Royale Company Network System

Overview:

The **Royale Company Network System** is designed to manage a hierarchical network of members within a company. The system allows users to create a binary tree structure representing the relationships between members, where each member can have up to two downlines (left and right). The application provides functionalities to add new members, display downlines, and visualize the entire network tree. It also tracks commissions earned by members based on their recruitment activities.

Objective:

1. **To Develop a Hierarchical Network Structure**: Create a binary tree representation of members in the company, allowing for efficient management of relationships and downlines.

2. **To Implement Member Recruitment Logic**: Enable users to add new members to the network while ensuring that the rules for downline placement are followed

3. **To Provide Commission Tracking**: Implement a system for tracking commissions earned by members based on their recruitment activities, encouraging growth within the network.

4. **To Facilitate Member Search and Display**: Allow users to search for specific members and display their downlines and commissions, enhancing the usability of the system.

5. **To Create a Visual Representation of the Network**: Develop a method to display the entire network tree in a clear and structured format, making it easier for users to understand the hierarchy.

6. **To Enhance User Experience**: Design a user-friendly interface that guides users through the various functionalities of the system, making it easy to navigate and operate.

Scope:

Royale Company Network System project outlines the functionalities, features, and limitations of the application. It defines what the project will deliver and the boundaries within which it will operate.

Limitations

1. **Binary Tree Structure:** Each member can have a maximum of two downlines, limiting network complexity.

2. **Single User Access:** Designed for one user at a time; no multi-user functionality.

3. **Basic User Interface:** Text-based interface may lack intuitiveness compared to graphical interfaces.

4. **Limited Error Handling:** Basic error handling may not cover all edge cases.

5. **No Advanced Features:** Lacks user authentication, role management, and reporting capabilities.

6. **No Graphical Visualization**: Network tree is displayed in text format without graphical tools.

Boundaries

1. **Project Focus:** Manages a hierarchical network of members specifically for a company.

2. **Functionality**: Includes adding members, managing downlines, and tracking commissions; does not extend to marketing tools or integrations.

3. **Target Audience:** Intended for company administrators or managers.

Conclusion

The scope outlines the limitations and boundaries of the Royale Company Network System, focusing on essential functionalities for managing a member network while setting realistic expectations for its capabilities.

Features:

1. **Member Creation:**

   - Users can create a new member in the network by specifying their name. Each member is initialized with a commission of zero and can have two downlines (left and right).

2. **Member Search:**

   - The system allows users to search for existing members by name. This is useful for verifying if a member already exists before adding a new downline.

3. **Adding Downlines:**
   - Users can add new members as downlines under existing members, with the option to place them in either the left or right position. If the recruiting member successfully recruits both a left and a right downline, they will receive an additional commission of P500.
4. **Commission Management:**

   - The system automatically calculates and updates the commissions of parent members based on the commissions of their downlines. Specifically, a parent member earns 50% of each child's commission.

5. **Displaying Downlines:**

   - The system provides functionality to display the downlines of a specific member, showing each downline's name and commission in a hierarchical format.

6. **Displaying the Entire Network:**

   - Users can visualize the entire network of members in a pre-order format, which shows the structure of the network along with each member's commission.

7. **Total Commission Calculation:**

   - The system can calculate and display the total commission earned by all members in the network, providing an overview of the network's financial performance.

8. **Counting Total Members:**

   - Users can count the total number of members in the network, which helps in understanding the size of the network.

9. **User -Friendly Interface:**

   - The program features a menu-driven interface that allows users to easily navigate through different functionalities. It prompts users for input and provides feedback on actions taken.

10. **Memory Management:**

   - The system includes functionality to free allocated memory for the entire network when the program exits, preventing memory leaks.

11. **Console Utilities:**

   - The program includes utility functions to clear the console screen and wait for user input, enhancing the overall user experience.

12. **Save File:**

   - Function to save data without Losing the Tree

Technologies Used:

Programming Language: C

Tools: C Compiler (GCC) Text editor (like Notepad++, Sublime Text, VS Code)

Project Structure:

RoyaleCompanyNetwork/

├── main.c   -Contains all the program logic

├── bin/   - Folder for the compiled executable

├── obj/   -Folder for intermediate object files

├── RoyaleCompanyNetwork.cbp   - Code::Blocks project file

Usage:

```
=================================================
|| Welcome to the Royale Company Network System ||
=================================================

Enter the name of the head of the Company: Casiano

Press Enter to continue...
```

```
==================== Royale Company Network Menu ====================
1. Add New Member
2. Display Downlines of a Member
3. Display Entire Network Tree
4. Display Total Commission of Network
5. Count Total Members
6. Exit
=====================================================================
Enter your choice: 1

Press Enter to continue...
```

```
Enter the recruiter's name: Casiano
Enter the name of the new member to be recruited by Casiano: Lean
Enter position (1 for left, 2 for right): 1
Downline Lean added successfully under Casiano.

Press Enter to continue...
```

```
Enter the recruiter's name: Casiano
Enter the name of the new member to be recruited by Casiano: Jared
Enter position (1 for left, 2 for right): 2
Downline Jared added successfully under Casiano.

Press Enter to continue...
```

```
Enter name of the member: Casiano
-> Casiano (P 500)
     -> Lean (P 0)
     -> Jared (P 0)

Press Enter to continue...
```

```
=================== Royale Company Network Tree ====================
-> Casiano (P500)
    -> Lean (P0)
    -> Jared (P0)
====================================================================

Press Enter to continue...
```

```
Total Commission of the entire network: P500

Press Enter to continue...
```

```
Total number of members in the network: 3

Press Enter to continue...
```

Contributions:

1. Fork the repository and clone it to your machine.

2. Create a branch: git checkout -b feature-branch.

3. Make changes and ensure they are well-tested.

4. Commit: git commit -m "Add feature".

5. Push: git push origin feature-branch.

6. Submit a pull request with a clear description.

GitHub - deraj101/RoyaleCompany

Acknowledgments:

We would like to extend our heartfelt thanks to all contributors, collaborators, and developers who played a vital role in the creation and success of this project. Additionally, we acknowledge the open-source libraries and external resources that provided invaluable support, tools, and inspiration throughout the development process. Your efforts and contributions are deeply appreciated and have made this project possible and also to our beloved teacher that make us this project happen our grateful thanks to our teacher.
https://stackoverflow.com/questions/12574506/binary-tree-in-c

blackbox

Contact Information:

• Author: **Group 6 :** Rich Lean Drew Casiano **,** Jared Cuerbo **,** Ivan Fernandez

, Carl Honmar Conde **,** Lawrence Ledesma , Kissha Babe Tibay **,** Matthew Pepito

• Email: richmandawecasiano@gmail.com

 • GitHub: https://github.com/deraj101/C