Student Grade Monitoring System

Team Name: PS5

Project Requirements and Commands

Project Description

The name of the project is the Student Grade Monitoring System. It helps track and manage students' grades efficiently using a balanced tree data structure. It has features that allow users to add, view, update, and delete grades while maintaining data integrity and quick access times.

- Problem/needs Addressed: Simplifies the process of managing and analyzing student grades. The number of courses and their respective grades can make it a hassle to track a student's progress. This project will simplify grade monitoring.
- Reason Chosen: Practical, small-scale system that demonstrates non-linear data structure implementation.
- Inspiration: Real-world school grade management portals and teacher grading systems.

Technical Requirements

The key technical requirements will be addressed through the following approach:

- Primary Data Structure: Binary Search Tree (BST)
- How it solves the problem: BST allows storing students' grades based on ID, enabling efficient searching, insertion, and deletion.

- Primary operations: Insert student grade, Search grade, Update grade, Delete record, Display grades.
- Interface: Command-line interface (menu-based).
- Persistent Storage: CSV file containing (studentID, name, course, grade, GPA).
- The project will be fully implemented in C++.

Application Commands

The following application commands will be supported by the application:

1. Add Grade – Inserts new student grade record into the BST.
2. Search Grade – Searches for a student's grade using ID or name.
3. Update Grade – Modifies the grade or GPA of an existing record.
4. Delete Grade – Removes a student's grade record.
5. Display Grades – Shows all student grades in sorted order (in-order traversal).
6. Save Data – Saves all grade records from BST into CSV file.
7. Load Data – Loads student grade records from CSV file into BST.
8. Exit – Exits the program after optional data saving.

Member Contributions

All of us brainstormed different ideas and landed on this topic, as it seemed the most suitable for our interests and met the prerequisites. We also discussed all three components of how we could implement the project. After that,

Sujal wrote the project description in his own words based on what we discussed.

Sambhav wrote the technical requirements portion, summarizing what we discussed.

Prashant wrote the application commands to fulfill the project based on our goals.

This section of the document had all three of us writing what we did. To keep the document consistent, we kept the language in second person.