Student Dashboard - Project Report

Overview

The Student Dashboard is a web application built using the Streamlit framework. It provides an interactive platform for managing and visualizing student data, including names, grades, and performance levels. The application is designed with a modern dark theme and includes various features for data input, visualization, and statistical analysis.

Features

- Dark Theme and Styling: A custom dark theme is applied using CSS for a visually appealing interface.

- Header and Footer: A header image is included to make the app more engaging, and a footer with branding is added.

- Data Input: Users can input student details (name, grade, and performance) through text fields.

- Data Display: Displays all student records in a sorted table (sorted by performance and grade).

- Statistics: Calculates and displays key statistics for grades (Mean, Median, Mode).

- Visualizations: Includes a histogram for grades and a pie chart for performance distribution.

- Individual Sorting: Allows sorting data by name, grade, or performance with interactive visuals.

Technical Highlights

- Streamlit Framework: Used for creating interactive web applications.

- Database Integration: Interacts with a database through the Student class.

- Plotly for Visualizations: Creates interactive and visually appealing charts.

- CSS for Custom Styling: Applies a dark theme and improves the app's appearance.

Strengths

- User-Friendly Interface: Intuitive and easy to use.

- Interactive Visualizations: Enhances data exploration and understanding.

- Modern Design: Dark theme and custom styling make the app visually appealing.

- Comprehensive Features: Combines data management, statistics, and visualization.

Suggestions for Further Improvements

- Error Handling: Add error messages for invalid inputs and handle database connection errors.

- Performance Optimization: Cache database queries to improve performance for large datasets.

- Export Options: Allow users to export the displayed data (e.g., as a CSV file).

- Authentication: Add user authentication to restrict access to sensitive data.

- Mobile Responsiveness: Optimize the layout for better usability on mobile devices.

Explanation of Algorithms

The Student Dashboard project uses several algorithms for sorting, statistical calculations, and data processing. Below is a detailed explanation of the algorithms used:

1. Merge Sort

Purpose: Used to sort grades in ascending order. Merge Sort is a divide-and-conquer algorithm that splits the list into smaller sublists, sorts them, and merges them back together.

Advantages: Stable sorting algorithm with a time complexity of $O(n \log n)$.

2. Quick Sort

Purpose: Used to sort names in alphabetical order. Quick Sort selects a pivot element, partitions the list into elements less than, equal to, and greater than the pivot, and recursively sorts the partitions.

Advantages: Efficient for large datasets with an average time complexity of $O(n \log n)$.

3. Radix Sort

Purpose: Used to sort performance levels based on a predefined order (e.g., 'A+', 'A', 'B+', etc.). Radix Sort groups data into buckets based on numeric mappings and concatenates the buckets to produce a sorted list.

Advantages: Non-comparative sorting algorithm with a time complexity of $O(n)$.

4. Statistical Calculations

The project includes algorithms to calculate mean, median, and mode for grades:

- Mean: The average of all grades, calculated in $O(n)$ time.

- Median: The middle value in a sorted list, calculated in $O(n \log n)$ time due to sorting.

- Mode: The most frequently occurring value, calculated in $O(n)$ time.

5. Sorting Students

Purpose: Sorts students by performance and grade. Students are first sorted by performance (using a predefined order) and then by grade in descending order.

Advantages: Custom sorting ensures meaningful organization of student data.

Conclusion

The Student Dashboard is a well-designed and feature-rich application for managing and visualizing student data. It effectively combines functionality, aesthetics, and interactivity, making it a valuable tool for educators and administrators. With minor enhancements, it can become even more robust and user-friendly.