2-10 PROJE RAPORU

PROJE ADI: Etkinlik Kayıt Sistemi

PROJE YAPANLAR:

Göktuğ Gezer

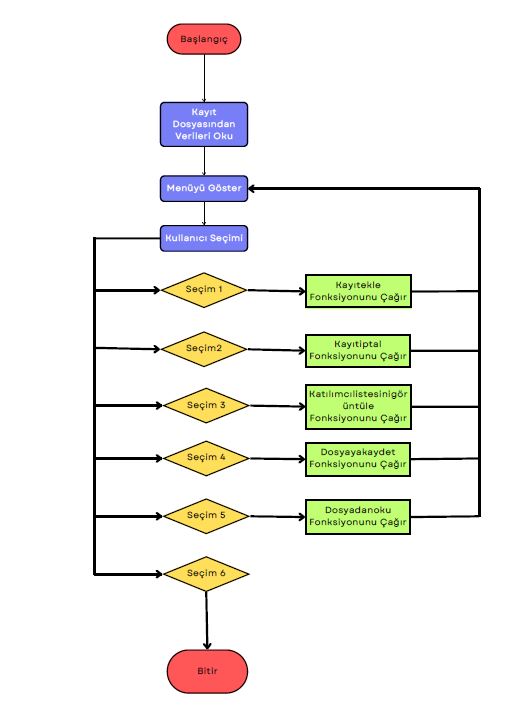
Kerem Çimentepe

Furkan İşeri

PROJE AÇIKLAMASI: Katılımcı detaylarını saklamak için yapılar içinde bir etkinlik kayıt sistemi tasarlayın. Kayıt, iptal ve katılımcı listesi görüntüleme fonksiyonlarını gerçekleştirin. Veriyi bir dosyaya kaydedin.

PROJE FAYDALARI:Kullanıcı ,kullanıcı dostu ekrandan kayıt ekleyebilir ,silebilir ,katılımcı listesi görüntüleme ,dosya kaydetme ,dosyadan oku gibi işlevleri gerçekleştirebilir.Bu etkinlikte kullanıcıların bilgilerini yönetmede kolaylık sağlar.

AKIŞ ŞEMASI:



KOD:

main.c dosyası

---------------------------------------------------------------------------------------

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "structs.c"

#include "kayıtekle.c"

#include "kayıtiptal.c"

#include "katılımcılistesinigörüntüle.c"

#include "dosyayakaydet.c"

#include "dosyadanoku.c"

// Programın başlangıcını belirten main fonksiyonu

int main() {

// Katılımcıları depolamak için bir dizi ve diğer gerekli değişkenleri tanımlayın

struct Katilimci katilimcilar[100];

int katilimciSayisi = 0;

int secim;

// Başlangıç verilerini bir dosyadan okuyun

dosyadanOku(katilimcilar, &katilimciSayisi);

// Kullanıcı girişini işleyen bir döngüde ana menüyü görüntüleyin ve işlemleri gerçekleştirin

do {

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*------------------------------------------------\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\* Etkinlik Kayit Sistemi \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*------------------------------------------------\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*------------------------------------------------\*\*\n");

printf("\*\*-------------###--------------###---------------\*\*\n");

printf("\*\*-------------###------##------###---------------\*\*\n");

printf("\*\*------##--------------##--------------##--------\*\*\n");

printf("\*\*--------##--------------------------##----------\*\*\n");

printf("\*\*----------###--------------------###------------\*\*\n");

printf("\*\*-------------####################---------------\*\*\n");

printf("\*\*------------------------------------------------\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("|| 1. Kayit Ekle ||\n");

printf("|| 2. Kayit Iptal Et ||\n");

printf("|| 3. Katilimci Listesini Goruntule ||\n");

printf("|| 4. Dosyaya Kaydet ||\n");

printf("|| 5. Dosyadan Oku ||\n");

printf("|| 6. Cikis ||\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("Seciminizi girin: ");

scanf("%d", &secim);

// Kullanıcının girişini bir switch ifadesi kullanarak işleyin

switch (secim) {

case 1:

kayitEkle(katilimcilar, &katilimciSayisi);

break;

case 2:

kayitIptal(katilimcilar, &katilimciSayisi);

break;

case 3:

katilimciListesiGoruntule(katilimcilar, katilimciSayisi);

break;

case 4:

dosyayaKaydet(katilimcilar, katilimciSayisi);

break;

case 5:

dosyadanOku(katilimcilar, &katilimciSayisi);

break;

case 6:

printf("Programdan cikiliyor.\n");

break;

default:

printf("Gecersiz secim. Tekrar deneyin.\n");

}

} while (secim != 6);

return 0;

}

structs.c dosyası

struct Katilimci {

char isim[50];

char soyisim[50];

int yas;

};

kayıtekle.c dosyası

// Yeni bir katılımcıyı diziyi ekleyen fonksiyon

void kayitEkle(struct Katilimci katilimcilar[], int \*katilimciSayisi) {

// Eğer katılımcı limitine henüz ulaşılmamışsa işleme devam et

if (\*katilimciSayisi < 100) {

// Yeni katılımcının adını ve soyadını al

printf("Ad: ");

scanf("%s", katilimcilar[\*katilimciSayisi].isim);

printf("Soyad: ");

scanf("%s", katilimcilar[\*katilimciSayisi].soyisim);

// Geçerli bir yaş alana kadar işleme devam et

do {

printf("Yas: ");

scanf("%d", &katilimcilar[\*katilimciSayisi].yas);

if (katilimcilar[\*katilimciSayisi].yas <= 0) {

printf("Gecersiz yas. Pozitif bir yas girin.\n");

}

} while (katilimcilar[\*katilimciSayisi].yas <= 0);

// Katılımcı sayısını artır ve işlemi başarıyla tamamla

(\*katilimciSayisi)++;

printf("Kayit basariyla eklendi.\n");

} else {

// Katılımcı limitine ulaşıldıysa uyarı ver

printf("Katilimci limitine ulasildi.\n");

}

}

kayıtiptal.c dosyası

// Bir katılımcının kaydını iptal eden fonksiyon

void kayitIptal(struct Katilimci katilimcilar[], int \*katilimciSayisi) {

// Eğer kayıt sayısı sıfırdan büyükse işleme devam et

if (\*katilimciSayisi > 0) {

// İptal edilecek kaydın sıra numarasını al

int iptalIndex;

printf("Iptal edilecek kayidin sira numarasini girin: ");

scanf("%d", &iptalIndex);

// Sıra numarası geçerli ise işleme devam et

if (iptalIndex > 0 && iptalIndex <= \*katilimciSayisi) {

// İptal edilecek kayıttan sonraki kayıtları bir önceki kayıta taşı

for (int i = iptalIndex - 1; i < \*katilimciSayisi - 1; i++) {

katilimcilar[i] = katilimcilar[i + 1];

}

// Kayıt sayısını azalt ve işlemi başarıyla tamamla

(\*katilimciSayisi)--;

printf("Kayit basariyla iptal edildi.\n");

} else {

// Geçersiz sıra numarası durumunda uyarı ver

printf("Gecersiz sira numarasi.\n");

}

} else {

// Kayıt sayısı sıfırsa uyarı ver

printf("Iptal edilecek kayit bulunmamaktadir.\n");

}

}

katılımcılistesinigörüntüle.c dosyası

// Katılımcı listesini görüntüleyen fonksiyon

void katilimciListesiGoruntule(struct Katilimci katilimcilar[], int katilimciSayisi) {

// Eğer kayıtlı katılımcı varsa işleme devam et

if (katilimciSayisi > 0) {

printf("\nKatilimci Listesi:\n");

// Tüm katılımcıları listeleyen bir döngü

for (int i = 0; i < katilimciSayisi; i++) {

// Her bir katılımcının bilgilerini ekrana yazdırma

printf("%d. %s %s, Yas: %d\n", i + 1, katilimcilar[i].isim, katilimcilar[i].soyisim, katilimcilar[i].yas);

}

} else {

// Kayıtlı katılımcı yoksa uygun bir uyarı göster

printf("Kayitli katilimci bulunmamaktadir.\n");

}

}

dosyayakaydet.c dosyası

#include <stdio.h>

// Katılımcı dizisini bir dosyaya kaydeden fonksiyon

void dosyayaKaydet(struct Katilimci katilimcilar[], int katilimciSayisi) {

// "katilimcilar.txt" adında bir dosyayı yazma modunda aç

FILE \*dosya = fopen("katilimcilar.txt", "w");

// Dosya başarıyla açılırsa işleme devam et

if (dosya != NULL) {

// Tüm katılımcıları dosyaya yazan bir döngü

for (int i = 0; i < katilimciSayisi; i++) {

// Her bir katılımcının bilgilerini dosyaya yaz

fprintf(dosya, "%s %s %d\n", katilimcilar[i].isim, katilimcilar[i].soyisim, katilimcilar[i].yas);

}

// Dosyayı kapat ve işlemi başarıyla tamamla

fclose(dosya);

printf("Veri dosyaya kaydedildi.\n");

} else {

// Dosya açılamazsa uygun bir hata mesajı göster

printf("Dosya acilirken hata olustu.\n");

}

}

dosyadanoku.c dosyası

#include <stdio.h>

// Dosyadan veri okuyup, katılımcı dizisini güncelleyen fonksiyon

void dosyadanOku(struct Katilimci katilimcilar[], int \*katilimciSayisi) {

// "katilimcilar.txt" adında bir dosyayı okuma modunda aç

FILE \*dosya = fopen("katilimcilar.txt", "r");

// Dosya başarıyla açılırsa işleme devam et

if (dosya != NULL) {

// Dosyadan okuma öncesinde sayacı sıfırla

\*katilimciSayisi = 0;

// Dosyadan veri okuma işlemi

while (fscanf(dosya, "%s %s %d", katilimcilar[\*katilimciSayisi].isim, katilimcilar[\*katilimciSayisi].soyisim, &katilimcilar[\*katilimciSayisi].yas) != EOF) {

// Okunan her bir satır için katılımcı sayısını artır

(\*katilimciSayisi)++;

}

// Dosyayı kapat ve işlemi başarıyla tamamla

fclose(dosya);

printf("Veri dosyadan okundu.\n");

} else {

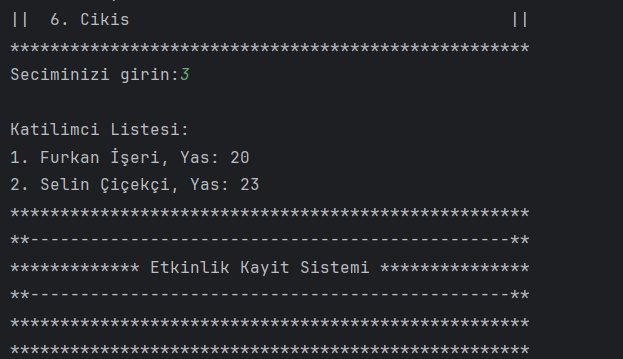
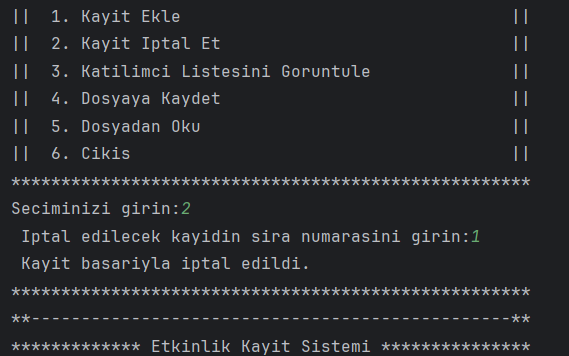
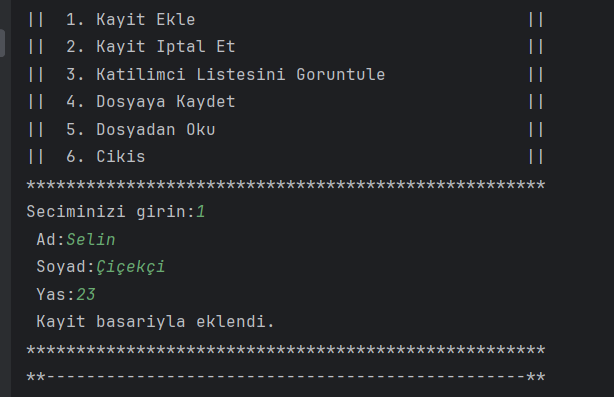
// Dosya açılamazsa uygun bir hata mesajı göster

printf("Dosya acilirken hata olustu.\n");

}

}

EKRAN ÇIKTILARI:



metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu