Task 2 Description:

Assume we have a server that allows two clients to connect:

• **Client 1 (Master):** The first client to connect becomes the "master" and is permitted to configure the server's global options. Specifically, the master can set any or all the server's Boolean options (such as buffer_data, enable_timeout).

- **Client 2 (Slave):** The second client to connect is the "slave."

o The slave can attempt to configure the same server options.

o If the slave tries to set any option to a different value than what the master has already configured (or what remains at the default if not set by the master), an error message is returned.

o If the slave sets all options to the same values already configured (including defaults not set by the master), no error occurs.

• **Default Behavior:**

o Any option not set by the master retains its default value (e.g., by default, buffer_data = TRUE, enable_timeout = TRUE).

o If the master specifies only one or a subset of options, all others remain at their default values.

Task Overview

1. Test Design:

o Plan to validate the server's logic by checking every combination of how master and slave might configure the global options.

2. Automation:

o Write a generic Python program that, for any given list of *Boolean* server options:

▪ Generates all combinations of master/slave configuration attempts.

▪ Determines and records the expected server response for each test case.

▪ Produces a CSV output in the format shown below

TestCase ID	Master Option For BufferData	Master Option For TimeOut	Client Option For BufferData	Client Option For TimeOut	Valid TC	Expected BufferData	Expected TimeOut
1	NA	NA	NA	NA	YES	TRUE	TRUE
2	TRUE	FALSE	NA	TRUE	NO	NA	NA
3	FALSE	TRUE	FALSE	NA	YES	FALSE	TRUE

Table Description:

TestCase ID	Unique identifier for each test case
Master/Client Option	Values for the option: TRUE/FALSE or NA if not set
Valid TC	YES: If no error occurred from the server. NO: Otherwise
Expected BufferData/TimeOUT	Expected Final values after attempted configuration

Program Input:

• List of server option names (e.g., buffer_data, enable_timeout, ...)

Program Output:

• CSV file structured as above, for all combinations and expected results.

Input Constraints:

1. Server must have at lease 1 option name as input. Empty option name is invalid.

2. Server options name must be unique. Repeated options name is invalid.

3. Special characters are considered invalid input.