

The features of the application based on the code line by line:

#### Data Structures:

The application defines two data structures: struct DiaryEntry and struct User.

struct DiaryEntry stores information about a diary entry, including the username, date, and content.

struct User stores information about a registered user, including the username and hashed password.

#### Function Prototypes:

The application declares several function prototypes that will be defined later in the code.

These function prototypes provide a structure to the code and allow functions to be used before their actual implementation.

#### Constants:

The application defines two constant arrays: usersFileName and diaryFileName.

These arrays store the file names where user data and diary entries will be stored, respectively.

encryptionKey is also defined as a constant, which is used to encrypt and decrypt diary content.

#### main() Function:

The main function is the entry point of the program.

It displays the main menu options using displayMenu() and takes user input for the choice using fgets() and sscanf().

Depending on the user's choice, it calls relevant functions such as registerUser(), loginUser(), and displayUserMenu().

#### displayMenu() Function:

This function displays the main menu options for the user to choose from.

#### displayUserMenu() Function:

This function displays the user menu options after successful login.

It takes the username as input and provides options to write a new entry, view previous entries, search entries, and logout.

Depending on the user's choice, it calls relevant functions like `writeEntry()`, `viewEntries()`, and `searchEntries()`.

`registerUser()` Function:

This function allows a new user to register by providing a username and password.

The password is hashed using `simpleHash()` function before being stored in the users file.

`loginUser()` Function:

This function allows an existing user to log in by providing their username and password.

The provided password is hashed and compared with the stored hashed password in the users file.

`writeEntry()` Function:

This function allows a logged-in user to write a new diary entry.

It takes the date and content of the entry as input from the user.

The content of the entry is encrypted using `encrypt()` function before being stored in the diary file.

`viewEntries()` Function:

This function allows a logged-in user to view their previous diary entries.

It reads entries from the diary file and decrypts the content using the `decrypt()` function before displaying it.

`searchEntries()` Function:

This function allows a logged-in user to search for specific diary entries based on a search term (date or content).

It reads entries from the diary file, decrypts the content using the `decrypt()` function, and checks if the search term matches the date or content.

`encrypt()` and `decrypt()` Functions:

These functions provide basic encryption and decryption functionality using a simple shift cipher based on the `encryptionKey`.

The `encrypt()` function encrypts the provided text, and the `decrypt()` function decrypts the encrypted text.

`validateDate()` Function:

This function checks if the provided date is in the valid format (DD-MM-YYYY) and within a reasonable range.

`clearBuffer()` Function:

This function clears the input buffer, ensuring that any extra characters from previous input are discarded.

Overall, the application allows users to register, login, write diary entries, view previous entries, search for entries, and log out. It encrypts the content of the diary entries for added privacy and security.