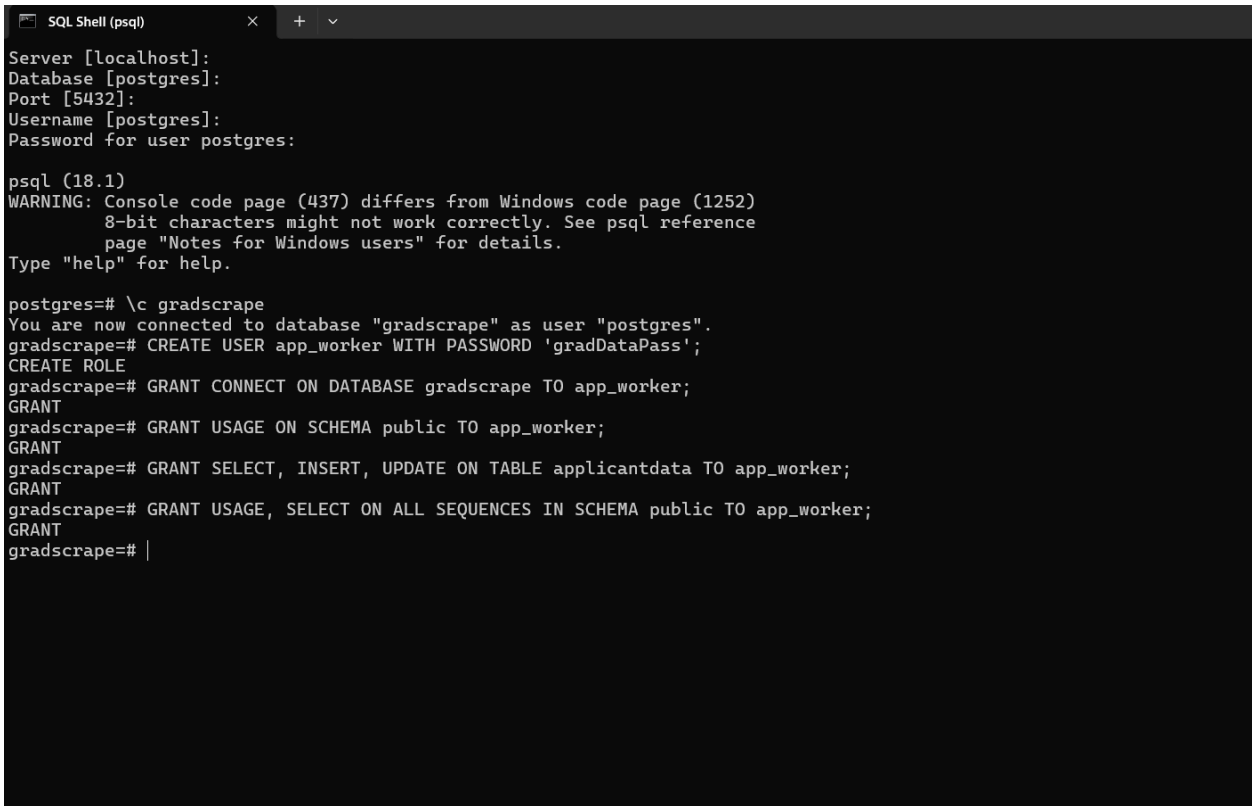
Database Hardening

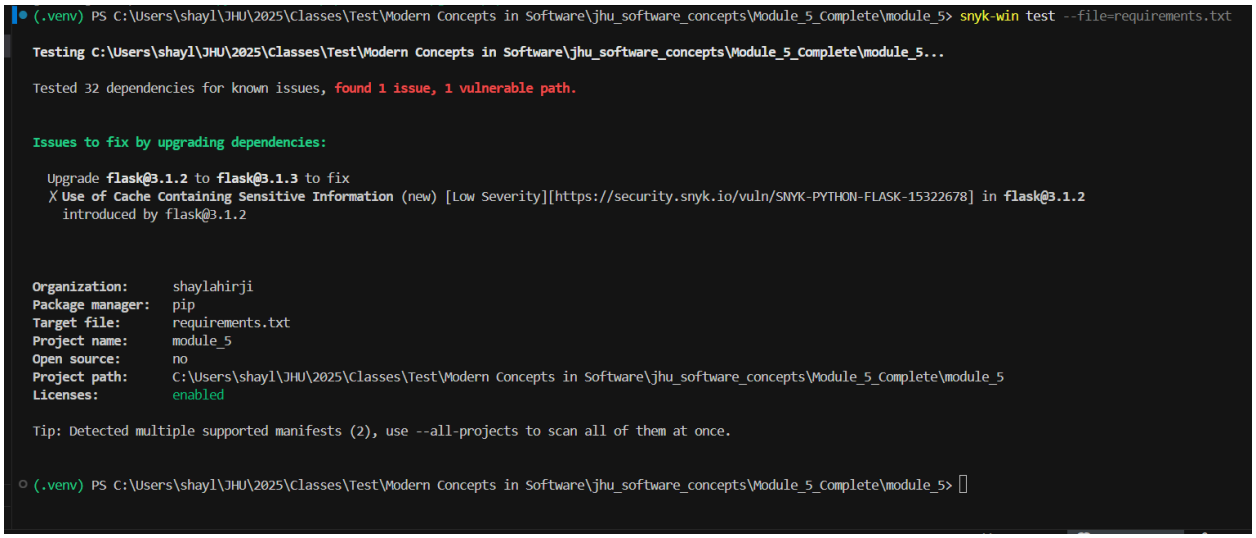


For my Least-Privilege Database user, I granted them the permissions of CONNECT, USAGE, SELECT, INSERT, and UPDATE. I gave the user CONNECT, USAGE, and SELECT permissions, so they are able to see information posted on my website. CONNECT, USAGE, and SELECT allow the user to connect to the database, look up the table associated with my website, and view it respectively. I gave the user INSERT and UPDATE permissions so that they are able to use the buttons on my webpage to scrape new data from the GradCafe survey and add it to the database, as well as update the answers to the queries on the Analysis page. These permissions are enough to allow the user to use my website as intended, without giving them any extra permissions.

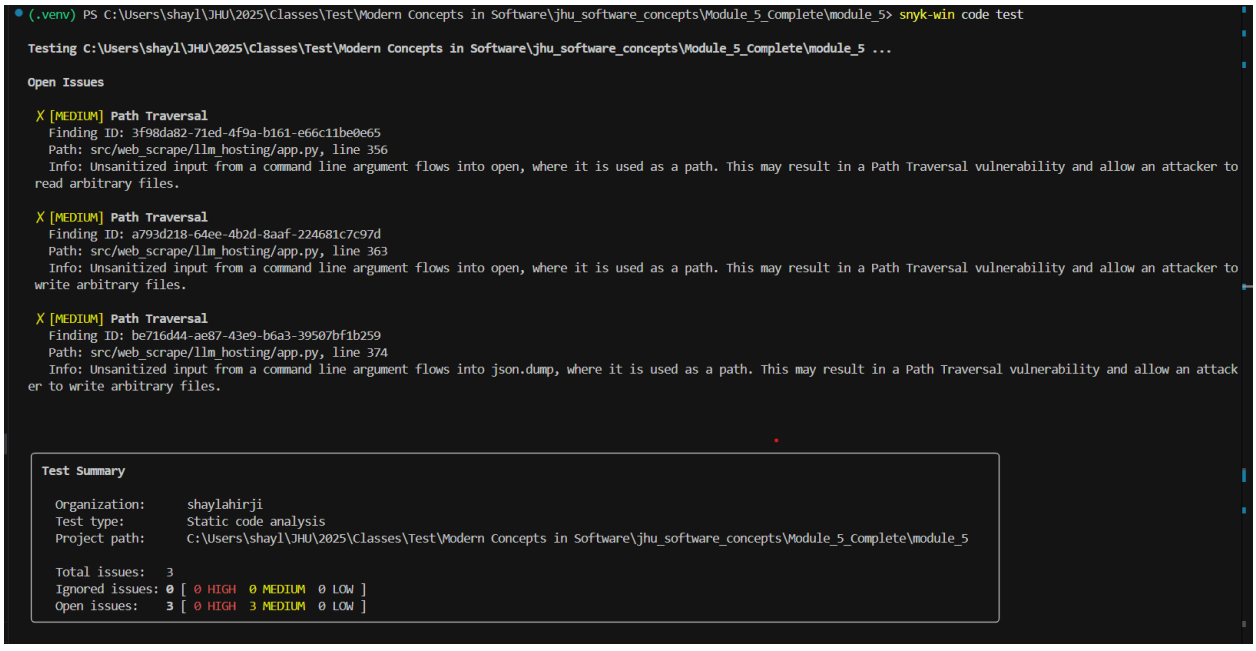
Python Dependency Graph

Based on the dependency.svg file, the key dependencies in my web app are dotenv, psycopg.sql, src.config, psycopg, flask, bs4, src.query_data, src.load_data, and web_app.run. Dotenv is a security dependency that load my environment variables into the application from the .env file. Another important security dependency is psycopg.sql, this provides tools for secure SQL queries, and helps defend against SQL Injections. Config acts as a main hub for my application settings, as it distributes authorized configuration details to the rest of my system. Flask is a key dependency for my web framework, and handles routing and requests. The key dependencies for my web scrape are bs4, which parses HTML content from The GradCafe, src.query_data, which gets information from the database, and src.load_data, which serves as an in-between for the web scraper and the database. My last key dependency is web_app.run,

which is the entry point to my web app, this is key as it initializes my flask app and ensures all dependencies are correctly linked before allowing users onto the server.



During the Snyk dependency scan, one Low Severity vulnerability was identified. My requirements.txt included flask==3.1.2, instead of flask==3.1.3, which uses cache containing sensitive information. To fix this vulnerability, I updated my requirements.txt to show flask==3.1.3 instead.



During the Snyk Code Test, 3 Medium Severity issues were identified. All three are identified to be within the llm_hosting/app.py file, and are all Path Traversal issues. The errors state that an unsanitized command line input is used as a path, which opens up the risk of attackers being able to read or write arbitrary files.