```c
//main.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include "prima_pagina.c"

int main(){
    firstPage();
    return 0;
}
//prima_pagina.c
#include "functii_angajati.c"
#include "functii_firma.c"
#include "functii_clienti.c"
#include "functii_imobile.c"
#include "inchidere.c"

void firstPage();

void secondPage(int option){
    switch (option)
    {
    case 1:
        option1();
        break;
    case 2:
        option2();
        break;
    case 3:
        option3();
        break;
    case 4:
        option4();
        break;
    case 0:
```

```
        option5();
        break;
    default:
        break;
    }

}
void menu(){
    printf("1 detalii firma \n2 angajati \n3 clienti\n4
imobile\n0 iesire\n");
}
void firstPage(){
    //system("cls");
    menu();
    int option;
    scanf("%d", &option);
    while(option < 0 || option > 4){
        menu();
        scanf("%d", &option);
    }
    secondPage(option);
}
//functii_firma.c
typedef struct firma{
    char nume[211], adresa[310], telefon[101];
    long long int cod;
}firma;

void printFirma(firma df){
    fflush(stdin);
    printf("Nume firma: %s\n", df.nume);
    printf("Cod firma: %lli\n\n", df.cod);
    printf("Adresa: %s\n", df.adresa);
    printf("Telefon: %s\n\n", df.telefon);
}
void option1(){
```

```
    system("cls");
    FILE *det_firma = fopen("detalii_firma.csv", "r");
    if(det_firma == NULL){
        perror("Unable to open the file");
        exit(1);
    }
    char line[1024];
    firma df;
    int row = 1;

    while(fgets(line, sizeof(line), det_firma)){
        char *token;
        int field = 0;
        token = strtok(line, ",");
        while(token != NULL){
            if(row == 1 && field == 1)
                strcpy(df.nume, token);

            else if(row == 2 && field == 1){
               df.cod = atoi(token);
            }

            else if(row == 3 && field == 1)
                strcpy(df.adresa, token);

            else if(row == 4 && field == 1)
                strcpy(df.telefon, token);

            field++;
            token = strtok(NULL, ",");
        }
        row++;
    }
    printFirma(df);

    fclose(det_firma);
```

```
    printf("Apasa orice pentru reintoarcere\n");
    getch();
    firstPage();
}
//functii_angajati
void firstPage();

typedef struct angajat{
    char nume[251], sediu[151];
}angajat;

void print_angajati(angajat angajati[], int row){
    for(int i = 0; i < row; i++){
        printf("%d. Nume: %s | Sediu: %s\n", i + 1,
angajati[i].nume, angajati[i].sediu);
    }
}

void adaugare_angajat(angajat date_angajat[], angajat a, int
len){
    int i, added = 0;
    angajat tmp;
    if(strcmp(a.nume, date_angajat[len - 1].nume) > 0){
        date_angajat[len] = a;
    }
    else{
        for(i = 0; i <= len + 1; i++){
            if(added){
                a = date_angajat[i];
                date_angajat[i] = tmp;
                tmp = a;
            }
            else if(strcmp(a.nume, date_angajat[i].nume) < 0){
                tmp = date_angajat[i];
                date_angajat[i] = a;
                added = 1;
```

```
            }
        }
    }
    FILE *temp = fopen("temp.csv", "w");

    fprintf(temp, "%s,%s", date_angajat[0].nume,
date_angajat[0].sediu);

    for(int i = 1; i <= len; i++){
        fprintf(temp,"%s,%s", date_angajat[i].nume,
date_angajat[i].sediu);
    }
    fclose(temp);
    remove("angajati_sedii.csv");
    rename("temp.csv", "angajati_sedii.csv");
}

void sterge_angajat(angajat s[], int len, int start_pos){
    for(start_pos; start_pos < len; start_pos++){
        s[start_pos] = s[start_pos + 1];
    }
    FILE *tmp = fopen("temp.csv", "w");
    if(len == 0){
        fprintf(tmp, "");
        fclose(tmp);
        remove("angajati_sedii.csv");
        rename("temp.csv", "angajati_sedii.csv");
        return;
    }
    fprintf(tmp, "%s,%s", s[0].nume, s[0].sediu);
    for(int i = 1; i < len; i++){
        fprintf(tmp,"%s,%s", s[i].nume, s[i].sediu);
    }
    fclose(tmp);
    remove("angajati_sedii.csv");
    rename("temp.csv", "angajati_sedii.csv");
```

```
}
void option2(){
    //system("cls");
    FILE *ang_sed = fopen("angajati_sedii.csv", "r");
    if(ang_sed == NULL){
        perror("Unable to open the file");
        exit(1);
    }
    angajat date_angajat[100];
    char line[1024];
    int row = 0;
    while(fgets(line, sizeof(line), ang_sed)){
        char *token;
        int field = 0;
        token = strtok(line, ",");

        while(token != NULL){
            if(field == 0){
                strcpy(date_angajat[row].nume, token);
            }
            else
                strcpy(date_angajat[row].sediu, token);
            field++;
            token = strtok(NULL, ",");
        }
        row++;
    }
    fclose(ang_sed);

    print_angajati(date_angajat, row);

    int option;
    printf("1 adaugare\n2 stergere\n3 back\n");
    scanf("%d", &option);
    switch (option)
    {
```

```
    case 1:
        angajat a;
        printf("\nIntroduceti numele: ");
        fflush(stdin);
        gets(a.nume);
        printf("\nIntroduceti sediul: ");
        fflush(stdin);
        gets(a.sediu);
        strcat(a.sediu, "\n");
        adaugare_angajat(date_angajat, a, row);
        option2();
        break;
    case 2:
        printf("Introduceti indexul angajatului pe care doriti
sa-l stergeti\n");
        int i;
        scanf("%d", &i);
        i--;
        sterge_angajat(date_angajat, row - 1, i);
        option2();
        break;
    case 3:
        firstPage();
        break;

    default:
        break;
    }
}
//functii_clienti.c
typedef struct client{
    char nume[251], tip[50], perioada[100];
}client;

void printClienti(client date_clienti[], int row){
    for(int i = 0; i < row; i++){
```

```
        printf("%d Nume: %s | Tip: %s | ", i + 1,
date_clienti[i].nume, date_clienti[i].tip);
        if(!strcmp(date_clienti[i].perioada, "nu\n")){
            printf("Nu are contract\n\n");
        }
        else{
            printf("Perioada: %s\n", date_clienti[i].perioada);
        }
    }
}

void adaugare_client(client date_client[], client c, int len){
    int i, added = 0;
    client tmp;
    if(strcmp(c.nume, date_client[len - 1].nume) > 0){
        date_client[len] = c;
    }
    else{
        for(i = 0; i <= len + 1; i++){
            if(added){
                c = date_client[i];
                date_client[i] = tmp;
                tmp = c;
            }
            else if(strcmp(c.nume, date_client[i].nume) < 0){
                tmp = date_client[i];
                date_client[i] = c;
                added = 1;
            }
        }
    }
    FILE *temp = fopen("temp.csv", "w");

    fprintf(temp, "%s,%s,", date_client[0].nume,
date_client[0].tip);
    if(!strcmp(date_client[0].perioada, "nu are contract\n"))
```

```
        fprintf(temp, "nu\n");
    for(int i = 1; i <= len; i++){
        fprintf(temp,"%s,%s,", date_client[i].nume,
date_client[i].tip);
        if(!strcmp(date_client[i].perioada, "nu are contract\n"))
            fprintf(temp, "nu\n");
        else
            fprintf(temp, "%s", date_client[i].perioada);
    }
    fclose(temp);
    remove("clienti.csv");
    rename("temp.csv", "clienti.csv");
}

void sterge_client(client date_client[], int len, int start_pos){
    for(start_pos; start_pos < len; start_pos++){
        date_client[start_pos] = date_client[start_pos + 1];
    }
    FILE *tmp = fopen("temp.csv", "w");
    if(len == 0){
        fprintf(tmp, "");
        fclose(tmp);
        remove("clienti.csv");
        rename("temp.csv", "clienti.csv");
        return;
    }
    fprintf(tmp, "%s,%s,%s", date_client[0].nume,
date_client[0].tip, date_client[0].perioada);
    for(int i = 1; i < len; i++){
        fprintf(tmp,"%s,%s,%s", date_client[i].nume,
date_client[i].tip, date_client[i].perioada);
    }
    fclose(tmp);
    remove("clienti.csv");
    rename("temp.csv", "clienti.csv");
}
```

```c
void option3(){
    system("cls");
    FILE *clienti = fopen("clienti.csv", "r");
    if(clienti == NULL){
        perror("Unable to open the file");
        exit(1);
    }
    char line[1024];
    client date_clienti[100];
    int row = 0;
    while(fgets(line, sizeof(line), clienti)){
        char *token;
        int field = 0;

        token = strtok(line, ",");

        while(token != NULL){
            if(field == 0){
                strcpy(date_clienti[row].nume, token);
            }
            else if(field == 1){
                strcpy(date_clienti[row].tip, token);
            }
            else{
                strcpy(date_clienti[row].perioada, token);
            }

            field++;
            token = strtok(NULL, ",");
        }
        row++;
    }
    fclose(clienti);

    printClienti(date_clienti, row);
```

```
    int opt;
    printf("1 adauga\n2 sterge\n3 back\n");
    scanf("%d", &opt);
    switch (opt)
    {
    case 1:
        client c;
        printf("Introduceti numele clientului: ");
        fflush(stdin);
        gets(c.nume);
        printf("Introduceti tipul clientului (fizic/firma): ");
        fflush(stdin);
        gets(c.tip);
        fflush(stdin);
        printf("Introduceti perioada contractului
(nedeterminat/perioada/nu are contract): ");
        gets(c.perioada);
        strcat(c.perioada, "\n");
        fflush(stdin);
        adaugare_client(date_clienti, c, row);
        option3();
        break;

    case 2:
        printf("Introduceti indexul clientului pe care doriti sa-
l stergeti (0 - cancel)\nIn cazul in care introduceti un index
mai mare decat indexul ultimului client, se va sterge ultimul\nla
fel si cu primul client\n");
        int i;
        scanf("%d", &i);
        if(i == 0){
            option3();
            break;
        }
        i--;
```

```
        sterge_client(date_clienti, row - 1, i);
        option3();
        break;

    case 3:
        firstPage();
        break;
    default:
        break;
    }
}
//functii_imobile
void option4(){
    system("cls");
    getch();
    firstPage();
}
//inchidere.c
void option5(){
    system("cls");
    char yesno[1];
    printf("Esti sigur ca vrei sa iesi? [y/n] ");
    scanf("%s", yesno);
    if(!strcmp(yesno, "n")){
        firstPage();
    }
    else system("cls");
}
```