

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>
```

```
struct project
{
    int proj_no;
    char title[100];
    char type[100];
    float amount;
    char status[100];
    char deadline[100];
    char department[100];
};
```

```
void loadingBar();
void addproject();
void displayproject();
void deleteproject();
void editproject();
```

```
int main()
{
    loadingBar();
    menu();
}
```

```
void loadingBar()
{
    system("color E0");

    char a = 177, b = 219;


    printf("\n\n\n\n");
    printf("\n\n\n\n\n\n\t\t\t\t\tInitializing...\n\n");
    printf("\t\t\t\t\t");


    // Print initial loading bar
    for (int i = 0; i < 26; i++)
        printf("%c", a);


    // Set the cursor again starting
    // point of loading bar
    printf("\r");
    printf("\t\t\t\t\t");


    // Print loading bar progress
    for (int i = 0; i < 26; i++)
    {
        printf("%c", b);


        // Sleep for 1 second
        Sleep(100);
    }

    system("cls");
```

```
printf("\n\n\n\n");
printf("\n\n\n\n\n\n\t\t\t\t\tPlease Wait...\n\n");
printf("\t\t\t\t\t");
```

```
// Print initial loading bar
```

```
for (int i = 0; i < 26; i++)
```

```
    printf("%c", a);
```

```
// Set the cursor again starting
```

```
// point of loading bar
```

```
printf("\r");
```

```
printf("\t\t\t\t\t");
```

```
// Print loading bar progress
```

```
for (int i = 0; i < 26; i++)
```

```
{
```

```
    printf("%c", b);
```

```
// Sleep for 1 second
```

```
    Sleep(50);
```

```
}
```

```
system ("cls");
```

```
printf("\n\n\n\n\n\n\n\n\t\t\t\t\tLoaded Successfully. Welcome!!\n\n");
```

```
printf("\t\t\t\t\t[Press any keys to continue]");
```

```
getch();
```

```
}
```

```

void menu()
{
    int choice;
    char again;

    // Create the file "information.txt" if it doesn't exist
    FILE *file;
    file = fopen("information.txt", "a");
    if (file == NULL)
    {
        printf("Failed to create the file.\n");
        return 1;
    }
    fclose(file);

    system("cls");
    do {
        system("cls");
        printf("\n\n\t\t\t-----");
        printf("\n\t\t\t|----- PROJECT MANAGEMENT SYSTEM FOR TUP MANILA -----|\n");
        printf("\t\t\t|-----|\n");
        printf("\t\t\t\t\t 1. Add a Project\t\t\t |\t\n");
        printf("\t\t\t\t\t 2. Display Projects\t\t\t |\t\n");
        printf("\t\t\t\t\t 3. Delete a Project\t\t\t |\t\n");
        printf("\t\t\t\t\t 4. Edit a Project\t\t\t |\t\n");
        printf("\t\t\t\t\t 5. Exit\t\t\t\t |\t\n");
        printf("\t\t\t\t\t-----\n");
        printf("\t\t\t\t\tEnter your choice number: ");
        scanf("%d", &choice);
    } while (choice < 0 || choice > 5);
}

```

```
printf("\t\t\t-----\n");
```

```
switch (choice)
```

```
{
```

```
    case 1:
```

```
        addproject();
```

```
        break;
```

```
    case 2:
```

```
        displayproject();
```

```
        break;
```

```
    case 3:
```

```
        deleteproject();
```

```
        break;
```

```
    case 4:
```

```
        editproject();
```

```
        break;
```

```
    case 5:
```

```
        return 0; // Exits the program
```

```
    default:
```

```
        printf("\t\t\t\tInvalid choice. Please try again.\n");
```

```
        printf("\t\t\t-----\n");
```

```
}
```

```
printf("\n\t\t\tPress 'Y' if you want to return to the menu and 'N' if No: ");
```

```
again = getch();
```

```
if(again == 'n' || again == 'N')
```

```
{
```

```
    menu();
```

```
}
```

```

}while(again == 'y' || again == 'Y');

return 0;
}

void addproject()
{
    char another;
    FILE *fp;
    struct project record;

    system("cls");
    printf("\n\n\t\t\t-----");
    printf("\n\t\t\t----- ADD PROJECT -----\\n");
    printf("\t\t\t-----\\n");

    do {
        printf("\n\n\t\t\tEnter Project Number    : ");
        while (scanf("%d", &record.proj_no) != 1)
        {
            printf("\n\t\t\tERROR. Invalid input for Project Number. Please try again and enter a
number.\\n");
            fflush(stdin); // Clear the input buffer
            printf("\n\n\t\t\tEnter Project Number    : ");
        }

        // to check the repeating project numbers
        fp = fopen("information.txt", "r");
        if (fp != NULL)

```

```

{
    struct project existing;

    while (fscanf(fp, "Project Number: %d\nTitle: %[^\\n]\\nProject Type: %[^\\n]\\nAmount Project
[PHP]: %f\\nProject Status: %[^\\n]\\nDeadline of the Project: %[^\\n]\\nSchool Department: %[^\\n]\\n\\n",
        &existing.proj_no, existing.title, existing.type, &existing.amount, existing.status,
        existing.deadline, existing.department) != EOF)
    {
        if (existing.proj_no == record.proj_no)
        {
            printf("\\n\\t\\t ERROR. Project number already exists. Please re-enter a different project
number.\\n");

            fclose(fp);
            fflush(stdin);

            record.proj_no = -1; // Set an invalid project number to re-entering

            break;
        }
    }

    fclose(fp);
}

if (record.proj_no == -1)
{
    continue; // Restart the loop to re-enter the project number
}

printf("\\n\\t\\t\\tEnter Title          : ");
scanf(" %[^\\n]", record.title);

printf("\\n\\t\\t\\tEnter Project Type      : ");
scanf(" %[^\\n]", record.type);

printf("\\n\\t\\t\\tEnter the Amount of the Project (PhP): ");

```

```

while (scanf("%f", &record.amount) != 1)
{
    printf("\n\t\t ERROR. Invalid input for Amount of the Project. Please enter a number.\n");
    fflush(stdin); // Clear the input buffer
    printf("\n\t\t\t\t Enter the Amount of the Project: ");
}

```

```

printf("\n\t\t\t\t Enter Project Status \n\t\t\t\t (In Progress, Completed, Cancelled): ");
scanf(" %[^\\n]", record.status);
printf("\n\t\t\t\t Enter the Deadline of the Project (mm/dd/yy): ");
scanf(" %[^\\n]", record.deadline);
printf("\n\t\t\t\t Enter the School Department of the Project: ");
scanf(" %[^\\n]", record.department);
printf("\t\t\t\t-----\\n");

```

```

fp = fopen("information.txt", "a");
if (fp == NULL)
{
    fprintf(stderr, "Can't open file");
    exit(1);
}

```

```

fprintf(fp, "Project Number: %d\\nTitle: %s\\nProject Type: %s\\nAmount Project [PHP]: %.2f\\nProject
Status: %s\\nDeadline of the Project: %s\\nSchool Department: %s\\n\\n",

```

```

    record.proj_no, record.title, record.type, record.amount, record.status, record.deadline,
    record.department);

```

```

fclose(fp);

```



```

printf("\t\t\t\tRecord stored successfully\n");
printf("\t\t\t-----\n");

printf("\n\t\t\tPress 'Y' if you want to return to the menu and 'N' if No: ");
another = getch();
if (another == 'n' || another == 'N')
{
    menu();
}
} while (another == 'y' || another == 'Y');

return 0;
}

```

```

void displayproject()
{
    int searchProjNo;
    int found = 0;
    char searchAgain = 'y';

    FILE *fp;
    struct project record;
    fp = fopen("information.txt", "r");

    if (fp == NULL)
    {
        fprintf(stderr, "\n\t\t\t\tCan't open file\n");
        exit(1);
    }
}

```

```

    } else
    {
        system("cls");
        printf("\n\t\t\t----- RECORDS ----- \n");
    }

while (searchAgain == 'y' || searchAgain == 'Y')
{
    found = 0;
    printf("\n\t\t\tDo you want to search by project number? (y/n): \n");

    char choice = getch();

    if (choice == 'y' || choice == 'Y')
    {
        // Prompt all project numbers and titles
        printf("\n\t\t\t-----");
        printf("\n\t\t\tAVAILABLE RECORDS:\n");
        rewind(fp); // Go back to the beginning of the file

        int recordCount = 0; // Track the number of records

        while (fscanf(fp, "Project Number: %d\nTitle: %[^\n]\nProject Type: %[^\n]\nAmount Project
[PHP]: %f\nProject Status: %[^\n]\nDeadline of the Project: %[^\n]\nSchool Department: %[^\n]\n\n",
            &record.proj_no, record.title, record.type, &record.amount, record.status,
            record.deadline, record.department) != EOF)
        {
            printf("\n\t\t\tPROJECT NUMBER:[%d] - TITLE:[%s]\n", record.proj_no, record.title);
            recordCount++;
        }
    }
}

```

```

    }

    printf("\t\t\t-----");

    if (recordCount == 0)
    {
        printf("\n\t\t\tNo records found. Please add records to continue.\n");
        printf("\t\t\t-----\n");
        break; // Exit the loop and return to the main menu
    }

    printf("\n\t\t\tEnter the project number to search: ");
    scanf("%d", &searchProjNo);

    rewind(fp); // Go back to the beginning of the file

    while (fscanf(fp, "Project Number: %d\nTitle: %[^\\n]\\nProject Type: %[^\\n]\\nAmount Project
[PHP]: %f\\nProject Status: %[^\\n]\\nDeadline of the Project: %[^\\n]\\nSchool Department: %[^\\n]\\n\\n",
        &record.proj_no, record.title, record.type, &record.amount, record.status,
        record.deadline, record.department) != EOF)
    {
        if (record.proj_no == searchProjNo)
        {
            found = 1;
            printf("\t\t\t-----\n");
            printf("\n\t\t\t\tProject Number          : %d", record.proj_no);
            printf("\n\t\t\t\tProject Title           : %s", record.title);
            printf("\n\t\t\t\tType                : %s", record.type);
            printf("\n\t\t\t\tAmount              : %.2f", record.amount);
            printf("\n\t\t\t\tStatus              : %s", record.status);

```

```

        printf("\n\t\t\t\tDeadline          : %s", record.deadline);
        printf("\n\t\t\t\tDepartment        : %s", record.department);
        printf("\n\n\t\t\t\t-----\n");
    }
}

if (!found)
{
    printf("\n\t\t\t\t-----");
    printf("\n\t\t\t\tNo records found for project number %d.\n", searchProjNo);
    printf("\t\t\t\tPlease add a record.\n");
    printf("\t\t\t\t-----\n");
}

printf("\t\t\t\tDo you want to search another project number? (y/n): ");
searchAgain = getch();

} else
{
    printf("\n\t\t\t\tInvalid choice. Please enter 'y' or 'n'.\n");
}

}

fclose(fp);
getch();
}

```

```

void deleteproject()

```

```

{
    struct project record;

    FILE *fp, *fp1;

    int proj_no, found = 0;


    system("cls");

    printf("\n\t\t\t\t----- DELETE A PROJECT ----- \n");

    fp = fopen("information.txt", "r");
    fp1 = fopen("temp.txt", "w");

    printf("\t\t\t\t-----");

    printf("\n\t\t\t\tAVAILABLE RECORDS:\n");


    // Display project numbers and titles

    int recordCount = 0; // Track the number of records


    while (fscanf(fp, "Project Number: %d\nTitle: %[^\\n]\nProject Type: %[^\\n]\nAmount Project [PHP]: %f\nProject Status: %[^\\n]\nDeadline of the Project: %[^\\n]\nSchool Department: %[^\\n]\n\n",
        &record.proj_no, record.title, record.type, &record.amount, record.status, record.deadline,
        record.department) != EOF)
    {
        printf("\t\t\t\tPROJECT NUMBER:[%d] - TITLE:[%s]\n", record.proj_no, record.title);

        recordCount++;
    }


    rewind(fp); // Reset file pointer to the beginning


    printf("\t\t\t\t-----");


    if (recordCount == 0)

```

```

{
    printf("\n\t\t\tNo records found. Please add records to continue.\n");
    printf("\t\t\t-----\n");
    fclose(fp);
    fclose(fp1);
    getch();
    return;
}

printf("\n\t\t\tEnter project number: ");
scanf("%d", &proj_no);

if (fp == NULL)
{
    fprintf(stderr, "\n\t\t\tCan't open file\n");
    menu(1);
}

while (fscanf(fp, "Project Number: %d\nTitle: %[^\\n]\\nProject Type: %[^\\n]\\nAmount Project [PHP]: %f\\nProject Status: %[^\\n]\\nDeadline of the Project: %[^\\n]\\nSchool Department: %[^\\n]\\n\\n",
    &record.proj_no, record.title, record.type, &record.amount, record.status, record.deadline,
    record.department) != EOF)
{
    if (record.proj_no == proj_no)
    {
        found = 1;
    } else
    {
        fprintf(fp1, "Project Number: %d\\nTitle: %s\\nProject Type: %s\\nAmount Project [PHP]: %.2f\\nProject Status: %s\\nDeadline of the Project: %s\\nSchool Department: %s\\n\\n",

```

```
        record.proj_no, record.title, record.type, record.amount, record.status, record.deadline,
record.department);
```

```
    }
```

```
}
```

```
fclose(fp);
```

```
fclose(fp1);
```

```
if (!found)
```

```
{
```

```
    printf("\n\t\t\t\t\t-----\n");
```

```
    printf("\t\t\t\t\t ERROR. Record does not exist.\n");
```

```
    printf("\t\t\t\t\t-----\n");
```

```
}
```

```
if (found)
```

```
{
```

```
    remove("information.txt");
```

```
    rename("temp.txt", "information.txt");
```

```
    printf("\n\t\t\t\t\t-----\n");
```

```
    printf("\t\t\t\t\t The record was deleted successfully\n");
```

```
    printf("\t\t\t\t\t-----\n");
```

```
}
```

```
getch();
```

```
}
```

```
void editproject() {
```

```

struct project record;

FILE *fp, *fp1;

int proj_no, found = 0;

int choice;

char continue_editing;


system("cls");

printf("\n\t\t\t\t----- EDIT A PROJECT ----- \n");

fp = fopen("information.txt", "r");
fp1 = fopen("temp.txt", "w");


printf("\t\t\t\t-----");

printf("\n\t\t\t\tAVAILABLE RECORDS:\n");


// Display project numbers and titles

int record_count = 0;

while (fscanf(fp, "Project Number: %d\nTitle: %[^\\n]\nProject Type: %[^\\n]\nAmount Project [PHP]: %f\nProject Status: %[^\\n]\nDeadline of the Project: %[^\\n]\nSchool Department: %[^\\n]\n\n",
               &record.proj_no, record.title, record.type, &record.amount, record.status, record.deadline,
               record.department) != EOF) {

    record_count++;

    printf("\t\t\t\tPROJECT NUMBER:[%d] - TITLE:[%s]\n", record.proj_no, record.title);

}

rewind(fp); // Reset file pointer to the beginning

if (record_count == 0) {

    printf("\t\t\t\t-----");

    printf("\n\t\t\t\tNo records found. Add records to continue.\n");

```





```
do {  
    printf("\n\t\t\t-----\n");  
    printf("\t\t\t\t Choose the information to edit:\n");  
    printf("\t\t\t\t-----\n");  
    printf("\t\t\t1. Project Number\n");  
    printf("\t\t\t2. Title\n");  
    printf("\t\t\t3. Project Type\n");  
    printf("\t\t\t4. Amount of the Project\n");  
    printf("\t\t\t5. Project Status\n");  
    printf("\t\t\t6. Deadline of the Project\n");  
    printf("\t\t\t7. School Department of the Project\n");  
    printf("\t\t\t8. All Information\n");  
    printf("\n\t\t\tEnter your choice: ");  
    scanf("%d", &choice);  
    printf("\n\t\t\t-----");  
  
    switch (choice) {  
        case 1:  
            printf("\n\t\t\tEnter Project Number: ");  
            while (scanf("%d", &record.proj_no) != 1) {  
                printf("\n\t\t\tERROR: Invalid input for Project Number. Please try again and enter a  
number.\n");  
                fflush(stdin); // Clear the input buffer  
                printf("\n\t\t\tEnter Project Number: ");  
            }  
  
            // Check if there are repeating numbers  
            FILE* fp_check = fopen("information.txt", "r");  
            if (fp_check != NULL) {
```

```

        struct project existing;

        while (fscanf(fp_check, "Project Number: %d\nTitle: %[^\\n]\nProject Type:
%[^\\n]\nAmount Project [PHP]: %f\nProject Status: %[^\\n]\nDeadline of the Project: %[^\\n]\nSchool
Department: %[^\\n]\n\n",

                &existing.proj_no, existing.title, existing.type, &existing.amount, existing.status,
existing.deadline, existing.department) != EOF) {

            if (existing.proj_no == record.proj_no) {

                printf("\n\t\t\tERROR: Project number already exists. Please enter a different project
number.\n");

                fclose(fp_check);

                fflush(stdin);

                record.proj_no = -1; // Set an invalid project number to re-entering

                break;

            }

        }

        fclose(fp_check);

    }

    break;

case 2:

    printf("\n\t\t\tEnter Title: ");

    scanf(" %[^\\n]", record.title);

    break;

case 3:

    printf("\n\t\t\tEnter Project Type: ");

    scanf(" %[^\\n]", record.type);

    break;

case 4:

    printf("\n\t\t\tEnter the Amount of the Project (Php): ");

    while (scanf("%f", &record.amount) != 1) {

```

```

        printf("\n\t\tERROR: Invalid input for Amount of the Project. Please enter a
number.\n");

        fflush(stdin); // Clear the input buffer

        printf("\n\t\tEnter the Amount of the Project: ");

    }

    break;

case 5:

    printf("\n\t\tEnter Project Status (In Progress, Completed, Cancelled): ");

    scanf("%[^\n]", record.status);

    break;

case 6:

    printf("\n\t\tEnter the Deadline of the Project: ");

    scanf("%[^\n]", record.deadline);

    break;

case 7:

    printf("\n\t\tEnter the School Department of the Project: ");

    scanf("%[^\n]", record.department);

    break;

case 8:

    printf("\n\t\tEnter Project Number: ");

    while (scanf("%d", &record.proj_no) != 1) {

        printf("\n\t\tERROR: Invalid input for Project Number. Please try again and enter a
number.\n");

        fflush(stdin); // Clear the input buffer

        printf("\n\t\tEnter Project Number: ");

    }

    // Check if there are repeating numbers

    FILE* fp_check_all = fopen("information.txt", "r");

```

```

    if (fp_check_all != NULL) {

        struct project existing;

        while (fscanf(fp_check_all, "Project Number: %d\nTitle: %[^\\n]\\nProject Type:
%[^\\n]\\nAmount Project [PHP]: %f\\nProject Status: %[^\\n]\\nDeadline of the Project: %[^\\n]\\nSchool
Department: %[^\\n]\\n\\n",

                    &existing.proj_no, existing.title, existing.type, &existing.amount, existing.status,
existing.deadline, existing.department) != EOF) {

            if (existing.proj_no == record.proj_no) {

                printf("\\n\\t\\t\\tERROR: Project number already exists. Please enter a different project
number.\\n");

                fclose(fp_check_all);

                fflush(stdin);

                record.proj_no = -1; // Set an invalid project number to re-entering

                break;

            }

        }

        fclose(fp_check_all);

    }

    printf("\\n\\t\\t\\tEnter Title: ");

    scanf(" %[^\\n]", record.title);

    printf("\\n\\t\\t\\tEnter Project Type: ");

    scanf(" %[^\\n]", record.type);

    printf("\\n\\t\\t\\tEnter the Amount of the Project (Php): ");

    while (scanf("%f", &record.amount) != 1) {

        printf("\\n\\t\\t\\tERROR: Invalid input for Amount of the Project. Please enter a
number.\\n");

        fflush(stdin); // Clear the input buffer

        printf("\\n\\t\\t\\tEnter the Amount of the Project (Php): ");

    }

```

```

printf("\n\t\t\tEnter Project Status (In Progress, Completed, Cancelled): ");
scanf("%[^\\n]", record.status);

printf("\n\t\t\tEnter the Deadline of the Project: ");
scanf("%[^\\n]", record.deadline);

printf("\n\t\t\tEnter the School Department of the Project: ");
scanf("%[^\\n]", record.department);

break;

default:

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

break;

}

printf("\t\t\t-----");

printf("\n\t\t\tDo you want to edit any other information? (y/n): ");

scanf(" %c", &continue_editing);

} while (continue_editing == 'y' || continue_editing == 'Y');

```

```

fprintf(fp1, "Project Number: %d\\nTitle: %s\\nProject Type: %s\\nAmount Project [PHP]:
%.2f\\nProject Status: %s\\nDeadline of the Project: %s\\nSchool Department: %s\\n\\n\\n",
        record.proj_no, record.title, record.type, record.amount, record.status, record.deadline,
        record.department);

```

```

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

printf("\\n\\t\\t\\t\\t\\t\\t\\tNew Project Details:\\n");

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

printf("\\t\\t\\tProject Number      : %d\\n", record.proj_no);

printf("\\t\\t\\tTitle                : %s\\n", record.title);

printf("\\t\\t\\tProject Type          : %s\\n", record.type);

printf("\\t\\t\\tAmount of the Project: %.2f\\n", record.amount);

printf("\\t\\t\\tProject Status       : %s\\n", record.status);

```

```

        printf("\t\tDeadline of the Project: %s\n", record.deadline);

        printf("\t\tDepartment of the Project: %s\n", record.department);

    } else {

        fprintf(fp1, "Project Number: %d\nTitle: %s\nProject Type: %s\nAmount Project [PHP]:
%.2f\nProject Status: %s\nDeadline of the Project: %s\nSchool Department: %s\n\n",

            record.proj_no, record.title, record.type, record.amount, record.status, record.deadline,
            record.department);

    }

}

fclose(fp);

fclose(fp1);

if (found == 0)

{

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

    printf("\n\t\tNo records found for the project number %d.\n", proj_no);

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

    remove("temp.txt");

} else

{

    remove("information.txt");

    rename("temp.txt", "information.txt");

    printf("\t\t-----");

    printf("\n\t\t\t\t The record was updated successfully\n");

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

}

}

```