

**In the name of Allah, the Most Gracious and The Merciful**



---

*Department of Computer Science*  
***Rachna College of Engineering and Technology, Gujranwala***  
*(A Constituent College of UET Lahore)*

---

**Multiplayer Game: Hangman (Console)**

**Computer Networks Lab (CSC-203L)**

**Group # 10**

2023-CS-454	Muhammad Wafa Abbas
2023-CS-456	Abdul Ahad
2023-CS-463	Abdul Rehman
2023-CS-464	Chaudhry Muneeb
2023-CS-480	Muhammad Zaman

**Semester: Fall 2024**  
**B.Sc. Computer Science**

## Introduction

This project showcases the implementation of a **CLI-based Hangman game** that operates over a **WiFi hotspot** using **laptops**. The server (run on one laptop) hosts the game, while multiple clients (other laptops) connect to it via the hotspot. The game utilizes **TCP/IP communication** for real-time interaction between the server and clients.

The primary goal of this project is to demonstrate the functioning of a **client-server application** over a local network. Players can guess letters in the Hangman game, and the server will update them on the current game state, including the word to be guessed and the number of attempts left. The game is played in a **command-line interface (CLI)**, where the server displays game prompts, and the client sends guesses via the terminal.

---

## LAN Using Home Router of WiFi Hotspot

### Details of How LAN is Created:

In this project, a **WiFi hotspot** is set up on one **laptop** acting as the **server**. Other laptops, acting as **clients**, connect to this hotspot to form a **Local Area Network (LAN)**. All the laptops can then communicate with each other over the same network, allowing the **server** to send updates and receive input from the **clients** in real-time.

### Devices Used:

- **Laptop** acting as the **server** (hosting the Hangman game).
- **Laptops** acting as **clients** (playing the game).
- **WiFi hotspot** created by one laptop, with others connecting to it.

### Working of LAN Using WiFi Hotspot:

1. The **server** laptop creates the WiFi hotspot, allowing the other laptops to connect to it.
2. Each **client** connects to the WiFi hotspot using its laptop and starts the game client.
3. The **server** listens for client connections on a specific **TCP port** (e.g., 12345) and waits for players to make guesses.
4. When a player inputs their guess, the **client** sends it to the **server** over the TCP connection.
5. The **server** processes the guess, updates the game state, and sends the updated status (e.g., word progress, remaining attempts) back to all connected clients.

### Screenshots of Testing and Working:

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\PC>ping 192.168.174.209

Pinging 192.168.174.209 with 32 bytes of data:
Reply from 192.168.174.209: bytes=32 time=329ms TTL=128
Reply from 192.168.174.209: bytes=32 time=261ms TTL=128
Reply from 192.168.174.209: bytes=32 time=296ms TTL=128
Reply from 192.168.174.209: bytes=32 time=334ms TTL=128

Ping statistics for 192.168.174.209:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 261ms, Maximum = 334ms, Average = 305ms

C:\Users\PC>
```

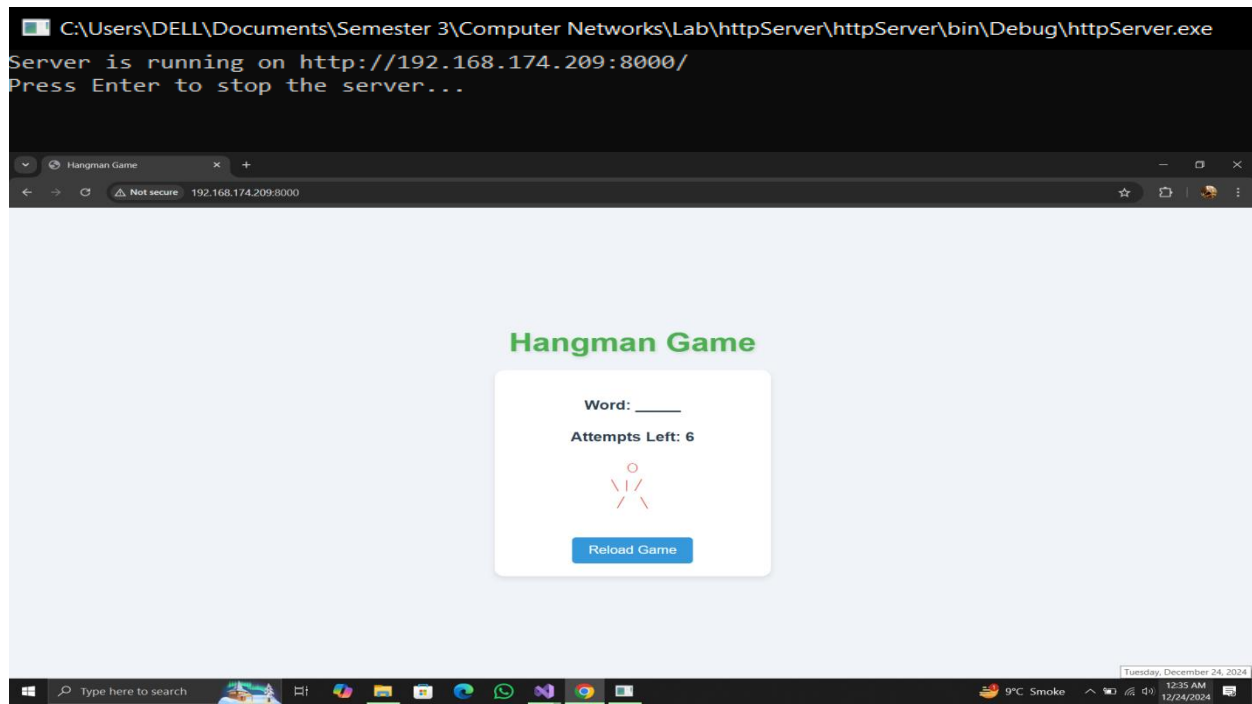
---

## HTTP Webserver (TCP/IP Communication)

### Technologies/ APIs/ Classes Used:

- **TCP/IP Protocol** for communication between the server and clients.
- **C#** programming language for implementing both the client and server applications.
- **.NET Libraries** used:
  - System.Net.Sockets for handling TCP connections.
  - System.Text for encoding and decoding messages.
  - System.IO to load game-related data like word lists.
- **Socket Programming** for managing server-client communication.

### Screenshots of Outputs:



## Brief Description of Working:

1. The **server** is hosted on a laptop that listens for incoming client connections on a pre-defined TCP port.
  2. Clients (other laptops) connect to the server and send their guesses over the network.
  3. After each guess, the **server** updates the game state (like the word progress and remaining attempts) and sends the updated information to the clients.
  4. The game continues until the word is guessed or the number of attempts runs out.
- 

## Application Task/Project

### Introduction/Description of Project:

The **Hangman Game** in this project is a **client-server application** played through a **command-line interface (CLI)**. The **server** runs on one laptop, managing the game's logic and state, while **clients** (other laptops connected to the WiFi hotspot) interact with the server by sending guesses. The server responds with updated game information, and the game continues until a player guesses the word or exhausts all attempts.

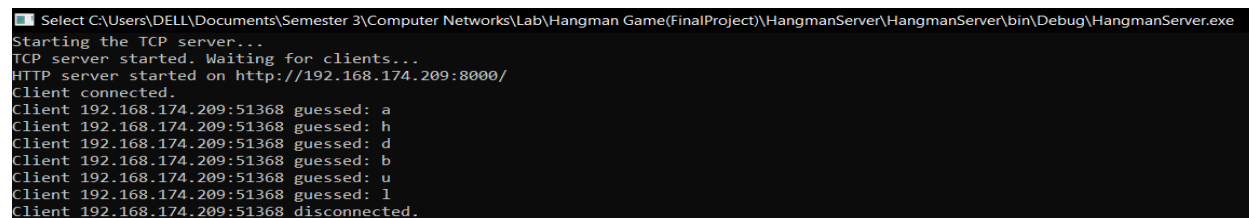
### How It Works:

- The **server** listens on a specific port (e.g., 12345) for client connections and waits for guesses from connected players.
- When a player guesses a letter, the **client** sends the guess to the **server**, which processes it and sends back the updated word state.
- The **game** continues in a loop, allowing players to keep guessing until they win or lose.

### Class/Tools/APIs/Libraries Used:

- **C#** programming language for the server and client applications.
- **System.Net.Sockets** for network communication.
- **System.Text** for encoding and decoding messages between the server and clients.
- **System.IO** for reading game-related data (such as the word list).
- **CLI-based game** where users input guesses via the terminal.

### Screenshots of Output:



```
Select C:\Users\DELL\Documents\Semester 3\Computer Networks\Lab\Hangman Game(FinalProject)\HangmanServer\HangmanServer\bin\Debug\HangmanServer.exe
Starting the TCP server...
TCP server started. Waiting for clients...
HTTP server started on http://192.168.174.209:8000/
Client connected.
Client 192.168.174.209:51368 guessed: a
Client 192.168.174.209:51368 guessed: h
Client 192.168.174.209:51368 guessed: d
Client 192.168.174.209:51368 guessed: b
Client 192.168.174.209:51368 guessed: u
Client 192.168.174.209:51368 guessed: l
Client 192.168.174.209:51368 disconnected.
```

```
C:\Users\DELL\Documents\Semester 3\Computer Networks\Lab\Hangman Game(FinalProject)\HangmanClient\HangmanClient\bin\Debug\HangmanClient.exe
Enter the server IP address:
192.168.174.209
Connecting to the server...
Connected to the server!
Word:
Attempts left: 6

Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
a
Word: a _ _ a _
Attempts left: 6

Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
h
Word: a _ _ a h _
Attempts left: 6

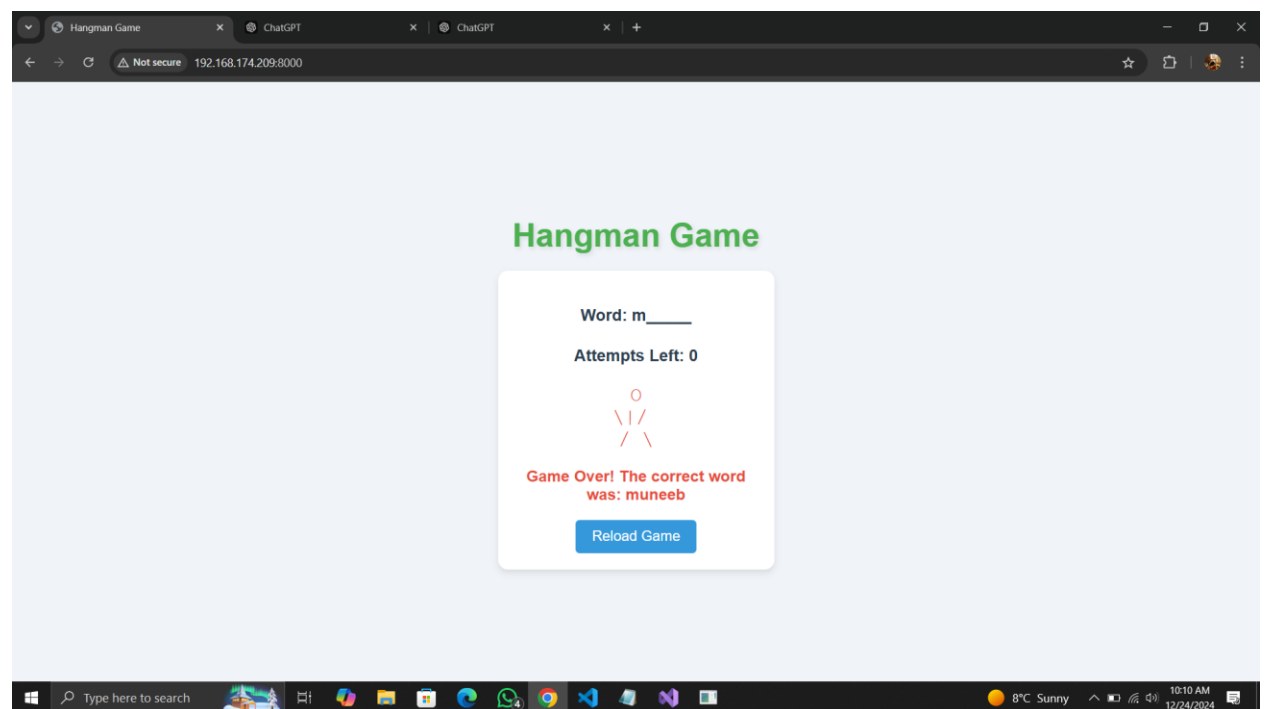
Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
d
Word: a d _ a h d
Attempts left: 6

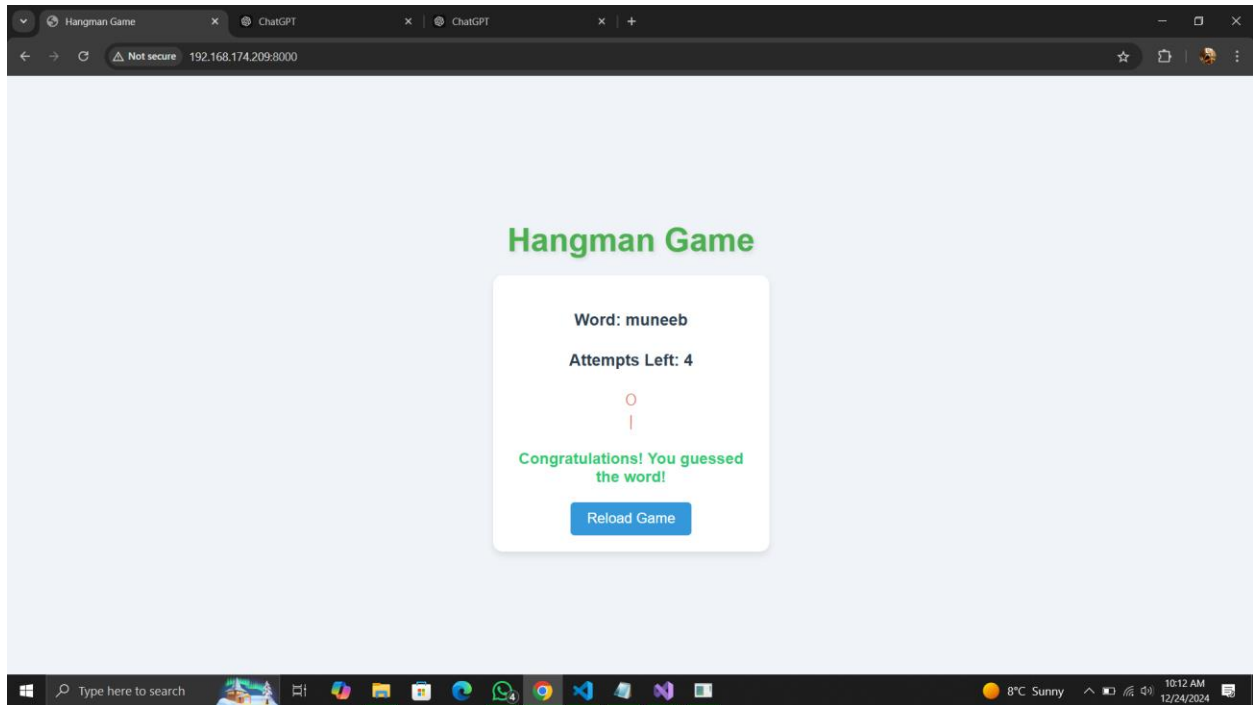
Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
b
Word: a b d _ a h d
Attempts left: 6

Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
u
Word: a b d u _ a h d
Attempts left: 6

Hangman Drawing:
  o
  \|/
  / \
Enter your guess:
l
Word: a b d u l _ a h d
Attempts left: 6

You guessed the word! Game Over.
Game over! Would you like to play again? (yes/no)
_
```





## Mobile Interface:

