NATIONAL UNIVERSITY OF COMPUTER
AND EMERGING SCIENCES

ACADEMIC PRODUCTIVITY TOOL

Group Members:

- Rameen Ramzan – 24k-0557
- Umama Zubair – 24k-0621
- Amna Sami – 24k-0561

Submission Date: May 12, 2025

1. EXECUTIVE SUMMARY

OVERVIEW:

The "Academic Productivity Tool" is a desktop application designed to streamline the academic life of students through effective time management, note organization, GPA calculation, and motivation via a streak system. It utilizes core Object-Oriented Programming (OOP) principles and is built using C++ with a Qt GUI.

KEY FINDINGS:

The tool successfully integrates multiple essential academic functionalities under one system. It offers a clean interface, modular architecture, and a gamified approach to study management, enhancing user engagement and learning outcomes.

2. INTRODUCTION

BACKGROUND:

Academic workloads can be overwhelming. Students often juggle GPA tracking, study planning, and note organization. Our OOP-based solution aims to tackle these challenges by offering a consolidated and efficient interface.

Project Objectives:

- Implement a C++ & Qt GUI desktop application.
- Apply Encapsulation, Inheritance, and Polymorphism.
- Provide tools: GPA Calculator, Study Planner, Note Organizer, and Streak System.
- Create a user-friendly and responsive interface.

3. PROJECT DESCRIPTION

SCOPE:

Included:

✓ GPA & Grade Calculator
✓ Study & Exam Planner
✓ Study Streak System
✓ Note Organizer
✓ Transcript Generator
✓ Task Completion Tracker

Excluded:

✗ AI-based recommendations
✗ Scholarship checker
✗ Career planning system

Technical Overview:

- Language: C++
- GUI Framework: Qt
- IDE/Tools: Qt Creator, Visual Studio Code
- Filing

4. METHODOLOGY

APPROACH:

We followed an iterative development model. Each module was first prototyped, coded, then refined through testing cycles.

ROLES AND RESPONSIBILITIES:

👤 ***Rameen Ramzan:***Dashboard & GPA Calculator
These modules likely deal with displaying data and academic performance.
• dashboard.cpp
• dashboard.h
• gpacalculator.cpp
• gpacalculator.h

👤 ***Amna Sami:*** Main Application & Note Organizer
Handles core UI and note management features.
• main.cpp
• mainwindow.cpp
• mainwindow.h
• noteorganizer.cpp
• noteorganizer.h

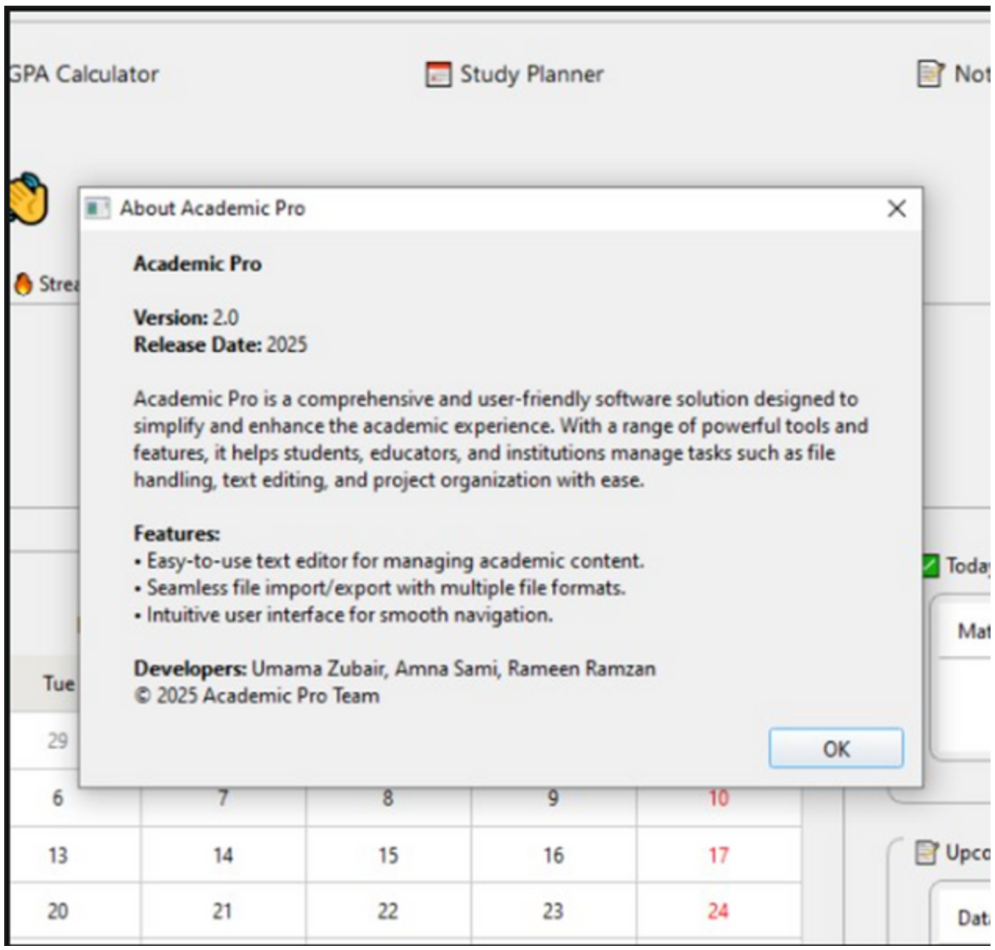
👤 ***Umama Zubair:*** Streak System & Study Planner
Focuses on user habits and planning.
• streaksystem.cpp
• streaksystem.h
• studyplanner.cpp
• studyplanner.h
🗂️ ***Common Files*** (Shared Responsibility or Maintainer Role)
These files are project-wide and can be handled by any member or assigned to someone managing integration:
• OOP_PROJECT.pro
• OOP_PROJECT.pro.user
• build/ (Generated build directory, typically auto-managed)

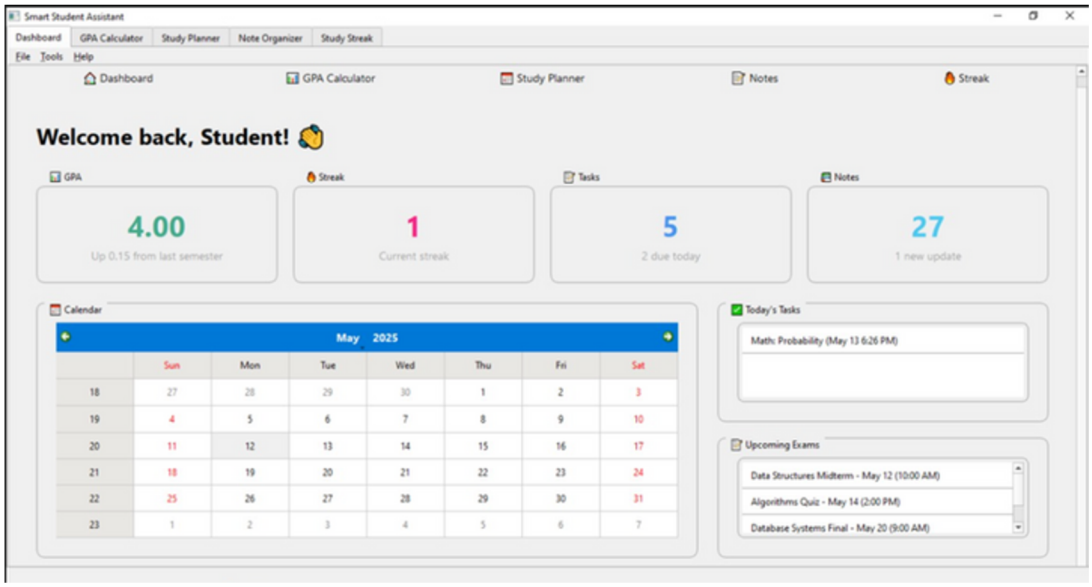
5. PROJECT IMPLEMENTATION

DESIGN AND STRUCTURE:

The project consists of several modules with distinct responsibilities, including:

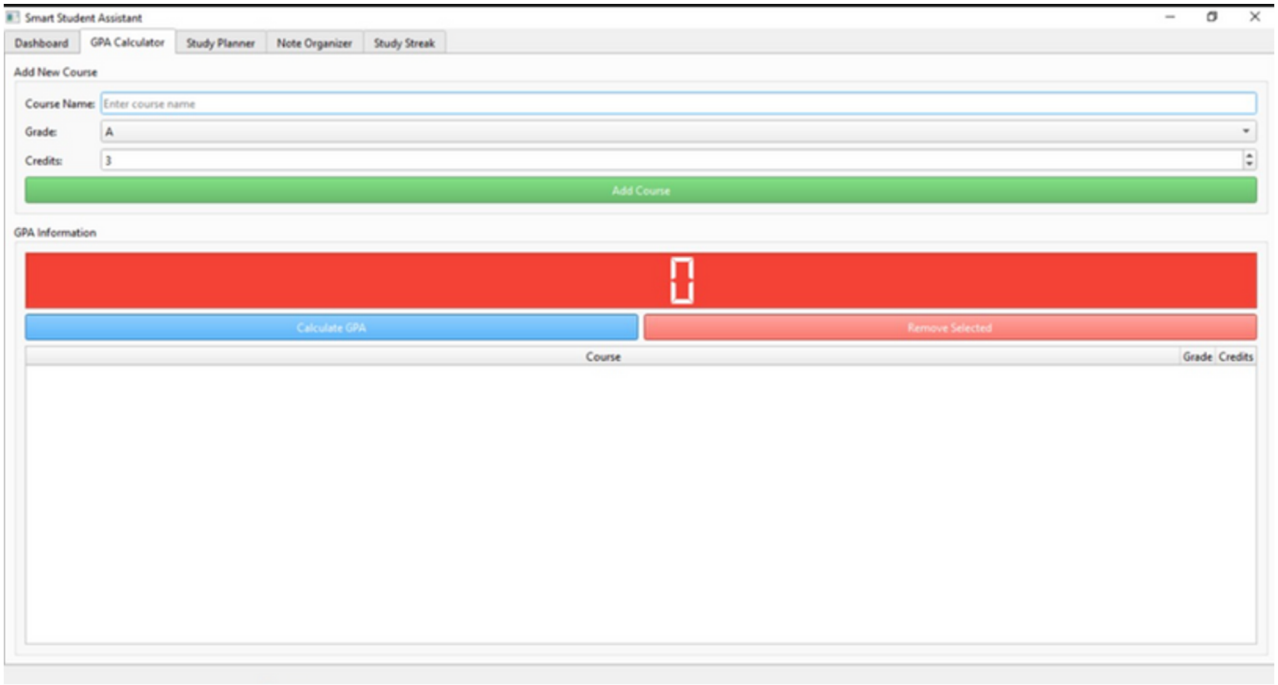
- main.cpp: Entry point of the application.
- dashboard.cpp/h: Central GUI panel connecting all modules.
- gpacalculator.cpp/h: Handles GPA input, processing, and result display.
- studyplanner.cpp/h: Manages time slots and exam preparation tasks.
- streaksystem.cpp/h: Implements a gamified reward system for studying.
- noteorganizer.cpp/h: Allows creation, editing, and deletion of notes.
- mainwindow.cpp/h: Qt logic for GUI management.

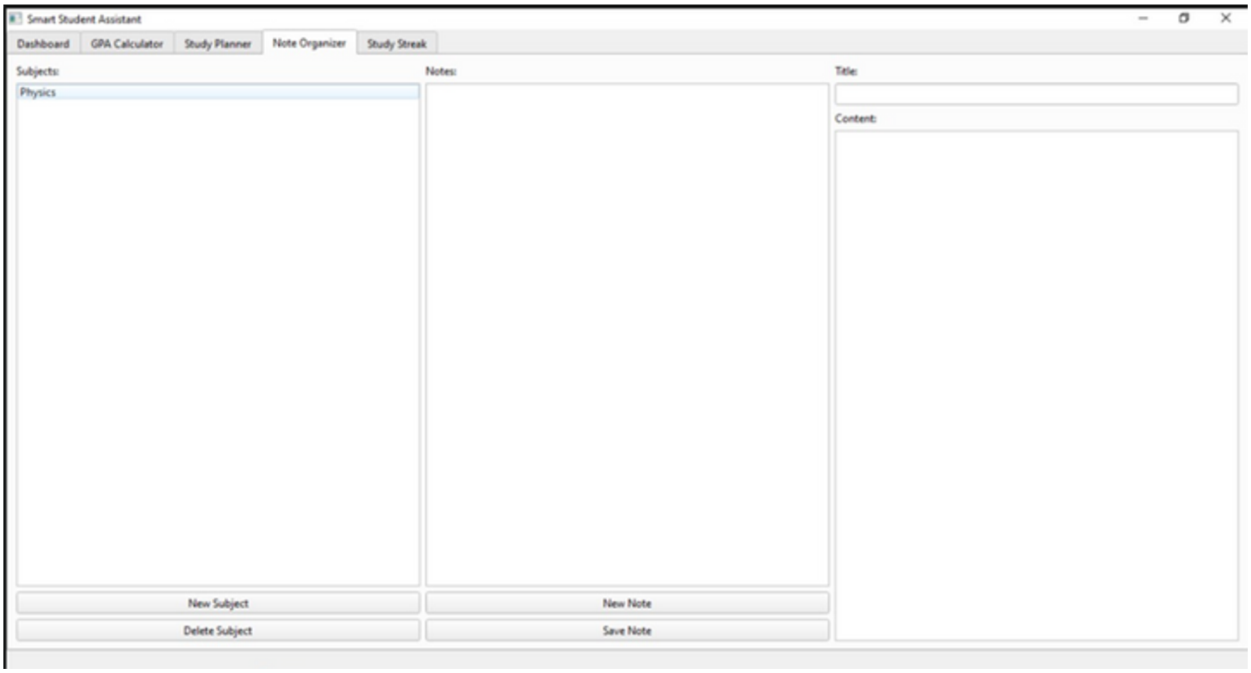




Functionalities Developed:

- GPA calculation using customizable grading systems
- Calendar-based planner with reminders
- Note organizer with search & categorization
- Study streak system using counters and motivational rewards
- Filing for saving user data





Challenges Faced:

• *Synchronizing GUI signals/slots in Qt*

• *Ensuring data persistence without loss on app restart*

• *GUI layout management across different screen resolutions.*

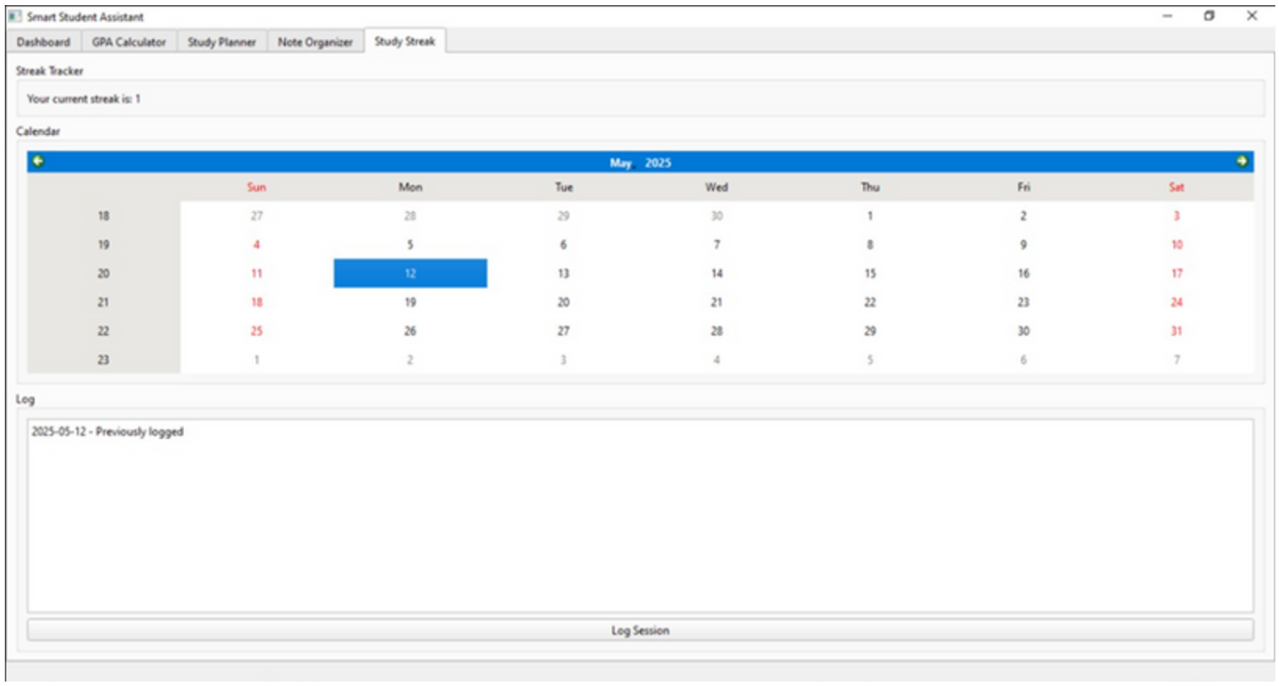
. Integrating filing in C++.

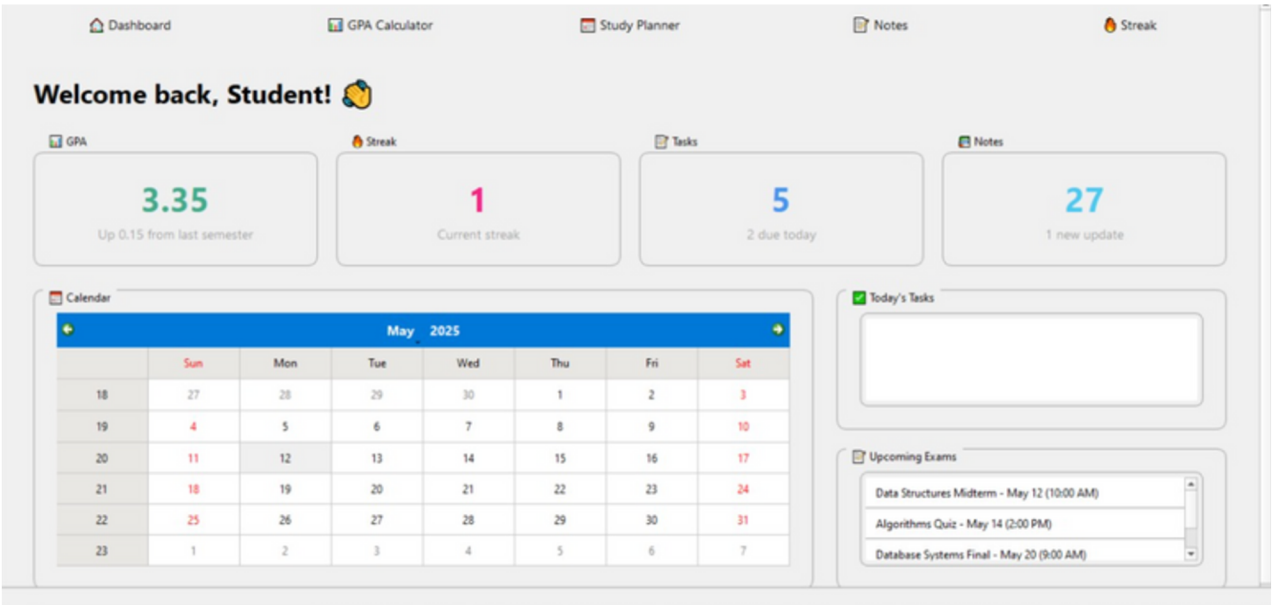
6. RESULTS

Project Outcomes:

The application meets all specified objectives, offering a responsive and fully functional student productivity assistant. Each module is independently testable and interlinked via a clean GUI.

Screenshots and Illustrations:





Testing and Validation:

- Manual testing was performed on each module.
- Edge cases (e.g., zero GPA credits, missing fields) were handled.
- Final integration testing confirmed smooth transitions between components.
- GUI responsiveness was validated on different systems.

7. CONCLUSION

Summary of Findings:

The project successfully addresses the problem of academic task management. Using Qt and OOP in C++, we developed a robust, modular, and practical tool for students.

Final Remarks:

This project not only enhanced our programming skills but also taught us collaborative software development, GUI design, and real-world application of Object-Oriented Programming principles.