2-2 Activity: SQL Injection Coding

A screenshot of a computer

AI-generated content may be incorrect.

To improve the security of the run\_query function, I focused on preventing SQL injection attacks. Originally, the function directly inserted user input into SQL queries, which could be exploited by attackers to manipulate the query. To fix this, I used prepared statements in SQLite, which separate SQL code from user input, making it impossible for attackers to alter the query structure. I also added a check for suspicious patterns, such as semicolons and SQL comments, which are commonly used in SQL injection attempts. When preparing the query, I made sure that no dangerous characters were present before executing it. If the input is safe, the query runs, and the results are collected. If anything unusual is detected, the function stops and returns an error message. This update makes the function more secure, ensuring that user input is handled safely and preventing potential SQL injection attacks.