**Project Title:**

**“Modernizing the Concept of Hierarchical Referral-Based Discount”**

**Project Description:**

This program focuses the idea on how the Hierarchical Referral-Based Discount works in digitalized flatform; through the logic of the program, it helps the user to make it more efficient that helps businesses attract user and foster a community effect.

Hierarchical Referral-Based Discount aims to solve the following problem of ineffective referral data tracking, inaccurate calculation of valid value discounts, and reduce the complexity of digitalized usage of this program.

This project acts as a marketing strategy to incentivize customers to recommend others by offering discounts at various points in the referral chain. One of the program's limitations is that users can only register using the company's authorized digital form. Additionally, the program's portability is limited to the company's built-in workstations, which protects the provided program.

**Features:**

**Feature 1: User Registration and Login Algorithm**

* Users can register with a referral code, generating unique referral and creates ID and PIN for themselves.
* Users can log in using their created ID and PIN.

**Feature 2: Referral Tracking and Referral Unlinking**

* Users can view who has used their referral code.
* User can unlink who uses their referral as long as that referral does not have any other associated account.

**Feature 3: Discount Updates**

* Discounts are updated dynamically based on the number of referrals.

**Feature 4: Root Node Maintenance and Checking of Referral Usage**

* Ensures the ROOT/HEAD node is always present in the line.

**Technologies Used:**

* **Programming Language:** C programming Language
* **Frameworks/Libraries:** stdio.h, stdlib.h, string.h, windows.h, conio.h, ctype.h
* **Tools**: GCC for compilation. VS Code or any text editor for code editing.
* **Database:** Flat-file storage (users.txt).

**Project Structure:**

/Application Files

/Program

|── admin.h // An independent function that admin takes in when logging in

|── functions.h // Shared function and logics belongs to here

|── homepage.h // Function were user takes after pin and id are valid

|── loginNregister.h // A function that takes the evaluation of logging data and registering data

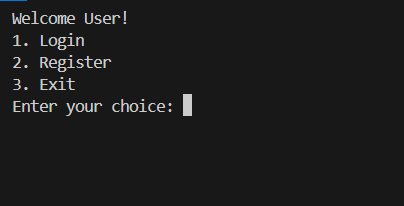
|── nodesNprototypes.h // This holds the identifier and variable function name as well as

|── users.txt // The base of data that will append when the user will register

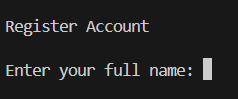
|── run\_Here.c // The function of the runnable source program

|── run\_Here.exe // The executable application to run the whole operation

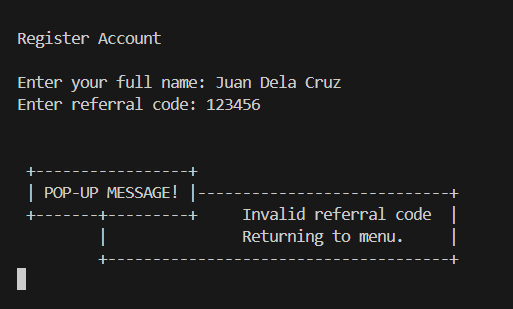
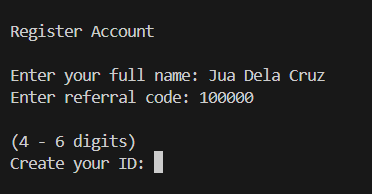
|── referrals.txt // The base of data that will append when the user uses their referral codes

**Usage:**

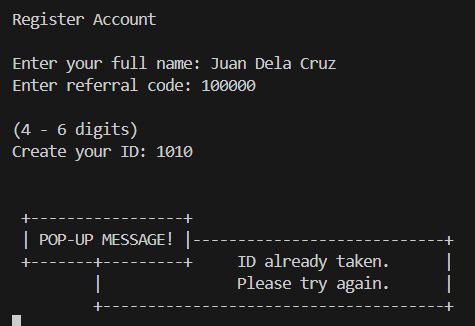
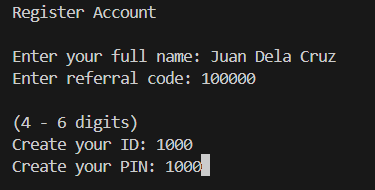
The porgram will first appear the menu text.

The user must essentially create an account since they cannot log in without one.

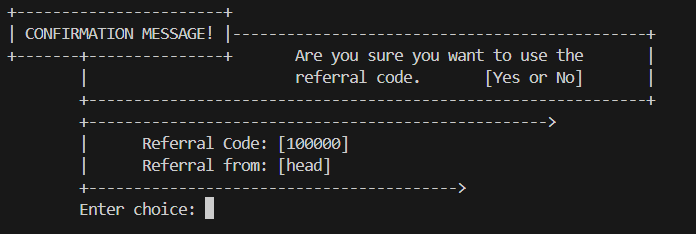
After the user entered its full name then user must go through referral linking.

If referral is not valid If its valid

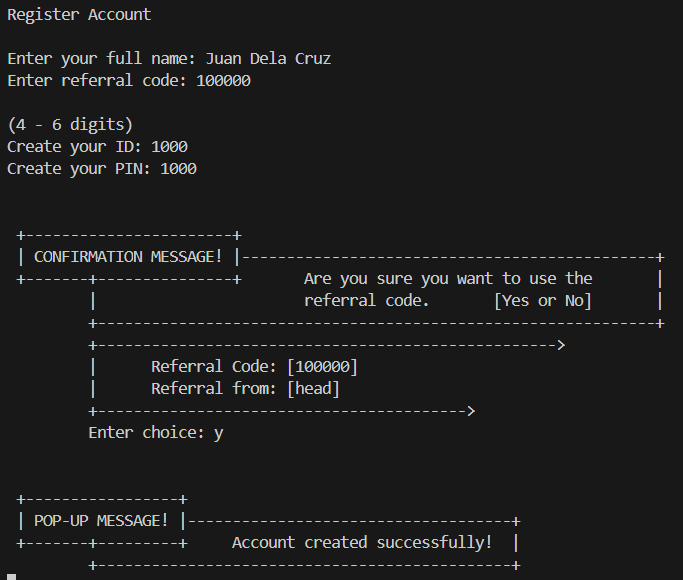
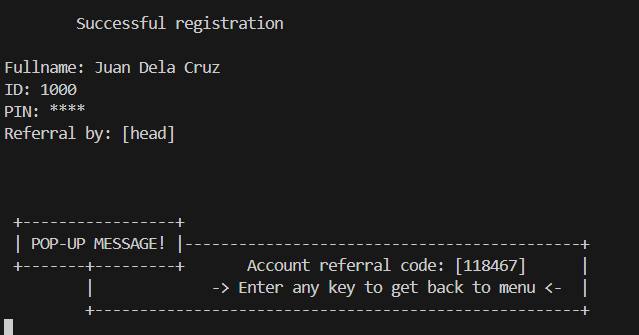
The user can create its unique ID and pin if it is valid. And if the ID is not unique then program will prompt the user:

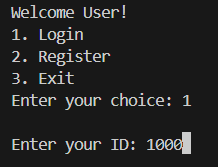
A confirmation message will be displayed by the program once the user has created their ID and PIN.



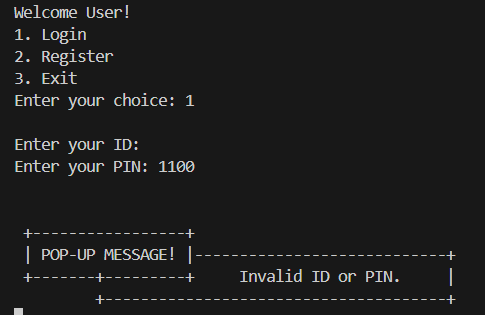
if the user chose Y or YES, then prompt the user details including its unique referral code, and command the user to enter any key

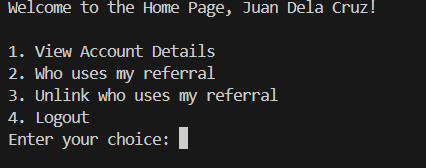


If the user wants to log-in make sure that his/her pin would me match the time the user created its data.



If the inputted ID and PIN, thee program will prompt In alternative way if the data match the

an invalid ID or PIN. correct ID and PIN according to our logic and with the help of data base then the user will be welcome. And go directly to homepage.

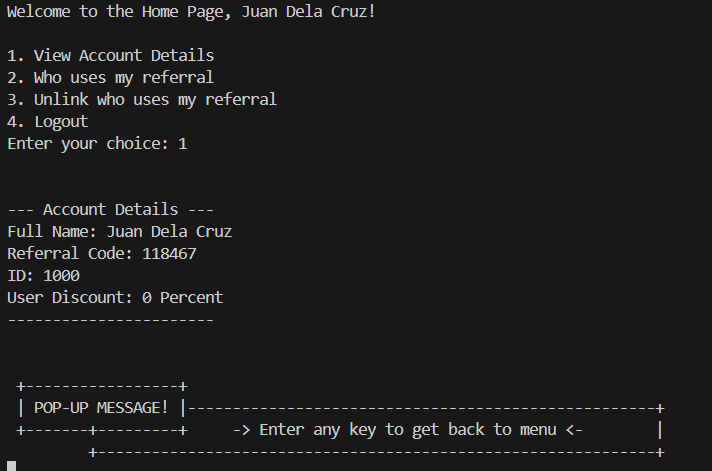


If the user selects the following:

Choice 1: The loading screen will appear for about 3 - 4 seconds

And after, your data will appear on the screen (PIN is not included)

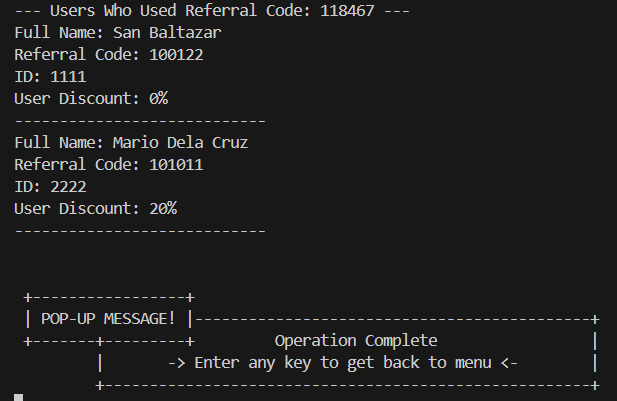
Then, the program will command the user to enter any key in order to get back to menu and proceed to another operation (if the user wants to).

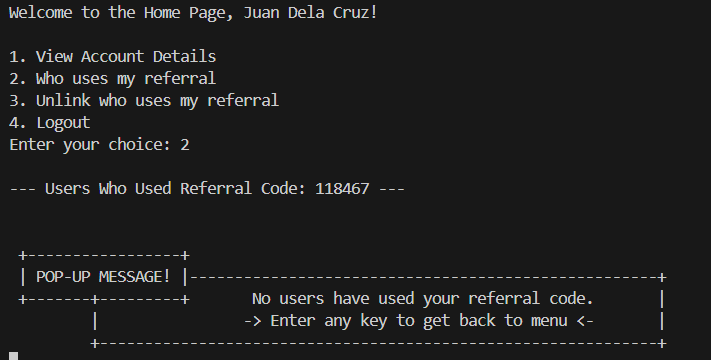


Choice 2: The loading screen will appear for about 3 - 4 seconds

In this function the one that uses your referral will append in this site:

If someone makes use of your referral code. And if no one uses, then:





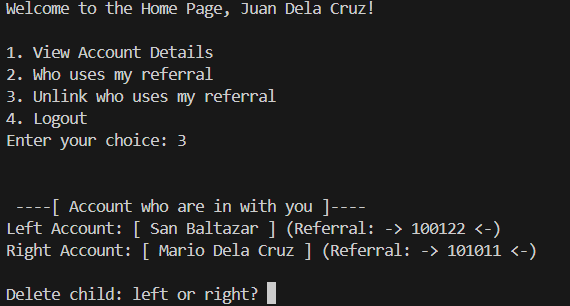
Then, the program will command the user to enter any key in order to get back to menu and proceed to another operation (if the user wants to).

Choice 3: The loading screen will appear for about 3 - 4 seconds

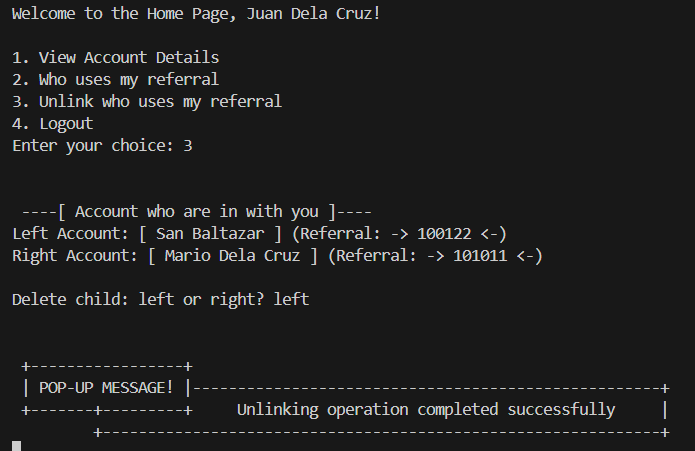
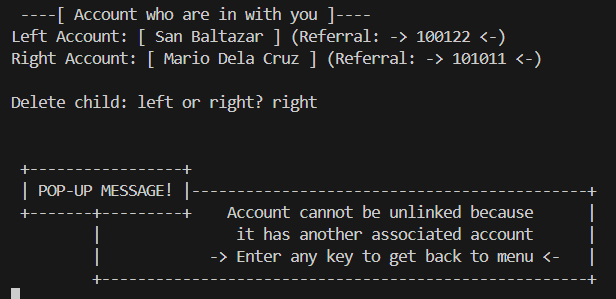
Right in this choice, the program implements this kind of algorithm that the user can unlink the one who uses its own referral.

Although the user is free to choose whether or not to unlink, the program ensures that the user can do so as long as there isn't an account linked.

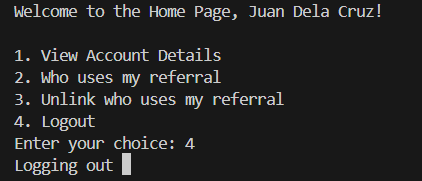
The user may only have the two side of choices: left or right.



If there is no associated account for the selected user: If the selected user has an associated account:



Choice 4: The loading screen will appear for about 3 - 4 seconds then the user is Logged out



ADMIN have feature also that can only view the users that was in the data base.

ADMIN ID and PIN: Id: 0 And Pin:0

Briefing on Admin area:

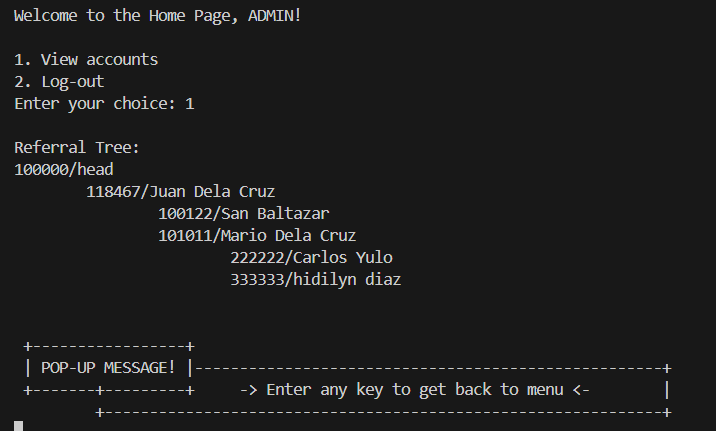
Admin can see the printed data according to who invited them

Admin see the hierarchical positions of data.

Admin only have the limitations of getting and printing the data

The data that was print are the following:

User Referral and User’s Full name



**Functionality Tests:**

1. Run the program.
2. User will see the user interface menu.
3. The program automatically checks if the data base was in the list (users.txt And referrals.txt).
4. The program automatically checks the data Base if the scanned line has the admin data and root data.
5. If the user can register their account through this referral link “100000”, then the program is correctly lunch.

**Acknowledgements:**

**Project Team**

* **Vence Peter D. Doble**
* **Christine D. Conde**
* **John P. Cadaro**
* **Alexa Suzane D. Loon**
* **Bea T. Dejito**
* **Kenneth Brian B. Tangkay**
* **Raniza A. Pepito**

**Academic Support**

* BSIT 2A CC214L Data Structures and Algorithms Course

**Technical Resources**

* C Libraries Built-in Function
  + stdio.h
  + stdlib.h
  + string.h
  + windows.h
  + conio.h
  + ctype.h

**Resources:**

* Educational online materials on binary tree structures and other essentials of program
  + <https://www.w3schools.com/>
  + <https://www.programiz.com/>
  + <https://www.reddit.com/r/cpp_questions/comments/swijos/why_does_everyone_declare_functions_in_headers/>
  + <https://www.geeksforgeeks.org/>
  + <https://wahyu-ehs.medium.com/compiling-and-linking-on-c-programming-b22eb634d27>
  + <https://www.javatpoint.com/>
  + <https://en.cppreference.com/w/>
  + <https://www.zenflowchart.com/guides/switch-case-flowchart>
  + <https://stackoverflow.com/questions/49663753/qsort-function-compare-confused-met>
  + <https://www.youtube.com/watch?v=mBT79DS8fl4>
  + <https://www.youtube.com/watch?v=hwx3C18eQwc>
  + <https://www.youtube.com/watch?v=0-kkDfCOXOI&t=449s>
  + <https://www.youtube.com/watch?v=GlybFFMXXmQ&t=48s>
  + <https://www.youtube.com/watch?v=tOQZlD-0Scc>
  + <https://www.youtube.com/watch?v=nuUEHZ4jCPQ>
  + <https://www.youtube.com/watch?v=qqHWIhNTBh0>

**Special Thanks**We would especially like to thank our professors and fellow group mates, leaders, and non-leaders for helping us develop this final project on data structures and algorithms.

**Contact Information:**

* **Author: John P. Cadaro**
* **E-mail:** johnperezcadaro@gmail.con
* **GitHub:** https://github.com/JohnPerez012