.setState({ currentPage: newPage });

};

componentDidMount(){ axios.get("http://127.0.0.1:5000/student\_data")

.then(response => {

this.setState({ data: response.data }); console.log(response.data)

})

.catch(error => {

console.error("Error fetching data:", error);

}); }

handleSubmit = (e)=> { e.preventDefault()

}

handlePageChange = (newPage) => { this.setState({ currentPage: newPage });

};

componentDidMount(){ axios.get("http://127.0.0.1:5000/student\_data")

.then(response => {

this.setState({ data: response.data }); console.log(data.length)

})

.catch(error => {

console.error("Error fetching data:", error);

}); }

renderTable() {

if (this.state.data.length === 0) { return <p>No data available.</p>; }

const { data, currentPage, itemsPerPage } = this.state; const startIndex = (currentPage - 1) \* itemsPerPage; const endIndex = startIndex + itemsPerPage;

const currentData = data.slice(startIndex, endIndex); return (

<div className='detail'>

<div className='button'>

<input type="file" name="file" accept=".csv" onChange={this.handleFileUpload} />

</div>

<div className='mydiv' style={{ overflowY: 'scroll', maxHeight: '700px',scrollbarWidth: 'none', WebkitOverflowScrolling: 'touch', scrollbarColor:'transparent' }}>

<table>

<thead>

<tr>

<th>NAMES</th>

<th>ACADEMICS</th>

<th>MOCK INTERVIEW</th>

<th>PYTHON</th>

<th>JAVA</th>

<th>DBMS</th>

<th>COMPUTER NETWORKS</th>

<th>MACHINE LEARNING</th>

<th>APTITUDE SKILLS</th>

<th>FULL STACK DEVELOPMENT</th>

<th>COMMUNICATION SKILLS</th>

<th>INTERNSHIPS</th>

<th>NUMBER OF CERTIFICATIONS</th>

<th>CERTIFICATIONS DOMAIN</th>

<th>NUMBER OF PROJECTS</th>

<th>PROJECT DOMAIN</th>

<th>SKILLS</th>

<th>PREDICTION</th>

</tr>

</thead>

<tbody>

{currentData.length > 3 ? ( currentData.map((student, index) => (

<tr key={student.id}>

<td>{student.NAMES}</td>

<td>{student.ACADEMICS}</td>

<td>{student["MOCK INTERVIEW"]}</td>

<td>{student.PYTHON}</td>

<td>{student.JAVA}</td>

<td>{student.DBMS}</td>

<td>{student["COMPUTER NETWORKS"]}</td>

<td>{student["MACHINE LEARNING"]}</td>

<td>{student["APTITUDE SKILLS"]}</td>

<td>{student["FULL STACK DEVELOPMENT"]}</td>

<td>{student["COMMUNICATION SKILLS"]}</td>

<td>{student.INTERNSHIPS}</td>

<td>{student["NUMBER OF CERTIFICATIONS"]}</td>

<td>{student["CERTIFICATIONS DOMAIN"]}</td>

<td>{student["NUMBER OF PROJECTS"]}</td>

<td>{student["PROJECT DOMAIN"]}</td>

<td>{student.SKILLS}</td>

<td>

<a href='#popup1'><button className='predict' onClick={() => this.setState({ predict:student.STATUS, skill: student.REMARKS

})}>PREDICT</button></a>

</td>

</tr>

))

) : (

<tr>

<td colSpan="18">No data available.</td>

</tr>

)}

</tbody>

</table>

<div className="pagination" style={{float:'left'}}>

{currentPage > 1 && (

<button onClick={() => this.handlePageChange(currentPage - 1)}>Previous</button> )}

{currentData.length === itemsPerPage && (

<button onClick={() => this.handlePageChange(currentPage + 1)}>Next</button> )}

</div>

</div>

</div>

); }

handleAction(id) {

console.log(`Button clicked for student ID ${id}`);} render(){

return(

<div>

{this.renderTable()}

<div id="popup1" className="overlay1">

<div className="popup1">

{/\* eslint-disable-next-line \*/}

<a className="close" href="#">&times;</a>

<h2>{this.state.predict}</h2>

<h2>{this.state.skill}</h2>

</div>

</div>

</div>

); }

}

export default Details;

export function ResultRouter(props){ const navigate=useNavigate()

return (<Details navigate={navigate}></Details>)

}

## Login.js

import React,{useState, useEffect} from "react"; import './Login.css';

import axios from "axios";

import {Link,useNavigate} from 'react-router-dom'; import Select from 'react-select';

function Login(){

const [mail,setmail]=useState(''); const [pwd,setpwd]=useState(''); const navigate=useNavigate(); const lcheck = (e)=>{ e.preventDefault();

try{ axios.post("http://localhost:5000/login",{ umail:mail,

upwd:pwd,

}).then((response)=>{

if(response.data==="Success"){ navigate("/user/home")}

else{

alert("Invalid Username/Password")

}

});

}

catch(error){ console.log(error)

}

}

useEffect(() => { window.addEventListener('popstate', (e) => { window.history.go(1);

});

}, []);

return(

<div className="body1">

<div className="box-form">

<div className="left">

<div className="overlay">

<h1>Let's Login</h1>

<span>

<Link to="/register">

<div> Register

</div>

</Link>

</span>

</div>

</div>

<div className="right">

<h4>Login</h4>

<p>Let's Log you in..It takes less than a minute</p>

<form onSubmit={lcheck}>

<div className="inputs">

<input type="email" placeholder="Enter Your mail" onChange={(e)=>{ setmail(e.target.value);

}}/>

<input style={{marginBottom:'30px'}} type="password" placeholder="Enter Password" onChange={(e)=>{

setpwd(e.target.value);

}}/>

</div>

<button style={{marginTop:'30px'}} type="submit">Login</button>

</form>

</div>

</div>

</div>

)

}

export default Login;

## About.js

import React from 'react';

import './About.css'; // Create this CSS file for styling

import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'; import {faEnvelope, faPhone, faLocation, faPeopleGroup} from '@fortawesome/free-solid-svg-icons';

import harsh from './images/harsh.jpeg' import dd from './images/dhivya.jpeg' import abbi from './images/abbi.jpeg'

const About = () => { return (

<div className="about-us-container">

<div className="about-us-content">

<h1>About Us - Placement Prediction</h1><br></br>

<p>

Welcome to Placement Prediction, your trusted platform for predicting and optimizing

placement opportunities. Our mission is to provide accurate placement predictions and help students and employers make informed decisions.

</p>

<p>

At Placement Prediction, we believe in the power of data and technology to streamline the placement process. Our team of experts is dedicated to ensuring that your placement journey is as smooth and successful as possible.

</p>

</div>

<div className="developers">

<br></br>

<h2>Meet Our Team <FontAwesomeIcon icon={faPeopleGroup}

/></h2><br></br>

<div className="developer-tiles">

<div className="developer-tile">

<img src={harsh} alt="harsh" width="50px" />

<h3>Harshithaa MS</h3>

<p>Front End Developer</p>

<p> Creates user interfaces and client-side functionality for websites and applications.</p>

</div>

<div className="developer-tile">

<img src={abbi} alt="harsh" width="50px" />

<h3>Abbijananee M</h3>

<p>Back End Developer</p>

<p>Develops database systems to support the functionality of websites and applications.</p>

</div>

<div className="developer-tile">

<img src={dd} alt="harsh" width="50px" />

<h3>Dhiya Dharshini G

</h3>

<p>PowerBi </p>

<p>Specializes in designing and implementing data visualizations, dashboards, and reports</p>

</div>

</div>

</div>

<div><br></br>

<div className="contact-us-container">

<div className="contact-details">

<h2>Contact Us</h2>

<div className="contact-icons">

<p><FontAwesomeIcon icon={faEnvelope} /> Email: it.sonatech.ac.in</p>

<p><FontAwesomeIcon icon={faPhone} /> Phone: +91 427 4099999</p>

<p><FontAwesomeIcon icon={faLocation} /> Address: Junction Main Road, Salem.</p>

</div>

</div>

</div>

</div>

</div>

);

};

export default About;

**Cluster.js**

import React, { useEffect, useState } from 'react';

import './Cluster.css';

function Cluster() {

const [data, setData] = useState([]);

const [filteredData, setFilteredData] = useState([]);

const [salaryInput, setSalaryInput] = useState('');

const [currentPage, setCurrentPage] = useState(1);

const itemsPerPage = 15; // Set the number of items per page

useEffect(() => {

fetchData();

}, []);

const fetchData = async () => {

try {

const response = await fetch('http://127.0.0.1:5000/get\_clusters');

const result = await response.json();

setData(result);

setFilteredData(result);

} catch (error) {

console.error('Error fetching data:', error);

}

};

const handleSalaryInputChange = (event) => {

setSalaryInput(event.target.value);

};

const handleFilter = () => {

const filtered = data.filter(item => item.SALARY === parseInt(salaryInput, 10));

setFilteredData(filtered);

setCurrentPage(1); // Reset to the first page when filtering

};

const indexOfLastItem = currentPage \* itemsPerPage;

const indexOfFirstItem = indexOfLastItem - itemsPerPage;

const currentItems = filteredData.slice(indexOfFirstItem, indexOfLastItem);

const paginate = (pageNumber) => setCurrentPage(pageNumber);

const Pagination = () => {

const totalPages = Math.ceil(filteredData.length / itemsPerPage);

const handlePrevClick = () => {

setCurrentPage((prevPage) => Math.max(prevPage - 1, 1));

};

const handleNextClick = () => {

setCurrentPage((prevPage) => Math.min(prevPage + 1, totalPages));

};

return (

<div className="pagination">

<button onClick={handlePrevClick} disabled={currentPage === 1}>

Previous

</button>

{Array.from({ length: totalPages }).map((\_, index) => (

<button

key={index + 1}

onClick={() => paginate(index + 1)}

className={currentPage === index + 1 ? 'active' : ''}

>

{index + 1}

</button>

))}

<button onClick={handleNextClick} disabled={currentPage === totalPages}>

Next

</button>

</div>

);

};

return (

<div className='content'>

<div className='contain'>

<label htmlFor="salaryInput">Enter Salary: </label>

<input

type="number"

id="salaryInput"

value={salaryInput}

onChange={handleSalaryInputChange}

/>

<button onClick={handleFilter}>Filter</button>

</div>

<div className='table-container'>

<table className='feature'>

<thead>

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

</thead>

<tbody>

{currentItems.map(item => (

<tr key={item.NAMES}>

<td>{item.NAMES}</td>

<td>{item.SALARY}</td>

</tr>

))}

</tbody>

</table>

</div>

<Pagination />

</div>

);

}

export default Cluster;

## Main.py (Backend)

from flask import Flask,request,jsonify

from flask\_cors import CORS

import random

from email.mime.multipart import MIMEMultipart

from email.mime.image import MIMEImage

from email.mime.text import MIMEText

import smtplib,ssl

import numpy as np

import mysql.connector

import joblib

import csv

import json

import os

from werkzeug.utils import secure\_filename

import pandas as pd

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import AdaBoostClassifier, AdaBoostRegressor

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler, LabelEncoder

# import requests

from sklearn.impute import SimpleImputer

import PyPDF2

import re

app=Flask(\_\_name\_\_)

CORS(app)

UPLOAD\_FOLDER = 'myfiles'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

mydb = mysql.connector.connect(

host="localhost",

user="root",

password="",

database="placement"

)

# create a mycursor object

mycursor = mydb.cursor()

#Define the list of required skills along with their required threshold values

required\_skills = {

"PYTHON": 5,

"JAVA": 5,

"DBMS": 5,

"COMPUTER NETWORKS": 5,

"MACHINE LEARNING": 5,

"APTITUDE SKILLS": 5,

"FULL STACK DEVELOPMENT": 5,

"COMMUNICATION SKILLS": 5

}

def pdf\_text\_extractor(pdf\_path):

text = ""

with open(pdf\_path, 'rb') as file:

pdf\_reader = PyPDF2.PdfReader(file)

num\_pages = len(pdf\_reader.pages)

for page\_num in range(num\_pages):

page = pdf\_reader.pages[page\_num]

text += page.extract\_text()

skills\_text = extract\_skills(text)

return skills\_text

def extract\_skills(resume\_text):

# Use a regular expression to capture the skills section

skills\_match = re.search(r'TECHNICAL SKILLS([\s\S]+?)(?:\n\n|\n\n\n)', resume\_text)

if skills\_match:

return skills\_match.group(1).strip()

return "Skills section not found."

# Function to identify skill gaps for a student

def identify\_skill\_gaps(student):

gaps = []

for skill, threshold in required\_skills.items():

if student[skill] < threshold:

gaps.append(skill)

return gaps

def mail\_send(otp,mail):

try:

s = smtplib.SMTP('smtp.office365.com', 587)

except Exception as e:

s = smtplib.SMTP\_SSL('smtp.office365.com', 465)

s.ehlo()

s.starttls()

s.login("abbijananee.20it@sonatech.ac.in", "Mother5abbi")

msg = MIMEMultipart()

msg['From']='abbijananee.20it@sonatech.ac.in'

msg['To']=mail

msg['Subject']="Registration Confirmation"

html=f'''\

<!DOCTYPE html>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<style type="text/css">

body,

table,

td,

a {"-webkit-text-size-adjust: 100%;-ms-text-size-adjust: 100%;"}

table,td {"mso-table-lspace: 0pt;mso-table-rspace: 0pt;"}

img {"-ms-interpolation-mode: bicubic;"}

/\* RESET STYLES \*/

img {"border: 0;height: auto;line-height: 100%;outline: none;text-decoration: none;"}

table {"border-collapse: collapse !important;"}

body {"height: 100% !important;margin: 0 !important;padding: 0 !important;width: 100% !important;"}

/\* iOS BLUE LINKS \*/

a[x-apple-data-detectors] {"color: inherit !important;text-decoration: none !important;font-size: inherit !important;font-family: inherit !important;font-weight: inherit !important;line-height: inherit !important;"}

/\* MOBILE STYLES \*/

/\* ANDROID CENTER FIX \*/

div[style\*="margin: 16px 0;"] {"margin: 0 !important;"}

</style>

</head>

<body style="background-color: #f4f4f4; margin: 0 !important; padding: 0 !important;">

<!-- HIDDEN PREHEADER TEXT -->

<div style="display: none; font-size: 1px; color: #fefefe; line-height: 1px; font-family: 'Lato', Helvetica, Arial, sans-serif; max-height: 0px; max-width: 0px; opacity: 0; overflow: hidden;"> Here is your One Time Password

</div>

<table border="0" cellpadding="0" cellspacing="0" width="100%">

<!-- LOGO -->

<tr>

<td bgcolor="#bf1591" align="center" style="padding: 60px 10px 0px 10px; background-color: linear-gradient(135deg, #f26ace 10%, #bf1591 100%)">

<table border="0" cellpadding="0" cellspacing="0" width="100%" style="max-width: 600px;">

<tr>

<td bgcolor="#ffffff" align="center" valign="top" style="padding: 40px 20px 20px 20px; border-radius: 4px 4px 0px 0px; color: #111111; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 48px; font-weight: 400; letter-spacing: 4px; line-height: 48px;">

<h1 style="font-size: 48px; font-weight: 400; margin: 2;">Hey there!</h1> <img src="https://i.ibb.co/G0t2czh/logo.jpg" width="125" height="120" srcset="" style="display: block; border: 0px;" alt="Logo" />

</td>

</tr>

</table>

</td>

</tr>

<tr>

<td bgcolor="#f4f4f4" align="center" style="padding: 0px 10px 20px 10px;">

<table border="0" cellpadding="0" cellspacing="0" width="100%" style="max-width: 600px;">

<tr>

<td bgcolor="#ffffff" align="left" style="padding: 20px 30px 10px 30px; color: #666666; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 18px; font-weight: 400; line-height: 25px;">

<h3 style="margin: 0; " align="center">Here is your One Time Password</h3>

</td>

</tr>

<tr>

<td bgcolor="#ffffff" align="left" style=" color: #666666; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 18px; font-weight: 400; line-height: 25px;">

<p style="margin: 0; " align="center">to validate your email address</p>

</td>

</tr>

<tr>

<td bgcolor="#ffffff" align="left">

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td bgcolor="#ffffff" align="center" style="padding: 0px 5px 0px 20px;">

<table border="0" cellspacing="0" cellpadding="0">

<tr>

<td align="center" style="border-radius: 3px; " ><h1 style="font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 70px; letter-spacing: 15px;">{otp}</h1></td>

</tr>

</table>

</td>

</tr>

</table>

</td>

</tr> <!-- COPY -->

<tr>

<td bgcolor="#ffffff" align="left" style=" color: #666666; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 18px; font-weight: 400; line-height: 25px; padding-bottom: 20px;">

<p style="margin: 0; color: #ff4d4d;" align="center" >Valid for 5 minutes only</p>

</td>

</tr>

<tr>

<td bgcolor="#ffffff" align="left" style="padding: 0px 30px 20px 30px; color: #666666; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 18px; font-weight: 400; line-height: 25px;">

<p style="margin: 0;" align="center">If you didn't request this , you can ignore this email.</p>

</td>

</tr>

<tr>

<td bgcolor="#ffffff" align="left" style="padding: 0px 30px 40px 30px; border-radius: 0px 0px 4px 4px; color: #666666; font-family: 'Lato', Helvetica, Arial, sans-serif; font-size: 18px; font-weight: 400; line-height: 25px;">

<p style="margin: 0; "align="center">Thanks!<br>CB Team</p>

</td>

</tr>

</table>

</td>

</tr>

</table>

</body>

</html>

'''

msg.attach(MIMEText(html, 'html'))

s.send\_message(msg)

return "Success"

def otp\_gen():

digit="0123456789"

password=""

i=0

for i in range(6):

password=password+random.choice(digit)

i+1

print("Your password is "+str(password))

file1 = open("myfile.txt","w")

file1.write(password)

file1.close()

return password

@app.route('/register',methods=["POST"])

def register():

mail=request.json['umail']

pwd=request.json['upwd']

opt=otp\_gen()

mm=mail\_send(opt, mail)

sql = "INSERT INTO login\_table (user\_mail,user\_password) VALUES (%s, %s)"

values = (mail, pwd)

mycursor.execute(sql, values)

mydb.commit()

return jsonify(opt)

@app.route('/sregister',methods=["POST"])

def sregister():

mail=request.json['umail']

pwd=request.json['upwd']

opt=otp\_gen()

mm=mail\_send(opt, mail)

sql = "INSERT INTO login\_table (user\_mail,user\_password) VALUES (%s, %s)"

values = (mail, pwd)

mycursor.execute(sql, values)

mydb.commit()

return jsonify(opt)

@app.route('/otp',methods=["GET"])

def otp():

file1 = open("myfile.txt","r+")

myotp=file1.read()

return jsonify(myotp)

@app.route('/student\_data',methods=["GET"])

def data():

sql = "SELECT \* FROM student\_data"

mycursor.execute(sql)

result = mycursor.fetchall()

if result:

data = [dict(zip([key[0] for key in mycursor.description], row)) for row in result]

return jsonify(data)

else:

return jsonify('Not')

@app.route('/login',methods=["POST"])

def login():

mail=request.json['umail']

pwd=request.json['upwd']

sql = "SELECT \* FROM login\_table WHERE user\_mail = %s AND user\_password = %s"

values = (mail, pwd)

mycursor.execute(sql, values)

result = mycursor.fetchone()

if result:

return jsonify('Success')

else:

return jsonify('Not')

@app.route('/upload\_students', methods=['POST'])

def upload\_students():

# Get the uploaded CSV file

csv\_file = request.files['file']

if not csv\_file:

return jsonify({'error': 'No file provided'}), 400

if not csv\_file.filename.endswith('.csv'):

return jsonify({'error': 'Invalid file format. Please upload a CSV file'}), 400

filename = secure\_filename(csv\_file.filename)

file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

csv\_file.save(file\_path)

if os.path.exists(file\_path):

df = pd.read\_csv(file\_path)

columns\_to\_fill = ['ACADEMICS', 'MOCK INTERVIEW', 'PYTHON', 'JAVA', 'DBMS', 'COMPUTER NETWORKS',

'MACHINE LEARNING', 'APTITUDE SKILLS', 'FULL STACK DEVELOPMENT',

'NUMBER OF CERTIFICATIONS', 'NUMBER OF PROJECTS']

numeric\_columns = df.select\_dtypes(include=[np.number]).columns

for column in columns\_to\_fill:

zero\_rows = df[df[column] == 0]

if not zero\_rows.empty:

train\_data = df[df[column] != 0]

y = train\_data[column]

X = train\_data[numeric\_columns]

imputer = SimpleImputer(strategy='mean')

X[numeric\_columns] = imputer.fit\_transform(X)

model = AdaBoostRegressor()

model.fit(X, y)

prediction\_features = zero\_rows[numeric\_columns]

prediction\_features[numeric\_columns] = imputer.transform(prediction\_features)

predicted\_values = model.predict(prediction\_features)

df.loc[df[column] == 0, column] = predicted\_values

selected\_features = df[['PYTHON', 'JAVA', 'DBMS', 'COMPUTER NETWORKS', 'MACHINE LEARNING', 'APTITUDE SKILLS', 'FULL STACK DEVELOPMENT', 'COMMUNICATION SKILLS']]

target\_variable = df['SKILLS']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(selected\_features, target\_variable, test\_size=0.2, random\_state=42)

base\_classifier = DecisionTreeClassifier(max\_depth=1)

adaboost\_classifier = AdaBoostClassifier(base\_classifier, n\_estimators=50, random\_state=42)

adaboost\_classifier.fit(X\_train, y\_train)

y\_pred = adaboost\_classifier.predict(X\_test)

y\_pred\_probabilities = adaboost\_classifier.predict\_proba(selected\_features)

threshold = 0.5

df['STATUS'] = ['Yes' if score[1] > threshold else 'No' for score in y\_pred\_probabilities]

df['REMARKS'] = ""

for index, row in df.iterrows():

student\_name = row['NAMES']

gaps = identify\_skill\_gaps(row)

if df.at[index, 'STATUS'] == 'Yes':

df.at[index, 'REMARKS'] = "Well-prepared"

else:

if not gaps:

df.at[index, 'REMARKS'] = "Well-prepared"

df.at[index, 'STATUS'] = "Yes"

else:

missing\_skills = ', '.join(gaps)

df.at[index, 'REMARKS'] = f"Missing skills: {missing\_skills}"

df.to\_csv('./myfiles/updated.csv', index=False)

with open('./myfiles/updated.csv', 'r', encoding='utf-8') as file:

csv\_reader = csv.DictReader(file)

for row in csv\_reader:

sql\_check = "SELECT NAMES FROM student\_data WHERE NAMES = %s"

values\_check = (row['NAMES'],)

mycursor.execute(sql\_check, values\_check)

result\_check = mycursor.fetchone()

if result\_check:

sql\_update = """

UPDATE student\_data

SET ACADEMICS = %s, `MOCK INTERVIEW` = %s, PYTHON = %s, JAVA = %s,

DBMS = %s, `COMPUTER NETWORKS` = %s, `MACHINE LEARNING` = %s,

`APTITUDE SKILLS` = %s, `FULL STACK DEVELOPMENT` = %s,

`COMMUNICATION SKILLS` = %s, INTERNSHIPS = %s,

`NUMBER OF CERTIFICATIONS` = %s, `CERTIFICATIONS DOMAIN` = %s,

`NUMBER OF PROJECTS` = %s, `PROJECT DOMAIN` = %s, SKILLS = %s,

STATUS = %s, REMARKS = %s

WHERE NAMES = %s

"""

values\_update = (row['ACADEMICS'], row['MOCK INTERVIEW'], row['PYTHON'],

row['JAVA'], row['DBMS'], row['COMPUTER NETWORKS'],

row['MACHINE LEARNING'], row['APTITUDE SKILLS'],

row['FULL STACK DEVELOPMENT'], row['COMMUNICATION SKILLS'],

row['INTERNSHIPS'], row['NUMBER OF CERTIFICATIONS'],

row['CERTIFICATIONS DOMAIN'], row['NUMBER OF PROJECTS'],

row['PROJECT DOMAIN'], row['SKILLS'], row['STATUS'],

row['REMARKS'], row['NAMES'])

mycursor.execute(sql\_update, values\_update)

else:

sql\_insert = """

INSERT INTO student\_data (NAMES, ACADEMICS, `MOCK INTERVIEW`, PYTHON,

JAVA, DBMS, `COMPUTER NETWORKS`, `MACHINE LEARNING`, `APTITUDE SKILLS`,

`FULL STACK DEVELOPMENT`, `COMMUNICATION SKILLS`, INTERNSHIPS,

`NUMBER OF CERTIFICATIONS`, `CERTIFICATIONS DOMAIN`, `NUMBER OF PROJECTS`,

`PROJECT DOMAIN`, SKILLS, STATUS, REMARKS)

VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)

"""

values\_insert = (row['NAMES'], row['ACADEMICS'], row['MOCK INTERVIEW'], row['PYTHON'],

row['JAVA'], row['DBMS'], row['COMPUTER NETWORKS'],

row['MACHINE LEARNING'], row['APTITUDE SKILLS'],

row['FULL STACK DEVELOPMENT'], row['COMMUNICATION SKILLS'],

row['INTERNSHIPS'], row['NUMBER OF CERTIFICATIONS'],

row['CERTIFICATIONS DOMAIN'], row['NUMBER OF PROJECTS'],

row['PROJECT DOMAIN'], row['SKILLS'], row['STATUS'], row['REMARKS'])

mycursor.execute(sql\_insert, values\_insert)

mydb.commit()

return jsonify({'message': 'Student data uploaded successfully'}), 200

return jsonify({'message': 'Student data uploaded successfully'}), 200

@app.route('/delete\_student',methods=['POST'])

def delete\_student():

mail=request.json['id']

sql = "DELETE FROM student\_data WHERE id= %s"

values = (mail,)

print(mail)

mycursor.execute(sql, values)

return jsonify('Success')

@app.route('/resume', methods=['POST'])

def resume():

csv\_file = request.files['file']

if not csv\_file:

return jsonify({'error': 'No file provided'}), 400

if not csv\_file.filename.endswith('.pdf'):

return jsonify({'error': 'Invalid file format. Please upload a PDF file'}), 400

filename = secure\_filename(csv\_file.filename)

file\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

csv\_file.save(file\_path)

extracted\_text = pdf\_text\_extractor(file\_path)

return jsonify(extracted\_text)

@app.route('/get\_clusters', methods=['GET'])

def get\_clusters():

df = pd.read\_csv("C:\\Users\\exam01\\Downloads\\data\\place.csv")

features = df[['ACADEMICS', 'MOCK INTERVIEW', 'PYTHON', 'JAVA', 'DBMS', 'COMPUTER NETWORKS', 'MACHINE LEARNING', 'APTITUDE SKILLS', 'FULL STACK DEVELOPMENT', 'COMMUNICATION SKILLS', 'INTERNSHIPS', 'NUMBER OF CERTIFICATIONS', 'NUMBER OF PROJECTS']]

le = LabelEncoder()

for col in features.columns:

if features[col].dtype == 'object':

features[col] = le.fit\_transform(features[col])

scaler = StandardScaler()

features\_scaled = scaler.fit\_transform(features)

k = 3

kmeans = KMeans(n\_clusters=k, random\_state=42)

df['CLUSTER'] = kmeans.fit\_predict(features\_scaled)

df['SALARY'] = np.random.choice([300000, 500000, 700000], size=len(df))

result = df[['NAMES', 'SALARY', 'CLUSTER']].to\_dict(orient='records')

return jsonify(result)

data = pd.read\_csv('C:\\Users\\exam01\\Downloads\\data\\place.csv')

@app.route('/api/get\_data', methods=['GET'])

def get\_data():

return jsonify(data.to\_dict(orient='records'))

@app.route('/api/calculate\_salary', methods=['POST'])

def calculate\_salary():

req\_data = request.get\_json()

x = np.array([req\_data['PYTHON'], req\_data['JAVA'], req\_data['DBMS'], req\_data['COMPUTER NETWORKS'], req\_data['MACHINE LEARNING'], req\_data['APTITUDE SKILLS'], req\_data['FULL STACK DEVELOPMENT'], req\_data['COMMUNICATION SKILLS']])

total\_score = np.sum(x)

assigned\_salary = 300000 if total\_score < 50 else (500000 if total\_score > 60 else 700000)

return jsonify({'assigned\_salary': assigned\_salary})

if '\_\_main\_\_'== \_\_name\_\_:

app.run(debug=True)

## REFERENCES

1. Kajal Rai, Pawan Kumar, Sanjay Sood, "Anticipating Placement Status of Engineering Students using Machine Learning based Prediction Models - A Case Study of Computer Science Stream", 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), pp.181-185, 2023.
2. T. Jeevalatha, N. Ananthi and D. Saravana Kumar, "Performance Analysis of Undergraduate Students Placement Selection using Decision Tree Algorithms", International Journal of Computer Applications, vol. 108, no. 0975 – 8887.
3. Shreyas Harinath, Aksha Prasad, H Suma and A. Suraksha, "Student Placement Prediction using Machine Learning", International Research Journal of Engineering and Technology (IRJIET), vol. 06, no. 04.
4. K. Sreenivasa Rao, N. Swapna and P. Praveen Kumar, "Educational data mining for student placement prediction using machine Learning algorithms Research Paper", International Research Journal of Engineering and Technology (IRJIET), vol. 20182015, no. 153, pp. 286-299.
5. "Campus placement Prediction Using Supervised Machine Learning Techniques", International Journal of Applied Engineering Research, vol. 14, no. 9, pp. 2188- 2191, 2019, ISSN 0973-4562.
6. C. Beaulac and J. S. Rosenthal, "Predicting university students' academic success and major using random forests", Research in Higher Education, vol. 60, no. 7, pp. 1048-1064, 2019.
7. T. Aravind, B. S. Reddy, S. Avinash and J. G, "A Comparative Study on Machine Learning Algorithms for Predicting the Placement Information of Undergraduate Students", 2019 Third International conference on I-SMAC (IoT in Social Mobile Analytics and Cloud) (I -SMAC), pp. 542-546, 2019.
8. E. Çakit and M. Dagdeviren, "Predicting the percentage of student placement: A comparative study of machine learning algorithms", Educ Inf Technol, vol. 27, pp. 997-1022, 2022.
9. Md Shadab Hussain and Sarita Singh, "Developing classifiers through Machine learning algorithms for student placement prediction based on academic performance", Applied Artificial Intelligence, vol. 35, no. 6, 2021.
10. Senthil Kumar Thangavel, Bharathi P Divya and Abhijith Shankar, "Student Placement Analyzer: A recommendation System Using Machine Learning", International Conference on advanced computing and Communication systems (ICACCS-2017), Jan 06–07,2017.