

Syed Ghayur Hussain

231570

CS4-B

Socket Programing project

Description:

### This project is a client-server based Tour Management System developed using C++ socket programming to demonstrate network communication and object-oriented design. It enables clients to connect to a central server that manages tour data. The server handles one client connection at a time, processing requests sequentially. This project is an upgraded version of my second-semester Object-Oriented Programming (OOP) project, enhanced by integrating socket programming to enable networked client-server communication. Architecture & Design

The system is divided into two main components:

* **Server (Socket.cpp):**  
  The server listens for incoming TCP client connections and processes requests related to tours. It manages all tour data storage, retrieval, updates, and deletions by interacting with persistent files. The server uses a modular approach, delegating requests to specific classes based on the type of tour.
* **Client (Client.cpp):**  
  The client provides a console-based menu-driven interface allowing users to send commands to the server and view responses. It communicates with the server over TCP sockets, sending formatted requests and displaying results.

### Core Classes and Functionality

The system models tours through three primary classes, encapsulating different types of tours and their attributes:

* **Guide Class:**  
  Represents tours with a dedicated guide. It stores details like guide ID, age, experience, languages spoken, and rating. Supports CRUD operations on guide-related tour data.
* **Business Class:**  
  Models business-class tours with attributes including destination, departure date, price, and any business-specific details. Enables users to add, view, update, or delete business tours.
* **Package Class:**  
  Handles pre-defined tour packages that clients can choose from. Contains package-specific attributes such as package ID, duration, included services, and price. Facilitates operations to browse and manage package tours.

### Key Features

* **Network Communication:**  
  Implements TCP sockets to enable reliable communication between clients and server.
* **Request Handling:**  
  The server parses client requests, routes them to appropriate classes, and sends back serialized tour data or status messages.
* **Persistence:**  
  All tour data is persisted using file I/O, ensuring data durability across sessions.
* **Menu-driven Client:**  
  User-friendly client interface that dynamically sends requests to the server based on user input and displays responses.
* **Extensible Design:**  
  The modular class structure allows easy addition of new tour types or features.

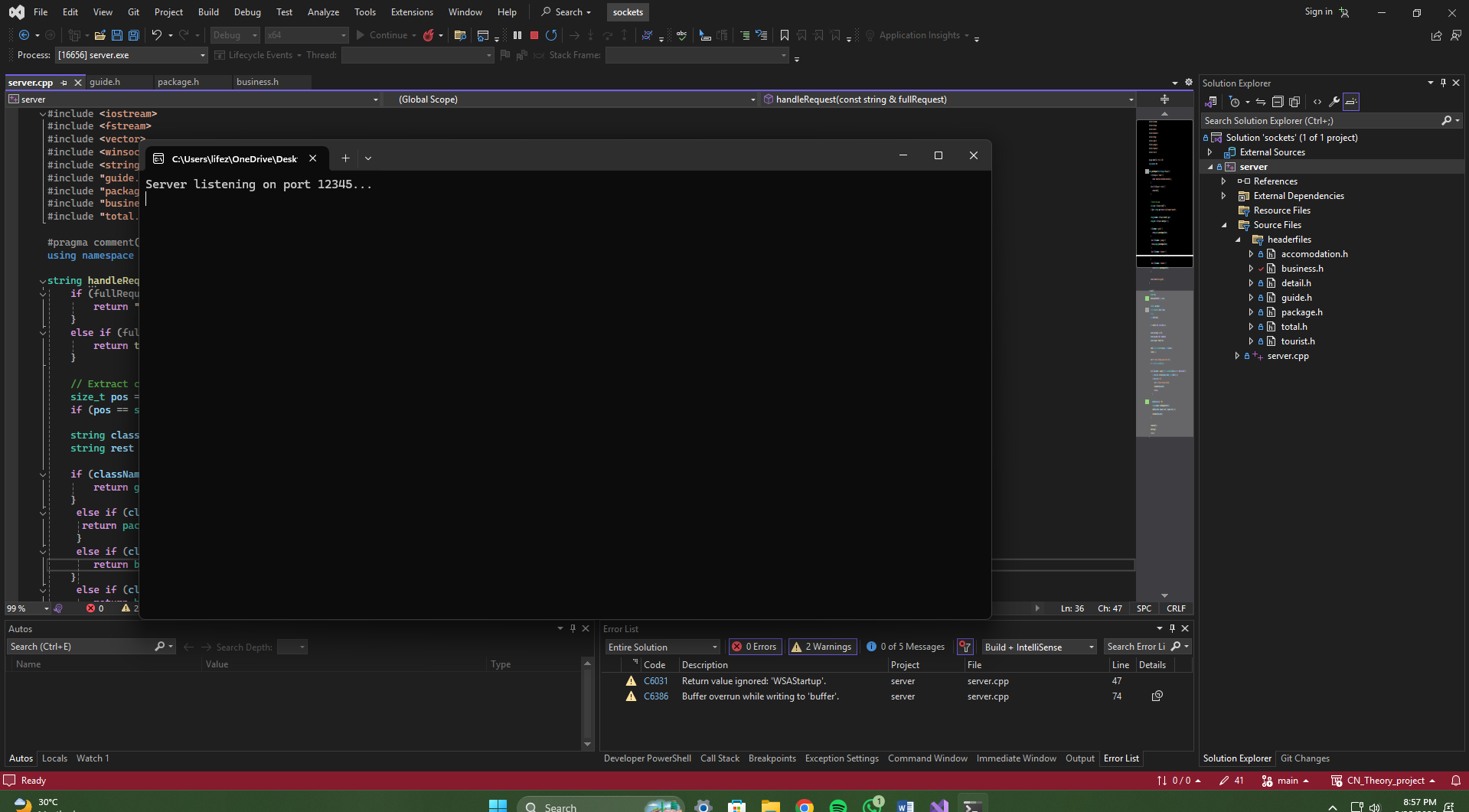
### Educational Outcomes

This project deepens understanding of:

* TCP/IP socket programming for client-server communication.
* Object-oriented principles like encapsulation, inheritance, and polymorphism.
* File handling for data persistence in C++.
* Designing interactive, networked applications with clean separation of concerns.

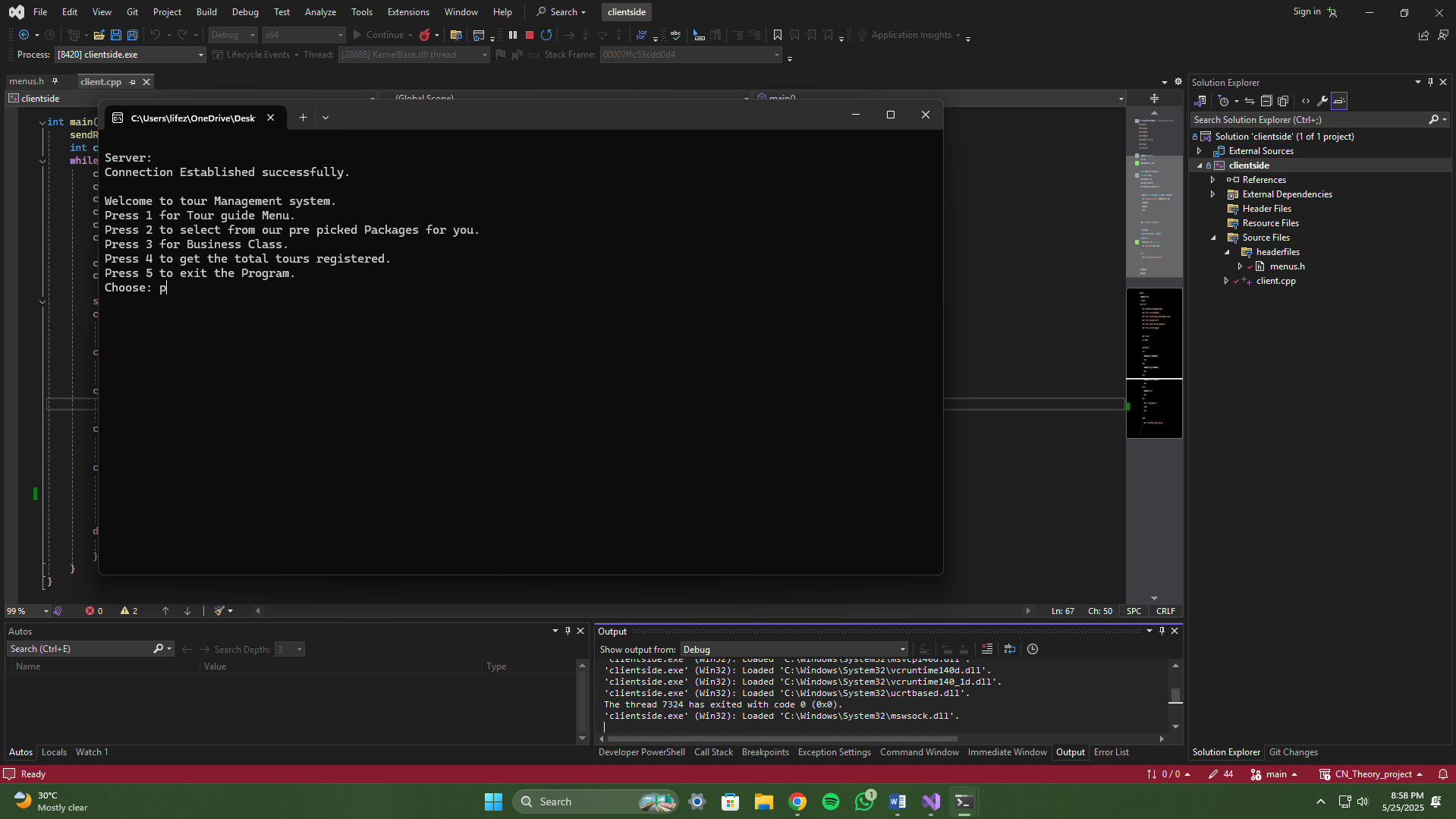
Demo:

First run the server.cpp:



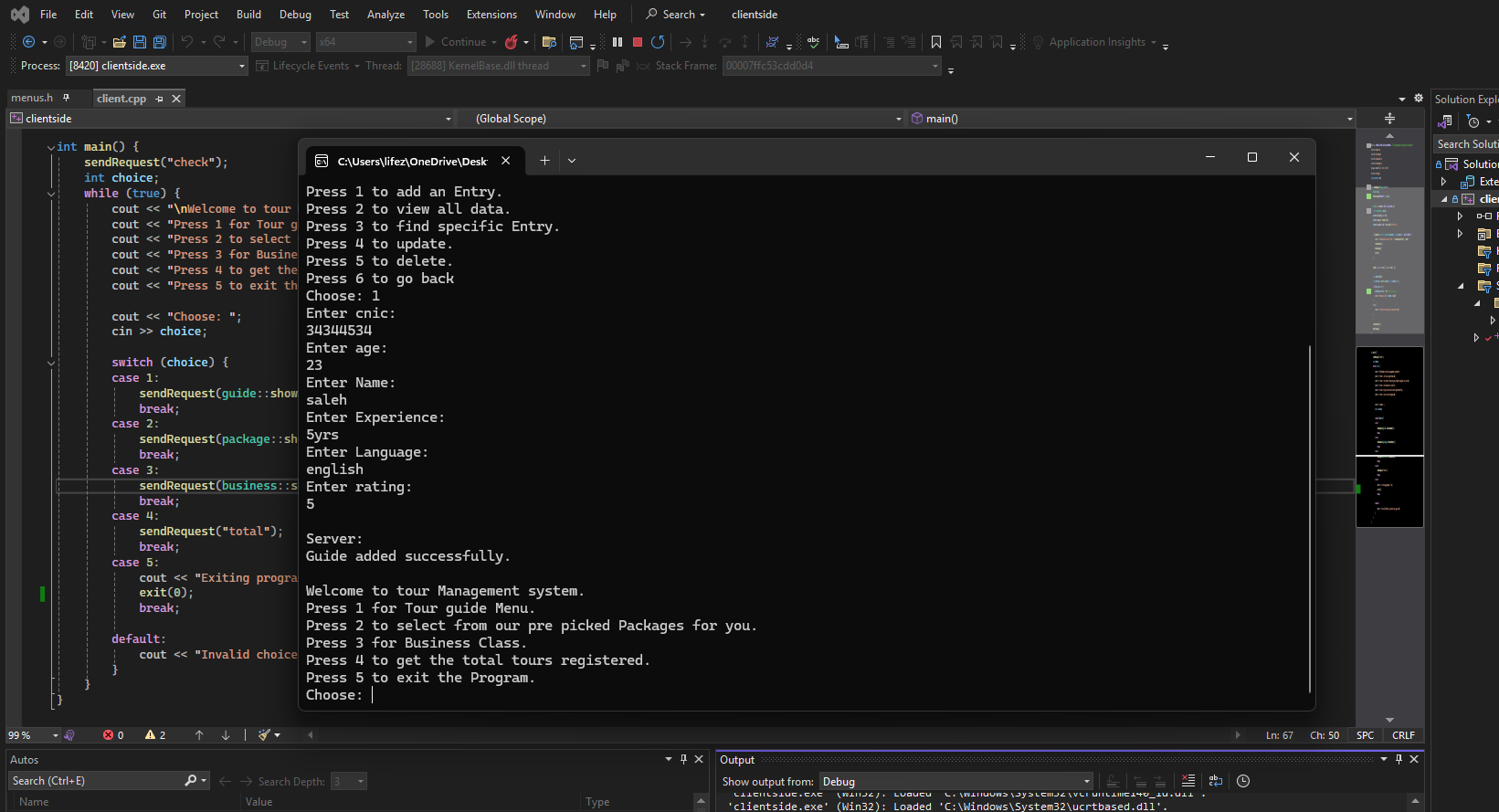
Then the client:

When the first time the client is launched, it will try to send a test packet and check the connectivity with the server. If it is successful, it will show the menu. And if not, it will show error and exit

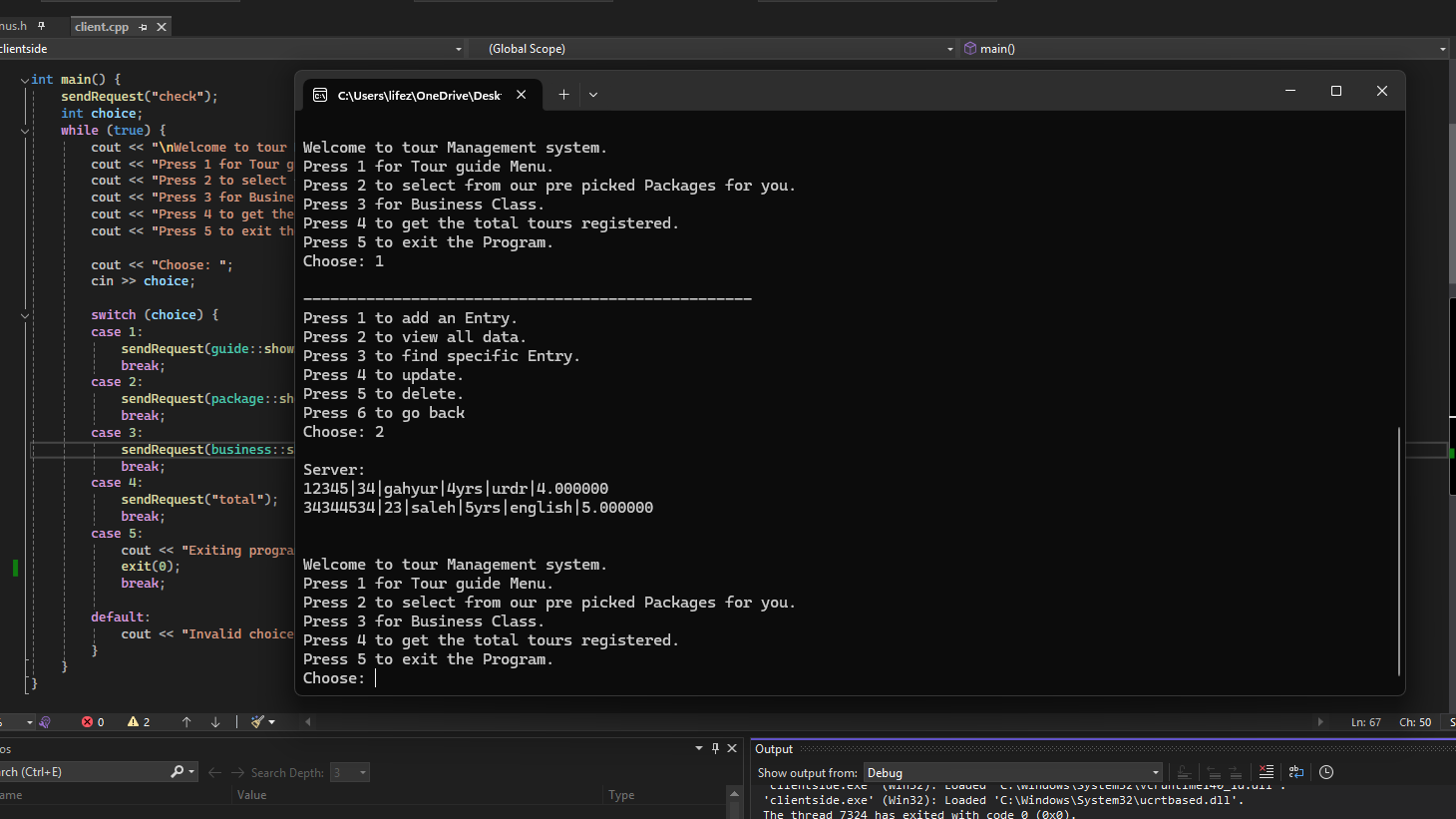


Let’s perform a simple add and read operation.

Adding:



Read:



Working flow:

⭐⭐⭐⭐⭐