Project Template: Database

In this guide we will:

- Connect to a database
- Initialise information in the database
- Display data from the database
- Save form input into the database

Before we start

The current project is split into 5 different types of files.

- Python files (flaskblog.py, forms.py)
    - Creates the instance of flask.
    - Stores the data to be displayed.
    - Performs routing.
    - Creates form functionality.
    - **Saves form input to the database**
- HTML files (blog.html, home.html, layout.html and register.html)
    - Handles the webpage's structure and contents.
    - Creates form elements.
- CSS files (main.css)
    - Handles the webpage's styling.
- SQL files (database.sql)
    - **Used for writing SQL queries**
- Database file (blog.db)
    - **Stores the contents of database (this is a system file, you should not try to edit it)**

Run the code in '**flaskblog.py'** and then make sure you can view both the web page and the code.

Step 1: Connect to a database

In this example we will be using SQLite. The key line to connect to the SQLite database, **blog.db**, is the following code in **flasblog.py:**

```
conn = sqlite3.connect('blog.db')
```

- The variable, **conn** (short for connection), can now be used when referring to the database.

Step 2: Initialise information in the database

Next let's look at the code in **database.sql**. Writing SQL queries is a concept that will be explored later in the course, so for now, focus on what the code is doing rather than how it's doing it.

Key Functionality:

- If tables with the name 'users' and 'blogs' already exist, delete them.
- Create tables 'users' and 'blogs'.
- Add users and blogs into the created tables.

Exercises

1. For each key piece of functionality, identify the relevant code in **database.sql.**
2. Consider why we first delete any tables with the names 'users' and 'blogs'?
3. Use the following link to learn about different variable types e.g. varchar:

   https://www.w3schools.com/sql/sql_datatypes.asp

   https://www.w3schools.com/sql/sql_autoincrement.asp

Step 3: Display data from the database

When you view the **home** page, you should see several posts are displayed. The information for each of these posts is stored in the database.

The code that display these posts is in the function, **home()**, in **flaskblog.py**.

```
    conn = sqlite3.connect('blog.db')

    conn.row_factory = dict_factory
    c = conn.cursor()

    c.execute("SELECT * FROM blogs")
    posts = c.fetchall()
```

Let's deconstruct the steps for accessing the database and displaying the blogs:

1. Connect to the database.
2. Change the database results into a dictionary format rather than a list / array.
3. Create a cursor object.
    - A cursor is like a temporary copy of the database where data can be retrieved or manipulated.
4. Execute a query to select all blogs.
5. Set a variable, **posts,** to be equal to all the selected blogs in a dictionary format.
    - In the **home.html**, **posts** will refer to this variable.

Exercises

1. Open the **Add Blog** page in your browser and identify the Username field dropdown.

   Now look at the function **blog() in** flaskblog.py and identify the code which displays this username field.

   Compare this code to the code for displaying the blogs. What are the similarities and differences?

Step 4: Save form input into the database

To save data from a form input into the database we need to adjust the form code in **flaskblog.py.**

The following code snippet is in the function, **register().**

```
if form.validate_on_submit():
    conn = sqlite3.connect('blog.db')
    c = conn.cursor()

    query = 'insert into users VALUES (' + "'" + form.username.data +
    "'," + "'" + form.email.data + "'," + "'" + form.password.data +
    "'" + ')'
    c.execute(query)
    conn.commit()
```

Let's deconstruct the steps for saving a new user.

1. Connect to the database.
2. Create a cursor object.
3. Create an SQL query (in the form of a string) to create a new entry in the **users** table. Notice that within the query are the form fields.
4. Run the query and commit the changes.

Exercises

1. Compare this code to the code for saving a new blog (in **blog()**) . What are the similarities and differences?