Student Profile Management System - Documentation



**Documentation**

Fundamentals of Data Science

UFCFK1-15-0

**Submitted by:**

**Name:** Shrawan Budhathoki and Jatan Timilsina and Biplab Gajurel

**Section:** I

**Student ID:** 25024656

# Table of Contents

1. Introduction  
2. Objective  
3. System Overview  
4. Module Breakdown  
 - A. Student 1: GUI and Authentication  
 - B. Student 2: Data Handling and User Management  
 - C. Student 3: Analytics and Visualization  
5. Technologies Used  
6. File Structure  
7. Conclusion

# 1. Introduction

This Student Profile Management System is a Python-based desktop application built using Tkinter for GUI, NumPy for data analytics, and Matplotlib for visualizing performance data. It supports two roles: Admin and Student.

# 2. Objective

The goal is to manage student academic profiles, including login credentials, personal information, grades, and extracurricular activities (ECAs), while providing administrators with the ability to visualize and analyze student performance.

# 3. System Overview

- Login System: Validates user credentials and redirects based on role.  
- Admin Features:  
 - Add/Delete/Update Student Info  
 - View Analytics Dashboard  
- Student Features:  
 - View Profile, Grades, and ECA  
 - Update Name  
- Analytics:  
 - Bar graph of average grades  
 - Scatter plot of ECA count vs. performance

# 4. Module Breakdown

## A. Shrawan Budhathoki: GUI and Authentication (from main.py)

Responsibilities:  
- Build the main application window and styling  
- Create login screen and handle login authentication  
- Implement frame switching for different pages  
- Define views for Admin and Student dashboards  
  
Key Functions:  
- LoginFrame, MainApp  
- login\_success(), build\_main\_ui(), show\_\*() functions  
- Sidebar navigation and Tkinter styling

## B. Biplov Gajurel: Data Handling and User Management (from file\_handler.py and parts of main.py)

Responsibilities:  
- File management (users, passwords, grades, eca)  
- Implement Admin actions like:  
 - Add new users  
 - Delete users  
 - Update grades and ECAs  
  
Key Files:  
- file\_handler.py for reading, writing, appending files  
- AdminAddUserFrame, AdminDeleteUserFrame, AdminUpdateInfoFrame  
  
Sample Functionality:  
- write\_file() and append\_file() used to modify users.txt, grades.txt, and eca.txt

## C. Jatan Timilsina: Analytics and Visualization (from analytics.py and main.py)

Responsibilities:  
- Analyze data from grades and ECAs  
- Use NumPy for average calculations  
- Use Matplotlib to plot graphs within the Tkinter GUI  
  
Key Files:  
- analytics.py – contains get\_analytics\_data() function  
- AdminAnalyticsFrame – shows bar and scatter plots  
  
Visual Output:  
- Average marks per subject (bar chart)  
- ECA count vs academic performance (scatter plot)  
- Highlights students with average grade < 40

# 5. Technologies Used

|  |  |
| --- | --- |
| Component | Technology |
| GUI | Tkinter + ttk |
| Data Handling | Python File I/O |
| Analytics | NumPy |
| Visualization | Matplotlib |

# 6. File Structure

project-folder/  
-main.py # GUI logic  
-analytics.py # Analytics functions  
-file\_handler.py # File read/write  
-users.txt # User data  
-passwords.txt # User credentials  
-grades.txt # Grade data (5 subjects)  
-eca.txt # ECA data (student ID + activities)  
-data # Folder where .txt files are stored

# 7. Samples

Login interface

A screenshot of a computer

AI-generated content may be incorrect.

Entering username and passwordA screenshot of a computer

AI-generated content may be incorrect.

Students detals

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Admin portal loginA screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# 8. Conclusion

This system integrates file-based data storage with a GUI and analytics module, ensuring an interactive experience for both students and administrators. Tasks are modular and manageable by three contributors, making collaboration smooth and logical.