Activity 7:Storing Student Mark

Name:Krushna Bhausaheb Kokane
Class:CE(2)
Roll no:65

Research:

The purpose of this research is to understand how file handling in the C programming language can be used to store and manage student information permanently. In many educational environments, student marks are recorded manually, which increases the chances of errors, data loss, and difficulty in organizing or retrieving information. Through this project, file handling is studied as a reliable method for storing data in a structured and accessible way.

The research focuses on the use of text files (`.txt`) to store student names and marks using C functions such as `fprintf()` for writing data and `fscanf()` for reading data. These functions allow the program to save information even after the program ends, demonstrating the concept of persistent storage. This is especially important in small systems where a database is not required, but data must still be saved safely.

Additionally, the research explores how basic file operations—such as opening, reading, writing, and closing files—form the foundation of larger data management systems. It helps understand how simple text-based storage can be

expanded into more advanced systems like databases or management applications in the future.

Overall, this research highlights the importance of file handling in building small yet effective data storage applications and shows how C programming can be used to create a simple Student Marks Management System.

ideate:

In this phase, different ideas were explored to design a simple system that can store and manage student marks using C programming. The goal was to find an easy, practical, and efficient method for keeping student records without using complex databases.

Several ideas were considered:

1. **Storing data temporarily using arrays**
   This method is simple but the data is lost when the program ends. So it was not suitable.

2. **Using structures to store name and marks**
   This helps organize data but still does not provide permanent storage unless combined with file handling.

3. **Using file handling to store data permanently**
   This idea allows saving student information even after the program is closed. It is simple, supports large data, and is easy to retrieve later.
   This became the final chosen idea.

4. **Using menu-driven system**
   Ideas like adding search, update, delete options were considered but kept aside to maintain simplicity for the

project.

After comparing all ideas, the most effective solution was to use file handling with simple write and read operations. This idea allows the system to:

- Store data permanently
- Display student records anytime
- Keep the program simple and easy to understand

Thus, the ideation process led to building a Student Marks Management System using File Handling as the best approach for this project.

Analyse:

The analysis of this project focuses on understanding how student information can be stored, retrieved, and displayed using file handling in C. The main goal is to examine the requirements, the process, and the behavior of the system.

The system fulfills the basic goal of storing and displaying student data.

It demonstrates the practical use of file handling in C and forms the foundation for more advanced student management systems.

Build:

```
#include<stdio.h>
int main(){
char name[100];
int marks;
int num,i,j,p,n;
FILE*f;
f=fopen("my_file.txt","w");
printf("enter number of students:");
scanf("%d",&num);
for(i=0;i<num;i++){
printf("enter name of student %d:",i+1);
scanf("%s",&name);
fprintf(f,"%s\n",name);
printf("enter marks of student:");
scanf("%d",&marks);
fprintf(f,"%d\n",marks);
}
fclose(f);
printf("\n----------------------\n");
printf("student list:\n");
f=fopen("my_file.txt","r");
for(j=0;j<num;j++){
fscanf(f,"%s",&name);
fscanf(f,"%d",&marks);
printf("%d)%s:%d\n",j+1,name,marks);
}
return 0;
}
```

Testing:

```
enter number of students:3
enter name of student 1:pushkar
enter marks of student:34
enter name of student 2:rudraksh
enter marks of student:78
enter name of student 3:krushna
enter marks of student:89
```

---
student list:
1)pushkar:34
2)rudraksh:78
3)krushna:89