Name: Kiley Neushul

Date: 04/11/2021

Course: CIS 170C Programming with Lab

Description: Write to a File, Display Contents from that file, Delete Contents from the file. \*\*In this portion of the project, I needed to convert my code from writing to a vector to writing to a file. This process proved relatively challenging.\*\*

/\*

Developer: Kiley Neushul

Project: Final Course Project

Date: 04/11/2021

\*/

#include<iostream>

#include<iomanip>

#include<string>

#include<ctime>

#include<fstream> //Allows you to read and write to files

using namespace std;

using std::cout;

using std::cin;

//prototypes

void addRecipe();

void lookUpRecipe();

void deleteRecipe();

void showRecipes();

//Global Variables (need to be defined outside of main function to be accessible by all functions)

const string FILENAME = "recipes.txt";

int main()

{

int choice = 0;

string name, ingredients, instructions;

int cookTime, servings;

string strCookTime;

int hours, minutes;

//seed the randomizer

srand(time(0));

//get password -- three tries and you are out!

string password = "";

string appPassword = "kiley123";

bool success = false;

for (int i = 0; i < 3; i++)

{

cout << "Enter the application password: ";

cin >> password;

if (password == appPassword)

{

success = true;

break;

}

else

{

cout << "\nError. Invalid password. Remember, the password is case sensitive." << endl;

cout << "Try again.\n" << endl;

}

if (success == true)

cout << "\nWelcome to the application!\n" << endl;

else

{

cout << "/nGet away from here you hacker!\n" << endl;

system("pause");

return -1;

}

}

do

{

system("cls");

cout << "Menu" << endl;

cout << "1) Add a Recipe" << endl;

cout << "2) Look up a Recipe" << endl;

cout << "3) Delete a Recipe" << endl;

cout << "4) Show All Recipes" << endl;

cout << "5) Exit" << endl;

cout << "\nEnter your choice (1-5): ";

cin >> choice; // 3 ENTER

if (cin.fail())

{

cin.clear();

cin.ignore(100, '\n');

cout << "\nError: Please enter a number 1-3: ";

return choice = -1;

}

else

{

cin.ignore(100, '\n');

}

switch (choice)

{

case 1:

addRecipe();

break;

case 2:

lookUpRecipe();

break;

case 3:

deleteRecipe();

break;

case 4:

showRecipes();

break;

case 5:

cout << "\t\t Goodbye!\n";

break;

default:

cout << "\nError. Please Select From the Menu." << endl;

break;

}

cout << endl;

system("pause");

} while (choice != 5);

return 0;

}

void addRecipe()

{

string name, ingredients, instructions;

int cookTime, servings;

string strCookTime;

int hours, minutes;

cout << "\t\t Add a Recipe Below";

cout << endl;

//Add Recipe Name

cout << "Recipe Name: ";

getline(cin, name);

cout << endl;

//Add Recipe Ingredients (getline() used to take in more strings with whitespaces (spaces))

cout << "Ingredients: ";

//First Ingredient and Quantity

getline(cin, ingredients);

cout << endl;

//Add Recipe Instructions (getline() used to take in more strings with whitespaces (spaces))

cout << "Instructions: ";

getline(cin, instructions);

cout << endl;

//Used to take in the float variable cookTime

cout << "Specify Cook Time in Minutes: ";

cin >> cookTime;

//Conditional Statement added here to check for number of minutes. If minutes exceeds 60, display in hours

if (cookTime > 60)

{

hours = cookTime / 60;

minutes = cookTime % 60;

strCookTime = to\_string(hours) + " hours " + to\_string(minutes) + " minutes"; //"2 hours 15 minutes"

}

else

{

minutes = cookTime;

strCookTime = to\_string(minutes) + " minutes";

}

cout << endl;

// Number of servings

cout << "Number of Servings: ";

cin >> servings;

cout << endl;

//Output the File

ofstream oFile(FILENAME, ios::app);

//Write the Record (we are not writing a csv so we cannot use commas)

oFile << name << ";" << ingredients << ";" << instructions << ";" << strCookTime << ";" << servings << endl;

//Close File

oFile.close();

//Notification

cout << "Recipe information was written to the file." << endl;

}

//With this method I have been unable to locate entries in the file other than the first entry

void lookUpRecipe()

{

string findRecipe;

bool found = false;

cout << "Look up a Recipe from the List\n" << endl;

cout << "Enter Recipe to Find: ";

getline(cin, findRecipe);

//open the file

ifstream iFile(FILENAME);

if (iFile.is\_open())

{

//read each record until we find the recipe

while (iFile.peek() > -1)

{

string name, ingredients, instructions;

int cookTime, servings;

string strCookTime;

int hours, minutes;

getline(iFile, name, ';');

getline(iFile, ingredients, ';');

getline(iFile, instructions, ';');

getline(iFile, strCookTime, ';');

iFile >> servings;

iFile.ignore(1);

if (name == findRecipe)

{

found = true; //flip the boolean

cout << "\n" << name << endl;

cout << "Ingredients: " << ingredients << endl;

cout << "Instructions: " << instructions << endl;

cout << "Cook Time: " << strCookTime << endl;

cout << "Servings: " << servings << endl;

cout << endl;

break; //break out of the loop because i found the recipe

}

}

//close the file

iFile.close();

//if bool is still false, tell user the recipe was not found

if (found == false)

{

cout << "\nThe recipe was not found in our database." << endl;

}

}

else

{

cout << "\nError. Unable to open the " << FILENAME << " file." << endl;

}

}

void deleteRecipe()

{

string deleteRecipe;

bool found = false;

cout << "\t\t Delete a Recipe from the List\n" << endl;

cout << "Enter the name of the recipe that you would like to delete here: ";

getline(cin, deleteRecipe);

//open the two files ... one to read and one to write

ifstream iFile(FILENAME);

ofstream oFile(FILENAME + ".tmp");

//read every record and write the records to the temporary file

//EXCEPT the record to delete

while (iFile.peek() > -1)

{

string name, ingredients, instructions;

int cookTime, servings;

string strCookTime;

getline(iFile, name, ';');

getline(iFile, ingredients, ';');

getline(iFile, instructions, ';');

getline(iFile, strCookTime, ';');

iFile >> servings;

iFile.ignore(1);

if (name == deleteRecipe)

{

//flip the boolean

found = true;

}

else

{

//write the record to the temporary file

oFile << name << ';' << ingredients << ';' << instructions << ';' << strCookTime << ';' << servings << endl;

}

}

//close the files

iFile.close();

oFile.close();

//delete the old file

remove(FILENAME.c\_str());

//rename the temp file to the old name

rename((FILENAME + ".tmp").c\_str(), FILENAME.c\_str());

//tell the user if the record was deleted

if (found == true)

cout << "\nRecord was deleted." << endl;

else

cout << "\nRecord was not found." << endl;

}

void showRecipes()

{

string name, ingredients, instructions;

int cookTime, servings;

string strCookTime;

int hours, minutes;

cout << "\nRecipe List:\n" << endl;

cout << endl;

//open the file

ifstream iFile(FILENAME);

if (iFile.is\_open())

{

//read the data

while (iFile.peek() > -1)

{

getline(iFile, name, ';');

getline(iFile, ingredients, ';');

getline(iFile, instructions, ';');

getline(iFile, strCookTime, ';');

iFile >> servings;

iFile.ignore(1);

cout << "Recipe Name: " << name << endl;

cout << "Reipe Ingredients: " << ingredients << endl;

cout << "Recipe Instructions: " << instructions << endl;

cout << "Recipe Cook Time: " << strCookTime << endl;

cout << "Recipe Servings: " << servings << endl;

cout << endl;

}

//close file

iFile.close();

}

}

SCREENSHOTS: