UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

WORKSHOP 1

REPORT

Name :MUHAMMAD DANISH FARHAN BIN HAIRULRIZAM

Matric Number : B032310702

Program : BITI S1G2

Project Title : PALAPES UTeM FITNESS SYSTEM

Supervisor Name : PROFESOR MADYA TS. DR. SHARIFAH SAKINAH BINTI SYED AHMAD

Evaluator Name : DR NOOR FAZILLA BINTI ABD YUSOF

EXECUTIVE SUMMARY

This project focuses on developing a user management and fitness tracking system for PALAPES UTeM. It aims to streamline the management of cadet, coach, and admin data while providing robust fitness tracking and reporting functionalities. The system is designed to enhance the accuracy and efficiency of data handling and generate valuable fitness insights.

Managing user data and fitness records manually is inefficient and prone to errors. The absence of a centralized system leads to scattered data, making it challenging to maintain data integrity and accessibility. Moreover, the lack of detailed fitness reports limits the ability to generate actionable insights, hindering the improvement of cadets' fitness levels.

The primary objective is to automate the management of cadet, coach, and admin data, enhancing the accuracy and efficiency of data handling. Additionally, the project aims to implement a robust fitness tracking system, create a centralized database for easy data access, and generate detailed fitness reports and analytics to provide actionable insights for fitness improvement.

The project includes user management functionalities such as adding, editing, viewing, and removing details for cadets, coaches, and admins. It will implement secure authentication mechanisms to ensure data protection. Fitness tracking features will cover various metrics like push-ups, sit-ups, and more, distinguishing between official and unofficial fitness records. The system will generate comprehensive fitness reports and feedback

Automating data management and fitness tracking will improve efficiency, save time, and reduce errors. A centralized system will ensure easy and secure access to all data, maintaining data integrity. Detailed fitness reports will provide actionable insights, helping cadets improve their fitness levels. Secure user authentication will enhance data protection, and the system's scalability will allow for future enhancements and added functionalities

TABLE OF CONTENTS

PAGE

CHAPTER 1: INTRODUCTION

1.1 Introduction

The "PALAPES UTeM Fitness System" is a role-based application designed to manage the fitness records of cadets, coaches, and administrators. It emphasizes structured data handling, efficient role management, and real-time database connectivity. By employing modular programming techniques, the system ensures that functionalities such as login, registration, and fitness record management are clear and adaptable. The integration of MySQL for database management further enhances the system's capability to store and retrieve data efficiently.

This project adopts standard coding practices like the use of arrays, structured selection mechanisms, and error handling to ensure reliability and maintainability. The system supports various functionalities for different user roles, such as adding fitness data, generating reports, and managing user details, with the goal of providing a streamlined interface for effective fitness tracking.

1.2 Problem Statement

The fitness assessment and management process for PALAPES cadets face several challenges, including inefficiency in data tracking and role-based access limitations. Traditional methods are prone to human error, limited scalability, and lack of centralized data storage, making it difficult to ensure accurate fitness monitoring and reporting.

Another critical issue lies in the dependency on live database connections. While the current implementation integrates MySQL, any disruption in connectivity can hinder access to vital information. The existing error handling is also insufficient for addressing unexpected runtime issues, limiting the system's robustness.

1.3 Objective (s) of the project

This project embarks on the following objectives:

To develop a structured role-based system that efficiently manages cadets, coaches, and admin user data.
To automate the process of recording, storing, and retrieving fitness information to reduce manual errors.
To implement a scalable and user-friendly interface for seamless interaction and data visualization.

To enhance the system's reliability with robust error-handling mechanisms and dynamic data management structures.

1.4 Scope

1.4.1 Modules to be developed

1.4.1.1 Login Module

This module is developed to authenticate the users before granting access to the system. Admins, coaches, and cadets are required to log in using their username and password. This ensures security and privacy for each user account. In case of incorrect credentials, an error message will be displayed. The login module supports all roles and determines their corresponding permissions to access specific functionalities.

1.4.1.2 Registration Module

This module allows the creation of new user accounts. Cadets, coaches, and admins can register by providing their details, such as name, age, phone number, and password. During registration, validation ensures accurate input. Admins can also create accounts on behalf of others. This module enforces role-specific rules, such as limiting the number of admins or coaches per squad.

1.4.1.3 User Management Module

This module enables admins to add, view, edit, and delete user details for all roles. Coaches and cadets can view and update their profiles but with restricted access to sensitive data. For example, cadets can only edit personal information such as phone number or address, while admins have full control over all user data.

1.4.1.4 Fitness Management Module

This module allows users to manage fitness records.

Admins can add, edit, delete, and view all fitness records, whether official or unofficial. Coaches can manage fitness records for their assigned squad. Cadets can view and update their unofficial fitness records but cannot modify official records.

Fitness data includes metrics such as push-ups, sit-ups, BMI, lunges, squats, and performance in physical tests like the beep test and 2.4km run.

1.4.1.5 Report Module

This module enables admins and coaches to generate and view fitness reports. Reports can include individual cadet performance, squad-based performance, or overall fitness trends. The module supports data visualization and printing options for better analysis. Cadets can view their personalized fitness reports to track progress.

1.4.2 Target User

The PALAPES UTeM Fitness System is designed to cater to three distinct user roles: Administrator, Coach, and Cadet. Each role is assigned specific responsibilities and access levels, ensuring proper data management and role-based permissions.

1.4.3 Administrator

The administrator is the highest-level user responsible for overseeing the system's overall functionality. Admins manage users, fitness records, and report generation while maintaining data integrity and system security.

Module Access

Admins must authenticate themselves to access the system using their username and password.

Registration Module. Admins can create accounts for new users, including cadets and coaches. Admins themselves can be registered by super-admins or pre-configured settings.

User Management Module. Admins have complete control over user data such as add, view, edit, or delete cadet, coach, and other admin accounts.

Fitness Management Module. Admins can add, edit, delete, and view both official and unofficial fitness records for all cadets.

Report Module. Admins can generate and view detailed reports for individual cadet performance, squad performance and overall system-wide fitness trends

Admin Module. Provides access to advanced system functions such as assigning coaches to specific batches and squads and managing overall system configurations.

1.4.4 Coach

Coaches are responsible for monitoring and managing the fitness records of cadets within their assigned squad. They also provide guidance and feedback to cadets based on their fitness performance.

Module Access

Login Module. Coaches must authenticate themselves using their credentials before accessing the system.

User Management Module. Coaches have limited access. They can view cadet details within their assigned squad but cannot modify or delete user information.

Fitness Management Module. Coaches can add, edit, delete, and view fitness records for cadets within their assigned squad. The coach also views only their squad's fitness records and cannot access records of other squads.

Report Module. Coaches can generate squad-based fitness reports, highlighting the overall performance of their assigned group. The coach view trends and individual progress for better performance assessment.

Coach Module. Coaches allow manage their details and interact with cadets under their supervision.

1.4.5 Cadet

Cadets are the primary users whose fitness data is recorded and tracked. They use the system to view their fitness progress, maintain unofficial fitness records, and generate personal reports.

Module Access

Login Module. Cadets log in with their username and password to access the system.

User Management Module. Cadets can view and update personal information such as phone number and address and cannot modify their official fitness data or sensitive system information.

Fitness Management Module. Cadets can view their official fitness records but cannot edit them also cadet can add and update their unofficial fitness records for self-tracking purposes.

Report Module. Cadets can view and generate personalized fitness reports, enabling them to track their progress and identify areas for improvement.

Cadet Module. Dedicated functionality for cadets to manage their unofficial records and monitor their fitness journey effectively.

1.5 Project Significance

The PALAPES UTeM Fitness System is a significant innovation aimed at streamlining the management and evaluation of fitness data for cadets, coaches, and administrators within the PALAPES UTeM program.

Improved Fitness Management Efficiency

The system digitizes and centralizes the management of fitness data, reducing manual errors and eliminating redundant paperwork. By automating processes like data entry, report generation, and user management, the system significantly enhances the efficiency of fitness evaluations and administrative tasks.

Enhanced Data Accuracy and Security

With role-based access control, the system ensures data integrity by limiting permissions to appropriate users. For example, admins manage overall data, coaches handle squad-specific fitness records, and cadets can only view or update their personal records. This minimizes unauthorized access and protects sensitive information.

Better Decision-Making through Analytics

The system's reporting module enables coaches and admins to generate comprehensive reports, track performance trends, and identify areas for improvement. This data-driven approach supports better decision-making for training programs, helping to enhance the overall fitness levels of cadets.

Personalized User Experience

Cadets can monitor their fitness progress through customized reports and manage their unofficial fitness data, fostering personal accountability and motivation. Coaches can focus on their squads without being overwhelmed by irrelevant data, improving the quality of training and mentorship.

Scalability for Larger Programs

By using structured modules and scalable data structures, the system can easily adapt to handle more users or expanded functionalities, making it suitable for other similar organizations or institutions.

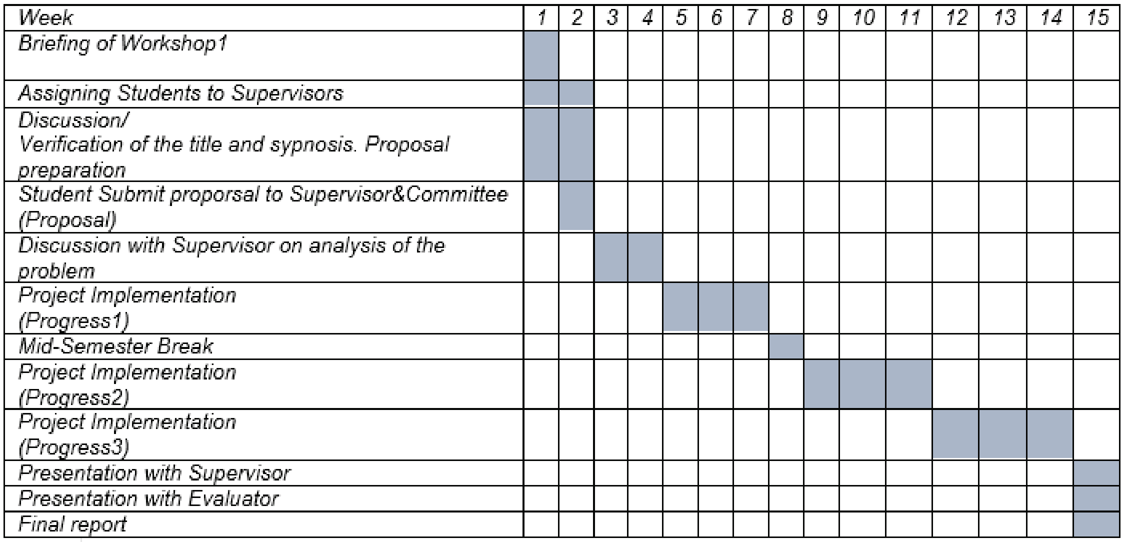
Alignment with Technological Advancement

The project leverages modern software practices and technologies, demonstrating the institution's commitment to embracing digital transformation. It sets an example for integrating technology into traditional systems to achieve better outcomes.

In summary, the PALAPES UTeM Fitness System promotes operational excellence, fosters a culture of fitness and self-improvement, and ensures the effective management of resources and personnel within the program.

1.6 Gantt Chart of Project Activities

Table 1.6.1: Gant Chart



Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Briefing of Workshop1	■														
Assigning Students to Supervisors	■	■													
Discussion/ Verification of the title and sypnosis. Proposal preparation	■	■													
Student Submit proporsal to Supervisor&Committee (Proposal)		■													
Discussion with Supervisor on analysis of the problem			■	■											
Project Implementation (Progress1)					■	■	■								
Mid-Semester Break								■							
Project Implementation (Progress2)									■	■	■				
Project Implementation (Progress3)												■	■	■	
Presentation with Supervisor															■
Presentation with Evaluator															■
Final report															■

Table 1.6.1 illustrate the Gantt Chart of the system project for the whole semester in Workshop 1.

CHAPTER 2: ANALYSIS OF PROBLEM

2.1 Problem Decomposition Description.

Table 2.1.1: Problem decomposition table

Problem	Solution
Inefficient fitness data tracking and management.	Develop a centralized database system for efficient data storage, retrieval, and organization.
Lack of role-based access control for cadets, coaches, and administrators.	Implement a secure, role-based login system tailored to each user type.
Dependency on live database connections causing disruptions during downtime.	Introduce offline functionality with local caching to maintain system usability during outages.
Manual data entry prone to errors and delays.	Automate data input with user-friendly forms and validation to minimize errors.
Difficulty in generating detailed and accurate fitness reports for analysis.	Create automated reporting tools with customizable templates for easy data analysis.

Table 2.1.1 illustrates a list of problems faced in managing fitness record for cadet to keep their fitness record and also to help coach view cadet fitness record.

2.2 Structure Chart

Structure chart shown as below is based on the modules that had in this system with the details.

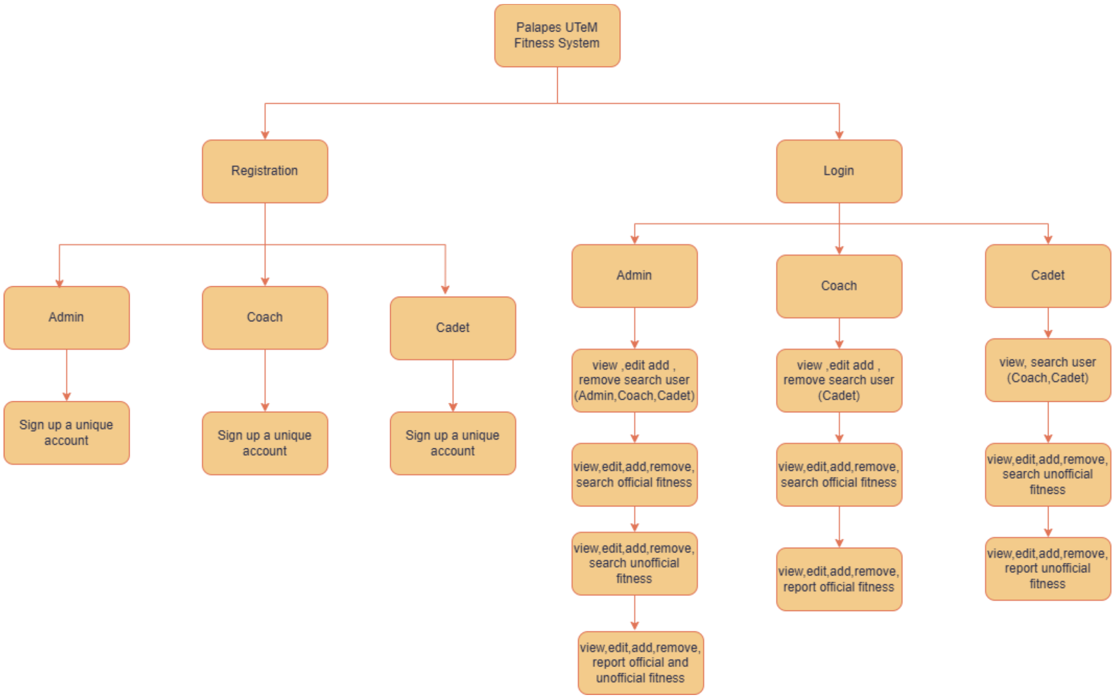

Palapes UTeM Fitness System
Registration
Login
Admin
Coach
Cadet
Sign up a unique account
Sign up a unique account
Sign up a unique account
Admin
Coach
Cadet
view ,edit add , remove search user (Admin,Coach,Cadet)
view,edit,add,remove, search official fitness
view,edit,add,remove, search unofficial fitness
view,edit,add,remove, report official and unofficial fitness
view ,edit add , remove search user (Cadet)
view,edit,add,remove, search official fitness
view,edit,add,remove, report official fitness
view, search user (Coach,Cadet)
view,edit,add,remove, search unofficial fitness
view,edit,add,remove, report unofficial fitness

Figure 2.3.1: Structure Chart of this system

Figure 2.3.1 illustrate the Structure Chart of Palapes UTeM System Fitness.

Chapter 3: DESIGN

3.1 Flowchart

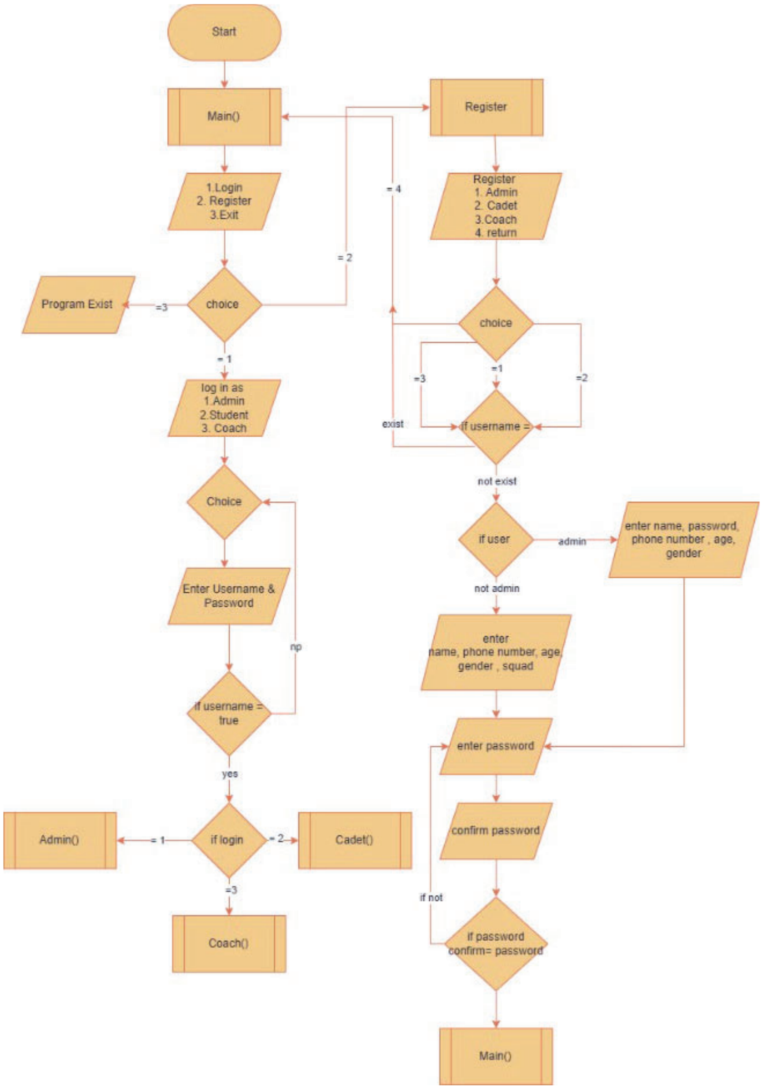


Figure 3.1.1: Login & Registration

Figure 3.1.1 illustrates the flowchart of login and registration page of this system.

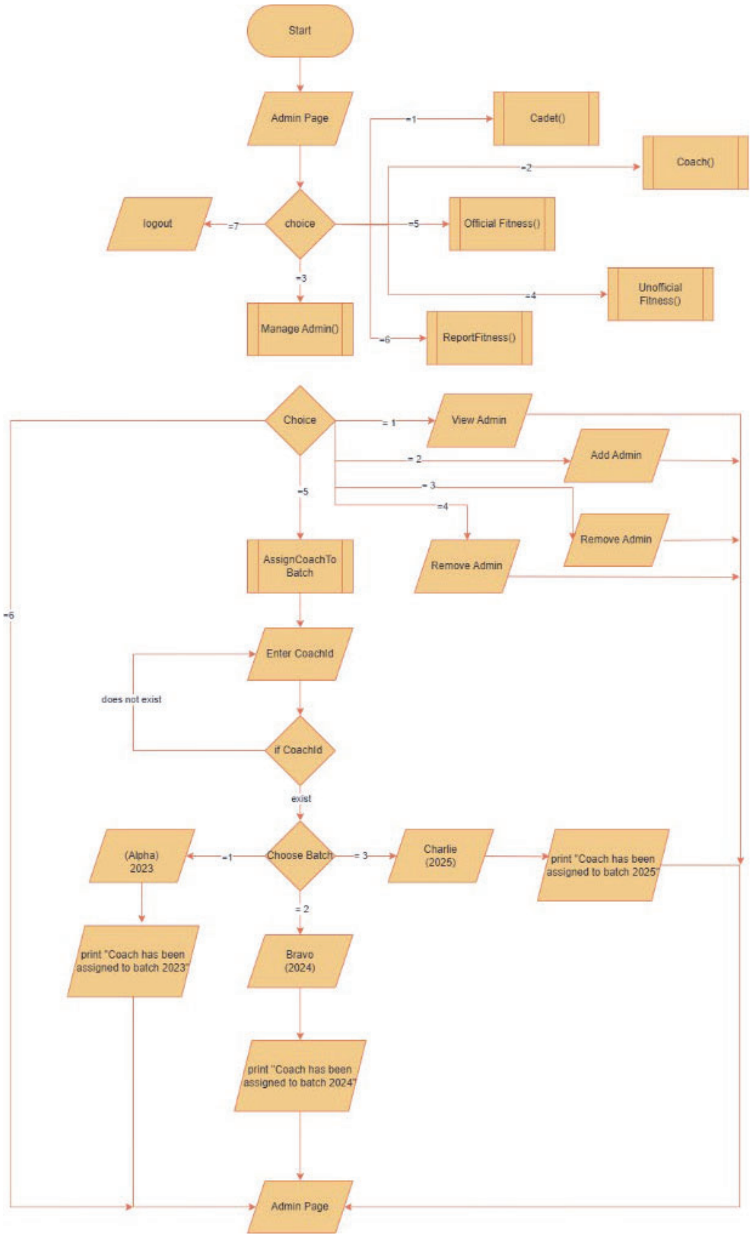


Figure 3.1.2 Admin Page

Figure 3.1.2 illustrates the flowchart of admin page.

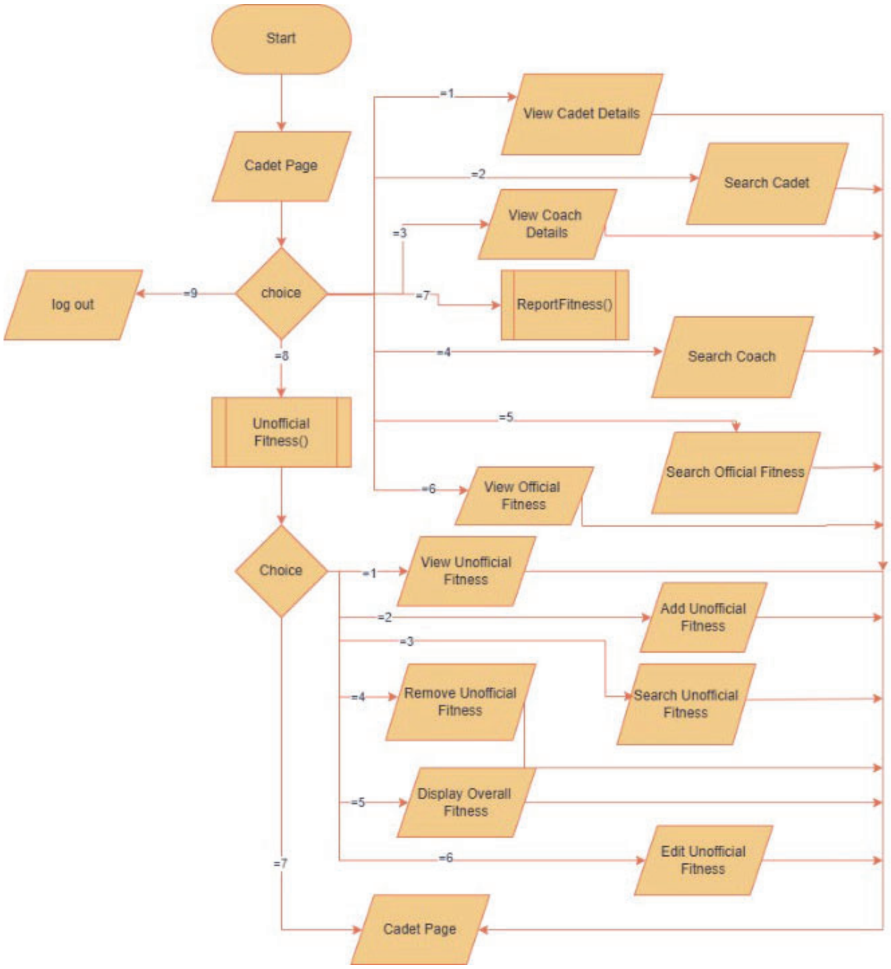


Figure 3.1.3 Cadet Page

Figure 3.1.3 illustrates the flowchart of cadet page.

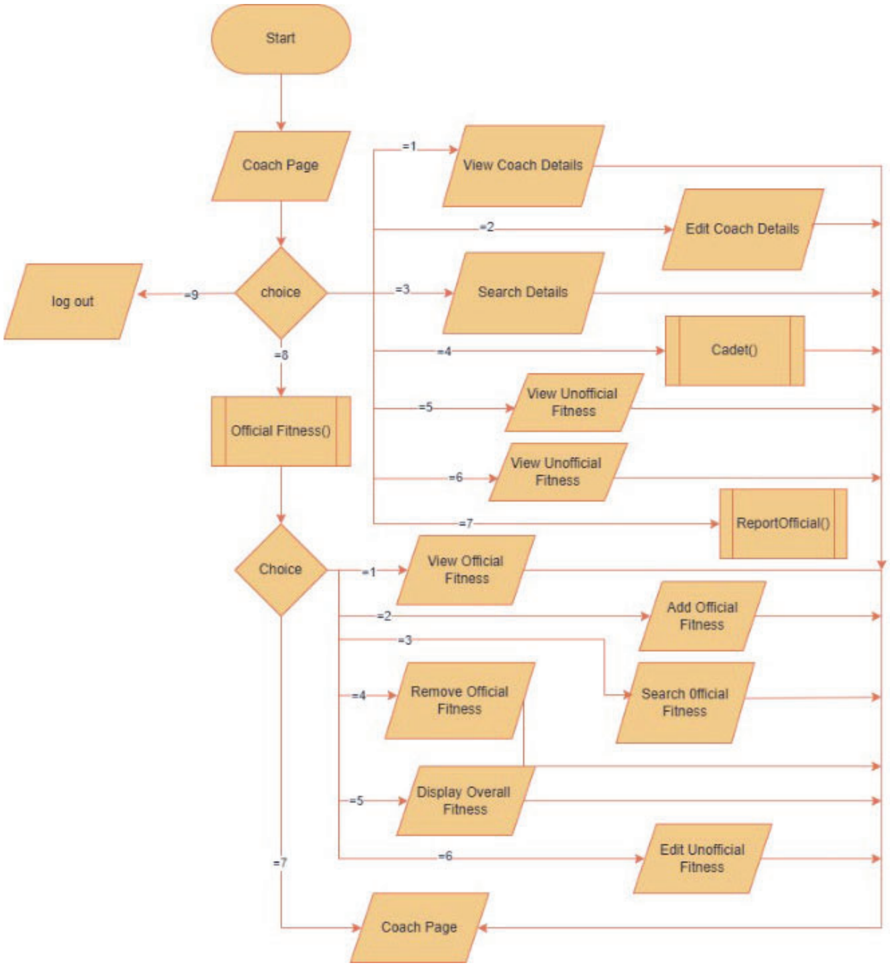


Figure 3.1.4 Coach Page

Figure 3.1.4 illustrates the flowchart of coach page.

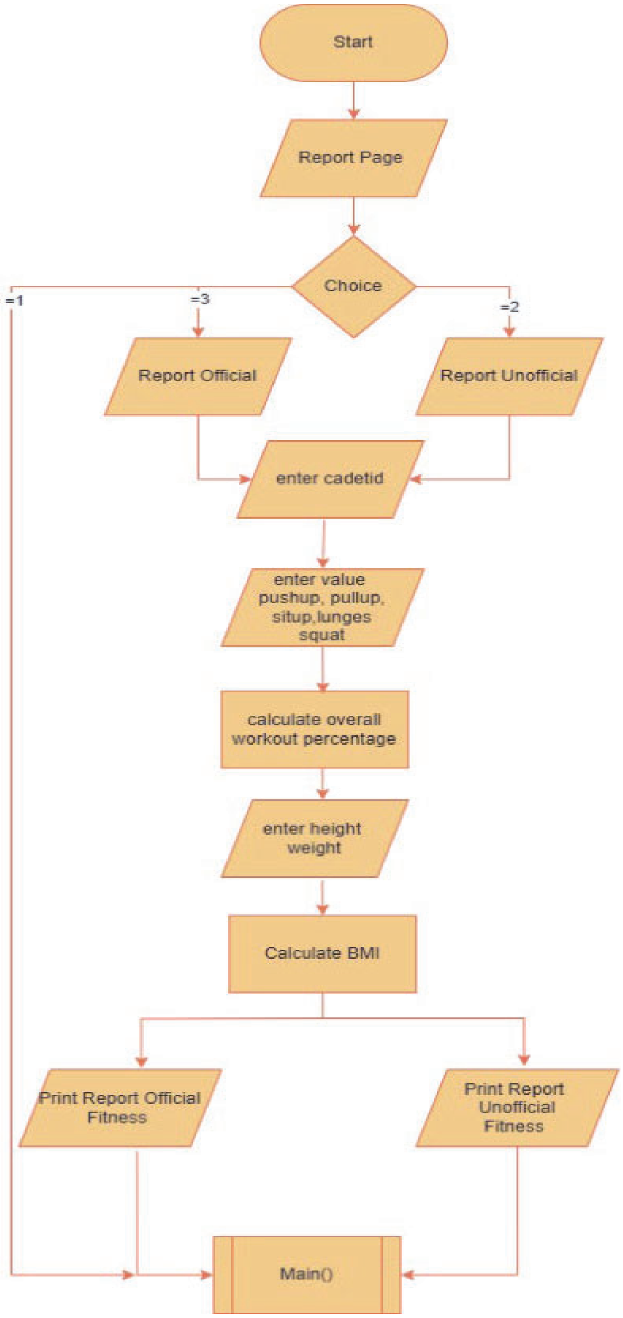


Figure 3.1.5 Report Page

Figure 3.1.5 illustrates the flowchart of report page.

3.2 ERD

This is Entity Relationship Diagram (ERD) of Palapes UTeM Fitness System

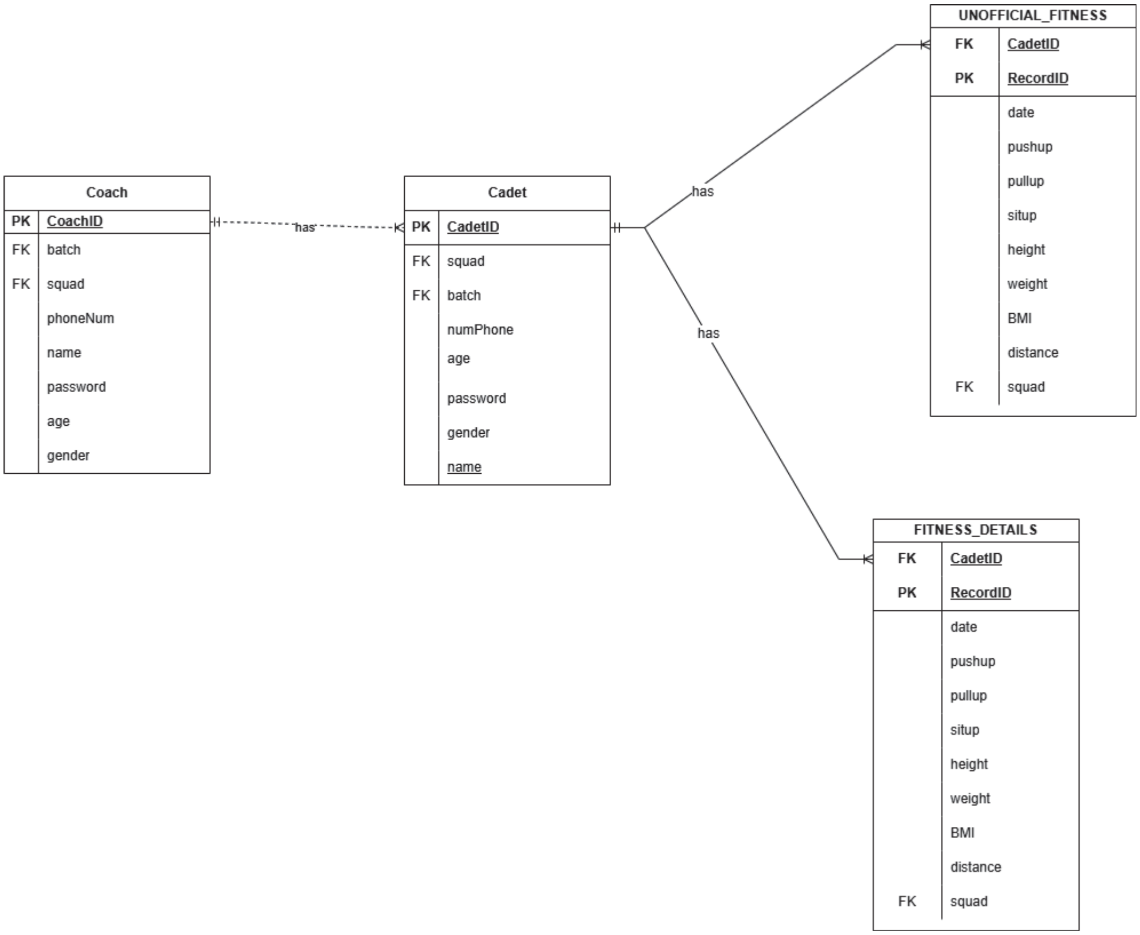


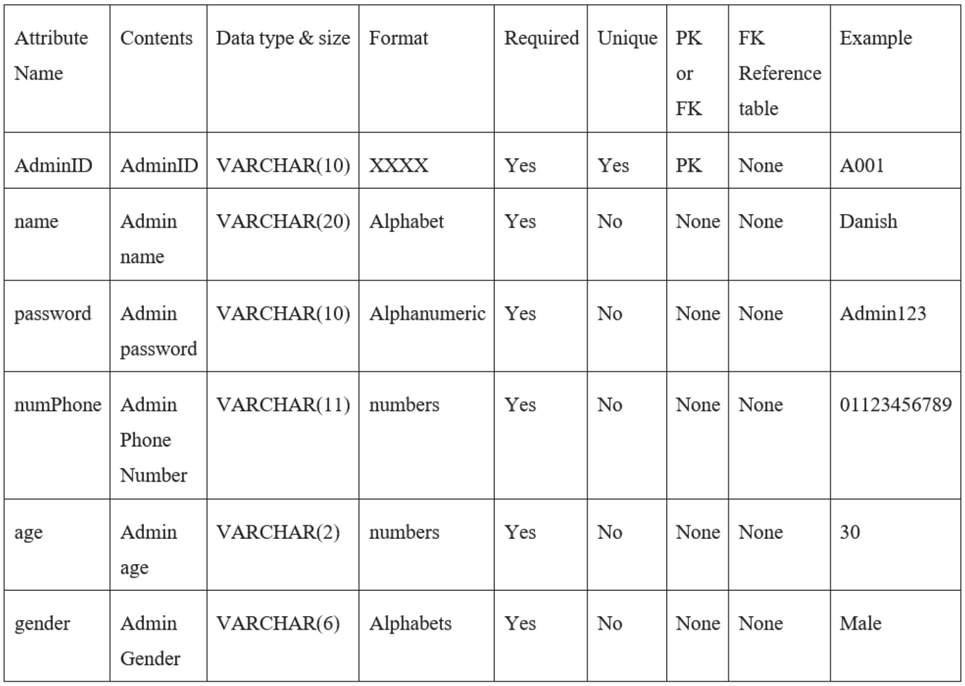
Figure 3.2.1: ERD of Fitness UTeM

Figure 3.2.1 illustrates the Entity Relation Diagram for the Fitness System

3.3 Data Dictionary

The data dictionary for all entities based on ERD are shown in these tables.

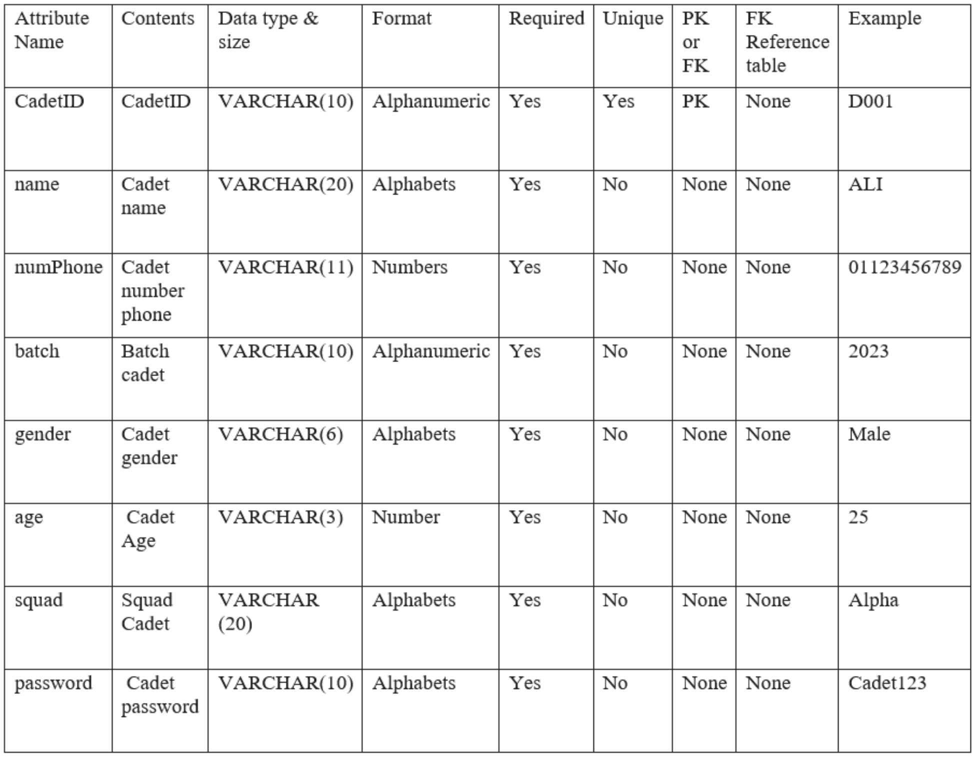
Table 3.3.1: Table ADMIN



Attribute Name	Contents	Data type & size	Format	Required	Unique	PK or FK	FK Reference table	Example
AdminID	AdminID	VARCHAR(10)	XXXX	Yes	Yes	PK	None	A001
name	Admin name	VARCHAR(20)	Alphabet	Yes	No	None	None	Danish
password	Admin password	VARCHAR(10)	Alphanumeric	Yes	No	None	None	Admin123
numPhone	Admin Phone Number	VARCHAR(11)	numbers	Yes	No	None	None	01123456789
age	Admin age	VARCHAR(2)	numbers	Yes	No	None	None	30
gender	Admin Gender	VARCHAR(6)	Alphabets	Yes	No	None	None	Male

Table 3.3.1 illustrates the data of database Admin

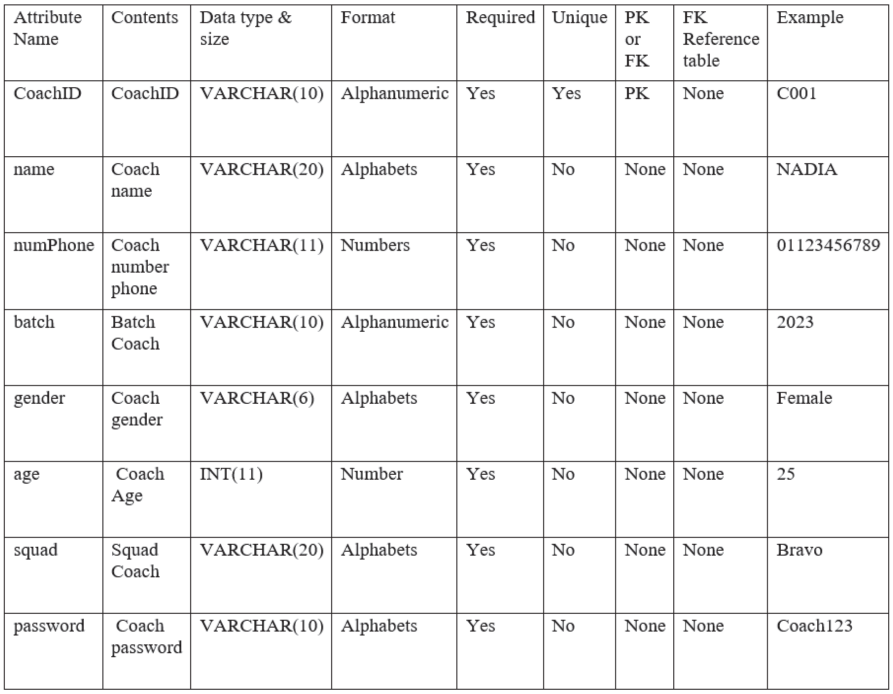
Table 3.3.2 : Table CADET



Attribute Name	Contents	Data type & size	Format	Required	Unique	PK or FK	FK Reference table	Example
CadetID	CadetID	VARCHAR(10)	Alphanumeric	Yes	Yes	PK	None	D001
name	Cadet name	VARCHAR(20)	Alphabets	Yes	No	None	None	ALI
numPhone	Cadet number phone	VARCHAR(11)	Numbers	Yes	No	None	None	01123456789
batch	Batch cadet	VARCHAR(10)	Alphanumeric	Yes	No	None	None	2023
gender	Cadet gender	VARCHAR(6)	Alphabets	Yes	No	None	None	Male
age	Cadet Age	VARCHAR(3)	Number	Yes	No	None	None	25
squad	Squad Cadet	VARCHAR (20)	Alphabets	Yes	No	None	None	Alpha
password	Cadet password	VARCHAR(10)	Alphabets	Yes	No	None	None	Cadet123

Table 3.3.2 illustrates the data of database Cadet

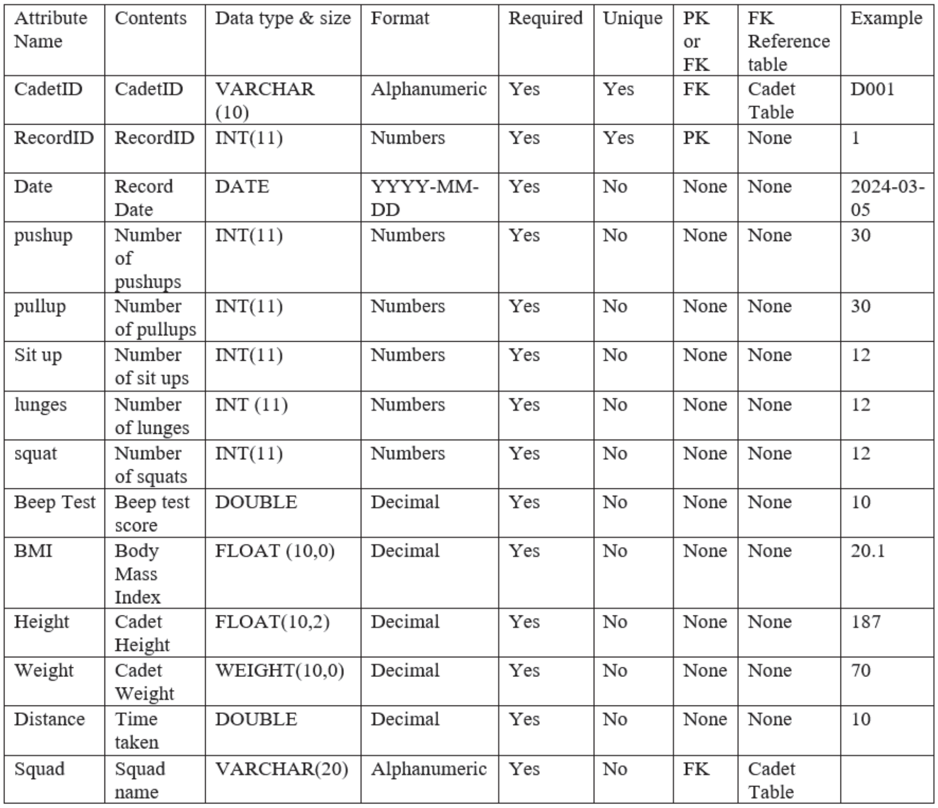
Table 3.3.3: Table COACH



Attribute Name	Contents	Data type & size	Format	Required	Unique	PK or FK	FK Reference table	Example
CoachID	CoachID	VARCHAR(10)	Alphanumeric	Yes	Yes	PK	None	C001
name	Coach name	VARCHAR(20)	Alphabets	Yes	No	None	None	NADIA
numPhone	Coach number phone	VARCHAR(11)	Numbers	Yes	No	None	None	01123456789
batch	Batch Coach	VARCHAR(10)	Alphanumeric	Yes	No	None	None	2023
gender	Coach gender	VARCHAR(6)	Alphabets	Yes	No	None	None	Female
age	Coach Age	INT(11)	Number	Yes	No	None	None	25
squad	Squad Coach	VARCHAR(20)	Alphabets	Yes	No	None	None	Bravo
password	Coach password	VARCHAR(10)	Alphabets	Yes	No	None	None	Coach123

Table 3.3.3 illustrates the data of database Coach

Table 3.3.4: Table OFFICIAL & UNOFFICIAL FITNESS

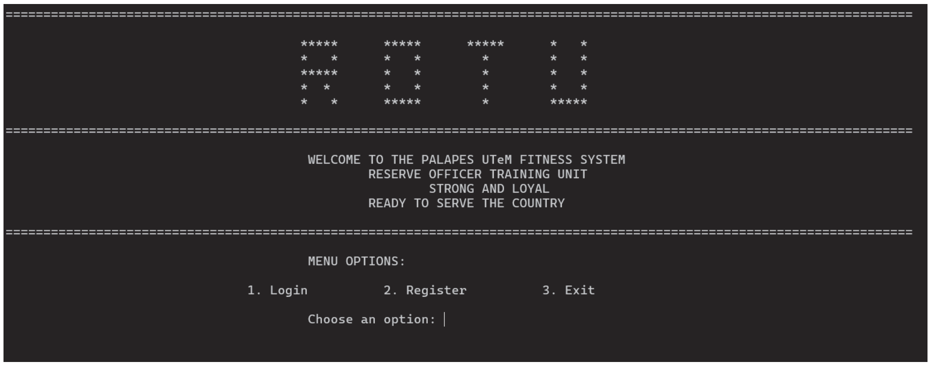


Attribute Name	Contents	Data type & size	Format	Required	Unique	PK or FK	FK Reference table	Example
CadetID	CadetID	VARCHAR (10)	Alphanumeric	Yes	Yes	FK	Cadet Table	D001
RecordID	RecordID	INT(11)	Numbers	Yes	Yes	PK	None	1
Date	Record Date	DATE	YYYY-MM-DD	Yes	No	None	None	2024-03-05
pushup	Number of pushups	INT(11)	Numbers	Yes	No	None	None	30
pullup	Number of pullups	INT(11)	Numbers	Yes	No	None	None	30
Sit up	Number of sit ups	INT(11)	Numbers	Yes	No	None	None	12
lunges	Number of lunges	INT (11)	Numbers	Yes	No	None	None	12
squat	Number of squats	INT(11)	Numbers	Yes	No	None	None	12
Beep Test	Beep test score	DOUBLE	Decimal	Yes	No	None	None	10
BMI	Body Mass Index	FLOAT (10,0)	Decimal	Yes	No	None	None	20.1
Height	Cadet Height	FLOAT(10,2)	Decimal	Yes	No	None	None	187
Weight	Cadet Weight	WEIGHT(10,0)	Decimal	Yes	No	None	None	70
Distance	Time taken	DOUBLE	Decimal	Yes	No	None	None	10
Squad	Squad name	VARCHAR(20)	Alphanumeric	Yes	No	FK	Cadet Table	

Table 3.3.4 illustrates the data of database official and unofficial Fitness record

3.4 Interface Design

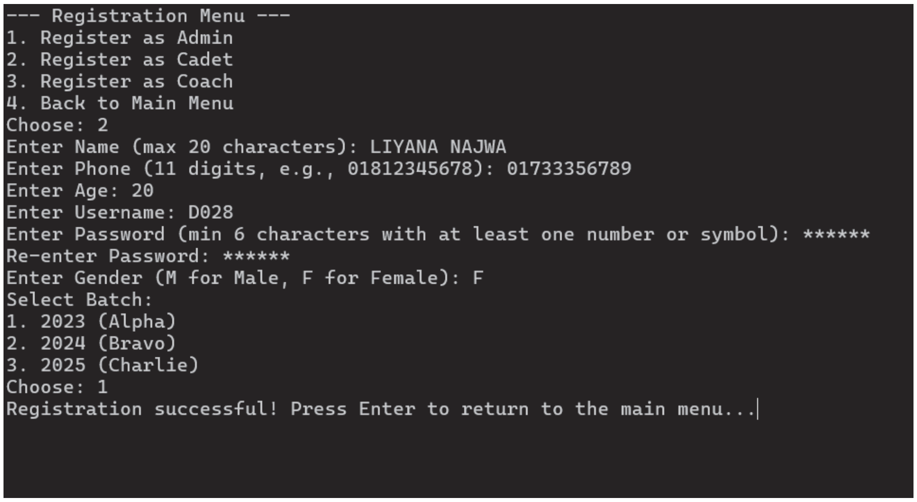
There are listed output in this interface design.



```
========================================================================================================================

                                       *****      *****      *****      *   *
                                       *   *      *   *        *        *   *
                                       *****      *   *        *        *   *
                                       *  *       *   *        *        *   *
                                       *   *      *****        *        *****

========================================================================================================================

                                        WELCOME TO THE PALAPES UTeM FITNESS SYSTEM
                                                RESERVE OFFICER TRAINING UNIT
                                                        STRONG AND LOYAL
                                                READY TO SERVE THE COUNTRY

========================================================================================================================

                                        MENU OPTIONS:

                                1. Login          2. Register          3. Exit

                                        Choose an option: 
```

Figure 3.4.1 Main Page

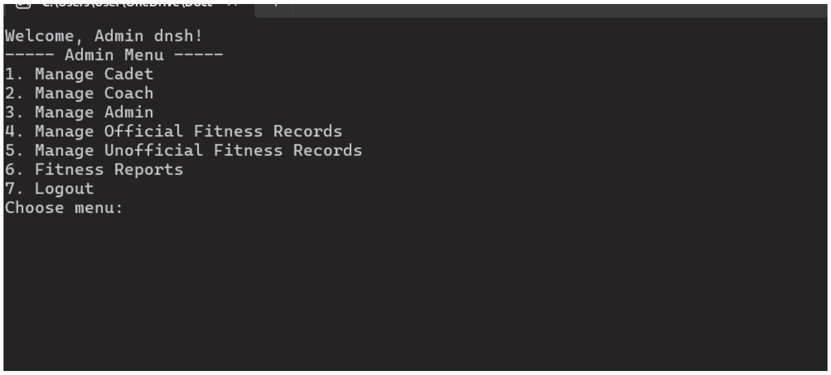
For figure 3.4.1 illustrate the main page where users can choose only 3 options which is 1. Login, 2. Register and 3. Exit



```
--- Registration Menu ---
1. Register as Admin
2. Register as Cadet
3. Register as Coach
4. Back to Main Menu
Choose: 2
Enter Name (max 20 characters): LIYANA NAJWA
Enter Phone (11 digits, e.g., 01812345678): 01733356789
Enter Age: 20
Enter Username: D028
Enter Password (min 6 characters with at least one number or symbol): ******
Re-enter Password: ******
Enter Gender (M for Male, F for Female): F
Select Batch:
1. 2023 (Alpha)
2. 2024 (Bravo)
3. 2025 (Charlie)
Choose: 1
Registration successful! Press Enter to return to the main menu...
```

Figure 3.4.2 Registration Page

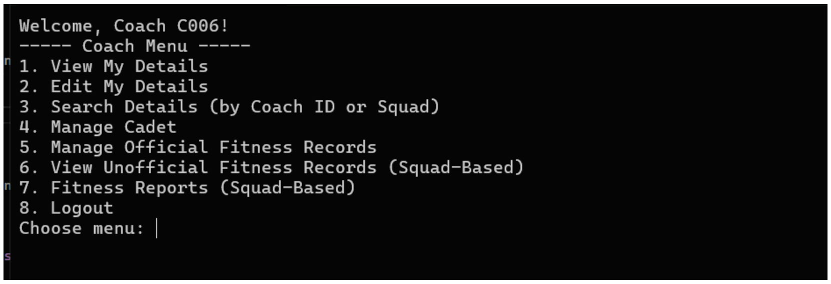
For figure 3.4.2 illustrate the registration page where users can register as admin, as coach or as cadet and the user need to fill the form to get registration successful. The password also is hiding and cadet or coach only select batch.



```
Welcome, Admin dnsh!
----- Admin Menu -----
1. Manage Cadet
2. Manage Coach
3. Manage Admin
4. Manage Official Fitness Records
5. Manage Unofficial Fitness Records
6. Fitness Reports
7. Logout
Choose menu:
```

Figure 3.4.3 Admin Page

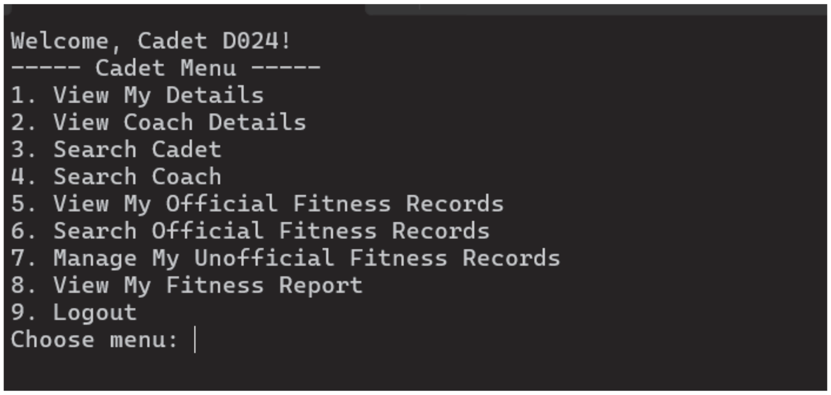
For figure 3.4.3 illustrate admin page where admin can choose the option to manage from option 1 to 7.



```
Welcome, Coach C006!
----- Coach Menu -----
1. View My Details
2. Edit My Details
3. Search Details (by Coach ID or Squad)
4. Manage Cadet
5. Manage Official Fitness Records
6. View Unofficial Fitness Records (Squad-Based)
7. Fitness Reports (Squad-Based)
8. Logout
Choose menu:
```

Figure 3.4.4 Coach Page

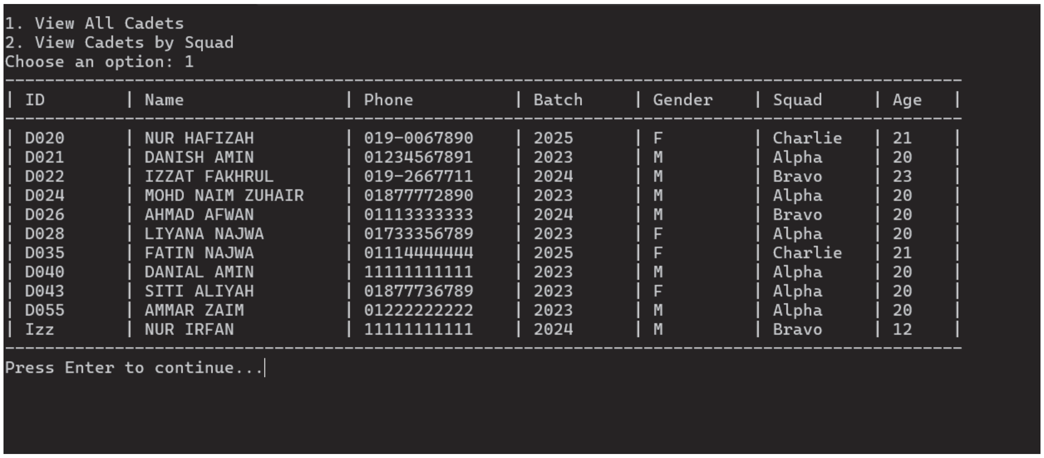
For figure 3.4.4 illustrate coach page where coach can choose the option to manage from option 1 to 8.



```
Welcome, Cadet D024!
----- Cadet Menu -----
1. View My Details
2. View Coach Details
3. Search Cadet
4. Search Coach
5. View My Official Fitness Records
6. Search Official Fitness Records
7. Manage My Unofficial Fitness Records
8. View My Fitness Report
9. Logout
Choose menu: 
```

Figure 3.4.5 Cadet Page

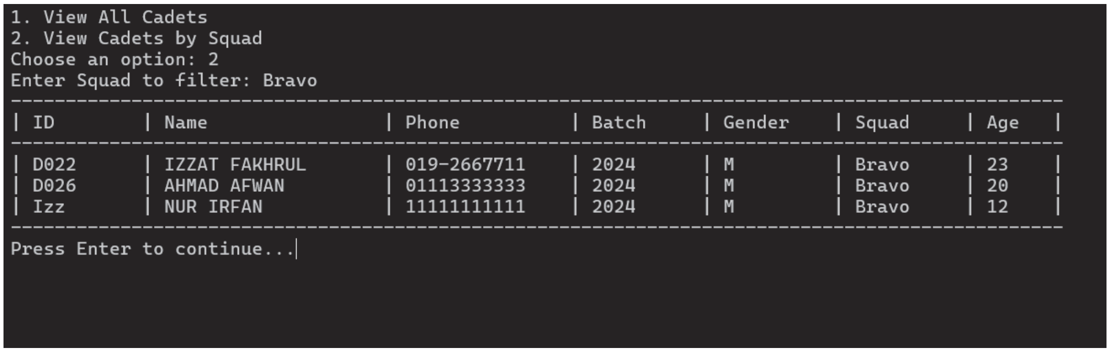
For figure 3.4.5 illustrate cadet page where cadet can choose the option to manage from option 1 to 9.



```
1. View All Cadets
2. View Cadets by Squad
Choose an option: 1
------------------------------------------------------------------------------------------------
| ID        | Name                | Phone          | Batch     | Gender    | Squad     | Age   |
------------------------------------------------------------------------------------------------
| D020      | NUR HAFIZAH         | 019-0067890    | 2025      | F         | Charlie   | 21    |
| D021      | DANISH AMIN         | 01234567891    | 2023      | M         | Alpha     | 20    |
| D022      | IZZAT FAKHRUL       | 019-2667711    | 2024      | M         | Bravo     | 23    |
| D024      | MOHD NAIM ZUHAIR    | 01877772890    | 2023      | M         | Alpha     | 20    |
| D026      | AHMAD AFWAN         | 01113333333    | 2024      | M         | Bravo     | 20    |
| D028      | LIYANA NAJWA        | 01733356789    | 2023      | F         | Alpha     | 20    |
| D035      | FATIN NAJWA         | 01114444444    | 2025      | F         | Charlie   | 21    |
| D040      | DANIAL AMIN         | 11111111111    | 2023      | M         | Alpha     | 20    |
| D043      | SITI ALIYAH         | 01877736789    | 2023      | F         | Alpha     | 20    |
| D055      | AMMAR ZAIM          | 01222222222    | 2023      | M         | Alpha     | 20    |
| Izz       | NUR IRFAN           | 11111111111    | 2024      | M         | Bravo     | 12    |
------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.6 View Cadet Details as Admin

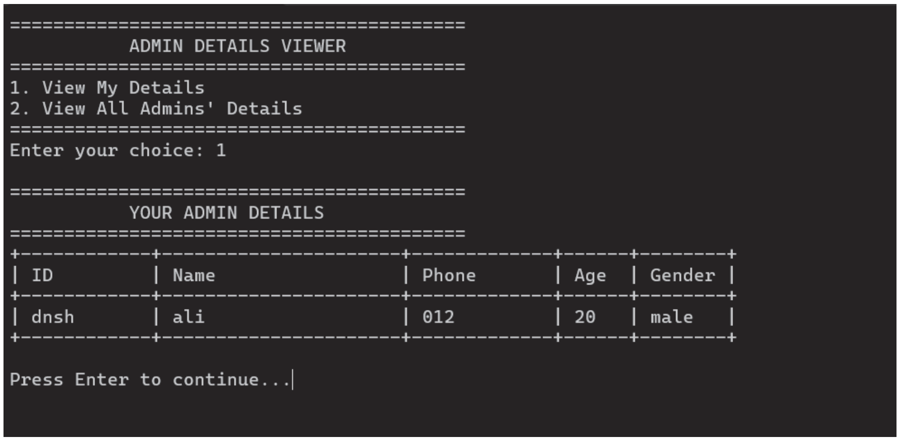
For figure 3.4.6 illustrate the overall cadet details as admin where admin has access to view cadet details.



```
1. View All Cadets
2. View Cadets by Squad
Choose an option: 2
Enter Squad to filter: Bravo
------------------------------------------------------------------------------------------------
| ID         | Name                 | Phone          | Batch      | Gender     | Squad      | Age   |
------------------------------------------------------------------------------------------------
| D022       | IZZAT FAKHRUL        | 019-2667711    | 2024       | M          | Bravo      | 23    |
| D026       | AHMAD AFWAN          | 01113333333    | 2024       | M          | Bravo      | 20    |
| Izz        | NUR IRFAN            | 11111111111    | 2024       | M          | Bravo      | 12    |
------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.7 View Cadet Details based on Squad as Admin

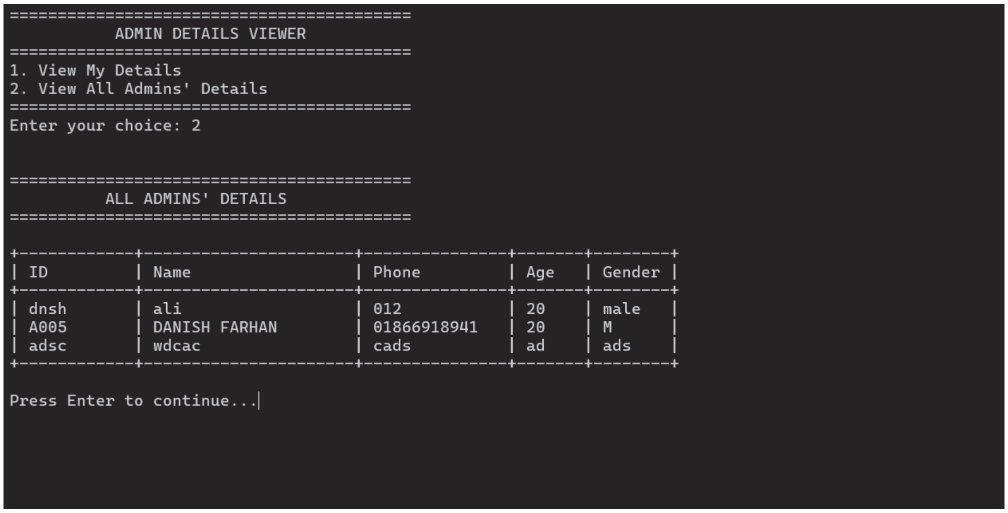
For figure 3.4.7 illustrate the cadet details as admin where admin has access to view cadet details based on squad.



```
==========================================
           ADMIN DETAILS VIEWER
==========================================
1. View My Details
2. View All Admins' Details
==========================================
Enter your choice: 1

==========================================
           YOUR ADMIN DETAILS
==========================================
+------------+----------------------+-------------+------+--------+
| ID         | Name                 | Phone       | Age  | Gender |
+------------+----------------------+-------------+------+--------+
| dnsh       | ali                  | 012         | 20   | male   |
+------------+----------------------+-------------+------+--------+

Press Enter to continue...
```

Figure 3.4.8 View Admin as Admin

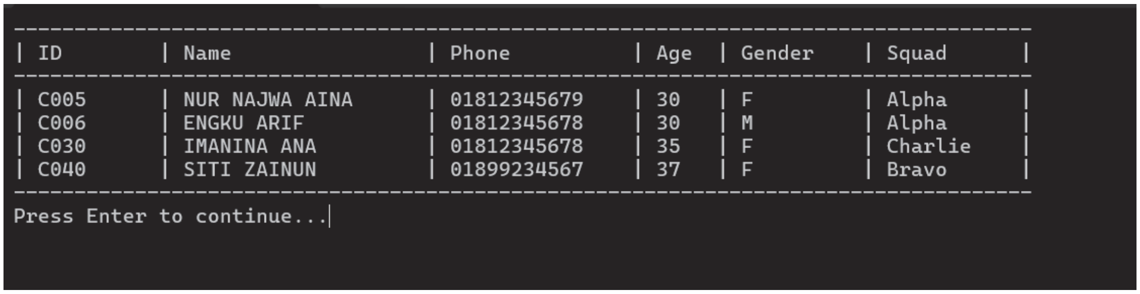
For figure 3.4.8 illustrate the admin details as admin where admin has access to his own details based on current admin login.



```
==========================================
           ADMIN DETAILS VIEWER
==========================================
1. View My Details
2. View All Admins' Details
==========================================
Enter your choice: 2


==========================================
          ALL ADMINS' DETAILS
==========================================

+------------+----------------------+---------------+-------+--------+
| ID         | Name                 | Phone         | Age   | Gender |
+------------+----------------------+---------------+-------+--------+
| dnsh       | ali                  | 012           | 20    | male   |
| A005       | DANISH FARHAN        | 01866918941   | 20    | M      |
| adsc       | wdcac                | cads          | ad    | ads    |
+------------+----------------------+---------------+-------+--------+

Press Enter to continue...
```

Figure 3.4.9 View Admin for all admin details

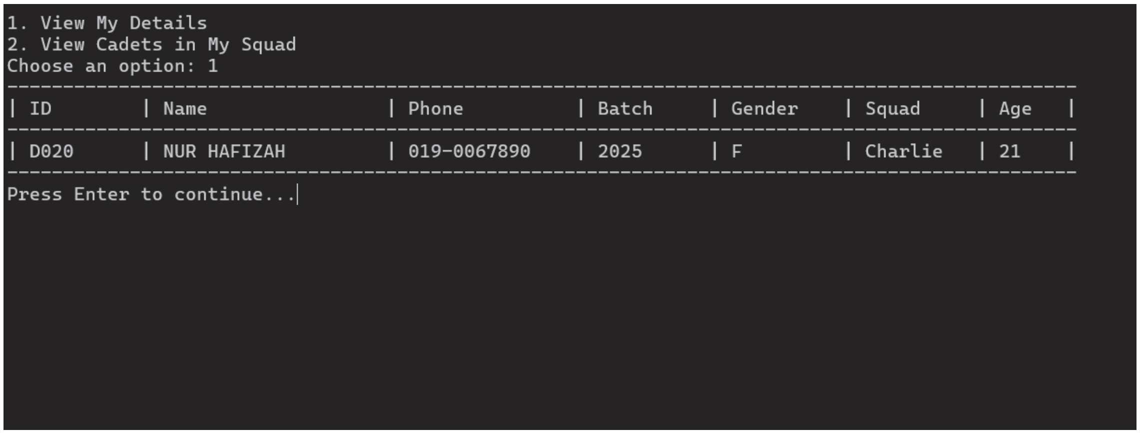
For figure 3.4.9 illustrate the admin details as admin where admin has access to view others admin details.



```
------------------------------------------------------------------------------------
| ID        | Name                | Phone           | Age  | Gender     | Squad       |
------------------------------------------------------------------------------------
| C005      | NUR NAJWA AINA      | 01812345679     | 30   | F          | Alpha       |
| C006      | ENGKU ARIF          | 01812345678     | 30   | M          | Alpha       |
| C030      | IMANINA ANA         | 01812345678     | 35   | F          | Charlie     |
| C040      | SITI ZAINUN         | 01899234567     | 37   | F          | Bravo       |
------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.10 view Coach as Admin

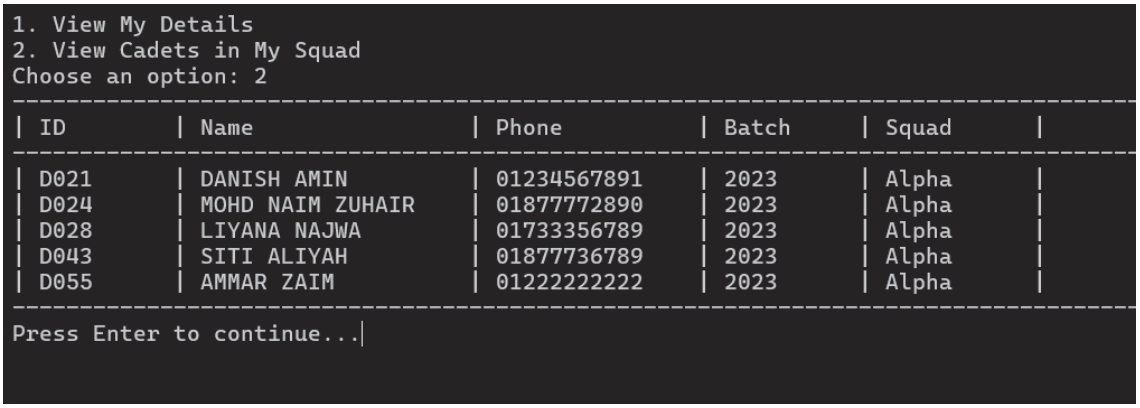
For figure 3.4.10 illustrate the admin has access to view the details of coaches.



```
1. View My Details
2. View Cadets in My Squad
Choose an option: 1
------------------------------------------------------------------------------------------------
| ID        | Name                 | Phone           | Batch      | Gender     | Squad      | Age   |
------------------------------------------------------------------------------------------------
| D020      | NUR HAFIZAH          | 019-0067890     | 2025       | F          | Charlie    | 21    |
------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.11 View Details as Cadet

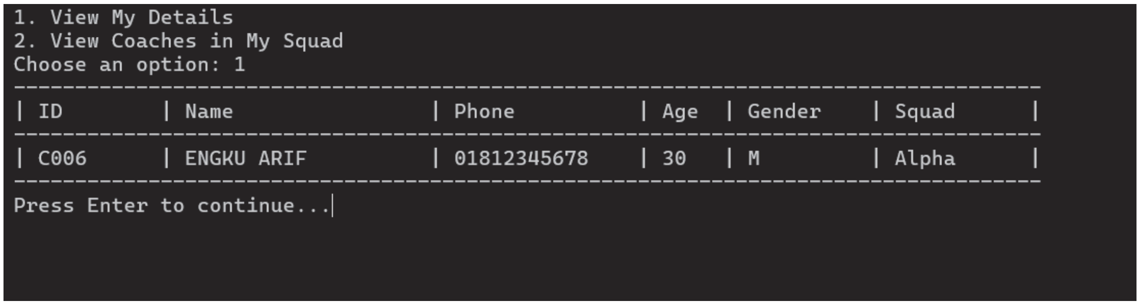
For figure 3.4.11 illustrate the details of cadet based on the current login as cadet.



```
1. View My Details
2. View Cadets in My Squad
Choose an option: 2
------------------------------------------------------------------------------------------
| ID        | Name                 | Phone           | Batch      | Squad      |
------------------------------------------------------------------------------------------
| D021      | DANISH AMIN          | 01234567891     | 2023       | Alpha      |
| D024      | MOHD NAIM ZUHAIR     | 01877772890     | 2023       | Alpha      |
| D028      | LIYANA NAJWA         | 01733356789     | 2023       | Alpha      |
| D043      | SITI ALIYAH          | 01877736789     | 2023       | Alpha      |
| D055      | AMMAR ZAIM           | 01222222222     | 2023       | Alpha      |
------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.12 View Squad Details as Cadet

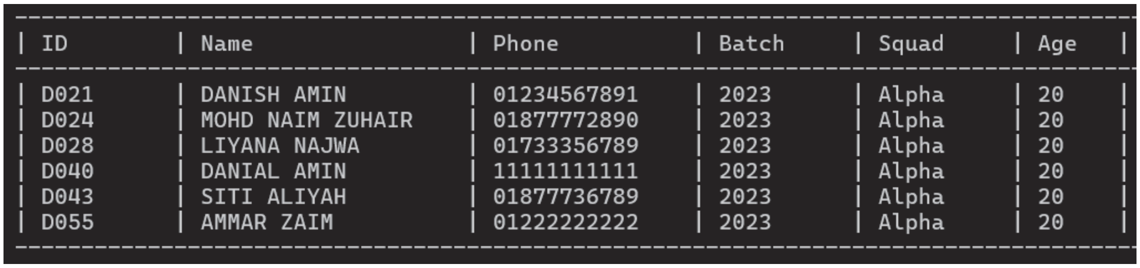
For figure 3.4.12 illustrate the squad details where cadet who login as squad Alpha can view those who same squad with squad Alpha.



```
1. View My Details
2. View Coaches in My Squad
Choose an option: 1
------------------------------------------------------------------------------------
| ID        | Name                 | Phone           | Age  | Gender     | Squad       |
------------------------------------------------------------------------------------
| C006      | ENGKU ARIF           | 01812345678     | 30   | M          | Alpha       |
------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.13 View Details as Coach

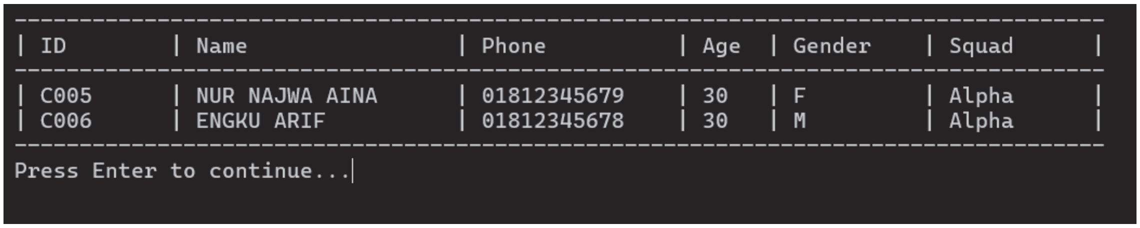
For figure 3.4.13 illustrate the coach details as coach based on current coach login.



```
---------------------------------------------------------------------------------------
| ID        | Name                 | Phone           | Batch      | Squad      | Age    |
---------------------------------------------------------------------------------------
| D021      | DANISH AMIN          | 01234567891     | 2023       | Alpha      | 20     |
| D024      | MOHD NAIM ZUHAIR     | 01877772890     | 2023       | Alpha      | 20     |
| D028      | LIYANA NAJWA         | 01733356789     | 2023       | Alpha      | 20     |
| D040      | DANIAL AMIN          | 11111111111     | 2023       | Alpha      | 20     |
| D043      | SITI ALIYAH          | 01877736789     | 2023       | Alpha      | 20     |
| D055      | AMMAR ZAIM           | 01222222222     | 2023       | Alpha      | 20     |
---------------------------------------------------------------------------------------
```

Figure 3.4.14 View Squad Details as Coach

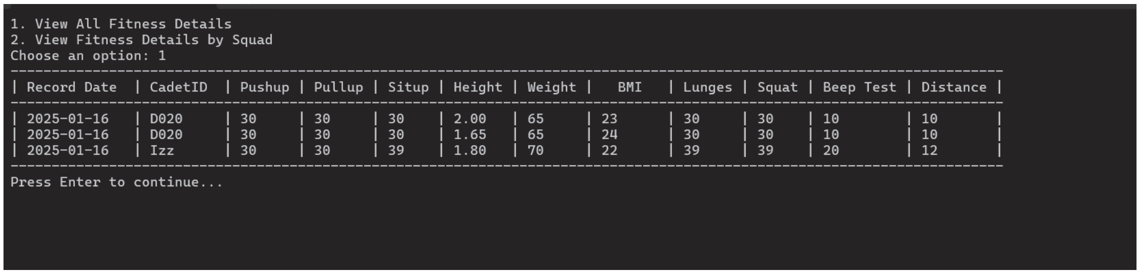
For figure 3.4.14 illustrate the squad details as coach where coach can view his own cadet squad based on the current squad.



```
------------------------------------------------------------------------------------
| ID        | Name                 | Phone           | Age  | Gender     | Squad       |
------------------------------------------------------------------------------------
| C005      | NUR NAJWA AINA       | 01812345679     | 30   | F          | Alpha       |
| C006      | ENGKU ARIF           | 01812345678     | 30   | M          | Alpha       |
------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.15 View Squad Coach as Cadet

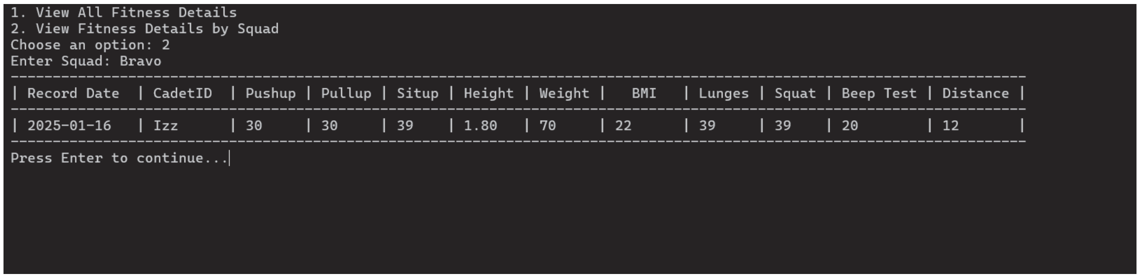
For figure 3.4.15 illustrate the cadet can view his own coaches based on current squad.



```
1. View All Fitness Details
2. View Fitness Details by Squad
Choose an option: 1
-------------------------------------------------------------------------------------------------------------------------
| Record Date  | CadetID  | Pushup | Pullup | Situp | Height | Weight |   BMI   | Lunges | Squat | Beep Test | Distance |
-------------------------------------------------------------------------------------------------------------------------
| 2025-01-16   | D020     | 30     | 30     | 30    | 2.00   | 65     | 23      | 30     | 30    | 10        | 10       |
| 2025-01-16   | D020     | 30     | 30     | 30    | 1.65   | 65     | 24      | 30     | 30    | 10        | 10       |
| 2025-01-16   | Izz      | 30     | 30     | 39    | 1.80   | 70     | 22      | 39     | 39    | 20        | 12       |
-------------------------------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.16 View Official Fitness as Admin

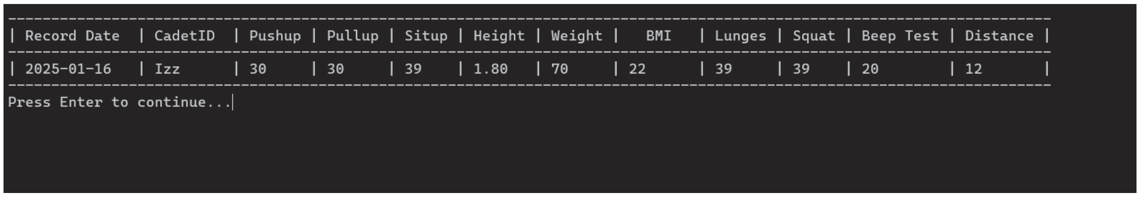
For figure 3.4.16 illustrate the official fitness as admin where admin can view the fitness of cadets.



```
1. View All Fitness Details
2. View Fitness Details by Squad
Choose an option: 2
Enter Squad: Bravo
-------------------------------------------------------------------------------------------------------------------------
| Record Date  | CadetID  | Pushup | Pullup | Situp | Height | Weight |   BMI   | Lunges | Squat | Beep Test | Distance |
-------------------------------------------------------------------------------------------------------------------------
| 2025-01-16   | Izz      | 30     | 30     | 39    | 1.80   | 70     | 22      | 39     | 39    | 20        | 12       |
-------------------------------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.17 View Official Fitness Bravo as Admin

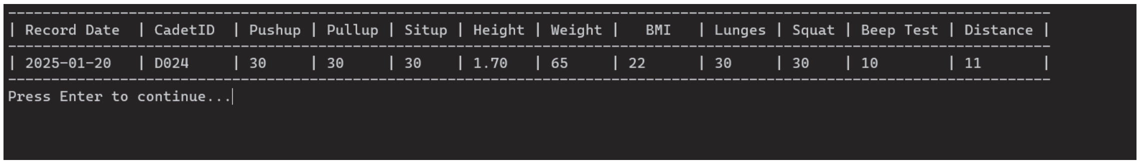
For figure 3.4.17 illustrate the official fitness as admin where admin can view the fitness of cadet based on squad.



```
-------------------------------------------------------------------------------------------------------------------------
| Record Date  | CadetID  | Pushup | Pullup | Situp | Height | Weight |   BMI   | Lunges | Squat | Beep Test | Distance |
-------------------------------------------------------------------------------------------------------------------------
| 2025-01-16   | Izz      | 30     | 30     | 39    | 1.80   | 70     | 22      | 39     | 39    | 20        | 12       |
-------------------------------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.18 View Official Fitness as Cadet

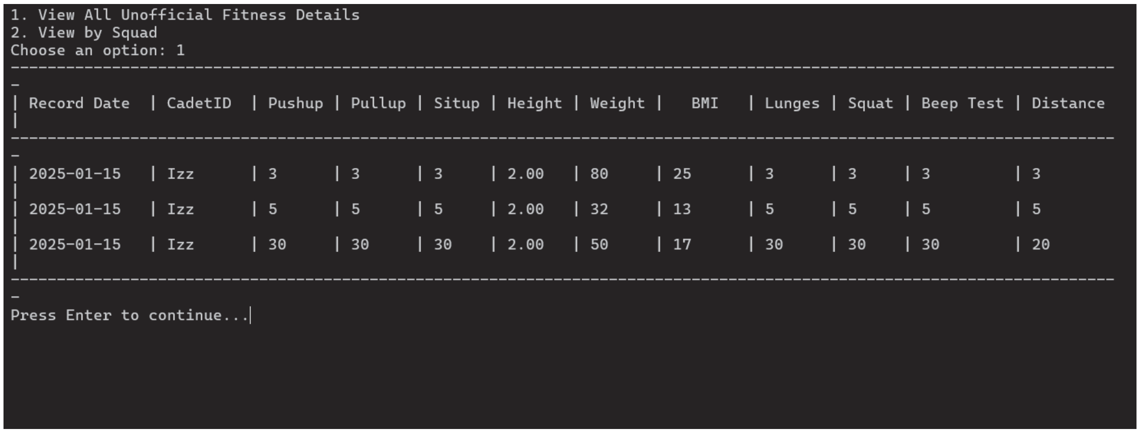
For figure 3.4.18 illustrate the official fitness as cadet where cadet can view his own fitness record.



```
---------------------------------------------------------------------------------------------------------------------------------
| Record Date  | CadetID  | Pushup | Pullup | Situp | Height | Weight |   BMI   | Lunges | Squat | Beep Test | Distance |
---------------------------------------------------------------------------------------------------------------------------------
| 2025-01-20   | D024     | 30     | 30     | 30    | 1.70   | 65     | 22      | 30     | 30    | 10        | 11       |
---------------------------------------------------------------------------------------------------------------------------------
Press Enter to continue...
```

Figure 3.4.19 View Official Fitness as Coach

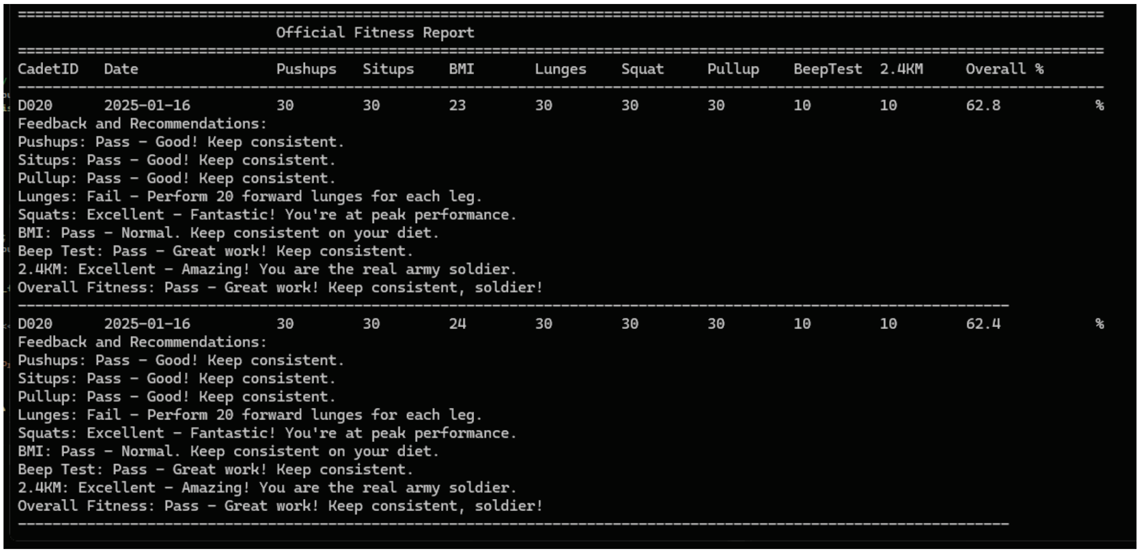
For figure 3.4.19 illustrate the official fitness as coach where coach can view the fitness of cadet based on current squad.



```
1. View All Unofficial Fitness Details
2. View by Squad
Choose an option: 1
--------------------------------------------------------------------------------------------------------------------------------
-
| Record Date  | CadetID  | Pushup | Pullup | Situp | Height | Weight |   BMI   | Lunges | Squat | Beep Test | Distance
|
--------------------------------------------------------------------------------------------------------------------------------
-
| 2025-01-15   | Izz      | 3      | 3      | 3     | 2.00   | 80     | 25      | 3      | 3     | 3         | 3
|
| 2025-01-15   | Izz      | 5      | 5      | 5     | 2.00   | 32     | 13      | 5      | 5     | 5         | 5
|
| 2025-01-15   | Izz      | 30     | 30     | 30    | 2.00   | 50     | 17      | 30     | 30    | 30        | 20
|
--------------------------------------------------------------------------------------------------------------------------------
-
Press Enter to continue...
```

Figure 3.4.20 View Unofficial Fitness as Coach

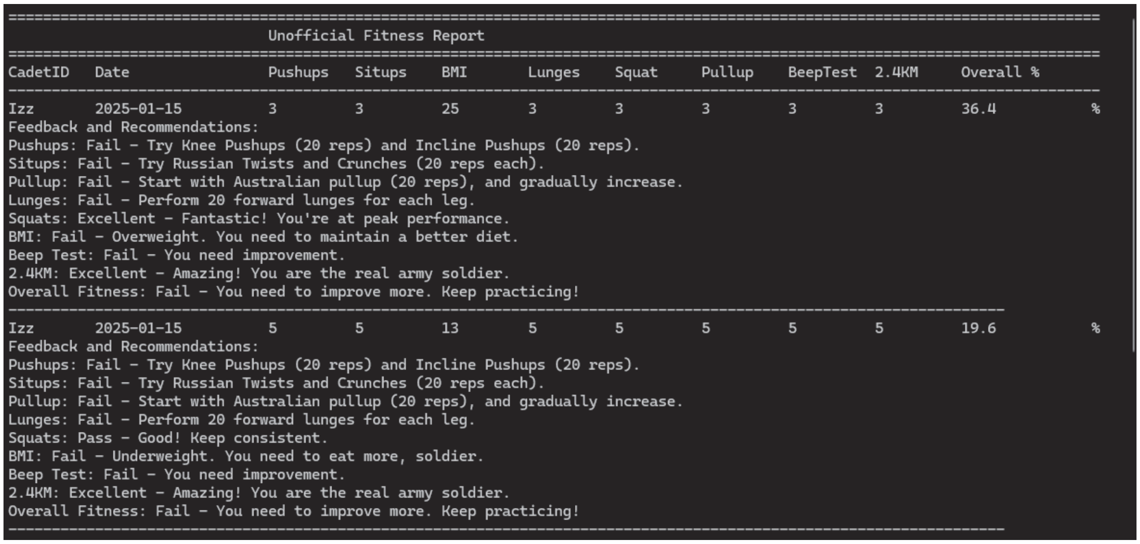
For figure 3.4.20 illustrate the Unofficial fitness as coach where coach can view the fitness of cadet based on current squad.



```
==========================================================================================================================
                              Official Fitness Report
==========================================================================================================================
CadetID   Date                Pushups   Situps    BMI       Lunges    Squat     Pullup    BeepTest  2.4KM     Overall %
--------------------------------------------------------------------------------------------------------------------------
D020      2025-01-16          30        30        23        30        30        30        10        10        62.8          %
Feedback and Recommendations:
Pushups: Pass - Good! Keep consistent.
Situps: Pass - Good! Keep consistent.
Pullup: Pass - Good! Keep consistent.
Lunges: Fail - Perform 20 forward lunges for each leg.
Squats: Excellent - Fantastic! You're at peak performance.
BMI: Pass - Normal. Keep consistent on your diet.
Beep Test: Pass - Great work! Keep consistent.
2.4KM: Excellent - Amazing! You are the real army soldier.
Overall Fitness: Pass - Great work! Keep consistent, soldier!
---------------------------------------------------------------------------------------------------------------
D020      2025-01-16          30        30        24        30        30        30        10        10        62.4          %
Feedback and Recommendations:
Pushups: Pass - Good! Keep consistent.
Situps: Pass - Good! Keep consistent.
Pullup: Pass - Good! Keep consistent.
Lunges: Fail - Perform 20 forward lunges for each leg.
Squats: Excellent - Fantastic! You're at peak performance.
BMI: Pass - Normal. Keep consistent on your diet.
Beep Test: Pass - Great work! Keep consistent.
2.4KM: Excellent - Amazing! You are the real army soldier.
Overall Fitness: Pass - Great work! Keep consistent, soldier!
---------------------------------------------------------------------------------------------------------------
```

Figure 3.4.21 Report Official Fitness

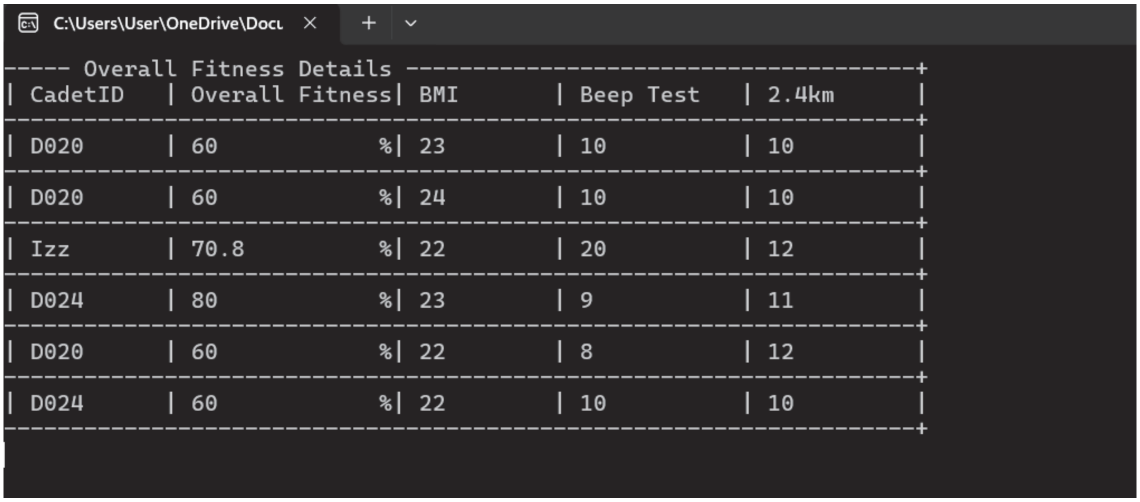
For figure 3.4.21 illustrate the report official fitness where report show overall cadet fitness with feedback and recommendation.



```
==========================================================================================================================
                              Unofficial Fitness Report
==========================================================================================================================
CadetID   Date                Pushups   Situps    BMI       Lunges    Squat     Pullup    BeepTest  2.4KM     Overall %
--------------------------------------------------------------------------------------------------------------------------
Izz       2025-01-15          3         3         25        3         3         3         3         3         36.4          %
Feedback and Recommendations:
Pushups: Fail - Try Knee Pushups (20 reps) and Incline Pushups (20 reps).
Situps: Fail - Try Russian Twists and Crunches (20 reps each).
Pullup: Fail - Start with Australian pullup (20 reps), and gradually increase.
Lunges: Fail - Perform 20 forward lunges for each leg.
Squats: Excellent - Fantastic! You're at peak performance.
BMI: Fail - Overweight. You need to maintain a better diet.
Beep Test: Fail - You need improvement.
2.4KM: Excellent - Amazing! You are the real army soldier.
Overall Fitness: Fail - You need to improve more. Keep practicing!
---------------------------------------------------------------------------------------------------------------
Izz       2025-01-15          5         5         13        5         5         5         5         5         19.6          %
Feedback and Recommendations:
Pushups: Fail - Try Knee Pushups (20 reps) and Incline Pushups (20 reps).
Situps: Fail - Try Russian Twists and Crunches (20 reps each).
Pullup: Fail - Start with Australian pullup (20 reps), and gradually increase.
Lunges: Fail - Perform 20 forward lunges for each leg.
Squats: Pass - Good! Keep consistent.
BMI: Fail - Underweight. You need to eat more, soldier.
Beep Test: Fail - You need improvement.
2.4KM: Excellent - Amazing! You are the real army soldier.
Overall Fitness: Fail - You need to improve more. Keep practicing!
---------------------------------------------------------------------------------------------------------------
```

Figure 3.4.22 Report Unofficial Fitness Report

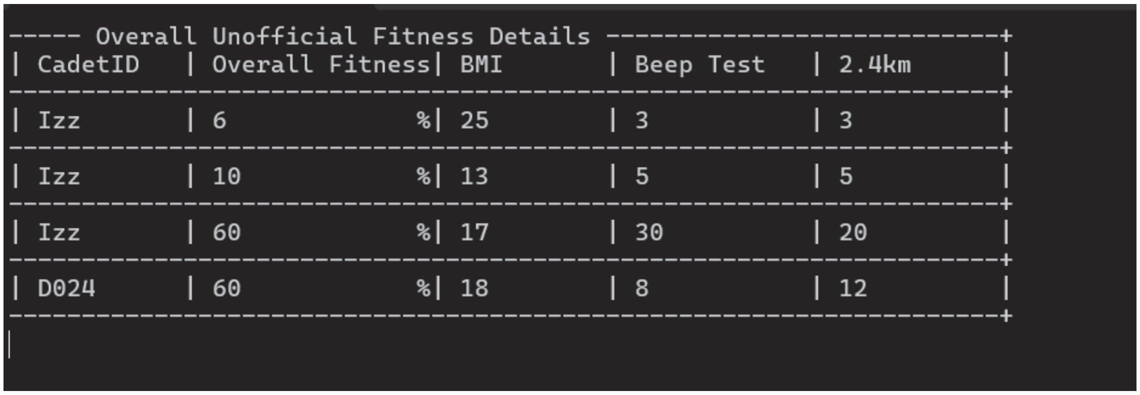
For figure 3.4.22 illustrate the report unofficial fitness where report show overall cadet fitness with feedback and recommendation



```
----- Overall Fitness Details ------------------------------------------+
| CadetID    | Overall Fitness| BMI        | Beep Test    | 2.4km        |
-------------------------------------------------------------------------+
| D020       | 60            %| 23         | 10           | 10           |
-------------------------------------------------------------------------+
| D020       | 60            %| 24         | 10           | 10           |
-------------------------------------------------------------------------+
| Izz        | 70.8          %| 22         | 20           | 12           |
-------------------------------------------------------------------------+
| D024       | 80            %| 23         | 9            | 11           |
-------------------------------------------------------------------------+
| D020       | 60            %| 22         | 8            | 12           |
-------------------------------------------------------------------------+
| D024       | 60            %| 22         | 10           | 10           |
-------------------------------------------------------------------------+
```

Figure   3.4.23 Display Overall Official Fitness Details

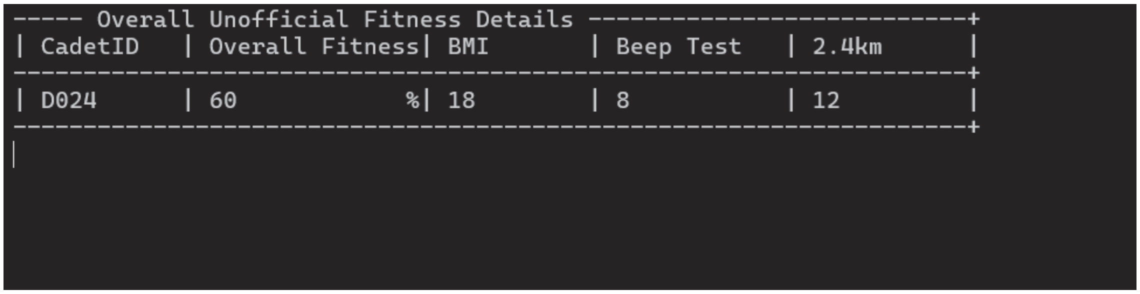
For figure 3.4.23 illustrate that display the overall fitness details as coach where coach can view overall fitness details with percentage.



```
----- Overall Unofficial Fitness Details ---------------------------+
| CadetID    | Overall Fitness| BMI        | Beep Test    | 2.4km        |
---------------------------------------------------------------------+
| Izz        | 6             %| 25         | 3            | 3            |
---------------------------------------------------------------------+
| Izz        | 10            %| 13         | 5            | 5            |
---------------------------------------------------------------------+
| Izz        | 60            %| 17         | 30           | 20           |
---------------------------------------------------------------------+
| D024       | 60            %| 18         | 8            | 12           |
---------------------------------------------------------------------+
```

Figure 3.4.24 Display Overall Fitness Unofficial Details

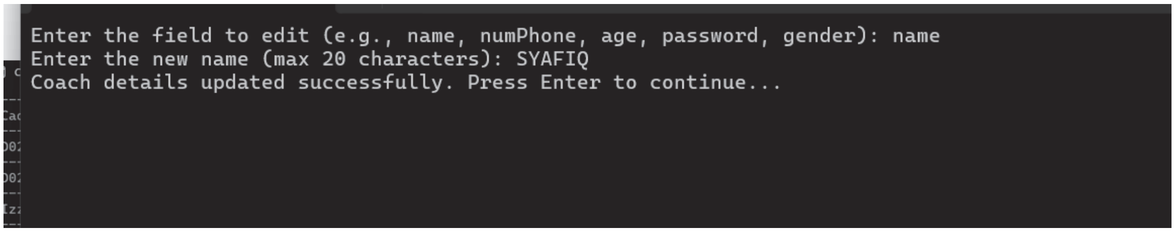
For figure 3.4.24 illustrate that display overall unofficial fitness where report show overall cadet fitness with percentage.



```
----- Overall Unofficial Fitness Details ---------------------------+
| CadetID   | Overall Fitness| BMI      | Beep Test   | 2.4km      |
--------------------------------------------------------------------+
| D024      | 60            %| 18       | 8           | 12         |
--------------------------------------------------------------------+
|
```

Figure 3.4.25 Display Overall Unofficial Fitness as Cadet

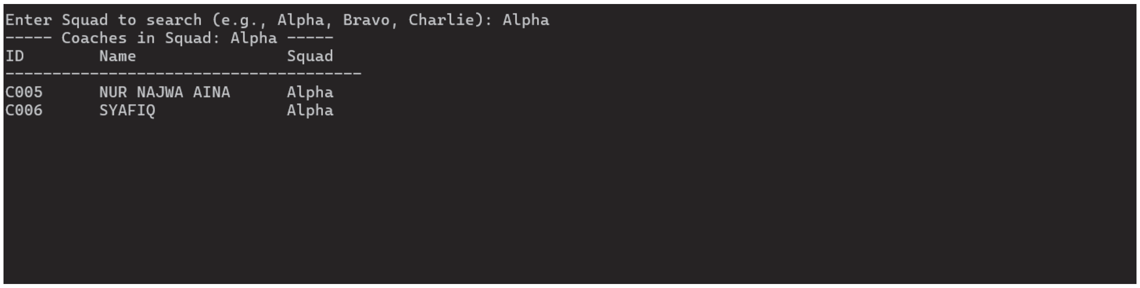
For figure 3.4.25 illustrate the report unofficial fitness where report show overall cadet own fitness with percentage.



```
Enter the field to edit (e.g., name, numPhone, age, password, gender): name
Enter the new name (max 20 characters): SYAFIQ
Coach details updated successfully. Press Enter to continue...
```

Figure 3.4.26 Edit Coach Details

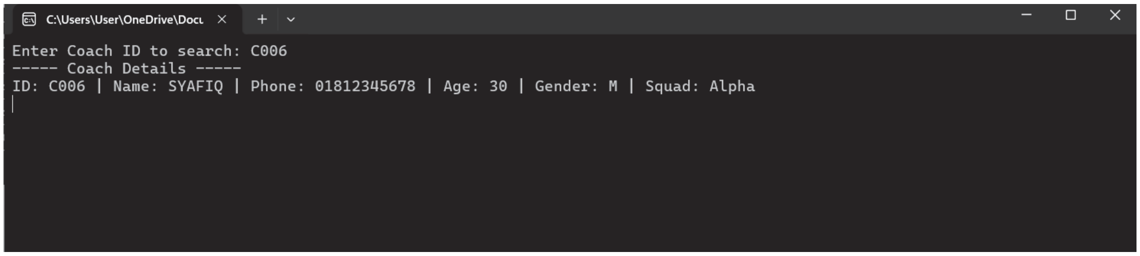
For figure 3.4.26 illustrate edit coach details



```
Enter Squad to search (e.g., Alpha, Bravo, Charlie): Alpha
----- Coaches in Squad: Alpha -----
ID        Name                Squad
--------------------------------------
C005      NUR NAJWA AINA      Alpha
C006      SYAFIQ              Alpha
```

Figure 3.4.27 Search Coach Details by Squad

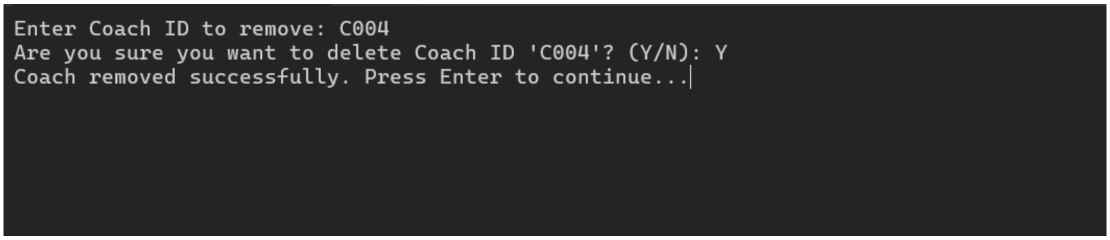
For figure 3.4.27 illustrate search coach details based on search squad.



```
Enter Coach ID to search: C006
----- Coach Details -----
ID: C006 | Name: SYAFIQ | Phone: 01812345678 | Age: 30 | Gender: M | Squad: Alpha
```

Figure 3.4.28 Search Coach Details by ID

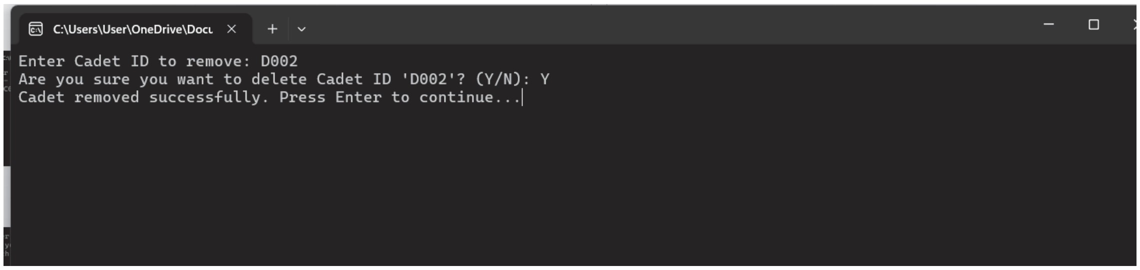
For figure 3.4.28 illustrate search coach details by CoachID



```
Enter Coach ID to remove: C004
Are you sure you want to delete Coach ID 'C004'? (Y/N): Y
Coach removed successfully. Press Enter to continue...
```

Figure 3.4.29 Remove Coach

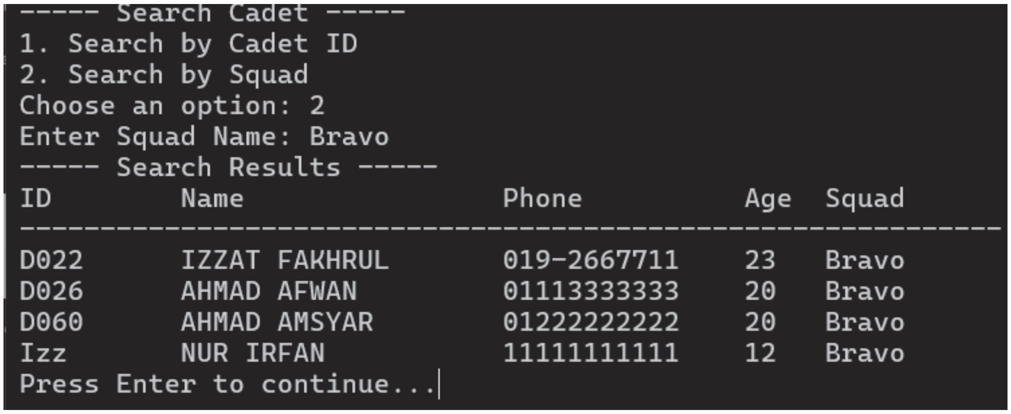
For figure 3.4.29 illustrate to remove coach where admin has access to remove coach



```
Enter Cadet ID to remove: D002
Are you sure you want to delete Cadet ID 'D002'? (Y/N): Y
Cadet removed successfully. Press Enter to continue...
```

Figure 3.4.30 Remove Cadet

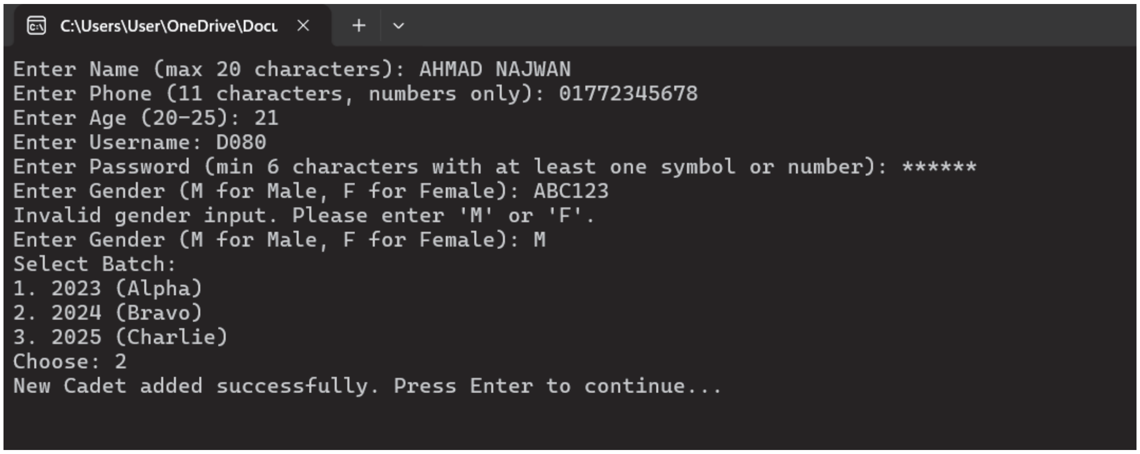
For figure 3.4.30 illustrate to remove cadet where admin and coach has access to remove cadet.



```
----- Search Cadet -----
1. Search by Cadet ID
2. Search by Squad
Choose an option: 2
Enter Squad Name: Bravo
----- Search Results -----
ID        Name                Phone          Age   Squad
-----------------------------------------------------------------
D022      IZZAT FAKHRUL       019-2667711    23    Bravo
D026      AHMAD AFWAN         01113333333    20    Bravo
D060      AHMAD AMSYAR        01222222222    20    Bravo
Izz       NUR IRFAN           11111111111    12    Bravo
Press Enter to continue...
```

Figure 3.4.31 Search Cadet

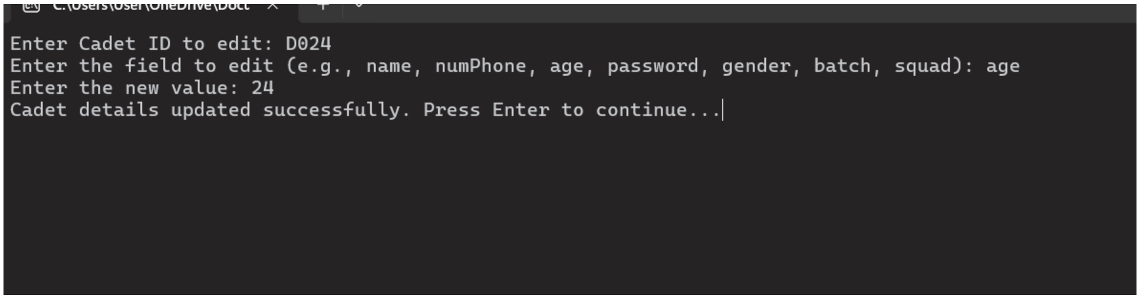
For figure 3.4.31 illustrate the search cadet based on squad name.



```
Enter Name (max 20 characters): AHMAD NAJWAN
Enter Phone (11 characters, numbers only): 01772345678
Enter Age (20-25): 21
Enter Username: D080
Enter Password (min 6 characters with at least one symbol or number): ******
Enter Gender (M for Male, F for Female): ABC123
Invalid gender input. Please enter 'M' or 'F'.
Enter Gender (M for Male, F for Female): M
Select Batch:
1. 2023 (Alpha)
2. 2024 (Bravo)
3. 2025 (Charlie)
Choose: 2
New Cadet added successfully. Press Enter to continue...
```

Figure 3.4.32 Add Cadet

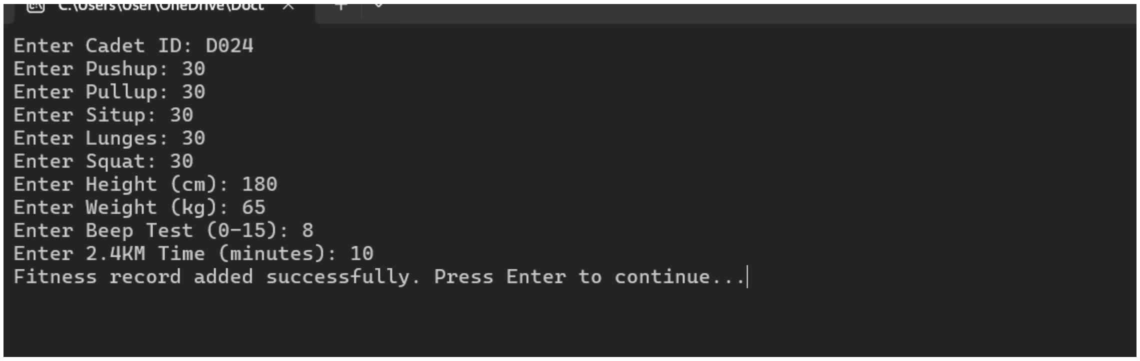
For figure 3.4.32 illustrate where coach and admin have access to add cadet.



```
Enter Cadet ID to edit: D024
Enter the field to edit (e.g., name, numPhone, age, password, gender, batch, squad): age
Enter the new value: 24
Cadet details updated successfully. Press Enter to continue...
```

Figure 3.4.33 Edit Cadet Details

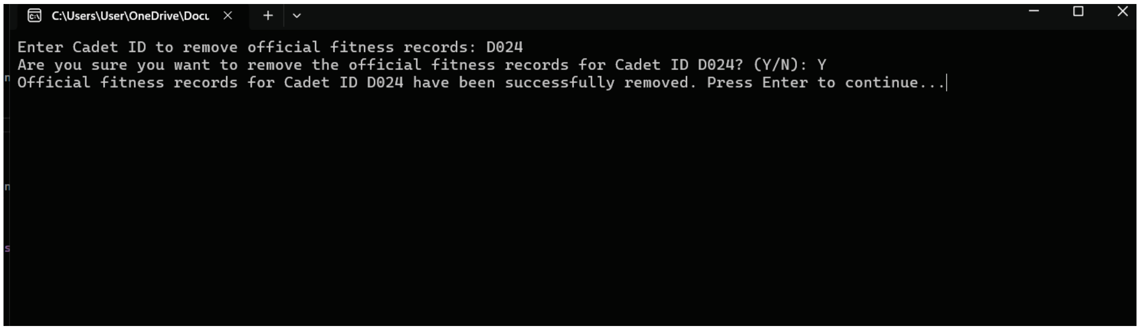
For figure 3.4.33 illustrate edit cadet details where admin and coach have access to edit cadet details.



```
Enter Cadet ID: D024
Enter Pushup: 30
Enter Pullup: 30
Enter Situp: 30
Enter Lunges: 30
Enter Squat: 30
Enter Height (cm): 180
Enter Weight (kg): 65
Enter Beep Test (0-15): 8
Enter 2.4KM Time (minutes): 10
Fitness record added successfully. Press Enter to continue...
```

Figure 3.4.34 Add Fitness Official & Unofficial

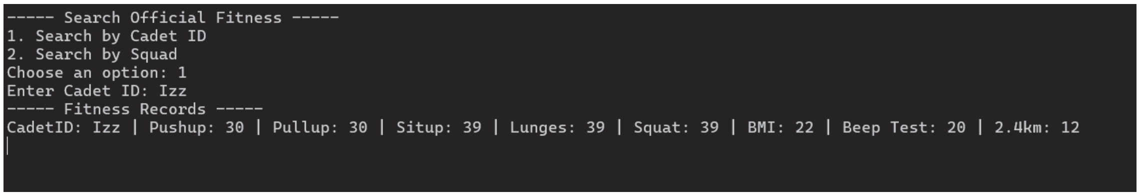
For figure 3.4.34 illustrate add fitness official and unofficial cadet fitness.



```
Enter Cadet ID to remove official fitness records: D024
Are you sure you want to remove the official fitness records for Cadet ID D024? (Y/N): Y
Official fitness records for Cadet ID D024 have been successfully removed. Press Enter to continue...
```

Figure 3.4.35 Remove Official Fitness

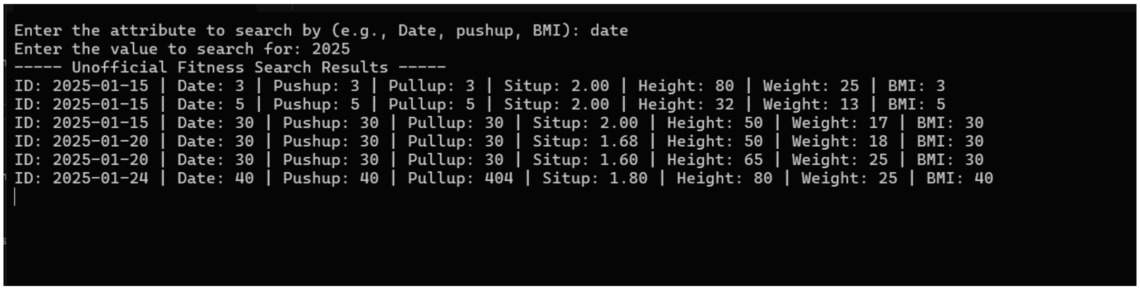
For figure 3.4.35 to remove official fitness record.



```
----- Search Official Fitness -----
1. Search by Cadet ID
2. Search by Squad
Choose an option: 1
Enter Cadet ID: Izz
----- Fitness Records -----
CadetID: Izz | Pushup: 30 | Pullup: 30 | Situp: 39 | Lunges: 39 | Squat: 39 | BMI: 22 | Beep Test: 20 | 2.4km: 12
```

Figure 3.4.36 Search Official Fitness

For figure 3.4.36 illustrate where search official fitness based on squad.



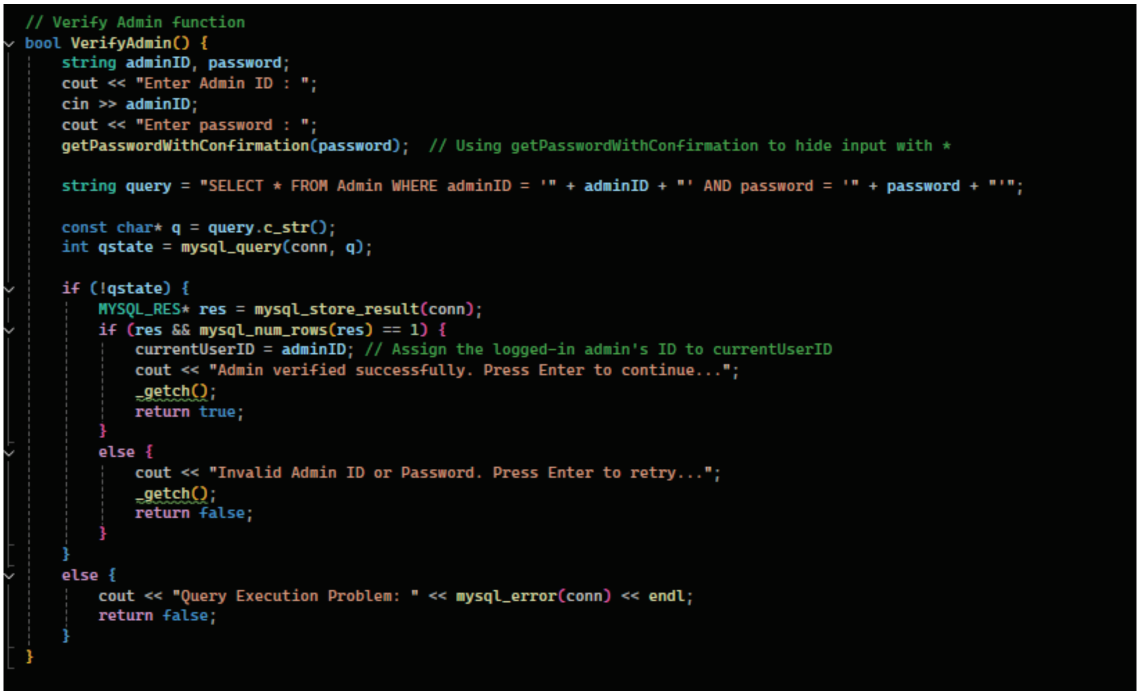
```
Enter the attribute to search by (e.g., Date, pushup, BMI): date
Enter the value to search for: 2025
----- Unofficial Fitness Search Results -----
ID: 2025-01-15 | Date: 3 | Pushup: 3 | Pullup: 3 | Situp: 2.00 | Height: 80 | Weight: 25 | BMI: 3
ID: 2025-01-15 | Date: 5 | Pushup: 5 | Pullup: 5 | Situp: 2.00 | Height: 32 | Weight: 13 | BMI: 5
ID: 2025-01-15 | Date: 30 | Pushup: 30 | Pullup: 30 | Situp: 2.00 | Height: 50 | Weight: 17 | BMI: 30
ID: 2025-01-20 | Date: 30 | Pushup: 30 | Pullup: 30 | Situp: 1.68 | Height: 50 | Weight: 18 | BMI: 30
ID: 2025-01-20 | Date: 30 | Pushup: 30 | Pullup: 30 | Situp: 1.60 | Height: 65 | Weight: 25 | BMI: 30
ID: 2025-01-24 | Date: 40 | Pushup: 40 | Pullup: 404 | Situp: 1.80 | Height: 80 | Weight: 25 | BMI: 40
```

Figure 3.4.37 Search Unofficial Details

For figure 3.4.37 illustrate search unofficial details where cadet can search based on date or any attribute.

CHAPTER 4: IMPLEMENTATION

4.1 Naming Convention



```
// Verify Admin function
bool VerifyAdmin() {
    string adminID, password;
    cout << "Enter Admin ID : ";
    cin >> adminID;
    cout << "Enter password : ";
    getPasswordWithConfirmation(password);  // Using getPasswordWithConfirmation to hide input with *

    string query = "SELECT * FROM Admin WHERE adminID = '" + adminID + "' AND password = '" + password + "'";

    const char* q = query.c_str();
    int qstate = mysql_query(conn, q);

    if (!qstate) {
        MYSQL_RES* res = mysql_store_result(conn);
        if (res && mysql_num_rows(res) == 1) {
            currentUserID = adminID; // Assign the logged-in admin's ID to currentUserID
            cout << "Admin verified successfully. Press Enter to continue...";
            _getch();
            return true;
        }
        else {
            cout << "Invalid Admin ID or Password. Press Enter to retry...";
            _getch();
            return false;
        }
    }
    else {
        cout << "Query Execution Problem: " << mysql_error(conn) << endl;
        return false;
    }
}
```

Figure 4.1.1 Naming Convention

Verify

This prefix clearly indicates that the function's purpose is to perform a verification task. “Verify" suggests that the function checks the validity or correctness.

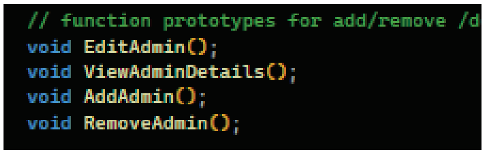
Admin

This suffix specifies what is being verified — the admin user in this context. This aligns with the context of your system, where Admins, Coaches, and Cadets have separate verification processes.

Verb-Noun Pattern

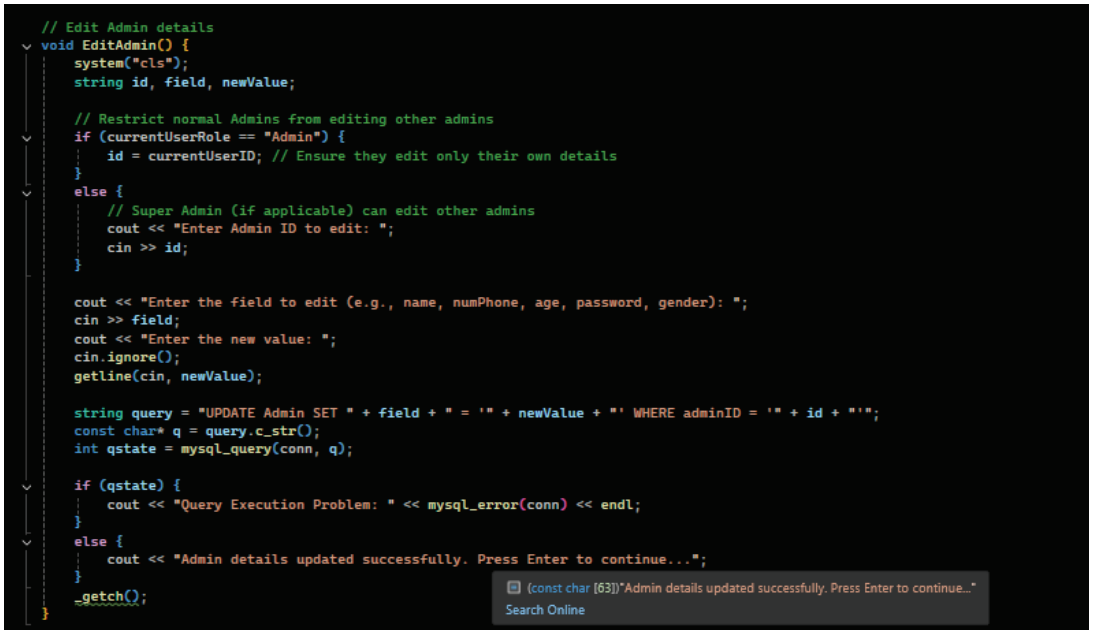
The function name follows a verb-noun pattern (Verify + Admin), which is a widely accepted in naming functions. The verb (Verify) represents the action, and the noun (Admin) represents the entity.

4.2 Function



```
// function prototypes for add/remove /d
void EditAdmin();
void ViewAdminDetails();
void AddAdmin();
void RemoveAdmin();
```

Figure 4.2.1 Function prototype



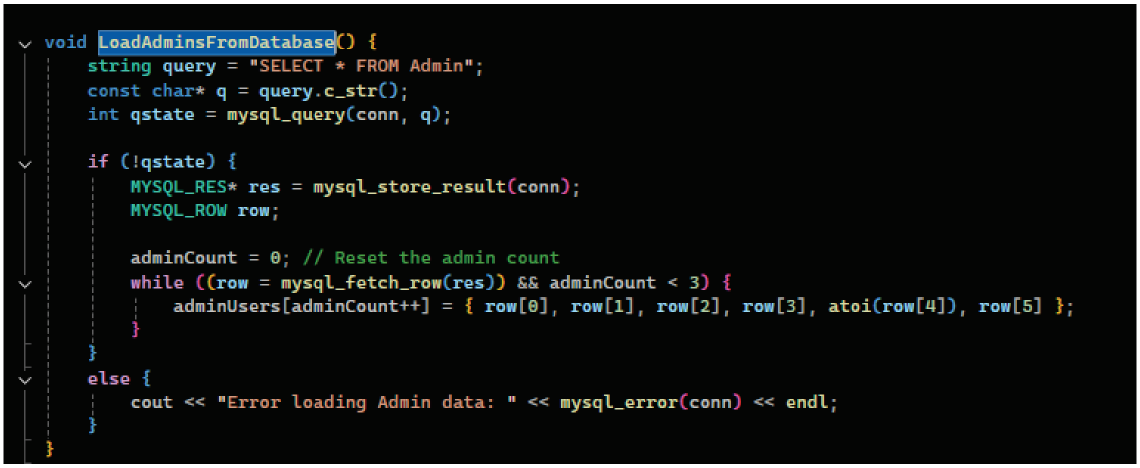
```
// Edit Admin details
void EditAdmin() {
    system("cls");
    string id, field, newValue;

    // Restrict normal Admins from editing other admins
    if (currentUserRole == "Admin") {
        id = currentUserID; // Ensure they edit only their own details
    }
    else {
        // Super Admin (if applicable) can edit other admins
        cout << "Enter Admin ID to edit: ";
        cin >> id;
    }

    cout << "Enter the field to edit (e.g., name, numPhone, age, password, gender): ";
    cin >> field;
    cout << "Enter the new value: ";
    cin.ignore();
    getline(cin, newValue);

    string query = "UPDATE Admin SET " + field + " = '" + newValue + "' WHERE adminID = '" + id + "'";
    const char* q = query.c_str();
    int qstate = mysql_query(conn, q);

    if (qstate) {
        cout << "Query Execution Problem: " << mysql_error(conn) << endl;
    }
    else {
        cout << "Admin details updated successfully. Press Enter to continue...";
    }
    _getch();
}
```

(const char [63])"Admin details updated successfully. Press Enter to continue..."

Figure 4.2.2 Function body

The EditAdmin() function allows the user (an Admin) to modify specific details of an Admin user in the database. It ensures that only valid inputs are accepted and performs necessary validation before updating the database. The function ensures the input for each field adheres to specific constraints, reducing the chances of invalid data being stored in the database. Normal Admins can only edit their own details, ensuring proper access control. The function provides clear error messages when inputs are invalid or the SQL query fails.

4.3 Array

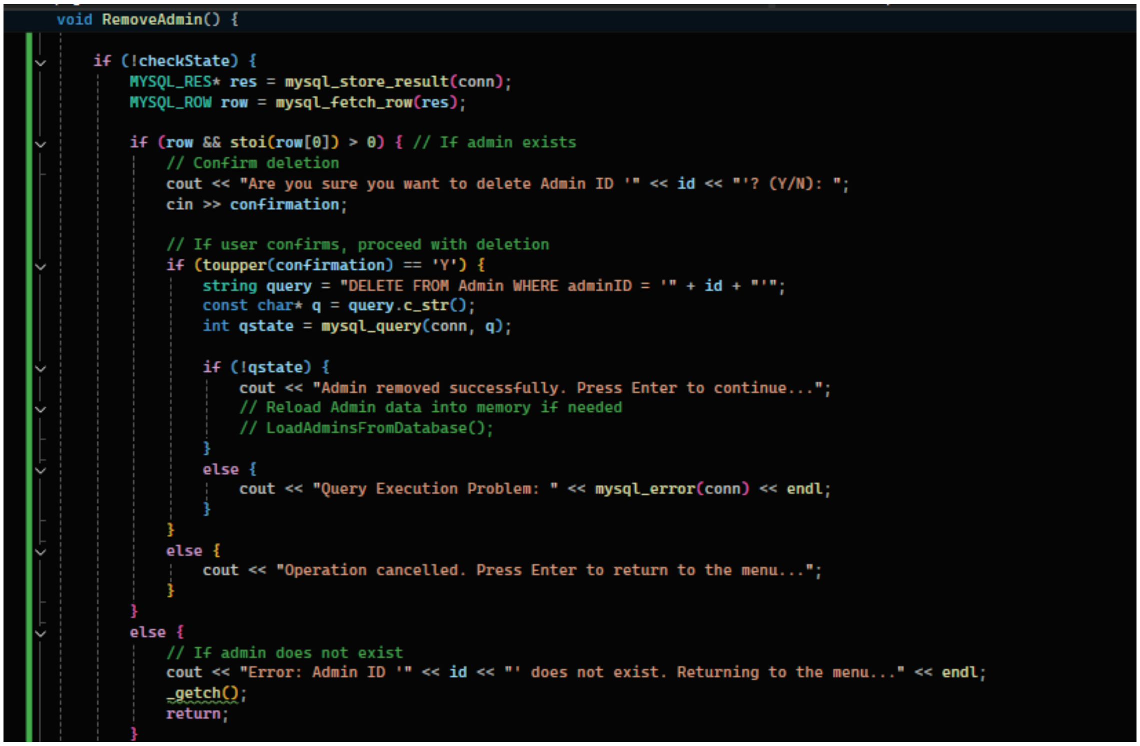


```
void LoadAdminsFromDatabase() {
    string query = "SELECT * FROM Admin";
    const char* q = query.c_str();
    int qstate = mysql_query(conn, q);

    if (!qstate) {
        MYSQL_RES* res = mysql_store_result(conn);
        MYSQL_ROW row;

        adminCount = 0; // Reset the admin count
        while ((row = mysql_fetch_row(res)) && adminCount < 3) {
            adminUsers[adminCount++] = { row[0], row[1], row[2], row[3], atoi(row[4]), row[5] };
        }
    }
    else {
        cout << "Error loading Admin data: " << mysql_error(conn) << endl;
    }
}
```

Figure 4.3.1 Array Function

The LoadAdminsFromDatabase function retrieves admin data from the database and loads it into an array called adminUsers. This array can hold up to three Admin records, each containing information like adminID, name, password, phone, age, and gender. The function processes the database results row by row and stores them sequentially in the array. The purpose of this function is to synchronize the admin data stored in the database with the local memory array (adminUsers). This enables faster access and manipulation of admin data without repeated database queries.

4.4 Selection

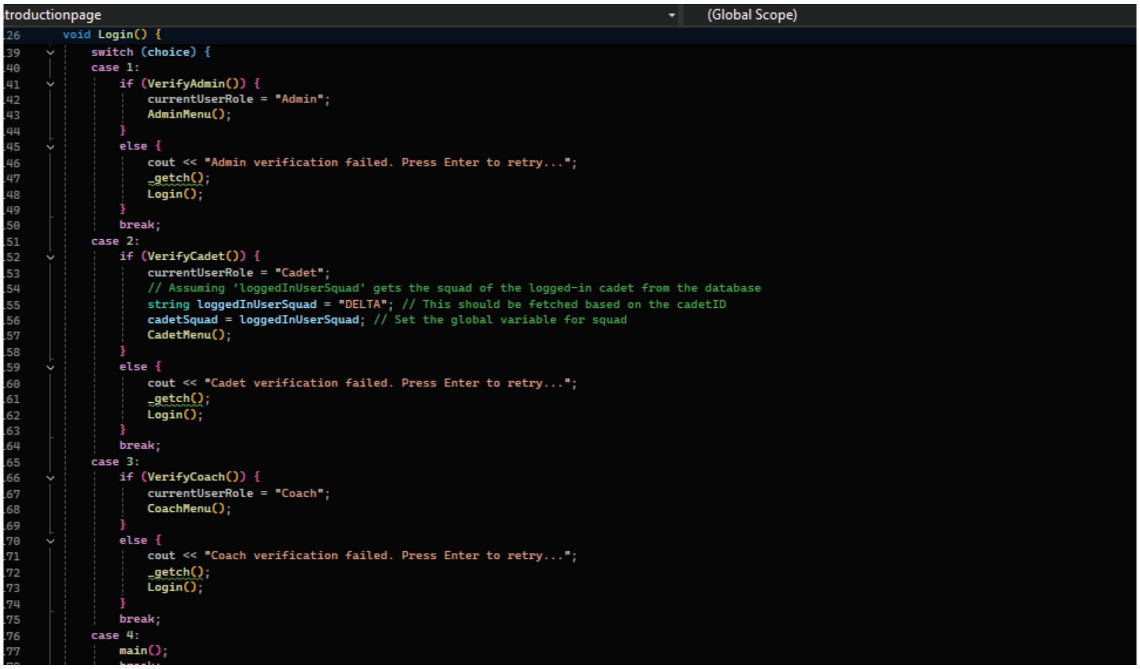


```
void RemoveAdmin() {

    if (!checkState) {
        MYSQL_RES* res = mysql_store_result(conn);
        MYSQL_ROW row = mysql_fetch_row(res);

        if (row && stoi(row[0]) > 0) { // If admin exists
            // Confirm deletion
            cout << "Are you sure you want to delete Admin ID '" << id << "'? (Y/N): ";
            cin >> confirmation;

            // If user confirms, proceed with deletion
            if (toupper(confirmation) == 'Y') {
                string query = "DELETE FROM Admin WHERE adminID = '" + id + "'";
                const char* q = query.c_str();
                int qstate = mysql_query(conn, q);

                if (!qstate) {
                    cout << "Admin removed successfully. Press Enter to continue...";
                    // Reload Admin data into memory if needed
                    // LoadAdminsFromDatabase();
                }
                else {
                    cout << "Query Execution Problem: " << mysql_error(conn) << endl;
                }
            }
            else {
                cout << "Operation cancelled. Press Enter to return to the menu...";
            }
        }
        else {
            // If admin does not exist
            cout << "Error: Admin ID '" << id << "' does not exist. Returning to the menu..." << endl;
            _getch();
            return;
        }
```

Figure 4.4.1. Selection if else in RemoveAdmin ()

The RemoveAdmin function uses selection statements (if, else, switch) to decide the flow of execution based on user inputs and database query results. These statements ensure that the program performs specific actions only when certain conditions are met. It ensures that critical actions like deleting records are performed only when necessary. It also protects against accidental deletions by requiring confirmation. It ensures meaningful feedback is provided if the operation cannot be completed.

4.5 Control



```
void Login() {
    switch (choice) {
    case 1:
        if (VerifyAdmin()) {
            currentUserRole = "Admin";
            AdminMenu();
        }
        else {
            cout << "Admin verification failed. Press Enter to retry...";
            _getch();
            Login();
        }
        break;
    case 2:
        if (VerifyCadet()) {
            currentUserRole = "Cadet";
            // Assuming 'loggedInUserSquad' gets the squad of the logged-in cadet from the database
            string loggedInUserSquad = "DELTA"; // This should be fetched based on the cadetID
            cadetSquad = loggedInUserSquad; // Set the global variable for squad
            CadetMenu();
        }
        else {
            cout << "Cadet verification failed. Press Enter to retry...";
            _getch();
            Login();
        }
        break;
    case 3:
        if (VerifyCoach()) {
            currentUserRole = "Coach";
            CoachMenu();
        }
        else {
            cout << "Coach verification failed. Press Enter to retry...";
            _getch();
            Login();
        }
        break;
    case 4:
        main();
```

Figure 4.5.1 Control function in Login ()

Control in programming refers to mechanisms used to dictate the flow of execution in a program. This includes decision-making, loops, and function calls that determine how and when specific parts of the code are executed. Control structures also ensure that the program behaves as intended based on conditions or user inputs. In your code, control mechanisms are demonstrated through constructs like if-else, switch, and loops.

Decision-Making

The if-else statements validate user credentials and guide the program’s flow based on the verification result. The switch statement determines the user's role and directs them to the correct process or menu.

Error Handling and Retry

If verification fails or an invalid choice is made, the program allows retry without crashing by recursively calling Login.

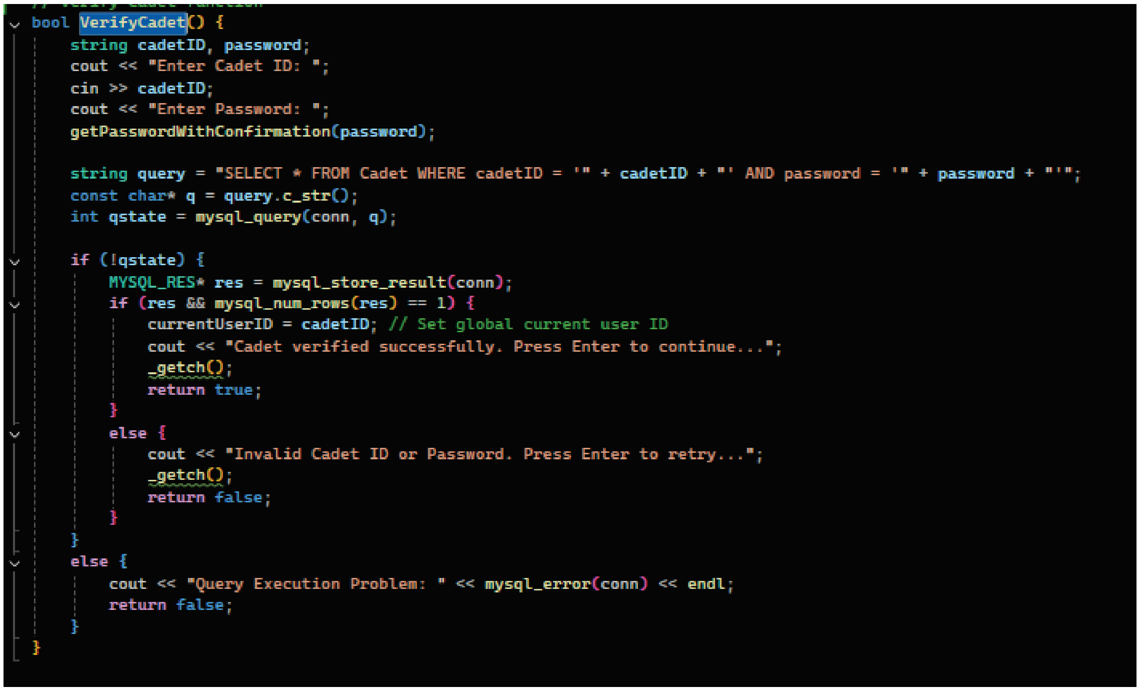
Role-Based Control Flow

Ensures that different roles (Admin, Coach, Cadet) access only their specific functionalities.

Sequential Execution

The control flow maintains logical order, ensuring that each step (input, verification, menu navigation) is executed in sequence.

4.6 Pointer



```
bool VerifyCadet() {
    string cadetID, password;
    cout << "Enter Cadet ID: ";
    cin >> cadetID;
    cout << "Enter Password: ";
    getPasswordWithConfirmation(password);

    string query = "SELECT * FROM Cadet WHERE cadetID = '" + cadetID + "' AND password = '" + password + "'";
    const char* q = query.c_str();
    int qstate = mysql_query(conn, q);

    if (!qstate) {
        MYSQL_RES* res = mysql_store_result(conn);
        if (res && mysql_num_rows(res) == 1) {
            currentUserID = cadetID; // Set global current user ID
            cout << "Cadet verified successfully. Press Enter to continue...";
            _getch();
            return true;
        }
        else {
            cout << "Invalid Cadet ID or Password. Press Enter to retry...";
            _getch();
            return false;
        }
    }
    else {
        cout << "Query Execution Problem: " << mysql_error(conn) << endl;
        return false;
    }
}
```

Figure 4.6.1 Pointer in VerifyCadet ()

MYSQL_RES* res

This pointer is used to store the result set of the query executed using mysql_store_result(conn). It points to the memory where the query result is stored.

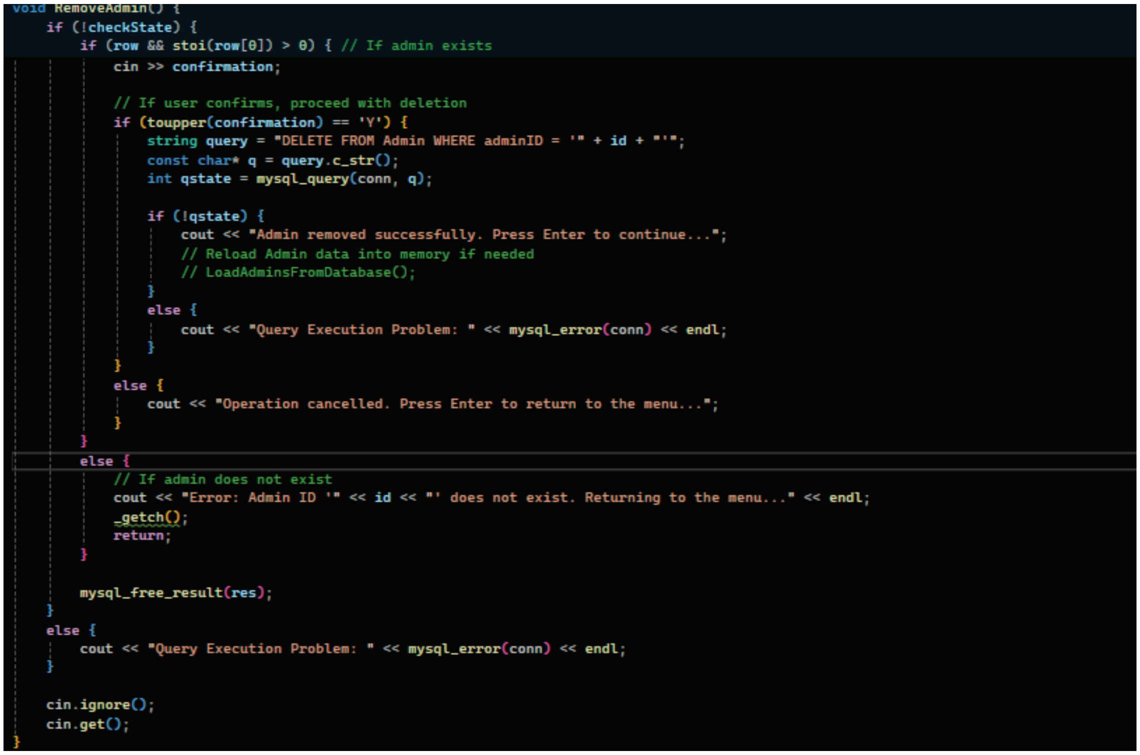
MYSQL_ROW row

While not explicitly visible in this function, row is used in other database-related functions to point to individual rows of data retrieved from the result set.

Query Results Management

The function ensures that memory for the query results (MYSQL_RES*) is allocated and used efficiently. The use of mysql_store_result and subsequent row handling demonstrates how pointers manage dynamic database data.

4.7 Error Handling



```cpp
void RemoveAdmin() {
    if (!checkState) {
        if (row && stoi(row[0]) > 0) { // If admin exists
            cin >> confirmation;

            // If user confirms, proceed with deletion
            if (toupper(confirmation) == 'Y') {
                string query = "DELETE FROM Admin WHERE adminID = '" + id + "'";
                const char* q = query.c_str();
                int qstate = mysql_query(conn, q);

                if (!qstate) {
                    cout << "Admin removed successfully. Press Enter to continue...";
                    // Reload Admin data into memory if needed
                    // LoadAdminsFromDatabase();
                }
                else {
                    cout << "Query Execution Problem: " << mysql_error(conn) << endl;
                }
            }
            else {
                cout << "Operation cancelled. Press Enter to return to the menu...";
            }
        }
        else {
            // If admin does not exist
            cout << "Error: Admin ID '" << id << "' does not exist. Returning to the menu..." << endl;
            _getch();
            return;
        }

        mysql_free_result(res);
    }
    else {
        cout << "Query Execution Problem: " << mysql_error(conn) << endl;
    }

    cin.ignore();
    cin.get();
}
```

Figure 4.7.1 Error Handling in RemoveAdmin()

Database Query Check

The program checks if the query (mysql_query) executes successfully. If it fails, the error is captured using mysql_error(conn) and displayed.

Validation of Results

After executing the query, the program verifies if the target Admin ID exists in the database by checking the result (row and stoi(row[0]) > 0). If the ID does not exist, an error message is displayed, and the function exits.

User Feedback

Clear and meaningful error messages are shown to the user, indicating the issue (e.g., "Admin ID does not exist").

CHAPTER 5: CONCLUSION

5.1 Constraints

The PALAPES UTeM Fitness System faces several limitations, including the use of static data structures such as fixed-size arrays, which restrict scalability and the system's ability to handle a growing number of users. Additionally, the system's heavy reliance on a live MySQL database creates a dependency on constant internet connectivity, making it vulnerable to network disruptions. The console-based interface limits user accessibility and engagement, particularly for users unfamiliar with text-based navigation. Role management is rigid, offering limited flexibility to adapt to complex or evolving organizational structures. Furthermore, the absence of advanced error handling mechanisms, such as exception management, reduces the system's resilience against unexpected runtime issues, and the lack of offline functionality hinders usability in environments with unreliable internet access.

5.2 Future Improvements

To overcome these constraints, the system can adopt dynamic data structures like std: vector or std: map to allow for better scalability and adaptability. Introducing offline functionality with local data caching and synchronization will ensure uninterrupted operations in case of connectivity issues. The implementation of advanced error handling through exception management and detailed logging will enhance the system's reliability. Transitioning to a graphical user interface (GUI) will improve user experience and accessibility, while optimizing database queries will enhance performance and responsiveness. Moreover, introducing flexible role-based permissions, fitness goal tracking, data visualization, and mobile app integration will modernize the system and ensure it meets the diverse and evolving needs of its users.

REFERENCES

Gym, H.G., n.d., Best military style fitness workouts, Hot Ground Gym, accessed 10 January 2025, https://www.hotgroundgym.com/blog/best-military-style-fitness-workouts.

W3Schools.com, n.d., Learn C++, accessed 10 January 2025, https://www.w3schools.com/cpp/.

User, S., n.d., Rejimen Askar Wataniah - PALAPES, accessed 10 January 2025, http://aw.army.mil.my/index.php/pages/palapes.

User, S., n.d., Berita Utama - 1/69, accessed 10 January 2025, http://aw.army.mil.my/index.php/berita-utama/146-1-69.

University of Virginia, n.d., Database setup with XAMPP, accessed 10 January 2025, https://www.cs.virginia.edu/~up3f/cs4750/supplement/DB-setup-xampp.html.

Microsoft, n.d., C++ development features in Visual Studio, accessed 10 January 2025, https://visualstudio.microsoft.com/vs/features/cplusplus/.