**Documentation for Online Library Server Client Application**

1. **Project Design**

The project consists of two main components: a server and a client. Both are implemented in C using TCP sockets for communication. The system utilizes SQLite3 as the backend database to manage users, books, and rental information.

*Server Part Design*

The server is responsible for handling client connections, processing user requests, and interacting with the SQLite database.

* The server initializes by setting up a TCP socket and binding it to a specific port (PORT 8080).
* It listens for incoming client connections and handles each in a separate process using fork().
* Upon receiving a request from a client, the server processes it based on the chosen action (e.g., login, create account, find books, rent a book, return a book).
* The database is initialized at the start, ensuring the required tables (users, books, and rented\_books) exist.
* Core functionalities include user account creation, login, displaying account information, listing available books, renting books, and returning books.

*Client Part Design*

* The client program initiates a connection to the server using a TCP socket.
* It provides an interactive menu to the user, allowing them to log in or create a new account.
* Upon login or account creation, the client displays options to view the user’s account details, find books, rent a book, return a book, or exit the application.
* The client sends user input as requests to the server and displays the server’s response.

*Communication between Server and Client*

* The communication is based on a request response model using TCP sockets.
* The client sends commands as text to the server, and the server processes these commands and responds with text messages.
* Each client request triggers a specific server function, such as querying the database or updating records.

1. **Instructions for Compiling and Running the Programs**

*Dependencies*

* Ensure you have **gcc** and **sqlite3** installed on your system. If you don’t have sqlite3 you can install it with the commands:

**sudo apt update**

**sudo apt install sqlite3 libsqlite3-dev**

* The source files are named server.c and client.c.
* make file to compile the program

*Compiling the Program*

Open the terminal in the project directory and run the command:

**make**

*Run the server*

Open the terminal in the project directory and run the command:

**make run-server**

*Run the Client*

In a separate terminal in the project directory and run the command:

**make run-client**

Note: The server and client should be executed in separate terminal windows for proper communication.

1. **Program Usage and Interface Instructions**

*Starting the Program*

First, start the server using **make run-server**. The server will initialize the database and listen for client connections.

In a separate terminal, start the client using **make run-client**. The client connects to the server and displays the main menu.

*Main Menu Options*

The client displays a menu with the following options:

* **Login**: The user enters their username and password to access the system.
* **Create Account**: The user provides a username, password, and ZIP code to create a new account.
* **Exit**: Closes the client connection.

Users can be able to login or create an account if they don’t have one

*Library Menu Options (after login)*

1. Account: Displays user details, including name, and rented books.
2. Find Book: Lists all available books in the store with details such as ID, name, author, time limit, and availability.
3. Rent Book: Allows the user to rent a book by entering its ID. The system updates the availability and records the rental.
4. Return Book: Allows the user to return a previously rented book by providing its ID. The system updates the availability.
5. Exit: Ends the session and disconnects the client.

1. **Software and Hardware Descriptions**

*Software Requirements*

Operating System: Linuxbased OS (e.g., Ubuntu, Fedora), macOS, or Windows Subsystem for Linux (WSL).

Compiler: GCC (GNU Compiler Collection)

Database: SQLite3

*Hardware Requirements*

CPU: Any processor capable of running GCC and SQLite3.

Network: Local network or internet connection to allow clientserver communication.

1. **Program Synopsis**

The online bookstore/library application simulates a book rental system using a client-server architecture. The server handles multiple client connections concurrently, allowing users to create accounts, log in, search for books, rent, and return books. The SQLite3 database stores user information, book details, and rental records.

*Key Features*

* User Management: Create accounts, login authentication.
* Book Management: View available books, track rented books, manage book availability.
* Rental Operations: Rent and return books with real-time updates in the database.

1. **Database Management System (DBMS)**

*SQLite3*

The server uses SQLite3 to manage data persistently.

*Database tables*

**users**: Stores user information (ID, name, password).

**books**: Stores book details (ID, name, author, time limit, availability).

**rented\_books**: Tracks which user has rented which book (user\_id, book\_id).