Ghulam Ishaque Khan Institute of Engineering Sciences and Technology

Semester Project
Data Structures and Algorithms
(ES221)

Project Title: **Student Helper Program**

Student Names	Reg #	Degree
Ayaan Azam	**2023596**	**ES**
Raahim Ali	**2023563**	**ES**
Anas Ahmed	**2023114**	**ES**

1. Project Title: Student Helper Program

Problem Statement:
Students often face challenges in effectively managing their time, budget, and academic integrity. The **Student Helper Program** is designed to assist students in these key areas by providing the following functionalities:

i. **Time Management:** Enables task creation, dependency management, priority handling, postponement, and tracking of active, pending, and completed tasks.
ii. **Budget Management:** Offers two budgeting options—automatic and user-defined allocation—with further subdivisions on a weekly or daily basis.
iii. **Plagiarism Detection:** Compares two text files for similarities, generates detailed reports, and analyzes text complexity using the Flesch Reading Ease Score.

2. Data Structures Selection:

- **Plagiarism Detection Components** – Hash Maps, List, Vectors
- **Read User Desired Files to Detect Plagiarism** – File Handling
- **Generate Report** – Strings, Vectors
- **Analysis of Text Complexity** – Strings
- **Task Creation** – No Specific Data Structure
- **Priority Management** – Priority Queue (Min-Heap)
- **Task Deferral and Reactivation** – Priority Queue (Min-Heap), Linked List
- **Task Completion** – Priority Queue (Min-Heap), Linked List
- **Task Display** – Priority Queue (Min-Heap), Linked List
- **Manage Task Dependency** – Graph, Topological Sort
- **Choosing Between Two Budget Allocation Choices** – No Specific Data Structure
- **Entering Priorities** – Doubly Linked List, Stack (using Linked List)
- **Entering Divisions & Subdivisions for Each Priority** – Doubly Linked List

- **Display & Search for Specific Priorities** – Doubly Linked List, Recursion
- **Viewing History** – Stack (using Linked List)

3. Algorithmic Approach:

- **Budget Manager:** Uses **Doubly Linked Lists** for priorities, **Stacks** for history, and **Recursion** for searching. Budget allocation follows a decision-based approach.
- **Time Manager:** Implements **Priority Queue (Min-Heap)** for task scheduling, **Graphs & Topological Sort** for dependency management, and **Linked Lists** for task deferral.
- **Plagiarism Detector:** Uses **N-gram Comparison** with **Hash Maps & Vectors**, **Threshold Matching**, and **Flesch Reading Ease Score** for text complexity analysis.

4. Input & Output Design:

- **Budget Manager:** Users input budget type and priorities. Outputs budget breakdown and history.
- **Time Manager:** Users enter tasks and dependencies. Outputs sorted tasks (active, pending, completed).
- **Plagiarism Detector:** Users upload files and set parameters. Outputs plagiarism percentage, reports, and readability scores.

5. Course Concepts Application:

- **Linked Lists (Weeks 4-5):** Used for **budget priorities** and **task deferral/history tracking**.
- **Stacks & Queues (Week 6):** Implemented for **undo operations in budget management** and **task scheduling**.
- **Recursion (Week 7):** Applied in **searching for priorities** efficiently.
- **Sorting Algorithms (Weeks 8-9):** Used for **priority sorting in task management**, utilizing **merge sort, quicksort, and heap sort** where needed.