Prison Management System - Project Report

**Course: Computer Programming (BSCS 1B)  
Instructor: Sir Mohsin Javaid Butt  
Semester: Spring 2025  
Deadline: 25 May 2025**  
**Group Members:**

**M Farrukh Mubeen (01-134251-033)**

**Imran Riaz (01-134251-071)**

# 1. Project Overview

This project simulates a real-world Prison Management System built using C++ fundamentals. It provides a secure, efficient, and structured way to manage prisoner records, allowing authorized users to add, update, delete, and search data. The program demonstrates object-less logic using arrays and functional decomposition.

# 2. Features Implemented

* **Add Prisoner:** Input prisoner details like name, gender, age, marital status, crime, sentence, and visitors.
* **Auto-Generate Unique ID**: Each prisoner is assigned a unique 2-digit ID using rand ().
* **Record Visitors:** Up to 5 visitors per prisoner can be stored.
* **Reduce Sentence:** Authorized users can reduce sentence duration in years.
* **Save to File:** All prisoner data can be exported to prisoners.txt.
* **Search by ID**: Users can look up prisoners using their unique ID.
* **Update Record (Admin Only):** Admins can update all fields of a prisoner's data.
* **Delete Record (Admin Only):** Allows the removal of prisoner data by ID.
* **Login System:** Basic login with roles - admin and staff, with different access levels.
* **Role-Based Access Control**: Only admins can delete or update records.
* **Colored Terminal Output:** Uses ANSI escape sequences to enhance terminal readability (e.g., red for errors, green for success).
* **Data Persistence:** All records are stored in a text file for future retrieval.

# 3. Concepts Applied

* Variables and Data Types
* Conditional Statements (if, else)
* Loops (for, while)
* Static Arrays (1D and 2D)
* Functions (Modular code)
* File I/O (ofstream)
* Random Number Generation (rand (), srand())
* ANSI Escape Sequences (for colored output)
* Basic Authentication (string-based login system)

# 4. Evaluation

|  |  |
| --- | --- |
| Criterion | Evaluation |
| Usability | Simple text-based interface |
| Data Processing | Efficient use of arrays and functions |
| Error Handling | Basic input validation for ID & limits |
| Data Storage | File-based output using ofstream |
| Authentication | Role-based login with password protection |

# 5. Conclusion

The Prison Management System effectively demonstrates the practical application of C++ basics in a real-world scenario. It allowed us to strengthen our understanding of:

* Data handling using arrays
* Functional programming
* File storage mechanisms
* Terminal interaction and UI formatting

Potential improvements include integrating file input, adding a graphical user interface, and supporting dynamic memory or database storage.

# 6. References

**ANSI Colored Output**: Structure code and Implementation guidance was taken from ChatGPT, specifically for using ANSI escape codes to enhance terminal user experience.

**Input Buffer Handling:** The use of Cin. Ignore () before get line() was guided by standard C++ practices and supported by ChatGPT's explanation of common input stream issues.

**Delete Function Logic:** The logic of shifting array data to remove a prisoner was also suggested by ChatGPT, which improved data consistency in static arrays.

# 7. Deliverables

- **Farrukh Mubeen**  
 - GitHub: <https://github.com/MFarrukhMubeen>  
 - LinkedIn: <http://www.linkedin.com/in/mfarrukhmubeen>  
  
- **Imran Riaz**  
 - GitHub: <https://github.com/Muhammad-Imran-Riaz>  
 - LinkedIn: <https://www.linkedin.com/in/muhammad-imran-riaz-bb5302352/>