**Restaurant Management System**

**Part:01(First)**

**ID: 232-35-667**

**How the program works:**

* **Registration**:
  1. The user enters their desired username and password.
  2. The program checks if the username already exists in the users.txt file.
  3. If it doesn’t exist, it writes the new username and password into the file.
* **Login**:
  1. The user enters their username and password.
  2. The program checks the users.txt file for the matching credentials.
  3. If the credentials match, the user is logged in; otherwise, the login fails.

**Registration And Login:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

char username[30];

char password[30];

} User;

void registerUser();

int loginUser();

int userExists(const char\* username);

void addUserToFile(User newUser);

int main() {

int choice;

printf("Restaurant Management System\n");

printf("1. Register\n");

printf("2. Login\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

registerUser();

break;

case 2:

if (loginUser()) {

printf("Login successful! Welcome to the Restaurant Management System.\n");

} else {

printf("Invalid credentials. Please try again.\n");

}

break;

case 3:

printf("Exiting... Goodbye!\n");

break;

default:

printf("Invalid choice. Please try again.\n");

break;

}

return 0;

}

void registerUser() {

User newUser;

FILE\* file;

printf("\nEnter username: ");

scanf("%s", newUser.username);

if (userExists(newUser.username)) {

printf("Username already exists. Try a different one.\n");

return;

}

printf("Enter password: ");

scanf("%s", newUser.password);

addUserToFile(newUser);

printf("Registration successful!\n");

}

int loginUser() {

User user;

FILE\* file;

char username[30];

char password[30];

int found = 0;

printf("\nEnter username: ");

scanf("%s", username);

printf("Enter password: ");

scanf("%s", password);

file = fopen("users.txt", "r");

if (file == NULL) {

printf("No users registered yet.\n");

return 0;

}

while (fscanf(file, "%s %s", user.username, user.password) != EOF) {

if (strcmp(username, user.username) == 0 && strcmp(password, user.password) == 0) {

found = 1;

break;

}

}

fclose(file);

return found;

}

int userExists(const char\* username) {

User user;

FILE\* file;

file = fopen("users.txt", "r");

if (file == NULL) {

return 0; // No users registered yet

}

while (fscanf(file, "%s %s", user.username, user.password) != EOF) {

if (strcmp(username, user.username) == 0) {

fclose(file);

return 1; // Username already exists

}

}

fclose(file);

return 0; // Username does not exist

}

void addUserToFile(User newUser) {

FILE\* file;

file = fopen("users.txt", "a");

if (file == NULL) {

printf("Error opening file for writing.\n");

return;

}

fprintf(file, "%s %s\n", newUser.username, newUser.password);

fclose(file);

}

**Output:**

Restaurant Management System

1. Register

2. Login

3. Exit

Enter your choice: 1

Enter username: john

Username already exists. Try a different one.

Enter username: alice

Enter password: mypassword

Registration successful!

Restaurant Management System

1. Register

2. Login

3. Exit

Enter your choice: 2

Enter username: alice

Enter password: mypassword

Login successful! Welcome to the Restaurant Management System.

**Restaurant Management System**

**Part:01(Updated Part)**

**ID: 232-35-100**

**How the program works:**

* **Registration**:
  1. The user enters their desired username, password, phone number.
  2. The program checks if the username already exists in the users.txt file.
  3. If it doesn’t exist, it writes the new username, password, phone number into the file.
* **Login**:
  1. The user enters their username, password, phone number.
  2. The program checks the users.txt file for the matching credentials.
  3. If the credentials match, the user is logged in; otherwise, the login fails.

**Registration And Login(code):**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define FILENAME "users.txt"

typedef struct {

char username[30];

char password[50]; // Increase the buffer size for password

char phoneNumber[15]; // Add a phone number field

} User;

void registerUser();

int loginUser();

int userExists(const char\* username);

int findUserInFile(const char\* username, const char\* password, const char\* phoneNumber, User\* user);

void addUserToFile(User newUser);

int main() {

int choice;

// Start a loop that will continuously prompt the user until they choose to exit

while (1) {

printf("\nRestaurant Management System\n");

printf("1. Register\n");

printf("2. Login\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

getchar(); // Consume the newline character left by scanf

switch(choice) {

case 1:

registerUser();

break;

case 2:

if (loginUser()) {

printf("Login successful! Welcome to the Restaurant Management System.\n");

} else {

printf("Invalid credentials. Please try again.\n");

}

break;

case 3:

printf("Exiting... Goodbye!\n");

return 0; // Exit the program when the user chooses option 3

default:

printf("Invalid choice. Please try again.\n");

break;

}

}

return 0;

}

void registerUser() {

User newUser;

// Prompt the user for username, password, and phone number

printf("\nEnter username: ");

scanf("%s", newUser.username);

getchar(); // Consume the newline character left by scanf

// Check if the username already exists

if (userExists(newUser.username)) {

printf("Username already exists. Try a different one.\n");

return;

}

// Prompt the user for password using fgets to allow spaces

printf("Enter password: ");

fgets(newUser.password, sizeof(newUser.password), stdin);

newUser.password[strcspn(newUser.password, "\n")] = '\0'; // Remove the trailing newline character

// Prompt the user for phone number

printf("Enter phone number: ");

fgets(newUser.phoneNumber, sizeof(newUser.phoneNumber), stdin);

newUser.phoneNumber[strcspn(newUser.phoneNumber, "\n")] = '\0'; // Remove trailing newline if present

// Add user to the file

addUserToFile(newUser);

printf("Registration successful!\n");

}

int loginUser() {

User user;

char username[30], password[50], phoneNumber[15];

// Prompt the user for login credentials

printf("\nEnter username: ");

scanf("%s", username);

getchar(); // Consume the newline character left by scanf

// Prompt the user for password using fgets

printf("Enter password: ");

fgets(password, sizeof(password), stdin);

password[strcspn(password, "\n")] = '\0'; // Remove the trailing newline character

// Prompt the user for phone number

printf("Enter phone number: ");

fgets(phoneNumber, sizeof(phoneNumber), stdin);

phoneNumber[strcspn(phoneNumber, "\n")] = '\0'; // Remove trailing newline if present

// Check if the entered credentials match any entry in the file

if (findUserInFile(username, password, phoneNumber, &user)) {

return 1; // Login successful

}

return 0; // Login failed

}

int userExists(const char\* username) {

User user;

return findUserInFile(username, NULL, NULL, &user);

}

int findUserInFile(const char\* username, const char\* password, const char\* phoneNumber, User\* user) {

FILE\* file = fopen(FILENAME, "r");

if (!file) {

return 0; // File doesn't exist or couldn't be opened

}

while (fscanf(file, "%s %s %s", user->username, user->password, user->phoneNumber) != EOF) {

// Check if username matches

if (strcmp(username, user->username) == 0) {

// If password and phone number are provided, check if they match as well

if (password != NULL && phoneNumber != NULL) {

if (strcmp(password, user->password) == 0 && strcmp(phoneNumber, user->phoneNumber) == 0) {

fclose(file);

return 1; // Credentials match

}

} else {

// Just check if the username exists

fclose(file);

return 1; // Username found

}

}

}

fclose(file);

return 0; // No matching user found

}

void addUserToFile(User newUser) {

FILE\* file = fopen(FILENAME, "a");

if (!file) {

printf("Error opening file for writing.\n");

return;

}

// Write username, password, and phone number to the file

fprintf(file, "%s %s %s\n", newUser.username, newUser.password, newUser.phoneNumber);

fclose(file);

}

**Output:**

Restaurant Management System

1. Register

2. Login

3. Exit

Enter your choice: 1

Enter username: johndoe

Enter password: password123

Enter phone number: 123-456-7890

Registration successful!

Restaurant Management System

1. Register

2. Login

3. Exit

Enter your choice: 2

Enter username: johndoe

Enter password: password123

Enter phone number: 123-456-7890

Login successful! Welcome to the Restaurant Management System.

Restaurant Management System

1. Register

2. Login

3. Exit

Enter your choice: 3

Exiting... Goodbye!