NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (KARACHI CAMPUS)
FAST School of Computing
Fall 2024

Project:

DataVault

Database Management System - DBMS

Group Members:

Muhammad Ali (Group Leader)	-	Member ID (Roll# 24K-3103)
Muhammad Hassan	-	Member ID (Roll#24K-3098)

Introduction

This project is a basic Database Management System (DBMS) built using the C programming language. The system is designed to simulate the core functionalities of a relational database, such as creating tables, adding records, updating data, deleting records, and viewing information. It uses CSV files to store data, with the first row acting as the schema to define attributes and the subsequent rows storing records.

The purpose of this system is to provide a simplified approach to understanding how databases work internally, focusing on key operations like CRUD (Create, Read, Update, Delete) and schema management. By implementing file handling and dynamic memory allocation, the project aims to mimic the behavior of modern database systems in a lightweight and educational manner.

Motivation

The motivation for building this DBMS from scratch is to gain a deeper understanding of the underlying principles of database design and operation. Instead of relying on prebuilt tools, this project offers a hands-on learning experience with concepts like file management, data indexing, and error handling in a programming context. This foundational knowledge can be a stepping stone for developing more advanced systems in the future.

System Overview

The system is designed as a simplified relational database, using a structured approach to manage data. It consists of multiple components working together, including schema handling, record management, and a user-friendly menu interface. The design is modular, with each part focusing on a specific task, making the system easy to use and maintain.

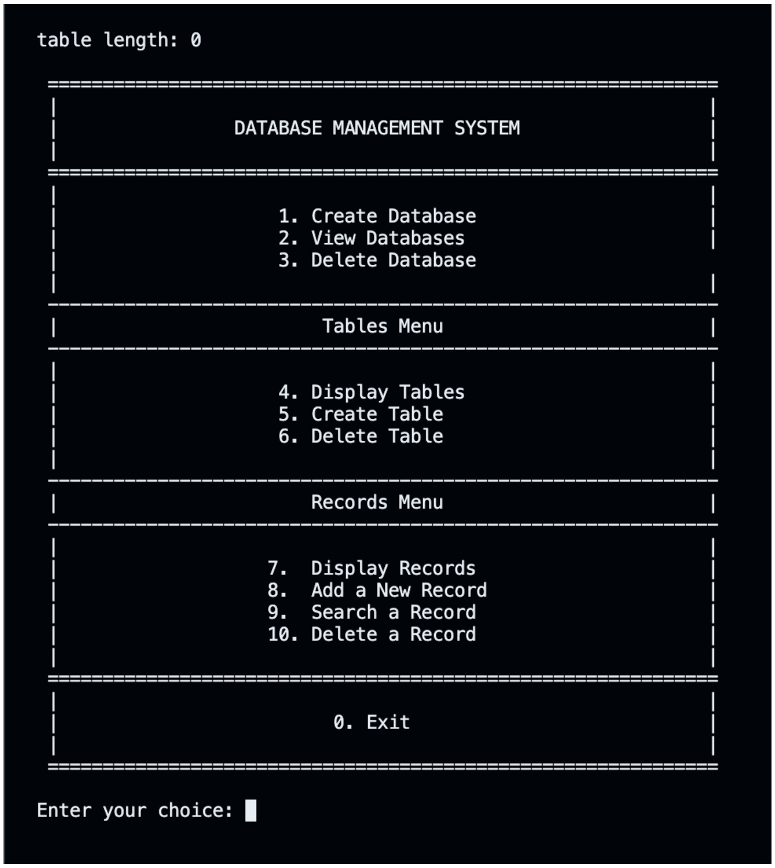
In this Database Management System, databases are represented as folders on the computer. Each folder acts as a container that holds tables, making it easy to organize and manage data for different projects or purposes. Within these folders, tables are stored as CSV files. These files contain rows of data, with the first row specifying the table's schema, which lists the attributes or columns of the table.

Records in each table are represented as arrays. Each array index stores the data of a single row, where the values correspond to the attributes defined in the schema. When working together, the system reads and writes data between these arrays and the CSV files, ensuring all changes are properly updated and saved.

In our system Databases group the related tables, tables define the structure of the data, and records provide the actual stored information. The design makes it easy to perform CRUD operations like adding, updating, or deleting data while maintaining a clear and simple file-based representation.

User Menu

The user menu shown in the image serves as the main navigation interface for the Database Management System (DBMS). It is designed with a clear structure, dividing operations into three primary categories: Database Management, Table Operations, and Record Operations. Each category contains related options, ensuring a logical flow for the user.



```
table length: 0

 =============================================================
 |                                                           |
 |                DATABASE MANAGEMENT SYSTEM                 |
 |                                                           |
 =============================================================
 |                                                           |
 |                    1. Create Database                     |
 |                    2. View Databases                      |
 |                    3. Delete Database
 |                                                           |
 -------------------------------------------------------------
 |                        Tables Menu                        |
 -------------------------------------------------------------
 |                                                           |
 |                    4. Display Tables                      |
 |                    5. Create Table                        |
 |                    6. Delete Table                        |
 |                                                           |
 -------------------------------------------------------------
 |                       Records Menu                        |
 -------------------------------------------------------------
 |                                                           |
 |                   7.  Display Records                     |
 |                   8.  Add a New Record                    |
 |                   9.  Search a Record                     |
 |                   10. Delete a Record                     |
 |                                                           |
 =============================================================
 |                                                           |
 |                         0. Exit                           |
 |                                                           |
 =============================================================

Enter your choice: █
```

1. **Database Management**
   - Options 1 to 3 allow users to create, view, and delete databases. These options enable users to manage their database directories efficiently.
2. **Table Operations**
   - Options 4 to 6 are focused on table management within the selected database. Users can display existing tables, create new ones, and delete tables.
3. **Record Operations**
   - Options 7 to 10 handle the records stored within a table. Users can display all records, add new entries, search a specific record, and delete specific records.
4. **Exit Option**
   - Option 0 allows the user to exit the system, providing a simple way to terminate the program.

System Architecture

The system is divided into several components, each responsible for specific tasks:

1. **Menu Operations Module**: This module handles all user interface-related operations. It displays menus, takes user inputs, and directs the program to the appropriate functionality, like creating, viewing, and deleting databases.

2. **Library Functions Module**: This module contains utility functions used across the system, such as handling file reading and writing, string manipulation, and other common tasks that support the main operations of the program.

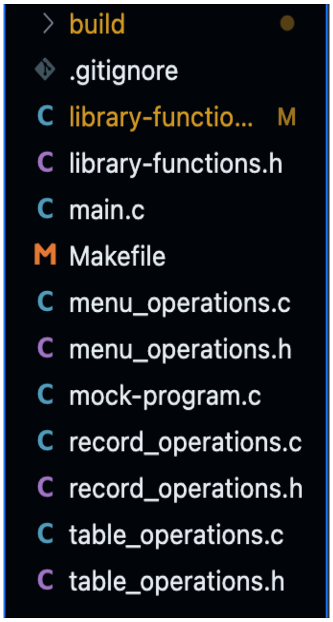
3. **Table Operations Module**: This module deals with the creation, display, and deletion of tables. It handles the manipulation of CSV files, representing tables, and ensures the proper management of table structures.

4. **Record Operations Module**: This module deals with the core CRUD operations that are to be performed on the records, It manipulates the data in the tables according to the selected operation.

Folder/File Structure:
The project structure is organized as follows:

- **Source Files**:
  - **main.c**: The entry point of the program.
  - **menu_operations.c & .h**: Handles the display of menus and database-related operations.
  - **library-functions.c & .h**: Contains utility functions used throughout the system.
  - **table_operations.c & .h**: Deals with table-related operations.
  - **record_operations.c & .h**: Handles record operations like adding, searching, and deleting records.

- **Output Files**:
  - **Makefile**: Used to automate the build process of the project.
  - **build/**:Directory containing compiled object files.



```
> build
.gitignore
C library-functio...   M
C library-functions.h
C main.c
M Makefile
C menu_operations.c
C menu_operations.h
C mock-program.c
C record_operations.c
C record_operations.h
C table_operations.c
C table_operations.h
```

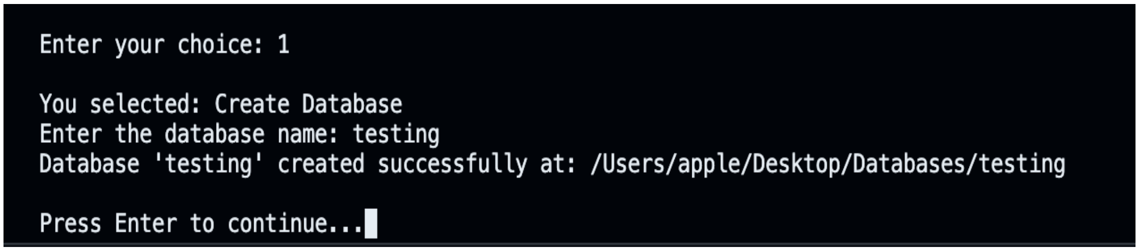
This modular structure ensures that each component of the system is independent and can be easily modified or extended without affecting other parts of the program.

Implementation

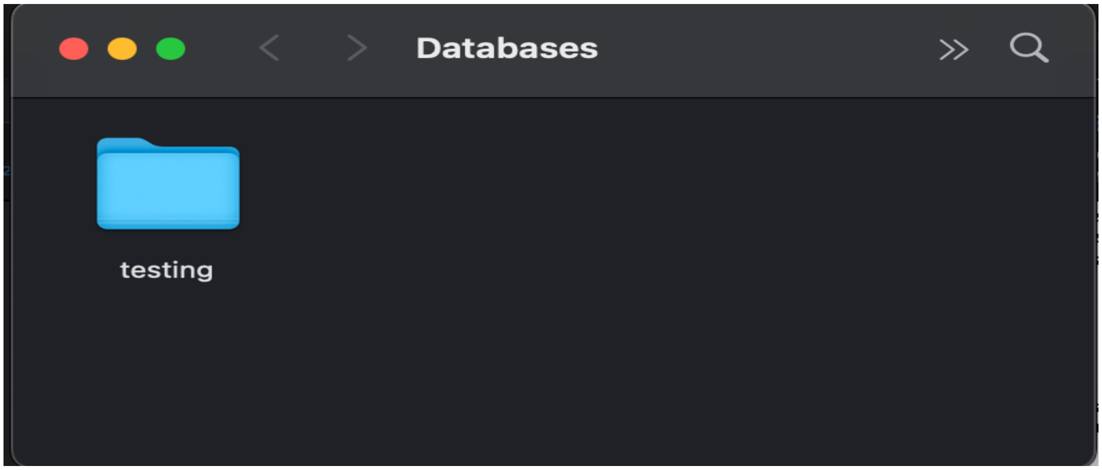
The user menu, as shown in the picture, provides a clear structure for interacting with the Database Management System (DBMS). The system is designed with various modules to handle different tasks related to database, table, and record management. Here's an overview of how the implementation aligns with the menu options:

Database Operations (1-3):

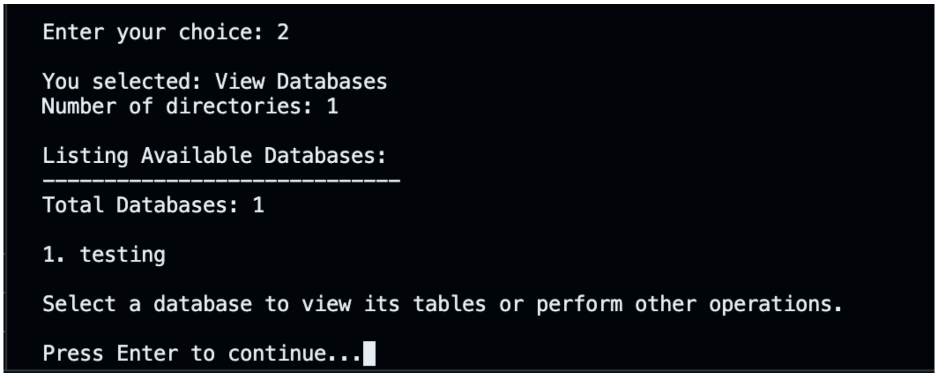
Create Database: Users can create a new database, which is represented as a folder on the system. This operation involves taking input from the user for the database name and creating a folder with the corresponding name.



```
Enter your choice: 1

You selected: Create Database
Enter the database name: testing
Database 'testing' created successfully at: /Users/apple/Desktop/Databases/testing

Press Enter to continue...
```

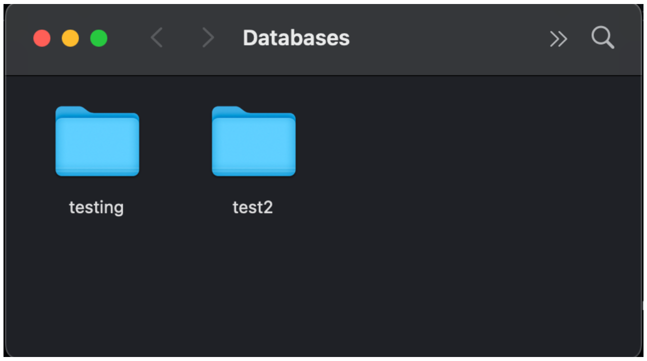


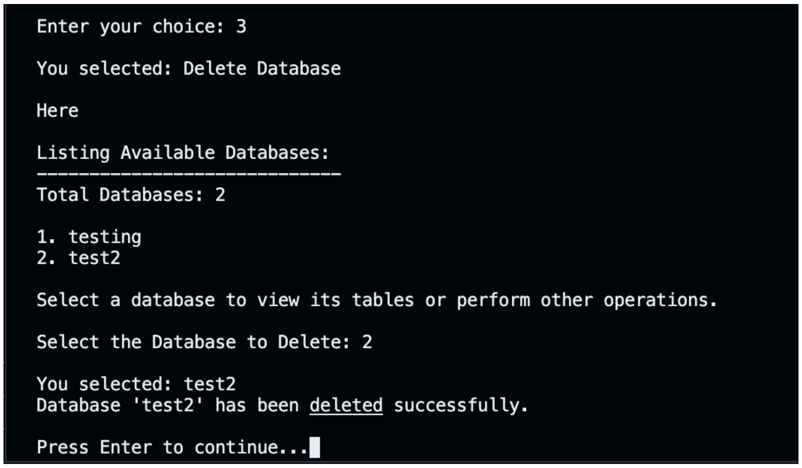
View Databases: Displays a list of available databases (folders) in the system. The user can select a database to perform further operations.



```
Enter your choice: 2

You selected: View Databases
Number of directories: 1

Listing Available Databases:
-----------------------------
Total Databases: 1

1. testing

Select a database to view its tables or perform other operations.

Press Enter to continue...
```

Delete Database: This option allows the user to delete an existing database (folder), which removes all associated tables and records.

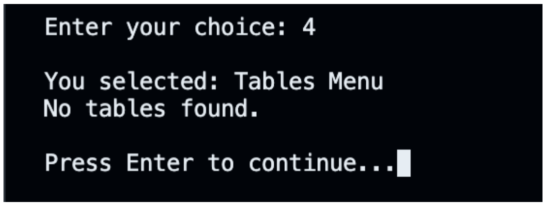




```
Enter your choice: 3

You selected: Delete Database

Here

Listing Available Databases:
-----------------------------
Total Databases: 2

1. testing
2. test2

Select a database to view its tables or perform other operations.

Select the Database to Delete: 2

You selected: test2
Database 'test2' has been deleted successfully.

Press Enter to continue...
```

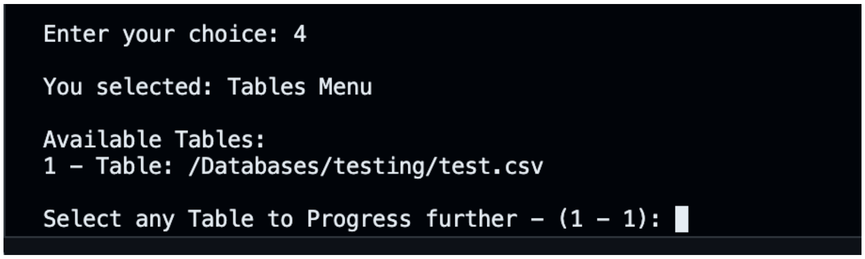
Table Operations (4-6):

Display Tables: Lists all tables in the selected database. Tables are represented as CSV files, and each table corresponds to a file that holds structured data (records).



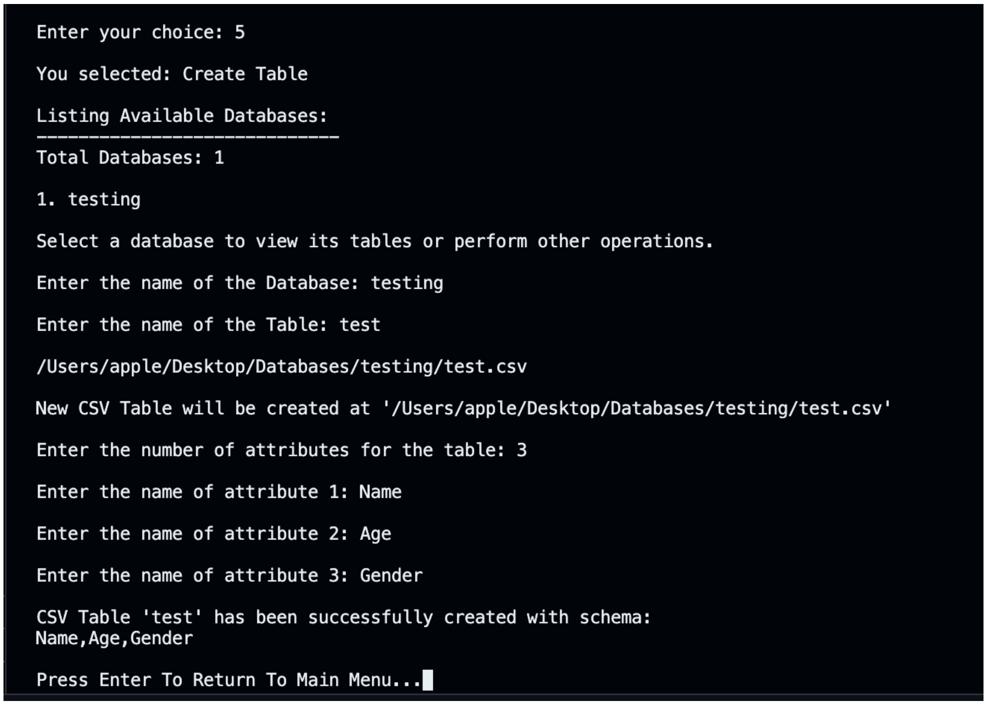
```
Enter your choice: 4

You selected: Tables Menu
No tables found.

Press Enter to continue...
```

// When No Tables are Present

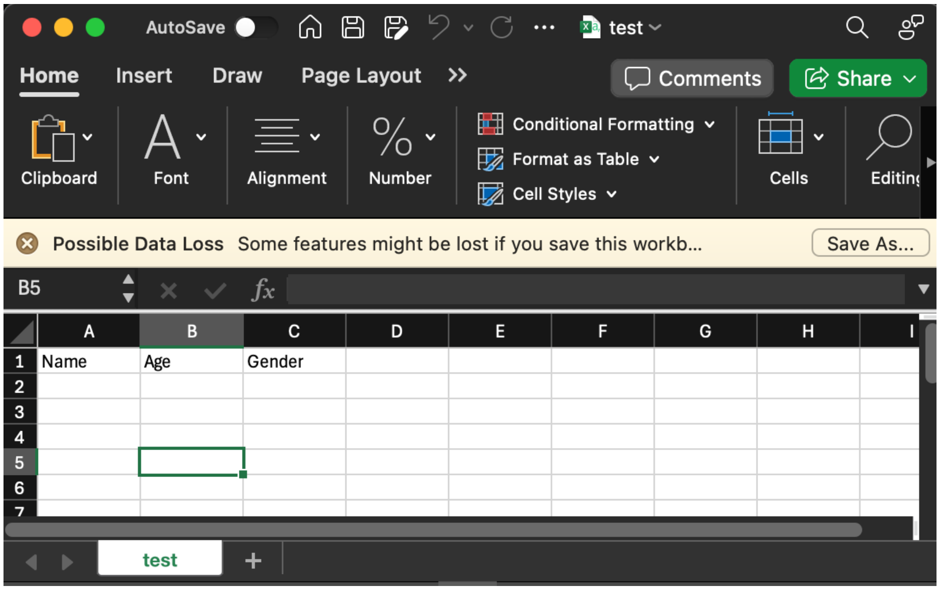


```
Enter your choice: 4

You selected: Tables Menu

Available Tables:
1 - Table: /Databases/testing/test.csv

Select any Table to Progress further - (1 - 1):
```

Create Table: This option first asks the user the name of the Database and then the table name, Users then specify its attributes (columns). A CSV file is created for the table, and the schema is stored as the first row of the file.



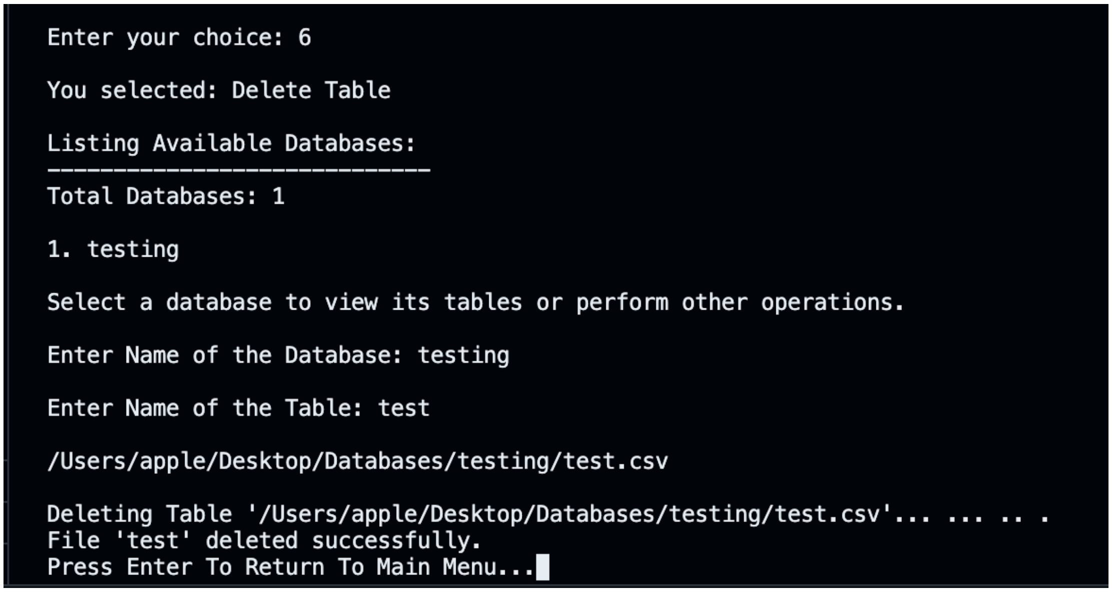
```
Enter your choice: 5

You selected: Create Table

Listing Available Databases:
-----------------------------
Total Databases: 1

1. testing

Select a database to view its tables or perform other operations.

Enter the name of the Database: testing

Enter the name of the Table: test

/Users/apple/Desktop/Databases/testing/test.csv

New CSV Table will be created at '/Users/apple/Desktop/Databases/testing/test.csv'

Enter the number of attributes for the table: 3

Enter the name of attribute 1: Name

Enter the name of attribute 2: Age

Enter the name of attribute 3: Gender

CSV Table 'test' has been successfully created with schema:
Name,Age,Gender

Press Enter To Return To Main Menu...
```



Possible Data Loss Some features might be lost if you save this workb...

	A	B	C	D	E	F	G	H	I
1	Name	Age	Gender						
2									
3									
4									
5									
6									
7									

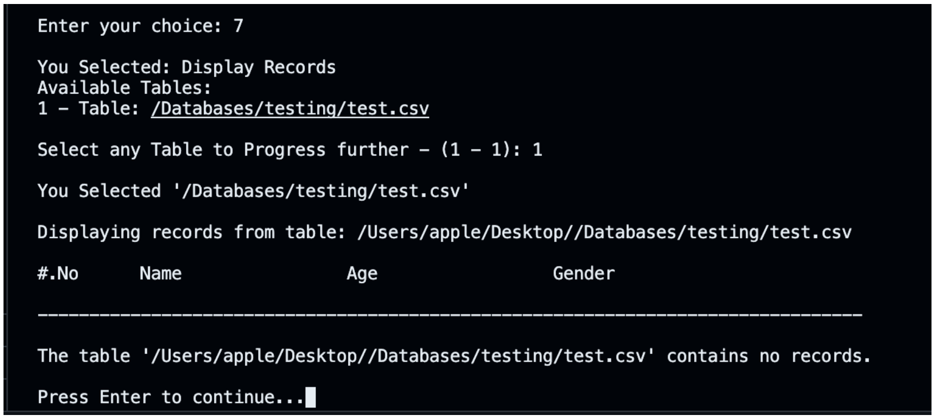
Delete Table: This option first asks the user the name of the Database and then the table name to delete then it deletes the selected table (CSV file) from the selected database.



```
Enter your choice: 6

You selected: Delete Table

Listing Available Databases:
-----------------------------
Total Databases: 1

1. testing

Select a database to view its tables or perform other operations.

Enter Name of the Database: testing

Enter Name of the Table: test

/Users/apple/Desktop/Databases/testing/test.csv

Deleting Table '/Users/apple/Desktop/Databases/testing/test.csv'... ... .. .
File 'test' deleted successfully.
Press Enter To Return To Main Menu...
```

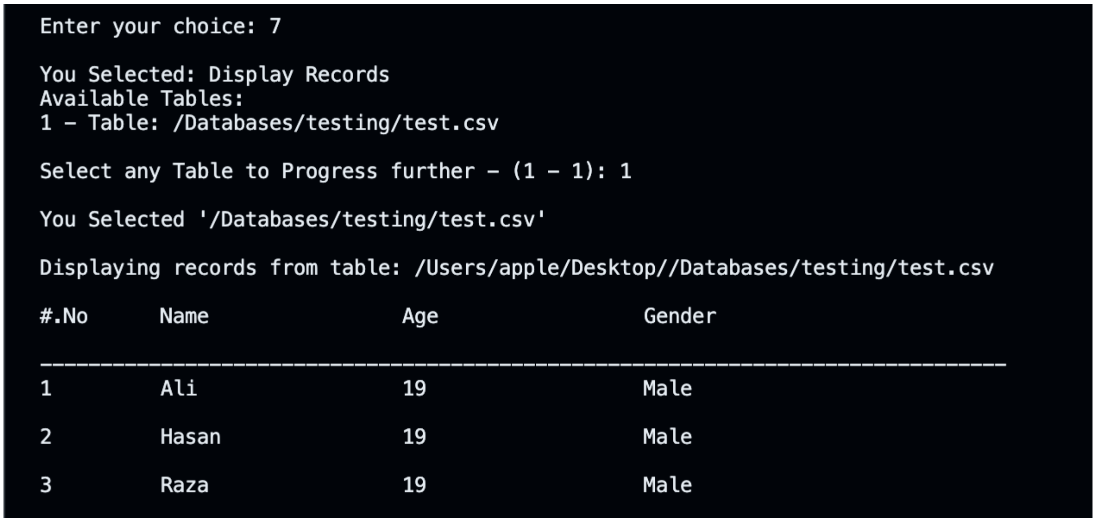
Record Operations (7-10):

Display Records: Displays all the records stored in the selected table (CSV file). Each record is represented as a row in the CSV file, and the program reads and displays this data in a user-friendly format.

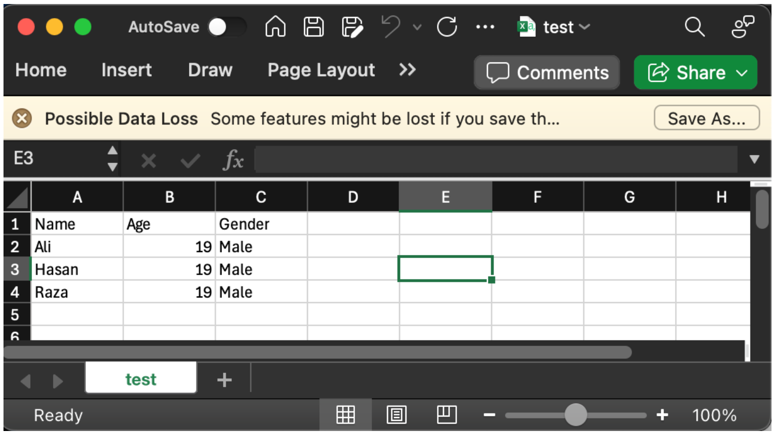


```
Enter your choice: 7

You Selected: Display Records
Available Tables:
1 - Table: /Databases/testing/test.csv

Select any Table to Progress further - (1 - 1): 1

You Selected '/Databases/testing/test.csv'

Displaying records from table: /Users/apple/Desktop//Databases/testing/test.csv

#.No      Name                Age                 Gender

--------------------------------------------------------------------------------

The table '/Users/apple/Desktop//Databases/testing/test.csv' contains no records.

Press Enter to continue...
```

// When no Records Present



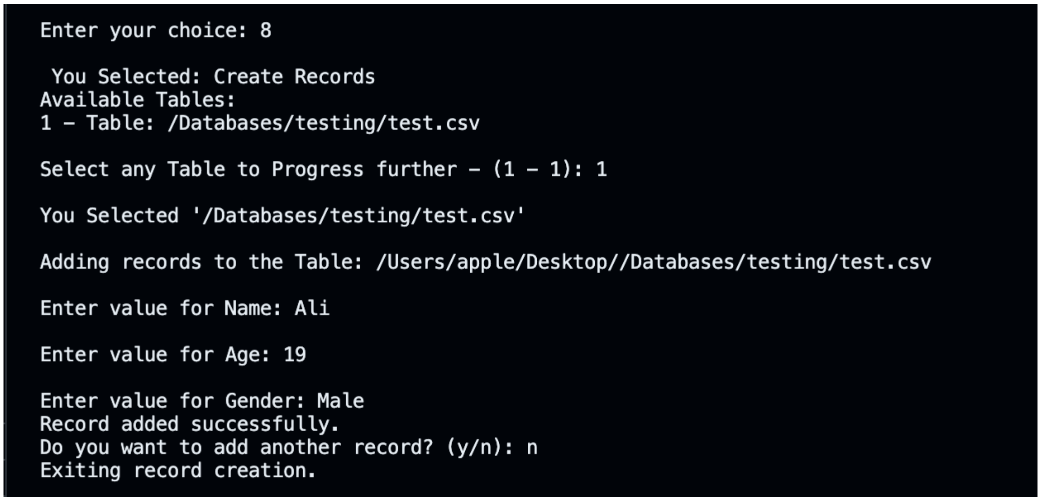
```
Enter your choice: 7

You Selected: Display Records
Available Tables:
1 - Table: /Databases/testing/test.csv

Select any Table to Progress further - (1 - 1): 1

You Selected '/Databases/testing/test.csv'

Displaying records from table: /Users/apple/Desktop//Databases/testing/test.csv

#.No      Name                Age                 Gender

--------------------------------------------------------------------------------
1         Ali                 19                  Male

2         Hasan               19                  Male

3         Raza                19                  Male
```



Possible Data Loss Some features might be lost if you save th...

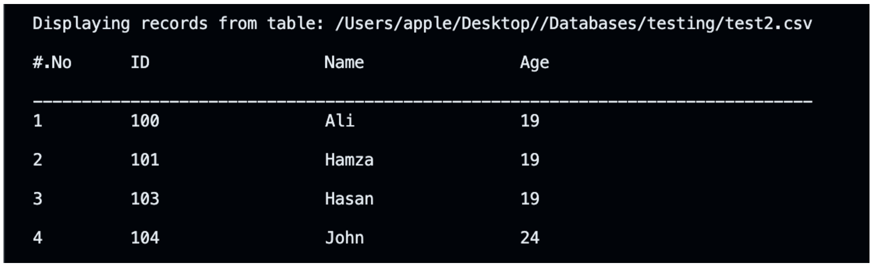
	A	B	C	D	E	F	G	H
1	Name	Age	Gender					
2	Ali	19	Male					
3	Hasan	19	Male					
4	Raza	19	Male					
5								
6								

Add a New Record: This option allows the user to add a new record to the table. The user is prompted to input values for each attribute, and the record is appended as a new row in the table's CSV file.

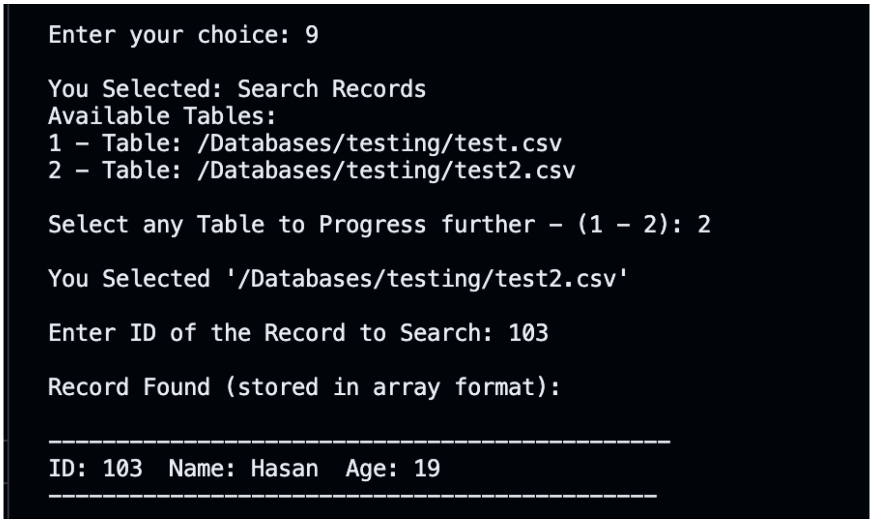


```
Enter your choice: 8

 You Selected: Create Records
Available Tables:
1 - Table: /Databases/testing/test.csv

Select any Table to Progress further - (1 - 1): 1

You Selected '/Databases/testing/test.csv'

Adding records to the Table: /Users/apple/Desktop//Databases/testing/test.csv

Enter value for Name: Ali

Enter value for Age: 19

Enter value for Gender: Male
Record added successfully.
Do you want to add another record? (y/n): n
Exiting record creation.
```

Search a Record: This option allows users to search for a specific record by ID, and view its attributes. (This assumes there is atleast one ID attribute of each row)

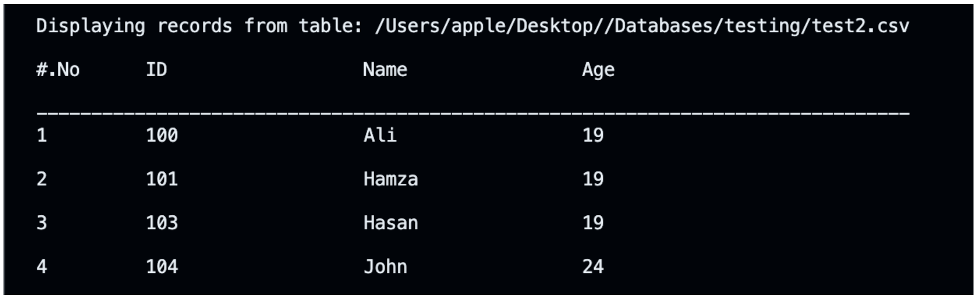


```
Displaying records from table: /Users/apple/Desktop//Databases/testing/test2.csv

#.No    ID              Name            Age

-------------------------------------------------------------------------------
1       100             Ali             19

2       101             Hamza           19

3       103             Hasan           19

4       104             John            24
```



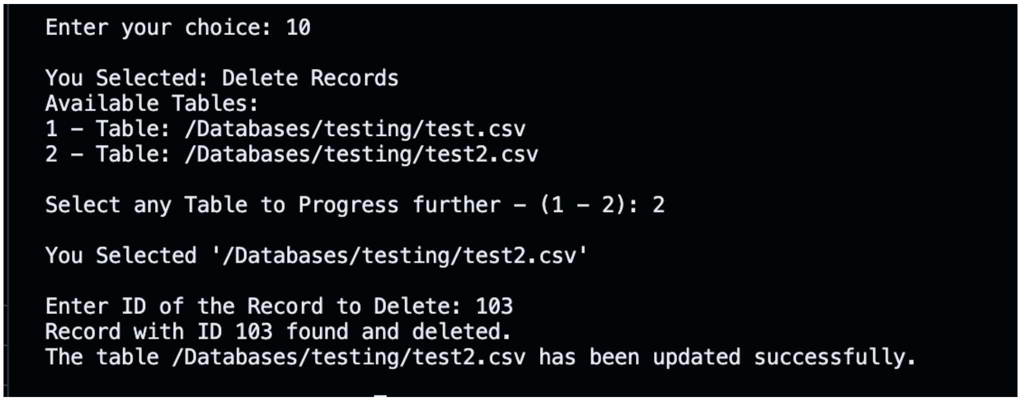
```
Enter your choice: 9

You Selected: Search Records
Available Tables:
1 - Table: /Databases/testing/test.csv
2 - Table: /Databases/testing/test2.csv

Select any Table to Progress further - (1 - 2): 2

You Selected '/Databases/testing/test2.csv'

Enter ID of the Record to Search: 103

Record Found (stored in array format):

----------------------------------------------
ID: 103  Name: Hasan  Age: 19
---------------------------------------------
```

Delete a Record: This option allows the user to delete a record from the table by specifying its ID. The record is removed from the CSV file, and the table is updated.



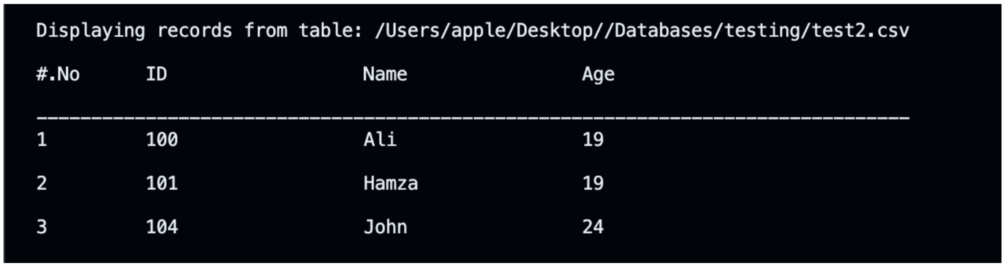
```
Displaying records from table: /Users/apple/Desktop//Databases/testing/test2.csv

#.No      ID                Name              Age

--------------------------------------------------------------------------------
1         100               Ali               19

2         101               Hamza             19

3         103               Hasan             19

4         104               John              24
```

// Before



```
Enter your choice: 10

You Selected: Delete Records
Available Tables:
1 - Table: /Databases/testing/test.csv
2 - Table: /Databases/testing/test2.csv

Select any Table to Progress further - (1 - 2): 2

You Selected '/Databases/testing/test2.csv'

Enter ID of the Record to Delete: 103
Record with ID 103 found and deleted.
The table /Databases/testing/test2.csv has been updated successfully.
```

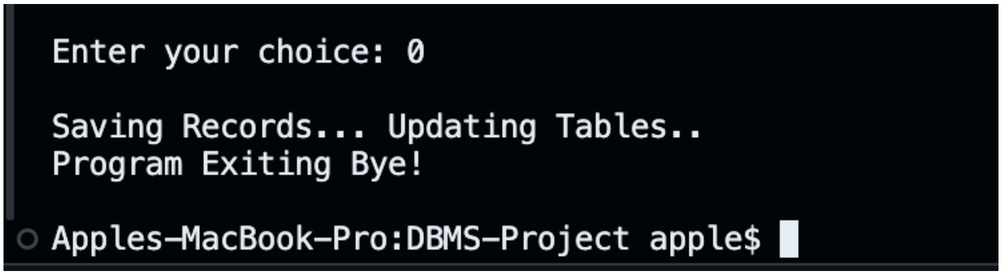
// Calling Function



```
Displaying records from table: /Users/apple/Desktop//Databases/testing/test2.csv

#.No      ID                Name              Age

--------------------------------------------------------------------------------
1         100               Ali               19

2         101               Hamza             19

3         104               John              24
```

// After Record Deleted

Exit: Inputting 0 allows the user to exit the program.



```
Enter your choice: 0

Saving Records... Updating Tables..
Program Exiting Bye!

Apples-MacBook-Pro:DBMS-Project apple$
```

By using each of these menu options which corresponds to a function within the system. The user is guided through different menus to manage databases, tables, and records effectively. The system uses a command-line interface to interact with the user and ensures that the operations are performed on files (CSV files for tables) and folders (for databases) on the file system.

Limitations:

1. **Limited Scalability**: The system handles basic CRUD operations but struggles with large datasets or high-performance requirements.
2. **No Advanced Querying**: It lacks support for complex queries such as joins, aggregations, or filters beyond basic operations.
3. **No Multi-User Support**: The system does not handle different user roles like admin, normal user.

Future Enhancements

In the future, I plan to enhance this database management system by integrating support for SQL-like commands, making it more user-friendly and versatile for performing advanced operations. I aim to implement data encryption to ensure security for stored information, particularly when handling sensitive data. Additionally, developing a graphical user interface (GUI) will significantly enhance the system's usability, providing an intuitive way for users to interact with the database. Finally, I plan to implement more robust error handling to improve reliability and prevent system crashes during invalid operations. These enhancements will make the system more scalable, secure, and practical for real-world applications.

Conclusion

This Database Management System is a foundational tool that mimics basic functionalities of a relational database. It provides a practical way to manage databases, tables, and records while focusing on CRUD operations. The system demonstrates how databases can be built from scratch using file handling and modular programming. While it has limitations, it serves as a great learning tool and a foundation for future enhancements like SQL integration, better scalability, and improved user experience.