Readme

- **Amit Chougule**

Before initiating any processes, the program ensures the presence of proper storage in CSV format to record user and book details.

To accomplish this, the code initially checks whether the user details CSV and book records details CSV are present in the current working directory. If these files are not found, the program automatically creates them. For the "`User_details.csv`" file, the columns include `User ID,` `Username,` `Password,` and `Issued` `Books.` Similarly, for the "`books_record_details.csv`" file, the columns consist of `Title,` `Author,` and `ISBN.`

This proactive approach guarantees that the necessary storage infrastructure is in place to facilitate the recording and management of user and book details throughout program execution.

After this the given program gives you the following 5 options,
1. Add Book
2. List Books
3. Add User
4. Checkout Book
5. Exit

1. Add Book

The `Add Book` module empowers users to seamlessly incorporate specific books into the collection. This module solicits input from users, including the `title,` `author,` and `ISBN` (International Standard Book Number), a unique identifier for each book.

Upon receiving the `title` input, the program ensures that the field is not left empty. If the user inadvertently submits an empty field, the program prompts an error message stating "`Title cannot be empty or blank`," guiding the user to provide the necessary input once more. The `title` input accepts both alphabetic and numeric characters.

Subsequently, the program prompts users to provide details of the `author's name` for the book entry. During this phase, the program verifies that the `author's name` is not left blank and does not contain any digits. Additionally, it confirms that all characters in the `author's name` are alphabetic, signalling an appropriate error or warning messages to the user if any discrepancies are found, and prompting for input correction.

In the final step, the program requests the `ISBN number` from the user. While capturing the input, it ensures that the field is not left empty and only numeric values are accepted as `ISBN` input entries.

Utilizing the information gathered from the three fields—`title of the book,` `author,` and `ISBN`—the program inserts the respective entry row into `book records details` CSV file.

2. List Book

The `List Book` module within the program facilitates the display of presently available books listed in the book records details CSV file. It showcases comprehensive details of available books, including the `title,` `author,` and `ISBN`.

This module specifically filters out books that are available for checkout by users. It excludes any books that have already been issued to other users. To achieve this, `ISBN` numbers of issued books are prefixed with '#' to signify their unavailability for issue. Consequently, when a user selects option 2, "`List Book`," the program only presents available books, omitting those whose `ISBN` starts with '#', indicating they are already issued and thus unavailable for checkout. It's important to note that entries of issued books are retained within the CSV database for future operational purposes.

3. Add User

The `Add User` module is intricately linked with user management, designed to facilitate the addition of new users to the user details records CSV. This process involves several steps to ensure the integrity and security of user information.

Initially, the module prompts the user to input a `username`. It validates that the entry for the `username` is not left blank or empty. If such cases are detected, the program rejects the input and displays a message stating "`Name cannot be empty or blank`," prompting the user to provide the necessary input once more.

Subsequently, the program requests a `user ID` input from the user. It ensures that the entry for the `user ID` is not left blank or empty. If such cases are identified, the program rejects the input and displays a message stating "`User ID cannot be empty or blank`," instructing the user to provide the required input again. Additionally, it verifies that the provided `user ID` consists only of numerical digits. If any other characters are detected, an error message is displayed stating "`User ID must contain only numbers`," prompting the user to correct the input.

Next, the module prompts the user to set a `password` for their account. Similar to previous inputs, the program ensures that the `password` entry is not left blank or empty. If such cases are found, the program rejects the input and displays a message

stating "`Password cannot be empty or blank`," guiding the user to provide the necessary input again. Prior to storing the `password`, the program employs a unique encryption mechanism. Specifically, it performs an `XOR operation with a key`, which is currently set to 10 (modifiable for future operations). This encryption ensures that the password stored in the CSV remains secure. Even in the event of unauthorized access to the CSV records, the encrypted format of the password prevents direct access by unauthorized individuals, thereby enhancing the privacy and security of user details and storage.

Finally, the `user ID, username,` and `encrypted version of the user's password` are stored in the `user details` CSV records.

4. Checkout Book

The `Checkout Book` module facilitates the issuance of available books to the current user. It streamlines the process of borrowing books, ensuring a seamless experience for users.

Initially, the module prompts the user to input their `User ID`. It ensures that the entry for the `User ID` is not left blank or empty. If such cases are identified, the program rejects the input and displays a message stating "`User ID cannot be empty or blank`," guiding the user to provide the required input once more. Additionally, it validates that the provided `User ID` consists only of numerical digits. If any other characters are detected, an error message is displayed stating "`User ID must contain only numbers,`" prompting the user to correct the input.

Subsequently, the program requests the `ISBN` number from the user. It ensures that the field is not left blank and only accepts numeric values as `ISBN` input entries.

Following this, the module verifies the `User ID` with the `user details` CSV records. If the `User ID` is found, the program proceeds with the next operation; otherwise, it displays an error message stating "`User with ID ___ not found`." Once the `User ID` is confirmed, the module checks the `ISBN` number against the `book records details` CSV. If a match is found, the program adds a '#' prefix to the `ISBN` number in the `book records details` CSV, indicating that the corresponding book has been issued. Simultaneously, it assigns the `ISBN` number to the "`Issued books`" column in the `user record details` CSV, associating the specific `ISBN` number with the given `user ID`, signifying that the book has been issued to the user.

This comprehensive process within the `Checkout` module ensures smooth and efficient book checkout and issuing procedures.

5. Exit

The `Exit` module serves to gracefully close the program and terminate all ongoing operations.

This module ensures a smooth and orderly exit from the program, allowing users to conclude their tasks and exit the application seamlessly.