Khulna University of Engineering & Technology

PROJECT REPORT

Department: Electrical and Electronic Engineering

Course No: EE 1222

Name of the Project: **Hall Management System**

Remarks

Name : Rezoan Ahmed Riam

Roll No : 2203004

Group No : 02

Year : 1st

Term : 2nd

Date of Submission : 09/09/2024

Objectives :

1. To create smart hall management system.

2. To ensure digital security and access control.

3. To enhance automatic meal tracking.

4. To provide a user-friendly interface.

5. To store hall related information for students.

Introduction:

The Hall Management System is a user-friendly software application designed to help students easily manage their personal details and essential services associated with their stay in university halls. It streamlines several important tasks, such as logging in securely, viewing personal information, managing meal preferences, and checking payment dues. In addition, the system provides features to help students change their passwords or reset them if forgotten, making the process efficient. The goal of this system is to make it easier for students to manage their day-to-day hall-related tasks in a secure and convenient way.

The program code is made by some blocks which are described below:

1. Functions:

A function is a block of code that performs a specific task and can be reused multiple times throughout a program. Functions help make programs modular by dividing them into small, manageable parts.

Key Points:

- Functions can take parameters (inputs) and return values (outputs).

2. Strings:

A string in C is essentially an array of characters that is terminated by a null character ('\0'). Strings are used to store and manipulate text.

Key Points:

- Strings are represented as arrays of char.

- They can be processed using standard library functions .

3. Loops:

Loops allow a block of code to be executed repeatedly as long as a given condition is true. They are commonly used for iteration.

Types:

1. For Loop: Used when the number of iterations is known.

2. While Loop: Used when the number of iterations is not fixed.

3. Do-While Loop: Similar to while loop, but it ensures the loop runs at least once.

4. Conditional Statements:

Conditional statements (like if, else, and switch) control the flow of a program by executing code blocks based on specific conditions.

Key Points:

- They allow decision-making within the program.

- Different branches of code can be executed based on varying conditions.

5. Arrays:

An array is a collection of elements (all of the same data type) stored in contiguous memory locations. Arrays store multiple values of the same type under a single variable name.

Key Points:

- Arrays have a fixed size.

- Elements are accessed using an index, starting from 0.

This creates an array of integers and prints the first element.

6. Pointers:

A pointer is a variable that stores the memory address of another variable. Pointers are used for dynamic memory allocation, array manipulation, and passing data by reference to functions.

Key Points:

- Pointers hold addresses, not values.

- They are powerful for manipulating memory directly.

This prints the value of x using a pointer.

7. Structures:

A structure (or struct) is a user-defined data type in C that allows combining variables of different types under a single name. Structures are used to represent complex data like student records, employee details, etc.

Key Points:

- Structures group different data types together.

- They allow handling complex data in a more organized way.

-Structures help manage and organize complex data types, especially when dealing with multiple related variables.

Program for the project:

```
#include <stdio.h>
#include <string.h>

// Structure to hold user details
struct student {
    char username[50];
    char password[50];
    char name[100];
    char dateofbirth[50];
    char hometown[50];
    char bloodgroup[10];
    char religion[50];
    int mealStatus;
    float paymentDue;
    char securityQuestion[100];
    char securityAnswer[50];
    int hall;
} users[50];
// Function used
void selectHall(int *selectedHall);
int login(struct student users[], int numUsers, int selectedHall);
void showPersonalDetails(struct student users[], int currentIndex);
void toggleMealStatus(struct student users[], int currentIndex);
void showPaymentDetails(struct student users[], int currentIndex);
void logout(int *isLoggedIn);
void changePassword(struct student users[], int currentIndex);
void forgotPassword(struct student users[], int numUsers, int selectedHall);
int main() {
    struct student users[3] = {
        {"sadat", "2203001", "Md Sadat Hossain", "01/06/2003","Dinajpur","AB+","Islam",0, 1200, "What is your favourite colour?", "blue", -1},
        {"junayad", "2203003", "Md Junayad","30/09/2003","Dhaka","O+","Islam", 1, 1500, "What is your favourite colour?", "green", -1},
        {"riam", "2203004", "Rezoan Ahmed Riam","24/09/2003","Madaripur","AB+","Islam", 1, 1600, "What is your favourite colour?", "yellow", -1}
    };
    int numUsers = 3;
    int isLoggedIn = 0;
    int choice;
    int currentUserIndex = -1;
    int selectedHall = -1;
    printf("Welcome to the Hall Management System\n");
```

```
// Display the user to select a hall before login
selectHall(&selectedHall);


while (1) {
    if (isLoggedIn!=1) {
        printf("\n1. Login\n2. Forgot Password\nEnter your choice: ");
        scanf("%d", &choice);
        if (choice == 1) {
            currentUserIndex = login(users, numUsers, selectedHall);
            if (currentUserIndex != -1) {
                isLoggedIn = 1;
            }
        } else if (choice == 2) {
            forgotPassword(users, numUsers, selectedHall);
        } else {
            printf("Invalid choice. Please try again.\n");
        }
    } else {
        printf("\nMenu:\n");
        printf("1. Personal Details\n");
        printf("2. Meal On/Off\n");
        printf("3. Payment Details\n");
        printf("4. Change Password\n");
        printf("5. Logout\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```
            switch (choice) {
                case 1:
                    showPersonalDetails(users, currentUserIndex);
                    break;
                case 2:
                    toggleMealStatus(users, currentUserIndex);
                    break;
                case 3:
                    showPaymentDetails(users, currentUserIndex);
                    break;
                case 4:
                    changePassword(users, currentUserIndex);
                    break;
                case 5:
                    logout(&isLoggedIn);
                    currentUserIndex = -1;
                    break;
                default:
                    printf("Invalid choice. Please try again.\n");
                    break;
            }
        }
    }
    return 0;
}
```

```
void selectHall(int *selectedHall) {
    printf("\nSelect a Hall:\n");
    printf("1. FAZLUL HAQUE HALL\n");
    printf("2. BANGABANDHU HALL\n");
    printf("3. AMAR EKUSHEY HALL\n");
    printf("4. LALAN SHAH HALL\n");
    printf("5. KHAN JAHAN ALI HALL\n");
    printf("6. M A RASHID HALL\n");
    printf("7. ROKEYA HALL\n");
    printf("Enter your choice: ");
    int hallChoice;
    scanf("%d", &hallChoice);

    if (hallChoice >= 1 && hallChoice <= 7) {
        *selectedHall = hallChoice;
        char *hallNames[] = {
            "FAZLUL HAQUE HALL", "BANGABANDHU HALL", "AMAR EKUSHEY HALL",
            "LALAN SHAH HALL", "KHAN JAHAN ALI HALL", "M A RASHID HALL", "ROKEYA HALL"
        };
        printf("Welcome to %s\n", hallNames[hallChoice - 1]);
    } else {
        printf("Invalid choice. Try again.\n");
        selectHall(selectedHall);
    }
}
```

```
int login(struct student users[], int numUsers,int selectedhall) {
    char inputUsername[50];
    char inputPassword[50];
    printf("\nLogin\n");
    printf("Username: ");
    scanf("%s", inputUsername);
    printf("Password: ");
    scanf("%s", inputPassword);
    for (int i = 0; i < numUsers; i++) {
        if (strcmp(inputUsername, users[i].username) == 0 && strcmp(inputPassword, users[i].password) == 0 ) {
            printf("Login successful!\n");
            return i;
        }
    }
    printf("Invalid username or password. Try again.\n");
    return -1;
}
void showPersonalDetails(struct student users[], int currentIndex) {
    printf("\nPersonal Details:\n");
    printf("Name: %s\n", users[currentIndex].name);
    printf("Roll: %s\n",users[currentIndex].password);
    printf("Date of Birth: %s\n", users[currentIndex].dateofbirth);
    printf("Hometown: %s\n",users[currentIndex].hometown);
    printf("Blood Group: %s\n", users[currentIndex].bloodgroup);
    printf("Religion: %s\n",users[currentIndex].religion);
    printf("Meal status: %s\n", users[currentIndex].mealStatus ? "On" : "Off");
    printf("Payment due: Tk.%.2f\n", users[currentIndex].paymentDue);
}
void toggleMealStatus(struct student users[], int currentIndex) {
    printf("\nCurrent meal status: %s\n", users[currentIndex].mealStatus ? "On" : "Off");
    printf("Do you want to change it? (1 for Yes, 0 for No): ");
    int change;
    scanf("%d", &change);
    if (change == 1) {
        users[currentIndex].mealStatus = !users[currentIndex].mealStatus;
        printf("Meal status changed to: %s\n", users[currentIndex].mealStatus ? "On" :
"Off");
    } else {
        printf("No changes made.\n");
    }
}
void showPaymentDetails(struct student users[], int currentIndex) {
    printf("\nPayment Details:\n");
    printf("Payment due: Tk.%.2f\n", users[currentIndex].paymentDue);
}
void changePassword(struct student users[], int currentIndex) {
    char oldPassword[50];
    char newPassword[50];
    printf("\nChange Password\n");
    printf("Enter current password: ");
    scanf("%s", oldPassword);
    if (strcmp(oldPassword, users[currentIndex].password) == 0) {
        printf("Enter new password: ");
        scanf("%s", newPassword);
        strcpy(users[currentIndex].password, newPassword);
        printf("Password changed successfully.\n");
    } else {
        printf("Incorrect current password. Password change failed.\n");
    }
}
void forgotPassword(struct student users[], int numUsers, int selectedHall) {
    char inputUsername[50];
    char securityAnswer[50];
    char newPassword[50];
    printf("\nForgot Password\n");
    printf("Username: ");
    scanf("%s", inputUsername);
    for (int i = 0; i < numUsers; i++) {
        if (strcmp(inputUsername, users[i].username) == 0 ) {
            printf("%s: ", users[i].securityQuestion);
            scanf("%s", securityAnswer);

            if (strcmp(securityAnswer, users[i].securityAnswer) == 0) {
                printf("Enter new password: ");
                scanf("%s", newPassword);
                strcpy(users[i].password, newPassword);
                printf("Password reset successfully.\n");
            return;
         } else {
            printf("Incorrect answer. Password reset failed.\n");
            return;
         }
      }
   }
   printf("Username not found.\nPassword reset failed.\n");
}
void logout(int *isLoggedIn) {
   *isLoggedIn = 0;
   printf("Logged out successfully.\n");
}
```

Output :

```
Welcome to the Hall Management System
Select a Hall:
1. FAZLUL HAQUE HALL
2. BANGABANDHU HALL
3. AMAR EKUSHEY HALL
4. LALAN SHAH HALL
5. KHAN JAHAN ALI HALL
6. M A RASHID HALL
7. ROKEYA HALL
Enter your choice: 1
Welcome to FAZLUL HAQUE HALL
1. Login
```

```
2. Forgot Password
Enter your choice: 1
Login
Username: sadat
Password: 2203001
Login successful!
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
4. Change Password
5. Logout
Enter your choice: 1
Personal Details:
Name: Md Sadat Hossain
Roll: 2203001
Date of Birth: 01/06/2003
Hometown: Dinajpur
Blood Group: AB+
Religion: Islam
Meal status: Off
Payment due: Tk.1200.00
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
```

```
4. Change Password
5. Logout
Enter your choice: 2
Current meal status: Off
Do you want to change it? (1 for Yes, 0 for No): 1
Meal status changed to: On
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
4. Change Password
5. Logout
Enter your choice: 3
Payment Details:
Payment due: Tk.1200.00
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
4. Change Password
5. Logout
Enter your choice: 4
Change Password
Enter current password: 2203001
Enter new password: 2024sadat
Password changed successfully.
```

```
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
4. Change Password
5. Logout
Enter your choice: 5
Logged out successfully.
1. Login
2. Forgot Password
Enter your choice: 1

Login
Username: sadat
Password: 2203001
Invalid username or password. Try again.

1. Login
2. Forgot Password
Enter your choice: 1

Login
Username: sadat
Password: 2024sadat
Login successful!
```

```
Menu:
1. Personal Details
2. Meal On/Off
3. Payment Details
4. Change Password
5. Logout
Enter your choice: 5
Logged out successfully.

1. Login
2. Forgot Password
Enter your choice: 2

Forgot Password
Username: sadat
What is your favourite colour?: blue
Enter new password: 2024sadat001
Password reset successfully.

1. Login
2. Forgot Password
Enter your choice: 1

Login
Username: sadat
```

```
Password: 2024sadat001

Login successful!
```

Discussion :

In this project, the program of hall management system was introduced. Various types of blocks of code were used in this program. Conditions were used to express the condition to evaluate true or false. Loops were used to repeat a block of code multiple times until a condition is met. Arrays and strings were used to form variables. Functions were used to perform a specific task separately besides the main function. Other blocks were also used to make this program.

This program has a simple, text-based interface that is easy to use. Hall selection, meal status toggle and payment due functionalities would be helpful to manage hall accommodations. Security question provides basic protection for password recovery. This program is not designed to handle many users, but it could be modified to do so.

Conclusion :

This system can be developed into a fully functional hall management system. By extending some features like transaction for payment and implementing admin controls, this could become a full-fledged application. It can be significantly improved in terms of security, scalability and user interface.

References :

1. Programming in ANSI C by E. Balagurusamy
2. https://www.programmiz.com/c-programming/c-structure-functions/