

Academic Result Management System Using Django

Mudupu sravya, Mujja vamshi, Putti srinath

Department of Computer Science & Engineering, Guru Nanak Institute of Technology, Hyderabad, India

Abstract--- The Academic Result Management System is a web-based application designed to efficiently manage and track student academic performance, including courses, grades, and related data. It simplifies administrative tasks by providing a centralized platform for teachers and administrators to enter, organize, and retrieve information, while offering modules like student, course, result, and report management. The system also enables students to securely access their results through a user-friendly interface. By reducing manual errors and improving efficiency, it supports better decision - making and enhances overall academic management in educational institutions.

Keywords--- Academic Result Management System, Web-based application, Student academic performance, Courses, Grades, Administrative tasks, Centralized platform, Administrators, Student management, Course management, Result management, Report management, Data organization, Information retrieval, Secure access, User-friendly interface, Manual error reduction Efficiency improvement, Decision-making, Academic management, Educational institutions, Student results, Performance tracking Database management, Online result system.

I. INTRODUCTION

The Academic Result Management System is a web-based application designed to simplify and improve the management of student academic records. Manual handling of student data often leads to errors, delays, and inefficiencies, especially in schools and universities with large numbers of students. This system digitalizes the process, making result management faster, more accurate, and transparent.

Developed using Django, the platform allows administrators to manage student profiles, courses, marks, and subject-wise results through a secure admin panel. Students can easily access their results online by entering their roll number, enabling quick and independent result checking. The system follows institutional academic rules to determine pass or fail status based on marks obtained. Its main objective is to reduce human error, save time, improve data accessibility, and provide an efficient solution for academic record management in educational institutions.

A. Objective

The objective of this project is to develop a web-based Academic Result Management System that simplifies the management of student academic records for educational institutions. The system reduces manual work, minimizes errors, and improves administrative efficiency by

providing a centralized platform for managing student details, courses, subjects, and results. Administrators can securely update academic data through a Django superuser login, while students can access their results using their roll numbers. The platform is designed to be user-friendly, secure, scalable, and transparent, ensuring quick and easy access to academic performance records without involving any machine learning or predictive features.

B. Problem Statement

Educational institutions often face difficulties in managing student academic results using traditional manual methods, which are time-consuming, error-prone, and inefficient. Maintaining student records, subject details, and examination results manually increases the chances of data loss and calculation mistakes. Students also experience delays in accessing their results, as they depend on physical notices or administrative staff for updates. In many cases, result management lacks transparency and proper organization, making it difficult to track academic performance effectively. To overcome these challenges, there is a need for a centralized and automated system that can securely store, manage, and retrieve academic data. The proposed Academic Result Management System aims to solve these issues by providing an efficient web-based platform for

administrators to manage student records and results easily. It enables students to access their results online instantly through roll numbers, reducing delays and paperwork. The system improves accuracy, transparency, and operational efficiency in academic result management. Overall, the project addresses the limitations of manual result processing and supports better data management in educational institutions.

Scope of the Project

The scope of this project is centered on the development of a web-based Academic Result Management System using the Django framework to simplify and automate the process of managing student academic records. The system is designed for educational institutions to replace traditional manual methods with a secure and centralized platform for handling student profiles, courses, subjects, and academic results. It allows administrators to efficiently manage student information and upload results, while students can easily access their academic performance online using their roll numbers. The project aims to reduce human errors, eliminate paperwork, improve data accuracy, and ensure transparency

II. LITERATURE SURVEY

Title: A survey of machine learning-based solutions for phishing website detection.

Author: L. Tang and Q. H. Mahmoud

Year: 2021

Description: With the growth of the Internet, network security has become important, especially because of phishing attacks where users are tricked into clicking fake links and their data is stolen. Traditional methods like blacklists cannot detect new phishing websites, so they are not enough. To solve this problem, machine learning is used to predict whether a website is phishing or not by analyzing different features. This helps in detecting new phishing links more accurately, and the paper compares different methods used for phishing detection.

Title: Early detection of students at risk—Predicting student dropouts using administrative data

Author: J. Berens, K. Schneider, S. Görtz, S. Oster, J. Burgh

Year: 2022

Description: This study focuses on identifying students at risk of dropping out using an Early Detection System (EDS). It uses machine learning techniques like regression, neural networks, decision trees, and AdaBoost to predict student performance. The system achieved accuracy between 79%–95% depending on the semester. The research highlights that early identification helps institutions take timely actions to reduce dropout rates, saving resources and improving student success.

Title: A survey of machine learning-based solutions for phishing website detection.

Author: L. Tang and Q. H. Mahmoud

Year: 2021

Description: With the growth of the Internet, network security has become important, especially because of phishing attacks where users are tricked into clicking fake links and their data is stolen. Traditional methods like blacklists cannot detect new phishing websites, so they are not enough. To solve this problem, machine learning is used to predict whether a website is phishing or not by analyzing different features. This helps in detecting new phishing links more accurately, and the paper compares different methods used for phishing detection.

III. SYSTEM DESIGN

A. System Architecture

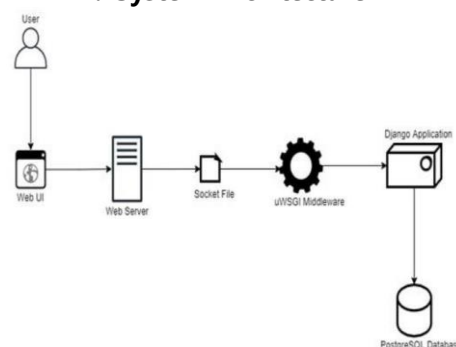


Fig. 1: System Architecture Diagram

2.2 METHODOLOGIES 2.2.1MODULES NAME: >
Admin Authentication Module > Student
Management Module > Class & Subject
Management Module > Result Entry Module >
Student Result View Module > Data Visualization
> Result Management & Validation Module >
Web Interface & User Experience Module >
Database Integration Module 2.2.2 MODULES
EXPLANATION:

1) Admin Authentication Module: The Admin Authentication Module is responsible for securely authenticating the administrator using Django's built-in superuser functionality. This ensures that only authorized users (administrators) can access the backend interface to manage the academic data of students, including results, profiles, and class information.

2) Student Management Module: This module allows the administrator to manage student records efficiently. Administrators can add, update, or delete student profiles, including personal information such as roll numbers, names, and academic details. It provides a centralized view for administrators to keep track of all enrolled students in the system.

3) Class & Subject Management Module: This module enables the administrator to define various classes and subjects for the institution. The admin can assign students to specific classes and subjects, facilitating the organization of student results by subject and class, ensuring that each student is associated with their respective courses.

4) Result Entry Module: The Result Entry Module allows the admin to input or update student results, including marks and final status (pass/fail) for each subject. It streamlines the process of entering academic performance data into the system, enabling efficient tracking and management of student outcomes.

5) Student Result View Module: This module enables students to view their academic results by simply entering their roll number. Once the administrator updates the results, students can securely log in and access their grades and status, ensuring timely access to performance data without requiring manual intervention.

6) Data Visualization: While this system doesn't include complex machine learning models, the Data Visualization module focuses on presenting student academic data in a user-friendly format. This could include charts, graphs, and reports to

help administrators and teachers analyze student performance trends, subject-wise results, or class-level data to make informed decisions.

7) Result Management & Validation Module: This module ensures that all data entered into the system is accurate and consistent. It validates student results, checking for missing or incorrect data, and ensures that the entered information meets the required format and standards before saving it to the database.

8) Web Interface & User Experience Module: The Web Interface & User Experience (UI/UX) module focuses on designing a simple and intuitive web interface using HTML, CSS, and JavaScript. It aims to provide users (both students and administrators) with an easy-to-navigate platform to interact with the system. The interface is built to ensure smooth usability, from logging in to viewing or updating results.

9) Database Integration Module This module handles all interactions with the database, where all academic data is stored. Using Django's ORM (Object-Relational Mapping), the module manages data storage, retrieval, and updates for student records, class

IV. EXISTING SYSTEM VS PROPOSED SYSTEM

A. Existing System

Most systems are manual or basic digital systems Student results are stored but not analyzed deeply Teachers identify weak students manually No early detection of at-risk students Extra assignments are given, but only after poor performance Systems mainly focus on marks entry and result display No automatic alerts or notifications for low-performing students

B. Proposed System —

Uses a web-based Django application for managing student data Early identification of at-risk students using data analysis / ML techniques Provides real-time tracking of student performance Generates automatic alerts/notifications for low-performing students Supports diverse assignments and projects to improve performance Offers personalized feedback and suggestions to students Helps teachers take timely actions Efficient data storage, management, and analysis in one system

V. IMPLEMENTATION

A. Development Environment

Install Python and Django, and set up the development environment

Create a Django project and a new app for the system
Design database models (Student, Course, Marks, Results)
Run migrations to create database tables
Register models in the admin panel for easy data management
Create forms for entering student details and marks
Write views to handle operations like add, update, and view results
Design templates (HTML pages) for user interface
Configure URLs to connect views and templates
Implement login and authentication for users
Test the system for errors and correct result calculation
Deploy the project so users can access it online.

VI. RESULTS AND DISCUSSION

A. Application Interface

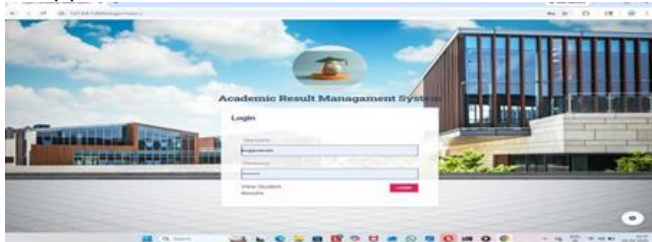


Fig. : Home Page

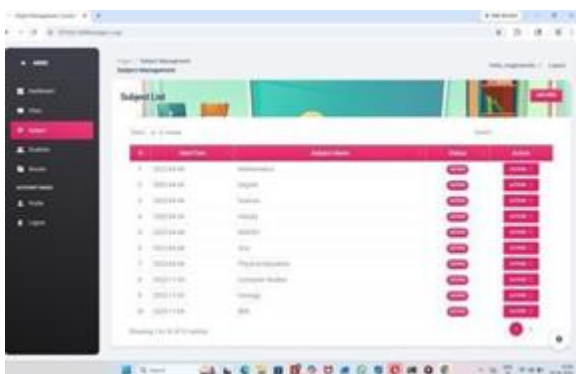


Fig. 11: Registration Page

VII. CONCLUSION

The goal of this project was to develop an efficient and user-friendly web-based system for managing academic results and student performance. By utilizing Django, a robust Python web framework, the system facilitates secure access for both administrators and students. The admin module allows administrators to easily input, manage, and validate student data and results, ensuring that academic performance information is accurate and up-to-date. The student module enables students to view their individual results, promoting transparency and enabling them to track their academic progress. Although the project does not incorporate machine learning algorithms, it effectively addresses the key challenges of academic result management by automating data entry, validation, and retrieval processes. This reduces manual errors and increases the efficiency of handling large amounts of student data.

Acknowledgment

The authors greatly appreciate the assistance of Mr. D V K, Assistant Professor in the CSE Department, for his direction and ongoing support. They are also thankful to Dr B Santhosh Kumar for his expert advice, as well as the professors and laboratory technicians in the CSE Department of Guru Nanak Institute of Technology, Hyderabad, who provided support and cooperation throughout this project.

REFERENCES

- [1] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," in Proc. 15th Eur. Concurrency Eng. Conf. (ECEC), 5th Future Bus. Technol. Conf. (FUBUTEC), 2008, pp. 5–12.

- [2] E. Er, "Identifying at-risk students using machine learning techniques: A case study with IS 100," *Int. J. Mach. Learn. Comput.*, vol. 2, no. 4, pp. 476–480, 2012, doi: 10.7763/ijmlc.2012.v2.171.
- [3] S. Isljamovic and M. Suknovic, "Predicting students' academic performance using artificial neural network: A case study from faculty of organizational sciences," *Eurasia Proc. Educ. Social Sci.*, vol. 1, pp. 68–72, May 2014.
- [4] I. Lykourantzou, I. Giannoukos, V. Nikolopoulos, G. Mparadis, and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Comput. Educ.*, vol. 53, no. 3, pp. 950–965, 2009, doi: 10.1016/j.compedu.2009.05.010
- [5] H. Lakkaraju, E. Aguiar, C. Shan, D. Miller, N. Bhanpuri, R. Ghani, and K. L. Addison, "A machine learning framework to identify students at risk of adverse academic outcomes," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1909–1918, doi: 10.1145/2783258.2788620.
- [6] H. Agrawal and H. Mavani, "Student performance prediction using machine learning," *Int. J. Eng. Res.*, vol. 4, no. 3, pp. 111–113, Mar. 2015, doi: 10.17577/ijertv4is030127.
- [7] L. A. B. Macarini, C. Cechinel, M. F. B. Machado, V. F. C. Ramos, and R. Munoz, "Predicting students success in blended learning—Evaluating different interactions inside learning management systems," *Appl. Sci.*, vol. 9, no. 24, p. 5523, Dec. 2019, doi: 10.3390/app9245523.
- [8] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, and H. Rangwala, "Predicting student performance using personalized analytics," *Computer*, vol. 49, no. 4, pp. 61–69, Apr. 2016, doi: 10.1109/MC.2016.119.



Mujja Vamshi

Student, Computer Science and Engineering,
Guru Nanak Institute of Technology



Putti Srinath

Student, Computer Science and Engineering,
Guru Nanak Institute of Technology

BIOGRAPHIES (Optional not mandatory)



Mudupu Sravya Reddy

Student, Computer Science and Engineering,
Guru Nanak Institute of Technology