1. **Project Title:** ONLINE CASINO NETWORK MANAGEMENT
2. **Project Description**

• Overview: Online Casino Network Management refers to a platform for gambling and entertainment where a variety of games of chance can be played. Additionally, agents can earn commissions by recruiting other players.

• Objective: To build a system that allows agents to track their earnings and display their downlines.

• Scope: This system focuses solely on recruiting agents and tracking their commissions per player for playing the game. It does not handle any other financial aspects of the game.

1. **Features**

* **Adding new agents** to the network, where each new agent is assigned a commission and placed as a child in the network tree.
* **Displaying the entire network tree**, showing agents and their commissions.
* **Displaying downlines** of a specific agent, where each agent's downlines (both left and right children) and their commissions are listed.
* **Searching for a specific agent** in the network based on the agent's name.

1. **Technologies Used**

• Programming Languages: C Langauge (Code Blocks)

1. **Project Structure**

CasinoNetwork/

│

├── main.c Contains the main function and menu handling logic

├── agent.c Contains functions like create\_agent, add\_agent, etc.

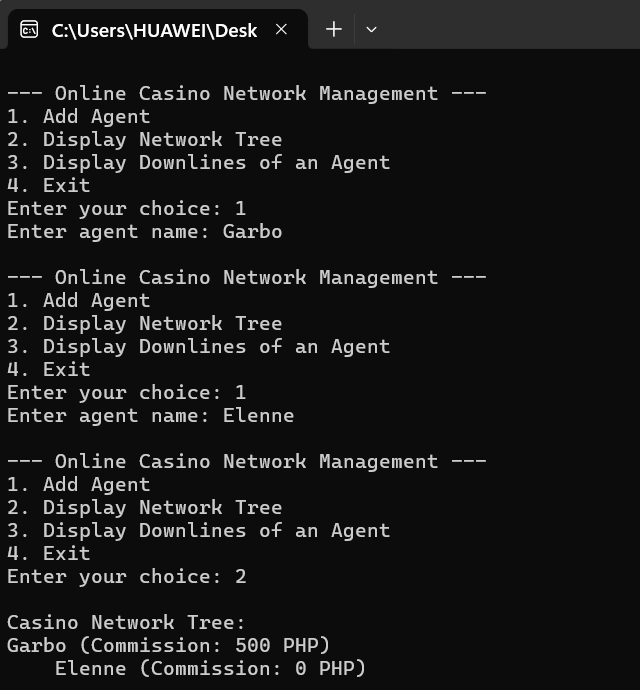
├── agent.h Header file for declaring Agent-related functions and structures

├── display.c Contains display\_tree and display\_downlines functions

├── display.h Header file for declaring display-related functions

└── Makefile Optional: Contains build instructions

1. **Usage**

****

1. **Testing**

**Compile and Run the Code**

* Save the code to a file, for example, casino\_network.c.
* Open a terminal or command prompt and navigate to the file's directory.
* Compile the code using a C compiler

**Use the Menu**

When the program runs, you'll see a menu with the following options:

--- Online Casino Network Management ---

1. Add Agent

2. Display Network Tree

3. Display Downlines of an Agent

4. Exit

Enter your choice:

**Add an Agent**

* Select option 1 to add an agent to the network.
* Enter the agent's name when prompted. Example:

Enter agent name: Alice

* The agent will be added to the network as per the binary tree structure.

**Display the Network Tree**

* Select option 2 to display the hierarchical structure of the network.
* The network is displayed as a tree, with each agent's name and commission shown.
* Example output:

Alice (Commission: 1000 PHP)

Bob (Commission: 0 PHP)

Charlie (Commission: 0 PHP)

**Display Downlines of an Agent**

* Select option 3 to view the downlines (left and right children) of a specific agent.
* Enter the agent's name when prompted. Example:

Enter agent name to display downlines: Alice

* The program will display all direct and indirect downlines of the specified agent, along with their commissions.

**Exit the Program**

* Select option 4 to exit the program.

1. **Contributions**

**Fork the Repository**

* Visit the repository hosting this code on GitHub.
* Click the **"Fork"** button at the top-right of the repository page to create your copy.

**Clone the Forked Repository**

* Copy the URL of your forked repository.

**Create a New Branch**

* Create a branch to work on a new feature or fix.

**Make Your Changes**

* Open the project files in a code editor (Code Blocks).
* Implement your changes or new features in the code.
* Save your changes

**Test the Changes**

* Compile the program to ensure your changes work correctly

**Commit the Changes**

* Add your changes to the staging area

**Push the Changes**

* Push your changes to your forked repository

S**ubmit a Pull Request**

* Go to your forked repository on GitHub.
* Click the **"Compare & Pull Request"** button.
* Add a title and description for your pull request (PR).
* Submit the PR for review.

1. **Acknowledgments**

We would like to express our deepest gratitude to everyone who contributed to the successful completion of this research.

First and foremost, we extend our sincere thanks to Mr. Kenneth Roi Novabos our subject Teacher, for his invaluable guidance, constructive feedback, and constant support throughout this study. Their expertise and encouragement have been instrumental in shaping this work.

We also wish to acknowledge ChatGPT and BlackBox,Ai for providing the resources and facilities necessary for conducting our research.

Special thanks to our peers and colleagues for their insightful discussions and moral support, which motivated us to persevere through challenges.

Finally, we are profoundly grateful to our families and loved ones for their unwavering encouragement and understanding, which served as a pillar of strength throughout this endeavor. To all those who, in one way or another, made this research possible, we extend our heartfelt appreciation.

1. **Contact Information**

• Author: Your Name

• Email: [your.email@example.com](mailto:your.email@example.com)

• GitHub: https://github.com/username