#include <iostream>

#include <sstream>

#include <vector>

#include <ctime>

#include <cstdlib>

#include <iomanip>

#include <fstream>

#include <unordered\_map>

using namespace std;

class TrieNode {

public:

bool isEndOfWord;

unordered\_map<char, TrieNode\*> children;

TrieNode() {

isEndOfWord = false;

}

};

class Trie {

private:

TrieNode\* root;

public:

Trie() {

root = new TrieNode();

}

void insert(const string& word) {

TrieNode\* current = root;

for (char ch : word) {

if (!current->children[ch])

current->children[ch] = new TrieNode();

current = current->children[ch];

}

current->isEndOfWord = true;

}

bool search(const string& word) {

TrieNode\* current = root;

for (char ch : word) {

if (!current->children[ch])

return false;

current = current->children[ch];

}

return current->isEndOfWord;

}

};

int testsCompleted = 0;

double totalWPM = 0;

int totalMistakes = 0;

vector<string> splitWords(const string& str) {

vector<string> words;

string word;

stringstream ss(str);

while (ss >> word) {

words.push\_back(word);

}

return words;

}

string assignBadge(double wpm, int mistakes) {

if (wpm >= 60 && mistakes == 0) {

return "[Gold] Typing Master";

} else if (wpm >= 40 && mistakes <= 2) {

return "[Silver] Typing Expert";

} else if (wpm >= 20 && mistakes <= 4) {

return "[Bronze] Typing Learner";

} else {

return "[Practice] Keep Practicing";

}

}

void showProgress() {

if (testsCompleted == 0) return;

double averageWPM = totalWPM / testsCompleted;

cout << "\n======================" << endl;

cout << "Your Progress Summary" << endl;

cout << "Tests Completed: " << testsCompleted << endl;

cout << "Average WPM: " << fixed << setprecision(2) << averageWPM << endl;

cout << "Total Mistakes: " << totalMistakes << endl;

cout << "======================" << endl << endl;

}

void logPerformance(const string& mode, const string& prompt, const string& userInput, double wpm, int mistakes, const string& badge) {

ofstream logFile("typing\_log.txt", ios::app);

time\_t now = time(0);

char\* dt = ctime(&now);

logFile << "==============================\n";

logFile << "Date and Time: " << dt;

logFile << "Mode: " << mode << endl;

logFile << "Prompt: " << prompt << endl;

logFile << "User Input: " << userInput << endl;

logFile << "Typing Speed: " << fixed << setprecision(2) << wpm << " WPM" << endl;

logFile << "Mistakes: " << mistakes << endl;

logFile << "Badge: " << badge << endl;

logFile << "==============================\n\n";

logFile.close();

}

void displayPerformanceHistory() {

cout << "\n[DEBUG] Entered displayPerformanceHistory()\n";

ifstream logFile("typing\_log.txt");

if (!logFile.is\_open()) {

cout << "\nNo performance history found.\n" << endl;

return;

}

cout << "\n====== Performance History ======\n" << endl;

string line;

while (getline(logFile, line)) {

cout << line << endl;

}

cout << "\n=================================\n" << endl;

logFile.close();

}

int main() {

srand(time(NULL));

char mainMenuChoice;

do {

cout << "---------------------------------" << endl;

cout << "Welcome to Typing Speed Test Game" << endl;

cout << "---------------------------------" << endl;

cout << "1. Start New Typing Test" << endl;

cout << "2. View Performance History" << endl;

cout << "3. Exit" << endl;

cout << "Enter your choice: ";

cin >> mainMenuChoice;

if (cin.fail()) {

cin.clear();

cin.ignore(1000, '\n');

continue;

}

cin.ignore();

if (mainMenuChoice == '1') {

char playAgain = 'y';

while (playAgain == 'y' || playAgain == 'Y') {

cout << "\nSelect Mode:" << endl;

cout << "1. Short Words" << endl;

cout << "2. Long Words" << endl;

cout << "3. Sentences" << endl;

int choice;

cout << "Enter your choice: ";

cin >> choice;

if (cin.fail()) {

cin.clear();

cin.ignore(1000, '\n');

continue;

}

cin.ignore();

vector<string> items;

string mode;

if (choice == 1) {

items = { "cat", "dog", "sun", "pen", "box", "run", "top", "red", "hat", "bat" };

mode = "Short Words";

} else if (choice == 2) {

items = { "elephant", "crocodile", "butterfly", "knowledge", "wonderful", "avalanche", "generation", "education", "beautifully", "enthusiasm" };

mode = "Long Words";

} else if (choice == 3) {

items = {

"The five boxing wizards jump quickly.",

"Sphinx of black quartz, judge my vow.",

"How quickly daft jumping zebras vex.",

"Jived fox nymph grabs quick waltz.",

"Bright vixens jump; dozy fowl quick."

};

mode = "Sentences";

} else {

cout << "Invalid choice. Returning to main menu..." << endl;

break;

}

int total = items.size();

int random = rand() % total;

string prompt = items[random];

cout << "\nType the following as fast and accurately as you can:\n" << endl;

cout << prompt << endl << endl;

time\_t start = time(NULL);

string userInput;

getline(cin, userInput);

time\_t end = time(NULL);

double timeUsed = difftime(end, start);

if (timeUsed == 0) timeUsed = 1;

vector<string> originalWords = splitWords(prompt);

vector<string> inputWords = splitWords(userInput);

Trie trie;

for (const string& word : originalWords) {

trie.insert(word);

}

int mistakes = 0;

int minLength = min(originalWords.size(), inputWords.size());

for (int i = 0; i < minLength; ++i) {

if (!trie.search(inputWords[i])) {

mistakes++;

}

}

mistakes += abs((int)(originalWords.size() - inputWords.size()));

int charCount = userInput.length();

double wpm = ((charCount / 5.0) / timeUsed) \* 60;

string badge = assignBadge(wpm, mistakes);

cout << fixed << setprecision(2);

cout << "\nTime elapsed: " << timeUsed << " seconds" << endl;

cout << "Your typing speed: " << wpm << " WPM" << endl;

cout << "Number of word mistakes: " << mistakes << endl;

cout << "\nBadge Earned: " << badge << endl;

if (mistakes == 0) {

cout << "Excellent! Perfect typing!" << endl;

} else if (mistakes <= 2) {

cout << "Good job! Minor mistakes." << endl;

} else {

cout << "Keep practicing to improve your accuracy!" << endl;

}

testsCompleted++;

totalWPM += wpm;

totalMistakes += mistakes;

logPerformance(mode, prompt, userInput, wpm, mistakes, badge);

showProgress();

cout << "Do you want to try another test? (y/n): ";

cin >> playAgain;

cin.ignore();

cout << endl;

}

} else if (mainMenuChoice == '2') {

displayPerformanceHistory();

} else if (mainMenuChoice == '3') {

cout << "Thank you for playing! Goodbye!" << endl;

} else {

cout << "Invalid choice! Please try again.\n" << endl;

}

} while (mainMenuChoice != '3');

return 0;

}