IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

Campus Management System

K Syam Kumari[1], Yekula Sathwika[2], Bandaru Radhika[3], Pinnamaneni Sreeja[4], Mekala Surekha[5]
[1]Assistant Professor, Department of Information Technology
[2,3,4,5]UG student, Department of Information Technology, Branch: Computer Science and Business System
SRKR Engineering College, Bhimavaram, Andhra Pradesh, India

Abstract: *The Campus Management System (CMS) represents a pivotal advancement in educational technology, offering an integrated software solution tailored to the complex needs of modern educational institutions. This comprehensive system is designed to revolutionize administrative tasks, enhance efficiency, and foster a more organized campus environment. Through seamless automation of various processes, CMS empowers administrators, faculty, and students to navigate campus operations with ease. Utilizing Full Stack Development, CMS ensures seamless integration of front-end and back-end components, resulting in a user-friendly interface accessible to all stakeholders. This approach streamlines administrative tasks, simplifying processes such as student enrolment, course management, and resource allocation. Furthermore, CMS harnesses the power of Machine Learning (ML) algorithms to optimize key processes. These ML algorithms enable intelligent automation of tasks such as resume evaluation, eligibility criteria assessment, and course recommendations based on student performance. By analyzing vast amounts of data, CMS empowers institutions to make data-driven decisions, enhancing the overall educational experience for students. The integration of technology within educational institutions through CMS represents a significant step forward in campus management. By leveraging Full Stack Development and Machine Learning algorithms, CMS offers a robust solution to the challenges faced by modern educational institutions, ultimately paving the way for a more efficient, effective, and student-centric campus environment.*

Keywords: Campus Management System (CMS), Full Stack Development, Machine Learning (ML), Administrative tasks, Student-centric environment, HTML, CSS, PHP, Python, Flask

I. INTRODUCTION

An integrated software system called the Campus Management System (CMS) was created to make administrative duties in educational institutions more efficient. By automating several procedures, it improves overall effectiveness, efficiency, and organization. The most recent announcements, articles, and other material are made available to all users, students, and other users through this web application. Utilizing data-driven insights, the challenge of predicting a candidate's eligibility for placement or not provides colleges and students with informed recommendations. We want to develop prediction models that, by analyzing historical campus placement data, combining various factors, and applying state-of-the-art algorithms, can determine the probability of a candidate's suitability for placement. By providing classes, approving resumes, and giving suggestions, it helps students develop their abilities.

1.1 PROBLEM STATEMENT:

Managing administrative responsibilities effectively presents many obstacles for educational institutions, particularly in tertiary education. Manual systems frequently need a lot of time, resources, and error-proneness, which results in process inefficiencies. In order to address these problems, a comprehensive Campus Management System (CMS) development is being proposed. The goal of this system is to increase overall campus administration effectiveness by streamlining workflows, automating various administrative chores.

1.2 PROPOSED SYSTEM :

In educational institutions, administrative task management can be difficult and time-consuming. Manual procedures frequently result in mistakes, inefficiencies, and a lack of transparency. We suggest creating a strong Campus

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

Management System (CMS) that simplifies a range of campus-related tasks in order to address these issues. The goals of the Campus Management System are to increase overall efficiency in educational institutions, improve student services, and expedite administrative chores. It offers a centralized platform for handling several facets of life on campus.

II. ARCHITECTURES

Following are the architectures that we used in our application:

2.1 Full Stack Architecture

High-Level Components:

1. Front-End (User Interface):

• Web-based interface accessible to students, faculty, and administrators.

• Allows users to interact with the system, view notices, register, and access personalized features.

2. Back-End (Server-Side):

• Manages data storage, processing, and business logic.

Consists of the following components:

Web Server: Handles HTTP requests and serves web pages.

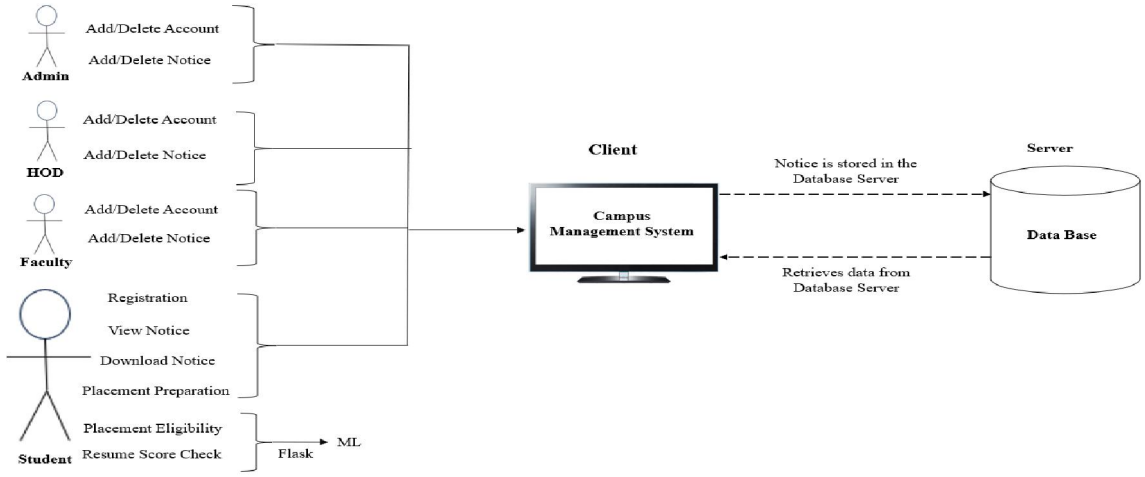
Application Server: Executes application logic (e.g., registration, resume scoring).

Database Server: Stores student profiles, notices, eligibility criteria, and course recommendations.

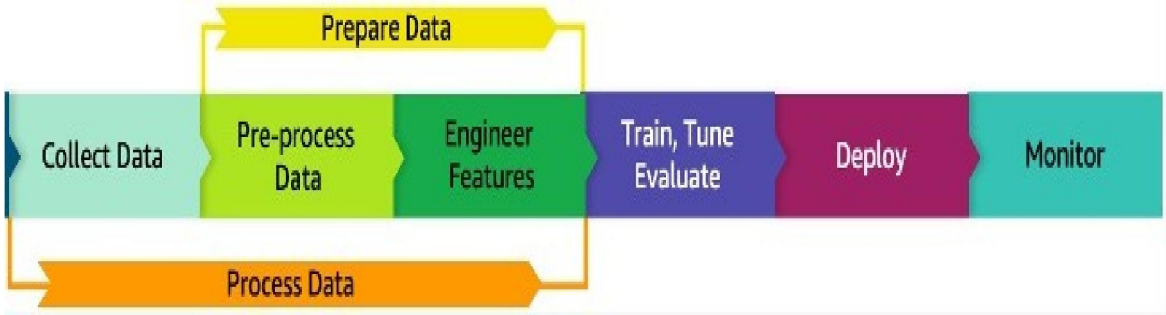
3. Database Management System (DBMS):

• Stores structured data related to students, courses, notices, and eligibility rules.

• Commonly used DBMS: MySQL**.**



2.2 Machine learning Architecture



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

Eligibility for campus placements is predicted by examining a number of variables that may affect a student's placement probability. Here is a suggested machine learning-based solution:

Data Collection: Collect information from prior campus placements, such as placement outcomes (eligible or not), employer profiles (industry, job roles offered, etc.), and student profiles (academic records, skills, etc.). Preparing data: Sort and prepare the data. Encode categorical variables, handle missing values, and normalize numerical features.
Feature Engineering and Selection: Determine which features are pertinent and have an effect on placement results.
Model Selection: Pick suitable machine learning algorithms. Taking into account methods such as Logistic Regression, Random Forest, Support Vector Machines, and Gradient Boosting, given that this is a binary classification problem (placed or not).
Model Training: Divide the data into training and testing sets for the model. Utilizing the training data, train the chosen models. Model Evaluation: Use measures such as accuracy, precision, recall, F1-score, and ROC-AUC to assess the model's performance on the testing data.
Hyperparameter Tuning: To enhance performance, optimize the model's hyperparameters with methods like grid search or random search.
Interpretability: Analyze the model's choices to determine the relative relevance of various features based on the selected algorithm.
Deployment: Use the model to generate predictions in real time if you're happy with its performance. This might be accomplished using an API or web application.
Continuous Learning: To maintain the model current and accurate, gather new placement data and retrain it on a regular basis.

III. IMPLEMENTATION

Requirements Gatherings

• Conduct interviews, surveys, and workshops with stakeholders to gather detailed requirements.
• Document functional and non-functional requirements for each module (notices, registration, etc.).

Development

Online Notices: To create an online notice board with CRUD operations, we use HTML for structure, CSS for styling, and JavaScript for functionality. HTML defines the layout, CSS adds visual appeal, and JavaScript handles adding, reading, updating, and deleting notices.
Deployment: Using Python flask for Placement eligibility check, Resume score checking in student interface.

3.1 TECHNOLOGY USING

Front-end Development: HTML, CSS, JavaScript.
Back-end Development: PHP, MySQL
Tech Stack : Python flask
Full-stack Development: Full-stack developers work on both the front-end and back-end, allowing them to create entire web applications from start to finish.
Machine Learning: Models such as logistic regression, random forests can help predict the suitability of a resume for specific job roles. Classification algorithms such as decision trees, SVMs are used to predict whether a student meets the eligibility criteria.

3.2 Logins :

1. Administrator Interface: Administrators have full control over the Campus Management System.
Administrators can perform any action within the system their interface enables them to:
Profile: Modify /update personal information such as contact details, profile picture, etc.
View Notice : Can view and download notices uploaded by Hod's ,Faculty.
Manage Notice : Can add or delete notices.

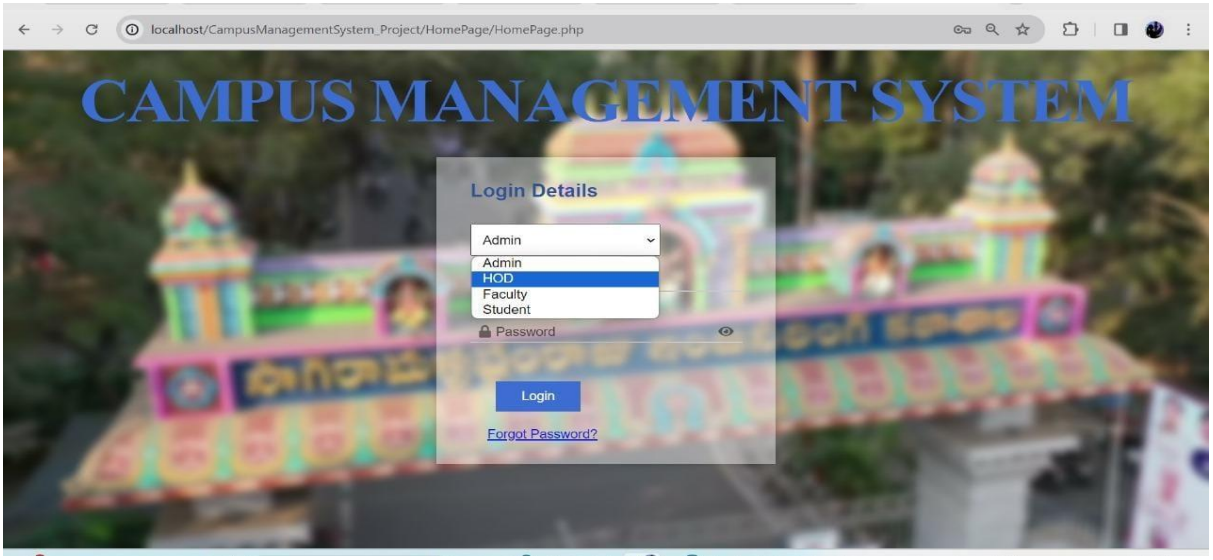
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

Manage HOD : Here admin can add or remove Hod's from data base.
Manage Faculty: Here admin can add or remove faculty from data base.
View Students : Here admin can view students list.
Logout: Can logout from login.

2. Head of Department (HOD) Interface:
Profile: Modify /update personal information such as contact details, profile picture, etc.
View Notice : Can view and download notices uploaded by Admin ,other Hod's ,Faculty.
Manage Notice : Can add or delete notices.
Manage Faculty: Here Hod can add or remove faculty from data base.
View Students : Here Hod can view students list.
Logout: Can logout from login.

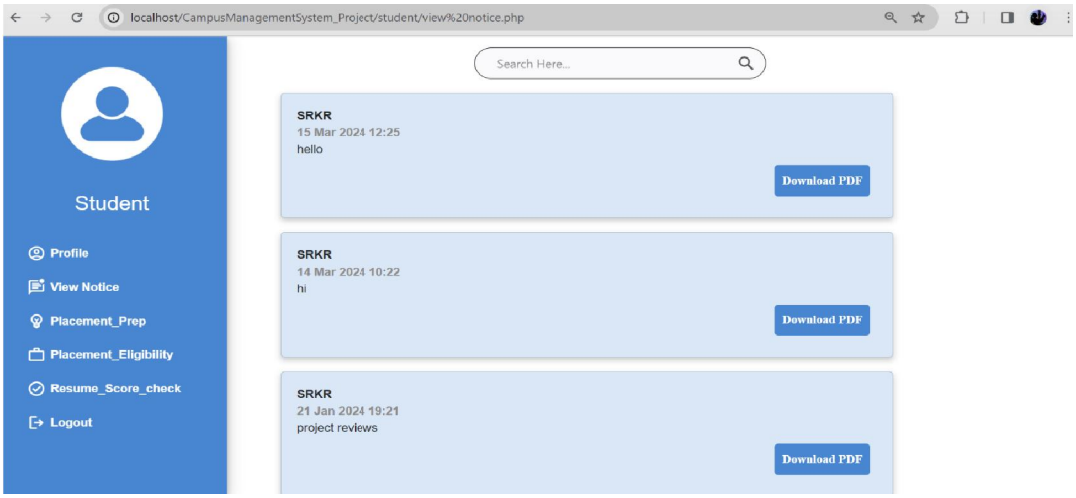
3. Faculty Interface:
Profile: Modify /update personal information such as contact details, profile picture, etc.
View Notice : Can view and download notices uploaded by Admin, Hod's ,other Faculty.
Manage Notice : Can add or delete notices.
Manage Students : Here faculty can add or delete students from data base.
Logout: Can logout from login.

Home page:



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

3.3 Student Interface:



4. Student Interface:

Profile: Modify /update personal information such as contact details, profile picture, etc.
View Notice : Can view and download notices uploaded by Admin, Hod ,other Faculty.
Placement_Prep: provides personalized course recommendations, These suggestions help students enhance their skills and improve their employability.
Placement_Eligibility: The Checking Eligibility Criteria feature allows students to determine whether they meet the requirements for specific job placements.
Resume_Score_Check :feature to evaluate student resumes based on predefined criteria
Logout: Can logout from login.

3.4 Algorithms:

XGB Classifier: The XGBoost (eXtreme Gradient Boosting) Classifier is a high-performance gradient boosting method for classification tasks. It excels at handling large datasets with excellent accuracy and efficiency. By constructing a sequence of decision trees, it corrects errors made by previous trees and offers multiple hyperparameters for fine-tuning model performance. XGBoost is widely used in real-world applications and machine learning contests due to its exceptional predictive capability and scalability.
Decision Tree Regressor: For regression problems, a machine learning algorithm called the Decision Tree Regressor is employed. Based on feature values, it divides the input space into regions and predicts the target variable for each region. It's renowned for being easy to understand, capable of managing non-linear relationships, and simple. Decision Tree Regressors are used to forecast continuous outcomes in environmental science, healthcare, and finance.
LGBM Classifier: The LGBM (LightGBM) Classifier is a scalable gradient boosting architecture that performs exceptionally well on big datasets. It is quick and effective for classification applications. With a leaf-wise tree growth method and histogram-based techniques, it attains quicker training and prediction speeds and provides more hyperparameters for optimising model performance. Because it can handle complex data and produce accurate predictions, LGBM is a popular choice in machine learning applications.
Decision Tree Classifier: Decision tree classifier is a machine learning algorithm used for classification tasks. It partitions the input space into regions based on feature values, selecting the best feature at each step to split the data into classes. It's popular for its interpretability and effectiveness in binary and multi-class classification problems across various domains like finance, healthcare, and marketing.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

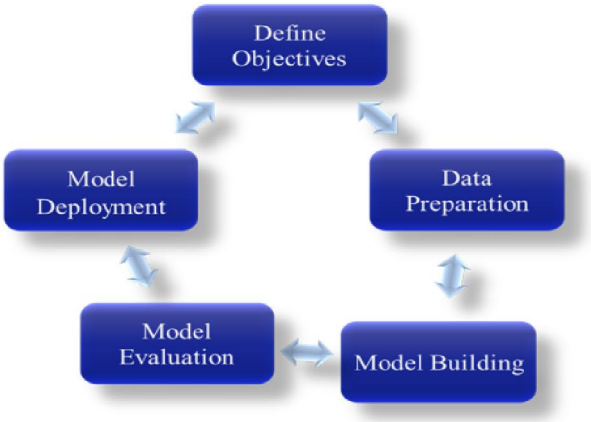
	Model Name	True_Positive	False_Negative	False_Positive	True_Negative	Accuracy	Precision	Recall	F1 Score	Specificity
0	LogisticRegression()	278	57	NaN	195	0.796	0.813	0.83	0.821	0.753
1	DecisionTreeClassifier()	284	51	NaN	233	0.87	0.916	0.848	0.881	0.9
2	(DecisionTreeClassifier(max_features='sqrt', r...	286	49	NaN	229	0.867	0.905	0.854	0.879	0.884
3	(ExtraTreeClassifier(random_state=1477099750),...	282	53	NaN	229	0.86	0.904	0.842	0.872	0.884
4	KNeighborsClassifier()	287	48	NaN	229	0.869	0.905	0.857	0.88	0.884
5	GaussianNB()	275	60	NaN	226	0.843	0.893	0.821	0.855	0.873
6	SVC(probability=True)	274	61	NaN	199	0.796	0.82	0.818	0.819	0.768
7	XGBClassifier(base_score=0.5, booster='gbtree'...	283	52	NaN	244	0.887	0.95	0.845	0.894	0.942
8	LGBMClassifier()	284	51	NaN	239	0.88	0.934	0.848	0.889	0.923
9	(DecisionTreeClassifier(random_state=46429706)...	286	49	NaN	230	0.869	0.908	0.854	0.88	0.888
10	([DecisionTreeRegressor(criterion='friedman_ms...	279	56	NaN	247	0.886	0.959	0.833	0.891	0.954

Campus Placements is a dataset that attempts to predict which students will be placed; with an accuracy score of 0.887, the XGBoost Classifier is the most accurate classifier. This suggests that the XGBoost model performs exceptionally well at accurately determining which students, according to the given attributes, are qualified for placements. Thus, the XGBoost Classifier would probably work best if the main objective is to maximize the accuracy of predicting qualified students for placements.

By converting the trained XGBoost Classifier model into a pickle file and deploying it on a website using Flask in Python, we have successfully integrated machine learning functionality into a web application. This deployment allows students to access the predictive capabilities of the model through a user-friendly web interface. The Flask framework provides a simple and efficient way to create web applications, while the use of the pickle file enables the loading of pre-trained models for real-time predictions.

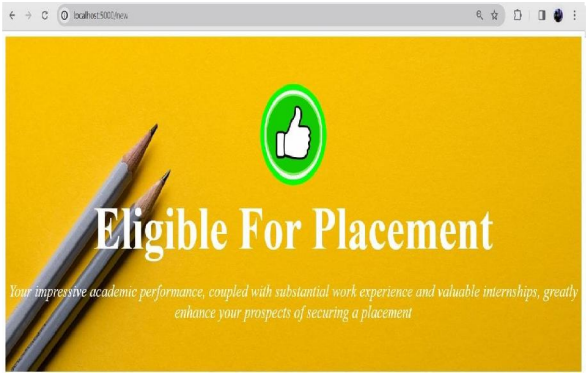
In conclusion, our deployment enables students to interact with the XGBoost model, providing predictions for campus placements eligibility based on input data provided through the website. This integration facilitates the practical application of machine learning in a web-based environment, offering a valuable tool for stakeholders involved in campus placements decision-making processes.

3.5 Deployment:

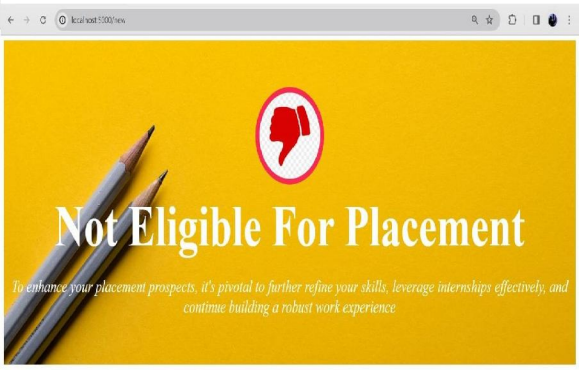


IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

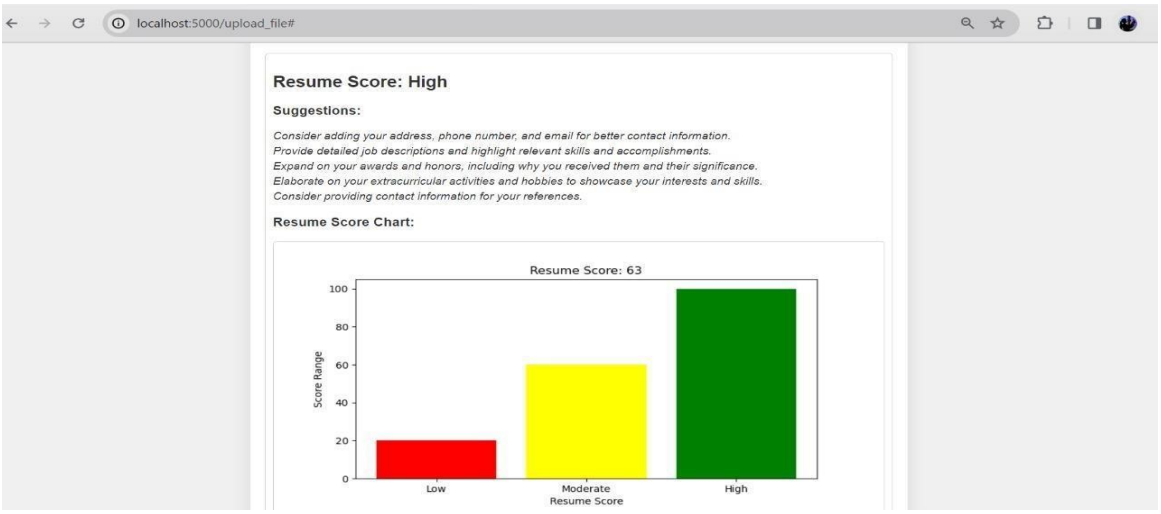
3.6 Results :



Predicts if student is eligible



b. Predicts if student not eligible



c .Whenwe upload our resumes, the results look like the one above.

IV. CONCLUSION

The campus management system project is a comprehensive initiative aimed at optimizing administrative processes and improving the student experience through integrated features. It includes an intuitive online notice board system for efficient communication within the campus community and personalized student registration forms with secure database storage. A key component is the incorporation of machine learning (ML) techniques in a resume score checker for automated resume evaluation. The system also defines eligibility criteria for placement drives and recommends supplementary courses based on student performance data. Emphasizing security, usability, and scalability, the project aims to foster fairness, transparency, and holistic student development while optimizing campus operations for enhanced efficiency and success.

REFERENCES

[1] Title: Design and Implementation of Campus Management System
Link: [https://www.ijstr.org/final-print/mar2016/Design-And-Implementation-Of-Campus-Management System.pdf]
[2] Title: Campus Management System - A Comprehensive Study
Link:[https://www.researchgate.net/publication/322324203_Campus_Management_System__A_Comprehensive_Study
[3] Title: Development of Web-based Campus Management System Link:
[https://ieeexplore.ieee.org/document/6742170]

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, April 2024

[4] Title: Campus Management System: A Case Study of Bahria University Islamabad Link:[https://www.researchgate.net/publication/332217965_Campus_Management_System_A_Case_Study_of_Bahria_University_Islamabad]