

# **Digital Personal Diary System**

**Farhan Zaman (ID: 2531094642)**

**Sadat Nasar (ID: 2533769042)**

**Anika Tasnim Maisha (ID: 2534689042)**

**Group no: 07**

**Course: CSE115**

**Section: 04**

---

programming concepts such as functions, loops, conditional statements, file handling, and structured program design.

The system enables users to create individual accounts, securely log in, and manage personal diary entries. Each user can add new entries, view existing entries, search for specific records, edit previous entries, delete unwanted entries, and export all diary data to a separate file. The project focuses on clarity, usability, and modularity rather than advanced security or graphical interfaces.

---

## **1. Introduction**

With the rapid growth of digital systems, personal information management has become increasingly important. One of the simplest yet most effective forms of personal data management is maintaining a diary. Traditionally, diaries were handwritten, but modern computing allows diary systems to be implemented digitally with better organization, searchability, and storage.

This project presents a **console-based Personal Diary Management System** developed using the C programming language. The project is designed as part of the CSE 115 course to apply foundational

## **2. Motivation and Problem Statement**

In the modern digital age, individuals increasingly rely on electronic systems to record, organize, and preserve personal information. Maintaining a diary is a common practice for documenting daily experiences, thoughts, and important events. However, traditional handwritten diaries can be difficult to organize, search, and preserve over long periods of time. As a result, there is a growing need for simple digital diary solutions that allow users to manage their personal records efficiently and securely.

The motivation behind this project stems from the desire to design a digital diary system that provides an organized and user-friendly way to store personal entries. A digital diary offers several advantages over traditional methods, such as easier access to past records, improved organization, and the ability to manage large amounts of information without physical limitations. By structuring diary entries digitally, users can maintain a clear and consistent format for recording dates, titles, and detailed content.

A digital diary system is particularly suitable because it naturally supports essential operations required for personal record management. Users need the ability to create new entries, view previously recorded information, update existing records when necessary, and remove entries that are no longer relevant. These operations ensure that the diary remains accurate, up-to-date, and meaningful over time. Additionally, a menu-driven interaction model makes the system easy to navigate, even for users with minimal technical background.

The core problem addressed by this project is how to manage personal diary information in a simple, organized, and persistent digital form. The system aims to ensure that personal entries can be stored reliably, accessed conveniently, and maintained over long periods without

complexity. By addressing this problem, the project highlights the importance of digital solutions for personal data organization while focusing on clarity, usability, and long-term accessibility.

---

### **3. Objectives of the Project**

The objectives of this project are divided into technical and learning objectives

#### **3.1 Technical Objectives**

- To design a functional diary system using C
- To implement a user authentication mechanism
- To ensure data persistence using text files
- To allow users to manage diary entries efficiently

#### **3.2 Learning Objectives**

- To gain hands-on experience with file input/output operations
  - To practice modular programming using functions
  - To understand menu-driven program design
  - To improve problem-solving and debugging skills
-

## 4. System Overview

The Personal Diary Management System is a **menu driven console application**. The program begins with a startup menu where users can either sign up, log in, or exit the application. Once authenticated, users gain access to the main diary menu.

Each user's diary entries are stored in a separate file, named using the username. This design ensures separation of data and simplifies file management. The system does not rely on external databases or libraries, making it lightweight and suitable for introductory programming courses.

---

## 5. System Architecture and Design

The system follows a **procedural programming architecture** with a clear separation of concerns.

### 5.1 Modular Design

Each major functionality is implemented as a separate function. This improves readability, maintainability, and ease of debugging. For example, signup,

login, entry management, and export operations are all handled independently.

### 5.2 File-Based Storage

Two types of files are used:

- A user credential file to store usernames and passwords
- Individual diary files for each user

This approach simplifies data persistence and avoids the complexity of database integration.

### 5.3 Menu-Driven Interaction

User interaction is controlled through menus using switch-case and loop structures. This ensures that users can easily navigate the system and perform desired actions without confusion.

---

## **6. Functional Requirements**

### **6.1 User Registration**

The system allows new users to create an account by choosing a unique username and password. The credentials are stored in a file so that users can log in during future sessions.

### **6.2 User Authentication**

Only registered users can access the diary system. The login process verifies credentials before granting access, ensuring basic data privacy.

### **6.3 Diary Entry Creation**

Users can create diary entries by providing a date, title, and content. The system supports multi-line content, allowing users to write detailed entries.

### **6.4 Viewing Diary Entries**

All diary entries can be viewed sequentially. Entries are clearly separated and numbered for easy identification.

### **6.5 Searching Diary Entries**

Users can search for a specific entry by its number. This feature helps users quickly locate specific records.

### **6.6 Editing Diary Entries**

Existing diary entries can be modified. Users can update titles and content while preserving the entry's date.

### **6.7 Deleting Diary Entries**

Users can permanently remove diary entries after confirmation. This prevents accidental data loss.

### **6.8 Exporting Diary Entries**

All diary entries can be exported to a separate text file. This feature supports data backup and sharing.

---

## **7. Non-Functional Requirements**

- The system should be easy to use
  - The program should execute efficiently for small to medium datasets
  - The interface should be clear and readable
  - The system should handle basic input errors gracefully
- 

## **8. Algorithms and Logical Flow**

The program relies heavily on sequential file processing. Operations such as viewing, searching, editing, and deleting entries involve scanning the diary file line by line.

Temporary files are used for editing and deletion operations to ensure that data integrity is maintained. Once an operation is completed, the temporary file replaces the original file.

---

## **9. Tools and Technologies Used**

- Programming Language: C
  - Compiler: GCC
  - IDE/Text Editor: Code::Blocks
  - Operating System: Windows, Linux and Mac OS.
- 

## **10. Testing and Validation**

The system was tested using various scenarios, including:

- Valid and invalid login attempts
- Adding multiple diary entries
- Editing and deleting entries
- Searching for existing and non-existing entries
- Exporting diary data

The system performed reliably under all test conditions relevant to the project scope.

---

## 11. Limitations of the Project

Despite its functionality, the system has several limitations:

- Passwords are stored in plain text
  - No encryption is used for diary files
  - No graphical user interface is provided
  - The system does not support multiple users accessing the same data simultaneously
- 

## 12. Possible Enhancements

Future improvements can further enhance the usability, security, and scalability of the diary application. These may include:

- **Password encryption and secure authentication** to protect user credentials and prevent unauthorized access
- **Graphical user interface (GUI)** implementation to improve usability and overall user experience
- **Database-based storage** to replace file handling and support better scalability and data management
- **Keyword-based and date-based search functionality** for

faster and more efficient retrieval of diary entries

- **Cloud-based backup options** to ensure data safety, accessibility, and recovery across multiple devices
- 

## 13. Contribution of Each Team Member

The project was developed collaboratively, with each team member contributing to specific functional areas of the system.

Anika Tasnim Maisha:

Responsible for the overall **program flow and integration**. Implemented loop structures, switch case logic, function prototypes, and managed the user entry menus and integrated all modules within the `main()` function to ensure smooth navigation between authentication and diary features.

Sadat Bin Nasar:

Focused on the **core diary functionalities**. Developed the main diary menu and implemented features for adding, viewing, searching, editing, and deleting diary entries and ensured structured data storage and efficient file handling for diary operations.

Farhan Zaman:

Handled **authentication and file management**. Implemented the signup and login systems, managed user-specific file creation, and developed the export functionality and ensured proper handling of files and data persistence across users.

---

## 14. Learning Outcomes

Through this project, we gained practical experience in the following areas:

- **File handling in C**, including reading from and writing to files for persistent data storage
  - **Modular program design**, by breaking the application into well-structured and reusable functions
  - **Debugging and testing**, identifying logical and runtime errors and improving program reliability
  - **Team-based software development**, collaborating effectively by dividing tasks and integrating individual components
- 

## 15. Conclusion

The **Personal Diary Management System** successfully fulfills the objectives of the CSE 115 project by demonstrating the practical application of fundamental C programming concepts. The system integrates file handling, modular design, and user interaction to create a functional and user-friendly application. While the current implementation has certain limitations, it provides a strong foundation for understanding file-based systems and offers ample scope for future enhancements with more advanced features.

---