# Reflective Essay: Student Management System Project Introduction

Our group developed a **Student Management System (SMS)** using Python, focusing on

user authentication, grade management, and extracurricular activity (ECA) tracking. The system caters to two user roles: **students** (who view grades/ECA) and **admins** (who

add/delete users, grades, and ECA details). This essay reflects on the code’s functionality,

challenges faced, team contributions, and personal learnings.

**Code Analysis**

1. **Key Features**
   * **Authentication**:
     + Validates credentials from adminpassword.txt (admins) or password.txt (students).
     + Implements password length checks and confirmation.

# User Management:

* + - Admins can **register/delete users** (register\_user(), delete\_user()).
    - Students can only view their data.

# Grade Management:

* + - Admins add grades via add\_marks(), stored in grades.txt.
    - Students view grades using Pandas for tabulation and Matplotlib for visualizations.

# ECA Tracking:

* + - Admins add ECA details (add\_eca()) to eca.txt; students view them (view\_eca()).

# Technical Strengths

* + **Modular Design**: Separate methods for each functionality (e.g., login(), add\_marks()).
  + **Error Handling**: Uses try-except blocks for file operations and input validation.
  + **Data Visualization**: Generates bar plots of student averages using Matplotlib.

# Limitations s Improvements

* + **Security**: Passwords are stored in plaintext (should use hashing).
  + **Scalability**: Text files are inefficient for large datasets; a database (SQLite) would be better.
  + **UI**: Command-line interface (CLI) is less user-friendly; a GUI (Tkinter) could enhance usability.

# Group Contributions

|  |  |  |
| --- | --- | --- |
| **Member** | **Role** | **Contributions** |
| **Resham**  **Shah** | Project Lead | Designed core architecture, implemented PDF  extraction (if used), integrated modules. |
| **Aaditya**  **Kumar** | Data Logic  Specialist | Developed grade calculation logic, optimized Pandas  operations, debugged file I/O. |
| **Devyani**  **Khadka** | Documentatio  n C Testing | Wrote user guides, tested edge cases, ensured code  comments and report clarity. |
| **Aaradhya**  **Bhattarai** | UI/Report  Designer | Formatted output files (e.g., grades/ECA tables),  contributed to the project report. |

**Personal Reflection** **Challenges Faced**

1. **File Handling**: Managing concurrent reads/writes to .txt files caused occasional data corruption.
2. **User Input Validation**: Ensuring numeric inputs for grades required robust error handling.
3. **Team Coordination**: Merging code via Git led to conflicts; resolved through clearer task delegation.

# Lessons Learned

* + **Collaboration**: Regular stand-up meetings improved task alignment.
  + **Debugging**: Writing test cases early saved time (e.g., validating ECA file formats).
  + **Adaptability**: Learned new libraries (Pandas, Matplotlib) for data analysis.

# Future Improvements

* + Implement **password encryption** (e.g., bcrypt).
  + Migrate to a **database** (SQLite/MySQL) for scalability.
  + Add a **GUI** using Tkinter or Flask for web access.

# Conclusion

This project honed our Python skills while highlighting the importance of teamwork and iterative testing. Each member’s expertise—from Resham’s leadership to Aaditya’s data logic—was instrumental in delivering a functional SMS. While the system meets basic requirements, future iterations could transform it into a production-ready tool for educational institutions.