Overall Progress :

1. Successfully defined the ethics this project needs to address. To be specific, addressing the people's freedom of accessing information and a code of conductivity.
2. Progressively Documenting Paper.
3. Implemented the laws and orders the group needs to address for ethical standards.
4. Finished Introduction.
5. Implemented and scheduled implementation.
6. The group successfully discussed the needs and prerequisites of the project.
7. Additionally stated and reviewing more related literature to strengthen the project's reliability.
8. Successfully explored the needs and the things the project needs to accomplish and its functionality.
9. Clearly defined the implementation of CRUD and the way of conduct on addressing the project.
10. Successfully visualized its functionality and its future use for the public.
11. Progressively coding and debugging in an attempt to address CRUD properly.
12. Planned the required IDE and compiler that will be used among the group.
13. Planned how the project will be executed and work step by step through outlining ideas for the project.
14. Finished Objectives.
15. Started final recommendations.
16. Visualized flow chart and pseudocode execution.
17. Learnings on how our project will be implemented.
18. Small progress on our PPT.

Final Progress Report

19. Finished the overall code and successfully implemented CRUD.
18. Each groupmate successfully shared their code and its meaning for documentation.
20. Finished Pseudocode.
21. Each groupmate did their own part in executing the project.
22. Most of the flow chart, data dictionaries are done, just finishing it.
22. Finalizing the paper.
23. Successfully explored libraries and its usage and implemented it to the project.
24. Successfully used the including header, cpp, and text file in one project.
25. Successfully executed 2 Online Meetings for discussion and presentation of progress.

Group Contribution:
Feria, Louis Andrei

1. Mainly leading the Group.
2. Made the specific problem, and solutions about the General Problem.
3. Creation of code outline, Progressively documenting paper.
4. RRL Citing.
5. Project Planning and Discussion.
6. Contributed on Objectives.
7. Code Ideas of Execution.
8. Contributed Learnings.
9. Code sharing.
10. Explained his code and executed his part as a team.
11. Linking of codes of all members(Each member is assigned to specific tasks)
12. Made the login system.
13. Handles User Registration consist of Username and Password.

Demain, **Jaime Luis**

1. Specific problem and Solution part of Introduction
2. Implementation of the project
3. Contributed on research of RRLs
4. Contributed on the creation of code outline
5. Progressively documenting paper
6. Searching RRL's, rewordings
7. Contributed Learnings.
8. Code sharing.
9. Explained his code and executed his part as a team.
10. Making Flowchart.
11. Code handles post displaying, post filtering, and post editing or modifying.

Mendoza, Nathaniel

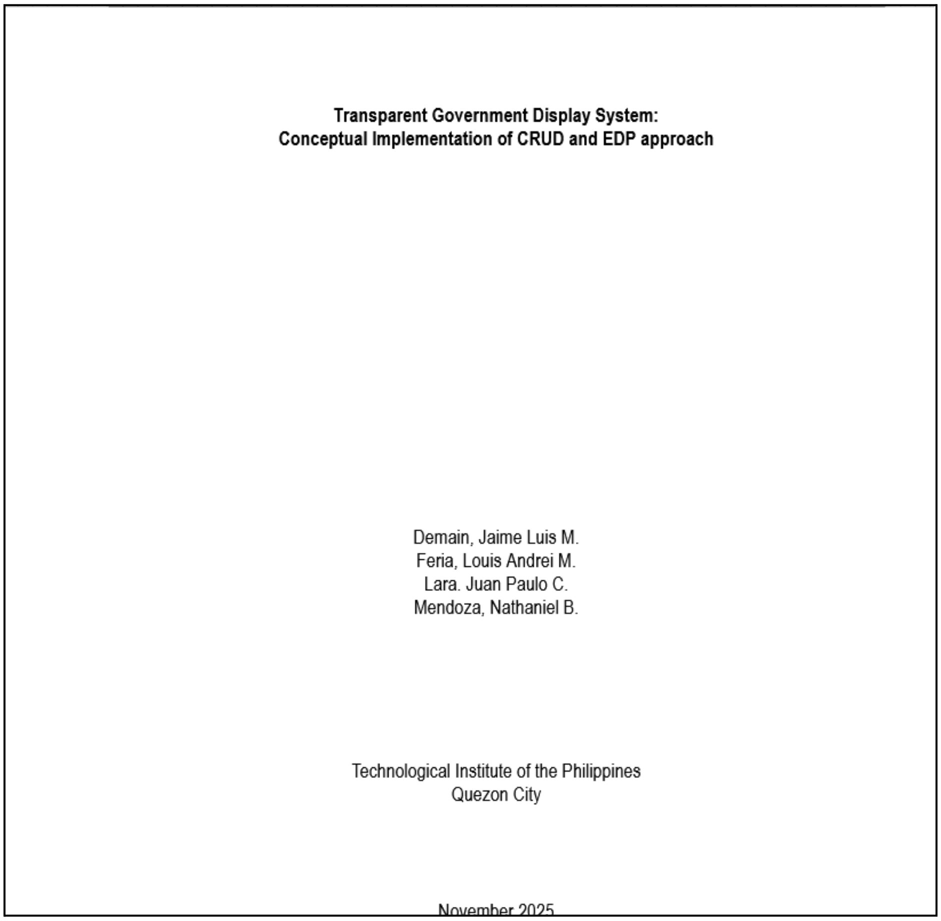
1. Idea in implementation of CRUD system for the project
2. Additional information for the specific problem.
3. Making of the specific problem and solutions about our general problem.
4. Idea of what is gonna be our general problem.
5. Sharing of ideas and planning.
6. Contributed to the creation of code outline.
7. Progressively documenting paper.
8. Searching for RRL's.
9. Contributed on Objectives,

10. Sharing of ideas of the code execution,
11. Contributed Learnings.
12. Code sharing.
13. Explained his code and executed his part as a team.
14. Making the Pseudocode.
15. Separation of Group Responsibilities and work.
16. Code Handles on deleting posts within the txt file and transferring to the archives.

Lara, Juan Paulo

1. Title writing
2. Implementation of the solution and project,
3. Additional findings and references on the specific problem,
4. Contributed on the creation of code outline,
5. Searching for RRL's.
6. Contributed in CRUD implementation
7. Contributed on Objectives.
8. Contributed Learnings.
9. Code sharing.
10. Explained his code and executed his part as a team.
11. Making a Data Dictionary.
12. Code Handles in the creation of posts.

Overall Documentation:



Transparent Government Display System:
Conceptual Implementation of CRUD and EDP approach

Demain, Jaime Luis M.
Feria, Louis Andrei M.
Lara. Juan Paulo C.
Mendoza, Nathaniel B.

Technological Institute of the Philippines
Quezon City

November 2025

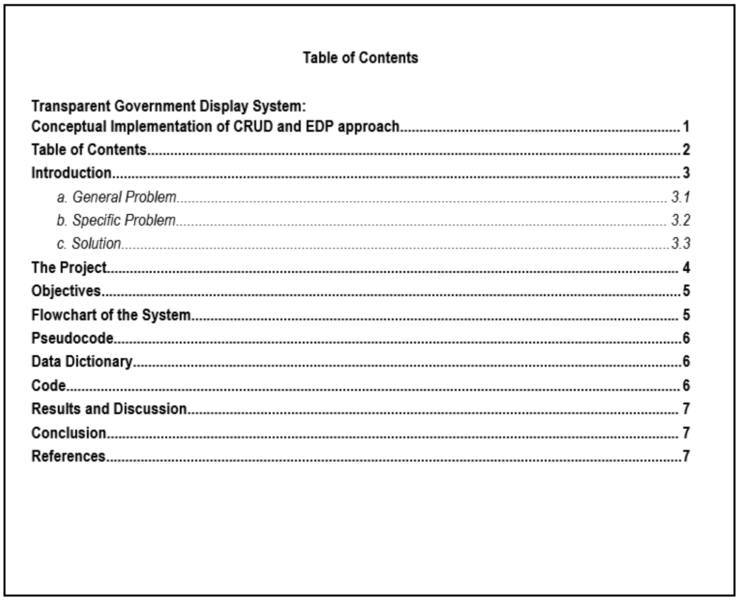
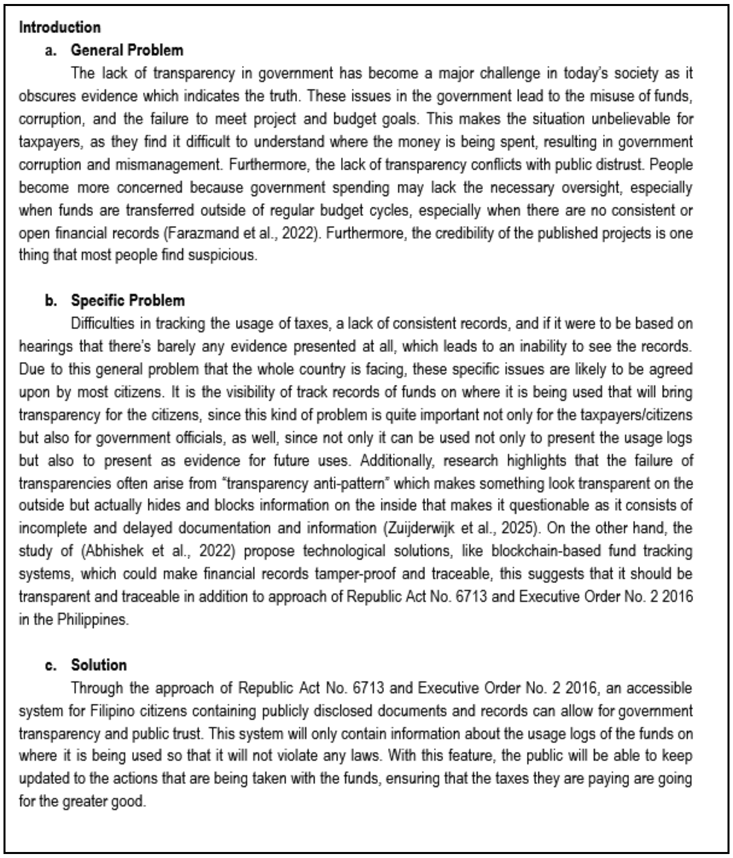


Table of Contents



Introduction

a. General Problem

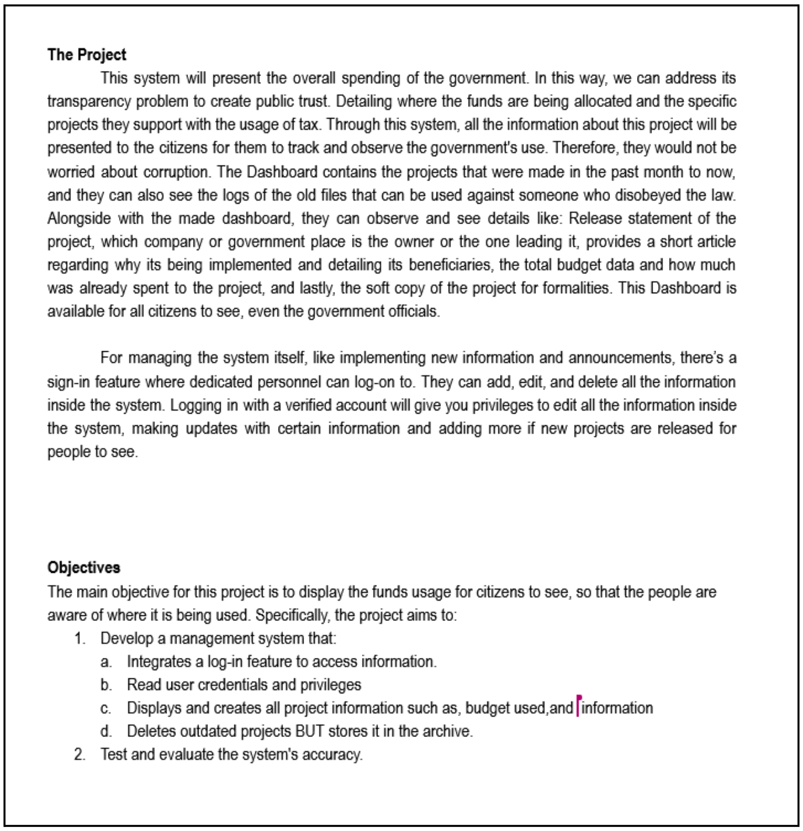
The lack of transparency in government has become a major challenge in today's society as it obscures evidence which indicates the truth. These issues in the government lead to the misuse of funds, corruption, and the failure to meet project and budget goals. This makes the situation unbelievable for taxpayers, as they find it difficult to understand where the money is being spent, resulting in government corruption and mismanagement. Furthermore, the lack of transparency conflicts with public distrust. People become more concerned because government spending may lack the necessary oversight, especially when funds are transferred outside of regular budget cycles, especially when there are no consistent or open financial records (Farazmand et al., 2022). Furthermore, the credibility of the published projects is one thing that most people find suspicious.

b. Specific Problem

Difficulties in tracking the usage of taxes, a lack of consistent records, and if it were to be based on hearings that there's barely any evidence presented at all, which leads to an inability to see the records. Due to this general problem that the whole country is facing, these specific issues are likely to be agreed upon by most citizens. It is the visibility of track records of funds on where it is being used that will bring transparency for the citizens, since this kind of problem is quite important not only for the taxpayers/citizens but also for government officials, as well, since not only it can be used not only to present the usage logs but also to present as evidence for future uses. Additionally, research highlights that the failure of transparencies often arise from "transparency anti-pattern" which makes something look transparent on the outside but actually hides and blocks information on the inside that makes it questionable as it consists of incomplete and delayed documentation and information (Zuijderwijk et al., 2025). On the other hand, the study of (Abhishek et al., 2022) propose technological solutions, like blockchain-based fund tracking systems, which could make financial records tamper-proof and traceable, this suggests that it should be transparent and traceable in addition to approach of Republic Act No. 6713 and Executive Order No. 2 2016 in the Philippines.

c. Solution

Through the approach of Republic Act No. 6713 and Executive Order No. 2 2016, an accessible system for Filipino citizens containing publicly disclosed documents and records can allow for government transparency and public trust. This system will only contain information about the usage logs of the funds on where it is being used so that it will not violate any laws. With this feature, the public will be able to keep updated to the actions that are being taken with the funds, ensuring that the taxes they are paying are going for the greater good.



The Project

This system will present the overall spending of the government. In this way, we can address its transparency problem to create public trust. Detailing where the funds are being allocated and the specific projects they support with the usage of tax. Through this system, all the information about this project will be presented to the citizens for them to track and observe the government's use. Therefore, they would not be worried about corruption. The Dashboard contains the projects that were made in the past month to now, and they can also see the logs of the old files that can be used against someone who disobeyed the law. Alongside with the made dashboard, they can observe and see details like: Release statement of the project, which company or government place is the owner or the one leading it, provides a short article regarding why its being implemented and detailing its beneficiaries, the total budget data and how much was already spent to the project, and lastly, the soft copy of the project for formalities. This Dashboard is available for all citizens to see, even the government officials.

For managing the system itself, like implementing new information and announcements, there's a sign-in feature where dedicated personnel can log-on to. They can add, edit, and delete all the information inside the system. Logging in with a verified account will give you privileges to edit all the information inside the system, making updates with certain information and adding more if new projects are released for people to see.

Objectives

The main objective for this project is to display the funds usage for citizens to see, so that the people are aware of where it is being used. Specifically, the project aims to:

1. Develop a management system that:
   a. Integrates a log-in feature to access information.
   b. Read user credentials and privileges
   c. Displays and creates all project information such as, budget used,and information
   d. Deletes outdated projects BUT stores it in the archive.
2. Test and evaluate the system's accuracy.

Conclusion

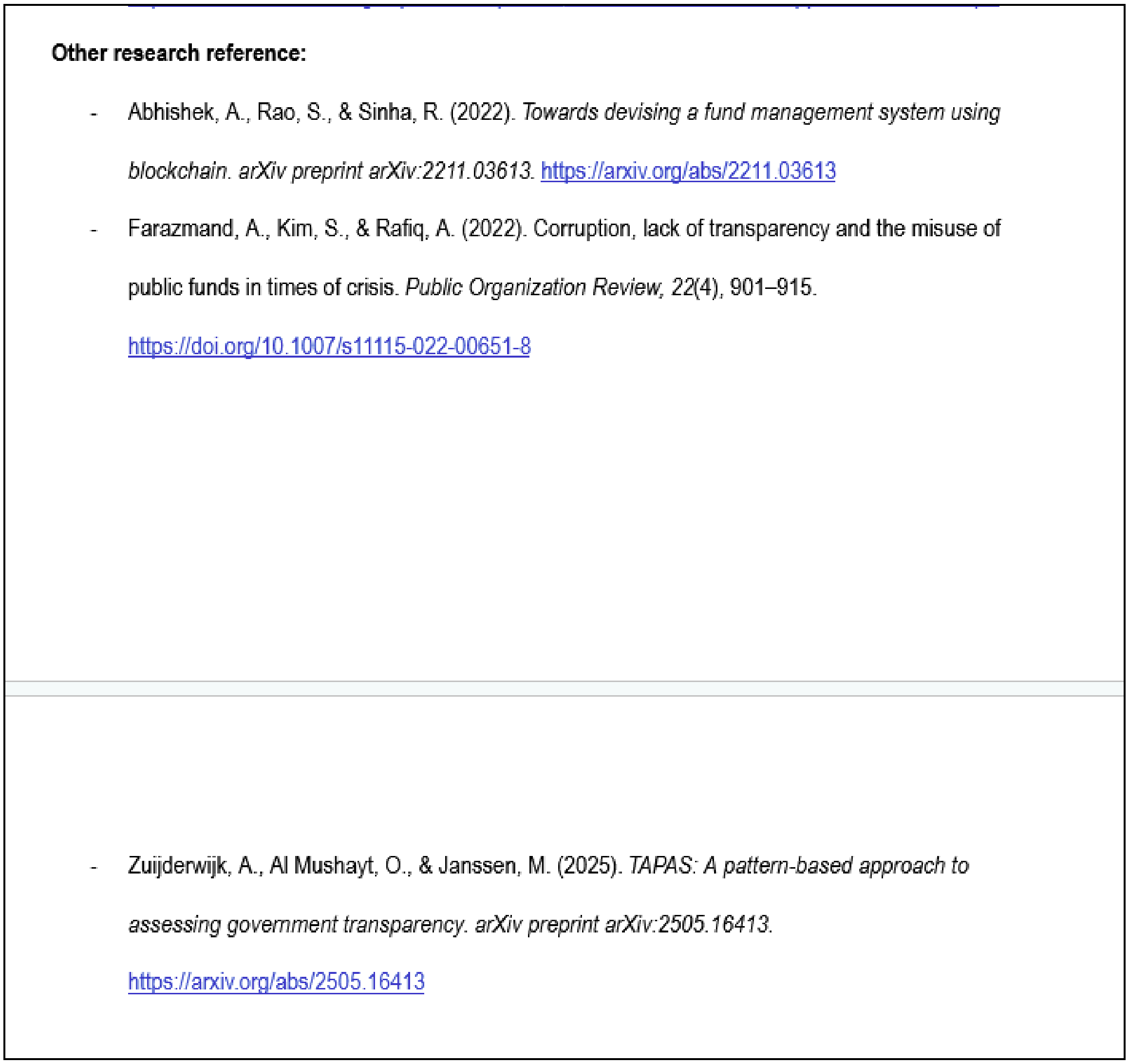
Recommendation:

Through the making of this project, the students had some recommendation to fully maximize its potential, specifically the should at least:

1. Consist a Graphic User Interface system for better user experience and beginner friendly..
2. Would also consist of user login for the citizens as well.
3. The Security System among accounts has vulnerabilities.
4. Consists of a larger data base as it is only a prototype.
5. More user interaction and tabs for deeper information regarding the project. (Example : Lists people that are connected to the project)

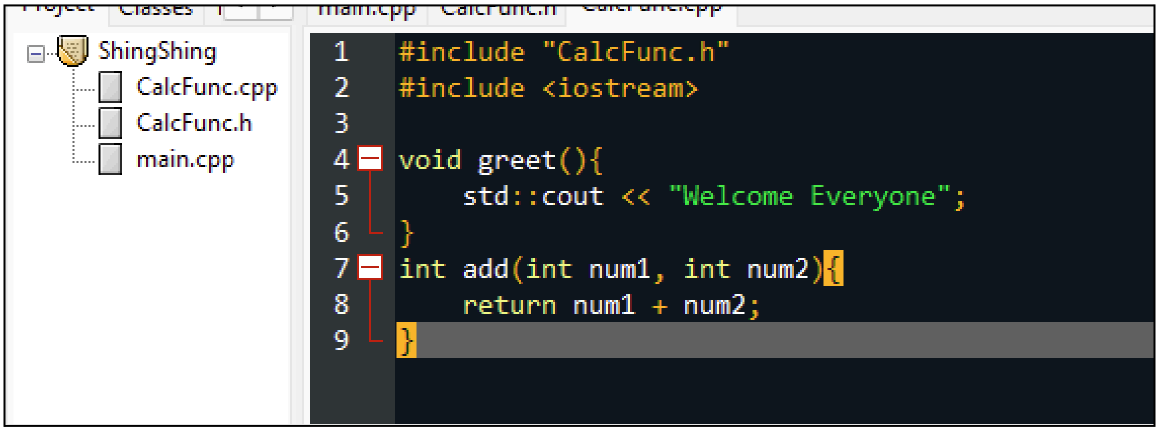
Justification Reference:

- *An act establishing a code of conduct and Ethical Standards for public officials and employees, to uphold the time-honored principles of public office being a public trust, granting incentives and rewards for exemplary service, enumerating prohibited acts and transactions and providing penalties for violations thereof and for other purposes. (1989). In Republic Act NO. 6713 [Legal document].* https://www.ombudsman.gov.ph/docs/republicacts/Republic_Act_No_6713.pdf
- *Duterte, R. R. & Philippines. (2016). Executive Order No. 02, s. 2016. In MALACAÑANG PALACE.* https://www.pdic.gov.ph/files/foi/Annex_A-EO_2_FOI.pdf
- *Ombudsman of the Philippines. (1989). Republic Act No. 6713: Code of conduct and ethical standards for public officials and employees.* https://www.ombudsman.gov.ph/docs/republicacts/Republic_Act_No_6713.pdf
- *The 1987 Constitution of the Republic of the Philippines, Article XI: Accountability of Public Officers.* (n.d.). Office of the Ombudsman (Philippines). https://www.ombudsman.gov.ph/docs/republicacts/Article_XI_1987_Philippine_Constitution.pdf

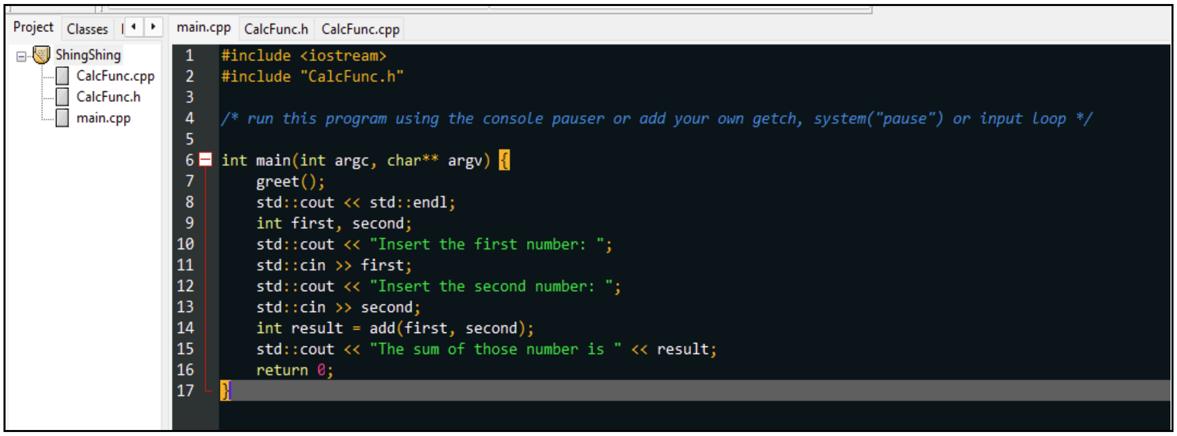


Other research reference:

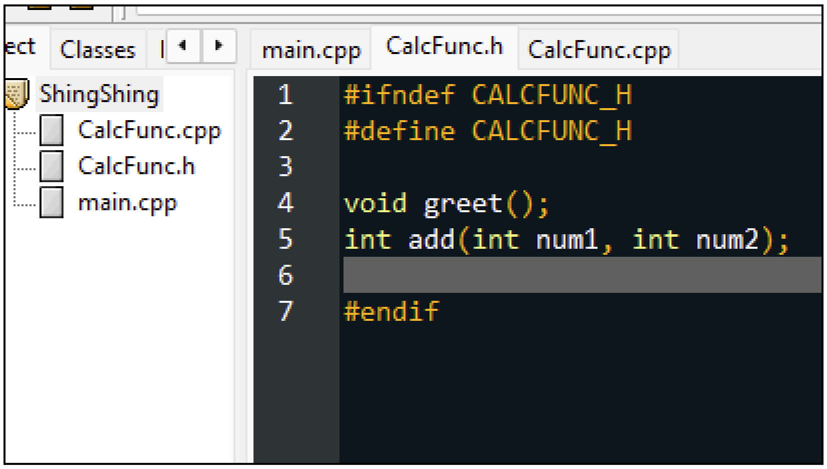
- Abhishek, A., Rao, S., & Sinha, R. (2022). *Towards devising a fund management system using blockchain. arXiv preprint arXiv:2211.03613.* https://arxiv.org/abs/2211.03613
- Farazmand, A., Kim, S., & Rafiq, A. (2022). Corruption, lack of transparency and the misuse of public funds in times of crisis. *Public Organization Review, 22*(4), 901–915. https://doi.org/10.1007/s11115-022-00651-8
- Zuijderwijk, A., Al Mushayt, O., & Janssen, M. (2025). *TAPAS: A pattern-based approach to assessing government transparency. arXiv preprint arXiv:2505.16413.* https://arxiv.org/abs/2505.16413



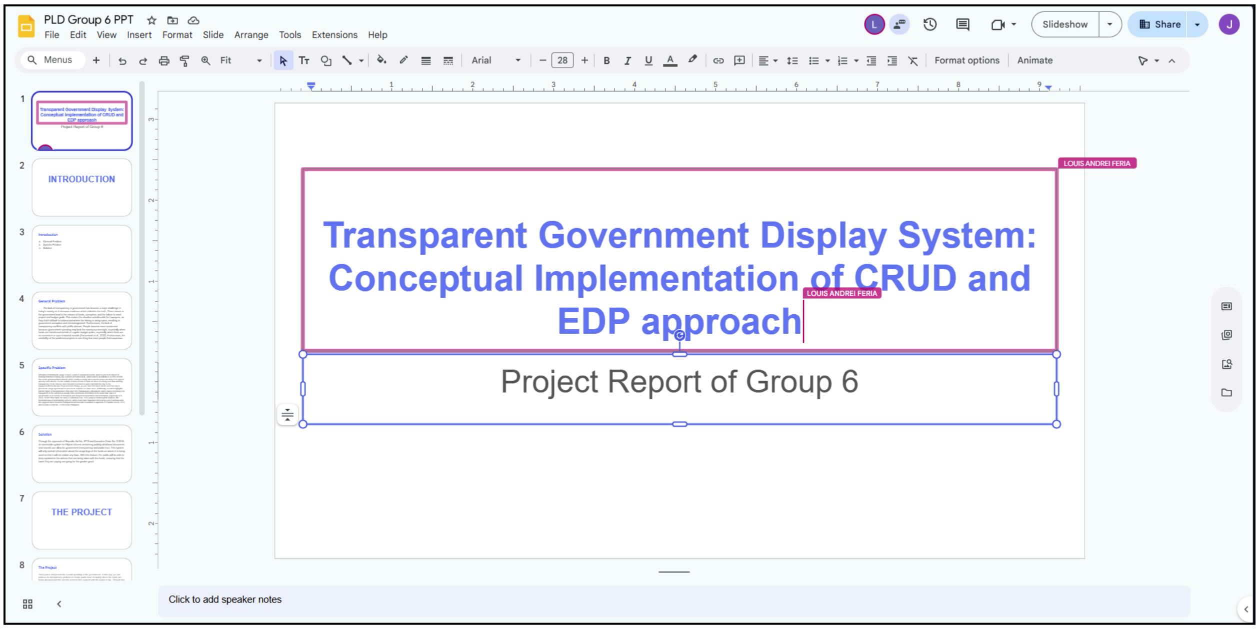
```cpp
#include "CalcFunc.h"
#include <iostream>

void greet(){
    std::cout << "Welcome Everyone";
}
int add(int num1, int num2){
    return num1 + num2;
}
```



```cpp
#include <iostream>
#include "CalcFunc.h"

/* run this program using the console pauser or add your own getch, system("pause") or input loop */

int main(int argc, char** argv) {
    greet();
    std::cout << std::endl;
    int first, second;
    std::cout << "Insert the first number: ";
    std::cin >> first;
    std::cout << "Insert the second number: ";
    std::cin >> second;
    int result = add(first, second);
    std::cout << "The sum of those number is " << result;
    return 0;
}
```

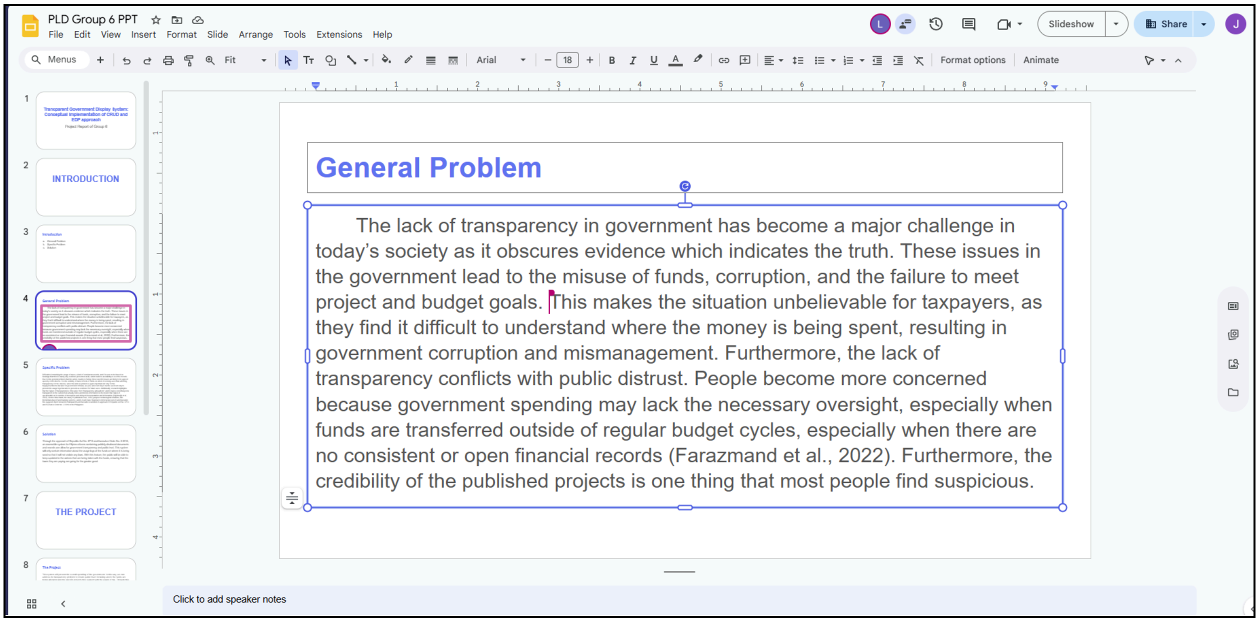


```cpp
#ifndef CALCFUNC_H
#define CALCFUNC_H

void greet();
int add(int num1, int num2);

#endif
```



Transparent Government Display System: Conceptual Implementation of CRUD and EDP approach

Project Report of Group 6

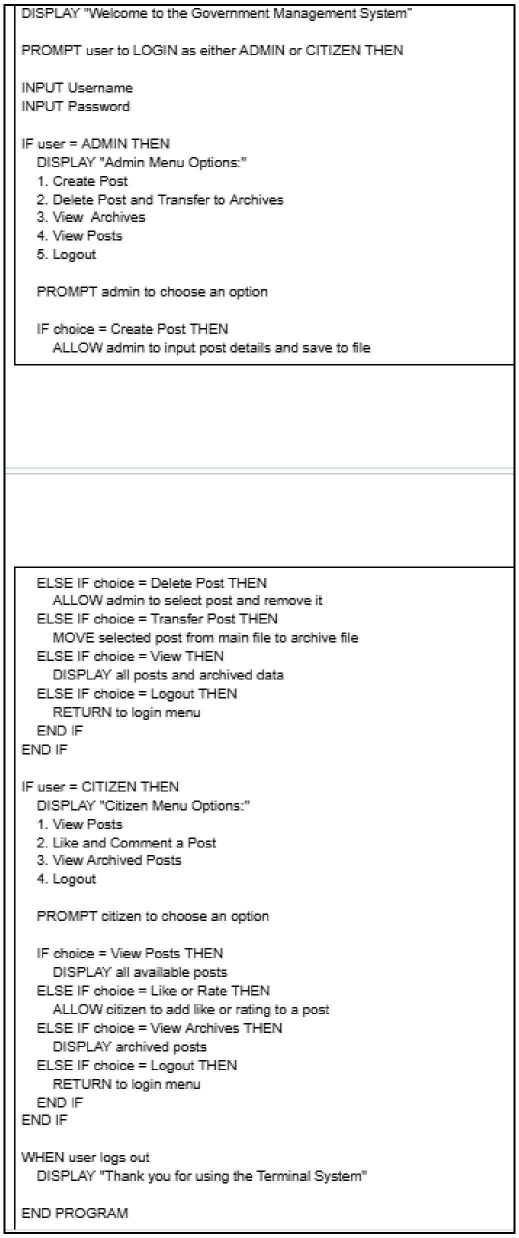

PLD Group 6 PPT
File Edit View Insert Format Slide Arrange Tools Extensions Help
Slideshow
Share
Menus
Fit
Arial
18
Format options
Animate
INTRODUCTION
THE PROJECT
Click to add speaker notes

General Problem

The lack of transparency in government has become a major challenge in today's society as it obscures evidence which indicates the truth. These issues in the government lead to the misuse of funds, corruption, and the failure to meet project and budget goals. This makes the situation unbelievable for taxpayers, as they find it difficult to understand where the money is being spent, resulting in government corruption and mismanagement. Furthermore, the lack of transparency conflicts with public distrust. People become more concerned because government spending may lack the necessary oversight, especially when funds are transferred outside of regular budget cycles, especially when there are no consistent or open financial records (Farazmand et al., 2022). Furthermore, the credibility of the published projects is one thing that most people find suspicious.

Final Progress :

Pseudocode (Inside Document)



```
DISPLAY "Welcome to the Government Management System"

PROMPT user to LOGIN as either ADMIN or CITIZEN THEN

INPUT Username
INPUT Password

IF user = ADMIN THEN
   DISPLAY "Admin Menu Options:"
   1. Create Post
   2. Delete Post and Transfer to Archives
   3. View  Archives
   4. View Posts
   5. Logout

   PROMPT admin to choose an option

   IF choice = Create Post THEN
      ALLOW admin to input post details and save to file
```

```
   ELSE IF choice = Delete Post THEN
      ALLOW admin to select post and remove it
   ELSE IF choice = Transfer Post THEN
      MOVE selected post from main file to archive file
   ELSE IF choice = View THEN
      DISPLAY all posts and archived data
   ELSE IF choice = Logout THEN
      RETURN to login menu
   END IF
END IF

IF user = CITIZEN THEN
   DISPLAY "Citizen Menu Options:"
   1. View Posts
   2. Like and Comment a Post
   3. View Archived Posts
   4. Logout

   PROMPT citizen to choose an option

   IF choice = View Posts THEN
      DISPLAY all available posts
   ELSE IF choice = Like or Rate THEN
      ALLOW citizen to add like or rating to a post
   ELSE IF choice = View Archives THEN
      DISPLAY archived posts
   ELSE IF choice = Logout THEN
      RETURN to login menu
   END IF
END IF

WHEN user logs out
   DISPLAY "Thank you for using the Terminal System"

END PROGRAM
```

Data Dictionary
(Inside Document)

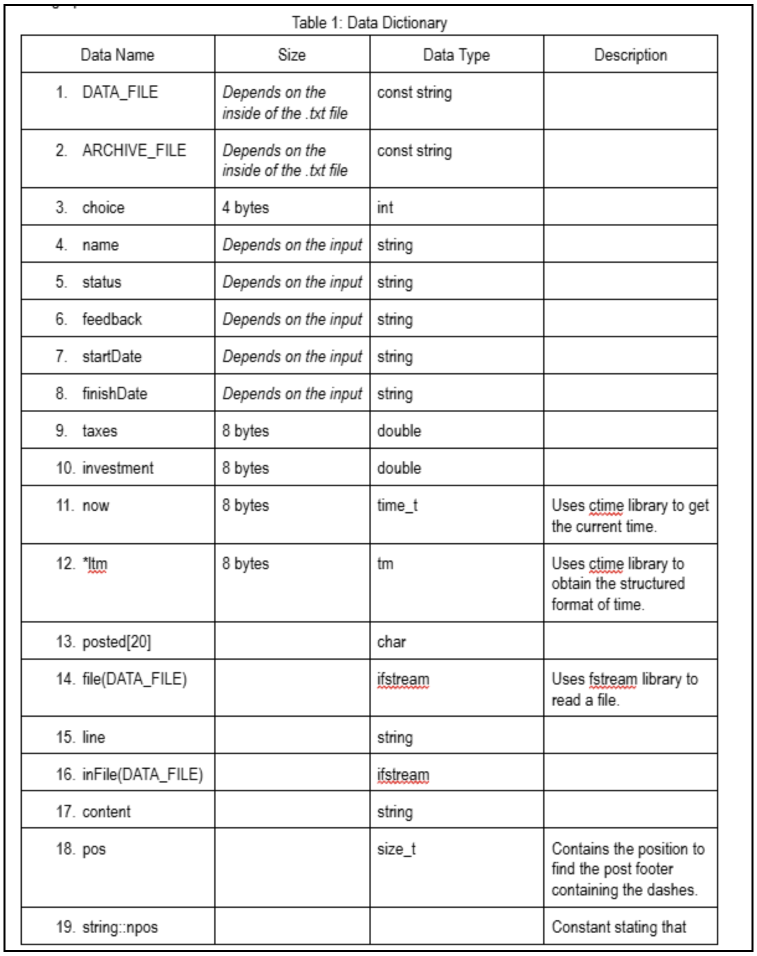
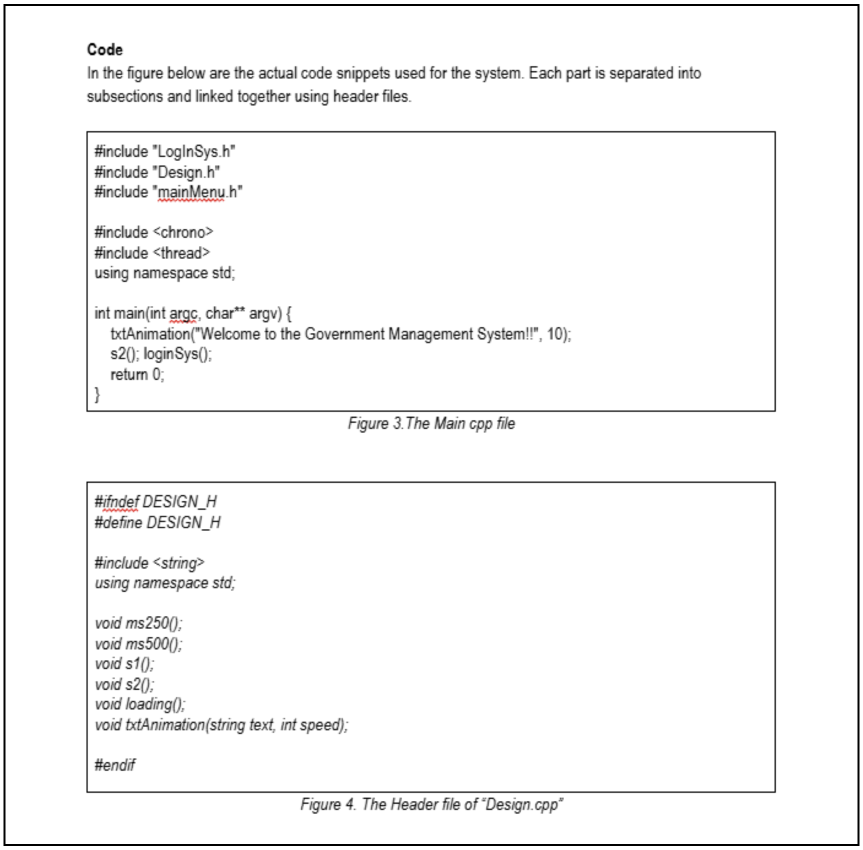


Table 1: Data Dictionary

Data Name	Size	Data Type	Description
1. DATA_FILE	*Depends on the inside of the .txt file*	const string	
2. ARCHIVE_FILE	*Depends on the inside of the .txt file*	const string	
3. choice	4 bytes	int	
4. name	*Depends on the input*	string	
5. status	*Depends on the input*	string	
6. feedback	*Depends on the input*	string	
7. startDate	*Depends on the input*	string	
8. finishDate	*Depends on the input*	string	
9. taxes	8 bytes	double	
10. investment	8 bytes	double	
11. now	8 bytes	time_t	Uses ctime library to get the current time.
12. *ltm	8 bytes	tm	Uses ctime library to obtain the structured format of time.
13. posted[20]		char	
14. file(DATA_FILE)		ifstream	Uses fstream library to read a file.
15. line		string	
16. inFile(DATA_FILE)		ifstream	
17. content		string	
18. pos		size_t	Contains the position to find the post footer containing the dashes.
19. string::npos			Constant stating that

Code
(Inside Document)



Code

In the figure below are the actual code snippets used for the system. Each part is separated into subsections and linked together using header files.

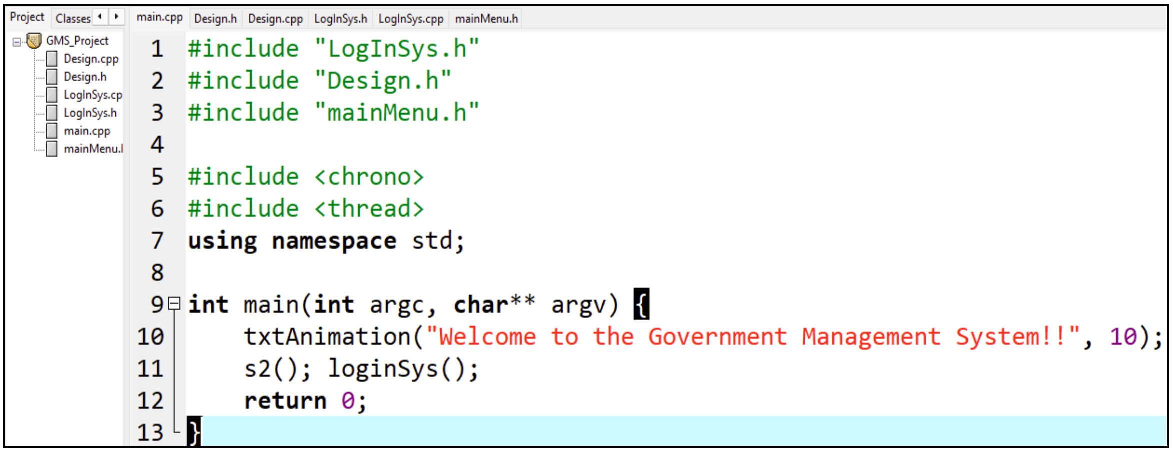
```
#include "LogInSys.h"
#include "Design.h"
#include "mainMenu.h"

#include <chrono>
#include <thread>
using namespace std;

int main(int argc, char** argv) {
    txtAnimation("Welcome to the Government Management System!!", 10);
    s2(); loginSys();
    return 0;
}
```

Figure 3.The Main cpp file

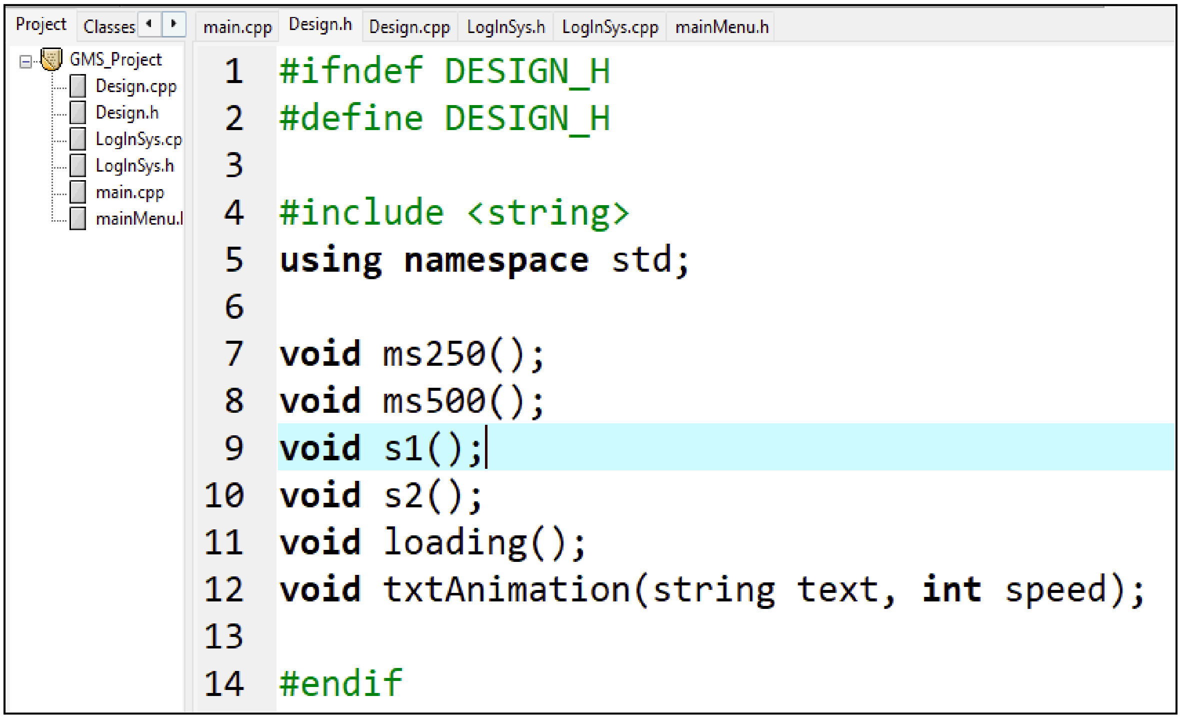
```
#ifndef DESIGN_H
#define DESIGN_H

#include <string>
using namespace std;

void ms250();
void ms500();
void s1();
void s2();
void loading();
void txtAnimation(string text, int speed);

#endif
```

Figure 4. The Header file of "Design.cpp"

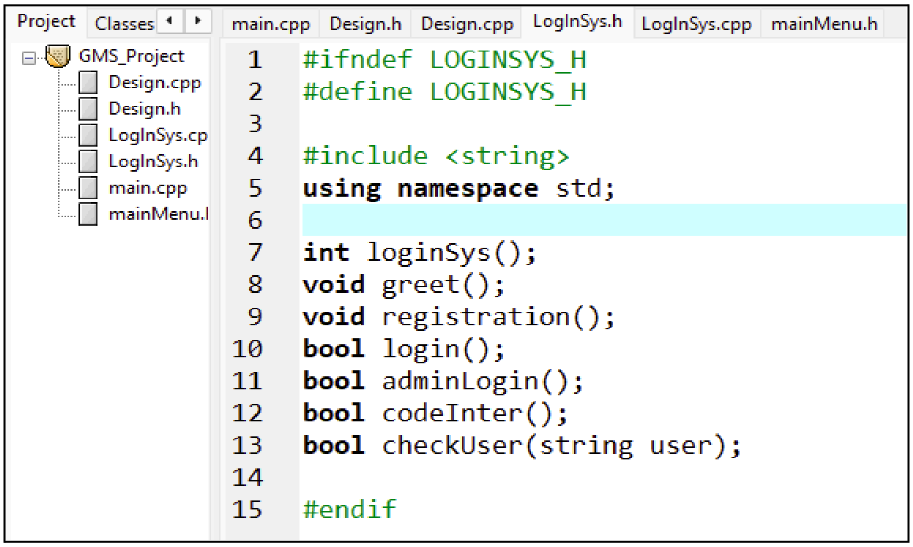
Code Finalization (Compiled into 1 project contributed by each member)



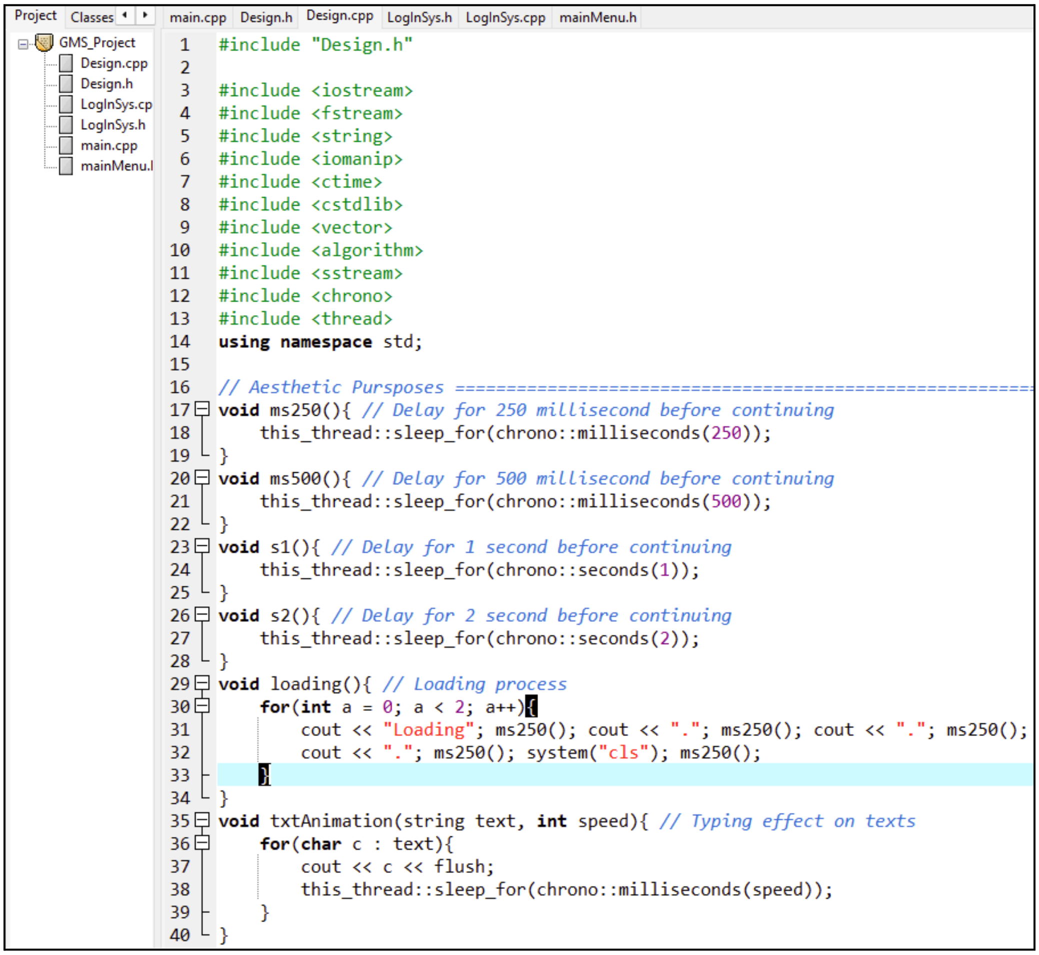
```cpp
#include "LogInSys.h"
#include "Design.h"
#include "mainMenu.h"

#include <chrono>
#include <thread>
using namespace std;

int main(int argc, char** argv) {
    txtAnimation("Welcome to the Government Management System!!", 10);
    s2(); loginSys();
    return 0;
}
```



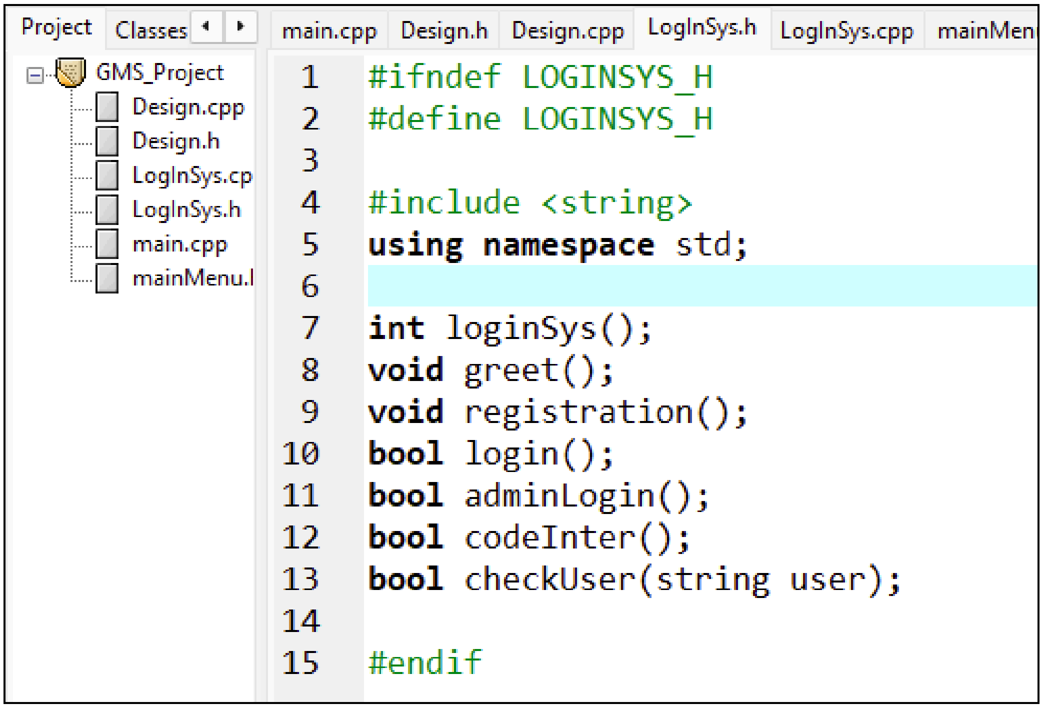
```cpp
#ifndef DESIGN_H
#define DESIGN_H

#include <string>
using namespace std;

void ms250();
void ms500();
void s1();
void s2();
void loading();
void txtAnimation(string text, int speed);

#endif
```



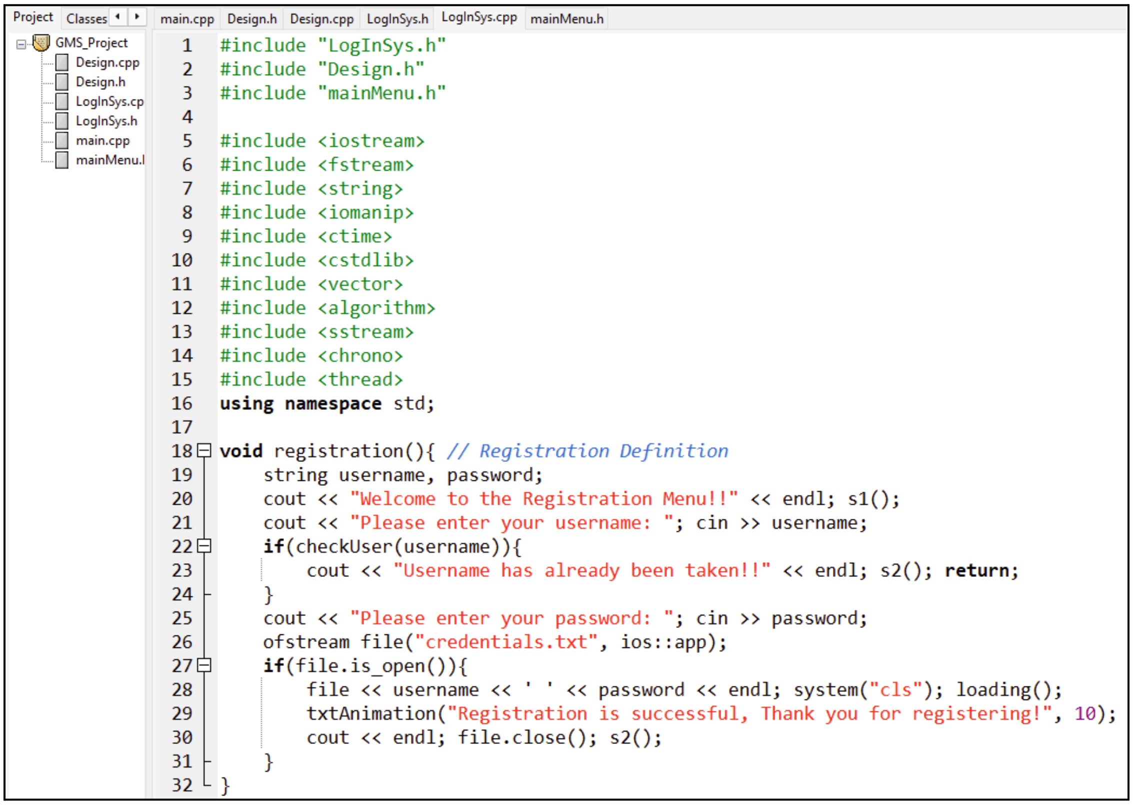
```cpp
#ifndef LOGINSYS_H
#define LOGINSYS_H

#include <string>
using namespace std;

int loginSys();
void greet();
void registration();
bool login();
bool adminLogin();
bool codeInter();
bool checkUser(string user);

#endif
```



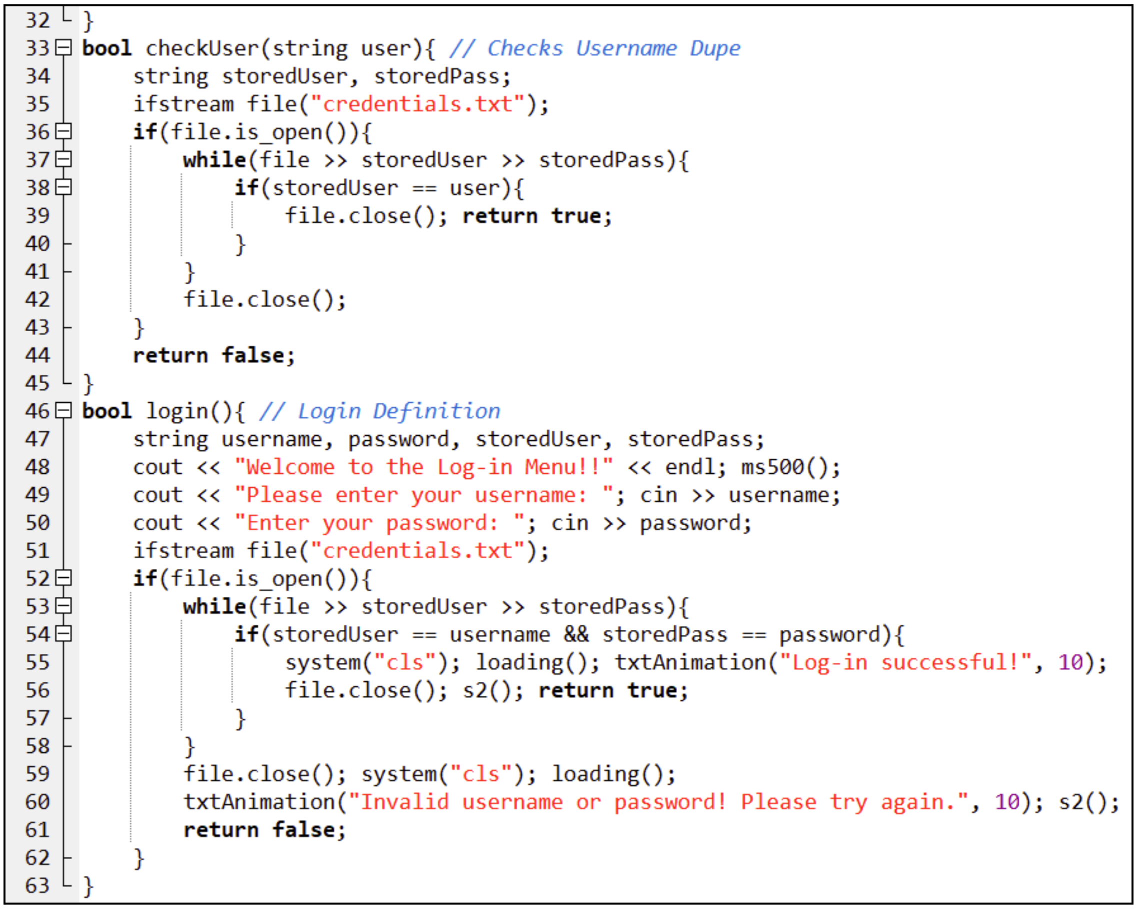
```cpp
#include "Design.h"

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <ctime>
#include <cstdlib>
#include <vector>
#include <algorithm>
#include <sstream>
#include <chrono>
#include <thread>
using namespace std;

// Aesthetic Pursposes ========================================================
void ms250(){ // Delay for 250 millisecond before continuing
    this_thread::sleep_for(chrono::milliseconds(250));
}
void ms500(){ // Delay for 500 millisecond before continuing
    this_thread::sleep_for(chrono::milliseconds(500));
}
void s1(){ // Delay for 1 second before continuing
    this_thread::sleep_for(chrono::seconds(1));
}
void s2(){ // Delay for 2 second before continuing
    this_thread::sleep_for(chrono::seconds(2));
}
void loading(){ // Loading process
    for(int a = 0; a < 2; a++){
        cout << "Loading"; ms250(); cout << "."; ms250(); cout << "."; ms250();
        cout << "."; ms250(); system("cls"); ms250();
    }
}
void txtAnimation(string text, int speed){ // Typing effect on texts
    for(char c : text){
        cout << c << flush;
        this_thread::sleep_for(chrono::milliseconds(speed));
    }
}
```



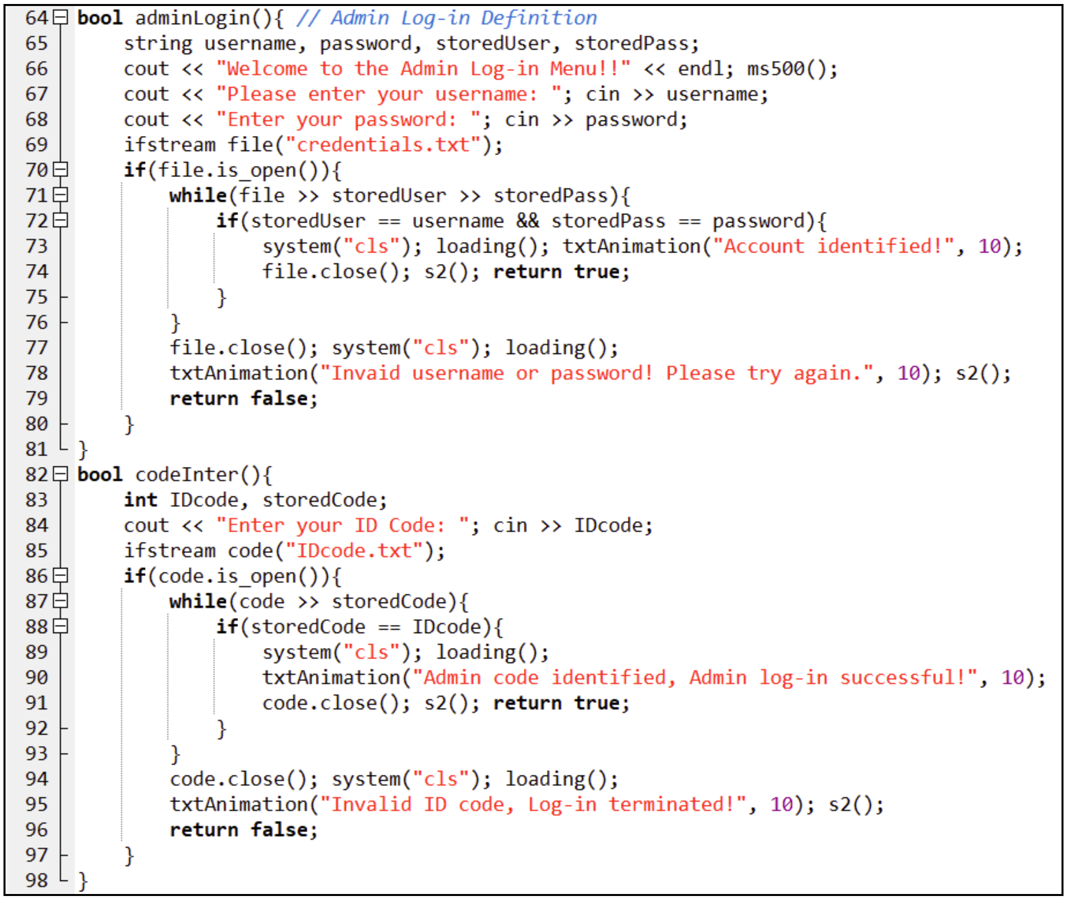
```cpp
#ifndef LOGINSYS_H
#define LOGINSYS_H

#include <string>
using namespace std;

int loginSys();
void greet();
void registration();
bool login();
bool adminLogin();
bool codeInter();
bool checkUser(string user);

#endif
```



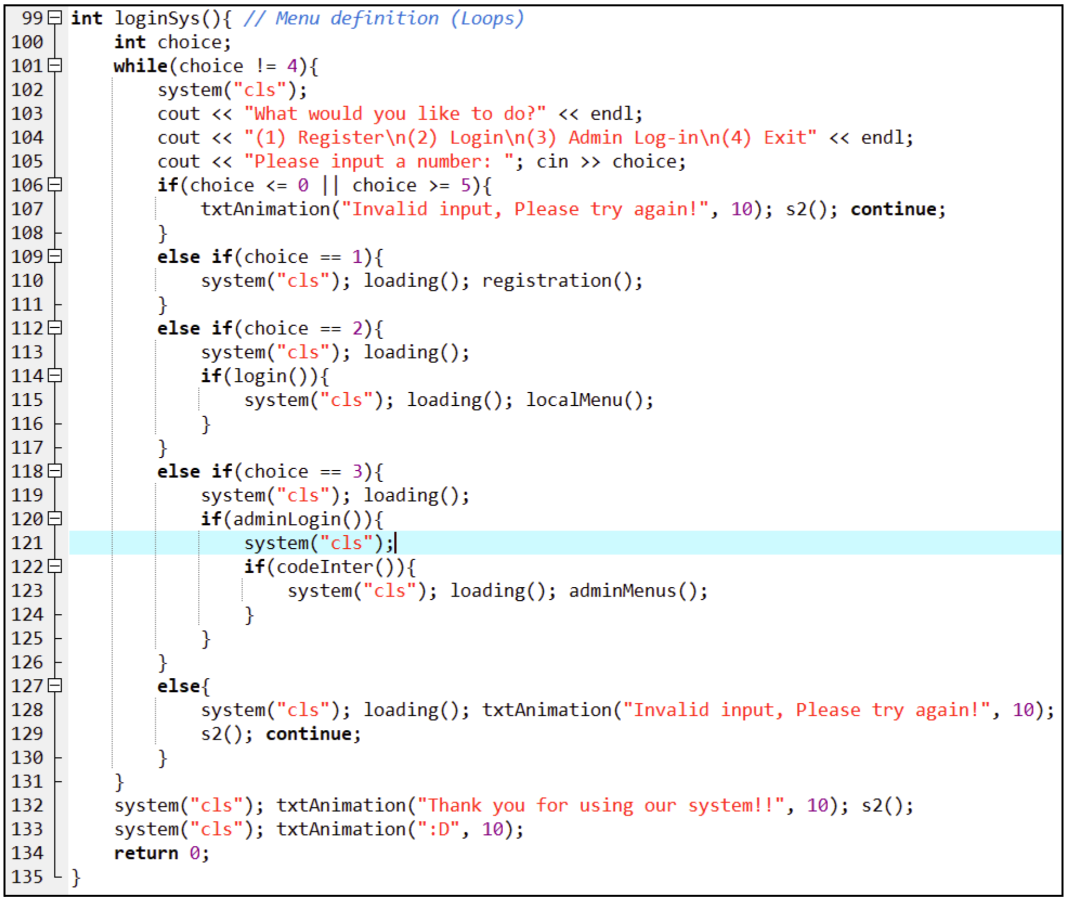
```cpp
#include "LogInSys.h"
#include "Design.h"
#include "mainMenu.h"

#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <ctime>
#include <cstdlib>
#include <vector>
#include <algorithm>
#include <sstream>
#include <chrono>
#include <thread>
using namespace std;

void registration(){ // Registration Definition
    string username, password;
    cout << "Welcome to the Registration Menu!!" << endl; s1();
    cout << "Please enter your username: "; cin >> username;
    if(checkUser(username)){
        cout << "Username has already been taken!!" << endl; s2(); return;
    }
    cout << "Please enter your password: "; cin >> password;
    ofstream file("credentials.txt", ios::app);
    if(file.is_open()){
        file << username << ' ' << password << endl; system("cls"); loading();
        txtAnimation("Registration is successful, Thank you for registering!", 10);
        cout << endl; file.close(); s2();
    }
}
```



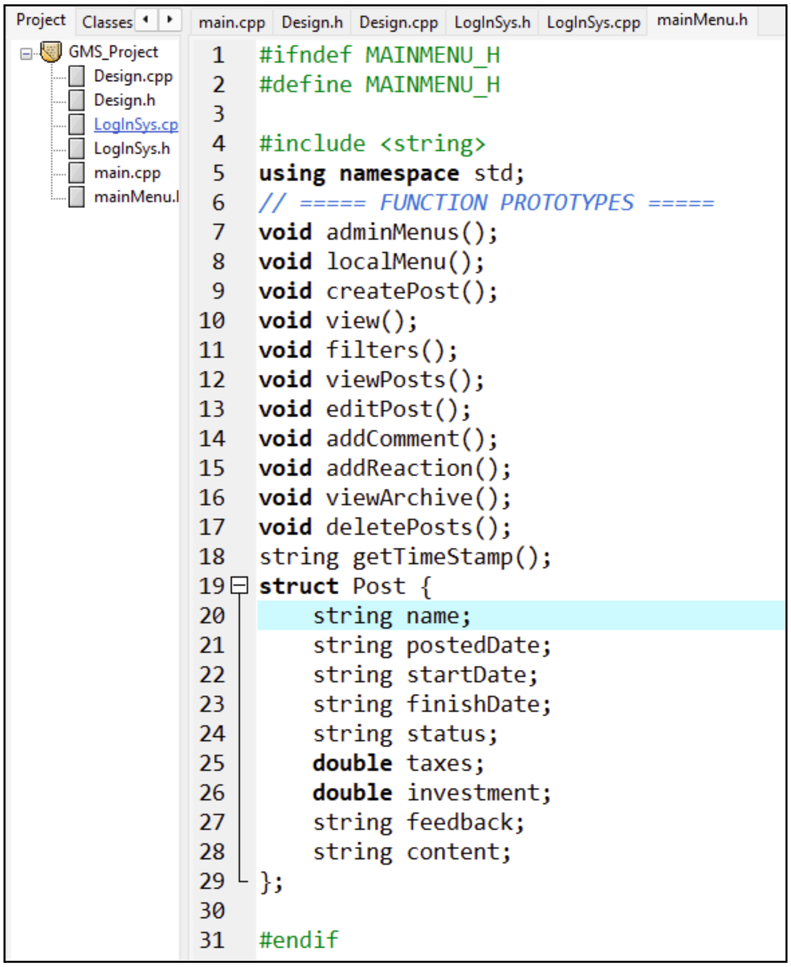
```cpp
}
bool checkUser(string user){ // Checks Username Dupe
    string storedUser, storedPass;
    ifstream file("credentials.txt");
    if(file.is_open()){
        while(file >> storedUser >> storedPass){
            if(storedUser == user){
                file.close(); return true;
            }
        }
        file.close();
    }
    return false;
}
bool login(){ // Login Definition
    string username, password, storedUser, storedPass;
    cout << "Welcome to the Log-in Menu!!" << endl; ms500();
    cout << "Please enter your username: "; cin >> username;
    cout << "Enter your password: "; cin >> password;
    ifstream file("credentials.txt");
    if(file.is_open()){
        while(file >> storedUser >> storedPass){
            if(storedUser == username && storedPass == password){
                system("cls"); loading(); txtAnimation("Log-in successful!", 10);
                file.close(); s2(); return true;
            }
        }
        file.close(); system("cls"); loading();
        txtAnimation("Invalid username or password! Please try again.", 10); s2();
        return false;
    }
}
```



```cpp
bool adminLogin(){ // Admin Log-in Definition
    string username, password, storedUser, storedPass;
    cout << "Welcome to the Admin Log-in Menu!!" << endl; ms500();
    cout << "Please enter your username: "; cin >> username;
    cout << "Enter your password: "; cin >> password;
    ifstream file("credentials.txt");
    if(file.is_open()){
        while(file >> storedUser >> storedPass){
            if(storedUser == username && storedPass == password){
                system("cls"); loading(); txtAnimation("Account identified!", 10);
                file.close(); s2(); return true;
            }
        }
        file.close(); system("cls"); loading();
        txtAnimation("Invaid username or password! Please try again.", 10); s2();
        return false;
    }
}
bool codeInter(){
    int IDcode, storedCode;
    cout << "Enter your ID Code: "; cin >> IDcode;
    ifstream code("IDcode.txt");
    if(code.is_open()){
        while(code >> storedCode){
            if(storedCode == IDcode){
                system("cls"); loading();
                txtAnimation("Admin code identified, Admin log-in successful!", 10);
                code.close(); s2(); return true;
            }
        }
        code.close(); system("cls"); loading();
        txtAnimation("Invalid ID code, Log-in terminated!", 10); s2();
        return false;
    }
}
```



```cpp
int loginSys(){ // Menu definition (Loops)
    int choice;
    while(choice != 4){
        system("cls");
        cout << "What would you like to do?" << endl;
        cout << "(1) Register\n(2) Login\n(3) Admin Log-in\n(4) Exit" << endl;
        cout << "Please input a number: "; cin >> choice;
        if(choice <= 0 || choice >= 5){
            txtAnimation("Invalid input, Please try again!", 10); s2(); continue;
        }
        else if(choice == 1){
            system("cls"); loading(); registration();
        }
        else if(choice == 2){
            system("cls"); loading();
            if(login()){
                system("cls"); loading(); localMenu();
            }
        }
        else if(choice == 3){
            system("cls"); loading();
            if(adminLogin()){
                system("cls");
                if(codeInter()){
                    system("cls"); loading(); adminMenus();
                }
            }
        }
        else{
            system("cls"); loading(); txtAnimation("Invalid input, Please try again!", 10);
            s2(); continue;
        }
    }
    system("cls"); txtAnimation("Thank you for using our system!!", 10); s2();
    system("cls"); txtAnimation(":D", 10);
    return 0;
}
```



```cpp
#ifndef MAINMENU_H
#define MAINMENU_H

#include <string>
using namespace std;
// ===== FUNCTION PROTOTYPES =====
void adminMenus();
void localMenu();
void createPost();
void view();
void filters();
void viewPosts();
void editPost();
void addComment();
void addReaction();
void viewArchive();
void deletePosts();
string getTimeStamp();
struct Post {
    string name;
    string postedDate;
    string startDate;
    string finishDate;
    string status;
    double taxes;
    double investment;
    string feedback;
    string content;
};

#endif
```