Module 5 — Software Assurance Report

Flask + PostgreSQL Grad Café Analytics

Student: Oliver

Date: February 23, 2026

1. Installation and Running (pip + venv and uv)

The project supports two fresh-install pathways: (A) pip + venv and (B) uv. Both workflows use an editable install so imports behave consistently in local runs, tests, and CI.

A) pip + venv

```
python -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
pip install -r requirements.txt
pip install -e .
```

B) uv

```
python -m venv .venv
source .venv/bin/activate
pip install uv
uv pip sync requirements.txt
pip install -e .
```

Run tests and coverage

```
pytest --cov=src --cov-fail-under=100
```

Run the Flask app

Set DB environment variables (recommended) and run the module entrypoint:

```
export DB_HOST=localhost DB_PORT=5432 DB_NAME=gradcafe DB_USER=gradcafe_app
DB_PASSWORD=...
python -m src
```

2. Dependency Graph Summary (pydeps + Graphviz)

A dependency graph was generated using pydeps and Graphviz (dot) and saved as dependency.svg at the module root. The graph shows that flask_app is the web entry point (routes), which calls query_data for analytics results and etl for loading data. Both query_data and etl depend on db for connecting to PostgreSQL and ensuring the schema exists. psycopg provides the database driver, and psycopg.sql is used to safely compose statements for dynamic components. The template layer renders results using Flask's Jinja templates, keeping presentation separate from database logic. This modular structure centralizes database access patterns and keeps route logic minimal, which improves maintainability and reduces risk.

Command used:

```
pydeps src --noshow -T svg -o dependency.svg
```

3. SQL Injection Defenses and LIMIT Enforcement

- SQL statements are constructed using psycopg SQL composition (sql.SQL / sql.Identifier / sql.Placeholder) for dynamic components.
- Variable values are never interpolated into SQL strings; they are passed as bound parameters to cursor.execute(stmt, params).
- Statement construction is separated from execution (composed SQL object vs. bound parameters).
- Every query includes an explicit LIMIT (LIMIT 1 for scalar/aggregate queries).
- Where a LIMIT could vary, code clamps the maximum allowed limit (e.g., 1–100) to prevent accidental large result sets.

4. Database Hardening (Least Privilege)

Database credentials are read from environment variables (DB_HOST, DB_PORT, DB_NAME, DB_USER, DB_PASSWORD), and no secrets are hard-coded in code. A .env.example file documents required variables, and .env is excluded via .gitignore.

Example least-privilege SQL (run as an admin):

```
CREATE USER gradcafe_app WITH PASSWORD 'STRONG_PASSWORD';
GRANT CONNECT ON DATABASE gradcafe TO gradcafe_app;
GRANT USAGE ON SCHEMA public TO gradcafe_app;
GRANT SELECT, INSERT ON TABLE applicants TO gradcafe_app;
GRANT USAGE, SELECT ON SEQUENCE applicants_p_id_seq TO gradcafe_app;
```

Rationale: the app reads analytics (SELECT) and loads rows via the ETL endpoint (INSERT). It does not require superuser privileges, DROP/ALTER, or ownership permissions.

5. Packaging (setup.py) and Reproducibility

A setup.py is included to make the project installable and to support editable installs (pip install -e .). Packaging improves consistency of imports and reduces environment-specific path issues across local development, testing, and CI.

6. Supply-Chain Scanning (Snyk)

Dependencies were scanned with Snyk CLI using snyk test. Proof is provided as snyk-analysis.png in the module root. If vulnerabilities are reported, remediation includes upgrading affected packages, removing unused dependencies, or documenting risk when no patch exists.

```
snyk auth
snyk test
```

7. CI Enforcement (GitHub Actions)

The GitHub Actions workflow enforces quality and security checks on every push and pull request:

- Pylint runs on src/ only and fails if score is below 10 (--fail-under=10).
- Dependency graph generation runs (pydeps + Graphviz) and fails if dependency.svg is missing.
- Pytest runs with a coverage threshold enforced.
- Snyk test runs when SNYK_TOKEN is configured in repository secrets (optional to fail the build depending on course expectations).