NBKR INSTITUTE OF SCIENCE AND TECHNOLOGY

MINI FILE EXPLORER

PRESENTED BY

M.Devika(24KB1A05AS)

M.Jeswitha(24KB1A05AV)

N.Sreeja(24KB1A05BK)

P.Padma(24KB1A05ES)

INTRODUCTION

- The Mini File Explorer is a lightweight application designed to provide users with a simple and efficient way to navigate, view, and manage their files and folders. Unlike full-scale file management systems, this mini explorer focuses on core functionalities such as browsing directories, opening files, and performing basic file operations like create, delete, and rename. It offers an intuitive interface, making it ideal for quick file access and organization without the complexity of advanced features. This project also serves as a foundation for learning file system handling and user interface design in software development.

OBJECTIVE

- The objective of the Mini File Explorer is to develop a simple and user-friendly tool that allows users to browse, manage, and organize files and folders efficiently. The project aims to implement basic file operations such as viewing directory structures, opening files, creating new folders, renaming, and deleting items. It focuses on providing a lightweight, responsive interface while enhancing understanding of file system navigation and event-driven programming concepts.

IMPORTANCE OF C AND DSA IN MINI FILE EXPLORER

- C is a powerful low-level programming language that provides direct access to memory and system resources, making it ideal for building efficient and fast applications like a Mini File Explorer. Using C allows developers to closely manage file system operations, memory usage, and performance.Data structures, such as trees, stacks, queues, and linked lists, play a crucial role in organizing and managing file and folder hierarchies effectively.

ALGORITHM

1.Start the program and initialize a list to store file names.

2. Show a menu: Show Files, Search File, Add File, Delete File, Exit.

3. Get user input.

4. Perform action:

    Show Files: Display all file names or "No files".

    Search File: Search and display result (found/not found).

    Add File: Add new file name to the list.

    Delete File: Remove file name if it exists.

5.Repeat until user selects exists.

6. End program.

C program link:https://onlinegdb.com/za1Dy9GMW

LESSON LEARNT

1.State Management is Key

Keeping track of the current path, selected files, and opened folders needs a clean, centralized state.Proper use of trees or recursion simplifies nested structure.

2. Error Handling

Necessary to handle missing files, permissions errors, and corrupted folders gracefully.

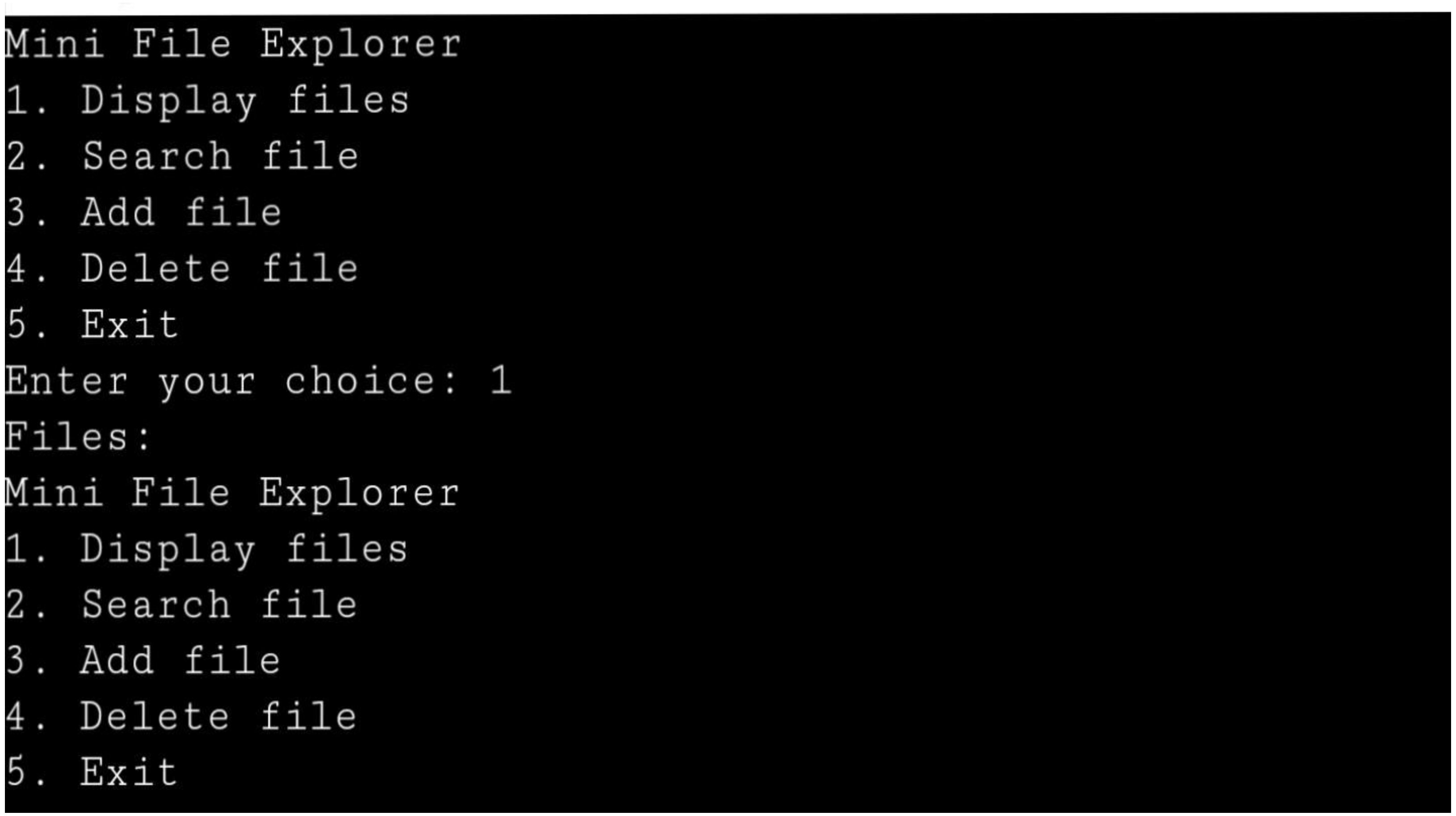
3.Testing and Debugging

Mock data is useful for testing without risking real files.Visual debugging (e.g., logging the current folder tree) can help.

4. Scalability in Design

Design components so that adding features like "file preview" or "search" later is easy.

Output images:



```
Mini File Explorer
1. Display files
2. Search file
3. Add file
4. Delete file
5. Exit
Enter your choice: 1
Files:
Mini File Explorer
1. Display files
2. Search file
3. Add file
4. Delete file
5. Exit
```

CONCLUSION

- The Mini File Explorer project provided valuable insights into handling file and folder structures in a user-friendly and efficient way. By focusing on core features like navigation, file management, and accessibility, it became clear that creating an intuitive and responsive interface is critical for user satisfaction. Implementing key functionalities such as right-click options, drag-and-drop support, and folder expansion/collapsing significantly enhanced the usability and interactivity of the tool.

Thank you!