Security Requirements Elicitation Tool

Units Covered: Unit II (Assurance case) and Unit III (SQUARE process)

Short Description:

A lightweight Flask web application to help teams collect, suggest, prioritize, and export security requirements. Users add functional requirements; the app auto-suggests security controls using keyword templates inspired by the SQUARE process. Data is stored in SQLite and can be exported as an HTML assurance case report.

Features

* Add / Edit / Delete functional requirements
* Real-time auto-suggested security requirements using keyword templates
* Prioritize requirements (High / Medium / Low)
* Search and filter by keyword or category
* Dashboard stats (High / Medium / Low counts)
* Export a prioritized security assurance case report (HTML)
* Simple Bootstrap-based UI (templates included)

Tech Stack

* Python (Flask)
* Jinja2 templates (Bootstrap 5 UI)
* SQLite (requirements.db)
* Optional: NLP (NLTK / spaCy) to improve keyword detection

Files (included)

* `app.py` — Main Flask app with routes: `/`, `/add`, `/edit/<id>`, `/delete/<id>`, `/suggest`, `/export`
* `requirements.db` — SQLite DB (auto-created on first run)
* `templates/base.html` — Base layout (Bootstrap)
* `templates/index.html` — Dashboard and listing
* `templates/add.html` — Create requirement form (with realtime suggestion)
* `templates/edit.html` — Edit requirement form
* `README.md` — Project README and run instructions

> Note: The templates and `app.py` are already present in the project folder. The DB is created automatically.

Database Schema

```
CREATE TABLE IF NOT EXISTS requirements (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  functional_req TEXT NOT NULL,
  security_req TEXT NOT NULL,
  priority TEXT NOT NULL,
  category TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
```

How to Setup & Run (Quick)

1. Create a virtual environment and activate it:

```bash
python -m venv venv
# Windows
venv\Scripts\activate
# macOS/Linux
source venv/bin/activate
```

2. Install Flask:

```bash
pip install flask
```

3. Run the app:

```bash
python app.py
```

4. Open `http://127.0.0.1:5000/` in your browser.

API / Routes Overview

* `GET /` — Dashboard, search and filter functionality
* `POST /suggest` — Accepts `{functional_req: "..."}` JSON and returns `{'security_req': '...'}` (keyword-based suggestions)
* `GET, POST /add` — Add a new requirement
* `GET, POST /edit/<id>` — Edit existing requirement
* `GET /delete/<id>` — Delete requirement
* `GET /export` — Returns HTML assurance case report as a downloadable file

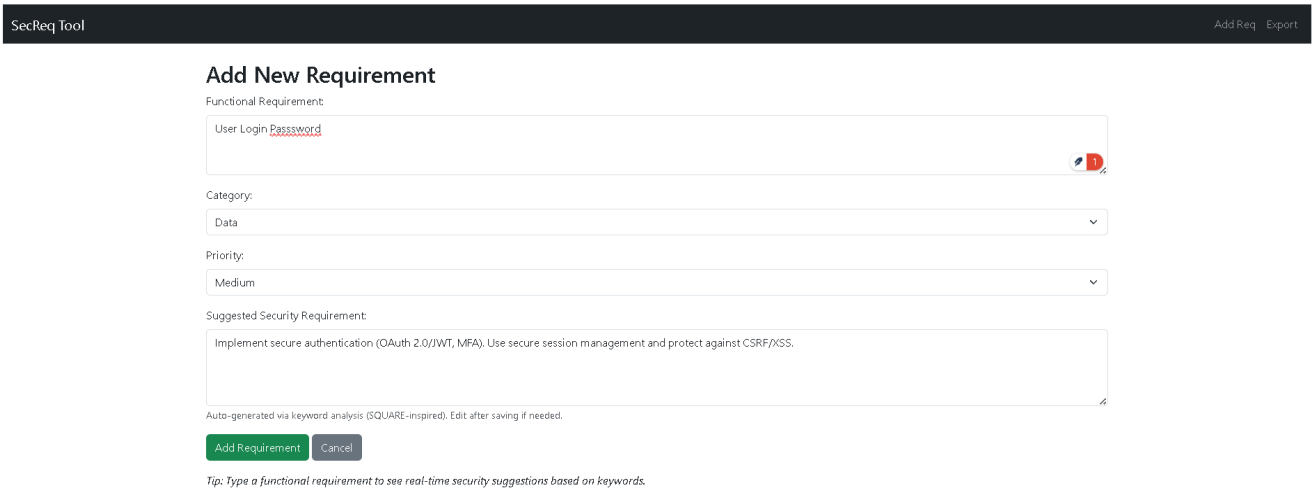
Security & Best-Practices Notes

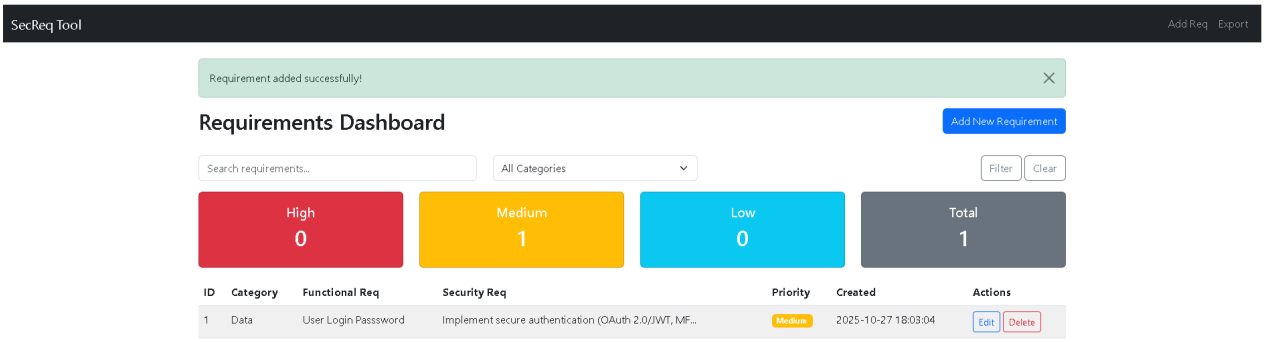
* Use parameterized queries (the app uses `?` placeholders) — avoids SQL injection.
* For production, set a secure `app.secret_key` and disable `debug=True`.
* Use hashed storage for any sensitive data; do not store PII in plaintext.
* Add CSRF protection (`Flask-WTF`) if you plan to expose this app outside controlled environments.
* Sanitize user-provided data shown in exported reports to prevent injection/HTML issues.

Extensions & Improvements

* Integrate simple NLP (spaCy/fastText) to classify functional requirements more accurately.
* Add user accounts and per-project scopes (multi-tenant support).
* Export additional formats (PDF via WeasyPrint or wkhtmltopdf).
* Add unit tests for suggestion logic and CRUD operations.
* Add audit logging for changes (who changed what and when).

``


SecReq Tool
Add Req
Export
Add New Requirement
Functional Requirement:
User Login Passsword
Category:
Data
Priority:
Medium
Suggested Security Requirement:
Implement secure authentication (OAuth 2.0/JWT, MFA). Use secure session management and protect against CSRF/XSS.
Auto-generated via keyword analysis (SQUARE-inspired). Edit after saving if needed.
Add Requirement
Cancel
Tip: Type a functional requirement to see real-time security suggestions based on keywords.


SecReq Tool
Add Req
Export
Requirement added successfully!
Requirements Dashboard
Add New Requirement
Search requirements...
All Categories
Filter
Clear
High
0
Medium
1
Low
0
Total
1
ID
Category
Functional Req
Security Req
Priority
Created
Actions
1
Data
User Login Passsword
Implement secure authentication (OAuth 2.0/JWT, MF...
Medium
2025-10-27 18:03:04
Edit
Delete