A screenshot of a computer

AI-generated content may be incorrect.

In this project, I had to fix a bug in a C++ program that could let a user accidentally or intentionally break part of the program by typing in too much information. This is called a buffer overflow. The program had a private account number stored in memory, and if the user typed too many characters into the input box, they could overwrite that account number without even realizing it. That’s a big security problem, especially in programs that handle private data like a banking application.

To solve the issue, I changed how the input was handled. Normally, the program used std::cin, which only reads input up to the first space and doesn’t catch when the user types too much. I switched it to use std::getline, which reads everything the user types, including spaces. After getting the full line of input, I added a check to see if the user entered more than 19 characters. If they did, the program prints a warning saying “Too many characters entered! Only the first 19 will be used.” That way, the user knows they went over the limit.

I also made sure the user’s input was safely copied into a fixed-size character array. This part is important because C++ lets you write past the end of a variable if you're not careful. To prevent that, I used a safer function called strncpy\_s, which is built into Visual Studio to help protect memory. It only copies the allowed number of characters and avoids security risks.

Now, no matter how much the user types, the program will only use the first 19 characters and keep the account number safe. I tested it by entering different lengths of input, and everything worked as expected. The fix protects the program from buffer overflows and gives a clear message to the user when they type too much.