Looking at my dependency graph, my app.py serves as the main entry point that brings together all the components of my Flask application. I'm using Flask's core modules including flask-app, flask-ctx, flask-globals, flask-wrappers, flask-helpers, flask-json, flask-templating, and flask-blueprints to handle the web framework functionality. For database connectivity, I've integrated psycopg and psycopg-sql to work with PostgreSQL, which connect to a psycopg_pool system that manages connection pooling through various components like pool async, pool errors, and null pool. My custom application logic is organized in the module_5 package, specifically in module_5.src and module_5.src.query_helpers, which handles my database queries and business operations. I'm also using flask-config to manage my application settings and flask-signals for event handling. The architecture shows a clear separation between the web framework layer, database management, and my custom code, all of which come together in my app.py file.