Full Stack Development with AI
Lab 10.3 – Creating a Complete Web Application with Flask

Lab Overview

In this lab, you will learn how to create a server-side rendering web application using Flask. This web application is similar to the one that you have created in Lab 10.2.

You will be using the Jinja2 template engine and persist product data into an SQLite database.

You will be using the same synthetic product data as Lab 10.2.

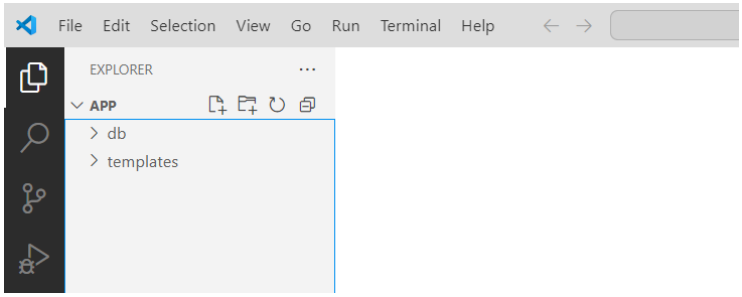
Exercise 1 – Creating a new Flask Application

1. Use the file explorer or terminal on your computer to create a new folder `app`.

2. Start Visual Studio Code (VS Code) and open the newly created `app` folder.

3. Create a new folder `db`. Repeat the same steps to create another new folder `templates`.

4. Your Explorer should now resemble the figure below:



5. Start a terminal in VS Code.

6. Run the following commands to install the required Python libraries using `pip`:

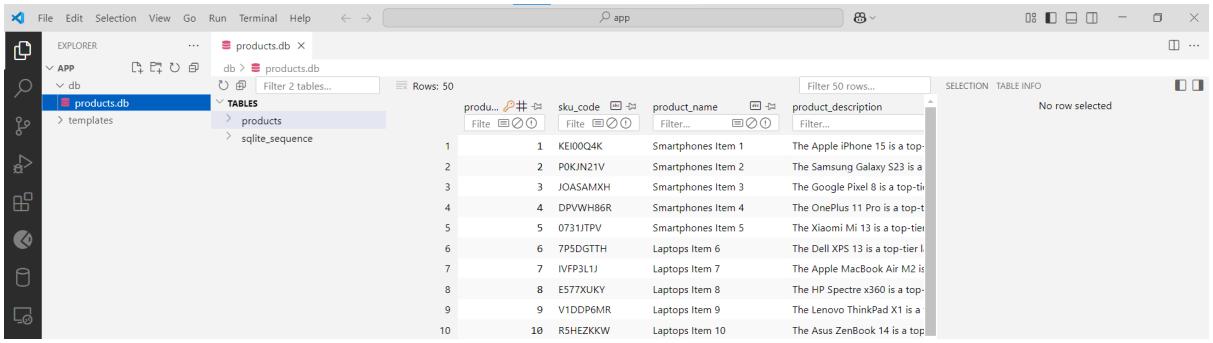
```
python -m pip install flask
```

Exercise 2 – Cloning the SQLite Database

For this lab, we will be reusing the "products.db" SQLite database that you have created in Lab 10.2.

Copy the file "products.db" into the `app\db` folder of your new Flask application.

Return to VS Code and your Explorer should now resemble the figure below:



Exercise 3 – Creating New Endpoints

In this exercise, you will be creating a series of endpoints using the GET and POST HTTP methods to implement the create, retrieve, update and delete use cases for product records.

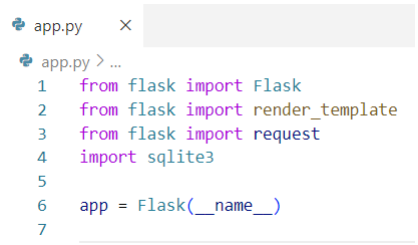
1. Right click on the Explorer in the area outside of any subfolder and select "New File". This will create the new file in the root of the web application or the `app` folder itself. Name the file as "app.py"

2. Paste the following code fragment representing the basic structure of a Flask application into "app.py"

```
from flask import Flask
from flask import render_template
from flask import request
import sqlite3

app = Flask(__name__)
```

3. Your VS Code should resemble the following figure:



4. After line 6, add a `GET` endpoint to the root path `/` that renders the Jinja2 template "index.html" as response.

5. Add a `GET` endpoint to the path `/viewAllProducts` that performs the following two tasks:

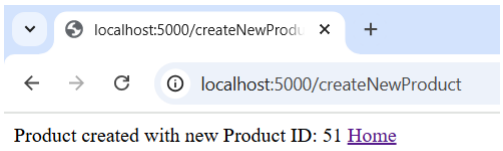
    a. Retrieves all the product records from the `products` table using a `SELECT` query.
    b. Renders the Jinja2 template "viewAllProducts.html" as response

6. Add a `GET` endpoint to the path `/createNewProduct` that renders the Jinja2 template "createNewProduct.html" as response.

    Note that this template will render a blank HTML form for user to input the data for a new product record.

7. Add a `POST` endpoint to the path `/createNewProduct` that performs the following three tasks. Note that in Flask, the same `@app.route` decorator of a Python handler function can be used to add multiple HTTP methods to a particular path.

    a. Retrieves the form data in the Jinja2 template "createNewProduct.html".
    b. Insert the data for the new product record into the `products` table using an `INSERT` statement.
    c. Return an information message together with a link back to the homepage.



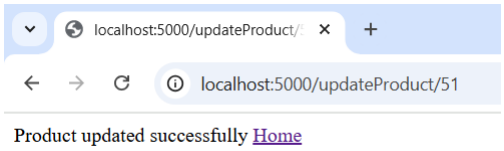
8. Add a `GET` endpoint to the path `/updateProduct` that performs the following two tasks:

    a. Accepts a path parameter for the product id of the product to be updated
    b. Renders the Jinja2 template "updateProduct.html" as response.

    Note that this template will render a HTML form prepopulated with the existing data of the selected product for user to edit.

9. Add a `POST` endpoint to the path `/updateProduct` that performs the following four tasks. Again, note that in Flask, the same `@app.route` decorator of a Python handler function can be used to add multiple HTTP methods to a particular path.

    a. Accepts a path parameter for the product id of the product to be updated
    b. Retrieves the form data in the Jinja2 template "updateNewProduct.html".
    c. Update the data for the existing product record into the `products` table using an `UPDATE` statement.
    d. Return an information message together with a link back to the homepage.



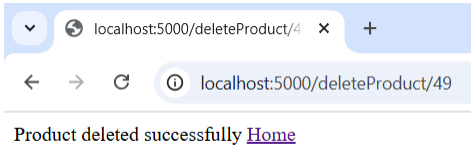
10. Add a `GET` endpoint to the path `/deleteProduct` that performs the following two tasks:

    a. Accepts a path parameter for the product id of the product to be deleted
    b. Renders the HTML template "deleteProduct.html" as response.

    Note that this template will render a table showing the existing data of the selected product for user to confirm the deletion.

11. Add a `POST` endpoint to the path `/deleteProduct` that performs the following three tasks. Once again, do note that in Flask, the same `@app.route` decorator of a Python handler function can be used to add multiple HTTP methods to a particular path.

    a. Accepts a path parameter for the product id of the product to be deleted
    b. Delete the existing product record from the `products` table using a `DELETE` statement.
    c. Return an information message together with a link back to the homepage.



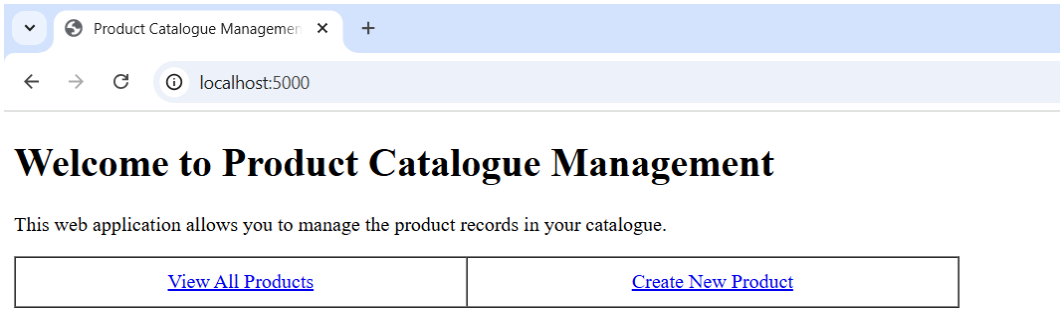
Exercise 4 – Creating the Views as Jinja2 Templates

In this exercise, you will be creating a series of Jinja2 templates representing the views of the web application. Recall that Jinja2 uses HTML directly.

Index or Homepage of the Web Application

1. Create a new file "index.html" in the `templates` folder.

2. Input the HTML code to create a web page resembling the figure below:



The two links should link to the following paths:

- View All Products – /viewAllProducts
- Create New Product – /createNewProduct

View All Products

1. Create a new file "viewAllProducts.html" in the `templates` folder.

2. Input the HTML code to create a web page resembling the figure below:

Product ID	SKU Code	Product Name	Product Description	Brand	Model	Category	Unit Price	Action
1	KEI00Q4K	Smartphones Item 1	The Apple iPhone 15 is a top-tier smartphones designed for premium performance and reliability.	Apple	iPhone 15	Smartphones	$34	Update | Delete
2	P0KJN21V	Smartphones Item 2	The Samsung Galaxy S23 is a top-tier smartphones designed for premium performance and reliability.	Samsung	Galaxy S23	Smartphones	$80	Update | Delete
3	JOASAMXH	Smartphones Item 3	The Google Pixel 8 is a top-tier smartphones designed for premium performance and reliability.	Google	Pixel 8	Smartphones	$40	Update | Delete
4	DPVWH86R	Smartphones Item 4	The OnePlus 11 Pro is a top-tier smartphones designed for premium performance and reliability.	OnePlus	11 Pro	Smartphones	$78	Update | Delete
5	0731JTPV	Smartphones Item 5	The Xiaomi Mi 13 is a top-tier smartphones designed for premium performance and reliability.	Xiaomi	Mi 13	Smartphones	$43	Update | Delete
6	7P5DGTTH	Laptops Item 6	The Dell XPS 13 is a top-tier laptops designed for premium performance and reliability.	Dell	XPS 13	Laptops	$97	Update | Delete
7	IVFP3L1J	Laptops Item 7	The Apple MacBook Air M2 is a top-tier laptops designed for premium performance and reliability.	Apple	MacBook Air M2	Laptops	$30	Update | Delete
8	E577XUKY	Laptops Item 8	The HP Spectre x360 is a top-tier laptops designed for premium performance and reliability.	HP	Spectre x360	Laptops	$69	Update | Delete
9	V1DDP6MR	Laptops Item 9	The Lenovo ThinkPad X1 is a top-tier laptops designed for premium performance and reliability.	Lenovo	ThinkPad X1	Laptops	$62	Update | Delete
10	R5HEZKKW	Laptops Item 10	The Asus ZenBook 14 is a top-tier laptops designed for premium performance and reliability.	Asus	ZenBook 14	Laptops	$41	Update | Delete

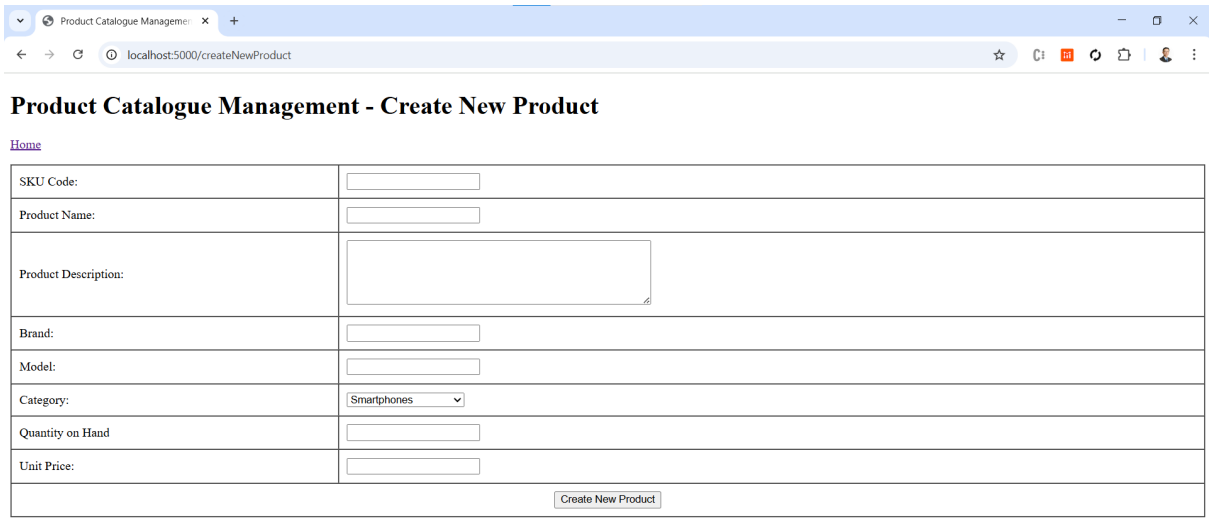
The two links should link to the following paths:

- Update – /updateProduct/<product_id>
- Delete – /deleteProduct/<product_id>

Create New Product

1. Create a new Jinja2 template file “createNewProduct.html”

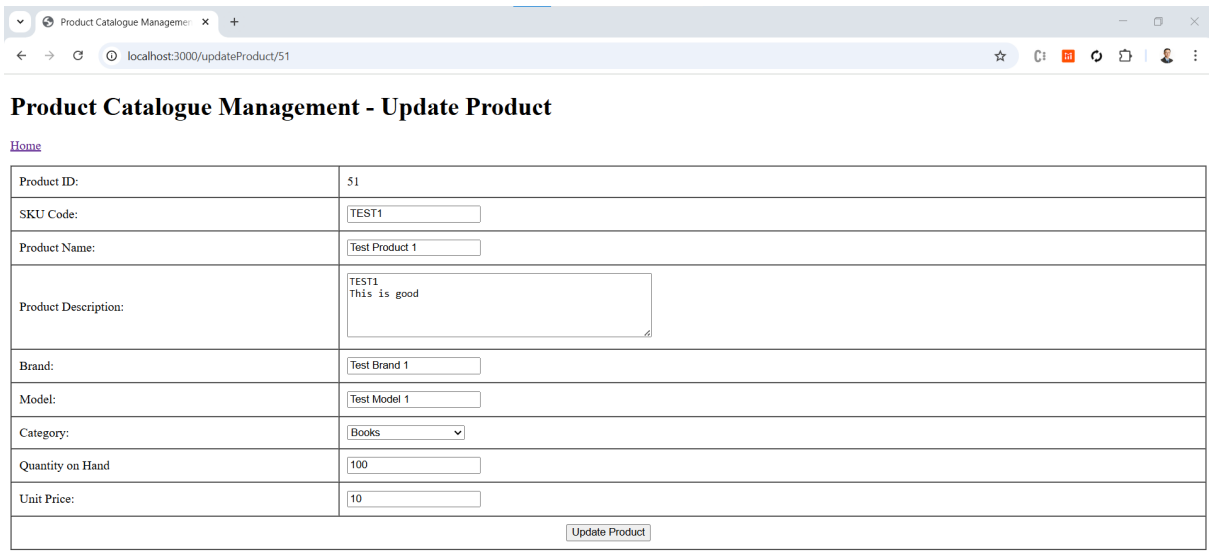
2. Input the HTML code to create a web page resembling the figure below:



Update Product

1. Create a new Jinja2 template file “updateProduct.html”

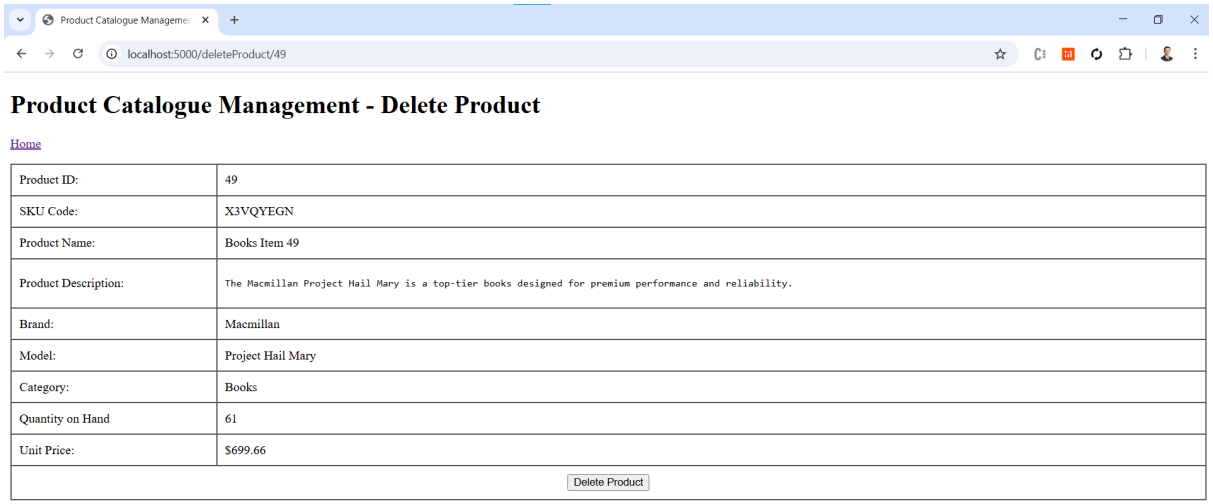
2. Input the HTML code to create a web page resembling the figure below:



Delete Product

1. Create a new Jinja2 template file “deleteProduct.html”

2. Input the HTML code to create a web page resembling the figure below:



Exercise 5 – Running and Testing the Web Application

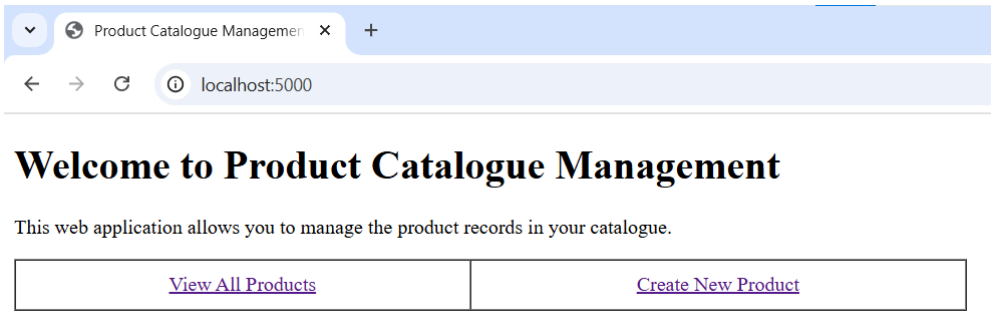
Follow the instructions below to run the web application:

1. Start a new terminal in VS Code.

2. Run the following command to run the web application:

```
python -m flask --app app run
```

3. Open your web browser and navigate to the address http://localhost:5000/. Ensure that you are able to view the index or homepage.



4. Click on the "View All Products" link to view the product records and perform some update and delete operations.

5. Click on the "Create New Product" link to add a new product record. Then return to the view all products webpage and ensure that you can view the new product record.

-- End of Lab --