DLITHE PROJECT REPORT

PROJECT ID : CP060

PROJECT TITLE : RESEARCH TRACKER

TEAM MEMBERS : Srijan K **(4MT21AI057)**

Aniruddha **(4MT21AI042)**

Anshad Aziz **(4MT21AI009)**

Hitha B Mendon **(4MT21AI021)**

Chayasri K **(4MT21AI014)**

REPORT

1. Abstract:

The Research Tracker is a comprehensive application designed to streamline the management of research papers published by MITE (MITE Institute of Technology and Education). This system enables users to effortlessly maintain a database of research papers, facilitating efficient storage, retrieval, and organization. Users can easily access and browse all the files associated with a specific research project, and upon selection, view the content of individual files, ensuring seamless access to valuable research materials. This project aims to enhance accessibility and organization in the academic research domain, providing a user-friendly platform for researchers and educators affiliated with MITE.

1. Introduction

1.1. Background:

- The MITE (MITE Institute of Technology and Education) is actively involved in research activities across various disciplines.
- The volume of research papers generated by MITE has been steadily increasing over the years.
- Traditional methods of paper and file management have become cumbersome and inefficient for tracking and accessing research materials.
- The need for a centralized and user-friendly system to manage and access research papers has become imperative to support MITE's research endeavors.

1.2. Objectives:

- Develop a Research Tracker application to centralize the storage and management of research papers published by MITE.
- Enable easy and efficient browsing of all files associated with each research project.
- Provide a user-friendly interface for researchers, faculty, and students to access and retrieve research materials.

- Improve accessibility, organization, and searchability of research papers within MITE.
- Streamline the process of adding,viewing, updating, and deleting research papers from the system.
- Enhance collaboration among researchers by facilitating seamless access to shared research documents.
- Ensure data security and privacy measures are in place to protect sensitive research information.

2. Technologies Used

- Programming Languages: C
- File Handling for data storage

3. System Architecture:

The program's architecture can be summarized as a standalone console application that handles data input and output and manages research papers locally on the user's machine.

4.1. Front-End:

The user interface is text-based and operates through a command-line interface (CLI).

It uses standard input and output to communicate with the user, displaying menus and receiving user input for various operations.

The front-end provides options for adding research papers, viewing the list of research papers, viewing the content of individual research papers, and exiting the program.

4.2. Back-End:

The back-end logic of the program is written in C and is responsible for managing research paper data and performing operations on that data.

It includes functions for writing research papers to a file, reading research papers from a file, adding new research papers, and viewing research paper information.

The back-end also manages the creation and storage of individual research paper files.

4.3. Database:

In this context, there is no traditional database system like MySQL or SQLite. Instead, research paper metadata (titles, authors, file paths) and content are stored as structured text data in local files.

Each research paper is represented by a text file named after its title, and the metadata is stored within these files.

These text files serve as a basic form of data storage and retrieval, similar to a file-based database.

5. Project Modules

5.1 Module 1: Research Paper Management

- Function: This module is responsible for managing the core functionality of the Research Tracker application.
- Description: It allows users to add new research papers to the system, view the list of existing research papers, and view the content of individual research papers. It also handles the reading and writing of research paper data to and from files.
- Interactions: This module interacts with the user interface (front-end) to receive user input and display information. It also interacts with Module 2 (File Management) for creating and managing individual research paper files.

5.2. Module 2: File Management

- Function: This module is responsible for managing the creation and storage of individual research paper files.
- Description: When a new research paper is added, this module creates a text file named after the paper's title, stores metadata (title, authors, file path) within the file, and allows users to input and store the content of the research paper in that file.
- Interactions: Module 1 (Research Paper Management) communicates with this module to create and update research paper files. It also interacts with Module 1 to provide file paths for viewing research paper content. Additionally, it interacts with the user interface to receive content input.

6. Design and Implementation

6.1. Front-End Design:

- User Interface: The front-end design focuses on creating a user-friendly command-line interface (CLI) for the Research Tracker application. It utilizes text-based menus and prompts to interact with users.
- Menu System: The user interface presents a menu with options to add research papers, view research papers, view file content, and exit the program. It uses printf and scanf functions for input and output.
- User Input Handling: Input from users, such as research paper titles, authors, file paths, and content, is handled using scanf for strings and integers. The program ensures proper input validation and handling of newline characters.
- Output Presentation: The front-end presents research paper information in a structured and readable format, displaying titles, authors, and file paths in a user-friendly manner.

6.2 Back-End Design:

- Functionality: The back-end design focuses on the core functionality of the Research Tracker application. It manages data, handles user requests, and performs file operations.
- Data Structures: The primary data structure used is an array of structs (struct Research Paper) to store research paper metadata (title, authors, file path). This array allows for the management of multiple research papers.
- File Operations: The back-end handles file operations, including reading and writing research paper data to and from a local text file ("research_papers.txt"). It also creates individual research paper files with their respective content.
- Data Flow: The back-end communicates with the front-end to receive user input and display information. It also interacts with the File Management module to create and manage research paper files.

6.3 Database Design:

- File-Based Database: In this simple system, there is no traditional database management system. Instead, the design relies on text files for data storage.
- Data Organization: Each research paper is represented by a text file named after its title. Within these files, metadata (title, authors, file path)

is stored in a structured format. Content is stored in the same file after the metadata.

- Data Retrieval: To retrieve data, the program reads and parses these text files. When viewing file content, it reads the content from the specific research paper's file.
- Data Storage and Maintenance: The program manages data consistency by ensuring that updates (adding new research papers) are reflected in the metadata files. The "research_papers.txt" file acts as a central index for research papers.

7. Features and Functionality

7.1 Feature 1: Adding Research Papers

- Functionality: Users can add new research papers to the system through a user-friendly command-line interface.
- Details: Users are prompted to enter the title, authors, and file path of the research paper.
- A new text file is created for the paper, with the title as its name, and metadata (title, authors, file path) are stored within this file.
- Users can input the content of the research paper directly into the file.
- The system enforces a maximum limit on the number of papers that can be added.

7.2 Feature 2: Viewing Research Papers

- Functionality: Users can view a list of existing research papers and their details.
- Details: The program displays a list of research papers, including their titles, authors, and file paths.
- Users can select a specific research paper from the list to view more details.
- The detailed view includes the title, authors, and file path of the selected paper.

7.3 Feature 3: Viewing File Content

- Functionality: Users can view the content of individual research papers.
- Details: Users select a specific research paper from the list.
- The program reads and displays the content of the selected research paper's associated text file.
- The content is presented in the console, allowing users to read the research paper.

7.4 Feature 4: Data Persistence

- Functionality: The program ensures that research paper data is stored and can be accessed across different program sessions.
- Details: Data, including research paper metadata (titles, authors, file paths) and content, is stored in text files on the local machine.
- Existing data is loaded when the program starts, allowing users to work with previously added research papers.
- Data is saved to files when new research papers are added or when the program exits to ensure persistence.

7.5 Feature 5: Error Handling

- Functionality: The program includes error-handling mechanisms to handle unexpected situations.
- Details:
- The program checks for errors when opening, creating, and reading files, providing error messages to the user when issues occur.
- Input validation is implemented to handle invalid user input, ensuring the program's stability and robustness.

8.Challenges Faced:

- User Interface Complexity: Designing a user-friendly command-line interface that is intuitive and accessible to users with varying levels of technical expertise posed a challenge.
- File Handling: Managing file operations, such as creating and updating files, reading and writing content, and ensuring data consistency, required careful implementation to avoid errors and data loss.
- Error Handling: Implementing robust error-handling mechanisms to gracefully handle unexpected situations and provide informative error messages was a challenge.

9.Future Enhancements:

- Search and Filter Functionality: Implement a search feature to allow users to search for specific research papers by keywords or authors. Add filtering options to sort research papers by criteria such as date or author.
- File Attachment: Allow users to attach files (PDFs, images, etc.) to research papers, providing a comprehensive repository for research materials.

- Collaboration Features: Enable multiple users to collaborate on research papers, with features like version control, comments, and access control.
- Database Integration: Replace the current file-based storage with a database system for improved scalability, performance, and data management.

10. Conclusion:

The Research Tracker program provides a functional solution for managing research papers within MITE. It offers features for adding, viewing, and accessing research papers along with error handling and data persistence. While there were challenges in designing the user interface and handling file operations, the program successfully fulfills its purpose. Future enhancements can further expand its capabilities and usability.

References:

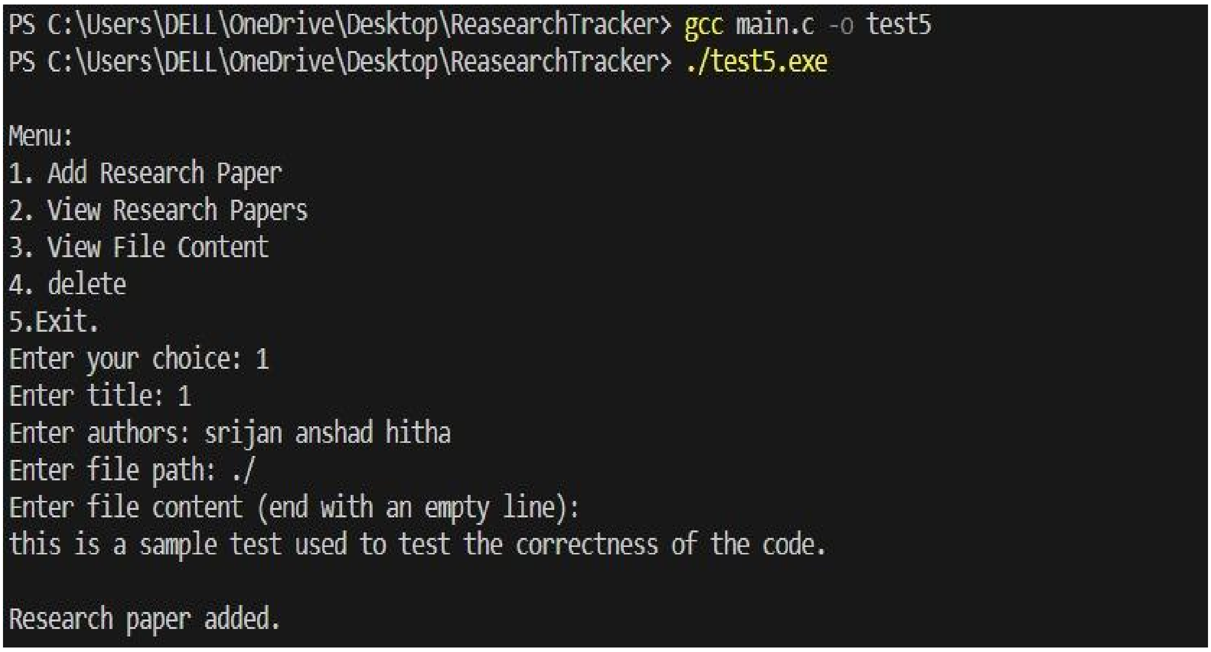
C Programming Language Documentation

File Handling in C

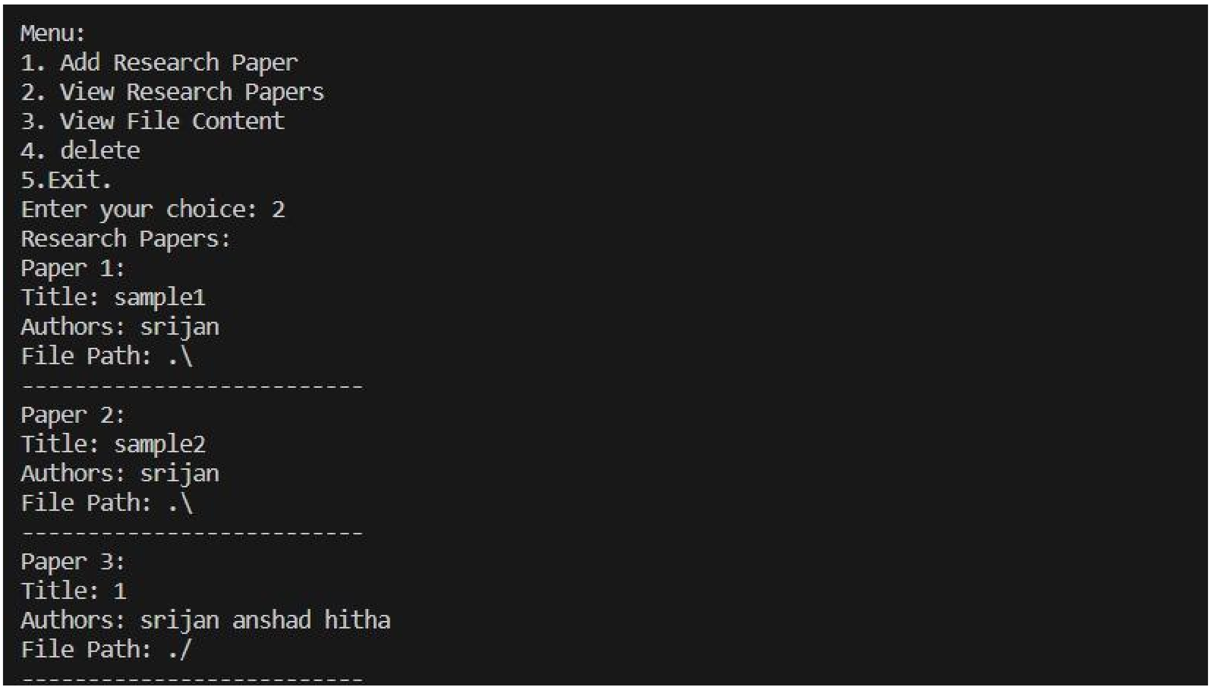
User Acceptance Testing (UAT)

11.Appendices:

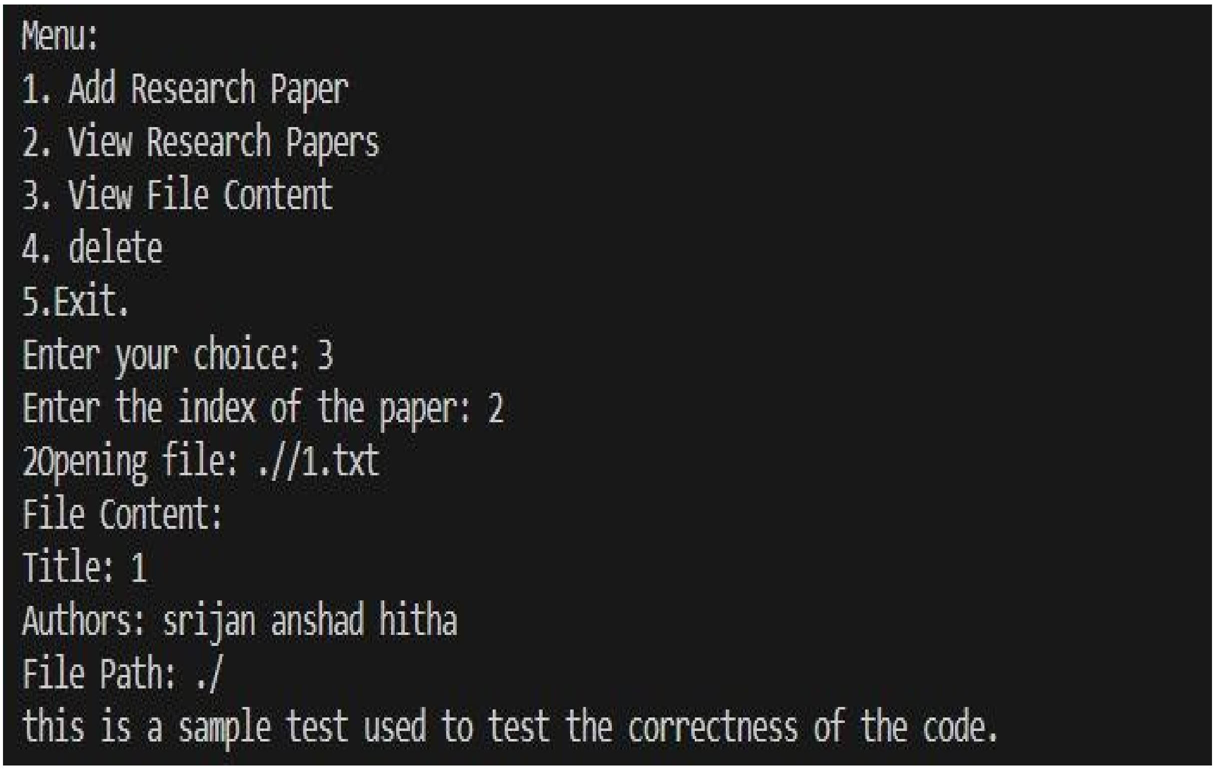
Screenshots



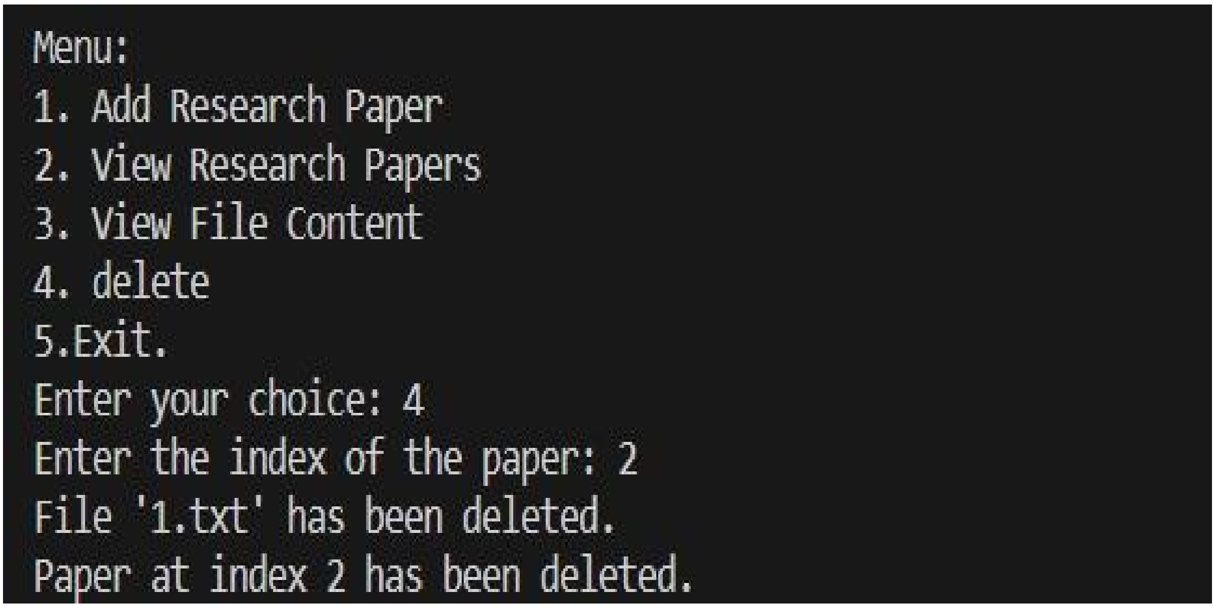
```
PS C:\Users\DELL\OneDrive\Desktop\ReasearchTracker> gcc main.c -o test5
PS C:\Users\DELL\OneDrive\Desktop\ReasearchTracker> ./test5.exe

Menu:
1. Add Research Paper
2. View Research Papers
3. View File Content
4. delete
5.Exit.
Enter your choice: 1
Enter title: 1
Enter authors: srijan anshad hitha
Enter file path: ./
Enter file content (end with an empty line):
this is a sample test used to test the correctness of the code.

Research paper added.
```



```
Menu:
1. Add Research Paper
2. View Research Papers
3. View File Content
4. delete
5.Exit.
Enter your choice: 2
Research Papers:
Paper 1:
Title: sample1
Authors: srijan
File Path: .\
--------------------------
Paper 2:
Title: sample2
Authors: srijan
File Path: .\
--------------------------
Paper 3:
Title: 1
Authors: srijan anshad hitha
File Path: ./
--------------------------
```



```
Menu:
1. Add Research Paper
2. View Research Papers
3. View File Content
4. delete
5.Exit.
Enter your choice: 3
Enter the index of the paper: 2
2Opening file: .//1.txt
File Content:
Title: 1
Authors: srijan anshad hitha
File Path: ./
this is a sample test used to test the correctness of the code.
```



```
Menu:
1. Add Research Paper
2. View Research Papers
3. View File Content
4. delete
5.Exit.
Enter your choice: 4
Enter the index of the paper: 2
File '1.txt' has been deleted.
Paper at index 2 has been deleted.
```