Project Documentation

Overview

Project Name	Attendance Marker
Team name	Dream IIT
Team Members	Abhishek Patra(2023ug2010) Pratik Chaurasia(2023ug51) Govind Mehta(2023ug2020) Prince Kumar(2023ug2021)
Project Dates	Start Date: Nov 24, 2024 End Date: Nov 25, 2024
Background	A Python-based attendance management system using NumPy to record, analyze, and manage student attendance efficiently with options for data persistence.
Objectives	1.Streamline Attendance Tracking: Provide an efficient and user-friendly system to record daily attendance for multiple students. 2. Analyze Attendance Trends: Calculate and display attendance statistics, enabling better monitoring of student participation. 3. Ensure Data Persistence: Save and load attendance records to maintain continuity and enable long-term data management.
Target Audience	our faculty members.

Project Specifics

The provided Python code implements a simple attendance management system using the **NumPy** library for efficient data handling. The system offers functionalities for marking attendance,

displaying records, calculating statistics, and saving or loading attendance data.

Functions and Their Roles

1.mark attendance():

- This function collects attendance data for the day.
- It initializes a NumPy array (today) to store daily attendance for all students as binary values (1 for present, 0 for absent).
- Using a loop, it prompts the user to mark each student's status through input (y/n).
- The np.vstack() method appends the daily attendance as a new row to the main attendance matrix.

2.display attendance():

- Displays the current attendance matrix, where rows represent dates and columns correspond to students.
- The function uses simple print() statements to show the attendance records alongside student names.

3.attendance statistics():

- Calculates and displays the attendance percentage for each student.
- It utilizes np.sum() to compute the total presence of each student and divides it by the total number of recorded days.
- The result is presented as a percentage, providing insights into student participation.

4.File Operations (save and load):

- The system employs np.savetxt() to save the attendance matrix into a .csv file for persistence.
- np.loadtxt() is used to reload data, ensuring continuity. Error handling ensures smooth operations if no file exists.

Advantages over a traditional Excel file:

1. Automation and Efficiency

- Automates repetitive tasks like calculating attendance percentages, which would require manual formulas or additional effort in Excel.

- Updates attendance records dynamically with minimal user input, reducing the risk of human error.

2. Flexibility and Scalability.

- Handles large datasets more efficiently using NumPy, while Excel may slow down as data volume increases

3. Data Integrity and Safety

- Prevents accidental overwrites or changes to formulas, which is a common issue in Excel files.

Contributions:

1.Prince kumar(2023ug2023)

Attendance Initialization and Management

- Designed the structure for storing attendance data using **NumPy** for efficiency.
- Created the **attendance matrix**, with rows for dates and columns for students.
- Implemented the logic to add daily attendance (np.vstack) and initialized the students list.

2.Abhishek Patra(2023ug2010)

Attendance Marking System

- Developed the **mark_attendance() function**, enabling the system to record daily attendance interactively.
- Implemented the loop to iterate through students and collect user input (y/n) to mark each student's status (1 for present, 0 for absent).
- Managed the integration of daily attendance into the main attendance matrix.

3.Govind Mehta(2023ug2020)

Attendance Statistics and Analysis

- Built the **attendance_statistics() function** to calculate and display each student's attendance percentage.
- Used **NumPy functions** like np.sum() for efficient data processing.
- Implemented the logic to calculate and present statistical insights, making the data actionable.

4.Pratik Chaurasia(2023ug51)

Data Persistence and User Interface

- Implemented **file operations** using np.savetxt() and np.loadtxt() to save and load attendance data in .csvformat.
- Added error handling for loading files, ensuring smooth user experience.
- Designed the **menu-driven interface** for seamless navigation between options like marking attendance, displaying records, and saving/loading data.

Conclusion

This code demonstrates effective use of functions for modular programming and NumPy for efficient data processing. It provides a simple yet robust framework for managing attendance in educational or organizational contexts.

THANK YOU❤️