

## EXPERIMENT-14

Construct a C program to organise the file using a single level directory.

### AIM:-

To construct a C program to organize files using a single-level directory structure.

### ALGORITHM:-

1. Initialize Directory:
2. Create a structure to represent the directory, including file names and a count of files.
3. Menu Options:
4. Provide options to create, delete, search, and display files in the directory.
5. Add Files:
6. Prompt the user to enter the file name and add it to the directory.
7. Delete Files:
8. Search for the file in the directory and delete it if found.
9. Search Files:
10. Check if a given file exists in the directory.
11. Display Files:
12. Print the list of all files in the directory.

### CODE:-

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#define MAX_FILES 100
```

```
struct Directory {
```

```
    char files[MAX_FILES][50];
```

```
int fileCount;

};

void addFile(struct Directory* dir, char* fileName) {

    if (dir->fileCount < MAX_FILES) {

        strcpy(dir->files[dir->fileCount], fileName);

        dir->fileCount++;

        printf("File '%s' added successfully.\n", fileName);

    } else {

        printf("Directory is full. Cannot add more files.\n");

    }

}
```

```
void deleteFile(struct Directory* dir, char* fileName) {

    int found = 0;

    for (int i = 0; i < dir->fileCount; i++) {

        if (strcmp(dir->files[i], fileName) == 0) {

            found = 1;

            for (int j = i; j < dir->fileCount - 1; j++) {

                strcpy(dir->files[j], dir->files[j + 1]);

            }

            dir->fileCount--;

            printf("File '%s' deleted successfully.\n", fileName);

            break;

        }

    }

}
```

```

    }

}

if (!found) {

    printf("File '%s' not found in the directory.\n", fileName);

}

}

```

```

void searchFile(struct Directory* dir, char* fileName) {

    for (int i = 0; i < dir->fileCount; i++) {

        if (strcmp(dir->files[i], fileName) == 0) {

            printf("File '%s' is present in the directory.\n", fileName);

            return;

        }

    }

    printf("File '%s' not found in the directory.\n", fileName);

}

```

```

void displayFiles(struct Directory* dir) {

    if (dir->fileCount == 0) {

        printf("Directory is empty.\n");

    } else {

        printf("Files in the directory:\n");

        for (int i = 0; i < dir->fileCount; i++) {

            printf("%d. %s\n", i + 1, dir->files[i]);

        }

    }

}

```

```
    }  
}  
}
```

```
int main() {  
  
    struct Directory dir;  
  
    dir.fileCount = 0;  
  
    int choice;  
  
    char fileName[50];  
  
    do {  
  
        printf("\n--- Single Level Directory Menu ---\n");  
  
        printf("1. Add File\n");  
  
        printf("2. Delete File\n");  
  
        printf("3. Search File\n");  
  
        printf("4. Display Files\n");  
  
        printf("5. Exit\n");  
  
        printf("Enter your choice: ");  
  
        scanf("%d", &choice);  
  
        switch (choice) {  
  
            case 1:  
  
                printf("Enter file name to add: ");
```

```
    scanf("%s", fileName);

    addFile(&dir, fileName);

    break;

case 2:

    printf("Enter file name to delete: ");

    scanf("%s", fileName);

    deleteFile(&dir, fileName);

    break;

case 3:

    printf("Enter file name to search: ");

    scanf("%s", fileName);

    searchFile(&dir, fileName);

    break;

case 4:

    displayFiles(&dir);

    break;

case 5:

    printf("Exiting the program.\n");

    break;

default:

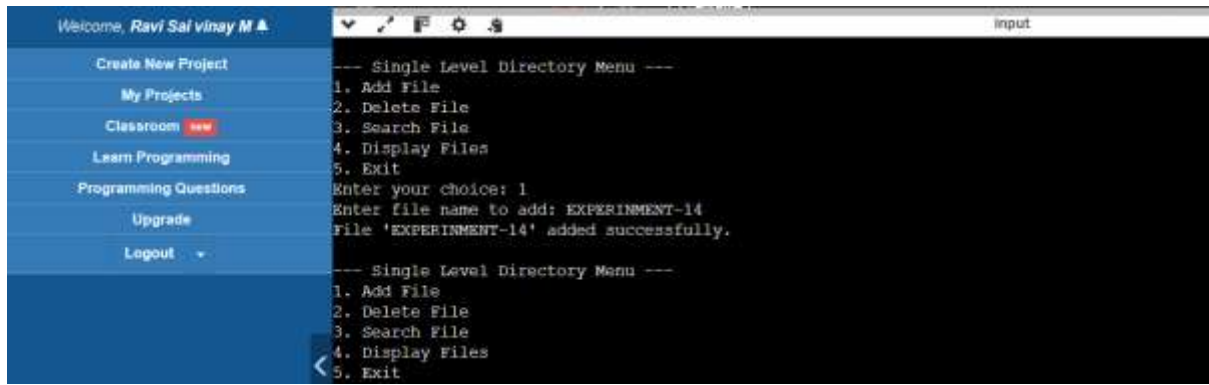
    printf("Invalid choice. Please try again.\n");

}

} while (choice != 5);
```

```
return 0;  
  
}
```

## OUTPUT:-



The screenshot shows a web application with a blue sidebar menu on the left and a terminal window on the right. The sidebar menu includes options like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The terminal window displays a 'Single Level Directory Menu' with options: 1. Add File, 2. Delete File, 3. Search File, 4. Display Files, and 5. Exit. The user has entered '1' for 'Add File', then 'EXPERINMENT-14' as the file name, and the terminal confirms 'File 'EXPERINMENT-14' added successfully.' The menu is then shown again.

```
Welcome, Ravi Sai vinay M ▲  
Create New Project  
My Projects  
Classroom new  
Learn Programming  
Programming Questions  
Upgrade  
Logout ▾  
  
--- Single Level Directory Menu ---  
1. Add File  
2. Delete File  
3. Search File  
4. Display Files  
5. Exit  
Enter your choice: 1  
Enter file name to add: EXPERINMENT-14  
File 'EXPERINMENT-14' added successfully.  
  
--- Single Level Directory Menu ---  
1. Add File  
2. Delete File  
3. Search File  
4. Display Files  
5. Exit
```

## RESULT:-

The program successfully simulates a single-level directory structure. It allows adding, deleting, searching, and displaying files, effectively organizing them in a simple directory structure