Key dependencies in dependency.svg

app.py is the central hub at the bottom of the graph. Nearly every other source module and third-party library flows into app.py, making it the main entry point (Flask dashboard). Flask provides the web framework for routing, request handling, JSON responses, and Jinja2 template rendering. Psycopg is the POSTgreSQL adapter used by app.py, query_data, cleanup_data and load_data for all database operations. Query_data is a shared module that flows into app.py, cleanup_data, and load_data. It supports DB_CONFIG, MAX_QUERY_LIMIT and run_queries used by all database-touching modules. Cleanup_data depends on query_data (for DB_CONFIG and MAX_QUERY_LIMIT) and flows into app.py, where its fix_gre_aw() and fix_uc_universities() functions are called after data ingestion. Llm_standardizer depends on llama_cpp and llama_cpp.Llama (the local LLM inference engine) and huggingface_hub (for downloading the TinyLlama model), and flows into both app.py (for standardizing new scraped entries) and cleanup_data (sharing UC campus normalizing patterns). Bs4 (BeautifulSoup) flows into scrape providing the HTML parsing capability that extracts applicant data from GradCafe survey pages. Scrape depends on bs4 and robots_checker, and flows into app.py. It handles web scraping while robots_checker ensures compliance with the site’s robots.txt rules before fetching pages. Load_data depends on psycopg, psycopg.sql and query_data and flows into app.py. It handles the initial one-time JSON-to-POSTgreSQL data load, creating the database and table schema that all other modules rely on.