

M Square Consulting

Michael De La Cruz

December 10, 2025

Creating a Custom HTML Page with Microsoft Copilot

Initial Request

Create a Python script that does the following:

- 1) Creates a database called c:\Temp\project20251210\dirtyMinds.db
- 2) Creates a questions table.
- 3) Creates an answers table.
- 4) Creates one sample record entry in each table.
- 5) Creates an HTML page that allows you to view the questions and answers with a SQL query.

Response to Initial Request

Here's a **proof-of-concept Python script** that does exactly what you asked:

- Creates a SQLite database at c:\Temp\project20251210\dirtyMinds.db
- Creates questions and answers tables
- Inserts one sample record into each
- Spins up a Flask app that serves an HTML page where you can enter a SQL query and view results.

Python script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds.py

How It Works

1. **Database creation:** Ensures c:\Temp\project20251210\dirtyMinds.db exists.
2. **Tables:** Creates questions and answers tables with a foreign key link.
3. **Sample data:** Inserts one question and one answer if tables are empty.
4. **Web interface:** Flask serves a page with a <textarea> for SQL input. Results are shown in a table.

Supplementary Request 1

Extend this so the HTML page automatically shows all questions with their answers (without needing to type SQL), while still keeping the SQL query box for custom exploration? That would make it more user-friendly.

Response to Supplementary Request 1

Python script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev1.py

What Changed

- **Automatic display:** The page now shows all questions with their linked answers in a table at the top.
- **Custom SQL box:** Still available below, so you can run queries like `SELECT * FROM questions;` or `SELECT * FROM answers;`.
- **Sample data:** One question and one answer are preloaded if the DB is empty.

This way, you get a **user-friendly view** of the Q&A content while still having the flexibility to run arbitrary SQL queries.

Supplementary Request 2

Extend the Flask app so you can **upload a CSV file via an HTML form** and automatically import its contents into the database. This way you don't have to manually run SQL inserts.

Response to Supplementary Request 2

Python script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev2.py

How the CSV Import Works

- Upload a .csv file via the new form.
- Each row should have **two columns**: question_text, answer_text
- The script inserts the question into questions, retrieves its ID, then inserts the answer into answers.

This way, you can bulk-import Q&A pairs directly from a CSV file.

Supplementary Response 3

Extend the HTML/Flask app so you can **delete records directly from the displayed Questions/Answers table**. We'll add a "Delete" button next to each row and wire it up to a Flask route that removes the record from the database.

Response to Supplementary Response 3

Python Script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev3.py

What Changed

- Added a **Delete button** in the Questions/Answers table.
- Each button posts to `/delete/<id>` where `<id>` is the question's primary key.
- The route deletes both the question and its linked answers.
- Added a JavaScript `confirm()` prompt to avoid accidental deletions.

This way, you can manage your dataset directly from the browser — view, query, import CSV, and now delete records.

Supplementary Request 4

Merge the **Import CSV feature back into the current app** while keeping the **Delete buttons** in the display. That way you can view, query, import, and delete records all from the same HTML page.

Response to Supplementary Request 4

Python script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev4.py

Features Now Included

- **View Q&A:** Table shows all questions with answers.
- **Delete Records:** Each row has a Delete button.
- **Import CSV:** Upload a CSV file with question_text,answer_text rows to bulk-import.
- **Custom SQL Query:** Still available for advanced exploration.

Supplementary Request 5

Add the **Export CSV feature** alongside your existing Import and Delete functionality. This way you can back up or share all current questions/answers directly from the browser.

Response to Supplementary Request 5

Python Script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev5.py

Features Now

- **View Q&A:** Table shows all questions with answers.
- **Delete Records:** Each row has a Delete button.
- **Import CSV:** Upload a CSV file (question_text,answer_text) to bulk-import.
- **Export CSV:** Download all current questions/answers as dirtyMinds_export.csv.
- **Custom SQL Query:** Still available for advanced exploration.

This gives you a full CRUD-style workflow: import, view, delete, query, and export.

Supplementary Request 6

Extend the app so you can **edit questions and answers inline**. This will give you full CRUD functionality: view, import, export, delete, and now update records directly from the browser.

Response to Supplementary Request 6

Python script: belliniseven.store/caseOrganizer/project20251210/PY/dirtyMinds-rev6.py

Access Custom HTML Page: <https://127.0.0.1:5000>

Local DB: c:\Temp\project20251210\dirtyMinds.db

Dirty Minds Database Tool

Questions and Answers

ID	Question	Answer	Actions
1	What has keys but can't open locks?	A piano	<button>Delete</button> <button>Edit</button>
3	Question 2	Answer 2	<button>Delete</button> <button>Edit</button>
4	Question 3	Answer 3	<button>Delete</button> <button>Edit</button>
5	Question 4	Answer 4	<button>Delete</button> <button>Edit</button>
6	Question 2	Answer 2	<button>Delete</button> <button>Edit</button>
7	Question 3	Answer 3	<button>Delete</button> <button>Edit</button>
8	Question 4	Answer 4	<button>Delete</button> <button>Edit</button>
9	Question 2	Answer 2	<button>Delete</button> <button>Edit</button>
10	Question 3	Answer 3	<button>Delete</button> <button>Edit</button>

Import CSV

Choose File No file chosen

Upload & Import

Export CSV

Download CSV

Custom SQL Query

Run SQL