**COMSATS University Islamabad,**

**COMSATS Road, off GT Road, Sahiwal, Pakistan**

# Project DOCUMENTATION

# for

## DataNexus: Data analyzer with Class-Wise and Course Wise Academic Performance Analyzer for CUI

***By***

| | |
|---|---|
| **Habib Ur Rehman** | **CIIT/FA23-BCS-251/SWL** |
| **Maryam** | **CIIT/FA23-BCS-454/SWL** |
| **Aqsa** | **CIIT/FA23-BCS-457/SWL** |
| **Maham Ayesha** | **CIIT/FA23-BCS-461/SWL** |

***Supervisor***

**Mr. Imran Shehzad**

***Bachelor of Science in Computer Science (2023-2027)***

# Table of Contents

# 1 Abstract

This project presents the development of a **desktop-based Data Analysis System** using **Python and PyQt5** that aims to streamline the way academic data is interpreted and visualized. The system is tailored for **educational institutions** that handle large amounts of student records, and for **students or developers** who wish to explore data trends and comparisons in a user-friendly interface.

## 1.1 Interactive GUI and Visualization

Built using **PyQt5**, the graphical user interface is designed with a clean layout and modern components such as:

- A **sidebar menu** for navigating between Batch, Course, Section, and Profile views,
- A **header panel** with a "Choose File" button for easy file uploading,
- **Modular cards** that dynamically update based on the loaded data,
- Error handling pop-ups for unsupported files or invalid formats.

Each dashboard view offers **interactive charts or structured tables**, providing users with a visual representation of student performance trends. This makes the system especially useful in **classroom analytics, internal assessments, and academic audits**.

# 2 Introduction

In modern academic institutions, large amounts of student-related data are generated every semester, including course enrollments, grades, attendance, and section distributions. Managing and analyzing this data manually using spreadsheets or basic tools is often inefficient, time-consuming, and prone to human error.

This makes the system highly suitable for **teachers**, **academic coordinators**, and **university administrators** who need to make data-driven decisions about student performance and curriculum effectiveness.

# 3 Problem Statement

- Academic institutions generate large volumes of student data every semester.
- Data includes grades, attendance, batch details, and section distributions.
- Manual analysis using tools like Excel is often time-consuming and inefficient..

# 4 Proposed Solution

To overcome the challenges described in the problem statement, this project introduces a **desktop-based Data Analysis System** built using **Python and PyQt5**. The system is designed to be both **user-friendly** and **functionally rich**, with features specifically tailored for academic data analysis.

## 4.1 Key Features of the Proposed System

| Feature | Description |
|---|---|
| **Excel File Upload Support** | Allows users to load .xlsx files with structured student data. |
| **Course-Wise Visualization** | Analyzes and displays statistics for each course offered. |
| **Section Comparison** | Compares student performance across different sections within a course. |
| **Student Profile Viewer** | Provides a complete academic overview of an individual student. |
| **Modern GUI Design** | Sidebar navigation and dashboard layout ensure a clean and intuitive UI. |
| **Student Profile Viewer** | Sidebar navigation and dashboard layout ensure a clean and intuitive UI. |
| **Error Handling** | Automatically detects and informs users about invalid files or unsupported formats. |

## 4.2 Visual Navigation through Sidebar

The system includes a **modern left-hand sidebar** that acts as a navigation menu. It contains buttons for:

### 4.2.1 Batch View

Displays overall statistics of a specific batch, including total students and average performance.

### 4.2.2 Course Analysis

Shows detailed performance metrics for each course offered in the dataset. It Allows teachers to identify which courses are performing well.

### 4.2.3 Section Comparison

Compares academic results across different sections of the same course. Useful for identifying section-wise strengths or performance gaps.

### 4.2.4 Student Profile

Presents an individual student's complete academic record, including enrolled courses and grades.

#### 4.2.5 Complete List

Displays a searchable list of all students along with key details and scores

| System | Weakness | Our Solution |
|---|---|---|
| Excel | Manual analysis required | Automated visualization |
| Tableau | Paid software | Free and open-source alternative |
| Raw Python | No GUI | PyQt5-based interactive interface |

# 5 Advantages

- **Drag-and-drop** support with **file validation**.
- **Interactive charts** and **data cards** for better analysis.
- **Modular design** allows easy addition of **new features**.
- **User-friendly interface** for **non-technical users**.
- **Real-time updates** of data views after **file upload**.
- **Offline functionality** after installation.
- Built entirely with **free** and **open-source tools**.
- **Educational value** for students learning **data analysis** and **GUI development**.

# 6 Scope

- Designed for universities to analyze performance of students across different semesters.
- Useful for teachers to track student progress efficiently through visual reports.
- Allows students to review their academic data to understand strengths and weaknesses.
- Helpful for developers and learners exploring visualization using academic data.

# 7 Modules:

## 7.1 GUI Frontend (main.py)

- Sidebar with buttons: Batch, Course, Profile, Section, List.
- Header with “Choose File” upload option.

### 7.2 File Reader (Data.py)

- Reads Excel (.xlsx) files.
- Extracts and stores structured student data.

### 7.3 CourseCard

- Displays courses and related performance.

### 7.4 SectionComparisonCard

- Compares performance across different sections.

### 7.5 ResultListCard

- Displays complete student list with stats.

### 7.6 ProfileCard

- Shows individual student profiles.

### 7.7 HomeCard

- Dashboard with batch-level data overview.

## 8 System Limitations

- Currently supports only **.xlsx** format for file uploads.
- May struggle with large files that have inconsistent **formatting**.
- Does not yet offer report **download** or export features.

## 9 Development Methodology

The project was developed using the Incremental Development methodology, which focuses on building the system step by step. Initially, the graphical user interface (GUI) was designed using PyQt5 to ensure a clean and intuitive layout. After the interface was in place, the next phase involved integrating Excel file handling, enabling the system to read and process academic data.

## 10 Tools and Technologies:

| Tool | Version | Purpose |
|---|---|---|
| Python | 3.x | Backend logic |
| PyQt5 | Latest | GUI and Widget Framework |
| VS Code | Latest | Code writing and debugging |

# 11 Data Collection Approach

- Sample Excel files from class datasets.
- Manually prepared and cleaned for uniform structure.
- Fields: Student ID, Name, Batch, Section, Courses, Grades.

# 12 Concepts

### 12.1.1 GUI Development (PyQt5)

The user interface was built using PyQt5, with components like QStackedWidget and QFrame to manage different screens within the same window.

### 12.1.2 Event-Driven Architecture

The system follows an event-based structure where actions like button clicks trigger responses such as switching das hboard cards or loading files.

### 12.1.3 Data Extraction

Excel data is extracted using pandas or openpyxl libraries, allowing smooth reading and processing in the background

# 13 User Accessibility

- **Simple Interface:** The clean and organized layout makes navigation intuitive and effortless.
- **One-Click File Upload:** Users can upload Excel files with a single click using the "Choose File" button.
- **Sidebar Navigation:** A visual sidebar provides direct access to all major sections like Batch, Course, and Profile views.
- **Error Handling Messages:** Friendly pop-up alerts guide users when a file is missing, incorrect, or unreadable.
- **Cross-Level Usability:** Suitable for teachers, students, coordinators, and academic admins.

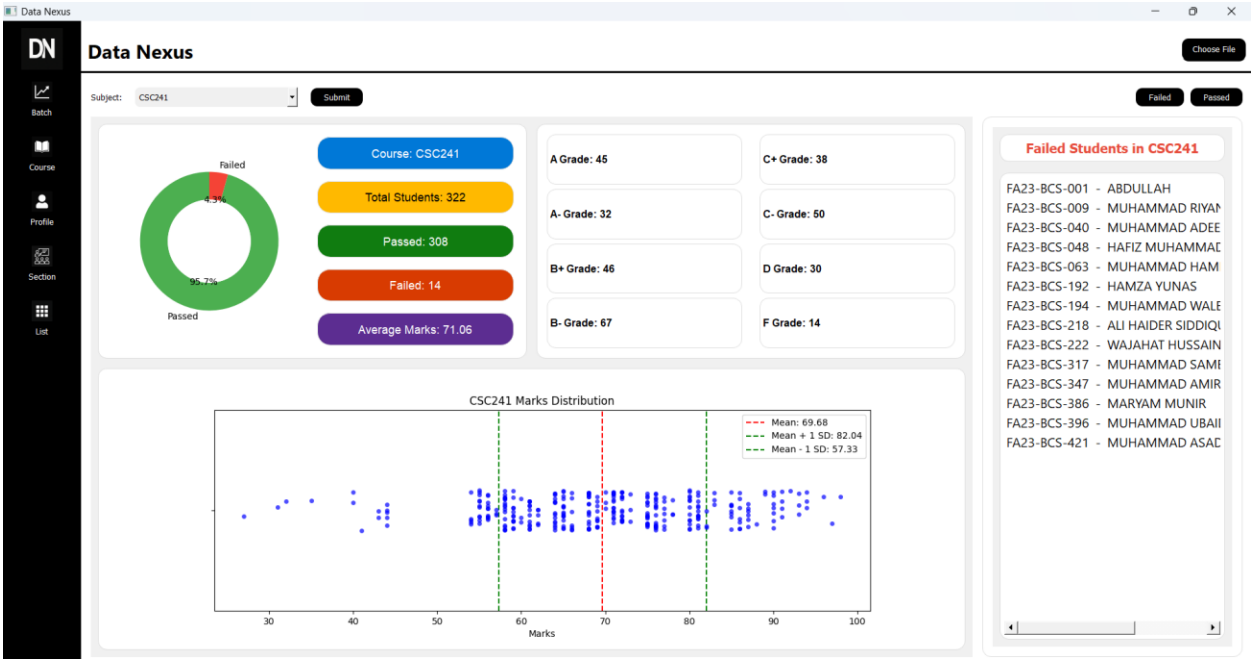
# 14 Mockups

The mockups provide a visual representation of the system's interface before actual development. These designs were created to plan the layout, structure, and user interaction flow. Key screens include:

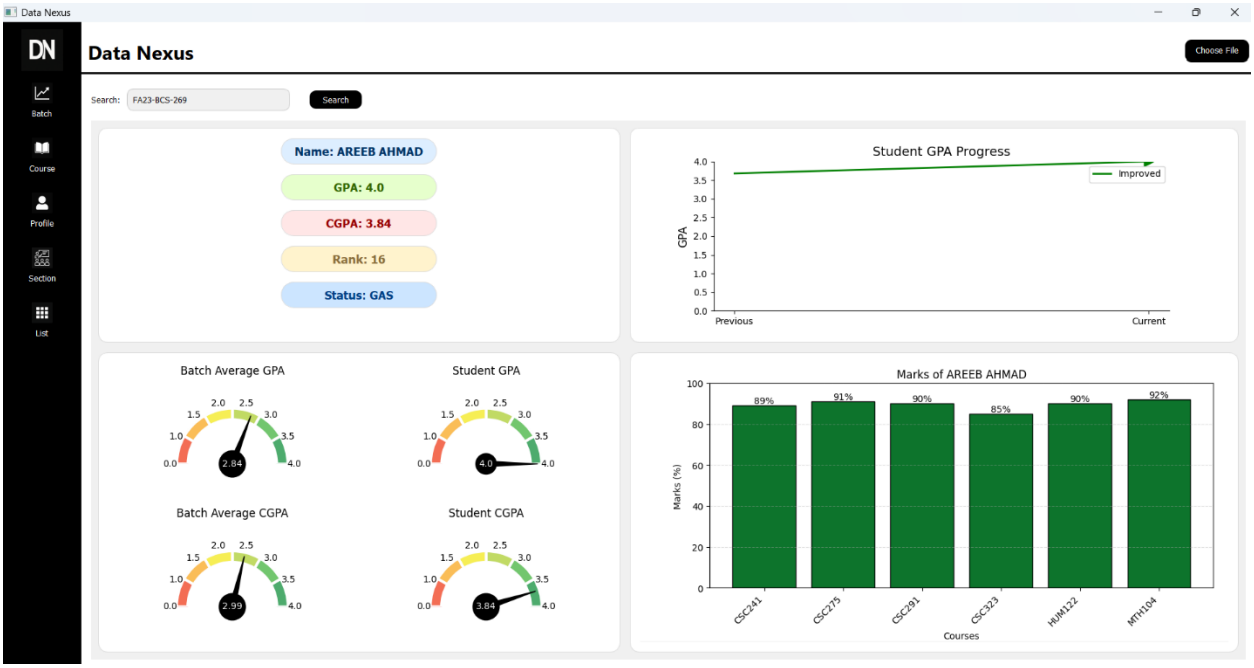
- **Dashboard View** – Main panel showing batch-wise, course-wise, and profile-based data cards.
- **Upload Section** – Area where users can upload Excel files for processing.
- **Charts & Reports** – Visual representations of performance data using bar charts and summary cards.

## Batch Analysis

# Course wise analysis




# Student profile



# Section wise comparison

# 15 Conclusion:

This project successfully shows how **PyQt5** and Python can be used to build a simple and organized data analysis system. It reads and processes student **data** from Excel files and shows the results across different categories like batches, courses, and profiles. The modular **architecture** makes it easy to add new features in the future, such as cloud support or advanced charting.