SafeSurfer: A Website Monitor for Parents - Final Project Guide

WHAT IS THIS PROJECT?

This project is a simple program written in C++ that simulates how a parent can monitor what websites their kid is going to. If the kid visits a site that isn't on the parent's "approved list," the program will alert the parent and give a description of what the website is used for.  
  
This is a basic version that just runs in the terminal. It doesn’t really spy on your browser, but it shows how a real system might work.

# FILES IN THE PROJECT

## main.cpp:

This is where the program starts. It runs everything. It sets up the monitor, loads the allowed sites and descriptions, and checks some websites (pretending the kid visited them).

## WebsiteMonitor.h:

This is a header file. It just tells the computer what functions (like tools) our monitor has.

## WebsiteMonitor.cpp:

This is where we write what each tool does. It reads the approved sites, reads site descriptions, and checks if the visited site is good or bad.

## approved\_sites.txt:

A simple text file with a list of sites that are allowed. Parents can change this.

## site\_descriptions.txt:

Another text file. It has sites that are not allowed and explains what those sites are.

# CODE FILES EXPLAINED (LIKE A HIGH SCHOOLER WOULD)

## approved\_sites.txt

youtube.com  
pbs.org  
mathgames.com

## site\_descriptions.txt

tiktok.com | Social media app popular among teens, contains potentially inappropriate content.  
reddit.com | Community forums where users post content. May include adult material.

## WebsiteMonitor.h

#ifndef WEBSITE\_MONITOR\_H  
#define WEBSITE\_MONITOR\_H  
  
#include <string>  
#include <vector>  
#include <map>  
  
class WebsiteMonitor {  
private:  
 std::vector<std::string> approvedSites; // list of okay sites  
 std::map<std::string, std::string> siteDescriptions; // bad site + explanation  
  
public:  
 void loadApprovedSites(const std::string& filename);  
 void loadSiteDescriptions(const std::string& filename);  
 void checkVisitedSite(const std::string& site);  
};  
  
#endif

## WebsiteMonitor.cpp

#include "WebsiteMonitor.h"  
#include <iostream>  
#include <fstream>  
#include <sstream>  
  
void WebsiteMonitor::loadApprovedSites(const std::string& filename) {  
 std::ifstream file(filename); // Open the file  
 std::string site;  
 while (getline(file, site)) { // Read each line (each site)  
 approvedSites.push\_back(site); // Add to the approved list  
 }  
}  
  
void WebsiteMonitor::loadSiteDescriptions(const std::string& filename) {  
 std::ifstream file(filename); // Open the file  
 std::string line;  
 while (getline(file, line)) { // Go through each line  
 std::stringstream ss(line); // Split the line at the '|'  
 std::string site, desc;  
 getline(ss, site, '|'); // Get the site name  
 getline(ss, desc); // Get the description  
 siteDescriptions[site] = desc; // Add to map  
 }  
}  
  
void WebsiteMonitor::checkVisitedSite(const std::string& site) {  
 bool approved = false;  
 for (const auto& allowed : approvedSites) { // Check all approved sites  
 if (site == allowed) {  
 approved = true;  
 break;  
 }  
 }  
  
 if (!approved) {  
 std::cout << "[ALERT] Unapproved site visited: " << site << "\n";  
 if (siteDescriptions.count(site)) {  
 std::cout << "→ Description: " << siteDescriptions[site] << "\n";  
 } else {  
 std::cout << "→ No description available.\n";  
 }  
 } else {  
 std::cout << "[OK] Approved site: " << site << "\n";  
 }  
}

## main.cpp

#include <iostream>  
#include "WebsiteMonitor.h"  
  
int main() {  
 WebsiteMonitor wm;  
 wm.loadApprovedSites("approved\_sites.txt");  
 wm.loadSiteDescriptions("site\_descriptions.txt");  
  
 std::vector<std::string> visitedSites = {  
 "youtube.com",  
 "tiktok.com",  
 "reddit.com",  
 "mathgames.com"  
 };  
  
 for (const auto& site : visitedSites) {  
 wm.checkVisitedSite(site);  
 }  
  
 return 0;  
}

# HOW TO RUN IT (TO SHOW YOUR TEACHER)

1. Open your code editor (or terminal)  
2. Put all files in the same folder  
3. In terminal, compile it:  
 g++ main.cpp WebsiteMonitor.cpp -o monitor  
4. Then run it:  
 ./monitor  
5. It will show the output in the terminal, showing what sites are OK and which get flagged.

# WHAT TO SAY IN YOUR VIDEO OR PRESENTATION

"This is a simulation of a parental website monitor. It checks sites the kid visits against a list of safe websites. If it’s not approved, it gives a warning and explains the site. This version is for demo purposes, but it could be turned into a background app or browser plugin in the future."

# NOTE:

This is a simulation for educational use. It doesn't actually monitor real browsing, but it shows how a real tool could work.