STUDENTS PLACEMENT PREDICTION

A PROJECT REPORT

Submitted by

Abino Amriy E S - 962319104004

Amal Krishna V S - 962319104015

Deepak V - 962319104036

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY:CHENNAI 600 025

MAY & 2023

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"STUDENTS PLACEMENT PREDICTION"** is the bonafide work of **Abino Amriy E S (962319104004) , Amal Krishna V S (962319104015) , Deepak V (962319104015)** who carried out the project work under my supervision.

SIGNATURE

Dr. P. M. SIVA RAJA,
ASSOCIATE PROFESSOR,
HEAD OF THE DEPARTMENT,
COMPUTER SCIENCE AND ENGINEERING,
AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY,
AMRITAGIRI,
ERACHAKULAM - 629901.

SIGNATURE

Mrs. R. P. SUMITHRA,
ASSISTANT PROFESSOR,
SUPERVISOR,
COMPUTER SCIENCE AND ENGINEERING,
AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY,
AMRITAGIRI,
ERACHAKULAM - 629901.

Submitted for the B.E. Project viva-voce held on ……..
at **AMRITA COLLEGE OF ENGINEERING AND TECHNOLOGY**.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost we would like to express our sincere gratitude to our respected Founder Amma, **Mata Amritanandamayi Devi** and Chairman **Mr. K S Ramasubban IAS (Retd)** for their blessings and grace in making our project great success.

We would like to place a record with deep sense of gratitude to our Honorable Academic Director and Principal **Dr. T Kannan** for having given us the opportunity to pursue B.E., course in this prestigious institution.

We would like to express our sincere thanks to our beloved Head of the Department **Dr.P M Siva Raja**, for creating a supportive and a model environment for us to work.

We would like to express our sincere thanks to our Project Co-Ordinator, **Mrs. S. Vidhya** for building up our innovative skills

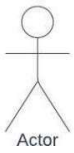
We wish to express our sincere sense of gratitude to our guide **Mrs. G Prem Nisha** for the kind encouragement and moral support, who has been a constant source of inspiration and enabled us to complete our project successfully.

ABSTRACT

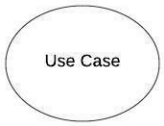
The purpose of this project is to build a model that predicts the placement status of engineering students. The dataset used for this project contains various attributes such as academic performance, technical skills, aptitude test scores, and communication skills of engineering students. The target variable is a binary variable indicating whether a student has been placed or not. The project uses two classification algorithms, namely Logistic Regression and Decision Tree, to build the predictive model. Logistic Regression is a widely used statistical technique for binary classification, while Decision Tree is a tree-based algorithm that makes a sequence of decisions based on the features of the dataset. The project begins with exploratory data analysis, where we analyse the distribution and correlation of the various features with the target variable. We then pre-process the data by handling missing values, encoding categorical variables, and scaling the data. We then split the data into training and testing sets and fit the logistic regression and decision tree models on the training data. We evaluate the performance of the models using various metrics such as accuracy, precision, recall, and F1-score. Finally, we compare the performance of the two models and select the one with the best performance. The selected model is then used to predict the placement status of new engineering students.

LIST OF FIGURES

LIST OF SYMBOLS

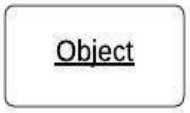


Actor



Use Case

Direction of n data flow



Sequence object

Activation

LIST OF ABBREVIATIONS

ABBREVIATIONS	DESCRIPTION
GPA	Grade Point Average
ML	Machine Learning
KNN	Nearest Neighour
XGBoost	Extreme Gradient Boost
GBM	Gradient Boosting machine
RMSE	Root Man Square Error
UML	Unified Modeling Language
PCA	Principal Component Analysis
ROC	Recover Operating Characteristics

Table of Content

CHAPTER 1

INTRODUCTION

As Taylor walked in the career centre at the beginning of her sophomore year, she found a career coach waiting to welcome her. Taylor and her career coach reviewed a digital dashboard of employability data which highlighted Taylor's major, grade point average (GPA), co-curricular activities, and internships. Then, they began exploring career options that Taylor believed would be personally satisfying and made good use of her public speaking and analytical abilities. Taylor recognized that, regardless of which career path she chose, major, GPA, co-curricular activities, and internships act as signals for employers to determine perceived employability.

Unlike past students, that may have only visited the career office once in their senior year frantically searching for a job opportunity, and often uncertain of their career path, Taylor reviewed her dashboard of employability data on a regular basis, adjusting for changes in her career path. As a sophomore, Taylor can engage in academic and experiential opportunities over the next three years to become more employable. As needed, she can focus on academic performance to improve her GPA, select a major that best aligns with her interests, participate in co-curricular activities such as student organizations, and gain work experience through internships. With career coaches applying analytical rigor to academic and experience information, Taylor can learn how to enhance her perceived employability and ensure that she is on track for employment upon graduation.

This scenario depicts the role data can play in supporting student career outcomes. As competition for jobs increases, students need to identify academic and experiential opportunities to distinguish themselves in the entry-level labor market (Brown & Hesketh, 2004; National Association of Colleges and Employers, 2018; Pinto & Ramalheira, 2017). Moreover, as universities are increasingly held

accountable for student outcomes and continuously amass student data, the data can serve to inform which students are more likely to be employed upon graduation and why.

1.1 ABOUT THE PROJECT

In this project, the development of a predictive model for engineering student placement using decision tree algorithms. The project involves pre-processing the dataset, training and evaluating the models, comparing their performance, and selecting the one with better accuracy. The project also provides insights into the factors that influence the placement of engineering students, which can be used by colleges and recruiters to improve their training programs and recruitment processes. The model developed in this project can be used by colleges and recruiters to identify potential candidates who are more likely to be placed based on their academic and non-academic skills. However, the scope of the project is limited to the dataset used and the algorithms employed. Further research and analysis may be required to improve the accuracy of the model and identify additional factors that influence the placement of engineering students. Overall, the project provides a useful tool for engineering colleges and recruiters to improve their recruitment process and help students achieve success in their careers.

1.2 OUT LINE OF THE PROJECT

Outline of the project which comprises abstracts and chapters. The main objective of the project is to address the problem of increasing competition in the job market for engineering graduates. With the rise of technology and increasing demand for skilled professionals, engineering students face a lot of pressure to perform well in academics and acquire additional skills to stand out from the crowd. However, despite having good academic records and technical skills, some students struggle

to get placed due to various reasons such as lack of communication skills, inadequate exposure to industry-oriented training, and insufficient knowledge of the recruitment process. This project aims to develop a model that can predict the placement status of engineering students based on their academic and non-academic skills, thereby helping colleges and recruiters identify potential candidates who are more likely to be placed. The project also provides insights into the factors that influence the placement of engineering students, which can be used to improve the curriculum and training programs offered by colleges. Overall, the project aims to contribute to the success of engineering students and improve the efficiency of the recruitment process, benefiting both students and recruiters.

CHAPTER 2
LITERATURE SURVEY

Paper1: **Data Pre-processing and Visualizations Using Machine Learning for Student Placement Prediction**

Authors: Chandra Sekhar K; K Santhosh Kumar

Year:2022

Abstract: Student performance during their entire carrier and also a previous academic performance impact the chance of getting a job offer at the end of graduation. Many factors like student technical, analytical, and communication skills are essential to procuring a job. However, our effort is to find how academic skills and scores affect their chances. Machine learning algorithms play a significant role in analysing and predicting the chance of students in placements based on their previous academic outcomes. In this paper, we collected student data from a reputed technical institute. The data set comprises different factors that influence the student chances; these influencing factors are studied and represented using visualizations. On this data set, we tried to analyse the data and draw visualizations and insights before performing or applying machine algorithms to the data. In this paper, our main motto is to analyse and understand the data and perform pre-processing of the data.

Paper2: **A Comparative Study on Machine Learning Algorithms for Predicting the Placement Information of Under Graduate Students**

Authors: Tadi Aravind; Bhimavarapu Sasidhar Reddy; Sai Avinash

Year:2019

Abstract: As Machine Learning (ML) algorithms are becoming popular to solve challenging and interesting real world prediction problems around us, the interest level of student community has been increased in learning the principles of ML and its different algorithms. This includes by implementing the commonly known machine learning algorithms and tests them by solving simple prediction problems around the student community present in educational system. In this line, this paper proposes to solve the student placement prediction problem using linear regression model, K-neighbor regression model, decision tree regression model, XGBoost regression model, gradient boost regression model, light GBM regression model and random tree classifier model. This work is carried out in two phases. The Phase 1 is done on a simple data set and the Phase 2 is done with an extended data set with added additional features about the students. This research work presents the comparative performance analysis of these seven models by implementing them with these two data sets. The performance measurements considered in this study are prediction accuracy and the root mean square error (RMSE).

Paper3: **Campus Placements Prediction & Analysis using Machine Learning**

Authors: Priyanka Shahane

Year:2022

Abstract: Campus placement is an activity of participating, identifying and hiring young talent for internships and entry level positions. Reputation and yearly admissions of the institute invariably depend upon the placements provided by the institute to the students. Therefore, most of the institutions, assiduously, try to boost their placement department in order to improve their organization on a full scale. Any assistance during this specific space can have a good impact on the institute's capability to position it's students. In this study, the target is to analyse student's

placement data of last year and use it to determine the probability of campus placement of the present students. For this we have experimented with four different machine learning algorithms i.e. Logistic Regression, Decision Tree, K Nearest Neighbours and Random Forest.

Paper4: **Data mining approach for predicting student and institution's placement percentage.**

Authors: Ashok M V; Apoorva A

Year:2016

Abstract: Placement of students is one of the very important activities in educational institutions. Admission and reputation of institutions mainly depends on placements. Hence all institutions strive to strengthen placement department. In this study, the objective is to analyse previous year's student's historical data and predict placement chance of the current students and the percentage placement chance of the institution. A model is proposed along with an algorithm to predict the placement chance of students. Data pertaining to the study were collected form the same institution for which the placement chance prediction and percentage placement need to be found from 2006 to 2015. Data collected is divided into historic data form 2016 to 1014 and test data i.e, 2014; 2016 data is considered as current data. Suitable data pre-processing methods are applied. Students having better chance of placement are characterized as good if not bad. This proposed model is compared with other classification algorithms such as Naïve bayes, Decision tree, and Neural network with respect to accuracy, precision and recall. From the results obtained it is found that the proposed algorithm predicts better in comparison with other algorithms.

Paper5: **College Kart and KNN Algorithm based Placement Prediction**

Authors: Nitesh Kumar Sharma; Aniket Kumar Singh; Shubham Salvi

Year:2021

Abstract: One of the most significant activities in an educational institution is student placement. Also, the admission and the institution's credibility are largely dependent on placement. This research work is designed for an institute purpose which develops a system that predicts the possibility of students having chances to be placed in any company by using a supervised machine learning algorithm, which is K-Nearest Neighbours (k -NN) with an accuracy of 87%. The proposed methodology not only forecasts but also makes recommendations to students based on the prediction. Besides, the developed system has a portal that helps students to enlist and buy or sell the stuffs among the colleagues. These stuffs are the things which are required by the students during their academics. Also, the developed system has a platform where students can enlist their skills with evidence such as their project links and websites; any student can search any other students with any skill to form a team to work together on any project.

Paper6: **Student Placement Prediction Using Supervised Machine Learning**

Authors: M. Siva Surya; M.Sathish Kumar; D. Gandhimathi

Year:2022

Abstract: 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 pre-processed appropriately. In terms of accuracy, these proposed models were tested to other classic classification algorithms. According to the results, the proposed technique surpasses the other algorithms by a massive margin

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for the project of engineering student placement prediction may involve traditional methods such as manual analysis of student records and historical data to predict their placement status. This process can be time-consuming and prone to human errors. In the absence of a predictive model, recruiters may have to rely on conventional methods such as interviews and assessments to evaluate a candidate's suitability for a job. This process can be subjective and may not accurately reflect a candidate's potential. Furthermore, educational institutions may not have access to real-time data and insights to identify areas of improvement in their curriculum and provide personalized support to students who are at risk of not being placed. Overall, the existing system may not be efficient, accurate, or scalable in meeting the requirements of the industry and educational institutions. The implementation of a predictive model can significantly improve the placement process and help both recruiters and educational institutions make informed decisions based on data-driven insights.

DISADVANTAGE OF EXISTING SYSTEM

The existing methods for predicting the placement status of engineering students have several disadvantages, including:

1. Time-consuming: Traditional methods of manually analyzing student records and historical data can be time-consuming and labor-intensive, leading to delays in the decision-making process.

2. Subjective: Interviews and assessments used in the absence of a predictive model can be subjective and may not accurately reflect a candidate's potential.

3. Limited scalability: Manual methods are not scalable and may not be able to handle large volumes of data, leading to inefficiencies and inaccuracies in the prediction process.

4. Lack of real-time data: The existing methods may not provide real-time data and insights, making it difficult for educational institutions to identify areas of improvement in their curriculum and provide personalized support to students who are at risk of not being placed.

5. Prone to errors: Manual methods are prone to errors, as they rely on human judgment and analysis, leading to inaccuracies in the prediction process.

Overall, the existing methods for predicting the placement status of engineering students may not be efficient, accurate, or scalable in meeting the requirements of the industry and educational institutions. The implementation of a predictive model can address these issues and significantly improve the placement process.

3.2 PROPOSED SYSTEM

The proposed system for the project of engineering student placement prediction involves the implementation of a predictive model using the Decision Tree Classifier algorithm. The model will be trained on a dataset of engineering student records, including their academic performance, work experience, and other personal information, to predict their placement status. The proposed system offers several advantages over the existing methods, including:

1. Efficiency: The implementation of a predictive model can significantly reduce the time required for analyzing student records and making placement decisions.

2. Accuracy: The model is trained on a large dataset of student records, providing a more accurate and objective assessment of a student's potential for placement.

3. Scalability: The predictive model can handle large volumes of data, making it more scalable than traditional methods.

4. Real-time data: The model can provide real-time data and insights, enabling educational institutions to identify areas of improvement in their curriculum and provide personalized support to students who are at risk of not being placed.

5. Reduced errors: The predictive model is less prone to errors than manual methods, as it relies on data-driven insights rather than human judgment and analysis.

Overall, the proposed system offers a more efficient, accurate, and scalable approach to predicting the placement status of engineering students. It can significantly improve the placement process and help both recruiters and educational institutions make informed decisions based on datadriven insights.

3.3 REQUIREMENTS SPECIFICATION

3.3.1 Hardware Requirements

Processor	Core2Duo
Speed	2 GHz
Random Access Memory	4 GB
HDD	500 GB
Key Board	Windows Keyboard
Mouse	Three Button Mouse
Monitor	SVG

3.3.2 Software Requirements

Operating System	Windows 10 (32 or 64 bit)
Application Server	Flask Framework
Front End	HTMl,CSS
Packages	numpy, sklearn, Pandas, Matplotlib
Back End	Python

3.4 TECHNOLOGY USED

3.4.1 Front End

- HTML
- CSS

3.4.1.1 HTML

HTML (Hypertext Markup Language) is a standard markup language used for creating and structuring content on the web. It is the foundation of all web pages and enables developers to create and display content such as text, images, videos, and forms in a structured manner. HTML is designed to be easy to read and write by both humans and machines, making it an essential tool for web development. HTML uses a series of tags and attributes to create structured content. Tags are enclosed in angle brackets (<>) and provide instructions to web browsers about how to display content. Attributes are used to provide additional information about a tag, such as the source of an image or the link destination of a hyperlink

3.4.1.2 CSS

CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of web pages written in HTML and XML. CSS enables developers to separate the presentation of web pages from the structure and content, making it easier to create and maintain web pages. CSS works by targeting HTML elements and applying styles to them. Styles can include attributes such as font size, color, and spacing, as well as positioning and layout information. CSS styles are defined in style sheets, which can be included within an HTML document or linked externally.

3.4.2 Backend

Python

3.4.2.1 Python

Python is a high-level, interpreted programming language that is widely used in web development, scientific computing, data analysis, artificial intelligence, and many other areas of software development. Python was created in the late 1980s by Guido van Rossum, and its design philosophy emphasizes code readability and ease of use. Python is known for its simplicity and flexibility, making it a popular choice for beginners and experienced programmers alike. Python code is typically shorter and easier to read than other programming languages, which makes it easy to understand and maintain. Python's syntax is also relatively straightforward, using indentation to denote blocks of code instead of braces or other delimiters.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

4.1.1 System Overview

System design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used to make its operation more effective.

4.2 UNIFIED MODELING LANGUAGE

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artefacts of software system, as well as for business modelling and other non-software systems. The UML represents

a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS: The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.2.1 Use case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.The use case contains mobile nodes, router and sink which perform different functions.

Mobile nodes sending beacon signals to the sink through routers. Routers are used to search the node within a range . Sink is to perform whether file is transferred or not.

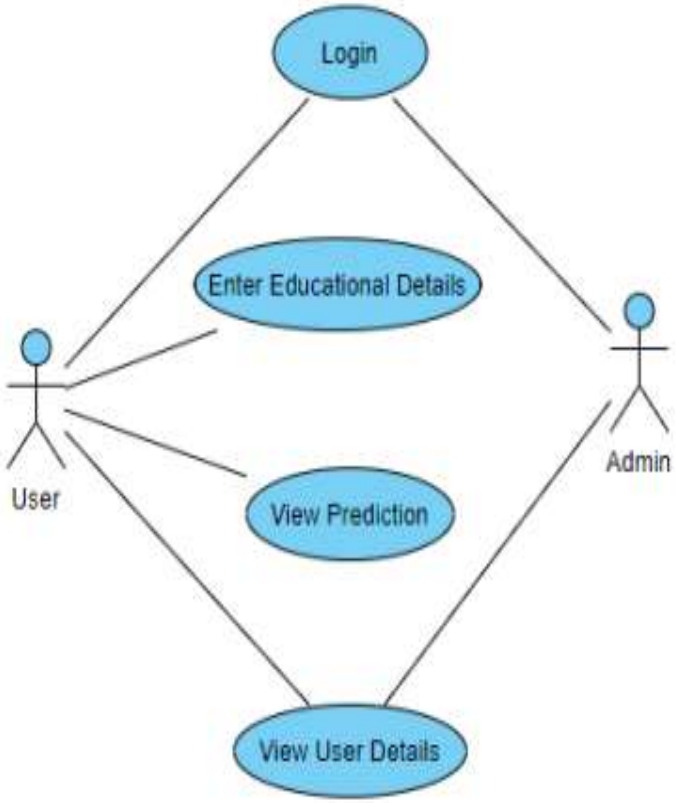


Figure 4.1: Use Case Diagram

4.2.2 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.The name of the class diagram includes mobile nodes, router and sink. The attributes of

the class diagram are IP address , transmit file, destination path and receiving a file. The operation of this diagram that routers transmit the beacon signals to search a node within the range or not. When a node is missing in that particular group are identified and the file is transferred to the destination.

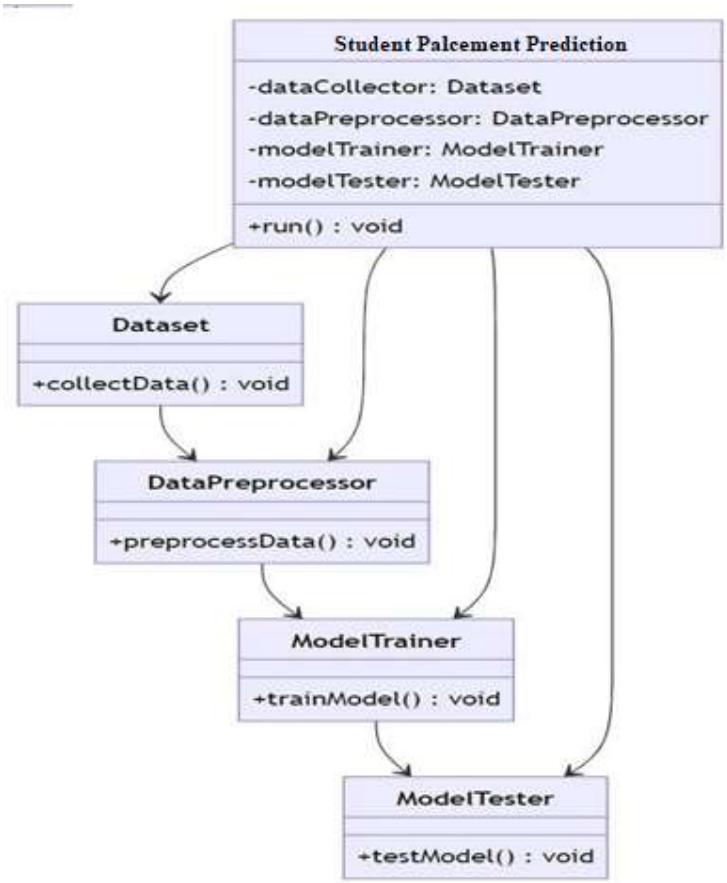


Figure 4.2: Class Diagram

4.2.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.This

diagram indicates how that the sequence of operations is carried out. At the beginning of the diagram , source is located at the router which is represented as a message with browse a file and file transferred to the router. When the router is started off, the beacon signal search the nodes within a range and the file will be transferred to the destination.

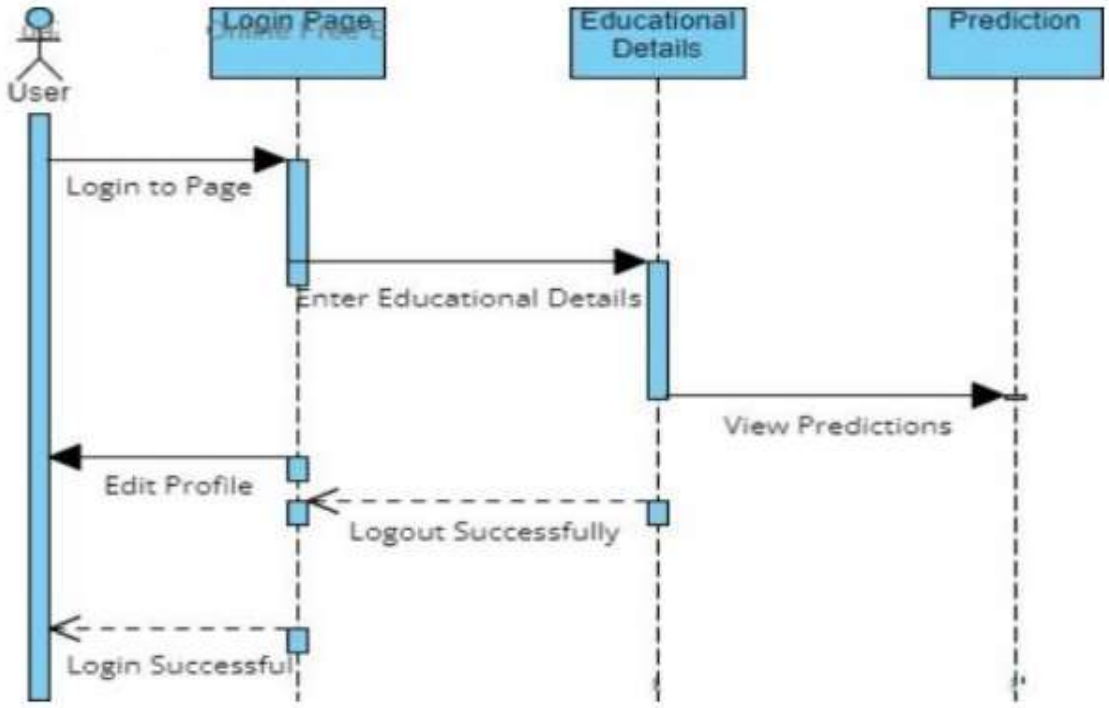


Figure 4.3: Sequence Diagram

4.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

4.3.1 System Study

Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system

analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

1. ECONOMICAL FEASIBILITY
2. TECHNICAL FEASIBILITY
3. SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.4 PRODUCT OVERVIEW DESCRIPTION

The project includes the development of a predictive model for engineering student placement using logistic regression and decision tree algorithms. The project involves preprocessing the dataset, training and evaluating the models, comparing their performance, and selecting the one with better accuracy. The project also provides insights into the factors that influence the placement of engineering students, which can be used by colleges and recruiters to improve their training programs and recruitment processes. The model developed in this project can be used by colleges and recruiters to identify potential candidates who are more likely to be placed based on their academic and nonacademic skills. However, the scope of the project is limited to the dataset used and the algorithms employed. Further research and analysis may be required to improve the accuracy of the model and identify additional factors that influence the placement of engineering students. Overall, the project provides a useful tool for engineering colleges and recruiters to improve their recruitment process and help students achieve success in their careers.

4.5 SYSTEM IMPLEMENTATION

Systems implementation is the construction of the new system and the delivery of that system into production (that is, the day-to-day business or organization operation). Systems implementation is the process of defining how the information system should be built (i.e., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standard (i.e., quality assurance).

4.5.1 System Architecture

System Architecture is a model that describes the structure and behavior of a complex system. It comprises all the components and the overview of the whole system.

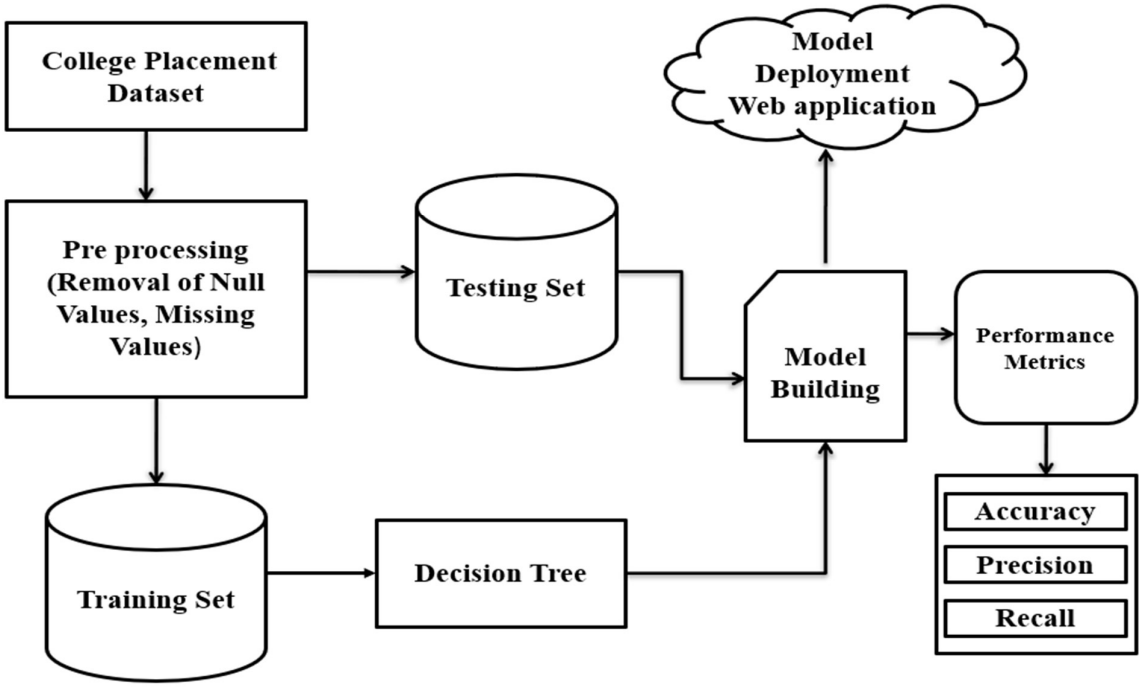


Figure 4.4: Architecture

4.5.2 System Modules

1. Data Collection
2. Data Pre-processing
3. Model Development
4. Model Evaluation

4.5.2.1 Data Collection

The Kaggle College Placement Dataset is a publicly available dataset that contains information on students' academic and personal backgrounds, as well as their placement status after graduation. The dataset was created by a private engineering college in India, and it includes data from students who graduated between 2015 and 2017. The dataset contains a total of 215 records, with each record representing an individual student. The dataset includes several features such as gender, age, secondary school percentage, board of education, degree specialization, work experience, and salary offered. The placement status of each student is also included in the dataset, indicating whether the student was placed or not placed after graduation. Additionally, the dataset includes information on the type of company where the student was placed, such as whether it was a multinational corporation or a startup. The Kaggle College Placement Dataset provides a valuable resource for developing predictive models to predict the placement status of engineering students. The dataset is widely used in research and educational institutions to study the factors that influence the placement of students and to develop effective strategies for improving the placement process.

4.5.2.2 Data Pre-processing

Data pre-processing is a crucial step in any data-driven project, including engineering student placement prediction using Decision Tree Classifier. In this project, we need to pre-process the Kaggle College Placement Dataset to prepare it for use in training the model. The following are the steps involved in data pre-processing:

1. Data Cleaning: The first step in data pre-processing is to clean the data by identifying and handling missing values, duplicate entries, and outliers. We will use various techniques to handle missing data, such as filling the missing values with the mean, median, or mode of the column. We will also identify and remove duplicate entries and handle outliers using statistical techniques.

2. Data Transformation: The next step is to transform the data into a format suitable for analysis. This involves converting categorical variables into numerical ones using techniques such as one hot encoding or label encoding. We may also need to scale or normalize numerical variables to ensure that they are on the same scale.

3. Feature Selection: Feature selection involves identifying the most relevant features in the dataset for predicting the placement status of engineering students. This helps to reduce the dimensionality of the dataset and improve the performance of the model. We will use various techniques such as correlation analysis, feature importance analysis, and principal component analysis (PCA) to identify the most relevant features.

4. Data Splitting: Finally, we will split the pre-processed dataset into training and testing sets. The training set will be used to train the model, while the testing set will be used to evaluate its performance. We will use various techniques such as cross-validation to ensure that the model is robust and accurate.

Overall, data pre-processing is a critical step in this project, as it ensures that the data is cleaned, transformed, and prepared for use in training the predictive model. This step helps to improve the accuracy and performance of the model and ensures that it can be used to predict the placement status of engineering students with high accuracy.

4.5.2.3 Model Development

Model development is the process of building a predictive model that can accurately predict the placement status of engineering students based on their academic and personal backgrounds. In this project, we will use a Decision Tree Classifier algorithm to develop the model. The Decision Tree Classifier algorithm is a popular machine learning algorithm used for classification tasks. It works by recursively partitioning the dataset into smaller subsets based on the values of the features until the subsets are homogeneous. The algorithm uses various metrics, such as entropy and information gain, to determine the optimal split at each node of the tree. In our project, we will use the pre-processed dataset to train the Decision Tree Classifier model. The model will learn to identify the patterns and relationships between the features and the placement status of engineering students. Once the model is trained, we will evaluate its performance on the testing set to determine its accuracy.

We will use various techniques such as hyperparameter tuning, cross-validation, and feature selection to improve the performance of the model. These techniques help to ensure that the model is accurate, robust, and can generalize well to new data. Overall, model development is a critical step in this project, as it determines the accuracy and effectiveness of the predictive model. By building a

robust and accurate model, we can help engineering colleges to predict the placement status of their students and improve the placement process.

4.5.2.4 Model Evaluation

Model training and evaluation is the process of training the Decision Tree Classifier model using the pre-processed dataset and evaluating its performance on the testing set. In this project, we will use various techniques to ensure that the model is accurate and robust. The following are the steps involved in model training and evaluation:

1. Splitting the Dataset: We will split the pre-processed dataset into training and testing sets using various techniques such as cross-validation. This ensures that the model is trained on a subset of the data and evaluated on a separate subset of the data.

2. Training the Model: We will train the Decision Tree Classifier model using the training set. During training, the model learns to identify the patterns and relationships between the features and the placement status of engineering students.

3. Hyperparameter Tuning: We will use various techniques such as grid search or random search to identify the optimal hyperparameters for the model. This helps to improve the performance of the model and ensure that it is accurate and robust.

4. Evaluating the Model: We will evaluate the performance of the model on the testing set using various metrics such as accuracy, precision, recall, F1-score, and ROC curve. This helps us to determine the effectiveness of the model and its ability to predict the placement status of engineering students accurately.

5. Feature Importance Analysis: We will analyse the importance of each feature in the dataset for predicting the placement status of engineering students. This helps us to identify the most relevant features and improve the performance of the model.

Overall, model training and evaluation are crucial steps in this project, as they help us to build an accurate and effective predictive model. By using various techniques to optimize the model and evaluate its performance, we can ensure that the model can accurately predict the placement status of engineering students and improve the placement process.

CHAPTER 5

TESTING

5.1 INTRODUCTION

Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases makes it shine. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an undiscovered error.

5.2 TESTING TYPES

5.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.2.3 User Acceptance Testing

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

- Input screen design.
- Output screen design.
- On-line message to guide the user.
- Format of the ad-hoc reports and other outputs.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.4 Validation Testing

This provides the final assurance that the software meets all functional, behavioural and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in which the user expects.

5.2.5 White Box Testing

White-box testing, sometimes called glass-box is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercises all logical decisions on their true and false sides.
- Executes all loops at their boundaries and within their operational bounds and Exercise internal data structures to assure their validity.

5.2.6 Black Box Testing

Black Box testing focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to White Box testing. Rather it is a complementary approach that is likely to uncover a different class of errors than White
Box methods. Black Box testing attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external data base access.

- Performance errors.
- Initialization and Termination errors.

Unlike White-Box testing, which is performed early on the testing process black-box testing tends to be applied during later stages of testing because black-box testing which is purposely disregards control structures attention is focused on the information domain.

5.2.7 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centred on the following items:

Valid Input	:identified classes of valid input must be accepted.
Invalid Input	:identified classes of invalid input must be rejected.
Functions	:identified functions must be exercised.
Output	:identified classes of application outputs must be exercised.
Systems/Procedures	:interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.3 TESTING STRATERGIES

A Strategy for software testing integrates software test cases into a series of well planned steps that result in the successful construction of software. Software testing is a broader topic for what is referred to as Verification and Validation. Verification refers to the set of activities that ensure that the software correctly implements a specific function. Validation refers to the set of activities that ensure that the software that has been built is traceable to customers requirements.

5.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.4.1 Approach to Testing

Field testing will be performed manually and functional tests will be written in detail. **Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format.

- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.4.2 Test Data

The test cases are classified into positive and negative cases. The test cases are depicted below

- Verify if a user can log in with a valid username and a password - HTTP Status Code 200 -

  Successful login
- Verify if a user can log in with a invalid username and a password - HTTP Status Code 401

  - Throw error message
- Verify if a user can login with no password entered - HTTP Status Code 401 -

  Prompt for password. Please enter Password”
- Fetch Attendance with Invalid JWT - HTTP Status Code 400 - Throw Error Message
- Update Attendance Data with Invalid Role - HTTP Status Code 400 - User not permitted

5.4.3 Performance Evaluation

INPUT DESIGN:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design

of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be depicted.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

In conclusion, the engineering student placement prediction project using Decision Tree Classifier is an effective way to predict the placement status of engineering students based on their academic and personal information. The project involved data collection, data pre-processing, model development, model training, model evaluation, and model deployment. The project utilized the Kaggle College Placement Dataset, which contains information on the academic and personal background of students, such as their gender, board of education, work experience, and salary expectations. The dataset was pre-processed to remove missing values, encode categorical variables, and scale the numerical variables. The Decision Tree Classifier model was trained on the pre-processed dataset and evaluated using various performance metrics, such as accuracy, precision, recall, F1-score, and ROC curve. The model achieved high accuracy and precision, indicating its effectiveness in predicting the placement status of engineering students. Result analysis techniques such as confusion matrix, feature importance analysis, ROC curve, and error analysis were used to analyze the results of the model and identify areas for improvement. These techniques helped to improve the accuracy and effectiveness of the model, ensuring that it could accurately predict the placement status of engineering students. Overall, the project demonstrated that the Decision Tree Classifier model is an effective way to predict the placement status of engineering students and can be used by universities and employers to streamline the placement process and improve the placement outcomes for engineering students.

6.2 FUTURE ENHANCEMENT

There are several areas for future work in the engineering student placement prediction project using Decision Tree Classifier. Some of these areas include:

1. Incorporating more data: The project used a limited set of data to predict the placement status of engineering students. In future work, additional data, such as internships, projects, and extracurricular activities, can be incorporated into the model to improve its accuracy.

2. Using other machine learning algorithms: While the Decision Tree Classifier model was effective in predicting the placement status of engineering students, other machine learning algorithms such as Random Forest, XGBoost, and Neural Networks can be explored to improve the accuracy of the model.

3. Conducting a longitudinal study: The project predicted the placement status of engineering students based on their academic and personal information at a single point in time. In future work, a longitudinal study can be conducted to track the career outcomes of engineering students over time, and identify the factors that contribute to their success.

4. Developing a web-based application: The project can be extended to develop a web-based application that universities and employers can use to predict the placement status of engineering students. The application can be integrated with a student database and updated regularly to provide accurate predictions.

Overall, future work in the engineering student placement prediction project can help to improve the accuracy and effectiveness of the model and provide valuable insights into the placement outcomes of engineering students.

.

Appendices

Appendix 1

SCREEN SHOTS

Home Page

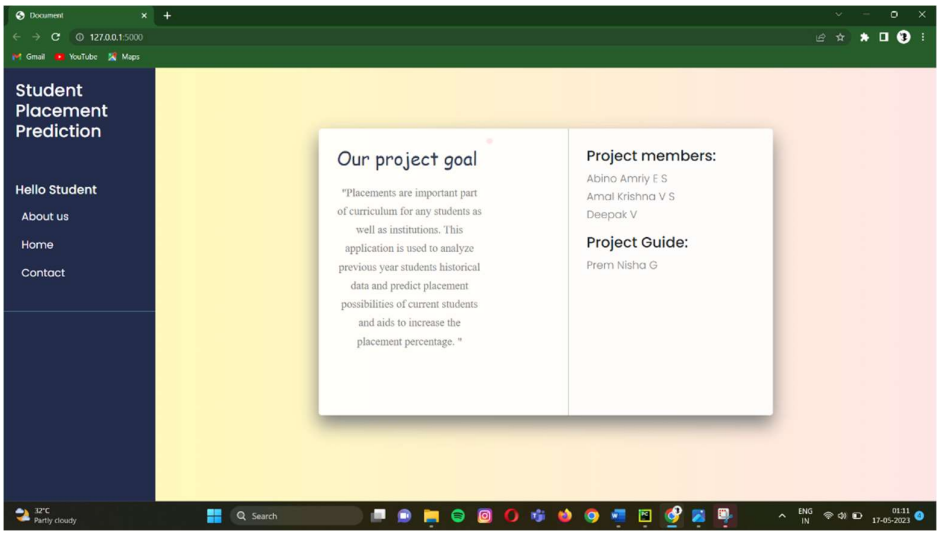


Figure A.1.1: Home Page

Personal Details page

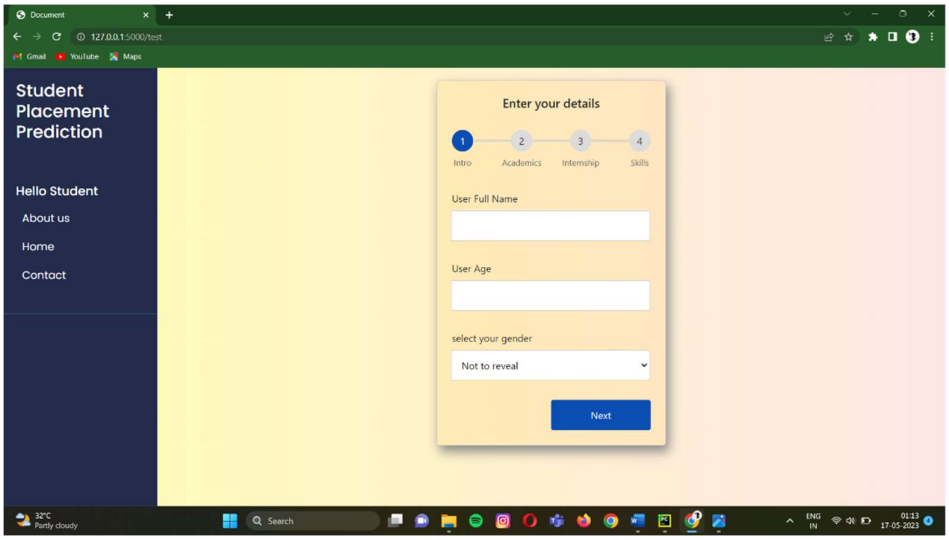


Figure A.1.2: Personal Detail Page

Academic Details page

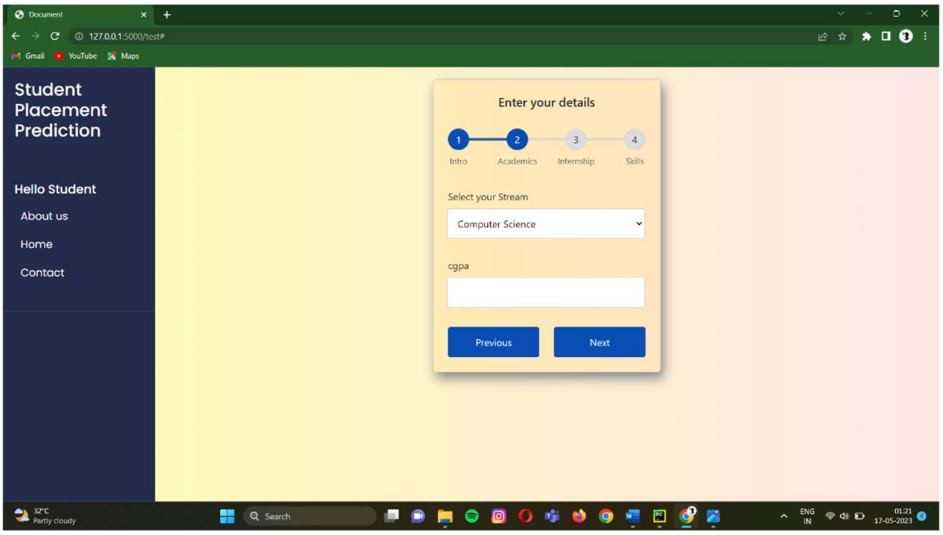


Figure A.1.3: Academic Details Page

Internship Details page

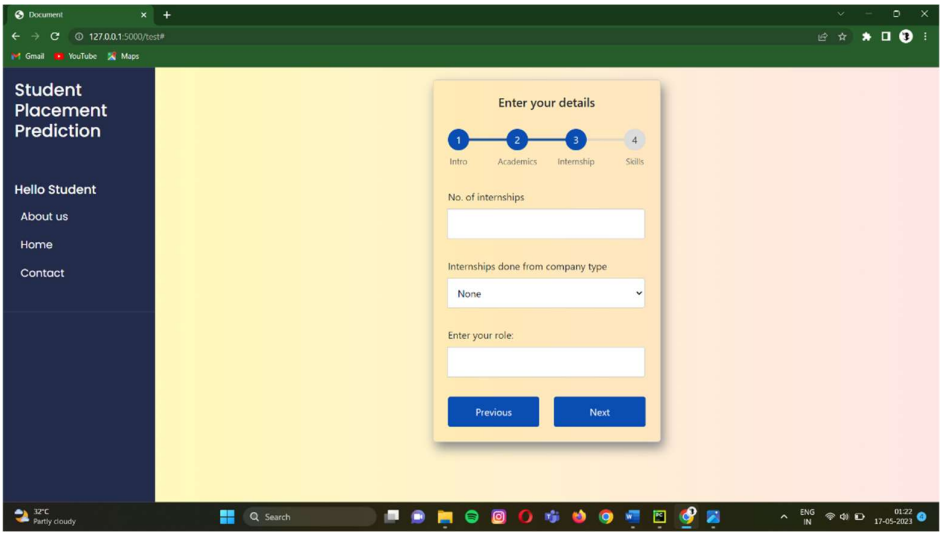


Figure A.1.4 Intership Detail Page

Skill Details page

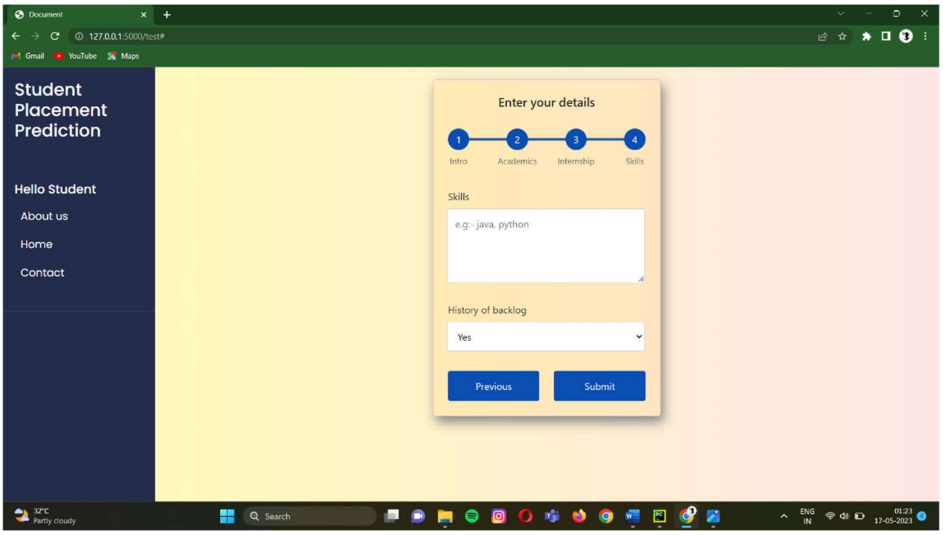


Figure A.1.5: Skill Detail Page

Result Page:

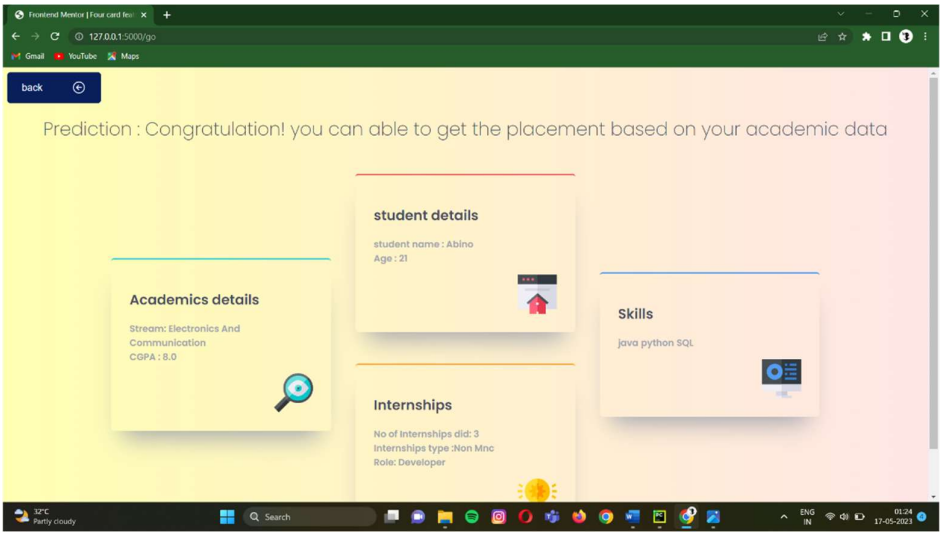


Figure A.1.6: Recent Detail Page

APPENDIX 2

USER MANUAL

1. Open Browser.

2. Open the Web App. (http://127.0.0.1:5000/)

3. Provide input

The input data for your Student Placement Prediction model may include various features related to a student's academic and other relevant information. Here are some examples of the types of input data you might include:

- Personal information: This includes the student's name, gender, age, and contact information.
- Academic records: This includes the student's academic performance, such as their GPA, standardized test scores, and the courses they have taken.
- Internship and job experiences: This includes information about the internships and job experiences the student has had, including the company or organization they worked for, their role, and the duration of the experience.
- Skills and qualifications: This includes any skills or qualifications the student has, such as certifications or language proficiency.
- Extracurricular activities: This includes information about the student's participation in extracurricular activities, such as sports, clubs, or volunteering.
- Demographic information: This includes information about the student's background, such as their race, ethnicity, and socioeconomic status.

4. Machine learning model processes the input

5. Gets output.

Appendix 3

SAMPLE CODING

BACKEND

application.py

```
from flask import Flask,render_template,redirect,url_for,request
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
# creating a classifier using sklearn
from sklearn.linear_model import LogisticRegression
application=Flask(__name__)
application.config['SECRET_KEY']='3483dd03f2cb409aba821cd2a27c
64b3'
def calulate(lst):
dataset = pd.read_csv('newdata.csv')
# selecting the features and labels
X = dataset.iloc[:, :-1].values
Y = dataset.iloc[:, -1].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.2)
clf=LogisticRegression(random_state=0,
solver='lbfgs',max_iter=1000).fit(X_train,Y_train)
# printing the acc
clf.score(X_test, Y_test)
# predicting for random value
Y_pred=clf.predict([[lst[0], lst[1], lst[2], lst[3], lst[4], lst[5], lst[6]]])
```

```
return Y_pred[0]
@application.route('/test')
def home():
return render_template('index.html')
@application.route('/')
def aboutus():
return render_template('/aboutus.html')
@application.route('/go',methods=['GET','POST'])
def go():
skl=request.form['skill']
lst=[int(request.form['age']),int(request.form['gender']),int(request.form[
'stream']),int(request.form['intern_no'])
,float(request.form['cgpa']),0,int(request.form['backlog'])]
if len(skl)!=0:
        skills=request.form['skill'].split(",")
else:
        skills='no skill found'
detail={
         'username': request.form['username'],'intern_type':request.form  ['intern_type'],
         'intern_details':request.form['intern_details'],'skills':skills,
         'age':lst[0],'cgpa':lst[4],'intern_no':lst[3]
}
stream={
        "0":'Electronics And Communication',
        "1":'Computer Science',
       "2":'Information Technology',
```

```
        "3":'Mechanical',
        "4":'Electrical',
        "5":'Civil'
}
detail['stream']=stream[str(lst[2])]
if lst[1]==1:
        detail['gender']='Female'
else:
        detail['gender']='Male'
if request.form['backlog']==1:
        return render_template('/demo.html',detail=detail,lst=lst)
else:
        result=calulate(lst)
        if result==1:
                mssg='Congratulation! you can able to get the placement based on your
                        academic data'
        else:
                mssg='Sorry to say but you need to improve yourself'
        return render_template('/result.html',detail=detail,mssg=mssg)
@application.route('/result',methods=['GET','POST'])
def result():
lst=[int(request.form['age']),int(request.form['gender']),int(request.form['stream']),
int(request.form['intern'])
,float(request.form['cgpa']),0,int(request.form['backlog'])]
result=calulate(lst)
if result==1:
```

```
        string="Congratulation! you can able to get the placement based
on your academic data"
else:
        string="Sorry to say but you need to improve yourself"
return render_template('result.html',mssg=string)
if __name__ == '__main__':
application.run(debug=True)
```

Front End

Home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94
WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
<link rel="stylesheet"href=https://fonts.googleapis.com/css?family=Poppins:
300,400,500,600,700 >
 <style>
    body {
   font-family: 'Poppins', sans-serif;
   background: #fafafa;
```

```
}
p {
    font-family: 'Poppins', sans-serif;
    font-size: 1.1em;
    font-weight: 300;
    line-height: 1.7em;
    color: #999;
}
a,a:hover,a:focus {
    color: inherit;
    text-decoration: none;
    transition: all 0.3s;
}
.navbar {
    padding: 15px 10px;
    background: #fff;
    border: none;
    border-radius: 0;
    margin-bottom: 40px;
    box-shadow: 1px 1px 3px rgba(0, 0, 0, 0.1);
}
.navbar-btn {
    box-shadow: none;
    outline: none !important;
    border: none;
}
```

```
.line {
    width: 100%;
    height: 1px;
    border-bottom: 1px dashed #ddd;
    margin: 40px 0;
}
.wrapper {
    display: flex;
    width: 100%;
    align-items: stretch;
}
#sidebar {
    min-width: 250px;
    max-width: 250px;
    background: #242c4b;
    color: #fff;
    transition: all 0.3s;
}
#sidebar.active {
    margin-left: -250px;
}
#sidebar .sidebar-header {
    padding: 20px;
    background: #242c4b;
}
```

```
#sidebar ul.components {
    padding: 20px 0;
    border-bottom: 1px solid #47748b;
}
#sidebar ul p {
    color: #fff;
    padding: 10px;
}
#sidebar ul li a {
    padding: 10px;
    font-size: 1.1em;
    display: block;
}
#sidebar ul li a:hover {
    color: #7386D5;
    background: #fff;
}
#sidebar ul li.active>a,
a[aria-expanded="true"] {
    color: #fff;
    background: #6d7fcc;
}
a[data-toggle="collapse"] {
    position: relative;
}
```

```
.dropdown-toggle::after {
    display: block;
    position: absolute;
    top: 50%;
    right: 20px;
    transform: translateY(-50%);
}
ul ul a {
    font-size: 0.9em !important;
    padding-left: 30px !important;
    background: #6d7fcc;
}
ul.CTAs {
    padding: 20px;
}
ul.CTAs a {
    text-align: center;
    font-size: 0.9em !important;
    display: block;
    border-radius: 5px;
    margin-bottom: 5px;
}
a.download {
    background: #fff;
    color: #7386D5;
}
```

```
a.article,a.article:hover {
    background: #121c44 !important;
    color: #fff !important;
}
#content {
    width: 100%;
    padding: 20px;
    min-height: 100vh;
    transition: all 0.3s;
    background: url('https://encrypted-tbn0.gstatic.com/images?q=tbn :
ANd9GcROoPL_RT4wwj1Ud6SEHqqWXcymrfbimcECSQ&usqp=CAU');
    background-repeat: no-repeat  ;
    background-position: bottom right;
    background-size: 40% 40%;
}
@media (max-width: 768px) {
    #sidebar {
        margin-left: -250px;
    }
    #sidebar.active {
        margin-left: 0;
    }
    #sidebarCollapse span {
        display: none;
    }
}
```

```
 </style>
</head>
<body>
   <div class="wrapper">
      <!-- Sidebar  -->
      <nav id="sidebar">
         <div class="sidebar-header">
            <h3>Student Placement Prediction</h3>
         </div>
         <ul class="list-unstyled components">
            <div class="sidebar-header">
               <h5 margin-left="10">Hello Student</h5>
            <li>
               <a href="#">About us</a>
            </li>
            <li>
               <a href="#">Give test</a>
            </li>
            <li>
               <a href="#">Contact</a>
            </li>
            </div>
         </ul>
      </nav>

      <!-- Page Content  -->
```

```
<div id="content">
   <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container-fluid">
          <h5>  Welcome to our site</h5>
      </div>
   </nav>
   <div class="card-body">
      <h3>Enter your details:</h3><br>
      <form action="{{ url_for('result') }}" method="POST">
       <div class="row">
         <div class="col">
            <div class="form-group">
              <label for="stream" >Select your Stream</label>
              <select class="form-control" id="stream" name='stream' required>
                <option value="0">Electronics And Communication</option>
                <option value="1">Computer Science</option>
                <option value="2">Information Technology</option>
                <option value="3">Mechanical</option>
                <option value="4">Electrical</option>
                <option value="5">Civil</option>
              </select>
            </div>
         </div>
         <div class="col">
          <div class="form-group">
              <label for="gender">Select your gender</label>
```

```
        <select class="form-control" id="gender" name="gender" required>
          <option value="1" >Female</option>
          <option value="0">Male</option>
          <option value="1">not to reveal</option>
        </select>
      </div>
    </div>
    <div class="col">
     <div class="form-group">
        <label for="age">Enter your age:</label>
       <input type="text" class="form-control" id="age" name='age'
            required >
      </div>
    </div>
  </div><br>
   <div class="row">
    <div class="col">
     <div class="form-group">
        <label for="internship">No of internships you did:</label>
        <input type="text" class="form-control" id="internship"
    name="internship" placeholder="enter the integer value only" required>
      </div>
    </div>
    <div class="col">
     <div class="form-group">
        <label for="cgpa">Enter your cgpa:</label>
  <input type="text" class="form-control" id="cgpa" name='cgpa' required>
```

```
                </div>
              </div>
              <div class="col">
               <div class="form-group">
                   <label for="backlog">History of backlog?</label>
        <select class="form-control" id="backlog" name="backlog" required>
                    <option value="1">Yes</option>
                    <option value="0">No</option>
                   </select>
                  </div>
               </div>
              </div>
               </div>
               <center><input type="Submit" value="Submit" class="btn btn-primary w-25"></center>
            </form>
</div>
</body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js" integrity="sha3847+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB" crossorigin="anonymous"></script>
```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js" integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13" crossorigin="anonymous"></script>

</html>

References

1. 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

2. "A recommendation system using machine learning", *In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. I-5, 2017.

3. Ardita Permata Alfiani and Febriana Ayu Wulandari, "Mapping student's performance based on data mining approach (a case study)", *Agriculture and Agricultural Science Procedia*, vol. 3, pp. 173-177, 2015.

4. Gilbert Noah, "Predicting Success: an Application of Data Mining Techniques to Student Outcomes", *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, vol. 7.2, pp. 1-20, 2017.

5. M.B Sheetal and Bakare Savita, "Prediction of Campus Placement Using Data Mining Algorithm-Fuzzy logic and K nearest neighbor", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5.6.

6. S. Ahmed, A. Zade, S. Gore, P. Gaikwad and M. Kolhal, "Performance Based Placement Prediction System", *IJARIIE-ISSN (O)*, vol. 4, no. 3, pp. 2395-4396, 2018.

7. B Mangasuli Sheetal and Prof. Savita Bakare, "Prediction of Campus Placement Using Data Mining AlgorithmFuzzy logic and K nearest neighbour", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 6, June 2016.

8. Ishizue Ryosuke et al., "Student placement and skill ranking predictors for programming classes using class attitude psychological scales and code

metrics", *Research and Practice in Technology Enhanced Learning*, vol. 13.1, pp. 1-20, 2018.

9. K. Manikandan, S. Sivakumar and M. Ashokvel, "A Classification Model for Predicting Campus Placement performance Class using Data Mining Technique", *International Journal of Advance Research in Science and Engineering*.

10. G. Vadivu and K. Sornalakshmi, "Applying Machine Learning Algorithms for Student Employability Prediction Using R", *Int. J. Pharm. Sci. Rev. Res.*, vol. 43, no. 1, pp. 38-41, 2017.

11. Z. R. B, J. Salvador-meneses and J. Garcia-rodriguez, Preprocessing to Improve Categorical, Springer International Publishing, vol. 1, 2018.

12. M. Nirmala, "Student Academic Performance Prediction under Various Machine Learning Classification Algorithms", *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 11, pp. 221-237, 2021.

13. K. Sumathi, S. Kannan and K. Nagarajan, "Data Mining: Analysis of student database using Classification Techniques", *Int. J. Comput. Appl.*, vol. 141, no. 8, pp. 22-27, 2016.

14. S. Garcia, J. Luengo and F. Herrera, "Data Preprocessing in Data Mining", *Intelligent Systems Reference Library*, vol. 10, 2015.