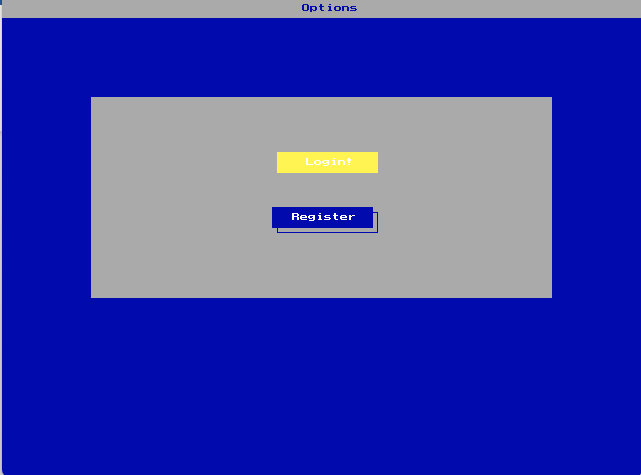
**Namaz management system**

|  |  |
| --- | --- |
| No. | Contents of application |
| 1 | Login/Register Page |
| 2 | User’s application panel |
| 3 | CMS/admin panel |
| 4 | Namaz assigning tab |
| 5 | Namaz De-assigning tab |
| 6 | Namaz searching tab |
| 7 | Delete user tab |
| 8 | List of namaz |
| 9 | List of users |
| 10 | User self-de-assignation |
| 11 | Error page |
| 12 | Success page |

**Login/Register page: -**



**cONTROLS: -**

By pressing the ‘s’ key the active option will change. If there is yellow background color on the button it means that is active option. If you want to go to that option you will have to press “**ENTER**”. By pressing enter you either will go to the login page or registration page. According to your choice.

**LOGIN: -**

In login page you will be asked username and password. If you are administrator then username will be **admin** and password will be **admin123.** Admin page is only for that person who wants to have full authority on that application. Usually the owner of that application have credentials. If the user is not an admin then first user must register himself first in order to login. If he is already registered then he will have to write his credentials.

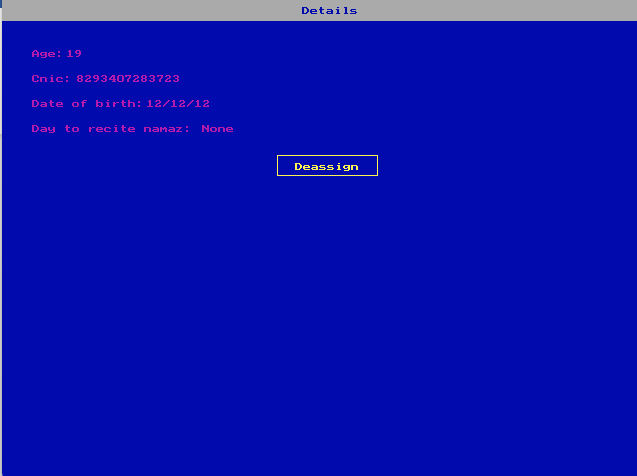
**register: -**

If the user is not registered. Then he will have to give these information:

1. Name
2. CNIC
3. Password
4. Date of birth
5. Age

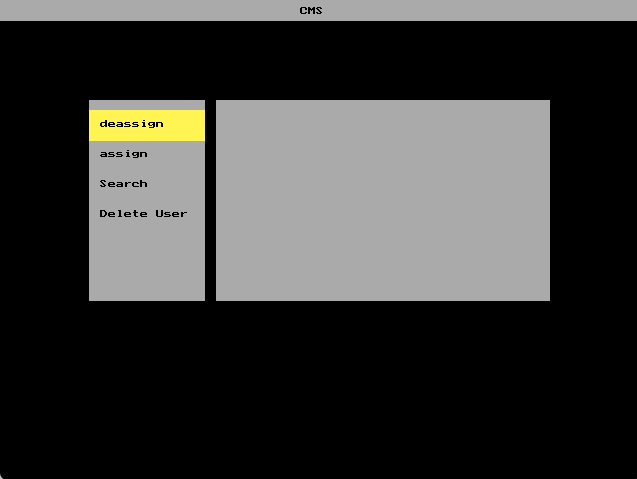
By giving this information his account will be created. And not person will be able to login into his account.

**user’s application panel: -**

**: -**

After entering his credentials this page will appear to the user. His entire information will be shown to him. And in the last information **“Day to recite namaz”.** There will be a specific namaz which he will recite on that specific day. If there is none written. Then it means no namaz is assigned to that person yet. If one namaz is assigned to that person it means only one namaz is assigned to him. One person can only have two namazes(Prayers).

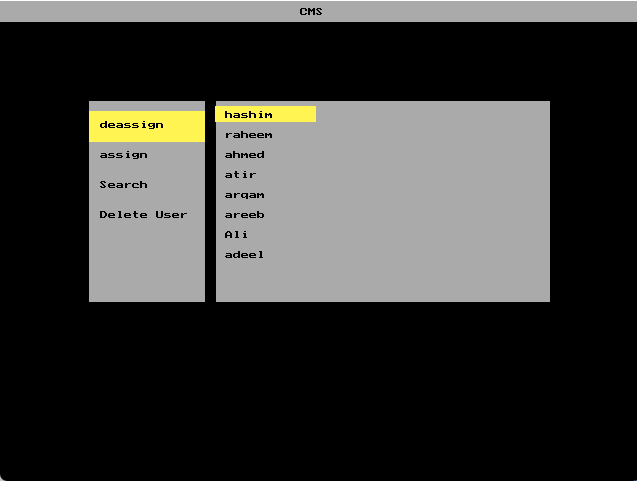
**CMS/admin panel: -**



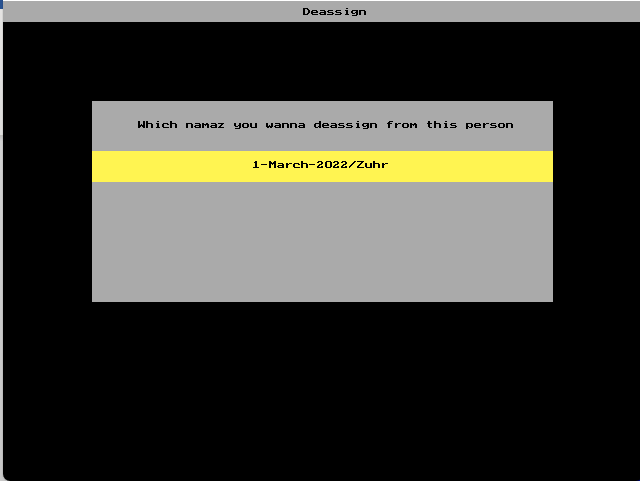
This is the page where user will control the entire application. On the left side there are multiple options where admin wants to go. By pressing ‘s’ the active option will change continuously. Yellow highlighter indicates that admin is currently on that option of admin presses “**ENTER**”. He will be redirected to that option functionality. There are multiple options such as:

1. **De-assign:** In this option namaz can be de-assigned from specific person
2. **Assign:** In this option namaz can be assigned to that specific person
3. **Search:** In this option. You will be able to search that who will recite namaz on that specific day. The namaz of the prayer will appear as a result
4. **Delete-user:** In this option you can delete users.

**Namaz de-assigning tab: -**



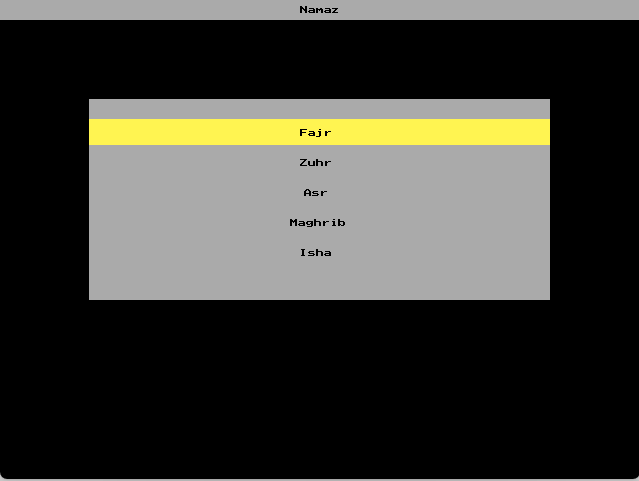
By pressing enter on that option. The list of the user will appear. You can change user by pressing ‘s’. If admin wants to assign namaz to that person he just have to press “**ENTER”.** By pressing enter you will be asked what namaz you want to de-assign from this person.



The option will look like this. In this page you are being asked what namaz you want to de-assign from him. There can be two options too. But in this case there is one option. By pressing enter on the specific option. The namaz will be de-assigned from that person immediately.

**Namaz assigning tab: -**

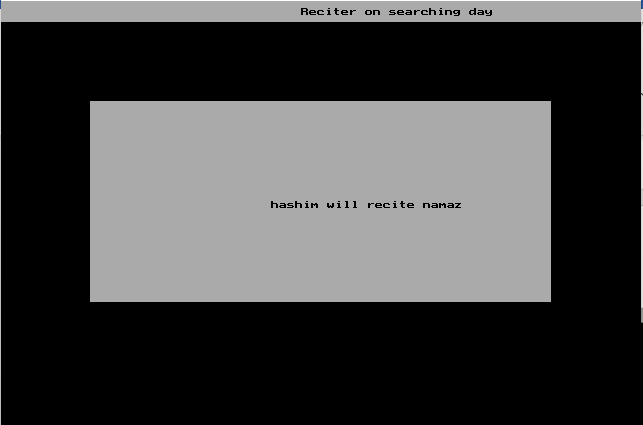
This tab is the opposite of the de-assign tab. In the tab you will decide what namaz you wan to assign to the specific person (**Only two namaz can be assigned per person. Otherwise it’ll show error message**). Front of the assign page is similar to the de-assign page. After choosing a user to assign namaz. You will be asked which namaz he will recite. It can be **Fajr. Zuhr, Asr, Maghrib and Isha**.



By choosing any namaz you will be asked in which day you wanna assign namaz. A text field will appear on that page where you have to write day. In case if you write the alphabet mistakenly on that page. Then you can de-assign that namaz whenever you want. It will appear like this **a-March-2022/Zuhr.** There will not be any bug by entering the character/string. After doing all of the process namaz will be assigned to that person. And that person will be able to see his recitation days on the user’s panel.

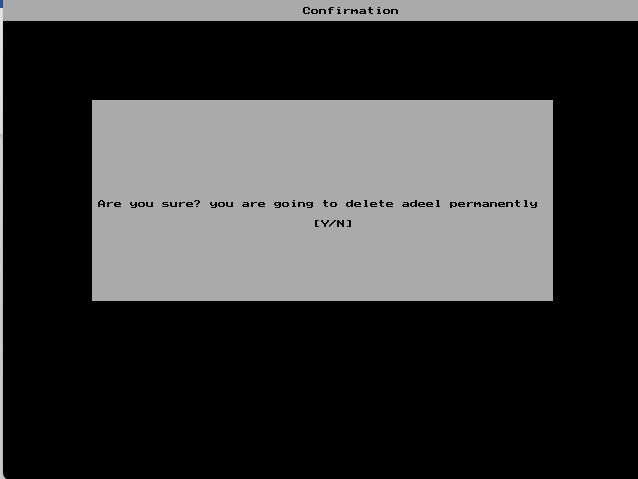
**Namaz searching tab: -**

This page is useful for that moment if the administrator forgets who is going to recite namaz on that day. After entering into this tab you will be asked what day you are searching for. Then after list of namaz will appear means which namaz you are searching. After getting all of this information. The box will appear where it will be written “**hashim will recite namaz**”. If no one is assigned to that day in specific namaz then it will show “**No one will recite namaz**”.



**delete user tab: -**

After entering to this option. List of the users will appear. If you choose any of them then it will ask again when he really wants to delete that person or not? By pressing ‘**y’** Key that person will be deleted. And by pressing ‘**n’** that person won’t be deleted and will redirected to previous page.

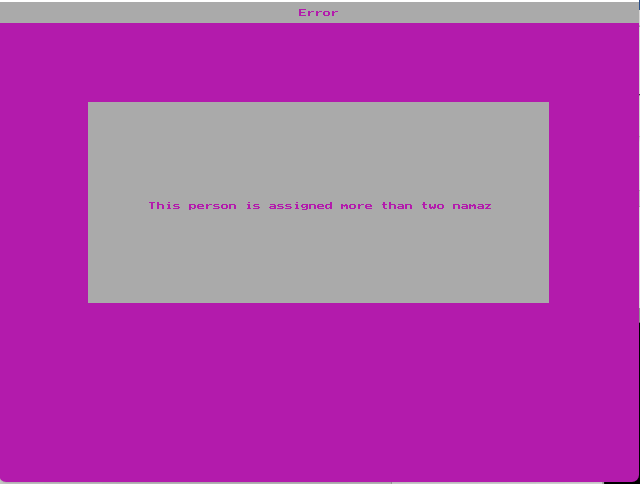


**user self de-assignation: -**

When administrator assigns namaz to the specific person. And that person don’t feel satisfy to be assigned on that day. Or incase of an emergency he can de-assign himself. By pressing ‘**D’** on the user’s application panel. By pressing ‘**D’** user will have to write excuse. Why he wants to de-assign himself. After that he will be asked which namaz you want to de-assign yourself. After choosing the specific namaz that user’s namaz will be de-assigned immediately.

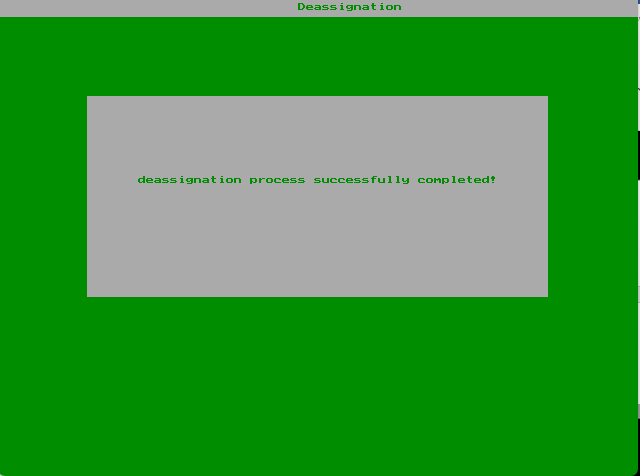
**error page: -**

Error page may appear for the multiple reasons:

1. User is already assigned for the two namaz
2. In de-assignation page if person is not assigned yet and admin enter to that user name then it will show that **”No namaz is assigned to that person yet”**
3. 

**success page: -**

This page appears whenever assignation, deassignation, delete user completes successfully. Then it shows the success page.



**code: -**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#include<graphics.h>

#include<dos.h>

// Function prototypes and global variables

void dialogueBox();

void header(char title[10], int bkColor);

void regSample(char title[10]);

void cms();

int files;

int count = 0;

char buffer2[50];

int loopCounter;

int success(char message[50], char successMessage[50]);

char dessUsername[50];

int tempInt;

char dessCnic[50];

char dessDob[50];

char dessAge[10];

int distance = 0;

void active(int optNum);

void cmsRegSample(char title[10]);

void options();

void userList(int distance);

void namazOpt(int namazOption);

int check(int id);

int assignNamaz(char day[10], char namaz[10], int id);

void deassignOption(int distance);

int searching(int quantityFiles, char searchedFor[50]);

void main() {

// Variable Declarations

FILE \*ptr = NULL;

char namaz[50];

int activeList = 0;

int namazOption = 0;

char sample[50];

char day[10];

char temp[50];

char confirm;

char nameOfUser[50];

char deletePerson[50];

int fileQuantity;

char secondRecitation[50];

char directory[50] = "C:\\TURBOC3\\FILES\\";

char currentMonth[50];

int gd = DETECT, gm, a, status;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

char action, files;

int opt = 1, result;

int fileChar;

int adminOpt = 0;

char buffer[50];

char path[50] = "C:\\TURBOC3\\FILES\\";

char userAge[50];

char execuses[50];

char currentYear[10];

char adminUsername[50];

int distance = 0;

int date;

char username[50], cnic[50], password[50], dob[50];

int age;

int quantity;

char nm[50];

char fileName[50];

int fileNumber;

char registrationName[50];

char deassignation2[50];

char deassignation[50];

char loginName[50];

char loginPass[50];

char data[50];

int error, i;

char loginNm[50];

char userPass[50];

char garbageData[50];

char userDob[50];

char userCnic[50];

char recitation[50];

char userAction;

// User interface of login panel

top:

header("Options", 1);

dialogueBox();

if(opt == 1) {

setcolor(14);

setfillstyle(1, 14);

line(270 + 5, 150 + 5, 370 + 5, 150+ 5);

line(370 + 5, 150 + 5, 370 + 5, 170 + 5);

line(370 + 5, 170 + 5, 270 + 5, 170 + 5);

line(270 + 5, 170 + 5, 270 + 5, 150 + 5);

floodfill(271 + 5, 151 + 5, 14);

} else {

setcolor(0);

setfillstyle(1, 0);

line(270, 150, 370, 150);

line(370, 150, 370, 170);

line(370, 170, 270, 170);

line(270, 170, 270, 150);

floodfill(271, 151, 0);

}

if(opt != 1) {

setcolor(0);

setfillstyle(1, 0);

line(270 + 5, 150 + 5, 370 + 5, 150+ 5);

line(370 + 5, 150 + 5, 370 + 5, 170 + 5);

line(370 + 5, 170 + 5, 270 + 5, 170 + 5);

line(270 + 5, 170 + 5, 270 + 5, 150 + 5);

floodfill(271, 151, 0);

}

if(opt == 2) {

setcolor(14);

setfillstyle(1, 14);

line(270 + 5, 155 + 60, 375, 155 + 60);

line(370 + 5, 155 + 60, 375, 175 + 60);

line(370 + 5, 175 + 60, 275, 175 + 60);

line(270 + 5, 175 + 60, 275, 155 + 60);

floodfill(276, 155 + 61, 14);

} else {

setcolor(0);

setfillstyle(1, 0);

line(270, 150 + 60, 370, 150 + 60);

line(370, 150 + 60, 370, 170 + 60);

line(370, 170 + 60, 270, 170 + 60);

line(270, 170 + 60, 270, 150 + 60);

floodfill(271, 151 + 60, 0);

}

if(opt != 2) {

setcolor(0);

setfillstyle(1, 0);

line(270 + 5, 150 + 65, 370 + 5, 150 + 65);

line(370 + 5, 150 + 65, 370 + 5, 170 + 65);

line(370 + 5, 170 + 65, 270 + 5, 170 + 65);

line(270 + 5, 170 + 65, 270 + 5, 150 + 65);

floodfill(271, 151 + 60, 0);

}

setcolor(15);

if(opt != 2) {

outtextxy(270 + 20, 150 + 66, "Register");

} else {

outtextxy(270 + 25, 150 + 71, "Register");

}

if(opt != 1) {

outtextxy(270 + 29, 150 + 6, "Login!");

} else {

outtextxy(270 + 34, 150 + 11, "Login!");

}

action = getch();

if((opt == 1) && (action == 13)) {

goto login;

} else if((opt == 2) && (action == 13)){

goto registration;

} else {

printf("something bad happened");

}

if(action == 's' || action == 'S') {

if(opt == 2) {

opt = 1;

} else {

opt++;

}

cleardevice();

goto top;

}

login:

tryAgain:

cleardevice();

header("Login", 1);

dialogueBox();

setcolor(4);

if(error == 1)

outtextxy(230, 130, "Wrong username or password");

regSample("Username");

gets(loginName);

strcpy(loginNm, loginName);

ptr = fopen(strcat(directory, strcat(loginName, ".txt")), "r");

if(ptr == 0) {

if(strcmp(loginNm, "admin") == 0) {

cleardevice();

header("Admin", 1);

dialogueBox();

regSample("Password");

gets(loginPass);

if(strcmp(loginPass, "admin123") == 0)

goto admin;

else

exit(0);

} else {

exit(0);

}

} else {

cleardevice();

header(loginNm, 1);

dialogueBox();

regSample("Password");

gets(loginPass);

for(i = 0; i < 3; i++) {

if(i == 1) {

fscanf(ptr, "%s", &userPass);

} else {

fscanf(ptr, "%s", &garbageData);

}

}

if(strcmp(loginPass, userPass) == 0) {

fclose(ptr);

goto user;

} else

exit(0);

}

getch();

exit(0);

// Registration panel

registration:

cleardevice();

header("Registration", 1);

dialogueBox();

regSample("Username");

gets(username);

cleardevice();

header("Registration", 1);

dialogueBox();

regSample("CNIC");

gets(cnic);

cleardevice();

header("Registration", 1);

dialogueBox();

regSample("Password");

gets(password);

cleardevice();

header("Registration", 1);

dialogueBox();

regSample("DOB");

gets(dob);

cleardevice();

header("Registration", 1);

dialogueBox();

regSample("Age");

scanf("%d", &age);

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

if(!ptr) {

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "w");

fprintf(ptr, "%d", 1);

fclose(ptr);

} else {

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &files);

files++;

fclose(ptr);

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "w");

fprintf(ptr, "%d", files);

fclose(ptr);

}

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &fileChar);

fclose(ptr);

strcpy(nm, username);

ptr = fopen(strcat(path, strcat(username, ".txt")), "w");

fprintf(ptr, "%s\n", nm);

fprintf(ptr, "%s\n", password);

fprintf(ptr, "%d", fileChar);

fprintf(ptr, "%s\n", ".txt");

fclose(ptr);

getch();

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &quantity);

fclose(ptr);

sprintf(buffer, "%d", quantity);

ptr = fopen(strcat("C:\\TURBOC3\\FILES\\", strcat(buffer, ".txt")),"w");

fprintf(ptr, "%s\n", nm);

fprintf(ptr, "%s\n", cnic);

fprintf(ptr, "%s\n", dob);

fprintf(ptr, "%d\n", age);

fclose(ptr);

exit(0);

// User view panel

user:

cleardevice();

header("Details", 1);

ptr = fopen(strcat("C:\\TURBOC3\\FILES\\", strcat(loginNm, ".txt")), "r");

for(i = 0; i < 3; i++) {

if(i == 1)

fscanf(ptr, "%s", garbageData);

}

strcpy(fileName, garbageData);

fclose(ptr);

strcpy(directory, "C:\\TURBOC3\\FILES\\");

ptr = fopen(strcat(directory, garbageData), "r");

strcpy(garbageData, NULL);

strcpy(userCnic, NULL);

strcpy(userDob, NULL);

strcpy(userAge, NULL);

strcpy(recitation, NULL);

strcpy(secondRecitation, NULL);

fscanf(ptr, "%s", &garbageData);

fscanf(ptr, "%s", &userCnic);

fscanf(ptr, "%s", &userDob);

fscanf(ptr, "%s", &userAge);

fscanf(ptr, "%s", &recitation);

fscanf(ptr, "%s", &secondRecitation);

setcolor(5);

outtextxy(30, 50, "Age: ");

outtextxy(65, 50, userAge);

outtextxy(30, 75, "Cnic: ");

outtextxy(75, 75, userCnic);

outtextxy(30, 100, "Date of birth: ");

outtextxy(145, 100, userDob);

outtextxy(30, 125, "Day to recite namaz: ");

if(strlen(recitation) > 0 || strlen(secondRecitation) > 0) {

outtextxy(200, 125, recitation);

outtextxy(361, 125, "-");

outtextxy(371, 125, secondRecitation);

} else {

outtextxy(200, 125, "None");

}

setcolor(14);

setfillstyle(1, 5);

line(270 + 5, 150 + 5, 370 + 5, 150 + 5);

line(370 + 5, 150 + 5, 370 + 5, 170 + 5);

line(370 + 5, 170 + 5, 270 + 5, 170 + 5);

line(270 + 5, 170 + 5, 270 + 5, 150 + 5);

floodfill(271, 151, 0);

outtextxy(293, 163, "Deassign");

userAction = getch();

if(userAction == 'd' || userAction == 'D')

goto deassignation;

else

exit(0);

deassignation:

cleardevice();

header("De-assignation", 1);

dialogueBox();

regSample("Execuse");

gets(execuses);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%s", loginNm);

ptr = fopen(buffer, "r");

for(i = 0; i < 2; i++) {

fscanf(ptr, "%s", &garbageData);

}

fscanf(ptr, "%s", &fileName);

fclose(ptr);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%s", fileName);

printf("%s", buffer);

ptr = fopen(buffer, "r");

for(i = 0; i < 4; i++) {

fscanf(ptr, "%s", &garbageData);

}

fscanf(ptr, "%s", &recitation);

fscanf(ptr, "%s", &secondRecitation);

fclose(ptr);

distance = 0;

userTop:

cleardevice();

header("Deassign", 0);

dialogueBox();

deassignOption(distance);

setcolor(0);

outtextxy(135, 120, "Which namaz you wanna deassign from this person");

outtextxy(250, 160, recitation);

outtextxy(250, 190, secondRecitation);

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(distance == 1) {

distance = 0;

} else {

distance++;

}

goto userTop;

}

if(userAction == 13) {

cleardevice();

sprintf(buffer, "C:\\TURBOC3\\FILES\\%s", fileName);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &dessUsername);

fscanf(ptr, "%s", &dessCnic);

fscanf(ptr, "%s", &dessDob);

fscanf(ptr, "%s", &dessAge);

fscanf(ptr, "%s", &recitation);

fscanf(ptr, "%s", &secondRecitation);

fclose(ptr);

ptr = fopen(buffer, "w");

fprintf(ptr, "%s\n", dessUsername);

fprintf(ptr, "%s\n", dessCnic);

fprintf(ptr, "%s\n", dessDob);

fprintf(ptr, "%s\n", dessAge);

if(distance == 0) {

fprintf(ptr, "%s\n", secondRecitation);

} else {

fprintf(ptr, "%s\n", recitation);

}

success("Deassigned", "You just deassigned yourself successfully");

}

getch();

exit(0);

// Beginning of admin panel/CMS panel

admin:

adminStart:

cleardevice();

header("CMS", 0);

cms();

active(adminOpt);

options();

userAction = getch();

if((adminOpt == 0) && (userAction == 13))

goto deassign;

else if((adminOpt == 1) && (userAction == 13))

goto assign;

else if((adminOpt == 2) && (userAction == 13))

goto search;

else if((adminOpt == 3) && (userAction == 13))

goto deleteUser;

if(userAction == 's' || userAction == 'S') {

if(adminOpt == 3)

adminOpt = 0;

else

adminOpt++;

goto adminStart;

}

resetMemory:

strcpy(deassignation, NULL);

strcpy(deassignation2, NULL);

count = 0;

distance = 0;

deassignRefresh:

cleardevice();

header("CMS", 0);

cms();

active(adminOpt);

options();

deassign:

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &fileQuantity);

fclose(ptr);

userList(activeList);

i = 1;

distance = 0;

up:

if(i <= fileQuantity) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", i);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &adminUsername);

outtextxy(225, 110 + distance, adminUsername);

fclose(ptr);

distance += 20;

i++;

goto up;

}

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(activeList == (fileQuantity - 1)) {

activeList = 0;

} else {

activeList++;

}

goto deassignRefresh;

}

if(userAction == 13) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", activeList + 1);

ptr = fopen(buffer, "r");

for(i = 0; i < 4; i++) {

fscanf(ptr, "%s", &garbageData);

}

refreshDeassign:

count = 0;

cleardevice();

header("CMS", 0);

cms();

active(adminOpt);

options();

fscanf(ptr, "%s", &deassignation);

// fprintf(stdout, deassignation);

fscanf(ptr, "%s", &deassignation2);

fclose(ptr);

setcolor(0);

cleardevice();

header("Deassign", 0);

dialogueBox();

setcolor(0);

if(strlen(deassignation) > 0)

count++;

if(strlen(deassignation2) > 0)

count++;

setcolor(14);

if(count != 0) {

deassignOption(0 + distance);

} else {

goto noNamaz;

}

setcolor(0);

outtextxy(135, 120, "Which namaz you wanna deassign from this person");

outtextxy(250, 160, deassignation);

outtextxy(250, 190, deassignation2);

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(distance == 0) {

distance++;

} else {

distance = 0;

}

goto refreshDeassign;

} else if(userAction == 13) {

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &dessUsername);

fscanf(ptr, "%s", &dessCnic);

fscanf(ptr, "%s", &dessDob);

fscanf(ptr, "%s", &dessAge);

fclose(ptr);

ptr = fopen(buffer, "w");

fprintf(ptr, "%s\n", dessUsername);

fprintf(ptr, "%s\n", dessCnic);

fprintf(ptr, "%s\n", dessDob);

fprintf(ptr, "%s\n", dessAge);

if(distance == 0) {

fprintf(ptr, "%s\n", deassignation2);

} else {

fprintf(ptr, "%s\n", deassignation);

}

success("Deassignation", "deassignation process successfully completed!");

goto resetMemory;

// getch();

} else {

strcpy(dessUsername, NULL);

strcpy(dessCnic, NULL);

strcpy(dessDob, NULL);

strcpy(dessAge, NULL);

goto resetMemory;

}

noNamaz:

cleardevice();

header("Error", 5);

dialogueBox();

setcolor(0);

outtextxy(160, 200, "No namaz is assigned to that person yet!");

getch();

goto resetMemory;

}

getch();

exit(0);

refresh:

cleardevice();

header("CMS", 0);

cms();

active(adminOpt);

options();

// Assignation process of namaz

assign:

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &fileQuantity);

fclose(ptr);

userList(activeList);

i = 1;

distance = 0;

deassignUp:

if(i <= fileQuantity) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", i);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &adminUsername);

outtextxy(225, 110 + distance, adminUsername);

fclose(ptr);

distance += 20;

i++;

goto deassignUp;

}

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(activeList == (fileQuantity - 1)) {

activeList = 0;

} else {

activeList++;

}

goto refresh;

}

if(userAction == 13) {

namazRefresh:

cleardevice();

header("Namaz", 0);

dialogueBox();

namazOpt(namazOption);

setcolor(0);

outtextxy(300, 130, "Fajr");

outtextxy(300, 160, "Zuhr");

outtextxy(304, 190, "Asr");

outtextxy(290, 220, "Maghrib");

outtextxy(300, 250, "Isha");

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(namazOption == 4) {

namazOption = 0;

goto namazRefresh;

} else {

namazOption++;

goto namazRefresh;

}

}

if((namazOption == 0) && (userAction == 13)) {

strcpy(namaz, "Fajr");

} else if((namazOption == 1) && (userAction == 13)) {

strcpy(namaz, "Zuhr");

} else if((namazOption == 2) && (userAction == 13)) {

strcpy(namaz, "Asr");

} else if((namazOption == 3) && (userAction == 13)) {

strcpy(namaz, "Maghrib");

} else if((namazOption == 4) && (userAction == 13)) {

strcpy(namaz, "Isha");

}

cleardevice();

header("Namz", 0);

dialogueBox();

regSample("Day");

gets(day);

// check(activeList + 1);;

getch();

if(check(activeList + 1) != 1) {

assignNamaz(day, namaz, (activeList + 1));

success("Assigned", "Namaz has been assigned to this person");

getch();

} else {

cleardevice();

header("Error", 5);

dialogueBox();

setcolor(5);

outtextxy(150, 200, "This person is assigned more than two namaz");

}

}

exit(0);

// Execuses panel

execuses:

outtextxy(320, 240, "Execuses will appear here");

getch();

exit(0);

// Search panel

search:

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &quantity);

fclose(ptr);

printf("%d", quantity);

cleardevice();

header("Search", 0);

dialogueBox();

regSample("Day");

gets(day);

searchRefresh:

cleardevice();

header("Search", 0);

dialogueBox();

namazOpt(namazOption);

setcolor(0);

outtextxy(300, 130, "Fajr");

outtextxy(300, 160, "Zuhr");

outtextxy(304, 190, "Asr");

outtextxy(290, 220, "Maghrib");

outtextxy(300, 250, "Isha");

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(namazOption == 4) {

namazOption = 0;

goto searchRefresh;

} else {

namazOption++;

goto searchRefresh;

}

}

if((namazOption == 0) && (userAction == 13)) {

strcpy(namaz, "Fajr");

} else if((namazOption == 1) && (userAction == 13)) {

strcpy(namaz, "Zuhr");

} else if((namazOption == 2) && (userAction == 13)) {

strcpy(namaz, "Asr");

} else if((namazOption == 3) && (userAction == 13)) {

strcpy(namaz, "Maghrib");

} else if((namazOption == 4) && (userAction == 13)) {

strcpy(namaz, "Isha");

}

cleardevice();

header("Reciter on searching day", 0);

dialogueBox();

searching(quantity, strcat(day, strcat("-", strcat("March-2022", strcat("/", namaz)))));

getch();

exit(0);

// Deleting process

deleteUser:

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "r");

fscanf(ptr, "%d", &fileQuantity);

fclose(ptr);

activeList = 0;

secondRefresh:

cleardevice();

header("CMS", 0);

cms();

active(adminOpt);

options();

userList(activeList);

i = 1;

distance = 0;

delUp:

if(i <= fileQuantity) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", i);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &adminUsername);

outtextxy(225, 110 + distance, adminUsername);

fclose(ptr);

distance += 20;

i++;

goto delUp;

}

userAction = getch();

if(userAction == 's' || userAction == 'S') {

if(activeList == (fileQuantity - 1)) {

activeList = 0;

} else {

activeList++;

}

cleardevice();

goto secondRefresh;

}

if(userAction == 13) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", (activeList + 1));

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &adminUsername);

fclose(ptr);

getch();

cleardevice();

header("Confirmation", 0);

dialogueBox();

setcolor(0);

sprintf(buffer, "Are you sure? you are going to delete %s permanently", adminUsername);

outtextxy(95, 200, buffer);

outtextxy(310, 220, "[Y/N]");

userAction = getch();

if(userAction == 'Y' || userAction == 'y') {

cleardevice();

header("Confirmation", 0);

dialogueBox();

setcolor(0);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", activeList + 1);

remove(buffer);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%s.txt", adminUsername);

remove(buffer);

fileQuantity--;

ptr = fopen("C:\\TURBOC3\\FILES\\files.txt", "w");

fprintf(ptr, "%d", fileQuantity);

fclose(ptr);

success("Deleted", "This user has been deleted!");

goto sorting;

} else {

goto admin;

}

}

sorting:

for(i = 1; i <= fileQuantity; i++) {

topSorting:

loopCounter++;

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", loopCounter);

ptr = fopen(buffer, "r");

if(ptr == 0) {

goto topSorting;

} else {

files++;

sprintf(buffer2, "C:\\TURBOC3\\FILES\\%d.txt", files);

rename(buffer, buffer2);

}

}

getch();

exit(0);

monthSet:

getch();

exit(0);

getch();

closegraph();

}

// Function descriptions

int assignNamaz(char day[10], char namaz[10], int id) {

FILE \*ptr = NULL;

char buffer[50];

char assignedNamaz[50];

char assignedDay[50];

strcpy(assignedNamaz, namaz);

strcpy(assignedDay, day);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", id);

ptr = fopen(buffer, "a");

fprintf(ptr, "%s\n", strcat(day, strcat("-", strcat("March-2022", strcat("/", namaz)))));

fclose(ptr);

return 0;

}

int check(int id) {

FILE \*ptr = NULL;

char buffer[50];

char buffer2[50];

char buffer3[50];

char name[50];

char cnic[50];

char dob[50];

char recitation[50];

char age[10];

char secondRecitation[50];

int i;

int status;

strcpy(name, NULL);

strcpy(cnic, NULL);

strcpy(dob, NULL);

strcpy(age, NULL);

strcpy(recitation, NULL);

strcpy(secondRecitation, NULL);

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", id);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &name);

fscanf(ptr, "%s", &cnic);

fscanf(ptr, "%s", &dob);

fscanf(ptr, "%s", &age);

fscanf(ptr, "%s", &recitation);

fscanf(ptr, "%s", &secondRecitation);

int length1;

int length2;

fclose(ptr);

length1 = strlen(recitation);

length2 = strlen(secondRecitation);

printf("%d", length1);

if(length1 > 0) {

if(length2 > 0) {

status = 1;

} else {

status = 0;

}

} else {

status = 0;

}

return status;

}

int searching(int quantityFiles, char searchedFor[50]) {

int i;

char buffer[50];

char garbageData[50];

char personName[50];

char firstRecitation[50];

char secondRecitation[50];

FILE \*ptr = NULL;

strcpy(firstRecitation, NULL);

strcpy(secondRecitation, NULL);

strcpy(garbageData, NULL);

for(i = 1; i <= quantityFiles; i++) {

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", i);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &garbageData);

fscanf(ptr, "%s", &garbageData);

fscanf(ptr, "%s", &garbageData);

fscanf(ptr, "%s", &garbageData);

fscanf(ptr, "%s", &firstRecitation);

fscanf(ptr, "%s", &secondRecitation);

if((strcmp(searchedFor, firstRecitation) == 0) || (strcmp(searchedFor, secondRecitation) == 0)) {

goto finish;

}

fclose(ptr);

}

finish:

sprintf(buffer, "C:\\TURBOC3\\FILES\\%d.txt", i);

ptr = fopen(buffer, "r");

fscanf(ptr, "%s", &personName);

fclose(ptr);

setcolor(0);

sprintf(buffer, "%s will recite namaz", personName);

if(strcmp(personName, "URF2") != 0)

outtextxy(270, 200, buffer);

else

outtextxy(140, 200, "No one is reciting namaz on your searched day");

return 0;

}

void namazOpt(int namazOption) {

int loop;

int i = 0;

for(loop = 0; loop < namazOption; loop++) {

i += 30;

}

setcolor(14);

setfillstyle(1, 14);

line(getmaxx() - 550, 120 + i, getmaxx() - 90, 120 + i);

line(getmaxx() - 90, 120 + i, getmaxx() - 90, 145 + i);

line(getmaxx() - 90, 145 + i, getmaxx() - 550, 145 + i);

line(getmaxx() - 550, 145 + i, getmaxx() - 550, 120 + i);

floodfill(getmaxx() - 400, 130 + i, 14);

}

void userList(int distance) {

int loop;

int i = 0;

for(loop = 0; loop < distance; loop++) {

i += 20;

}

setcolor(14);

setfillstyle(1, 14);

line(215, 105 + i, 315, 105 + i);

line(315, 105 + i, 315, 120 + i);

line(315, 120 + i, 215, 120 + i);

line(215, 120 + i, 215, 105 + i);

floodfill(216, 106 + i, 14);

setcolor(0);

}

void options() {

outtextxy(100, 120, "deassign");

outtextxy(100, 150, "assign");

outtextxy(100, 180, "Search");

outtextxy(100, 210, "Delete User");

}

void active(int optNum) {

int loop;

int i = 0;

if(optNum > 0) {

for(loop = 1; loop <= optNum; loop++) {

i += 30;

}

}

setcolor(14);

setfillstyle(1, 14);

line(getmaxx() - 550, 110 + i, getmaxx() - 435, 110 + i);

line(getmaxx() - 435, 110 + i, getmaxx() - 435, 140 + i);

line(getmaxx() - 435, 140 + i, getmaxx() - 550, 140 + i);

line(getmaxx() - 550, 140 + i, getmaxx() - 550, 110 + i);

floodfill(getmaxx() - 500, 115 + i, 14);

setcolor(0);

}

void dialogueBox() {

setcolor(7);

setfillstyle(1, 7);

line(getmaxx() - 550, 100, getmaxx() - 90, 100);

line(getmaxx() - 90, 100, getmaxx() - 90, 300);

line(getmaxx() - 90, 300, getmaxx() - 550, 300);

line(getmaxx() - 550, 300, getmaxx() - 550, 100);

floodfill(getmaxx() - 549, 101, 7);

}

void cms() {

dialogueBox();

setcolor(0);

setfillstyle(1, 0);

line(205, 90, 215, 90);

line(215, 90, 215, 310);

line(215, 310, 205, 310);

line(205, 310, 205, 90);

floodfill(210, 300, 0);

}

void deassignOption(int distance) {

int i, space = 0;

for(i = 0; i < distance; i++) {

space += 30;

}

setcolor(14);

setfillstyle(1, 14);

line(getmaxx() - 550, 150 + space, getmaxx() - 90, 150 + space);

line(getmaxx() - 90, 150 + space, getmaxx() - 90, 180 + space);

line(getmaxx() - 90, 180 + space, getmaxx() - 550, 180 + space);

line(getmaxx() - 550, 180 + space, getmaxx() - 550, 150 + space);

floodfill(getmaxx() - 540, (150 + space) + 5, 14);

}

void header(char title[10], int bkColor) {

setcolor(7);

setfillstyle(1, 7);

line(0, 0, getmaxx(), 0);

line(getmaxx(), 0, getmaxx(), 20);

line(getmaxx(), 20, 0, 20);

line(0, 20, 0, 0);

floodfill(1, 1, 7);

setbkcolor(bkColor);

setcolor(0);

outtextxy(300, 7, title);

}

void regSample(char title[10]) {

setcolor(0);

outtextxy(270 + 29, 150 + 16, title);

setfillstyle(1, 0);

line(270 + 10, 150 + 56, 370 + 10, 150 + 56);

line(370 + 10, 150 + 56, 370 + 10, 170 + 56);

line(370 + 10, 170 + 56, 270 + 10, 170 + 56);

line(270 + 10, 170 + 56, 270 + 10, 150 + 56);

floodfill(270 + 11, 150 + 58, 0);

gotoxy(37, 14);

}

void cmsRegSample(char title[10]) {

setcolor(0);

int i = 80;

outtextxy(360, 160, title);

setfillstyle(1, 0);

line(270, 110 + i, 500, 110 + i);

line(500, 110 + i, 500, 130 + i);

line(500, 130 + i, 270, 130 + i);

line(270, 130 + i, 270, 110 + i);

floodfill(270 + 11, (110 + i) + 3, 0);

gotoxy(36, 13);

}

int success(char message[50], char successMessage[50]) {

cleardevice();

header(message, 7);

setbkcolor(2);

dialogueBox();

setcolor(2);

outtextxy(140, 180, successMessage);

getch();

return 0;

}