

# Report for DCR-B Part I Project

Pegah Moradpour

---

## Introduction

The primary objective of this project is to develop a robust search facility for a subtree of a local file system. The search functionality must support both file name and content searches within a specified directory structure. This report outlines the implementation, database schema, loading process, search engine functionality, and the results of predefined search scenarios.

## Project Requirements

### Search Functionality:

The system must match both file names and contents.

Results should include the file's full path, type, and the number of occurrences of the search term within the content.

### Directory Structure:

The subtree must have a depth of at least six levels.

It must include a specific subtree named "DCRB" with a depth of at least four levels, containing at least 50 Wikipedia files distributed across at least four levels.

### Persistence:

The system should index necessary fields for efficient search operations.

The database must store metadata about the files, including their paths, names, types, sizes, and content.

## Database Schema

The database schema comprises two primary tables: `files_info` and `search_results_user`.

### `files_info` Table:

Stores metadata about each file, including:

`id`: Auto-incremented unique identifier.

`file_name`: Name of the file.

full\_path: Full path of the file, unique for each file.

file\_extension: File extension (e.g., .txt, .html).

file\_size: Size of the file.

file\_type: Type of the file (e.g., Directory, File).

content: Full content of the file.

Indexed fields for efficient searching:

file\_name

full\_path

content (FULLTEXT index for efficient text searching).

### **search\_results\_user Table:**

Stores the results of user search queries, including:

id: Auto-incremented unique identifier.

file\_name: Name of the file.

full\_path: Full path of the file.

occurrences: Number of occurrences of the search term in the content.

file\_type: Type of the file.

Implementation Details

### **Database Loading:**

The database is initialized with metadata from the specified directory.

The directory is continuously monitored, and any new or updated files are automatically inserted into the files\_info table.

The system uses transactions to ensure data consistency and rollback in case of errors.

### **Search Engine:**

Users can search for terms within both file names and contents.

Search results are stored in the search\_results\_user table.

Each new search query truncates the previous results, maintaining the current search context.

**Primary Keys and Indexing:**

files\_info Table: Composite primary key on id and file\_name ensures uniqueness.

search\_results\_user Table: Composite primary key on id and file\_name ensures uniqueness.

**User Interface:**

Provides options to search for a term or exit the application.

Prompts the user to enter a search term, performs the search, and displays the results.

**Summary of Implemented Features****Database Initialization:**

A database named file\_storage is created with two tables: files\_info and search\_results\_user.

The files\_info table stores detailed metadata of files, including their contents for full-text searching.

**Directory Monitoring:**

The specified directory (D:\1) is continuously monitored.

New or updated files are detected and their metadata is inserted into the files\_info table.

**Search Functionality:**

Users can perform searches on both file names and contents.

Search results are stored in the search\_results\_user table and displayed to the user.

The occurrences of the search term in file contents are accurately counted and stored.

**User Interface:**

Provides an interactive interface for users to perform searches or exit the application.

**Conclusion**

The developed system efficiently indexes and searches file names and contents within a specified directory structure. The results are stored in a dedicated search results table, ensuring that users can easily retrieve and view their search queries. The project adheres to the specified requirements and provides a robust solution for file searching on a local file system.