Project Report

On

IMPLEMENT WIRELSESS ACCESS SYSTEM

Submitted in partial fulfilment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

by

Ms. R ASHRITHA(22WH1A6626)

Ms. M SAATVIKA (22WH1A6628)

Ms. M NITHYA SRI(22WH1A6630)

Ms. B RISHITA(22WH1A6631)

Under the esteemed guidance of

Ms. P Anusha

Assistant Professor, CSE(AI&ML)

BVRIT HYDERABAD College of Engineering for Women

(UGC Autonomous Institution | Approved by AICTE | Affiliated to JNTUH)

(NAAC Accredited - A Grade | NBA Accredited B.Tech. (EEE, ECE, CSE and IT)

Bachupally, Hyderabad – 500090

2024-25

ABSTRACT

This program implements a foundational Wireless Access System Server designed to authenticate users and securely log their connections. Its primary goal is to provide a simple yet effective method of managing and monitoring user access to wireless resources. By leveraging a predefined set of credentials stored in memory, the system validates user access requests and ensures that only authorized individuals gain entry.

The program is built around a straightforward command-line interface (CLI) that allows users to input their credentials for authentication. The system operates with minimal configuration, making it a lightweight and portable solution for basic wireless access management.

1. Initialization
    - During startup, the system preloads a predefined list of usernames and passwords into memory. This setup ensures rapid credential validation and eliminates the need for external databases, improving efficiency for small-scale implementations.
2. Authentication
    - User credentials are validated through a robust string comparison mechanism. Upon input of a username and password, the system checks the combination against the stored records, ensuring accurate verification.
3. Logging
    - All successful login attempts are logged into a file with detailed information, including the timestamp, username, and connection details. This feature enhances traceability and allows administrators to audit connection histories effectively.

PROBLEM STATEMENT

Create a C program for a basic Wireless Access System that authenticates users with predefined username-password pairs. The program will prompt users to input their credentials and validate them against the hardcoded set. If the credentials are valid, the program grants access; otherwise, it denies access and allows the user to retry until they enter correct credentials.

Upon successful authentication, the program will log the username along with the current timestamp into a file named connections.log. This logging ensures traceability and accountability for each connection. The log file must be updated reliably, and any issues with file operations, such as inability to open or write to the file, should be handled gracefully, with appropriate error messages displayed to the user.

The program must provide clear feedback to users. For example, successful authentication should display a message like "Login successful!" while failed attempts should show "Invalid credentials. Please try again." The program should not expose sensitive information, such as the list of valid credentials, through error messages or debugging output.

The program will terminate only after a successful login and completion of the logging process. It should handle resources properly, such as closing files and freeing memory, to ensure a clean exit. This system highlights basic user authentication principles and emphasizes secure, reliable handling of credentials and log**s.**

FUNCTIONAL REQUIREMENTS

1. User Authentication:

The system must authenticate users by comparing the provided username and password with a predefined list of valid credentials.

2. Access Control:

The system should deny access for invalid username-password combinations and allow the user to try again.

3. Connection Logging:

After a successful login, the system must log the username and timestamp of the connection to a file (connections.log).

4. System Exit:

The program must exit after successfully logging the connection for the first valid user login.

5. User Feedback:

The system must provide feedback messages for:

- Successful login: "Access granted" message.
- Failed login: "Access denied" message.

6. Error Handling:

The system should handle file operations (reading, logging) and provide appropriate error messages if files cannot be accessed or created.

NON FUNCTIONAL REQUIREMENTS

1. **Usability:**

The system should provide a simple and user-friendly command-line interface with clear prompts for user input and error messages.

2. **Performance:**

The system should respond quickly to user input and authenticate users in real time without significant delays.

3. **Reliability:**

The program should function consistently without crashes or unexpected behavior during normal operation, even if the user enters incorrect credentials multiple times.

4. **Security:**

The system should store credentials securely in the code and not expose sensitive data. It should prevent unauthorized access and log all successful connections.

5. **Maintainability:**

The code should be clean, well-structured, and easy to modify, including adding more users or changing functionality, such as switching to a different method of storing credentials.

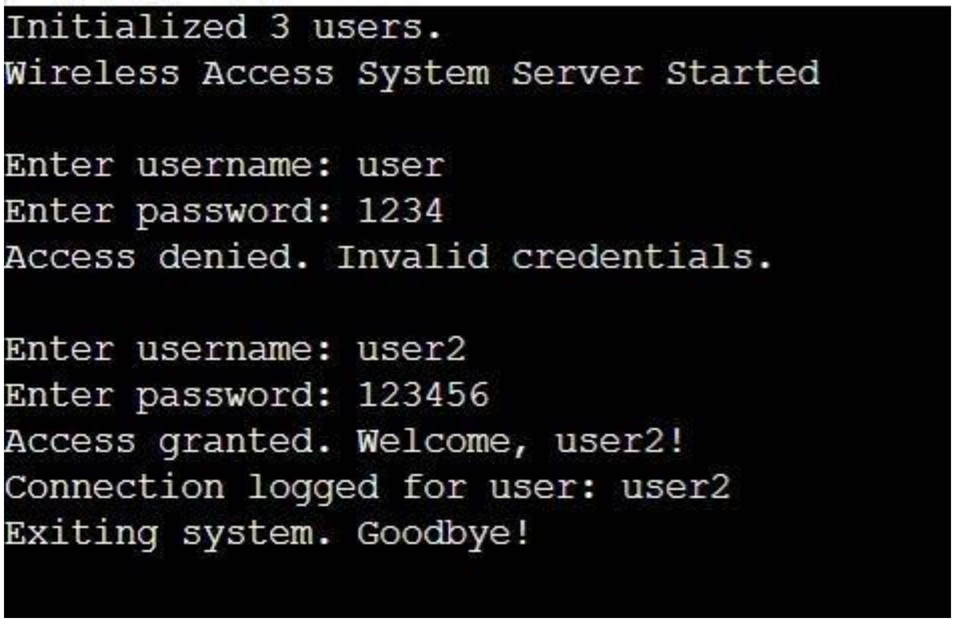
6. **Portability:**

The system should be portable and able to run on standard C compilers across various platforms with minimal configuration.

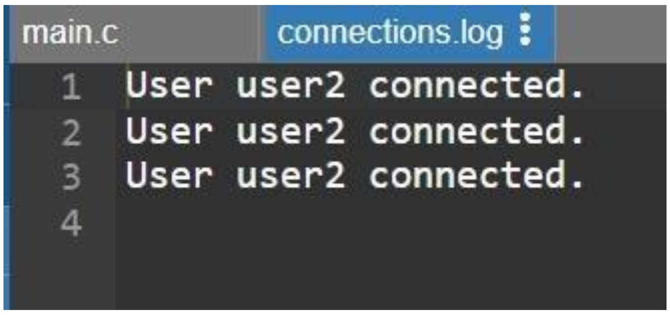
SOURCE CODE

```
#include <stdio.h>

 #include <stdlib.h>

#include <string.h>
#include <unistd.h>

#define MAX_USERS 100
#define USERNAME_LENGTH 50
#define PASSWORD_LENGTH 50

// Structure to store user credentials typedef struct
{

char username[USERNAME_LENGTH];

char password[PASSWORD_LENGTH];

}
 User;

// Array to hold registered users
User users[MAX_USERS];
int user_count = 0;

// Function to initialize users directly in code void
initialize_users()

{

strcpy(users[0].username, "admin");
strcpy(users[0].password, "password123");
strcpy(users[1].username, "user1");
strcpy(users[1].password, "passw0rd");
strcpy(users[2].username, "user2");
strcpy(users[2].password, "123456");
user_count = 3; // Number of initialized users
printf("Initialized %d users.\n", user_count);
}

// Function to authenticate a user
int authenticate_user(const char *username, const char *password)
 {
for (int i = 0; i < user_count; i++)
{
if (strcmp(users[i].username, username) == 0 &&
strcmp(users[i].password,password) == 0) {
return 1;
// Authentication successful
        }
    }
    return 0;
 // Authentication failed
}

// Function to log connected users void
log_connection(const char *username) {
FILE *logfile = fopen("connections.log", "a");
 if (!logfile) {
        perror("Error opening log file");

return;

    }
    fprintf(logfile, "User %s connected.\n", username);

 fclose(logfile);

    printf("Connection logged for user: %s\n", username);
}
// Main server loop void
start_server() {

Char
username[USERNAME_LE
NGTH];

char
password[PASSWORD_LE
NGTH];

 printf("Wireless Access System Server Started\n");
  while (1) {
printf("\nEnter username: ");

scanf("%s", username);

printf("Enter password: ");

scanf("%s", password);


if (authenticate_user(username, password)) {

 printf("Access granted. Welcome, %s!\n", username);
log_connection(username);

printf("Exiting system. Goodbye!\n");
exit(0); // Exit after successful connection

        }
else {
            printf("Access denied. Invalid credentials.\n");
        }
    }
}

int main() {
    // Initialize registered users
initialize_users();

start_server();

return 0;

}
```

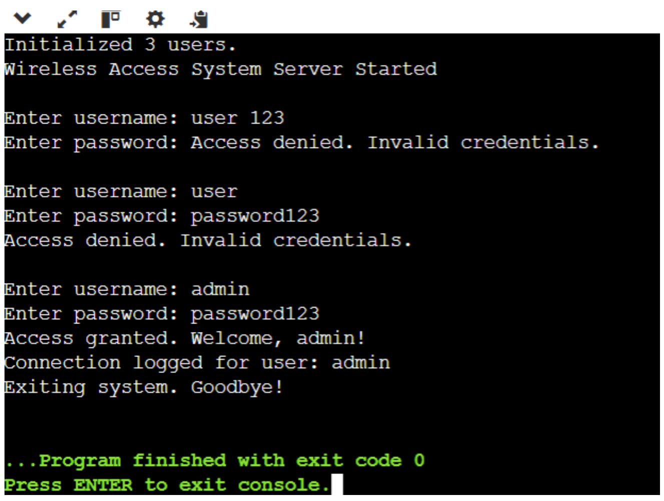
OUTPUT



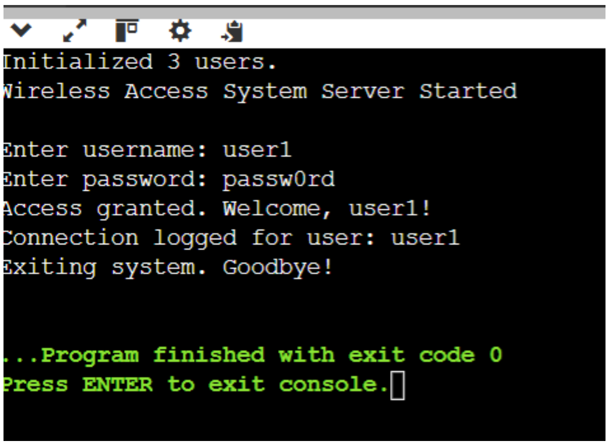
```
Initialized 3 users.
Wireless Access System Server Started

Enter username: user
Enter password: 1234
Access denied. Invalid credentials.

Enter username: user2
Enter password: 123456
Access granted. Welcome, user2!
Connection logged for user: user2
Exiting system. Goodbye!
```



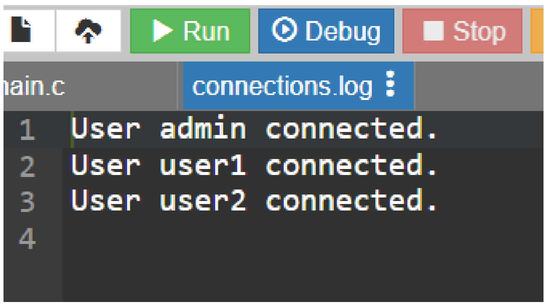
```
User user2 connected.
User user2 connected.
User user2 connected.
```



```
Initialized 3 users.
Wireless Access System Server Started

Enter username: user 123
Enter password: Access denied. Invalid credentials.

Enter username: user
Enter password: password123
Access denied. Invalid credentials.

Enter username: admin
Enter password: password123
Access granted. Welcome, admin!
Connection logged for user: admin
Exiting system. Goodbye!


...Program finished with exit code 0
Press ENTER to exit console.
```



```
Initialized 3 users.
Wireless Access System Server Started

Enter username: user1
Enter password: passw0rd
Access granted. Welcome, user1!
Connection logged for user: user1
Exiting system. Goodbye!


...Program finished with exit code 0
Press ENTER to exit console.
```



```
User admin connected.
User user1 connected.
User user2 connected.
```