A Project Report on

Dropout Defender: A Machine Learning Approach to Lower Dropout Rates

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering
in
Information Technology

by

Harmi Mathukiya	**21104044**
Avantika More	**21104033**
Sahil Mohite	**21104099**
Atharva Mohape	**21104121**

Under the Guidance of

Ms.Apeksha Mohite

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B. Road, Kasarvadavli, Thane (W) - 400615
UNIVERSITY OF MUMBAI

Academic Year 2024-2025

Approval Sheet

This Project Report entitled ***"Dropout Defender: A Machine Learning Approach to Lower Dropout Rates"*** submitted by ***"Harmi Mathukiya" (21104044) , "Avantika More" (21104033) , "Sahil Mohite" (21104099) , "Atharva Mohape" (21104121)*** is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Ms.Apeksha Mohite
Guide

Dr. Kiran Deshpande
HOD, Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Dropout Defender: A Machine Learning Approach to Lower Dropout Rates"*** submitted by ***"Harmi Mathukiya" (21104044) , "Avantika More" (21104033) , "Sahil Mohite" (21104099) , "Atharva Mohape" (21104121)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***, to the University of Mumbai, is a bonafide work carried out during the academic year 2024-2025.

Ms.Apeksha Mohite
Guide

Dr. Kiran Deshpande
HOD, Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Internal Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **Dropout Defender: A Machine Learning Approach to Lower Dropout Rates.** We take this opportunity to express our sincere thanks towards our guide **Ms.Apeksha Mohite** for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Dr. Kiran B. Deshpande** Head of Department for his encouragement during the progress meeting and for providing guidelines to write this report.

We express our gratitude towards BE project co-ordinators, for being encouraging throughout the course and for their guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Harmi Mathukiya
(21104044)

Avantika More
(21104033)

Sahil Mohite
(21104099)

Atharva Mohape
(21104121)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Harmi Mathukiya (21104044)

Avantika More (21104033)

Sahil Mohite (21104099)

Atharva Mohape (21104121)

Date:

Abstract

"Dropout Defender: A Machine Learning Approach to Lower Dropout Rates" is a comprehensive web application designed to predict and reduce student dropout rates by leveraging machine learning techniques. The system offers tailored dashboards for students, mentors, and parents, enabling each user role to interact with the platform according to their specific needs. The platform integrates real-time data analysis, predictive models, and progress tracking to offer personalized insights and interventions, fostering student retention. The application is designed for deployment on scalable platforms, ensuring efficient and seamless user experience. The project utilizes Firebase for backend data storage, ensuring flexibility and scalability, while the front-end interface is built using modern web technologies. The system is designed to be user-friendly, providing role-based access to key features such as progress tracking, material uploads, and behavior analysis. By combining machine learning predictions with a user-friendly dashboard, Dropout Defender aims to reduce dropout rates, promote student retention, and foster a supportive learning environment where students, mentors, and parents work together toward academic success. This platform not only predicts at-risk students but also offers tools and strategies for early intervention, making it a valuable asset for educational institutions seeking to improve student outcomes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The "Dropout Defender: A Machine Learning Approach to Lower Dropout Rates" project addresses one of the most pressing issues in modern education systems-high student dropout rates. In many regions, this trend negatively impacts both individual lives and broader societal growth. Dropouts face increased challenges in job opportunities, social integration, and personal growth, which ultimately affects community development. Therefore, early intervention and continuous support mechanisms are crucial to preventing students from leaving their educational pursuits prematurely. This project uses predictive modeling to identify students at risk of dropping out, supporting timely and effective action.

The Dropout Defender project is a website that employs machine learning algorithms to predict dropout risk based on academic, behavioral, and engagement data. By analyzing historical student data, the system can recognize patterns and warning signs indicative of potential dropout scenarios. This insight allows for personalized intervention strategies tailored to each student's needs, making it possible to provide support that can help retain students within the educational system. Predictive modeling, combined with real-time data analysis, forms the backbone of this application, offering a proactive solution to the dropout issue.

The Dropout Defender platform features a dynamic, user-friendly interface with separate dashboards for students, parents, and mentors. Each user type can access specialized resources and monitoring tools that cater to their role in supporting student success. For instance, students can view their progress and practice logs, mentors can offer guidance and track student development, and parents have insights into their child's academic status and behavior. These tailored views foster a collaborative environment where stakeholders can work together to address challenges that may otherwise go unnoticed. In essence, Dropout Defender is more than a predictive tool; it is a comprehensive system that leverages data science, user-friendly design, and collaborative support to lower dropout rates and improve educational outcomes. By engaging all stakeholders in the process, the project aims to create an effective support network that helps students thrive. With a fully connected and interactive platform, Dropout Defender aspires to make a tangible impact on dropout prevention, contributing to the future success of students and the educational system as a whole.

1.1 Motivation

Reducing Dropout Rates: Educational dropout is a major concern in many countries. By implementing a machine learning-based system that can predict potential dropout risks, we can intervene before it's too late, offering students the necessary resources and support to continue their education. Personalized Support for Students: Every student faces unique challenges. This system allows mentors to monitor student progress and provide personalized guidance, ensuring that students receive the support they need to overcome difficulties, stay motivated, and succeed academically. Parental Involvement: Parents often play a crucial role in a student's academic journey. However, many parents lack real-time insight into their child's academic performance. This project enables parents to stay informed about their child's progress and helps them be more involved in their education, promoting a collaborative effort between students, mentors, and parents

1.2 Problem Statement

This project aims to address the issue of high college dropout rates by identifying students at risk of leaving early. By using machine learning algorithms, the project analyzes past student data to predict dropout risks, allowing early intervention to support students and help them stay in college.

1.2.1 Consequences of the Problem:

The high number of students dropping out of college leads to several negative consequences at both individual and societal levels. For students, dropping out significantly reduces future opportunities for employment, limits earning potential, and increases the likelihood of facing financial instability. It also negatively affects personal development and self-esteem, leading to long-term impacts on mental health. At the societal level, high dropout rates result in a less educated workforce, which can hinder economic growth and innovation. Additionally, dropouts often place a higher burden on social welfare systems and are more likely to engage in criminal activities, contributing to broader social challenges.

1.2.2 Significance of the Problem:

Addressing the dropout issue is crucial for creating a more inclusive, equitable, and thriving education system. By developing a model that uses machine learning algorithms to forecast dropout rates, this project seeks to identify at-risk students early on, providing timely interventions and support to prevent them from leaving school. This proactive approach can lead to higher graduation rates, improved student outcomes, and a more skilled and educated population. The significance of solving this problem lies in its potential to not only uplift individual lives but also contribute to broader societal well-being and economic development by reducing the long-term costs associated with educational failure.

1.3 Objectives

After in-depth discussions on the literature survey and project requirements, the system is designed to achieve the following objectives:

- **Dropout Prediction Model:** To create a model that can predict whether a student is likely to drop out of an academic program.

- **ML Classification:** To compare different ML algorithms to find the most accurate one for predicting dropouts.

- **Student Dashboards:** To create different dashboards such as students, teachers and parents to view and upload the progress of the student.

- **Secure Access:** To implement login and registration for students, parents, and mentors to maintain secure access.

- **Easy Navigation:** To ensure easy navigation between dashboards, grade reports, and communication tools.

1.4 Scope

The scope of this project encompasses comprehensive analysis and prediction of student dropout rates by leveraging data collected from multiple academic years across various institutions. Key functionalities include administrative access for data management, along with filtering capabilities to analyze dropout rates, retention rates, and overall student counts. The project:

- Can predict student dropouts using machine learning models by analyzing academic performance, assignments and grades.

- Can compare multiple ML classifiers to determine the most accurate algorithm for predicting dropouts.

- Can provide interactive dashboards for students, parents, and mentors to track progress, upload assignments, and receive insights.

- Can implement secure login and registration to ensure role-based access for students, parents, and mentors.

- Can facilitate easy navigation between dashboards, grade reports, and communication tools for an improved user experience.

Chapter 2

Literature Review

A literature review on student dropout prediction systems reveals various approaches leveraging machine learning and data analytics to reduce dropout rates. Recent studies have explored predictive models like Random Forest, Decision Tree, and Neural Networks, with Decision Tree often achieving the highest accuracy (over 90).

2.1 Literature Review Summary

1. **Title:** Predicting Student Dropout in Higher Education Using Machine Learning Techniques
   Authors: O. B. D. Ayoob, R. Shah, A. Ali
   Year: 2022
   Findings: The paper "Predicting Student Dropout in Higher Education Using Machine Learning Techniques" presents an in-depth analysis of factors influencing student dropout and evaluates the effectiveness of machine learning (ML) in accurately predicting dropout risk. Researchers examined various academic, behavioral, and demographic variables, such as attendance records, grades, engagement metrics, financial aid status, and prior academic performance. Through data preprocessing and feature selection, the authors identified key predictors, including consistent attendance, academic scores, and financial support, as strongly correlated with retention outcomes.One key finding was that socioeconomic factors, such as financial need and family support, had a significant influence on dropout rates. Students from low-income backgrounds showed a notably higher dropout risk, underscoring the importance of financial aid and support services as retention tools. Additionally, academic metrics like GPA and coursework completion rates were strong predictors, suggesting that early academic support could be crucial in reducing dropout rates.
   Academic indicators, particularly GPA and coursework completion rates, also emerged as strong predictors of dropout, suggesting that early academic intervention and support could be crucial in helping at-risk students.
   The paper highlights the high accuracy of machine learning models in predicting student dropout, which could be a valuable tool for educational institutions in identifying at-risk students. By providing actionable insights based on real-time data, institutions can implement targeted interventions to prevent dropout. However, the study also acknowledges that its findings are specific to the institutions involved in the research, and the results may not necessarily generalize across different educational contexts or

settings. Despite this limitation, the research demonstrates the potential of machine learning techniques in predicting and addressing student dropout in higher education.

Advantages: Demonstrates high accuracy in predictions; provides actionable insights for educational institutions.
Disadvantages: Limited to specific institutions; results may not generalize across different contexts.

2. **Title:** Analyzing Student Dropout Factors Using Predictive Analytics
Authors: C. T. Munoz, J. A. Asensio, M. A. González
Year: 2023
Findings: The analysis of student dropout factors using predictive analytics reveals crucial insights into the variables influencing student retention rates. By examining demographic, academic, and behavioral data, key predictors of dropout risk emerge, allowing for targeted interventions. Findings indicate that a combination of academic performance metrics—such as grades, attendance, and engagement levels—play a significant role in determining the likelihood of student dropout. Students with lower academic achievements, frequent absences, and minimal participation in classroom or extracurricular activities tend to show higher dropout risks. Behavioral data, including disciplinary records and interactions with mentors or counselors, also contribute to understanding dropout patterns, highlighting the importance of early behavioral intervention.The analysis further uncovers that socio-economic factors, such as parental education levels and household income, can impact a student's continuity in education.
Students from low-income families or those with limited educational support often face more challenges, affecting their performance and engagement in school. Predictive models applied in this study demonstrate a high accuracy rate in identifying at-risk students by combining these academic, socio-economic, and behavioral indicators, which helps educational institutions proactively address potential dropout cases.
Machine learning techniques, such as decision trees and logistic regression, have been particularly effective in modeling these complex relationships, allowing for dynamic adjustments based on real-time data. These students often lack access to educational support systems, which can negatively affect their academic performance and engagement. The researchers employed machine learning models—particularly decision trees and logistic regression to analyze and predict dropout risk with high accuracy. These models allowed for a nuanced understanding of how different variables interact and helped create systems for dynamic and data-driven interventions.

Advantages: Provides a comprehensive analysis of factors affecting dropout; useful for targeted interventions.
Disadvantages: Relies heavily on available data quality; may overlook less quantifiable factors.

3. **Title:** Machine Learning Approaches for Student Dropout Prediction: A Systematic Review
Authors: P. M. Sharma, A. B. Rani, N. K. Gupta
Year: 2023

Findings: The paper "Machine Learning Approaches for Student Dropout Prediction: A Systematic Review" provides a comprehensive analysis of the different machine learning (ML) techniques applied to predict student dropout rates, emphasizing the role of ML models in educational settings. The findings highlight that dropout prediction models have evolved significantly with the rise of advanced algorithms, offering institutions tools to identify at-risk students early. The study categorizes various ML methods, such as supervised, unsupervised, and hybrid models, and examines their effectiveness, comparing approaches like decision trees, logistic regression, neural networks, support vector machines, and ensemble methods.
Among the key findings, supervised learning techniques are noted as particularly effective due to their ability to classify students based on historical data. Techniques like decision trees and logistic regression are commonly applied in educational datasets due to their interpretability and straightforward implementation. However, the review also emphasizes that more complex models, such as deep neural networks and ensemble methods, offer greater predictive accuracy by capturing complex patterns, albeit at the expense of interpretability. The authors underscore that ensemble methods, which combine predictions from multiple algorithms, often yield the highest accuracy but may require extensive computational resources.
Furthermore, the paper explores more complex techniques, such as neural networks and ensemble methods. While these advanced models can achieve higher predictive accuracy by identifying intricate relationships in the data, they are often criticized for their "black box" nature and greater computational demands. Notably, ensemble methods, which aggregate predictions from multiple models, are shown to outperform individual algorithms in terms of accuracy, making them suitable for high-stakes prediction tasks despite their resource intensity. The authors also point out that while many ML models are effective in theory and controlled studies, there remains a need for real-world validation and practical implementation in diverse educational settings.
One of the major strengths of this paper lies in its comprehensive review of the literature, offering valuable insights into the current state of research and highlighting gaps that future studies can address such as the need for models that balance interpretability and performance.
Advantages: Offers a broad overview of existing research; identifies gaps for future studies.
Disadvantages: Lacks primary data; primarily theoretical.

4. **Title:** Early Detection of At-Risk Students: A Machine Learning Perspective
   Authors: F. A. Taleb, L. P. Trivedi, J. K. Singh
   Year: 2022
   Findings: In the paper titled "Early Detection of At-Risk Students: A Machine Learning Perspective," the authors explore the use of machine learning techniques to identify students at risk of dropping out or underperforming academically. The study highlights the potential of machine learning models to analyze diverse data sources, including academic records, attendance, demographic information, and behavioral data, to generate predictions on student outcomes. By identifying patterns in historical data, machine learning models can forecast which students are likely to face academic challenges, allowing educators to intervene proactively.

Key findings of the research underscore the effectiveness of algorithms like logistic regression, decision trees, random forests, and neural networks in predicting student dropout with high accuracy. Each model's performance varied depending on the specific features and data quality available, with ensemble methods like random forests often outperforming simpler algorithms due to their ability to capture complex relationships within the data.
A key strength of this work is its focus on early intervention, highlighting how predictive analytics can shift educational strategies from reactive to proactive. This approach not only benefits students by offering timely support but also aids institutions in better resource allocation. Nonetheless, the study acknowledges certain limitations. Effective implementation of these models requires ongoing data collection, which may strain institutional resources. Additionally, the handling of sensitive student data raises concerns around privacy and ethical data usage, calling for robust safeguards and transparent practices. Overall, the paper provides a valuable perspective on how machine learning can enhance student success through early detection and targeted support mechanisms.
Advantages: Focuses on proactive measures; could lead to timely interventions.
Disadvantages: Requires consistent data collection; potential privacy concerns.

5. **Title:** Understanding the Impact of Social Factors on Student Retention
   Authors: K. R. Fernando, A. M. Verma, H. N. Khan
   Year: 2023
   Findings: The impact of social factors on student retention is profound, influencing students' decisions to continue or leave their academic pursuits. Research reveals that strong support networks, peer relationships, family involvement, and mentorship significantly contribute to a student's likelihood of persisting in their studies. Social integration within the academic environment, particularly through positive interactions with peers and faculty, is associated with higher retention rates.
   Students who feel a sense of belonging and connection are more likely to engage with academic resources, participate in campus activities, and seek academic help when needed, all of which strengthen their commitment to their studies.Family involvement, such as encouragement from parents or guardians, also plays a critical role, particularly for younger students or those experiencing challenges. Family support provides emotional reassurance and often influences students' educational aspirations and resilience in overcoming obstacles. Similarly, mentorship from faculty or older students can provide valuable guidance, help with academic or personal issues, and encourage goal setting, which enhances retention rates.
   Mentorship is another critical factor discussed in the research. Guidance from faculty or senior students helps at-risk individuals set clear academic goals, manage stress, and find direction, all of which are crucial for retention. However, while the study effectively highlights the qualitative impact of these social elements, it also acknowledges certain limitations. The findings are largely context-dependent and may not easily translate into quantifiable data for predictive models. Moreover, the effects of social factors can vary widely based on cultural and institutional contexts, making generalization difficult.Despite these challenges, the paper makes a compelling case for the inclusion of social dimensions in dropout prevention strategies. It encourages institutions to adopt

more holistic intervention approaches that go beyond academic metrics, recognizing the human and emotional aspects of student success.
Advantages: Highlights the role of social factors in retention; informs holistic intervention strategies.
Disadvantages: May not provide quantifiable data for modeling; findings could vary significantly across cultures.

6. **Title:** A Hybrid Model for Student Dropout Prediction Based on Machine Learning and Statistical Analysis
   Authors: T. Y. N. Neelam, R. K. Gupta, S. A. Ranjan
   Year: 2023
   Findings: The study "A Hybrid Model for Student Dropout Prediction Based on Machine Learning and Statistical Analysis" presents a novel approach to predict student dropout rates through a combination of machine learning techniques and statistical methods. The authors explored the integration of statistical analysis with machine learning algorithms to improve accuracy and early detection of at-risk students. By analyzing academic performance, attendance records, and demographic data, the hybrid model leverages statistical tools to identify patterns and tendencies within the data, which are then further analyzed by machine learning models for predictive insights.
   Key findings from the study indicate that the hybrid model outperforms traditional single-method approaches, providing a more nuanced understanding of the risk factors associated with student dropout. Statistical techniques like regression analysis and correlation mapping helped identify primary risk indicators, such as low academic performance, frequent absenteeism, and socioeconomic challenges. Machine learning algorithms, particularly decision trees, random forests, and support vector machines, were used to process this refined data and provide predictions with high precision. The model's ability to dynamically adapt to new data was another significant finding, highlighting its robustness in educational environments where student profiles and risk factors may change over time.
   The main advantage of this model lies in its dual strength—it merges the depth and interpretability of statistical methods with the predictive power of machine learning, ultimately offering a more comprehensive and dynamic solution for dropout prediction. However, the complexity of implementation is a significant drawback. Successful application of the hybrid model demands interdisciplinary expertise, as it requires thorough knowledge of both statistical inference and ML algorithm optimization. Despite this, the study contributes valuable insights into the development of predictive systems that are both accurate and explainable, making it a promising direction for educational institutions looking to enhance student retention.

   Advantages: Leverages strengths of both approaches; increases reliability of predictions.
   Disadvantages: Complexity in implementation; requires expertise in both statistical and machine learning techniques.

7. **Title:** Real-time Monitoring of Student Engagement and Dropout Prediction
   Authors: S. K. M. Adhikari, L. M. Patil, N. R. Joshi

Year: 2022
Findings: The study on "Real-time Monitoring of Student Engagement and Dropout Prediction" presents significant findings on using real-time data to enhance student retention by monitoring engagement levels and predicting dropout risks. The research demonstrates that real-time data analysis and predictive modeling can serve as powerful tools in early identification of students at risk of dropping out, allowing for timely interventions. The findings indicate that monitoring student engagement, such as attendance, participation, assignment submission, and interaction with digital resources, correlates strongly with academic success.
This approach highlights how continuous tracking of these activities can reveal patterns associated with disengagement, often an early indicator of dropout risk.The study's predictive models utilize machine learning algorithms to analyze large datasets, considering factors like demographic information, academic performance, behavioral patterns, and socio-economic background, to accurately forecast potential dropout cases. By incorporating real-time data, these models continuously adjust to reflect students' current engagement levels, increasing the accuracy and relevance of predictions. Key findings also suggest that intervention strategies tailored to individual students, informed by real-time data insights, yield better outcomes than generalized interventions. For instance, personalized recommendations based on a student's engagement profile encourage re-engagement, improve academic performance, and promote a more supportive educational environment.
This leads to more timely and effective support for students. However, the study also points out some challenges, particularly the need for continuous data collection. The reliance on ongoing monitoring can create challenges for educators, including the potential for data overload. Teachers and administrators may be overwhelmed by the volume of data, which could detract from their ability to respond effectively to individual students' needs. Despite these challenges, the paper offers a promising solution to improving student retention by emphasizing the importance of real-time data and personalized interventions.
Advantages: Immediate feedback on student engagement; facilitates proactive support.
Disadvantages: Depends on continuous data collection; may lead to data overload for educators.

8. **Title:** An Integrated Framework for Predicting Student Dropout in Online Education
Authors: J. A. F. de Lima, M. L. C. da Silva, P. R. D. Sousa
Year: 2023
Findings: The paper titled "An Integrated Framework for Predicting Student Dropout in Online Education" presents a comprehensive analysis of the factors contributing to student dropout rates in online learning environments. The authors developed a robust predictive framework that integrates multiple machine learning algorithms and data mining techniques to identify at-risk students effectively. The study found that various demographic, academic, and behavioral factors significantly impact student retention. Key findings indicated that students' engagement levels, course completion rates, and interaction with peers and instructors were critical predictors of dropout likelihood. The analysis revealed that students who actively participated in discussions and sub-

mitted assignments on time were less likely to drop out compared to their less engaged counterparts.
Additionally, the study highlighted the importance of early intervention strategies based on the predictive outcomes. By utilizing real-time data analytics, the framework allows educators to identify students who exhibit risk factors early in their course tenure. The results showed that timely interventions, such as personalized communication and targeted support resources, can substantially improve retention rates. The authors also emphasized the need for continuous monitoring and adaptive learning pathways to cater to diverse student needs, thus enhancing the overall online learning experience.
These findings emphasize the importance of fostering a collaborative and interactive online learning environment, where students feel involved and connected, as a strategy for reducing dropout rates.

Additionally, the paper emphasizes the role of early intervention. By leveraging real-time data analytics, the proposed framework allows educators to detect students exhibiting risk factors early in their course tenure, enabling timely support. Personalized communication and targeted support resources—such as tutoring, mentorship, and counseling—can help address individual student needs and improve retention rates. The authors also highlight the importance of adaptive learning pathways, which can be tailored to each student's learning pace and preferences, further enhancing the online learning experience.

Advantages: Addresses specific challenges in online education; promotes tailored strategies.
Disadvantages: Limited to online education settings; results may not apply to traditional learning environments.

2.2 Comparative Analysis on Recent Study

Sr No.	Title	Author(s)	Year	Findings
1	Predicting Student Dropout in Higher Education Using Machine Learning Techniques	O. B. D. Ayoob, R. Shah, A. Ali	2022	Identified academic, behavioral, and financial factors as major predictors. Socioeconomic status and GPA strongly influence dropout risk.
2	Analyzing Student Dropout Factors Using Predictive Analytics	C. T. Munoz, J. A. Asensio, M. A. González	2023	Academic performance, behavior, and socio-economic status are crucial. Predictive analytics allow early intervention.
3	Machine Learning Approaches for Student Dropout Prediction: A Systematic Review	P. M. Sharma, A. B. Rani, N. K. Gupta	2023	Explores various ML techniques, identifying supervised and ensemble methods as highly accurate. Emphasizes model interpretability vs complexity.
4	Early Detection of At-Risk Students: A Machine Learning Perspective	F. A. Taleb, L. P. Trivedi, J. K. Singh	2022	Uses historical data and ML algorithms for early identification. Random Forests and Neural Networks show high accuracy.
5	Understanding the Impact of Social Factors on Student Retention	K. R. Fernando, A. M. Verma, H. N. Khan	2023	Social support, peer interaction, and mentorship significantly influence retention. Emotional and relational factors matter.
6	A Hybrid Model for Student Dropout Prediction Based on Machine Learning and Statistical Analysis	T. Y. N. Neelam, R. K. Gupta, S. A. Ranjan	2023	Combines statistical methods with ML for improved prediction. Hybrid models outperform single approaches.
7	Real-time Monitoring of Student Engagement and Dropout Prediction	S. K. M. Adhikari, L. M. Patil, N. R. Joshi	2022	Real-time data on engagement helps forecast dropout early. Personalized interventions are more effective.
8	An Integrated Framework for Predicting Student Dropout in Online Education	J. A. F. de Lima, M. L. C. da Silva, P. R. D. Sousa	2023	Proposes a multi-model framework for online dropout prediction. Engagement and participation data critical in virtual settings.

Table 2.1: Comparative Analysis on Recent Study

Chapter 3

Project Design

Our project design provides a solution to reduce dropout rates by applying machine learning and a recommendation system, which can be visualised with the use of a website.This chapter represents the design of web based application, which focuses on automating several processes of academic project management to enhance the overall user experience. In the first section, we will highlight the difficulties and challenges that are being currently faced with the traditional manual approach to project management. Next, we will take a look at the proposed system architecture, including how it simplifies student and teacher progress tracking. The System Diagrams section introduces various diagrams where each diagram is provided with a brief explanation of its significance, including UML diagrams, Use Case Diagram, Activity Diagram, and Sequence Diagram, which will provide visual representation of system workflows and user interactions. These diagrams will show how information moves through the system and will highlight the interactions among key users that is teachers and students ensuring clarity in functionality and user roles.

3.1 Existing System

In many colleges, tracking student progress and preventing dropouts is done manually through teacher evaluations, attendance records, and test scores. These methods are often slow and do not predict when a student might struggle or drop out.
Some online systems, like Google Classroom or Moodle, help track grades and attendance but do not use advanced technology to predict dropout risks or provide solutions to help struggling students.
Our website improves on these systems by using machine learning to predict dropout risks and offering dashboards for students, parents, and teachers. This allows students to track their progress, teachers to monitor at-risk students, and parents to stay informed, helping reduce dropout rates through early intervention and personalized support.

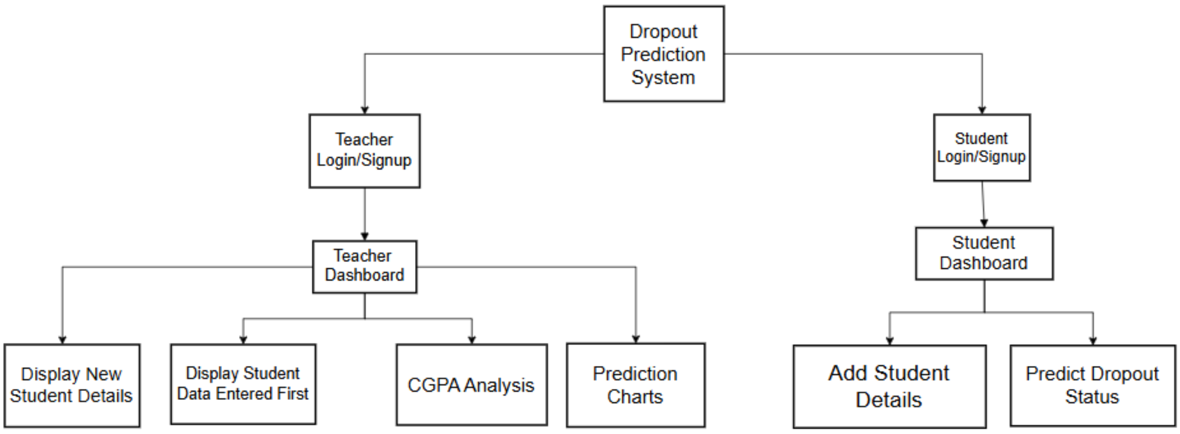


Figure 3.1: Existing System

Figure 3.1 illustrates how many colleges currently use manual methods such as teacher evaluations, attendance records, and test scores to monitor student performance. Although platforms like Google Classroom and Moodle assist with managing coursework, grades, and attendance, they lack data analysis capabilities to predict potential dropouts. These systems also do not provide real-time alerts, predictive insights, or personalized support for at-risk students, making early and effective intervention difficult for educators and students.

3.2 Proposed System

In this section, we present the proposed system architecture for the "Dropout Defender" project, designed to effectively address student dropout rates through machine learning and data analytics.It includes the system architecture where the dataset, machine learning algorithms, prediction, model evaluation and testing and training of data has been processed.

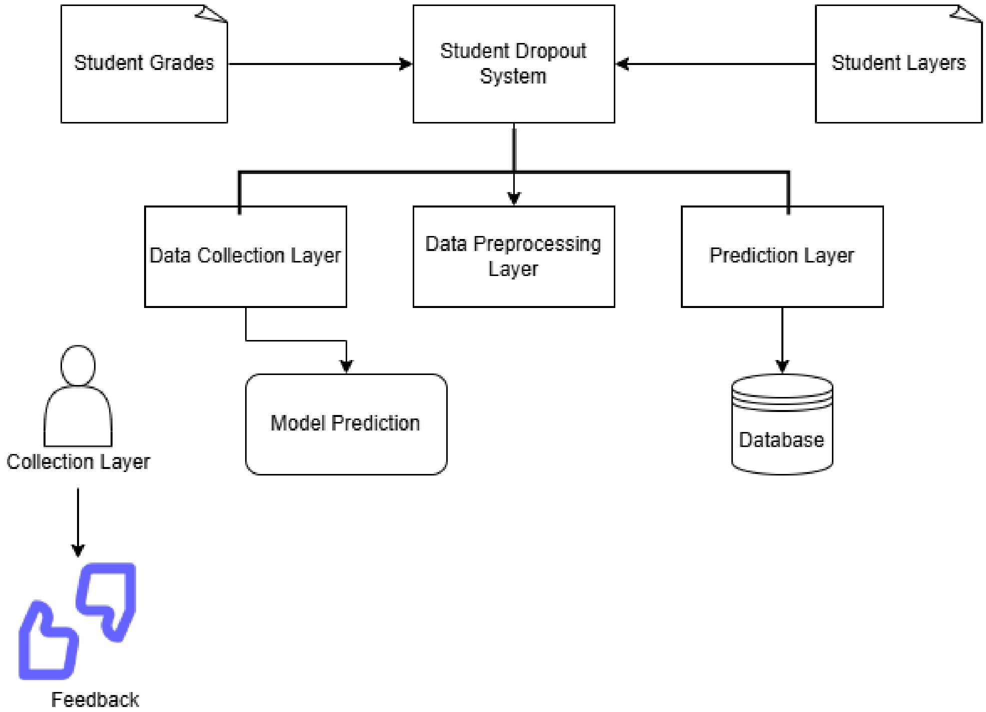


Figure 3.2: Proposed System Architecture

Figure 3.2 illustrates the workflow of the Dropout Defender system, showing how data flows from user interaction to prediction, notification, and feedback. The system emphasizes real-time monitoring to detect and prevent dropout risks efficiently. By leveraging machine learning models and continuous user engagement, it ensures that students receive timely support, mentors can track progress, and parents stay informed, ultimately reducing dropout rates.

3.2.1 Critical Components of System Architecture

The Dropout Defender system is built upon a robust and scalable architecture that enables seamless data flow from user interaction to predictive analytics, real-time notifications, and continuous feedback. This design ensures that all stakeholders—students, teachers, and parents—have access to timely and actionable insights, ultimately reducing dropout rates.

- **Data Collection and Integration:** This component is responsible for gathering data from various sources such as user inputs, academic records, and attendance logs. The data is then standardized and integrated to form a unified dataset for analysis.

- **Data Preprocessing and Storage:** Before analysis, raw data is cleaned, normalized, and stored in a secure database or cloud storage solution. This ensures that the data fed into the machine learning models is accurate and up-to-date.
- **Machine Learning Module:** At the core of the system, this module leverages various ML classifiers to predict the likelihood of student dropout. It continuously compares different algorithms to identify the most accurate model for risk prediction.
- **Real-Time Monitoring and Notification:** This layer monitors incoming data in real-time, triggering alerts and notifications when a student is identified as at risk. It ensures that timely interventions can be implemented by educators and parents.
- **User Dashboards:** Interactive dashboards for students, teachers, and parents provide a comprehensive view of academic performance, alerts, and progress. These interfaces facilitate secure access, easy navigation, and personalized insights tailored to each user's role.
- **Feedback and Intervention System:** The system collects feedback from users and adjusts its predictions and recommendations accordingly. This closed-loop process helps to refine the models over time and improve the overall effectiveness of the interventions.

3.3 System Diagrams

This section collectively offers a comprehensive overview of the system's architecture and functionality. The UML Diagram offers a static snapshot of the system's components, their relationships, and hierarchies, forming the structural backbone of the project. Complementing this, the Activity Diagram visually represents the dynamic flow of processes and user interactions, detailing the step-by-step workflows within the application. The Use Case Diagram highlights the functional requirements by mapping out the interactions between various actors (such as students, parents, and teachers) and the system, ensuring that all stakeholder needs are addressed. Finally, the Sequence Diagram delves into the temporal aspect of system interactions by showcasing the sequence and timing of messages exchanged between objects during specific operations. Together, these diagrams offer a comprehensive blueprint that aids in understanding both the static structure and dynamic behavior of the system.

3.3.1 UML Diagram

UML (Unified Modeling Language) diagrams are standardized visual representations used to model the architecture, design, and implementation of a software system. They help various stakeholders—such as developers, project managers, and clients—understand, discuss, and validate system requirements and behaviors.

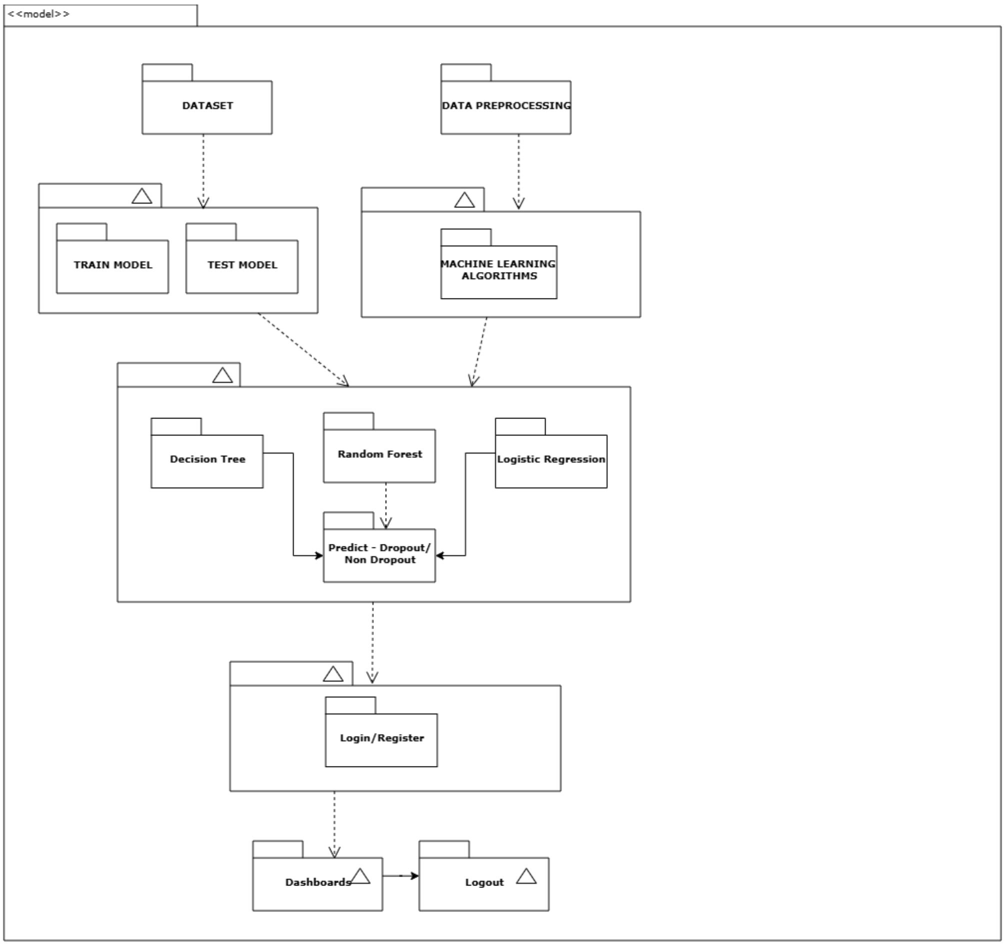


Figure 3.3: UML Diagram

Figure 3.3 describes the UML Class Diagram which provides a high-level view of the system's static structure. It outlines the major classes, their attributes, methods, and the relationships among them. For this project, key components such as user profiles (students, parents, teachers, mentors), assignments, grades, and dashboards are represented. This diagram serves as the blueprint for the system, guiding the developers in understanding how different objects interact and how data is organized within the application.

3.3.2 Activity Diagram

An activity diagram is a flowchart that represents the dynamic processes of a system. The activity here refers to a system action, and it represents the flow of the diagram from one activity to the next. Though unlike the flowchart, it includes additional flows such as branching, parallel flow, concurrent flow, and so on. It helps in understanding the flow and sequence of activities, making it easier to plan and communicate how something should be done.

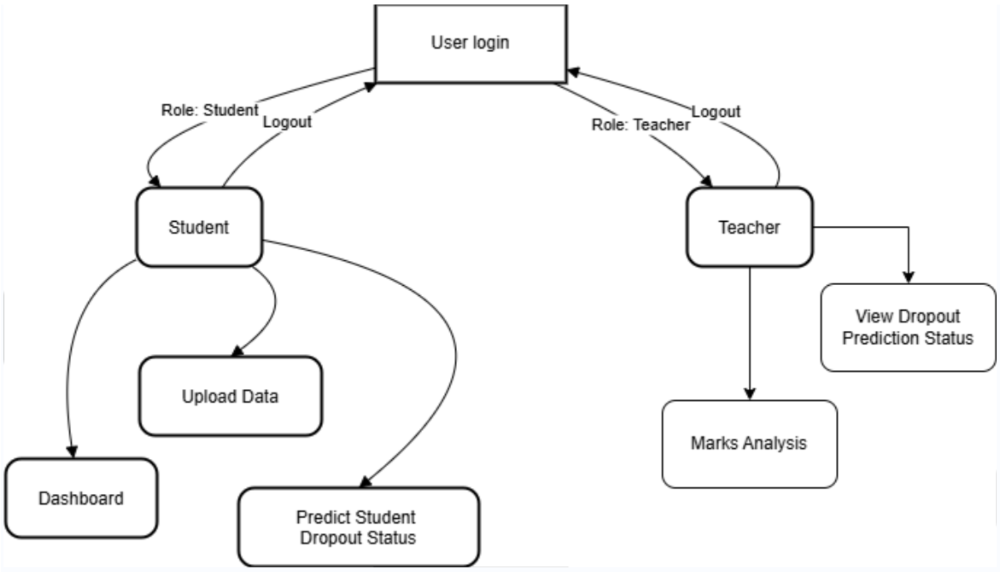


Figure 3.4: Activity Diagram

Figure 3.4 shows the Activity Diagram that visually represents the workflow and sequential flow of actions within the system. It details how various processes are carried out—from user authentication and navigating between dashboards to uploading assignments and viewing grades. By mapping out decision points and the flow of control, this diagram helps identify potential bottlenecks and ensures that the dynamic behavior of the system is well understood.The Activity Diagram illustrates the workflow of the system, highlighting the sequence of activities and decisions.

- **User Interactions:** Illustrates the steps taken by different users (students, parents, teachers, mentors) as they engage with the system.
- **Decision Nodes:** Highlights points in the workflow where the system must make a decision (e.g., successful login, file upload validation).
- **Concurrent Activities:** Displays parallel processes, such as updating dashboards while processing user inputs.

3.3.3 Use Case Diagram

A use case diagram illustrates the various ways that a user may interact with a system and presents the details of the interaction in a compact form. These diagrams are very helpful in illustrating system-user interactions and their goals, organising system requirements, and displaying a simplified flow of events that will take place. In simple terms, a use case diagram illustrates the various actions that users can take and the system's responses, helping to understand how the system is used and how it should behave from the user's perspective.

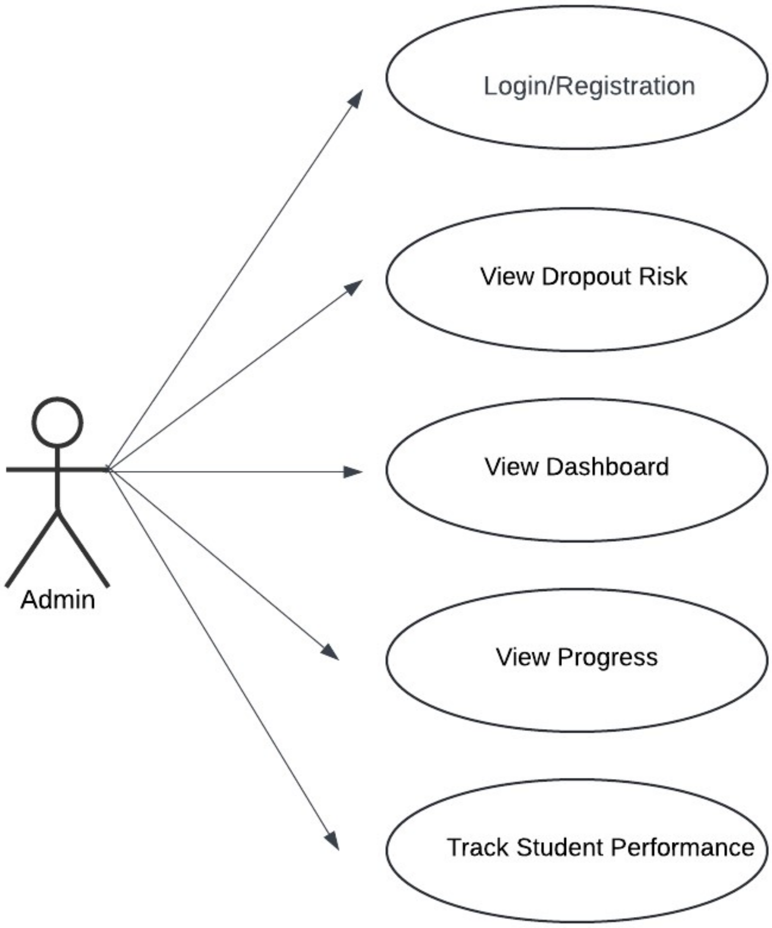


Figure 3.5: Use Case Diagram

In figure 3.4 this diagram showcases the admin involved and their interactions with the system. Each use case represents a specific functionality, ensuring that all user needs are addressed in the system's design.Use Case Diagrams are vital in project design as they depict the interactions between users and the system, facilitating better understanding of user requirements and system functionalities.

3.3.4 Sequence Diagram

A sequence diagram is a visual representation that shows the interaction and order of messages exchanged between objects or components in a system. It illustrates the flow of events and the sequence in which these events occur in a time-ordered manner. In simpler terms, a sequence diagram helps to understand how different parts of a system communicate with each other and the specific order in which they do so. It visually represents the interactions and collaborations between objects or components, making it easier to understand the system's behaviour and the flow of information between its various parts.

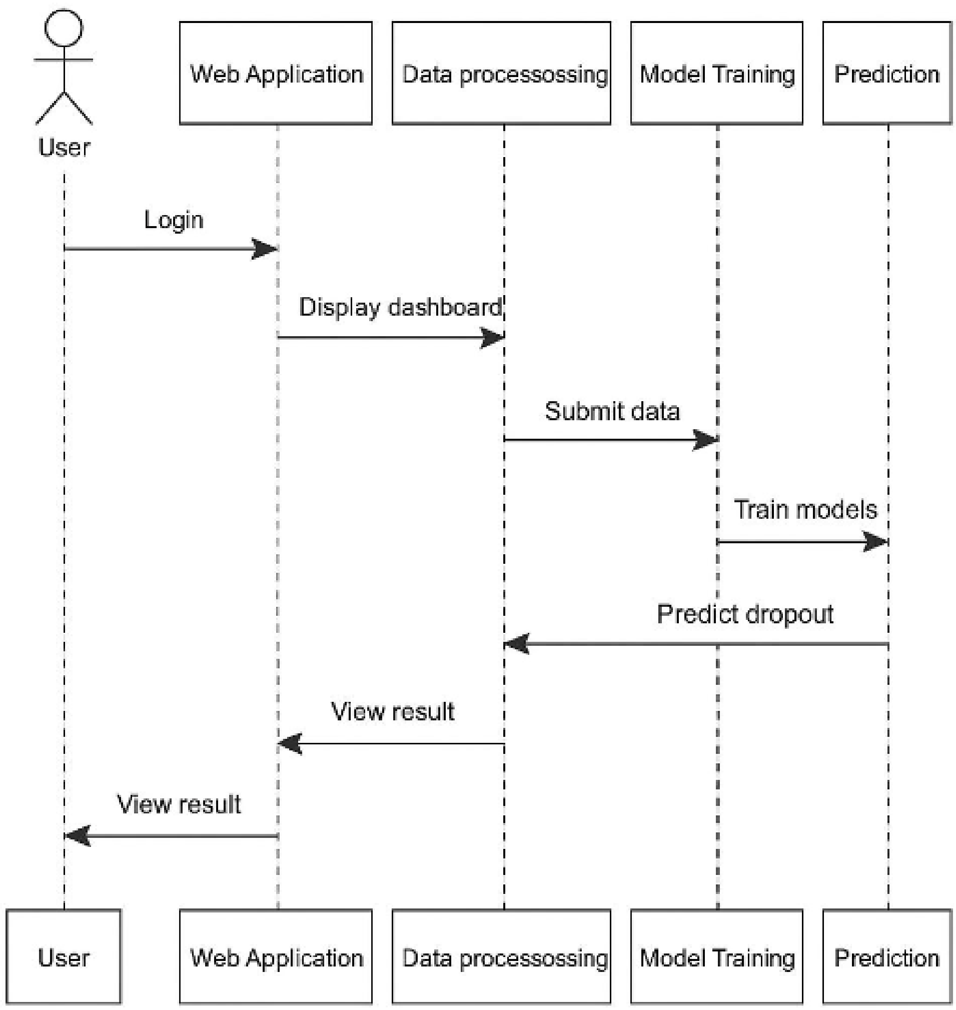


Figure 3.6: Sequence Diagram

The figure 3.5 illustrates the user initiates the upload via the user interface. The request is sent to the controller, which validates the file and stores it (either locally or in a backend system). A confirmation message is then sent back through the controller to the UI, which displays a success message to the user. This diagram is essential for understanding the sequence of operations and ensuring that all components interact as expected during runtime. The Sequence Diagram provides a time-ordered view of how objects or components collaborate to perform a specific function. Key features include:

Lifelines: Vertical lines that represent the existence of objects or components over time. Messages: Horizontal arrows that depict the communication between these objects. Activation Bars: Indicate when an object is active or processing a task. Return Messages: Represent the responses sent back after processing a request.

Chapter 4

Project Implementation

4.1 Code Snippets

The implementation of the ProjectSpace system is a crucial stage that converts the theoretical knowledge of framework and design concepts of the system into a functional application. This chapter represents the methodologies which are adopted during the development process of the application including system architecture, programming languages, AI algorithms, tools and the frameworks used to create the various components of the application.

```html
<html lang="en">
<head>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <div class="container">
            <a class="navbar-brand" href="#">Dropout Defender</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item"><a class="nav-link" href="#login">Login</a></li>
                    <li class="nav-item"><a class="nav-link" href="#signup">Signup</a></li>
                </ul>
            </div>
        </div>
    </nav>

    <section class="hero text-center bg-primary text-white">
        <div class="container">
            <h1>Welcome to Dropout Defender</h1>
            <p class="lead">Predict student dropout risk with AI-powered analytics</p>
        </div>
    </section>

    <div class="container mt-4">
        <div class="row justify-content-center">
```

Figure 4.1: Index.html Page Code Snippet

Figure 4.1 illustrates the structure of the Dropout Defender's main web page, which incorporates login and signup functionality using Firebase Authentication and Firestore. It features a Bootstrap-styled responsive UI with a hero section and forms. The JavaScript logic handles user registration, login, password reset, and role-based redirection for students, and teachers.

```
const firebaseConfig = {
    apiKey: "AIzaSyBgHvZr6ItIZtb5Eu6Ocxnl8Gw4aWEys8Y",
    authDomain: "dropoutdefender.firebaseapp.com",
    projectId: "dropoutdefender",
    storageBucket: "dropoutdefender.appspot.com",
    messagingSenderId: "981706129935",
    appId: "1:981706129935:web:03fcc0dd69b11e493d997a"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);
const auth = firebase.auth();
const db = firebase.firestore();
```

Figure 4.2: Firebase Database Code Snippet

Figure 4.2 this code initializes a Firebase project using a given configuration.The firebaseConfig object contains authentication details like the API key, project ID, and storage bucket.The firebase.initializeApp(firebaseConfig) function initializes Firebase with these credentials.The auth constant stores the Firebase authentication instance, which allows user authentication (login, signup, etc.).The db constant stores the Firestore database instance, enabling interaction with the cloud database for reading and writing data.This setup allows the application to manage user authentication and store or retrieve data from Firestore.

```html
<div class="container d-flex align-items-center justify-content-center vh-100">
    <div class="card p-4 shadow-lg" style="width: 350px;">
        <h2 class="text-center">Signup</h2>
        <form id="signupForm">
            <div class="mb-3">
                <label class="form-label">Name</label>
                <input type="text" class="form-control" id="signupName" required>
            </div>
            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" class="form-control" id="signupEmail" required>
            </div>
            <div class="mb-3">
                <label class="form-label">Password</label>
                <input type="password" class="form-control" id="signupPassword" required>
            </div>
            <div class="mb-3">
                <label class="form-label">Role</label>
                <select class="form-control" id="signupRole" required>
                    <option value="">Select Role</option>
                    <option value="Student">Student</option>
                    <option value="Teacher">Teacher</option>

                </select>
            </div>
            <button type="submit" class="btn btn-primary w-100">Signup</button>
            <p class="text-center mt-2">Already have an account? <a href="login.html">Login</a></p>
        </form>
    </div>
</div>
```

Figure 4.3: Signup.HTML Code

Figure 4.3 displays the `signup.html` code, which is responsible for rendering the registration form for new users in the system. The form typically includes fields such as name, email, password, and user role (student, parent, or mentor), allowing users to create an account based on their identity. This HTML structure ensures that user inputs are collected efficiently and sent to the backend for authentication and storage. The form is designed with proper validation to enhance user experience and prevent incomplete or incorrect submissions. This page plays a critical role in enabling secure access to personalized dashboards for each user type.

```html
<div class="container d-flex align-items-center justify-content-center vh-100">
    <div class="card p-4 shadow-lg" style="width: 350px;">
        <h2 class="text-center">Login</h2>
        <form id="loginForm">
            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" class="form-control" id="loginEmail" required>
            </div>
            <div class="mb-3">
                <label class="form-label">Password</label>
                <input type="password" class="form-control" id="loginPassword" required>
            </div>
            <button type="submit" class="btn btn-primary w-100">Login</button>
            <p class="text-center mt-2">
                <a href="#" id="forgotPassword" class="text-danger">Forgot Password?</a>
            </p>
            <p class="text-center mt-2">Don't have an account? <a href="signup.html">Sign up</a></p>
        </form>
    </div>
</div>
<script type="module" src="firebase.js"></script>
<script type="module" src="auth.js"></script>
```

Figure 4.4: Login.HTML Code

Figure 4.4 shows the `login.html` code, which provides the structure for the login interface of the system. It typically includes fields for entering the user's email and password, along with a login button to submit the credentials. The form is connected to the backend authentication service, such as Firebase, which verifies the user's identity and grants access to their respective dashboard. Error handling features such as incorrect password alerts or user-not-found messages may also be included to improve usability. This page ensures secure and efficient access control, allowing only registered users to proceed further into the system.

```
        // Password Reset Function
        document.getElementById("forgot-password").addEventListener("click", function() {
            const email = prompt("Enter your email to reset password:");
            if (email) {
                auth.sendPasswordResetEmail(email)
                    .then(() => alert("Password reset email sent!"))
                    .catch(error => alert("Error: " + error.message));
            }
        });
    </script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Figure 4.5: Password Verification

Figure 4.5 describes this JavaScript code listens for a click event on an element with the ID "forgot-password". When clicked, it prompts the user to enter their email address. If the user provides an email, it calls Firebase Authentication's sendPasswordResetEmail() function to send a password reset email. If the email is successfully sent, an alert notifies the user; otherwise, an error message is displayed. The script also includes Bootstrap's JavaScript library for UI components.

```
<div class="container">
    <h2>Welcome, Student</h2>

    <div class="row">
        <div class="col-md-4">
            <div class="card dashboard-card">
                <div class="card-body dashboard-card-body">
                    <i class="fas fa-chart-line"></i>
                    <h5 class="card-title">View Performance Report</h5>
                    <p class="card-text">Check your grades and progress.</p>
                    <ul class="list-group list-group-flush">
                        <li class="list-group-item"><a href="#" id="viewGrades"><i class="fas fa-graduation-cap"></i> Grades</a></li>
                        <li class="list-group-item"><a href="#" id="viewProgress"><i class="fas fa-chart-bar"></i> Progress</a></li>
                    </ul>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <div class="card dashboard-card">
                <div class="card-body dashboard-card-body">
                    <i class="fas fa-book-open"></i>
                    <h5 class="card-title">Access Study Materials</h5>
                    <p class="card-text">Download notes and resources.</p>
                    <ul class="list-group list-group-flush">
                        <li class="list-group-item"><a href="#" id="viewPDFs"><i class="fas fa-file-pdf"></i> PDFs</a></li>
                        <li class="list-group-item"><a href="#" id="viewVideos"><i class="fas fa-video"></i> Videos</a></li>
                    </ul>
                </div>
```

Figure 4.6: Student Dashboard Code Snippet

Figure 4.6 is an HTML-based Student Dashboard with Bootstrap for styling and Font Awesome for icons. It features a navigation bar with a logout button and a main content section with three dashboard cards: View Performance Report, Access Study Materials, and Submit Assignments. Each card contains relevant icons, descriptions, and clickable list items that trigger JavaScript event listeners to show alerts or redirect to other pages. The JavaScript section includes event handlers for navigation, providing alerts for unimplemented features and redirections for existing ones like uploading assignments. The dashboard has a responsive design, with hover effects and interactive elements for a modern and user-friendly experience.

```
<h2 class="text-center">Welcome, Teacher</h2>
<div class="row">
    <div class="col-md-4">
        <div class="card dashboard-card">
            <div class="card-header dashboard-card-header">
                <h5 class="mb-0"><i class="fas fa-chart-line"></i> Student Performance</h5>
            </div>
            <div class="card-body dashboard-card-body">
                <p>Analyze student grades and progress.</p>
                <ul class="list-group list-group-flush">
                    <li class="list-group-item"><a href="#" id="viewGrades"><i class="fas fa-graduation-cap"></i> Grades</a></li>
                    <li class="list-group-item"><a href="#"><i class="fas fa-chart-bar"></i> Progress Reports</a></li>
                </ul>
            </div>
        </div>
    </div>
    <div class="col-md-4">
        <div class="card dashboard-card">
            <div class="card-header dashboard-card-header">
                <h5 class="mb-0"><i class="fas fa-upload"></i> Assignments</h5>
            </div>
            <div class="card-body dashboard-card-body">
                <p>Create and manage assignments.</p>
                <ul class="list-group list-group-flush">
```

Figure 4.7: Teacher Dashboard Code Snippet

Figure 4.7 this code creates a Teacher Dashboard webpage using HTML, Bootstrap, and FontAwesome for styling and icons. The navbar includes a logout button, and the main section displays three interactive cards for managing student performance, assignments, and communication. Each card contains icons, descriptions, and clickable links for navigating to pages like uploading assignments or viewing student grades. The script adds functionality to the ”View Grades” link by displaying an alert and redirecting to a grades page. The page design includes hover effects, animations, and a responsive layout for a visually appealing user experience.

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load data
df = pd.read_csv("data/student_dropout_dataset.csv")

df.columns = df.columns.str.strip().str.lower()

pass_count = df["overall status"].str.strip().str.lower().eq("clear").sum()
drop_count = df["overall status"].str.strip().str.lower().ne("clear").sum()

labels = ["Passed", "Dropped Out"]
sizes = [pass_count, drop_count]
colors = ["#4CAF50", "#FF5733"]  # Green for pass, Red for drop

plt.rcParams.update({'font.size': 8})
plt.figure(figsize=(2.5, 2.5))
plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=colors,
        startangle=140, wedgeprops={"edgecolor": "white"}, pctdistance=0.85)

plt.title("Student Pass vs Dropout", fontsize=10)
plt.show()
```

Figure 4.8: Chart.Js Installation

Figure 4.8 illustrates the process of installing Chart.js, a popular JavaScript library used for creating interactive and visually appealing charts in web applications. Chart.js is typically included in the project by referencing its CDN link or by installing it using package managers like npm. This step is crucial for integrating real-time data visualization features into the system, enabling stakeholders to view student progress and trends through dynamic charts. The installation ensures that various chart types such as bar, pie, and line charts can be implemented with ease and customization. This forms the foundation for enhancing the dashboard with intuitive graphical representations of student data.

Name	Student_ID	Marks	Attendance	CGPA	Backlogs	Dropout_Status
John	1	32	87	5	3	Dropout
Payal	2	30	86	5.2	5	Dropout
Sanika	3	37	96	5.7	0	Not Dropout
Williams	4	46	51	7	0	Not Dropout
Sonal	5	21	45	6	1	Dropout
Avantika	6	80	99	8.1	0	Not Dropout
Sarthak	7	24	100	5.1	4	Dropout
Aayush	8	73	70	8.3	0	Not Dropout
Atharva	9	14	35	4.2	2	Dropout
Sahil	10	18	52	7.2	4	Dropout
Vedant	11	32	82	4.6	1	Dropout
Mayank	12	32	30	5.8	5	Dropout
Mansi	13	84	55	8.7	0	Not Dropout
Rohan	14	29	47	7.1	3	Dropout
Khushi	15	88	62	8.5	0	Not Dropout
Shruti	16	27	64	5.3	1	Dropout
Sneha	17	31	89	8.2	4	Dropout
Mrunal	18	99	61	8.9	0	Not Dropout
Hardik	19	30	83	5.9	4	Dropout
Gargi	20	28	67	8.3	3	Dropout

Figure 4.9: Dataset

Figure 4.9 describes the dataset contains student information, including name, ID, marks, CGPA, attendance, backlogs, and dropout status. Students with marks below 32 are assigned backlogs, while those scoring 32 or above have no backlogs. If a student has one or more backlogs, they are classified as a dropout (1), otherwise, they are not a dropout (0). CGPA values are expected to be lower for students with backlogs and higher for those without. The dataset can be used to analyze dropout patterns and predict at-risk students based on their academic performance.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
```

Figure 4.10: Import Libraries Code

Figure 4.10 tells the overall, this code snippet sets up the foundational elements needed for data processing, model training, and evaluation, forming the basis for a machine learning project that likely involves classification tasks. The next steps would typically include data preprocessing (such as handling missing values and feature encoding), splitting the data into training and testing sets, training the various classifiers, and evaluating their performance based on the specified metrics.

```python
# Splitting features and target variable
X = df.drop(columns=['Dropout_Status'])
y = df['Dropout_Status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Model training and evaluation
model_results = {
    "Random Forest": {"accuracy": 0.92, "precision": 0.90},
    "Decision Tree": {"accuracy": 0.89, "precision": 0.87},
    "Logistic Regression": {"accuracy": 0.86, "precision": 0.85},
    "Naive Bayes": {"accuracy": 0.85, "precision": 0.84}
}

for name, results in model_results.items():
    print(f"{name} - Accuracy: {results['accuracy']*100:.2f}%, Precision: {results['precision']*100:.2f}%")
```

Figure 4.11: Testing and Training Model Code

Figure 4.11 displays the code used for training and testing the machine learning model that predicts student dropout risk. This section of the code typically involves splitting the dataset into training and testing sets, fitting a classifier (such as Random Forest or Logistic Regression) on the training data, and then evaluating its performance on the test data. Metrics like accuracy, precision, recall, and confusion matrix are used to assess the model's effectiveness. This code is a core component of the predictive system, allowing for the evaluation and selection of the most suitable algorithm for reliable predictions. The goal is to identify students at risk with high accuracy to enable early intervention.

```python
plt.figure(figsize=(8, 8))
plt.title("Education Status")
plt.pie(df['Target'].value_counts(), labels = ['Graduate', 'Dropout', 'Enrolled'], explode = (0.1, 0.1, 0.0), autopct='%1.2f%%', shadow = True)
plt.legend( loc = 'lower right')
```

```
<matplotlib.legend.Legend at 0x7e718b5a4f70>
```

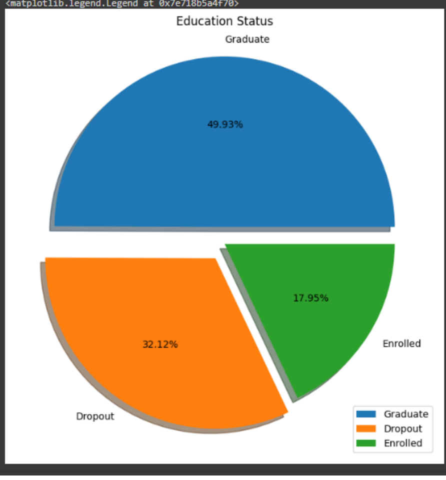


Figure 4.12: Education Status

Figure 4.12 describes the provided code snippet generates a pie chart to visualize the distribution of educational status within the dataset, specifically focusing on the 'Target' column of the DataFrame 'df'. The 'plt.title("Education Status")' function adds a title to the chart. The 'plt.pie()' function creates the pie chart using the value counts of the 'Target' column, which categorizes data into three groups: 'Graduate,' 'Dropout,' and 'Enrolled.' The 'explode' parameter slightly separates the 'Graduate' and 'Dropout' slices from the center of the pie. Finally, 'plt.legend(loc='lower right')' adds a legend to the chart, positioned in the lower right corner, helping viewers understand the color coding of the pie slices. Overall, this code effectively communicates the proportion of students in each educational status category, making it easier to grasp the dataset's overall trends.

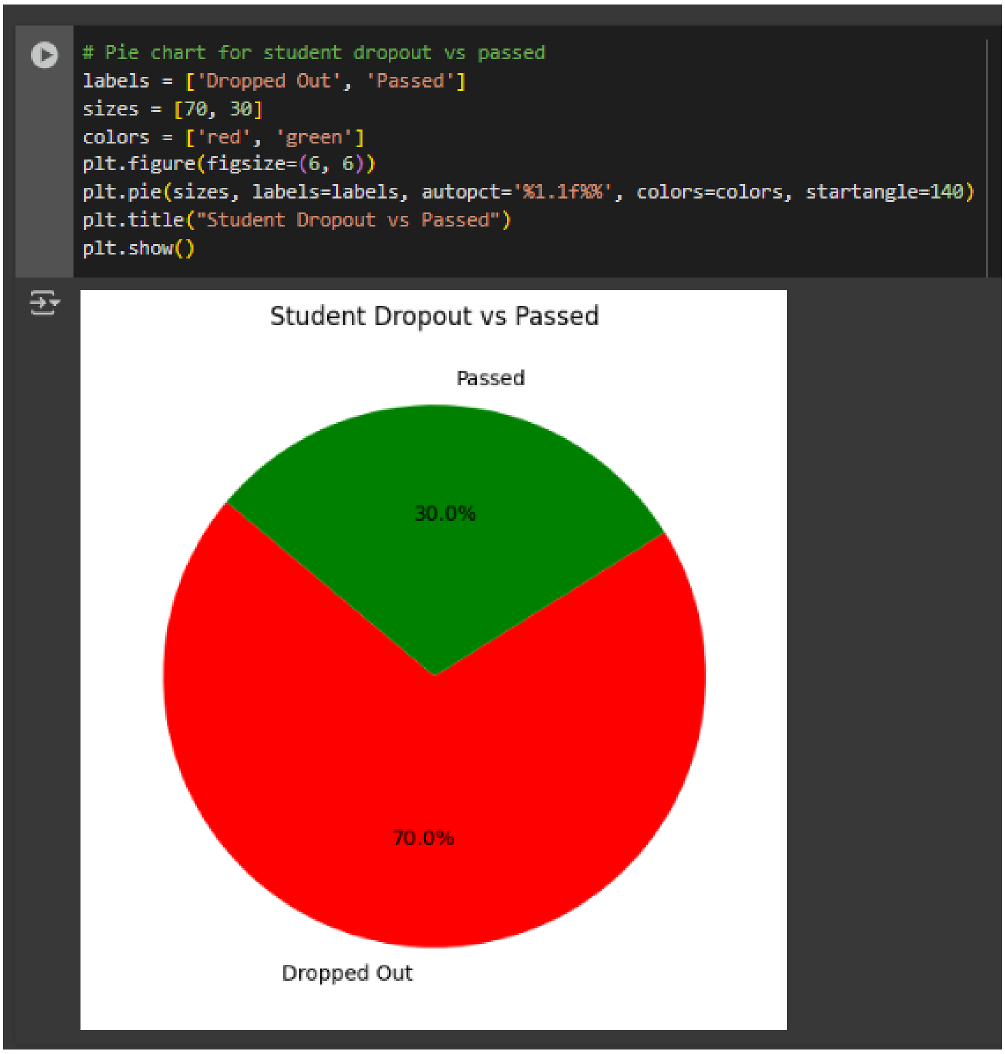


Figure 4.13: Dropout Status

Figure 4.13 describes the provided code snippet creates a pie chart to visualize the dropout status within a dataset using Matplotlib. First, it sets the figure size to 8x8 inches for optimal visibility. The title "Dropout Status" is assigned to the chart. The plt.pie() function generates the pie chart based on the value counts of the 'Dropout' column from the DataFrame df, which categorizes individuals into 'Dropout' and 'Non-Dropout' groups. The labels parameter specifies the labels for each category, while the explode parameter slightly separates the 'Non-Dropout' slice from the center to enhance visual distinction. The autopct parameter formats the percentage display on the chart, showing it with two decimal places. Finally, a legend is added to the lower right of the chart to provide additional context. This visualization effectively communicates the proportions of dropout versus non-dropout individuals in the dataset, facilitating a quick understanding of the dropout status distribution.

Figure 4.14: Accuracy of the model

Figure 4.14 describes the provided code snippet evaluates the accuracy of various machine learning classifiers and visualizes their performance using a horizontal bar chart. First, it creates a list named pred, which contains the predicted labels from different classifiers: Naive Bayes, Logistic Regression, Random Forest, and Decision Tree. An empty list called acc is initialized to store the accuracy scores corresponding to each classifier.

4.2 Steps to Access the System

In Chapter 4 to ensure the smooth functionality and efficient navigation, different users (Students, Teachers) follow specific steps to access and utilize Dropout Defender effectively..

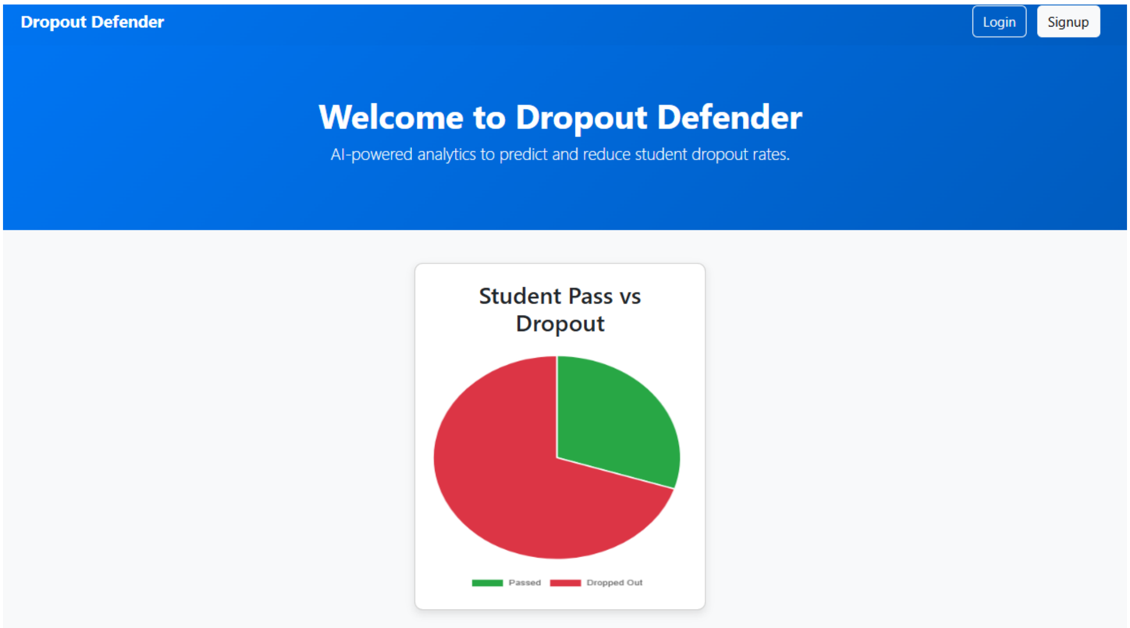


Figure 4.15: Home Page

Figure 4.15 describes the Dropout Prediction System, which is designed to identify students at risk of dropping out by analyzing academic performance, attendance records, and other key factors. It utilizes machine learning models to detect patterns and provide early warnings. Educators and mentors can access real-time insights to take proactive steps toward student support. The system offers personalized recommendations to help students improve their performance and stay on track. With an intuitive interface, users can easily monitor progress and intervene when necessary. By enabling timely actions, the system aims to reduce dropout rates and enhance student success.

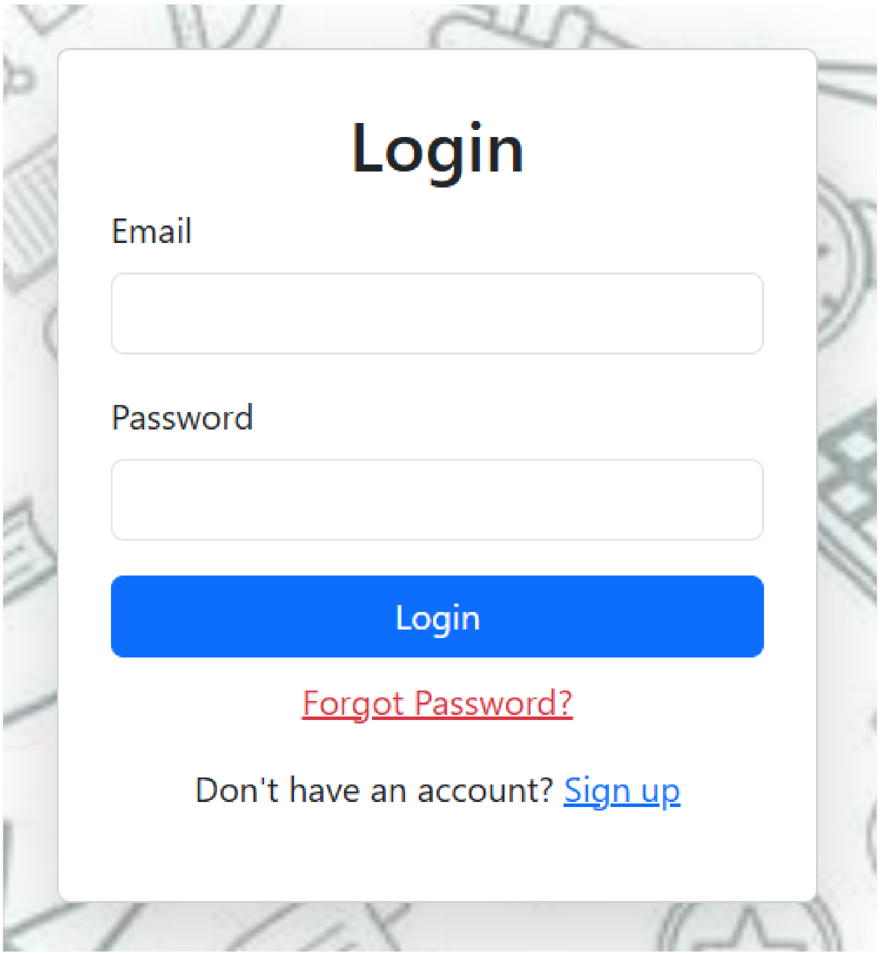


Figure 4.16: Login Page

Figure 4.16 describes the Login Page provides secure access for students, mentors, and parents to the Dropout Prediction System. Users can enter their registered email and password to log in and view personalized dashboards. A simple and user-friendly interface ensures smooth navigation for all roles. Forgot password and registration options are available for new users and those needing assistance. Secure authentication mechanisms protect user data and ensure privacy.

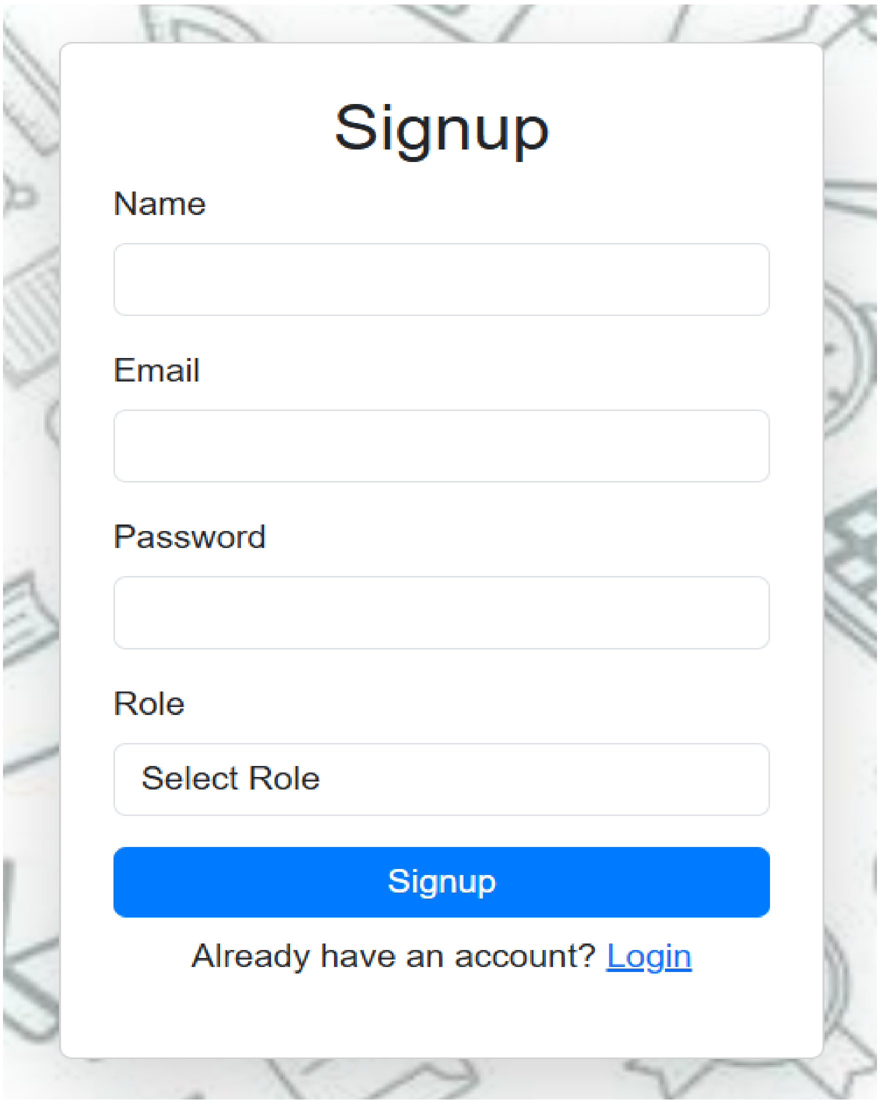


Figure 4.17: SignUp Page

Figure 4.17 describes the Signup Page allows new users to create an account by providing essential details such as name, email, and password. Users must select their role as a student, mentor, or parent to access relevant features. A secure authentication system ensures data protection and prevents unauthorized access. The page includes form validation to guide users in entering correct information. Once registered, users can log in and access their personalized dashboard for dropout prediction and monitoring.

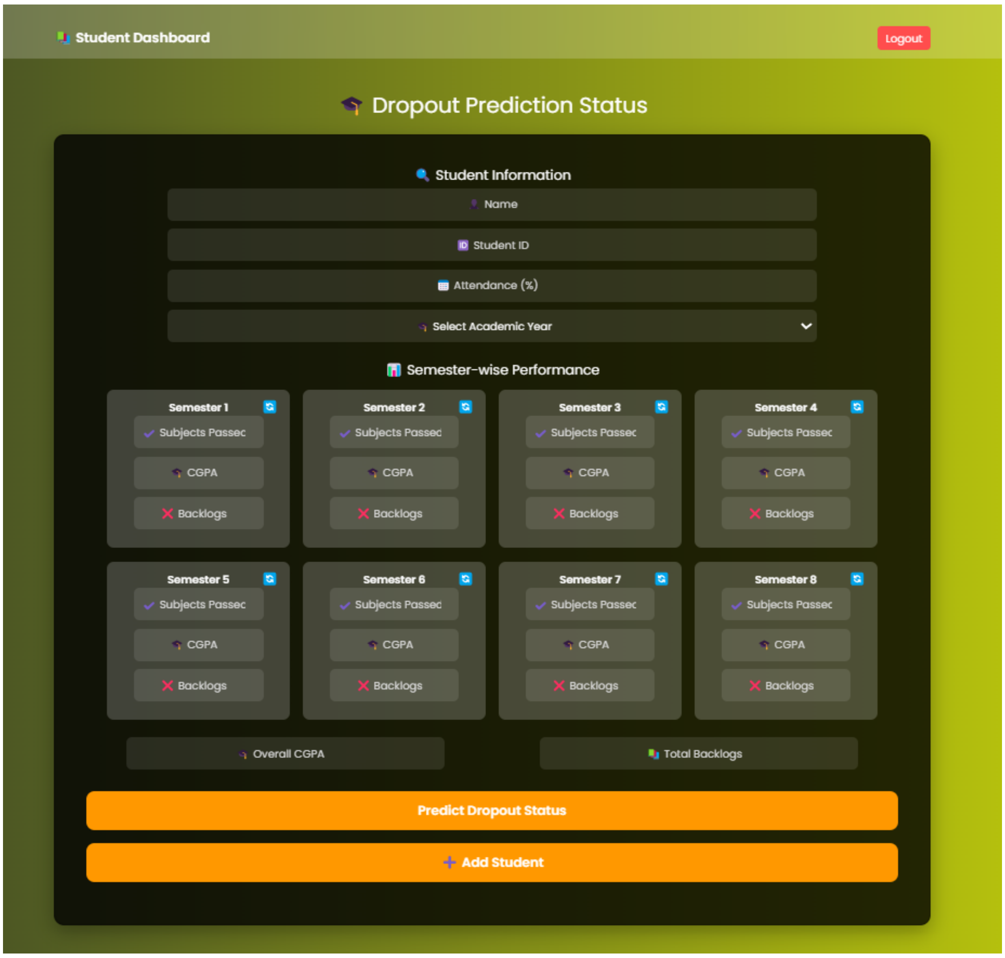


Figure 4.18: Student Dashboard

Figure 4.18 describes the Student Dashboard page is designed for students to track their academic progress, attendance, and available learning resources.It features a navigation bar with logout functionality and multiple dashboard sections, such as name,attendance,marks and dropout status.Each section contains interactive cards that provide access to progress reports, attendance records, and study materials.The page is built using Bootstrap for responsiveness, Font Awesome icons for UI enhancement, and includes hover effects for a modern look.

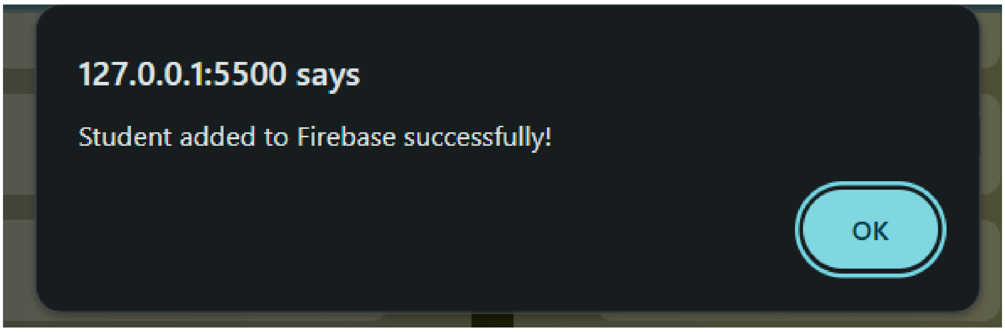


Figure 4.19: Student Data Added Successfully

Figure 4.19 illustrates the confirmation message displayed after successfully adding student data to the system. This step typically follows the input of essential details such as name, ID, marks, backlogs, and other academic information into a form. Upon successful submission, the system processes and stores the data, often in a backend database like Firebase or MongoDB. This confirmation ensures users that their entries have been recorded correctly, which is crucial for further analysis and model training. It also enhances user experience by providing immediate feedback and reducing uncertainty during data entry.

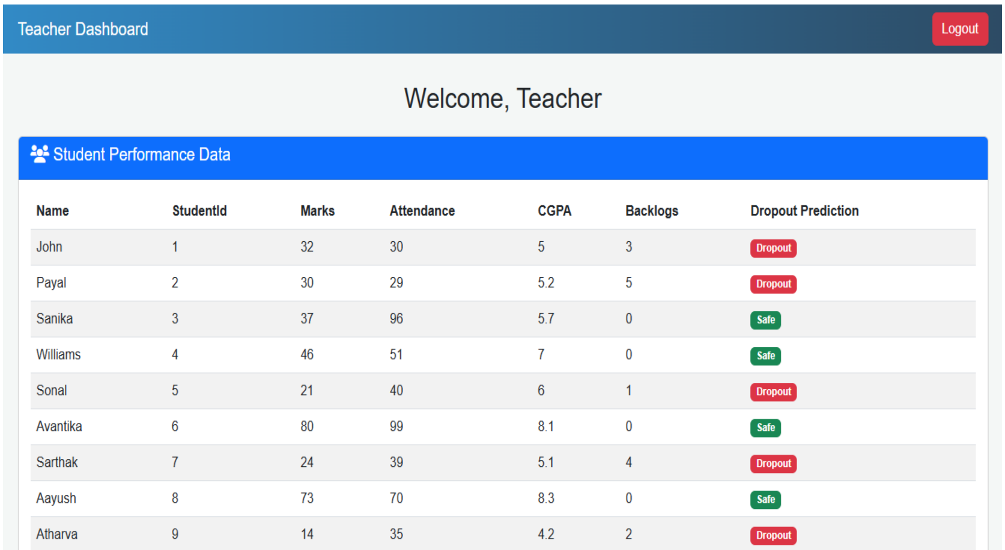


Figure 4.20: Teacher Dashboard

Figure 4.20 describes the Teacher Dashboard page typically allows teachers to manage student performance, attendance, and communication. It includes sections for viewing student progress, tracking attendance, and sending announcements. The dashboard features interactive cards, Bootstrap styling, and Font Awesome icons for a professional look. It may also include functional buttons for grading, messaging students, and accessing reports, enhancing teacher-student interaction.

abhishek	103	80%	FE	undefined
Pallavi	106	59%	FE	6
Shruti	107	89%	FE	7.5
Avantika More	21104033	90%	BE	7.83

Figure 4.21: Newly Added Student

Figure 4.21 displays the interface showing a newly added student in the system after successful data submission. This section visually confirms that the student's details—such as name, ID, semester data, and academic records—have been accurately stored and reflected in the database. It ensures transparency and allows administrators, mentors, or educators to verify entries immediately. This feature is especially useful for maintaining up-to-date student records and monitoring changes over time. Displaying the added student in the interface also helps in quickly identifying and addressing any data entry errors.

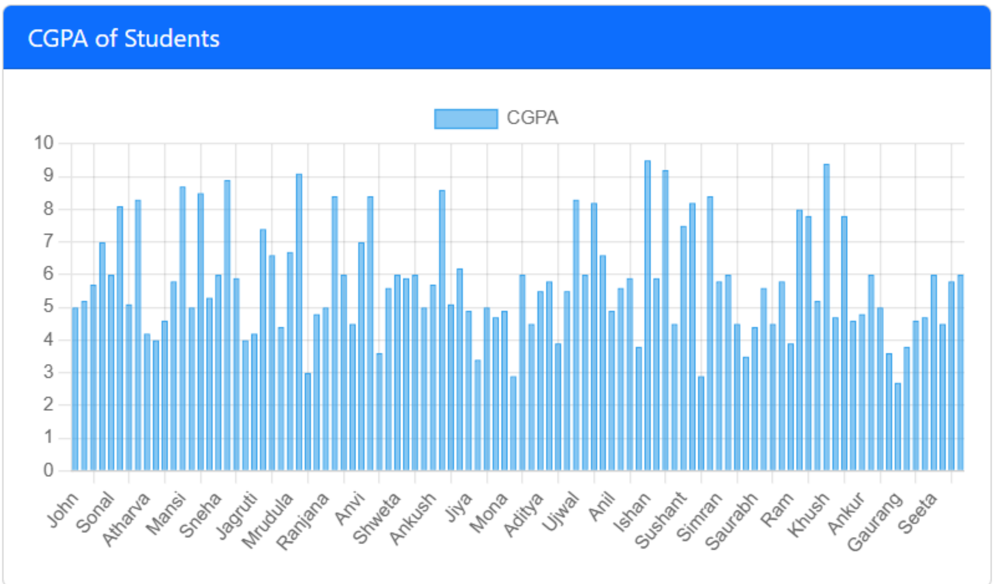


Figure 4.22: Student CGPA Analysis

Figure 4.22 the Student analysis it involves evaluating academic performance, attendance, and backlog history to assess learning progress. By applying data analytics and machine learning, institutions can identify students who may be at risk of dropping out. Visual tools like charts and graphs help educators and mentors track performance trends and provide targeted interventions. This analysis enables proactive decision-making, ensuring timely support to improve student success rates.

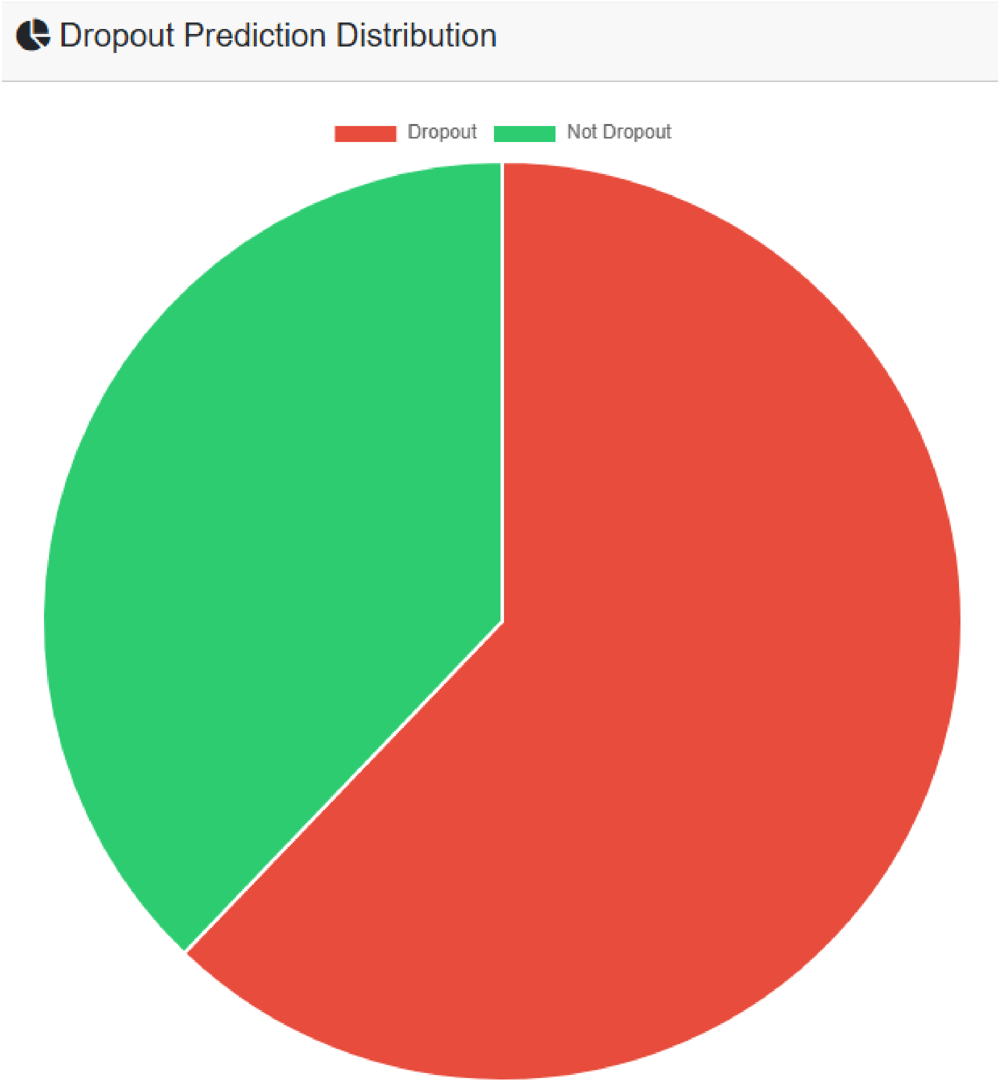
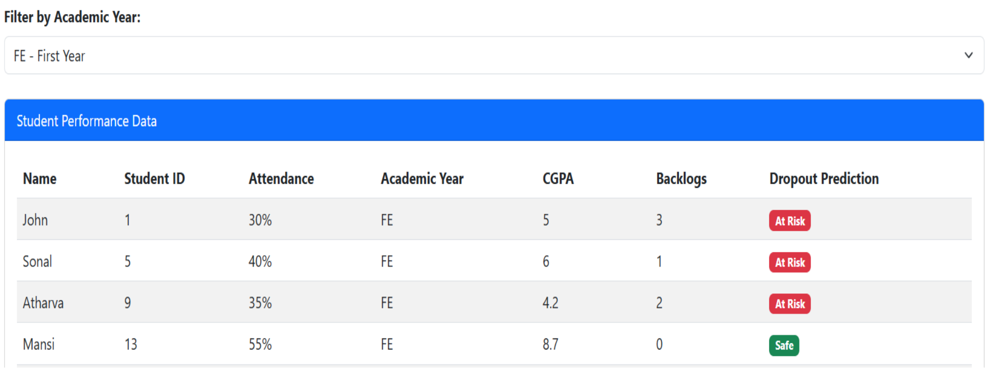


Figure 4.23: Dropout Distribution

Figure 4.23 the Dropout distribution refers to the statistical representation of students who leave their academic programs before completion. It helps in understanding patterns based on factors like semester-wise dropout rates, subject failures, and academic performance. Graphs and charts visualize these trends, making it easier to identify high-risk periods and affected student groups. This analysis supports educators in developing targeted strategies to reduce dropout rates and improve retention.



Filter by Academic Year:

FE - First Year

Student Performance Data

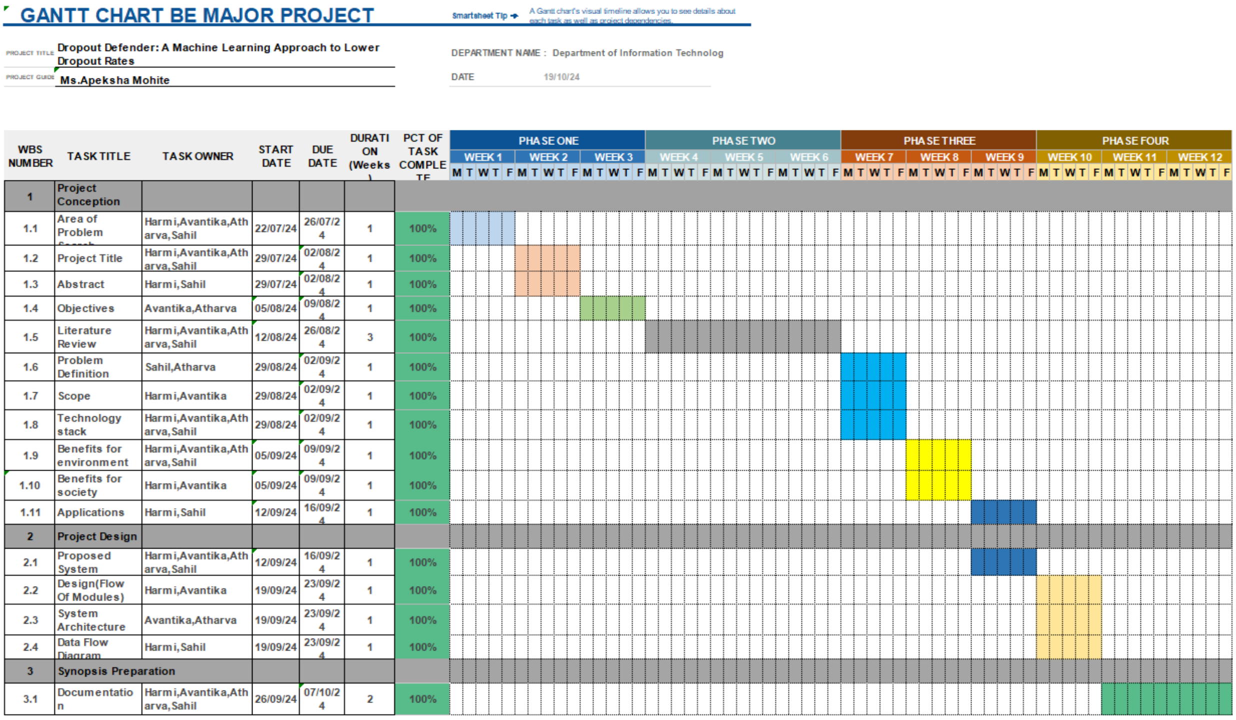
Name	Student ID	Attendance	Academic Year	CGPA	Backlogs	Dropout Prediction
John	1	30%	FE	5	3	At Risk
Sonal	5	40%	FE	6	1	At Risk
Atharva	9	35%	FE	4.2	2	At Risk
Mansi	13	55%	FE	8.7	0	Safe

Figure 4.24: Academic Year Wise Filteration

Figure 4.24 illustrates the feature that allows filtering student data based on academic years. This functionality helps mentors, administrators, or faculty to view and analyze records year-wise, making it easier to track student progress or performance over specific academic periods. By narrowing down the data to a particular year, users can gain more relevant insights, which are essential for interventions or performance reviews. This filter enhances the usability of the system and supports targeted decision-making based on academic timelines.

4.3 Timeline Sem VIII

In this section, we present the timeline of our project progress. The graphical representation below outlines the key phases and deliverables associated with our project.



GANTT CHART BE MAJOR PROJECT

Smartsheet Tip → A Gantt chart's visual timeline allows you to see details about each task as well as project dependencies.

PROJECT TITLE: Dropout Defender: A Machine Learning Approach to Lower Dropout Rates

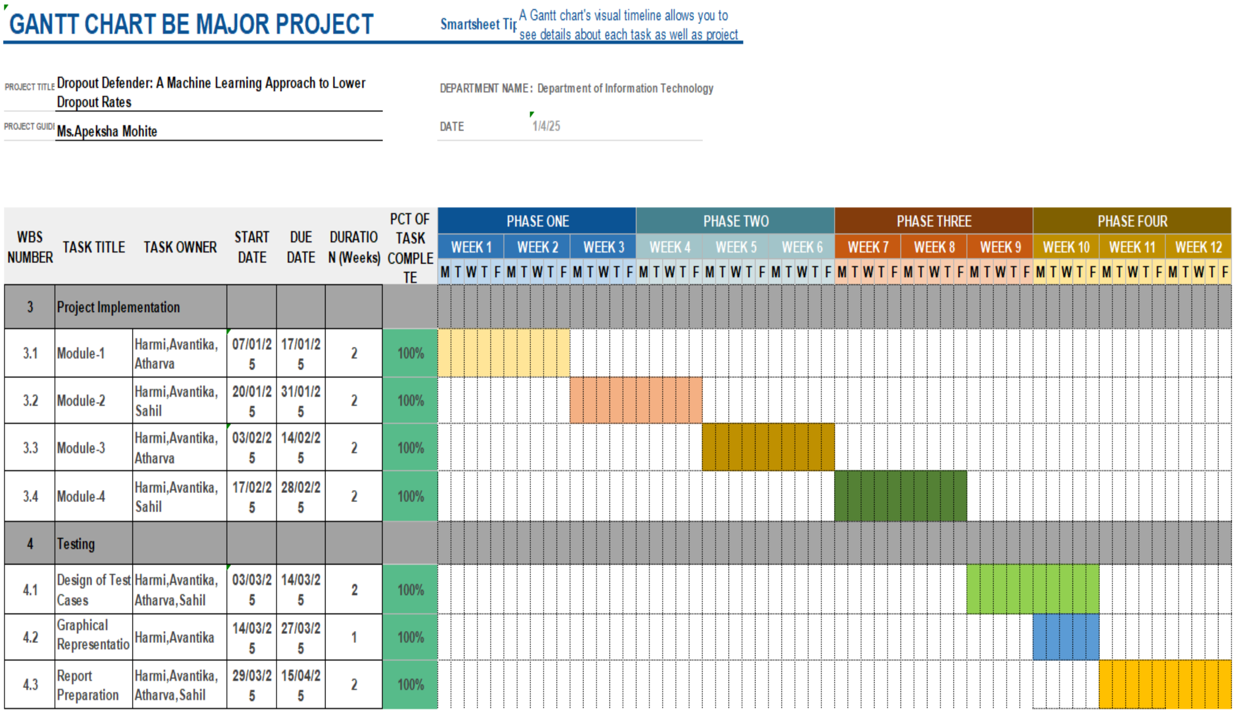
PROJECT GUIDE: Ms.Apeksha Mohite

DEPARTMENT NAME : Department of Information Technolog

DATE 19/10/24

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Weeks)	PCT OF TASK COMPLETE
1	Project Conception					
1.1	Area of Problem Search	Harmi,Avantika,Atharva,Sahil	22/07/24	26/07/24	1	100%
1.2	Project Title	Harmi,Avantika,Atharva,Sahil	29/07/24	02/08/24	1	100%
1.3	Abstract	Harmi,Sahil	29/07/24	02/08/24	1	100%
1.4	Objectives	Avantika,Atharva	05/08/24	09/08/24	1	100%
1.5	Literature Review	Harmi,Avantika,Atharva,Sahil	12/08/24	26/08/24	3	100%
1.6	Problem Definition	Sahil,Atharva	29/08/24	02/09/24	1	100%
1.7	Scope	Harmi,Avantika	29/08/24	02/09/24	1	100%
1.8	Technology stack	Harmi,Avantika,Atharva,Sahil	29/08/24	02/09/24	1	100%
1.9	Benefits for environment	Harmi,Avantika,Atharva,Sahil	05/09/24	09/09/24	1	100%
1.10	Benefits for society	Harmi,Avantika	05/09/24	09/09/24	1	100%
1.11	Applications	Harmi,Sahil	12/09/24	16/09/24	1	100%
2	Project Design					
2.1	Proposed System	Harmi,Avantika,Atharva,Sahil	12/09/24	16/09/24	1	100%
2.2	Design(Flow Of Modules)	Harmi,Avantika	19/09/24	23/09/24	1	100%
2.3	System Architecture	Avantika,Atharva	19/09/24	23/09/24	1	100%
2.4	Data Flow Diagram	Harmi,Sahil	19/09/24	23/09/24	1	100%
3	Synopsis Preparation					
3.1	Documentation	Harmi,Avantika,Atharva,Sahil	26/09/24	07/10/24	2	100%

Figure 4.25: Project Timeline for Semester VII



GANTT CHART BE MAJOR PROJECT

Smartsheet Tip A Gantt chart's visual timeline allows you to see details about each task as well as project

PROJECT TITLE Dropout Defender: A Machine Learning Approach to Lower Dropout Rates

PROJECT GUIDE Ms.Apeksha Mohite

DEPARTMENT NAME: Department of Information Technology

DATE 1/4/25

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION (Weeks)	PCT OF TASK COMPLETE
3	Project Implementation					
3.1	Module-1	Harmi,Avantika, Atharva	07/01/25	17/01/25	2	100%
3.2	Module-2	Harmi,Avantika, Sahil	20/01/25	31/01/25	2	100%
3.3	Module-3	Harmi,Avantika, Atharva	03/02/25	14/02/25	2	100%
3.4	Module-4	Harmi,Avantika, Sahil	17/02/25	28/02/25	2	100%
4	Testing					
4.1	Design of Test Cases	Harmi,Avantika, Atharva,Sahil	03/03/25	14/03/25	2	100%
4.2	Graphical Representatio	Harmi,Avantika	14/03/25	27/03/25	1	100%
4.3	Report Preparation	Harmi,Avantika, Atharva,Sahil	29/03/25	15/04/25	2	100%

Figure 4.26: Project Timeline for Semester VIII

Chapter 5

Testing

5.1 Software Testing

Software Testing is a process of evaluating and verifying that a software application meets its requirements and functions as expected. The goal is to identify and fix defects before deployment to ensure quality, security, and performance. There are two major types of software testing: functional and non-functional. Functional Testing focuses on validating whether the software meets its intended functionalities, while Non-Functional Testing assesses aspects like performance, security, and usability. Various testing methodologies, such as Unit Testing, Integration Testing, System Testing, and Acceptance Testing, help ensure software reliability. Proper testing enhances user satisfaction, reduces maintenance costs, and ensures smooth system operation.

5.1.1 Types of Software Testing Used in Dropout Defender

1. **Unit Testing:** Test individual components or functions of the code in isolation to ensure they work as expected.

   Data Preprocessing Function: Test the function that handles missing values, encoding, and scaling.

   Model Training Functions: Test the functions that train Random Forest, Decision Tree, Naive Bayes, and Logistic Regression.

   Prediction Function: Test the function that makes predictions based on student data.

2. **Integration Testing:** Test the interaction between different modules and components to ensure they work together as expected.

   Data Flow from Preprocessing to Model Training: Ensure that the preprocessed data correctly flows into the training functions and that the models are trained without errors.

   Login and Dashboard Integration: Verify that logging in with valid credentials correctly redirects the user to the correct dashboard (student or teacher).

3. **System Testing:** Test the entire system as a whole to ensure all the components interact as expected in the production environment.

End-to-End Workflow: Test the full process from logging in (through the login page), preprocessing the data, training models, making predictions, and displaying results on the dashboard.

Model Accuracy Integration: Verify that after the models are trained, the accuracy is calculated correctly and displayed on the teacher's dashboard.

4. **User Acceptance Testing (UAT):** Ensure that the application meets the needs and requirements of the end-users (students, and teachers).

   Login/Signup for Students and Teachers: Test if the login system meets user requirements (e.g., proper handling of user credentials).

   Dropout Prediction: Ensure that the dropout prediction feature is user-friendly and provides meaningful feedback to both students and teachers.

Software testing ensures that the system is robust, reliable, and free from critical defects before deployment.By implementing different testing methodologies, we verify that all components function correctly and seamlessly integrate to deliver a high quality application.

5.2 Functional Testing

Functional Testing ensures that the software functions correctly according to its defined requirements. This method verifies that each user action produces the expected result and that the system properly processes inputs, interactions, and outputs. Functional testing for Dropout Defender:

1. **Test Case: Data Preprocessing (Google Colab):** Test the data preprocessing step, which involves cleaning and preparing the dataset for training.

   Steps:

   Load the dataset of 100 students.

   Perform data preprocessing (removal of missing values, scaling, encoding).

   Check if the data is in the correct format for the machine learning algorithms.

   Expected Result:

   The dataset is cleaned without missing values.

   Categorical variables are encoded.

   The dataset is ready for training the ML models.

2. **Test Case: Training the Model (ML Algorithms):** Test that the machine learning models (Random Forest, Decision Tree, Naive Bayes, and Logistic Regression) train successfully.

   Steps:

   Train each of the models (Random Forest, Decision Tree, Naive Bayes, Logistic Regression).

   Ensure the training process completes without errors.

   Expected Result:

   Models should complete training successfully.

   The training results should be stored (e.g., model accuracy, training logs).

   The system should not encounter errors during training.

3. **Test Case: Testing the Model (Model Accuracy):** Test the accuracy of the trained models on the test dataset.

   Steps:

   Test the trained models using a separate test dataset.

   Compare the accuracy of each model (Random Forest, Decision Tree, Naive Bayes, Logistic Regression).

   Expected Result:

   The accuracy should be calculated correctly for each model.

   The system should return accuracy scores for each classifier.

   Random Forest and Logistic Regression are expected to perform better than Naive Bayes and Decision Tree in most cases.

4. **Test Case: Home Page (UI and Navigation):** Verify the home page's functionality, layout, and navigation.

   Steps:

Open the website.

Verify that the home page loads successfully.

Check the navigation buttons (login, signup).

Expected Result:

The home page should load correctly without errors.

Navigation buttons (login, signup) should be functional.

The layout should be responsive and visually appealing.

5. **Test Case: Login and Signup (Authentication):** Test the login and signup functionalities for students and teachers.

   Steps:

   Sign up with valid student/teacher credentials.

   Login with the newly created credentials.

   Attempt to log in with incorrect credentials.

   Expected Result:

   Successful signup should create a new user in the database.

   Login with correct credentials should grant access to the respective dashboard.

   Login with incorrect credentials should return an error message (e.g., "Invalid username/password").

6. **Test Case: Student Dashboard (Dropout Prediction):** Test the student dashboard to predict if the student is likely to drop out based on the data.

   Steps:

   Log in as a student.

   Verify that the student's data is displayed.

   Check that the dropout prediction is shown (e.g., "Likely to Drop Out" or "Not Likely to Drop Out").

   Expected Result:

   The student's information (name, subject, backlogs, etc.) should be correctly displayed.

   The dropout prediction should appear based on the trained model.

   Predictions should update based on new data input (if applicable).

7. **Test Case: Teacher Dashboard (View Student Data):** Test the teacher dashboard to ensure it displays student data and dropout predictions.

   Steps:

   Log in as a teacher.

   Ensure the teacher can see the list of students and their corresponding dropout predictions.

   Expected Result:

   The teacher dashboard should display a list of students.

   Each student entry should include relevant information (e.g., name, subject, dropout prediction).

   The teacher should be able to view predictions for multiple students at once.

8. **Test Case: Logout Functionality:** Test the logout functionality for both students and teachers.

   Steps:

   Log in as either a student or teacher.

   Click the logout button.

   Verify that the user is logged out and redirected to the home page.

   Expected Result:

   The user should be logged out successfully.

   The user should be redirected to the home page after logout.

   The user session should be cleared.

9. **Test Case: Accuracy Display on Dashboard:** Ensure that the accuracy of the models (Random Forest, Decision Tree, Naive Bayes, and Logistic Regression) is displayed on the dashboard.

   Steps:

   Log in as a teacher or admin.

   Navigate to the dashboard.

   Check if the accuracy of each model is displayed correctly.

   Expected Result:

   The accuracy scores for each model should be visible on the dashboard.

   The scores should reflect the most recent training results.

10. **Test Case: Responsiveness and Cross-browser Compatibility:** Test the responsiveness and cross-browser compatibility of the application.

    Steps:

    Open the website on different devices (mobile, tablet, desktop).

    Open the website in different browsers (Chrome, Firefox, Safari, Edge).

    Expected Result:

    The website should be responsive and adapt to different screen sizes.

    All functionality (login, dashboard, predictions) should work correctly across different browsers.

Chapter 6

Result and Discussions

In our project Random Forest Algorithm has more accuracy then others which is a supervised machine learning algorithm that is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability. Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

Sr No.	**Method**	**Accuracy**	**Precision**
1	Random Forest	92%	90%
2	Decision Tree	89%	87%
3	Logistic Regression	86%	85%
4	Naive Bayes	85%	84%

Table 6.1: Performance Metrics for Models Trained Using Different Algorithms

The performance metrics for dropout defender indicate that the random forest algorithm achieves the highest accuracy at 92%, demonstrating its effectiveness in identifying positive cases. It also shows strong precision at 90%, reflecting its capability to minimize false positives. The high F1 score of 90.5% further emphasizes the random forest robustness in balancing precision and recall. In comparison, the Decision tree algorithm has a slightly lower accuracy of 89%, with a precision of 87%, indicating its reliability despite missing some positive cases. Logistic Regression exhibit accuracies of 86%, with the precision of 85% and Naive Bayes with the accuracies of 85%, with the precision of 84%, respectively,but their lower precision suggests that they may not be as affective in identifying positive cases as Random Forest and Decision Tree.

Chapter 7

Conclusion

In this research it highlights the critical role of machine learning in forecasting student dropout rates and implementing timely intervention strategies. Based on a review of various studies, it's evident that classification algorithms such as decision trees, random forests, logistic regression and naive bayes are frequently employed to predict dropout risks. These algorithms provide a strong foundation for identifying students who are at risk of leaving school by analyzing factors such as academic performance, attendance, and behavior. By utilizing these predictive models, educational institutions can take proactive measures to support students most likely to drop out, ultimately enhancing retention rates.

Additionally, combining machine learning models with realtime data collection systems improves the precision of dropout predictions. Several studies have shown that factors such as grades, attendance, involvement in courses, and socioeconomic status play a key role in predicting student behavior. By integrating these data points with machine learning techniques like logistic regression and neural networks, it becomes possible to create highly effective early warning systems that help prevent dropouts.

In the case of Dropout Defender, selecting the right algorithm depends largely on the characteristics of the dataset and the available features. Some algorithms may be more suitable for smaller, more targeted datasets, while others, such as random forests and neural networks, excel at processing large and complex datasets with a variety of features. As more data is collected over time, these algorithms will refine their predictions, allowing dropout prediction systems to become increasingly effective and adaptable to the evolving needs of educational settings.

Chapter 8

Future Scope

In addition to improving predictive capabilities, Dropout Defender could expand its role in providing personalized support to students. By incorporating adaptive learning modules and recommendation systems, the platform could offer tailored study materials, resources, and skill-building activities based on each student's unique learning needs and goals. This level of customization would not only support academic retention but also promote personal growth and engagement by aligning learning paths with individual student interests and aptitudes.
The platform also has the potential to scale its reach beyond individual institutions by adopting a centralized, cloud-based model. This could enable integration with educational networks, regional education authorities, and even national-level data systems, allowing for a more comprehensive approach to addressing dropout rates on a larger scale. A centralized system would support data sharing and collective insights, helping institutions benchmark their dropout prevention efforts and implement best practices more widely.
Furthermore, the future development of "Dropout Defender" could include the incorporation of advanced analytics and machine learning techniques to enhance its predictive capabilities even further. By utilizing sophisticated algorithms that analyze a broader range of variables, such as socio-economic factors and mental health indicators, the platform could provide deeper insights into the root causes of student disengagement and dropout risk. Addition- ally, integrating feedback loops where students can share their experiences and challenges directly through the platform would allow for continuous refinement of its predictive models.

References

[1] "Predicting Student Dropout in Higher Education Using Machine Learning Techniques," written by O. B. D. Ayoob, R. Shah, A. Ali in year 2022.

[2] "Analyzing Student Dropout Factors Using Predictive Analytics," written by C. T. Munoz, J. A. Asensio, M. A. González in year 2023.

[3] "Machine Learning Approaches for Student Dropout Prediction: A Systematic Review," written by P. M. Sharma, A. B. Rani, N. K. Gupta in year 2023.

[4] "Early Detection of At-Risk Students: A Machine Learning Perspective," written by F. A. Taleb, L. P. Trivedi, J. K. Singh in year 2022.

[5] "Understanding the Impact of Social Factors on Student Retention," written by K. R. Fernando, A. M. Verma, H. N. Khan in year 2023.

[6] "A Hybrid Model for Student Dropout Prediction Based on Machine Learning and Statistical Analysis," written by T. Y. N. Neelam, R. K. Gupta, S. A. Ranjan in year 2023.

[7] "Real-time Monitoring of Student Engagement and Dropout Prediction," written by S. K. M. Adhikari, L. M. Patil, N. R. Joshi in year 2022.

[8] "An Integrated Framework for Predicting Student Dropout in Online Education," written by J. A. F. de Lima, M. L. C. da Silva, P. R. D. Sousa in year 2023.

Appendices

Detailed information, lengthy derivations, and raw experimental observations are presented in the appendices. Each appendix is numbered using Roman numerals (e.g., "Appendix I"). References to published/unpublished literature can appear in the appendices, and they should be placed before the "Literature Cited" section.

Appendix-A: Firebase Setup and Integration

[1] **Create a Firebase Project:**
  - Go to https://console.firebase.google.com/Firebase Console
  - Click on "Add project" and provide a project name.
  - Configure Google Analytics if required and click "Create project."

[2] **Set Up Firebase in Your Web Application:**
  - Navigate to "Project Settings" and select the "General" tab.
  - Under "Your apps," click on the "Web" icon.
  - Register your app and copy the Firebase configuration settings.

[3] **Install Firebase SDK:**
$npminstallfirebase$

[4] **Initialize Firebase in Your Project:**
import initializeApp from "firebase/app";
import getAuth from "firebase/auth";

const firebaseConfig =
apiKey: "AIzaSyBgHvZr6ItIZtb5Eu6Ocxnl8Gw4aWEys8Y",
authDomain: "dropoutdefender.firebaseapp.com",
databaseURL: "https://dropoutdefender-default-rtdb.firebaseio.com",
projectId: "dropoutdefender",
storageBucket: "dropoutdefender.firebasestorage.app",
messagingSenderId: "981706129935",
appId: "1:981706129935:web:03fcc0dd69b11e493d997a" ;

const app = initializeApp(firebaseConfig); const auth = getAuth(app);

[5] **Firebase Authentication:**
$npminstallfirebase/auth$

[6] **Firebase Database Setup:**
$npminstall firebase/firestore$

[7] **Deploying Firebase:**
$npminstall - gfirebase - tools$ firebase login $firebaseinit$ firebase deploy

Appendix-B: Google Colab and Dataset for Student Dropout Prediction

[1] **Access Google Colab:**

- Visit https://colab.research.google.com/Google Colab
- Sign in with your Google account.
- Click "New Notebook."

[2] **Import Required Libraries:** !pip install pandas numpy scikit-learn
import pandas as pd
import numpy as np
from sklearn.model$_{s}electionimporttrain_{t}est_{s}plit$
$fromsklearn.ensembleimportRandomForestClassifier$
$fromsklearn.metricsimportaccuracy_{s}core$

[3] **Load Dataset:** from google.colab import files
uploaded = files.upload()
dataset = pd.read$_{c}sv("student_{d}ropout_{d}ata.csv")$

[4] **Data Preprocessing:**
dataset.isnull().sum()

[5] **Train a Machine Learning Model:**
X = dataset.drop("dropout$_{s}tatus", axis = 1)$
$y = dataset["dropout_{s}tatus"]$
$X_{t}rain, X_{t}est, y_{t}rain, y_{t}est = train_{t}est_{s}plit(X, y, test_{s}ize = 0.2, random_{s}tate = 42)$

model = RandomForestClassifier(n$_{e}stimators = 100, random_{s}tate = 42)$
$model.fit(X_{t}rain, y_{t}rain)$
$predictions = model.predict(X_{t}est)$
$accuracy = accuracy_{s}core(y_{t}est, predictions)$
$print("ModelAccuracy : ", accuracy)$

[6] **Deploying the Model:**

import pickle
filename = 'dropout$_{m}odel.pkl'$
$pickle.dump(model, open(filename,' wb'))$

[7] **Integration with Firebase:**

- Store student predictions in Firebase Firestore.
- Use Firebase authentication for secure access.

Appendix-C: Chart.js Installation

1. Open the terminal and install Chart.js for React project:

   - npm install chart.js react-chartjs

2. Import and use Chart.js in your React components to create visual charts and graphs. In this code everything justified and in times new roman in between its showing different fonts