# **TITLE** PAGE

**DESIGN AND IMPLEMENTATION OF AN EXPERT SYSTEM FOR PREDICTING STUDENTS’ ACADEMIC PERFORMANCE (A CASE STUDY OF COMPUTER SCIENCE DEPARTMENT FEDERAL POLYTECHNIC BIDA)**

**SUBMTTED BY**

**GROUP 68**

**NAME MAT NO LEVEL**

**CHUKWU GODWIN.C. 2020/136381CS HND II**

**SALIHU MULIKAT OYIZA 2020/139268CS HND II**

**SHAABA NURUDEEN 2020/138274CS HND II**

**SUBMITTED TO**

**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**THE FEDERAL POLYTECHNIC BIDA**

**P.M.B. 55**

**BIDA, NIGER STATE**

**SUPERVISED BY**

**MRS. AHMAD .K. HAUWA**

**OCTOBER, 2022.**

# CERTIFICATION

This is to certify that this project on the Design and Implementation of an Expert System for Predicting Student Academic Performance is an original work carried out by Chukwu Godwin. C. with Matric: No 2020/136191CS, Salihu Mulikat Oyiza with Matric: No 2020/139268CS and Shaaba Nurudeen, Matric: No 2020/13827CS. And it has been toughly and duly read under the guidance and supervision ofMrs. Ahmad K.H. and found of acceptance by the Department of Computer Science for the award of Higher National Diploma (HND).

**MRS. AHMAD .K. HAUWA ...……………………**

***Project Supervisor Signature/Date***

**MR. CHARLES ENYINAYA ………………………**

***Project coordinator Signature/Date***

**MAL. ADAMU MUHAMMED ………………………..**

***Head of Department Signature/Date***

***Computer Science***

**…………………………….. ……………………….**

***External Supervisor Signature/Date***

# DEDICATION

This project is dedicated to God Almighty and our parents for their support.

# ACKNOWLEDGEMENT

Our sincere and greatest appreciation goes to God Almighty `for always having his blessing on us. We are also grateful to the Federal Polytechnic Bida for giving us a great platform through our project supervisor **MRS. KULUWA.H. AHMAD** for exploring our software developing skills during the making of this project. We express our heartfelt gratitude to **MAL. ADAMU MUHAMMED** our Head of Department for rendering all possible help and support during the development, implementation and presentation of the project.

We are also thankful to our project coordinator **MR. CHARLES ENYINAYA** our project coordinator and all other staff members of Computer Science Department and Registry Office for their constructive and helpful input. Our profound gratitude goes to our course mates and others who helped us directly and indirectly in solving problems and making our software project more efficient and working.

# ABSTRACT

The application of machine learning techniques in predicting students’ performance, based on their background and their in-term performance has proved to be a helpful tool for foreseeing poor and good performances in various levels of education. The major problem institutions face is the systematic monitoring of students’ academic progress in their course of study, which is influence by many factors. This research work assesses the potential of predicting student academic performance with an expert system in the Federal Polytechnic Bida. The expert system is develop using python programming language and it accepts data about a student in various category such as demographic background, attitudes study habits, etc. and through the partial activation of multiple rules, predicts an outcome for each student. The students’ data was collected via a user-friendly interface and analysed using fuzzy logic algorithm for prediction. The prediction analysis for students’ can only be viewed by the administrator (instructor) as it requires logins for forecasting students’ possible outcomes.

# LIST OF TABLES

Table 4.2.1 User Selection 23

Table 4.2.2 Administrator login 24

Table 4.2.3 User Registration Page 25

# LIST OF FIGURES

Fig 4.1 Flow Chart of the Student Academic Performance Prediction System 20

Fig 4.1.2 Use case diagram, for the different users of the system and their functions 21

Fig 4.1.3 Represents the data flow between datasets, user and prediction model. 22

Fig 4.2.1 User Selection Interface 23

Fig.4.2.2 Administrator Login Page 24

Fig. 4.2.3: User Registration Page 25

Fig 4.2.4 Prediction Page 26

Fig. 4.3.1 Invalid Login Credentials 27

Fig 4.3.2 Response Submitted 28

Fig 4.3.3 Prediction Record stored in database 28

# ABBREVIATIONS

**EDM**: Educational Data Mining

**ML**: Machine Learning

**ANN**: Artificial Neural Network

**LMT**: Logistic Model Tree

**SVM**: Support Vector Machine

**SRM**: Structural Risk Minimization

**SGD**: Stochastic Gradient Descent

**DFD:** Data Flow Diagram

**GUI**: Graphical User Interface

# TABLE OF CONTENTS

# 

[TITLE PAGE i](#_Toc118222222)

[CERTIFICATION ii](#_Toc118222223)

[DEDICATION iii](#_Toc118222224)

[ACKNOWLEDGEMENT iv](#_Toc118222225)

[ABSTRACT v](#_Toc118222226)

[LIST OF TABLES vi](#_Toc118222227)

[LIST OF FIGURES vii](#_Toc118222228)

[ABBREVIATIONS viii](#_Toc118222229)

[TABLE OF CONTENTS ix](#_Toc118222230)

[CHAPTER ONE 1](#_Toc118222231)

[1.0 INTRODUCTION 1](#_Toc118222232)

[1.1 BACKGROUND OF THE STUDY 1](#_Toc118222233)

[1.2 PROBLEM STATEMENT 2](#_Toc118222234)

[1.3 AIM AND OBJECTIVES OF THE STUDY 2](#_Toc118222235)

[1.4 SCOPE OF THE STUDY 2](#_Toc118222236)

[1.5 LIMITATION OF THE STUDY 2](#_Toc118222237)

[1.6 SIGNIFICANCE OF THE STUDY 3](#_Toc118222238)

[1.7 GAPS IN THE STUDY 3](#_Toc118222239)

[1.8 MOTIVATION 3](#_Toc118222240)

[1.9 OPERATIONAL DEFINITION OF TERMS 4](#_Toc118222241)

[CHAPTER TWO 5](#_Toc118222242)

[2.0 LITERATURE REVIEW 5](#_Toc118222243)

[2.1 REVIEW OF RELATED LITERATURE 5](#_Toc118222244)

[2.1.1 MACHINE Learning Models 9](#_Toc118222245)

[2.1.2 Decision Tree Decision 10](#_Toc118222246)

[2.1.3 Artificial Neural Network (ANN) 10](#_Toc118222247)

[2.1.4 Naive Bayes 11](#_Toc118222248)

[2.1.5 Logistic Regression 11](#_Toc118222249)

[2.1.6 Fuzzy Logic 11](#_Toc118222250)

[2.1.7 Support Vector Machine (SVM) 12](#_Toc118222251)

[2.1.8 Random Forest 12](#_Toc118222252)

[2.1.9 Stochastic Gradient Descent Binary Classifiers 12](#_Toc118222253)

[2.2.0 Multiple Regression 13](#_Toc118222254)

[2.2.1 K-Nearest Neighbour (K-NN) 13](#_Toc118222255)

[2.2 Expert System 14](#_Toc118222256)

[2.3 Academic Success 14](#_Toc118222257)

[2.4 Factors affecting Students’ Academic Performance 15](#_Toc118222258)

[CHAPTER THREE 16](#_Toc118222259)

[3.0 methodology 16](#_Toc118222260)

[3.1 description and analysis of previous system 16](#_Toc118222261)

[3.2 description and Analysis of the Current System 16](#_Toc118222262)

[3.3 SHORTCOMNGS OF the Current System 16](#_Toc118222263)

[3.4 data collection method for the new system 17](#_Toc118222264)

[3.5 Descripion and analysis of the new sysem 17](#_Toc118222265)

[3.6 Methodology for the new system 18](#_Toc118222266)

[3.7 Justification of The New System 18](#_Toc118222267)

[CHAPTER FOUR 19](#_Toc118222268)

[4.0 System Design and implemetation 19](#_Toc118222269)

[4.1 SYSTEM DESIGN 19](#_Toc118222270)

[4.1.0 Flowchart Diagram 19](#_Toc118222271)

[4.1.2 USE CASE DIAGRAM 21](#_Toc118222272)

[4.1.3 DATA FLOW DIAGRAM 22](#_Toc118222273)

[4.2 INPUT SPECIFICATION AND DESIGN 23](#_Toc118222274)

[4.2.1 User Selection 23](#_Toc118222275)

[4.2.2 Administrator Form Login 24](#_Toc118222276)

[4.2.3 User Registration Page 25](#_Toc118222277)

[4.3 OUTPUT SPECIFICATION AND DESIGN 27](#_Toc118222278)

[4.3.1. INVALID LOGIN CREDENTIAL 27](#_Toc118222279)

[4.3.2. RESPONSE SUBMTTED 27](#_Toc118222280)

[4.3.3. DISPLAY PREDICTION 28](#_Toc118222281)

[4.4 JUSTIFICATION FOR PROGRAMMING LANGUAGE AND TOOLS USED 29](#_Toc118222282)

[4.4.1Advantages of using Tkinter 29](#_Toc118222283)

[4.4.2 Advantages of using Excel 30](#_Toc118222284)

[4.5 SYSTEM REQUIREMENT 30](#_Toc118222285)

[Hardware Requirements 30](#_Toc118222286)

[Software Requirements 30](#_Toc118222287)

[4.6 SYSTEM UTILITY AND MAINTENACE 31](#_Toc118222288)

[CHAPTER FIVE 32](#_Toc118222289)

[5.0 SUMMARY, CONCLUSION AND RECOMMENDATION 32](#_Toc118222290)

[5.1 SUMMARY 32](#_Toc118222291)

[5.2 CONCLUSION 32](#_Toc118222292)

[5.3. RECOMMENDATIONS 33](#_Toc118222293)

[5.4. AREAS OF FURTHER STUDIES 33](#_Toc118222294)

[REFERENCES 34](#_Toc118222295)

[APPENDIX I 36](#_Toc118222296)

[APPENDIX II 38](#_Toc118222297)

# CHAPTER ONE

## 1.0 INTRODUCTION

The importance of artificial intelligence in education has risen tremendously over the past few years; this is because it enables us to learn fresh, interesting, and practical information about things. Recent analysis has indicated that technology improvements have had a considerable impact on education, just as they have on many other facets of society and human endeavour (Chen et. al., 2020). The effectiveness of educational institutions is crucial in generating graduates and post-graduates of the highest caliber. By discovering instructional strategies that work for students from a variety of background information, student success prediction will aid educational institutions in improving learning and teaching practices(Nosseir & Fathy, 2020). One indication of course learning outcomes in a learning institution is excellent student performance. Educational institutions required the result of the predicate process to improve students’ academic achievement and to improve learning. Therefore, educational institutions require the result of the predicate process to improve students’ academic achievement and also to improve learning(Lekan et al., 2019). The ability to predict student performance is a crucial area because it enables teachers to identify children who need further academic support(Hussain et al., 2019). Numerous methods is being suggested right now to assess student achievement. One of the most used methods for analysing student performance is data mining, which has recently seen a lot of use in the field of education.

## 1.1 BACKGROUND OF THE STUDY

In educational data mining, machine-learning approaches seek to model and identify significant hidden patterns and valuable information from educational environments. Additionally, machine-learning techniques are used in the academic sector to represent a variety of student attributes as data points by applying them to big datasets.

By fulfilling a variety of objectives, such as extracting patterns, anticipating behaviour, or finding trends, these strategies can be useful in a variety of sectors and help educators deliver the most effective teaching methodologies and track and monitor their students' development.

The usage of data mining techniques on educational data, such as student information, academic records, exam results, and student involvement in class, is referred to as educational data mining, or EDM(Lekan et al., 2019). Educational data mining, which is used to extract usable data and patterns from a sizable educational database, has recently emerged as an efficient technique for identifying hidden patterns in educational data, predicting academic accomplishment, and improving the learning and teaching environment. The useful data and patterns can be utilized to forecast student performance, which will help teachers devise efficient teaching strategies. The information gathered about instructional procedures provides fresh chances to enhance the educational process and maximize users' engagement with digital tools (Zhang et al., 2021). Improvements are made in a variety of areas as a result of the processing of educational data, including behaviour prediction for students, analytical learning, and novel approaches to education policies(Khalaf et al., 2017). In addition to enabling educational authorities to build databased policies, this extensive data gathering will serve as the foundation for software designed using artificial intelligence to aid in the learning process.

Predictive modelling is frequently use in educational data mining to forecast student achievement. Classification, regression, and categorization are three approaches used to develop prediction models. Classification is the most common job used to forecast pupils' success, several categorization techniques have been used to forecast student achievement, which includes Decision Tree, Artificial Neural Networks, Naive Bayes, Fuzzy Logic, K-Nearest Neighbour, and Support Vector Machine. To estimate students' academic performance for this research project, data mining techniques will implore using fuzzy logic and decision tree will be used to make the prediction.

## 1.2 PROBLEM STATEMENT

The academic performance of students has a significant impact on the production of top-notch graduates who will serve as outstanding leaders and human resources for the nation, contributing to its economic and social development. As a result, need to design and implement an expert system for predicting students' academic performance. This will help identify students who have weaknesses in their course of study so that special attention can be given to them to help improve their performance and produce quality graduates.

## 1.3 AIM AND OBJECTIVES OF THE STUDY

This project aims to design and implement an expert system for predicting students’ academic performance. The objectives of the study is stated below:

1. To identify some appropriate variables that influence students’ performance.
2. To design and implement a prediction system
3. To predict the academic performance of student in computer science department.

## 1.4 SCOPE OF THE STUDY

The Federal Polytechnic Bida, Niger state, Computer Science Department Students would be the focus of the project.

## 1.5 LIMITATION OF THE STUDY

To increase the level of accuracy in the prediction, the new system requires employing the services of experts (psychological, counsellor and academic), to help set the rules for evaluation and analysis of the data collected. This might result in additional financing, also, many factors contribute to the academic performance of students, the new system generalize the factors the student, due to time constraints.

## 1.6 SIGNIFICANCE OF THE STUDY

The new system can be use by the management to identify the struggling students at an early stage of the semester so they will have enough time to rework and attain a satisfactory result. An early prediction of students’ academic performance can help the management to provide the needed actions at the right moment, and to plan the appropriate training to improve the student’s success rate such as providing the suffering students with special attention e.g., counselling.

## 1.7 GAPS IN THE STUDY

To monitor and predict student academic performance based on various factors such as demographic background, academic record, etc., various expert systems have been designed and implemented in many institutions(Ao et al., 2008). This is because it is crucial for learning institutions to be able to predict student academic performance in real-time. However, such a system has not been implemented in the Federal Polytechnic Bida computer science department.

## 1.8 MOTIVATION

The ability to forecast a student's academic achievement is essential for parents and higher education officials. Institution administrators usually ignore numerous factors such as demographic and sociological whereas, these factors serve as catalysts for student performance. If these elements are thoroughly considered, the impacted student will be given extra assistance and study schedules to boost academic achievement.

## 1.9 OPERATIONAL DEFINITION OF TERMS

**Expert:** a person who is very knowledgeable about or skillful in a particular area. (Oxford Advanced Learners Dictionary, 2022).

**System:** a set of things working together as parts of a mechanism or an interconnecting network; a complex whole. (Oxford Advanced Learners Dictionary, 2022).

**Prediction:** this is the ability to be able to forecast the outcome of something. In science, a prediction is a rigorous, often quantitative, statement, forecasting what would be observed under specific conditions (Wikipedia, 2022).

**Expert system:** an expert system is a computer program that uses artificial intelligence (ai) technologies to simulate the judgment and behavior of a human or an organization that has expert knowledge and experience in a particular field. (Scott Peterson, 2016).

**Artificial intelligence:** the theory and development of computer systems able to perform tasks normally requiring human intelligence. (Wikipedia, 2022).

**Academic performance:** is the measurement of student achievement across various academic subjects. (Ballotpedia, 2022).

# CHAPTER TWO

## 2.0 LITERATURE REVIEW

The literature review will present the historical perspective, initial developments and the current situation, possibilities of new invention including problems and issues raised, argument discussed, strength and weaknesses to enhance knowledge, sharpen understanding and broaden researcher’s perspective on the application of expert system in predicting students’ academic performance.

## 2.1 REVIEW OF RELATED LITERATURE

(Song & Lu, 2015) explored the use of machine learning to forecast student success in computer engineering using a decision tree algorithm. The study, however, concentrated on predicting students without offering suggestions for how to raise students' performance.

Hussain et. al., (2019) carried out research on an artificial neural network-based prediction model for student performance based on internal assessment. The Artificial Neural Network produced a classification accuracy of 95.34%, which was the highest in this investigation, but its potential for enhancing learning outcomes was not investigated, therefore, creating a conceptual gap.

(Alyahyan & Düştegör, 2020)examined a machine learning-based model for predicting student performance in the context of Wolkite University. Support vector machines, neural networks, naive Bayes, and machine learning techniques algorithms were use. Naive Bayesian had a better performance of 95.7%, indicating that Naive Bayesian performed better than Support Vector Machine and Multi-layer perceptron networks. Despite being effective in predicting students' performance, neural networks, Naive Bayesian, and Support Vector Machine models have not been use to enhance learning outcomes for students who exhibit conceptual and methodological gaps.

Attempt to predict students' academic achievement departments at Blue-Crest College in Accra, Ghana utilizing the Naive Bayes algorithm, information presented was gathered using a feedback rating-scale questionnaire graded on a scale from 1 to 5. From the result, the Naive Bayes algorithm accurately predicts student performance 92.2% of the study show that the Naive Bayes algorithm was the most reliable machine learning model (Jayaprakash Sr Lecturer et al., n.d.) The study proposed testing the capacity of additional machine learning models to improve student outcomes since the optimal model for increasing learning outcomes was not tested.

(Yohannes & Ahmed, 2018)examined student data using data mining approaches to build a model for performance prediction. The study used categorization techniques like decision trees to evaluate student performance. The study further provided information on how the affected students’ performance can be improved, such as supporting students through their learning process and making prompt judgments to avoid academic risk and desertion.

(Alyahyan & Düştegör, 2020)studied the academic performance of engineering students using a regression model. A questionnaire was created based on the intended study to collect data from the students. The outcomes demonstrated that the regression model provides greater prediction accuracy. The study, however, was unable to demonstrate whether a regression model can be utilized to enhance learning outcomes

(Ulloa-Cazarez, 2014)demonstrated how prediction techniques can be used on educational data using the fuzzy logic methodology to construct a model to forecast the performance of online students using grades collected during the first half of the scholar cycle. The study suggested integrating fuzzy logic techniques and genetic algorithms to improve the membership functions for future studies after discovering that fuzzy logic does not achieve a good accuracy level when compared with the statistical regression model.

(Lekan et al., 2019), made use of support vector machines to predict student academic achievement by making use of the student's Grade Point Average. Their result showed that the Supportive vector machine performs better than other ML algorithms with an accuracy of 94% percent and 97% respectively.

(Renaningtias et al., 2018) conducted a study using neural networks to predict student performance. The study assessed the prediction performance of neural networks against six different classifiers on this dataset to determine the neural networks' applicability. From the study findings when generic training is employed, neural networks outperform other prediction models.

(Hussain et al., 2019) looked into how artificial neural networks could predict the performance of postgraduate students. The report conducted a thorough investigation on applying deep learning to predict postgraduate student success in R programming. The Artificial Neural Method was employed to enhance the model's performance and the predictability of results. From findings, Linear Regression could only provide an accuracy of 12% when the accuracy of several approaches such as Random Forest, Deep Learning (ANN), and Linear Regression are compared. Deep Learning/Artificial Neural Network can predict the final grades (G3) on the test dataset that wasn't used to create the model with an accuracy rate of 97% The focus of the models was on performance prediction for pupils. The study concluded that testing the effectiveness of various machine learning models to improve student outcomes is necessary because the best model for increasing learning outcomes has not yet been evaluated.

(Song & Lu, 2015) used a data mining classifier to predict undergraduate students' performance. Four different classifiers decision tree, random forest, Naive Bayes, and rule induction are used to assess student performance. Various classifiers exhibit varying degrees of accuracy depending on the various algorithms utilized in them. The findings showed that decision trees had a 90% prediction accuracy, Naive Bayes had an 84% prediction accuracy, the random forest had an 85% percent prediction accuracy, and the induction rule had an 82% prediction accuracy for predicting student performance. A conceptual gap still exists because the model that can be employed to enhance learning outcomes has never been tested.

Al-Shabandar et. al., (2017) investigated the different models to know which is the most correct machine learning technique to forecast learning outcomes in massive open online courses. According to the simulation findings from both studies, random forest and supportive vector machines performed at their absolute best respectively. The outcomes demonstrate that machine learning is a workable solution to prediction problems, offering a remarkable capacity to discern between successful and unsuccessful outcomes. Although, in the study, the model that can be used to improve learning outcomes has never been proposed.

(Renaningtias et al., 2018)used machine learning to forecast student performance; in his thesis, three different machine learning techniques were employed (Naive Bayes classification, decision trees, and linear regression). The findings demonstrate that machine learning may be used to accurately predict student achievement. The most accurate algorithms were naive Bayes classification for the first data set (98%) and decision trees for the second data set (78%). There is a conceptual gap since the model that can be used to improve learning outcomes has never been proposed.

(Yağcı, 2022)in research used machine learning algorithms to model students' academic performance. For each lesson, datasets were utilized to generate classification models using a different algorithm. The findings demonstrated that our algorithms are discovering helpful trends in the classification approaches that we may use to forecast the likelihood that students' outcomes will be unsuccessful. However, the findings suggest that utilizing the Logistic Model Tree (LMT) can help anticipate poor learning performance, allowing easy identification of students that are likely to struggle in their academic pursuits.

(Ulloa-Cazarez, 2014) used Random Forest, Artificial Neural Network, Naive Bayesian, and Logistic to predict academic success. In the study, logistic regression, naive Bayesian analysis, artificial neural networks, and the random forest technique were contrasted. The study discovered that the training dataset's makeup is crucial. Random forest is frequently overfitted in terms of accuracy and is not superior to alternative approaches for datasets with modest fitur. For datasets with modest fitur, Random Forest's accuracy is not any better than other approaches. In numerous testing sets, neural networks continue to perform better than others. Additionally, the performance of logistic regression and Naive Bayes is equal for this dataset.

## **2.1.1 MACHINE Learning Models**

The phrase "machine learning" is frequently used to describe an analytical technique created to find patterns in data and relationships between data variables. Furthermore, given that complex input variables are anticipated, a critical element of machine learning is the ability to understand complicated non-linear relationships (Alsariera et al., 2022). Depending on the applicability of the data collection and the goals of the data analysis process, several machine learning models can be modified to analyse the data, such as classification, clustering, and association rules mining. Machine learning is helpful in monitoring and analysing the learning process in schools, predicting learners' performance by providing necessary academic assistance, academic guiding and advice mentoring, examining the efficiency and effectiveness of learning methods, providing meaningful feedback for teachers and learners, and changing learning environments for the benefit of students, according to Hussain et al. (2019). A useful method for forecasting good and bad performances at different levels of education is the use of machine learning algorithms to anticipate students' performance based on their past knowledge and their in-term performance (Song & Lu, 2015).

Machine learning has an edge over conventional statistical analysis because it prioritizes predicted performance over a priori super-population assumptions and theoretically verifiable features. By doing this, tutors can support the strongest students while also promoting the weakest, which enhances learning. To achieve this goal, machine learning is utilized. To find models or patterns in data, machine learning techniques are applied, and they are beneficial for decision-making (Hussain et. al., 2018). In the current system of education, the capacity to anticipate student achievement is extremely important. The ideal machine learning model for forecasting student performance and optimizing learning outcomes, however, is not immediately apparent (Zhang et al., 2021). To uncover hidden knowledge from massive amounts of data, a variety of data mining techniques are used. Machine learning models include decision trees, neural networks, the nearest neighbour Bayesian classifier, support vector machines, random forests, logistic regression, linear discriminant analysis, multiple regression, and self-organized maps.

### 2.1.2 Decision Tree Decision

To find tiny or large data structures and predict values, decision trees have been employed widely by the majority of academics. After analysing the training set, decision tree classifiers are used in data mining to construct trees that will be utilized to make predictions. One of the respected and important classification tools is the decision tree classifier. Decision tree classifiers often have a tree-like structure that begins with the root attributes and finishes with leaf nodes. Additionally, it has several branches made up of distinct qualities, with each branch's leaf node standing for a class or type of class distribution. The relationship between qualities and their relative importance are explained by decision tree algorithms. Decision trees have the advantage of defining rules that are easily understood and interpreted by users, requiring no sophisticated data preparation, and working well with numerical and categorical variables (Song & Lu, 2015). ID3 is the name of the main decision tree construction algorithm.

### 2.1.3 Artificial Neural Network (ANN)

Unquestionably, one of the most well-liked methods for educational data mining is the artificial neural network (ANN). Synapses on the dendrites of the neural network are where signals are received. According to the ANN technique, the neuron is engaged and a signal is generated on the axon when the received signals are sufficiently strong (above the threshold). This signal may activate further neurons and be transmitted to other synapses (Alsariera et al., 2022). The basic building blocks of an artificial neural network are input (synapses) multiplied by the weight (the intensity of each signal), which is then calculated by a mathematical function to determine when to activate the neuron and generate an output. A neural network's ability to identify all potential interactions between predictor variables is a benefit (Nosseir & Fathy, 2020). Even when there is a complicated nonlinear relationship between the dependent and independent variables, neural networks are still capable of performing flawless detection. Neural network technology is chosen as one of the top prediction techniques as a result.

### 2.1.4 **Naive Bayes**

The Naïve Bayes method is a straightforward probabilistic classifier that relies on the Bayes theorem and naive independence assumptions. One of the effective and efficient inductive learning methods for machine learning and data mining is naive Bayes. These algorithms make use of Bayesian probability, which calculates future probabilities based on historical data. One of the most fundamental classification techniques, it has numerous uses including the detection of spam in email, sorting of personal emails, document classification, the identification of sexually explicit content, language detection, and sentiment detection. Unsophisticated Bayes performs admirably in a variety of challenging real-world issues, despite its simplistic assumptions and naive design (Yağcı, 2022). A Naive Bayes classifier is a straightforward probabilistic classifier built using the Bayes theorem and naive independence assumptions from Bayesian statistics. A naïve Bayes classifier, to put it simply, thinks that the presence (or lack) of one feature in a class has nothing to do with the presence (or absence) of any other feature.

### 2.1.5 Logistic Regression

Another machine learning technique, logistic regression, is used to assess the impact of one or more independent variables (x) on a single binary dependent variable (y) (Yağcı, 2022). There are criteria variables used: -A dummy variable used as the dependent variable should only have two possible values, such as yes or no, 1 or 0, etc.

### 2.1.6 **Fuzzy Logic**

Fuzzy logic is an approach to variable processing that allows for multiple possible truth values to be processed through the same variable. Fuzzy logic attempts to solve problems with an open, imprecise spectrum of data and heuristics that makes it possible to obtain an array of accurate conclusions. Fuzzy logic, among the various available Artificial Intelligence techniques, emerges as an advantageous technique for predicting future events. Subjective and Objective modelling are two types of fuzzy modelling. Fuzzy logic stems from the mathematical study of multivalued logic. Whereas ordinary logic deals with statements of absolute truth (such as, "Is this object green?"), fuzzy logic addresses set with subjective or relative definitions, such as "tall", "large", or "beautiful". This attempts to mimic the way humans analyse problems and make decisions, in a way that relies on vague or imprecise values rather than absolute truth or falsehood.

### 2.1.7 Support Vector Machine (SVM)

Support Vector Regression is a form of Support Vector Machine that makes predictions and assigns support vectors to differentiate features. SVMs are characterized as a collection of connected supervised learning methods for regression and classification (Lekan et al., 2019). They belong to the generalized linear classification family. SVM's ability to simultaneously minimize the empirical classification error and maximize the geometric margin is a key characteristic. SVM is hence also referred to as a maximum margin classifier. Structural Risk Minimization is the foundation of SVM (SRM). A maximal separating hyperplane is built when SVM maps the input vector to a higher dimensional space.

### 2.1.8 Random Forest

A collection of decision trees constructed with a random element is known as a random forest (Rincon-Flores et al., 2020). An ensemble learning technique for classification called random forest builds several unpruned classification trees using the bootstrap sampling approach on the training data. The random forest has been used to study several intriguing issues, and it is clear that this method has a lot of potential for producing helpful classification models (Yağcı, 2022).

### 2.1.9 Stochastic Gradient Descent Binary Classifiers

Large-scale supervised machine learning problems have become more and more common for tackling via stochastic gradient descent (SGD). Applicable to Support Vector Machine (SVM) and Logistic optimizations, it offers a quick way for minimizing a variety of loss functions. SGD, however, does not offer a useful stopping condition.

Typically, held-out data can be used to calculate the ideal number of iterations over the data. Machine learning algorithms have been successfully trained using SGD. In most cases, it has been discovered to confer a sizable reduction in training time without compromising accuracy (Alyahyan & Düştegör, 2020). SGD can be used to regularize ordinary convex loss functions with good results; however, (Diab, 2019) recommended utilizing SGD without the customary regularization term and regularizing with early halting. This method, which is now extensively used, is carried out by separating the training set into several sections that each contain a new training set and a validation set. When compared to other machine learning models, such as deep learning, stochastic gradient descent has the advantage of being effective and simple to apply.

### 2.2.0 Multiple Regression

A statistical technique for determining the relationship between the variables in the data is regression. It focuses primarily on the interaction between the dependent variable and independent variables, often known as predictors (Yohannes & Ahmed, 2018). Understanding how the dependent variable's value varies when an independent variable is altered is helpful. An equation containing the independent variables, certain coefficients, and the slope value is created using the independent variable's value (Tokunbo et al., 2018). Regression methods come in a wide variety. One such method is the principally predictive linear regression methodology. The link between one dependent variable and one or more independent variables is examined using one or more independent variables.

### 2.2.1 K-Nearest Neighbour (K-NN)

The k-Nearest Neighbour algorithms (k-NN) classify objects in the feature space based on the nearby training samples. K-NN is a type of instance-based learning, or lazy learning, in which the complete calculation is postponed in expectation of classification and the function is only roughly estimated nearby. The primary issue with the k-NN method is that loud or unsuitable characteristics can severely impair its accuracy. Similarly, its accuracy suffers if the feature balance is unreliable given its significance.

## Expert System

An expert system in artificial intelligence is a computer program that mimics the capacity for judgment of a human expert. Expert systems are created to reason through knowledge bases that are primarily represented as if-then rules rather than through traditional procedural code. The rules are founded on an understanding of the issue area and approaches. One should rely on a range of sources, including books, research articles, interviews, surveys, and protocol analysis, to gain knowledge for an expert system. An expert system is a system that receives user input on a particular topic and outputs a probabilistic result based on the knowledge that the system has stored in the form of rules (Patel et al., 2015). The system is used to make decisions based on information from the real world. In a rule-based expert system, the system uses information from the real world to inform its decisions.

## Academic Success

Student achievement is an important element of higher education institutions since it is seen as a crucial indicator for evaluating the quality of educational institutions (National Commission for Academic Accreditation, 2015). The literature offers several definitions of student success.

According to a summary of the literature by (Yohannes & Ahmed, 2018)"Student success is defined as academic achievement, engagement in educationally purposeful activities, satisfaction, acquisition of desired knowledge, skills and competencies, persistence, attainment of educational outcomes, and post-college performance.". Although this is a multi-dimensional definition, authors in (Molokomphale & Mhlauli, 2014) gave an amended definition focusing on the most crucial six components, namely "Academic achievement, satisfaction, acquisition of skills and competencies, persistence, attainment of learning objectives, and career success." The majority of published studies define academic success narrowly as academic achievement, despite calls for more nuanced understandings of the term.

In universities, grade systems are used to assign a scale for evaluating students' academic performance, such as the Grade Point Average (GPA) or Cumulative Grade Point Average (CGPA) (Parker et. al., 2018). Grades are also a major component of academic achievement (Lekan et al., 2019). Academic success has also been characterized about students' perseverance, also known as academic resilience, which is in turn primarily assessed through grades and GPA, the evaluations that are by far the most accessible in educational institutions.

## Factors affecting Students’ Academic Performance

In many universities around the world, research has been done to pinpoint the causes of poor academic performance. Most studies concentrate on the three intervening factors: parents (family causal factors), teachers (academic causal factors), and students (personal causal factors), even though the impact on academic performance varies depending on the academic setting, the particular students, and even the culture.

# CHAPTER THREE

## methodology

This chapter covers the analysis and limitations of the existing system, justification for the new system, project methodology, data collecting, and the suggested system design.

## 3.1 description and analysis of previous system

Currently, there are many prediction systems existing in various institutions in Nigeria. An example is a system adopted by the Department of Computer and Information Sciences, Tai Solarin University of Education, Ogun State Nigeria in which prediction is done using a neural network to establish and analyze the complex nonlinear relationship that exists between cognitive and psychological variables that influence the academic performance of secondary school students. In spite of the high level of prediction accuracy of ANN in nonlinear phenomena, however, the model does not easily allow the identification of how predictor variables are related to one another in the explanation of the academic outcome. In other words, ANN the model does not specify an explicit mathematical model for the relationship between inputs and outputs, hence the need for further research in this regard.

## description and Analysis of the Current System

In the Federal Polytechnic Bida, prediction of students’ academic performance has been done in various forms, but all the methods still boil down to the manual process, for instance in the exam’s office, students’ academic performance is easily predicted based on their performance on various courses, lecturers, on the other hand predict students’ performance based on responsiveness during lectures, coordinate, on and seriousness, though most times this methods involves generalizing the student performance using a key a factor.

## SHORTCOMNGS OF the Current System

Despite the use of various manual forms of prediction by the department, authenticity can still not yet be verified, as most times students’ performance is generalized and not done individually which in turn is not justifiable.

The current system is limited to:

1. Prediction is done based on outcomes of exams and student position in class
2. Factors influencing the performance are not determined
3. There is usually variation in prediction based on different individual perspectives.

## data collection method for the new system

To develop the model, a number of steps was followed which include: data collection, data processing, analysis, code before the prediction system will be realized. Based on the study, the questionnaire was prepared and answered by experts in various fields. The questionnaire is divided into two sections, (Personal Factors and Academic Factors). The questionnaire will be distributed to different experts for analysis. The data is analysed to build a predictive model. Table 1 & 2 shows the questionnaire for personal information and academic information of the students presented to various experts for their analysis.

## Descripion and analysis of the new sysem

Predicting a system is usually done by learning from the past for which historical data is obtained and analysed to study the results. The new system will have undergone data training as the questionnaires distributed were analysed by experts on areas and factors that will be used for prediction. There are numerous benefits of the proposed new system, this includes:

1. Prediction accuracy will increase since analysis was done by experts in the selected fields
2. The system analysis each variable separately before drawing conclusion
3. Effect of each individual variable can be determined e.g., academic factors
4. Students can be assisted in individual areas with emphasis
5. Prediction is fast and reliable
6. System can easily be modified and updated

## Methodology for the new system

The new will design using Python Tkinter GUI (Graphical User Interface). Tkinter is one of the most popular Python GUI libraries for developing desktop applications. It's a combination of the TK and python standard GUI framework. Tkinter provides diverse widgets such as labels, buttons, text boxes and, checkboxes that are used in a graphical user interface application. It's cross-platform, so the same code works on Windows, macOS, and Linux.

## Justification of The New System

The proposed system will follow Decision Tree and Fuzzy Logic algorithm in making predictable decisions, follow the Fuzzy logic which is well recognized prediction algorithm. Also, the data that will be used for prediction will be analysed by experts on the different fields to which the data were collected. With the proposed algorithm and the analysis of the data by experts, there will be high level of accuracy of the prediction.

# CHAPTER FOUR

## 4.0 System Design and implemetation

## 4.1 SYSTEM DESIGN

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them. For the purpose of the system UML diagrams is used to explain the system and its functionalities. A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

### 4.1.0 Flowchart Diagram

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows. Fig 4.1 below shows the flow chart of the student academic performance prediction system. The flow chart comprises of the following:

**SELECT USER:** The system gives room to select the type of users (Administrator or Student)

**ADMINISTRATOR:** This user is also known as the instructor; the system asks for login credentials before access is given to this user.

**STUDENT:** Every student is requested to fill a form; this form contains their personal data before they can be allowed to make prediction.

**ANSWER PREDICTION QUESTION:** The system prompts series of questions for the student user, these questions are analysed following the selected algorithm, before prediction is done for each student based on their response to the questions prompted by the system.

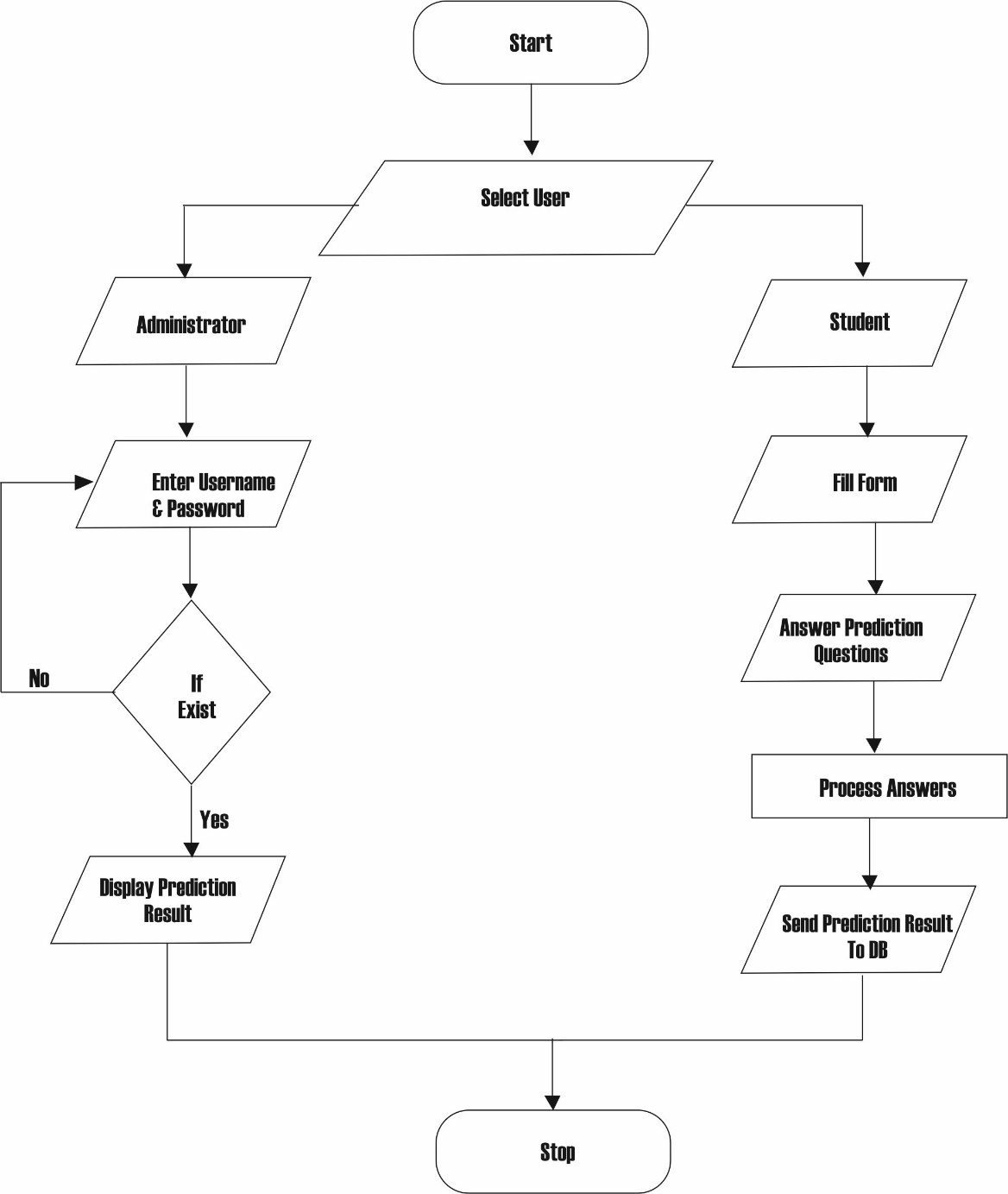
****

Fig 4.1 Flow Chart of the Student Academic Performance Prediction System

### 4.1.2 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. While a use case itself might drill into a lot of detail about every possibility, a use case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system". They provide the simplified and graphical representation of what the system must actually do.

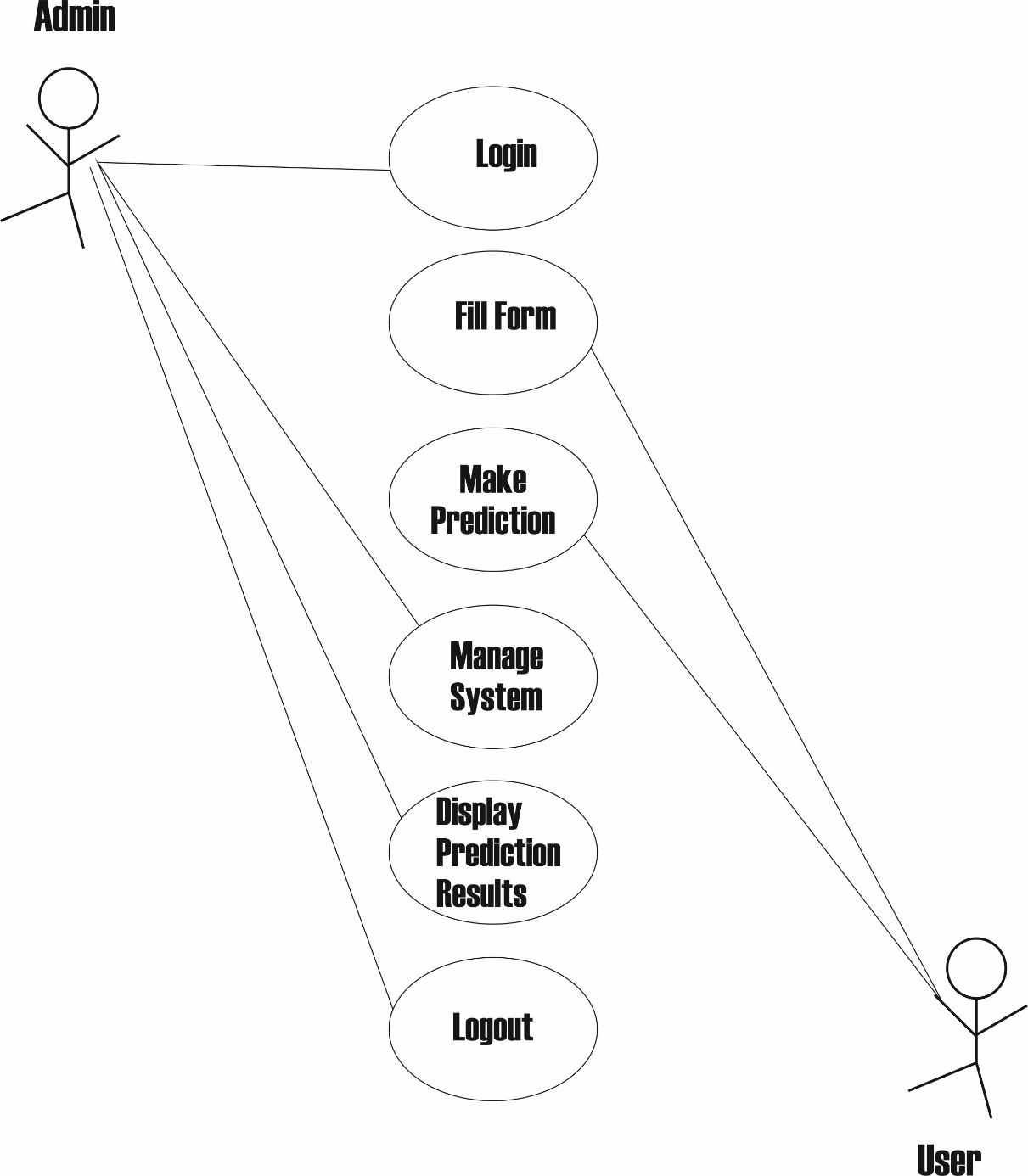


Fig 4.1.2 Use case diagram of the student’s academic performance prediction system.

The use case diagram for this prediction system as depicted in Fig. 4.1.2 above consists of two actors - the administrator and the student. The system also consists of six use cases – two of which are operated by user while the administrative officer operates on the rest of the cases. The activities of the student include fill a form, and make prediction. While the activities of the administrator comprise of the activities of the user (student) together with other activities such as view/ display prediction of students and management of the system.

### 4.1.3 Data Flow Diagram

A dataflow diagram is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into detail, which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

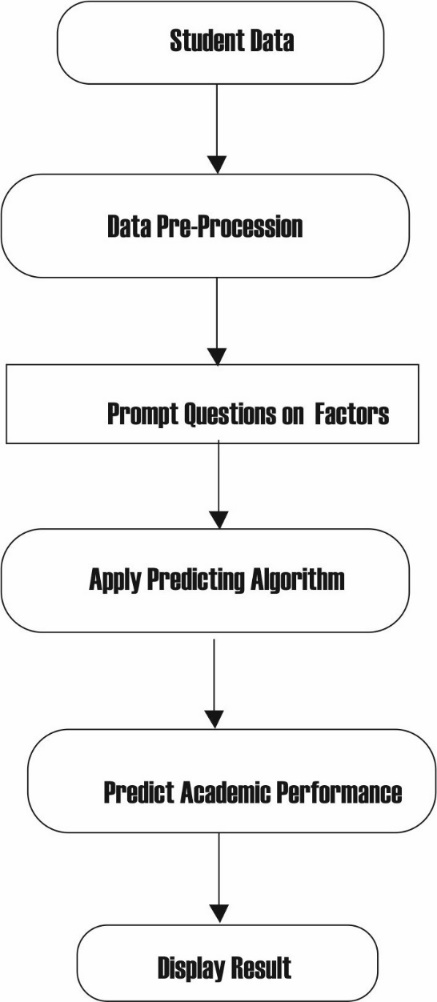


Fig 4.1.3 represents the data flow between datasets, user and prediction model.

The data flow diagram of the Fig. 4.1.3 above shows the movement of data. After successfully filling the form, questions are then prompted for the user to answer based on some of the selected factors that affects students’ academic performance. After which the system apply the algorithm for prediction and stores the user (student) prediction result in the database of the system.

## INPUT SPECIFICATION AND DESIGN

### 4.2.1 User Selection

Table 4.2.1 User Selection

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type(value)** | **Constraint** |
| Administrator | Button | Not null |
| User | Button | Not null |

Fig. 4.2.1 below shows the system user selection interface, the system comprises of two users an administrator (lecturer / instructor) and a user (student) each is expected to click on the button that designated for their level.

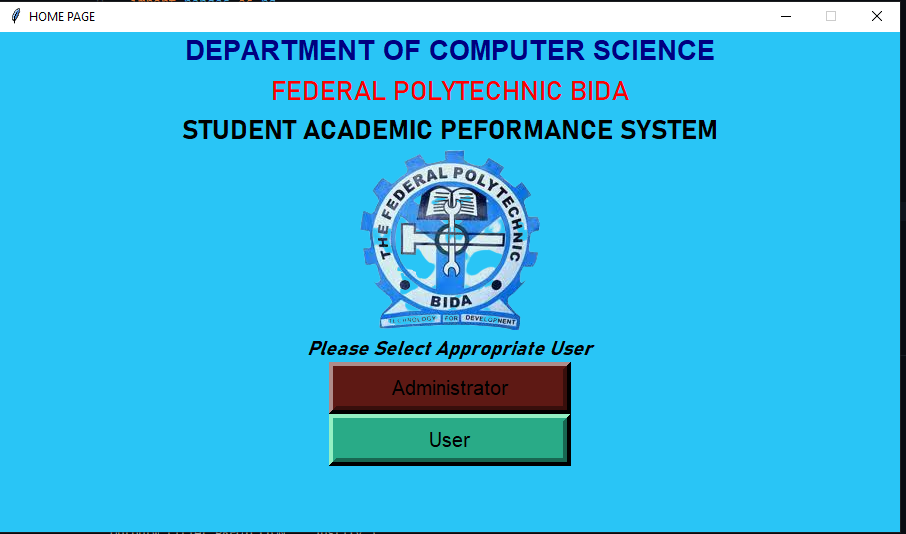


Fig 4.2.1 User Selection Interface

### 4.2.2 Administrator Form Login

Table 4.2.2 Administrator login

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type(value)** | **Constraint** |
| Username | VarChar (10) | Primary key |
| Password | VarChar (10) | Not null |

The administrator Login Page as depicted in Fig. 4.2.2 below provides the administrator with a GUI to login the software, it requires the administrator to enter a username and password.

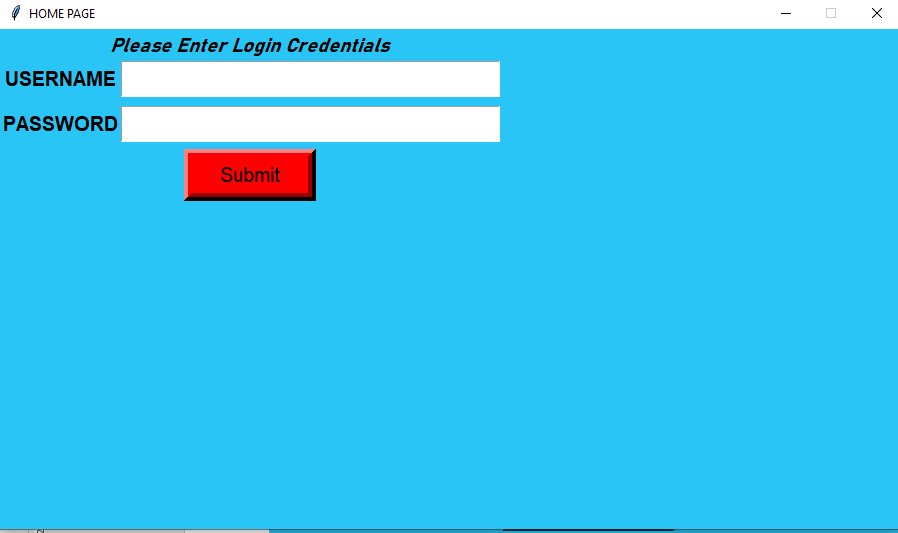


Fig.4.2.2 Administrator Login Page

### 4.2.3 User Registration Page

Table 4.2.3 User Registration Page

|  |  |  |
| --- | --- | --- |
| **Field name** | **Data type(value)** | **Constraint** |
| Full name | VarChar(30) | Not Null |
| Matric No | VarChar(10) | primary key |
| Department | VarChar (15) | Not Null |
| Semester | VarChar(7) | Not Null |
| Email | VarChar (20) | Not null |
| Telephone | VarChar(11) | Not Null |
| Address | VarChar (30) | Not null |

The user registration page as depicted in Fig. 4.2.3 is a form that pop-up whenever the button “Student” is selected in the user selection page. It is a form that a user is expected to fill before prediction can take place.

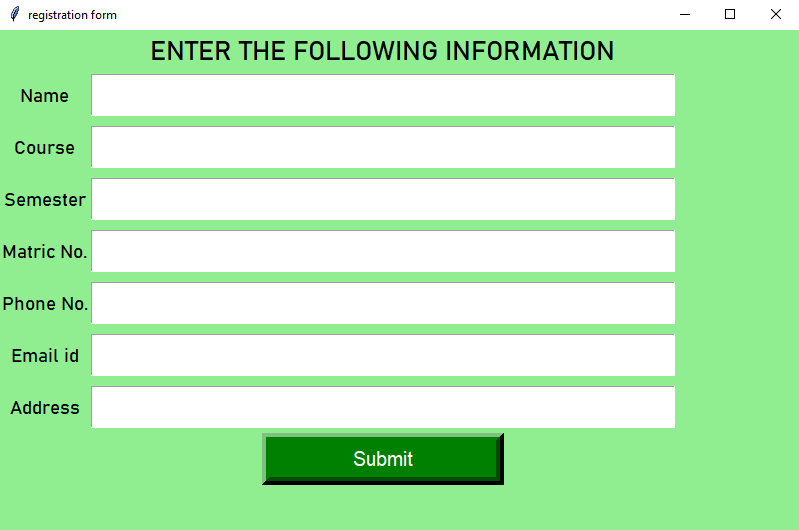


Fig. 4.2.3: User Registration Page

**4.2.4 Prediction Page**

The prediction page as depicted in Fig. 4.2.4 below is an interface which comprises of series of questions which are in Radio Button format.

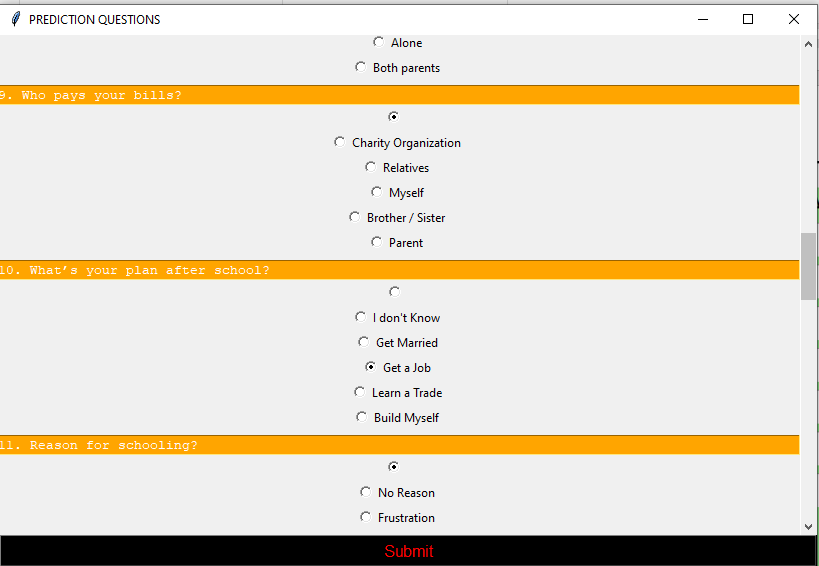


Fig 4.2.4 Prediction Page

## 4.3 OUTPUT SPECIFICATION AND DESIGN

### 4.3.1. Invalid Login Credential

The invalid login credentials as depicted in Fig. 4.3.1 below display whenever the admin login credentials entered does not match with that set by the system management or when a user enters wrong login credentials.

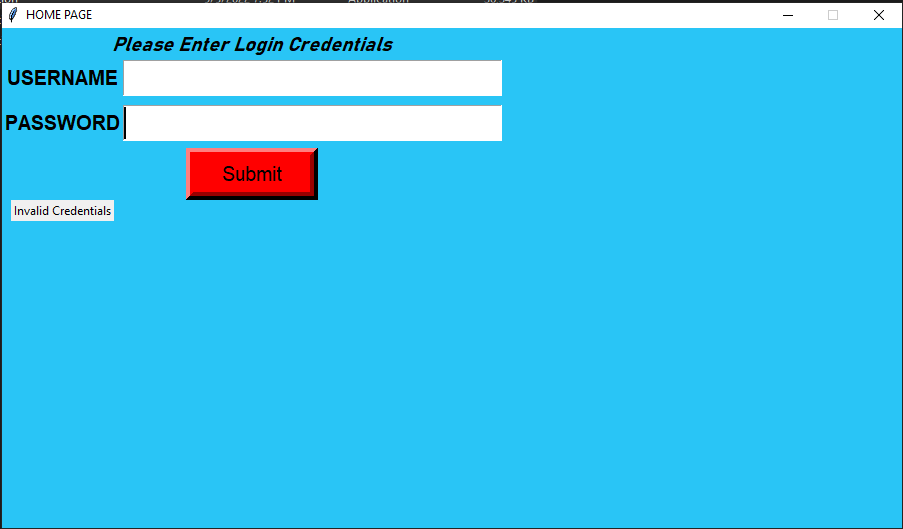


Fig. 4.3.1 Invalid Login Credentials

### 4.3.2. Response Submitted

The response submitted as depicted in Fig. 4.3.2 below is a message box whenever the user (student) has finished making prediction (answering prediction questions) it is a confirmation message showing the users answers has been submitted.

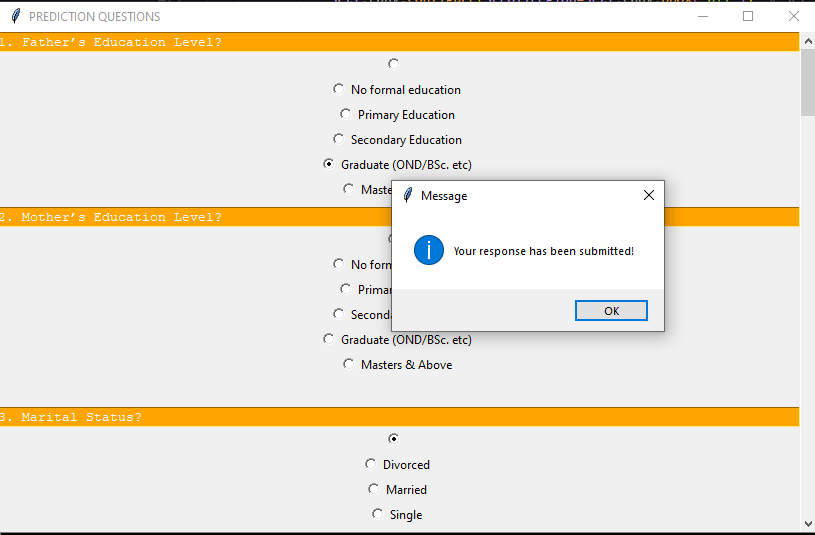


Fig 4.3.2 Response Submitted

### 4.3.3. Display Prediction

Display Prediction as depicted in Fig. 4.3.3 below is use by the administrator view list of users that (students) that have made prediction and the outcome of their prediction analysis.

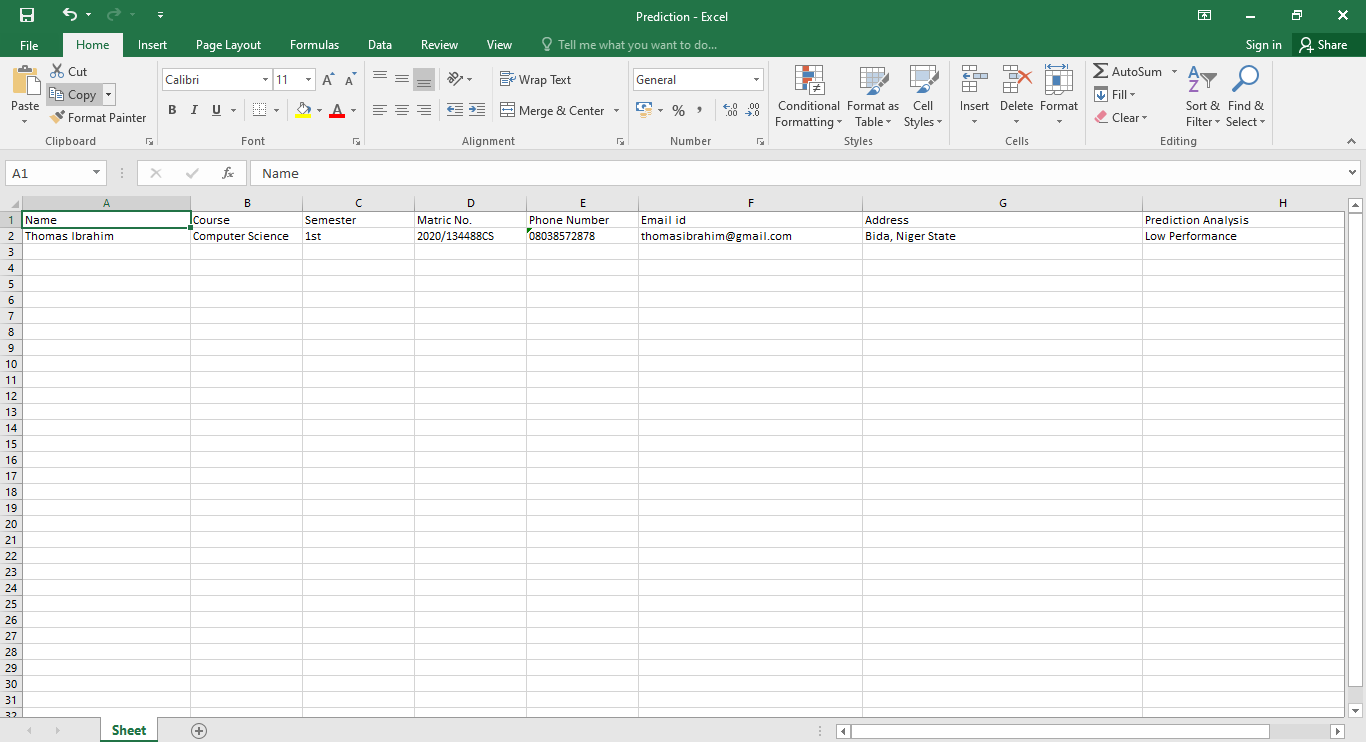


Fig 4.3.3 Prediction Record stored in database

## 4.4 JUSTIFICATION FOR PROGRAMMING LANGUAGE AND TOOLS USED

Python Tkinter is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [binding](https://en.wikipedia.org/wiki/Language_binding) to the [Tk](https://en.wikipedia.org/wiki/Tk_(software)) [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's [de facto standard](https://en.wikipedia.org/wiki/De_facto_standard) GUI. Tkinter is included with standard [Linux](https://en.wikipedia.org/wiki/Linux), [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [macOS](https://en.wikipedia.org/wiki/MacOS) installs of Python. As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete [Tcl](https://en.wikipedia.org/wiki/Tcl) [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) embedded in the Python [interpreter](https://en.wikipedia.org/wiki/Language_interpretation). Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application. [Tkinter](https://www.geeksforgeeks.org/python-gui-tkinter/) is an open-source Python Graphic User Interface (GUI) library well known for its simplicity. It comes pre-installed in Python, so you don’t even need to think about installing it. These characteristics make it a strong position for beginners and intermediates to begin with. Tkinter cannot be used for larger-scale projects.

### 4.4.1Advantages of using Tkinter

* Tkinter is easy and fast to implement as compared to any other GUI toolkit.
* Tkinter is more flexible and stable.
* Tkinter is included in Python, so nothing extra need to download.
* Tkinter provides a simple syntax.
* Tkinter is really easy to understand and master.
* Tkinter provides three geometry managers: place, pack, and grid. That is much more powerful and easy to use.

Microsoft Office Excel was used as the database. Excel is perfect for numerical data, making calculations, and it's easy to enter data. Excel databases provide a simple way to analyze data (such as sales numbers and forecasts), look at various calculations, and compare different data sets. Of course, there are advanced formulas and functions if you dive deeper and invest time in becoming a pro.

### 4.4.2 Advantages of using Excel

* Excel is already tabular, and it’s easy to view an Excel worksheet and access data
* Most business leaders have access to and know the basics of Excel
* Excel is perfect for numerical data, making calculations, and it’s easy to enter data
* The rows and columns are easier to understand than some more complex database layouts
* VBA allows users to automate tasks and link Excel to other software programs

## 4.5 SYSTEM REQUIREMENT

The system requirement is of two categories. The software requirement and the hardware requirement.

### Hardware Requirements

* Intel core i3 or more
* Hard disk: 256GB
* RAM: 2GB

### Software Requirements

* Operating System: Windows 7 and above.
* Python 3.10
* Visual Studio Code or PyCharm or Anaconda
* Python libraries: Tkinter

## 4.6 SYSTEM UTILITY AND MAINTENACE

The following measures should be observed to ensure smooth running of the system:

* Always sign-out after you have finish using the system
* When a new staff record is added to the system, ensure to clear the previous generated timetable and regenerate to appropriately add the new course to the timetable.
* Keep your access(login) detail safe.
* Keep your windows defender up-to date for windows 10 users.
* Keep an up-to date antivirus for windows 8 and below.

# CHAPTER FIVE

# 5.0 SUMMARY, CONCLUSION AND RECOMMENDATION

# 5.1 SUMMARY

The students’ performance plays an important role in producing best quality graduates who will become great leaders and manpower for the country thus responsible for the country’s economic and social development. Academic achievement is one of the major factors considered by employers in hiring workers especially for the fresh graduates. Thus, students have to put the greatest effort in their study to obtain good grades and to prepare themselves for future opportunities in their career at the same time to fulfil the employer’s demand. Being able to predict students’ academic performance will help institutions management to be able provide support for the students with low academic performance. Study revealed that certain factors affect student academic performance which can be categorized into: demographic, academic, social etc. The project presents a system which predict students’ academic performance by asking questions based on the outlined factors. The response from the student is used to grade the performance of the study.

# 5.2 CONCLUSION

The prediction of student’s performance in advance is very important issue as it helps educational institution to supply the student with additional support. An early performance prediction would be beneficial for at-risk students, facing difficulty in attaining good grades in the class. To support such students in their learning to improve their progress, it is important to periodically predict their performance so they can be supported. From research, it is believed that by going through the factors that affect a student, the student academic performance can be predicted. When the prediction is successfully done, for students with low academic performance, the instructor may devote some extra time to improve the final examinations marks since the student whose performance is low is said to be at risk.

# 5.3. RECOMMENDATIONS

Prediction of students’ academic performance plays a vital role in institutions, as it tells lectures students possible result, thereby giving room for necessary assistants to students. It is therefore recommended that the prediction system be implemented in the computer science department, to enable lecturers provide the necessary assistance to students with low performance.

# 5.4. AREAS OF FURTHER STUDIES

Many factors affect study academic performance which include; demographic, academic, social among others, the system used only makes use of general questions and does not state which of the factors has more impact on a student academic performance. Therefore, it is recommended that for further study, questions to be used for prediction should be categorized, this is to enable the instructor tell which factor is currently affecting a student performance and to know where to provide the necessary care for the student.

# REFERENCES

Alsariera, Y. A., Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A. A., & Ali, N. (2022). Assessment and Evaluation of Different Machine Learning Algorithms for Predicting Student Performance. Computational Intelligence and Neuroscience, 2022, 1–11. https://doi.org/10.1155/2022/4151487

Alyahyan, E., & Düştegör, D. (2020). Predicting academic success in higher education: literature review and best practices. In International Journal of Educational Technology in Higher Education (Vol. 17, Issue 1). Springer. https://doi.org/10.1186/s41239-020-0177-7

Ao, S. I., International Association of Engineers, WCECS (2008.10.22-24 San Francisco), & World Congress on Engineering and Computer Science (2008.10.22-24 San Francisco). (2008). WCECS 2008, World Congress on Engineering and Computer Science, San Francisco, USA, 22-24 October, 2008. IAENG International Association of Engineers.

Hussain, S., Muhsin, Z. F., Salal, Y. K., Theodorou, P., Kurtoğlu, F., & Hazarika, G. C. (2019). Prediction model on student performance based on internal assessment using deep learning. International Journal of Emerging Technologies in Learning, 14(8), 4–22. https://doi.org/10.3991/ijet.v14i08.10001

Jayaprakash Sr Lecturer, S., Assoc Professor, B. E., & Chandar Lecturer, V. (n.d.). Predicting Students Academic Perfomace using Naive Bayes Algorithm.

Khalaf, A., Majeed, A., Akeel, W., & Salah, A. (2017). Students’ Success Prediction based on Bayes Algorithms. International Journal of Computer Applications, 178(7), 6–12. https://doi.org/10.5120/ijca2017915506

Lekan, A. J., Oloruntoba, S. A., & Akinode, J. L. (2019). Student academic performance prediction using support vector machine design and development of automated lecturetime-tabling system view project computer security and networks view project ijesrt international journal of engineering sciences & research technology student academic performance prediction using support vector machine. International Journal of Engineering Sciences & Research Technology, 588. https://doi.org/10.5281/zenodo.1130905

Molokomphale, L., & Mhlauli, M. B. (2014). An Investigation on Students Academic Performance for Junior Secondary Schools in Botswana. In International Journal of Environmental & Science Education (Vol. 3). http://www.eurojedu.com

Nosseir, A., & Fathy, Y. M. (2020). A mobile application for early prediction of student performance using fuzzy logic and artificial neural networks. International Journal of Interactive Mobile Technologies, 14(2), 4–18. https://doi.org/10.3991/ijim.v14i02.10940

Patel, J., Patel, N. D., & Patel, N. S. (2015). International Journal of Computer Science and Mobile Computing A Research on Expert System using Decision Tree and Naive Bays Classifier. In International Journal of Computer Science and Mobile Computing (Vol. 4, Issue 5). www.ijcsmc.com

Renaningtias, N., Suseno, J. E., & Gernowo, R. (2018). Hybrid Decision Tree and Naïve Bayes Classifier for Predicting Study Period and Predicate of Student’s Graduation. In International Journal of Computer Applications (Vol. 180, Issue 49).

Rincon-Flores, E. G., Lopez-Camacho, E., Mena, J., & Lopez, O. O. (2020). Predicting academic performance with Artificial Intelligence (AI), a new tool for teachers and students. IEEE Global Engineering Education Conference, EDUCON, 2020-April, 1049–1054. https://doi.org/10.1109/EDUCON45650.2020.9125141

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. Shanghai Archives of Psychiatry, 27(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Tokunbo Olufemi, O., Adekunle Adediran, A., & Oyediran, W. (2018). Factors Affecting Students’ Academic Performance In Colleges Of Education In Southwest, Nigeria. In British Journal of Education (Vol. 6, Issue 10). www.eajournals.org

Ulloa-Cazarez, R. L. (2014). Fuzzy Logic Model for predicting student performance on online courses. https://doi.org/10.13140/RG.2.1.1236.6720/1

Yağcı, M. (2022). Educational data mining: prediction of students’ academic performance using machine learning algorithms. Smart Learning Environments, 9(1). https://doi.org/10.1186/s40561-022-00192-z

Yohannes, E., & Ahmed, S. (2018). Prediction of Student Academic Performance using Neural Network, Linear Regression and Support Vector Regression: A Case Study. International Journal of Computer Applications, 180(40), 39–47. https://doi.org/10.5120/ijca2018917057

Zhang, Y., Yun, Y., An, R., Cui, J., Dai, H., & Shang, X. (2021). Educational Data Mining Techniques for Student Performance Prediction: Method Review and Comparison Analysis. Frontiers in Psychology, 12. https://doi.org/10.3389/fpsyg.2021.698490

# APPENDIX I

**QUESTIONNAIRE**

**Factors That Can Affect Academic Performance**

**Instruction**: Being an instructor, it is believed that you are experience and have full knowledge on how the stated factors in the group below affect students’ academic performance. Using a scale of 1 – 5 where 5 is the highest and 1 lowest or 1 -3 where 3 is the highest and 1 lowest for each group. Please grade the following group of question on how they affect academic performance.

***Note: In each group, no two items in a group should be graded with the same score.***

1. **The student reason for schooling**

|  |  |
| --- | --- |
| Personal |  |
| Parents |  |
| Friends |  |
| Frustration |  |
| No reason |  |

1. **The student study habits**

|  |  |
| --- | --- |
| The student studies only when there is test or exam |  |
| The student studies always |  |
| The student is lazy to study |  |
| The student studies only when they feel like |  |
| The student has no time to study |  |

1. **The student report on assignments & classwork**

|  |  |
| --- | --- |
| They copy the assignments of friends |  |
| They get assignment done by themselves |  |
| They pay to get it done |  |
| They don’t do assignment |  |
| They do assignments only when it’s necessary |  |

1. **Availability of learning materials**

|  |  |
| --- | --- |
| The student has all necessary resources |  |
| The student does not have any resources |  |
| The student borrows resources from friends only when necessary |  |

1. **The student motivation to Study**

|  |  |
| --- | --- |
| To pass exams |  |
| To learn |  |
| To end up with a good job |  |

1. **The student** **level of accumulation**

|  |  |
| --- | --- |
| High |  |
| Average |  |
| Low |  |

1. **The student involvement in class activities**

|  |  |
| --- | --- |
| Very Active |  |
| Active |  |
| Not active |  |

1. **The student reason for absenteeism**

|  |  |
| --- | --- |
| Illness |  |
| Tiredness |  |
| Boredom |  |
| Choice |  |
| Prefer not to say |  |

1. **The student level of absenteeism in class**

|  |  |
| --- | --- |
| High |  |
| Average |  |
| Low |  |

1. **Previous academic performance**

|  |  |
| --- | --- |
| High |  |
| Average |  |
| Low |  |

1. **Father’s Education Level**

|  |  |
| --- | --- |
| No formal education |  |
| Primary School |  |
| Secondary School |  |
| OND / HND / BSc |  |
| Post-Graduate & Above |  |

1. **Mother’s Education Level**

|  |  |
| --- | --- |
| No formal education |  |
| Primary School |  |
| Secondary School |  |
| OND / HND / BSc |  |
| Post-Graduate & Above |  |

1. **Marital Status?**

|  |  |
| --- | --- |
| Single |  |
| Married |  |
| Divorced |  |

1. **Parental income**

|  |  |
| --- | --- |
| High |  |
| Average |  |
| Low |  |

1. **Gender**

|  |  |
| --- | --- |
| Male |  |
| Female |  |
| Prefer not to say |  |

1. **Parents’ involvement in academic activities e.g. joint reading**

|  |  |
| --- | --- |
| High |  |
| Average |  |
| Low |  |

1. **Age**

|  |  |
| --- | --- |
| Below 20 |  |
| 20 – 30 |  |
| Above 30 |  |

1. **Who do you live with?**

|  |  |
| --- | --- |
| Single Father |  |
| Single Mother |  |
| Both parent |  |
| Relatives |  |
| Alone |  |

1. **Who pays your Bills?**

|  |  |
| --- | --- |
| Parent |  |
| Myself |  |
| Brother / Sister |  |
| Charity organization |  |
| Relatives |  |

1. **What’s your plan after school?**

|  |  |
| --- | --- |
| Get a job |  |
| Learn a trade |  |
| Get married |  |
| Build myself |  |
| I don’t know |  |

# APPENDIX II

**SOURCE CODE**

#importing the various libraries

from tkinter import \*

from tkinter import ttk, filedialog

from PIL import ImageTk, Image

from openpyxl import \*

from openpyxl import \*

import pathlib

import openpyxl

import pandas as pd

global Prediction

#Admin final window to read prediction result

def admin2():

    #Homepage creation

    outputw = Tk()

    outputw.title('PREDICTION OUTPUT PAGE')

    outputw['bg'] = '#29C5F6'

    outputw.resizable(False, False)  # This code helps to disable windows from resizing

    #setting window default size for home page

    window\_height = 500

    window\_width = 900

    screen\_width = outputw.winfo\_screenwidth()

    screen\_height = outputw.winfo\_screenheight()

    x\_cordinate = int((screen\_width/2) - (window\_width/2))

    y\_cordinate = int((screen\_height/2) - (window\_height/2))

    outputw.geometry("{}x{}+{}+{}".format(window\_width, window\_height, x\_cordinate, y\_cordinate))

    # Frame for TreeView

    frame1 = LabelFrame(outputw, text="Excel Data", bg='#29C5F6')

    frame1.place(height=400, width=900)

    # Frame for open file dialog

    file\_frame = LabelFrame(outputw, text="Open File", bg = '#29C5F6')

    file\_frame.place(height=100, width=900, rely=0.8, relx=0)

    # Buttons

    button1 = Button(file\_frame, text="Browse A File", command=lambda: File\_dialog())

    button1.place(rely=0.65, relx=0.50)

    button2 = Button(file\_frame, text="Load File", command=lambda: Load\_excel\_data())

    button2.place(rely=0.65, relx=0.30)

    # The file/file path text

    label\_file = ttk.Label(file\_frame, text="No File Selected")

    label\_file.place(rely=0, relx=0)

    ## Treeview Widget

    tv1 = ttk.Treeview(frame1)

    tv1.place(relheight=1, relwidth=1) # set the height and width of the widget to 100% of its container (frame1).

    treescrolly = Scrollbar(frame1, orient="vertical", command=tv1.yview) # command means update the yaxis view of the widget

    treescrollx = Scrollbar(frame1, orient="horizontal", command=tv1.xview) # command means update the xaxis view of the widget

    tv1.configure(xscrollcommand=treescrollx.set, yscrollcommand=treescrolly.set) # assign the scrollbars to the Treeview Widget

    treescrollx.pack(side="bottom", fill="x") # make the scrollbar fill the x axis of the Treeview widget

    treescrolly.pack(side="right", fill="y") # make the scrollbar fill the y axis of the Treeview widget

    def File\_dialog():

        """This Function will open the file explorer and assign the chosen file path to label\_file"""

        filename = filedialog.askopenfilename(initialdir="/",

                                            title="Select A File",

                                            filetype=(("xlsx files", "\*.xlsx"),("All Files", "\*.\*")))

        label\_file["text"] = filename

        return None

    def Load\_excel\_data():

        """If the file selected is valid this will load the file into the Treeview"""

        file\_path = label\_file["text"]

        try:

            excel\_filename = r"{}".format(file\_path)

            if excel\_filename[-4:] == ".csv":

                df = pd.read\_csv(excel\_filename)

            else:

                df = pd.read\_excel(excel\_filename)

        except ValueError:

            messagebox.showerror("Information", "The file you have chosen is invalid")

            return None

        except FileNotFoundError:

            messagebox.showerror("Information", f"No such file as {file\_path}")

            return None

        clear\_data()

        tv1["column"] = list(df.columns)

        tv1["show"] = "headings"

        for column in tv1["columns"]:

            tv1.heading(column, text=column) # let the column heading = column name

        df\_rows = df.to\_numpy().tolist() # turns the dataframe into a list of lists

        for row in df\_rows:

            tv1.insert("", "end", values=row) # inserts each list into the treeview. For parameters see https://docs.python.org/3/library/tkinter.ttk.html#tkinter.ttk.Treeview.insert

        return None

    def clear\_data():

        tv1.delete(\*tv1.get\_children())

        return None

    outputw.mainloop()

#End of admin final window

#Admin credential checking

def adminn():

    root.destroy()

    #Homepage creation

    adcre = Tk()

    adcre.title('HOME PAGE')

    adcre['bg'] = '#29C5F6'

    adcre.resizable(False, False)  # This code helps to disable windows from resizing

    #setting window default size for home page

    window\_height = 500

    window\_width = 900

    screen\_width = adcre.winfo\_screenwidth()

    screen\_height = adcre.winfo\_screenheight()

    x\_cordinate = int((screen\_width/2) - (window\_width/2))

    y\_cordinate = int((screen\_height/2) - (window\_height/2))

    adcre.geometry("{}x{}+{}+{}".format(window\_width, window\_height, x\_cordinate, y\_cordinate))

    # Create an object for homepage

    instruction = Label(adcre, text = 'Please Enter Login Credentials', font=('Bahnschrift', 15, 'italic', 'bold'), bg ='#29C5F6',fg = 'Black')

    instruction.grid(columnspan=4, row=0)

    #Declaring Login credentials

    global password

    global username

    #Designing objects for login

    us = Label(adcre, text='USERNAME', font=('Arial', 15, 'bold'), bg ='#29C5F6').grid(row=1, column=0, pady=5)

    ps = Label(adcre, text='PASSWORD', font=('Arial', 15, 'bold'), bg ='#29C5F6').grid(row=2, column=0, pady=10)

    username = Entry(adcre, width=25, font=('Arial 20'))

    username.grid(row=1, column=1)

    password = Entry(adcre, show = '\*', width=25, font=('Arial 20'))

    password.grid(row=2, column=1)

    def second():

        en1 = username.get()

        en2 = password.get()

        if en1 == 'username'  and en2 =='password':

            adcre.destroy()

            admin2()

        else:

            label1 = Label(adcre, text='Invalid Credentials')

            label1.grid(row=4, column=0)

            username.delete(0, END)

            password.delete(0, END)

    Submit = Button(adcre, text = 'Submit',font=('Arial', 15), bg ='Red', width=10, border=8, command=second)

    Submit.grid(row=3,column=0, columnspan=2)

    adcre.mainloop()

#end of admin credential checking

#Homepage creation

root = Tk()

root.title('HOME PAGE')

root['bg'] = '#29C5F6'

root.resizable(False, False)  # This code helps to disable windows from resizing

#setting window default size for home page

window\_height = 500

window\_width = 900

screen\_width = root.winfo\_screenwidth()

screen\_height = root.winfo\_screenheight()

x\_cordinate = int((screen\_width/2) - (window\_width/2))

root.geometry("{}x{}+{}+{}".format(window\_width, window\_height, x\_cordinate, y\_cordinate)

# Create an object for homepage

title1 = Label(root, text = 'DEPARTMENT OF COMPUTER SCIENCE', font=('Pickwick', 20, 'bold' ), fg = '#000080', bg ='#29C5F6', justify=CENTER)

title1.pack()

title2 = Label(root, text = 'FEDERAL POLYTECHNIC BIDA', font=('Bahnschrift', 20), fg = 'Red', bg ='#29C5F6', justify=CENTER)

title2.pack()

title2 = Label(root, text = 'STUDENT ACADEMIC PEFORMANCE SYSTEM', font=('Bahnschrift', 20, 'bold'), fg = 'Black', bg ='#29C5F6',justify=CENTER)

title2.pack()

img = ImageTk.PhotoImage(Image.open("prediction4.png"))

label = Label(root, image = img, bg ='#29C5F6')

label.pack()

instruction = Label(root, text = 'Please Select Appropriate User', font=('Bahnschrift', 15, 'italic', 'bold'), bg ='#29C5F6',fg = 'Black')

instruction.pack()

#Form button creation

def new():

    root.destroy() #closing hompage

    #Loading an existing workbook

    file = pathlib.Path('Prediction.xlsx')

    if file.exists ():

        file=load\_workbook('Prediction.xlsx')

        sheet=file.active

    #creating a workbook for the form

    else:

        file = openpyxl.Workbook()

        sheet = file.active

        sheet.column\_dimensions['A'].width = 30

        sheet.column\_dimensions['B'].width = 20

        sheet.column\_dimensions['C'].width = 20

        sheet.column\_dimensions['E'].width = 20

        sheet.column\_dimensions['F'].width = 40

        sheet.column\_dimensions['G'].width = 50

        sheet.column\_dimensions['H'].width = 50

        # write given data to an excel spreadsheet

        # at particular location

        sheet.cell(row=1, column=1).value = "Name"

        sheet.cell(row=1, column=2).value = "Course"

        sheet.cell(row=1, column=3).value = "Semester"

        sheet.cell(row=1, column=4).value = "Matric No."

        sheet.cell(row=1, column=6).value = "Email id"

        sheet.cell(row=1, column=7).value = "Address"

        sheet.cell(row=1, column=8).value = "Prediction Analysis"

    def excel():

            sheet.column\_dimensions['A'].width = 30

            sheet.column\_dimensions['B'].width = 20

            sheet.column\_dimensions['C'].width = 20

            sheet.column\_dimensions['D'].width = 20

            sheet.column\_dimensions['E'].width = 20

            sheet.column\_dimensions['F'].width = 40

            sheet.column\_dimensions['H'].width = 50

            # write given data to an excel spreadsheet

            # at particular location

            sheet.cell(row=1, column=1).value = "Name"

            sheet.cell(row=1, column=2).value = "Course"

            sheet.cell(row=1, column=3).value = "Semester"

            sheet.cell(row=1, column=4).value = "Matric No."

            sheet.cell(row=1, column=5).value = "Phone Number"

            sheet.cell(row=1, column=7).value = "Address"

            sheet.cell(row=1, column=8).value = "Prediction Analysis"

    # Function to set focus (cursor)

    def focus1(event):

        # set focus on the course\_field box

        name\_field.focus\_set()

    def focus2(event):

        # set focus on the course\_field box

        course\_field.focus\_set()

        # Function to set focus

    def focus3(event):

        # set focus on the sem\_field box

        sem\_field.focus\_set()

        # Function to set focus

    def focus4(event):

        # set focus on the form\_no\_field box

        mat\_no\_field.focus\_set()

    # Function to set focus

    def focus5(event):

        # set focus on the contact\_no\_field box

        phone\_no\_field.focus\_set()

    # Function to set focus

    def focus6(event):

        # set focus on the email\_id\_field box

        email\_id\_field.focus\_set()

    # Function to set focus

    def focus7(event):

        # set focus on the address\_field box

        address\_field.focus\_set()

    # Function to set focus

    def focus8(event):

        # set focus on the address\_field box

        Prediction.focus\_set()

    # Function for clearing the

    # contents of text entry boxes

    def clear():

        # clear the content of text entry box

        name\_field.delete(0, END)

        course\_field.delete(0, END)

        sem\_field.delete(0, END)

        mat\_no\_field.delete(0, END)

        phone\_no\_field.delete(0, END)

        email\_id\_field.delete(0, END)

        address\_field.delete(0, END)

        # Function to take data from GUI

    # window and write to an excel file

    def insert():

        # if user not fill any entry

        # then print "empty input"

        if (name\_field.get() == "" and

            course\_field.get() == "" and

            sem\_field.get() == "" and

            mat\_no\_field.get() == "" and

            email\_id\_field.get() == "" and

            address\_field.get() == ""):

            print("empty input")

        else:

            # assigning the max row and max column

            # value upto which data is written

            # in an excel sheet to the variable

            current\_row = sheet.max\_row

            current\_column = sheet.max\_column

            # get method returns current text

            # as string which we write into

            # excel spreadsheet at particular location

            sheet.cell(row=current\_row + 1, column=1).value = name\_field.get()

            sheet.cell(row=current\_row + 1, column=2).value = course\_field.get()

            sheet.cell(row=current\_row + 1, column=3).value = sem\_field.get()

            sheet.cell(row=current\_row + 1, column=4).value = mat\_no\_field.get()

            sheet.cell(row=current\_row + 1, column=5).value = phone\_no\_field.get()

            sheet.cell(row=current\_row + 1, column=6).value = email\_id\_field.get()

            sheet.cell(row=current\_row + 1, column=7).value = address\_field.get()

            # save the file

            # set focus on the name\_field box

            name\_field.focus\_set()

            # call the clear() function

            clear()

        root1.destroy()

        # Create questions as a list of dictionaries

        # The GUI will be generated automatically

        questions = [

            {"question": "1. Father’s Education Level?",

            "answers": ('',"No formal education", "Primary Education", "Secondary Education", "Graduate (OND/BSc. etc)", "Masters & Above")},

            {"question": "2. Mother’s Education Level?",

            "answers": ('',"No formal education", "Primary Education", "Secondary Education", "Graduate (OND/BSc. etc)", "Masters & Above")},

            {"question": "3. Marital Status?",

            "answers": ('',"Divorced", "Married", "Single")},

            {"question": "4. Parental income?",

            "answers": ('', "Low", "High", "Average")},

            {"question": "5. Gender?",

            "answers": ('', "Female", "Male")},

            {"question": "6. Parents’ involvement in academic activities e.g joint reading?",

            "answers": ('', "Low", "High", "Average")},

            {"question": "7. Age",

            "answers": ('',"Above 30", "Below 20", "20 - 30")},

            {"question": "8. Who do you live with?",

            "answers": ('', "Single Father", "Single Mother", "Relatives", "Alone", "Both parents")},

            {"question": "9. Who pays your bills?",

            "answers": ('', "Charity Organization", "Relatives", "Myself", "Brother / Sister", "Parent")},

            {"question": "10. What’s your plan after school?",

            "answers": ('', "I don't Know", "Get Married", "Get a Job", "Learn a Trade", "Build Myself")},

            {"question": "11. Reason for schooling?",

            "answers": ('', "No Reason", "Frustration", "Friends", "Parent", "Personal")},

            {"question": "12. Study habits?",

            "answers": ('', "I have no time to study", "I study only when I like", "I am lazy to study", "I study only it’s necessary ", "I am lazy to study")},

            {"question": "13. Report on assignments & classwork?",

            "answers": ('', "I get my  assignment done by myself", "Copy the assignments of friends", "I pay to get it done", "I do assignment only when its necessary", "I don't do assignment",)},

            {"question": "14.Availability of learning materials?",

            "answers": ('', "The student does not have any resources", "The student borrows resources from friends only when necessary", "The student has all necessary resources")},

            {"question": "15. What's your motivation to Study",

            "answers": ('', "To end up with a good job", "To learn", "To pass exams ")},

            {"question": "16.   How would you rate your level of accumulation?",

            "answers": ('', "Low", "High", "Low")},

            {"question": "17.   How would you rate your invovlement in class activities ",

            "answers": ('', "Not Active", "Very Active", "Active")},

            {"question": "18. What's your reason for absenteeism in school? ",

            "answers": ('', "Prefer not to say", "Tiredness", "Boredom", "Choice", "Illness")},

            {"question": "19. What's your level of absenteeism in school?",

            "answers": ('', "High", "Average", "Low")},

            {"question": "20. Previous Academic Performance?",

            "answers": ('', "Low", "Average", "High")},

            ]

        class App(Tk):

            def \_\_init\_\_(self):

                super().\_\_init\_\_()

                self.title("PREDICTION QUESTIONS")  # set the window title

                self.resizable(False, True)  # make window unresizable by width

                canv\_frame = Frame(self)  # create the canvas frame

                # create the Canvas widget

                # highlightthickness=0 removes the black border when the canvas gets focus

                self.canv = Canvas(canv\_frame, highlightthickness=0, height=500, width=800)

                # add scrolling when mouse wheel is rotated

                self.canv.bind\_all("<MouseWheel>",

                                lambda event: self.canv.yview\_scroll(-1 \* (event.delta // 120), "units"))

                self.canv.pack(fill=BOTH, expand=YES, side=LEFT)  # pack the Canvas

                # Create a scrollbar

                # command=self.canv.yview tells the scrollbar to change the canvas yview

                # and canvas's yscrollcommand=self.yscrollbar.set tells the canvas to update

                # the scrollbar if canvas's yview is changed without it.

                self.yscrollbar = Scrollbar(canv\_frame, command=self.canv.yview)

                self.canv["yscrollcommand"] = self.yscrollbar.set

                for question\_id, question in enumerate(questions, 1):

                    qaframe = Frame(self.canv)  # create the question-answers (QA) frame

                    text = Text(qaframe, width=100, height=1.2, bg='orange', fg='white')  # create the Text widget for question

                    text.insert(END, question["question"])  # insert the question text there

                    text.pack(fill=X)  # pack the text widget

                    aframe = Frame(qaframe)  # create the answers frame

                    # Create the question variable and add it to the variables list

                    question\_var = IntVar(self)

                    question["variable"] = question\_var

                    # create the radiobuttons

                    for answer\_id, answer in enumerate(question["answers"]):

                        Radiobutton(aframe, variable=question\_var, text=answer, value=answer\_id).pack()

                    aframe.pack(fill=Y)  # pack the answers frame

                    self.canv.create\_window(210, question\_id \* 175, window=qaframe)  # insert the QA frame into the Canvas

                canv\_frame.pack(fill=BOTH, expand=YES)  # pack the canvas frame

                Button(self, text="Submit", command=self.submit, fg='Red', bg='Black', font=(12)).pack(fill=X)  # create the "Submit" button

                self.update()  # update everything to get the right scrollregion.

                self.canv.configure(scrollregion=self.canv.bbox("all"))  # set the canvas scrollregion

            def submit(self):

                sum\_ = 0  # initially, the sum\_ equals 0

                for question in questions:

                    sum\_ += question["variable"].get()  # and then, we add all the questions answers

                if sum\_>=80:

                    pout = 'High Performance'

                elif sum\_>= 55:

                    pout = 'Average Performance'

                else:

                    pout = 'Low Performance'

                sheet.cell(row=current\_row + 1, column=8).value =pout

                file.save("Prediction.xlsx")

                self.destroy()

        if \_\_name\_\_ == "\_\_main\_\_":  # if the App is not imported from another module,

            App().mainloop()  # create it and start the mainloop

    # Driver code

    if \_\_name\_\_ == "\_\_main\_\_":

        # create a GUI window

        root1 = Tk()

        # set the background colour of GUI window

        root1.configure(background='light green')

        # set the title of GUI window

        root1.title("registration form")

        # set the configuration of GUI window

        window\_height = 500

        window\_width = 800

        screen\_width = root1.winfo\_screenwidth()

        screen\_height = root1.winfo\_screenheight()

        x\_cordinate = int((screen\_width/2) - (window\_width/2))

        y\_cordinate = int((screen\_height/2) - (window\_height/2))

        root1.geometry("{}x{}+{}+{}".format(window\_width, window\_height, x\_cordinate, y\_cordinate))

        excel()

        # create a Form label

        heading = Label(root1, text="ENTER THE FOLLOWING INFORMATION", bg="light green", font=('Bahnschrift', 20))

        # create a Name label

        name = Label(root1, text="Name", bg="light green", font=('Bahnschrift', 14))

        # create a Course label

        course = Label(root1, text="Course", bg="light green", font=('Bahnschrift', 14))

        # create a Semester label

        sem = Label(root1, text="Semester", bg="light green", font=('Bahnschrift', 14))

        # create a Form No. label

        mat\_no = Label(root1, text="Matric No.", bg="light green", font=('Bahnschrift', 14))

        # create a Contact No. label

        phone\_no = Label(root1, text="Phone No.", bg="light green", font=('Bahnschrift', 14))

        # create a Email id label

        email\_id = Label(root1, text="Email id", bg="light green", font=('Bahnschrift', 14))

        # create a address label

        address = Label(root1, text="Address", bg="light green", font=('Bahnschrift', 14))

        # grid method is used for placing

        # the widgets at respective positions

        # in table like structure .

        heading.grid(row=0, column=1)

        name.grid(row=1, column=0)

        course.grid(row=2, column=0)

        sem.grid(row=3, column=0)

        mat\_no.grid(row=4, column=0)

        phone\_no.grid(row=5, column=0)

        email\_id.grid(row=6, column=0)

        address.grid(row=7, column=0)

        # create a text entry box

        # for typing the information

        name\_field = Entry(root1, font=(25))

        course\_field = Entry(root1, font=(25))

        sem\_field = Entry(root1, font=(25))

        mat\_no\_field = Entry(root1, font=(25))

        phone\_no\_field = Entry(root1, font=(25))

        email\_id\_field = Entry(root1, font=(25))

        address\_field = Entry(root1, font=(25))

        # bind method of widget is used for

        # the binding the function with the events

        # whenever the enter key is pressed

        # then call the focus1 function

        name\_field.bind("<Return>", focus1)

        # whenever the enter key is pressed

        # then call the focus2 function

        course\_field.bind("<Return>", focus2)

        # whenever the enter key is pressed

        # then call the focus3 function

        sem\_field.bind("<Return>", focus3)

        # whenever the enter key is pressed

        # then call the focus4 function

        mat\_no\_field.bind("<Return>", focus4)

        # whenever the enter key is pressed

        # then call the focus5 function

        phone\_no\_field.bind("<Return>", focus5)

        # whenever the enter key is pressed

        # then call the focus6 function

        email\_id\_field.bind("<Return>", focus6)

        # whenever the enter key is pressed

        # then call the focus6 function

        address\_field.bind("<Return>", focus7)

        # grid method is used for placing

        # the widgets at respective positions

        # in table like structure .

        name\_field.grid(row=1, column=1, ipadx="200", ipady="10", pady=5)

        course\_field.grid(row=2, column=1, ipadx="200", ipady="10", pady=5)

        sem\_field.grid(row=3, column=1, ipadx="200", ipady="10", pady=5)

        mat\_no\_field.grid(row=4, column=1, ipadx="200", ipady="10", pady=5)

        phone\_no\_field.grid(row=5, column=1, ipadx="200", ipady="10", pady=5)

        email\_id\_field.grid(row=6, column=1, ipadx="200", ipady="10", pady=5)

        address\_field.grid(row=7, column=1, ipadx="200", ipady="10", pady=5 )

        # call excel function

        #Function to go back to home page

        # create a Submit Button and place into the root1 window

        submit = Button(root1, text="Submit", fg="White",font=('Arial', 15),

                                 bg ='Green', width=20, border=8, command=insert)

        submit.grid(row=8, column=1)

        # start the GUI

        root1.mainloop()

admin = Button(root, text = 'Administrator',font=('Arial', 15), bg ='#5e1914', width=20, border=8, command=adminn)

admin.pack()

user = Button(root, text = 'User', font=('Arial', 15),  bg ='#29ab87', width=20, border=8, command = new,)

user.pack()

# Create a Label Widget to display the text or Image

root.mainloop()

a