

Sweetleaf & Co. — Part 4 Execution Proof
CS 334 Web Application Project
Zoe Martinez

Links:

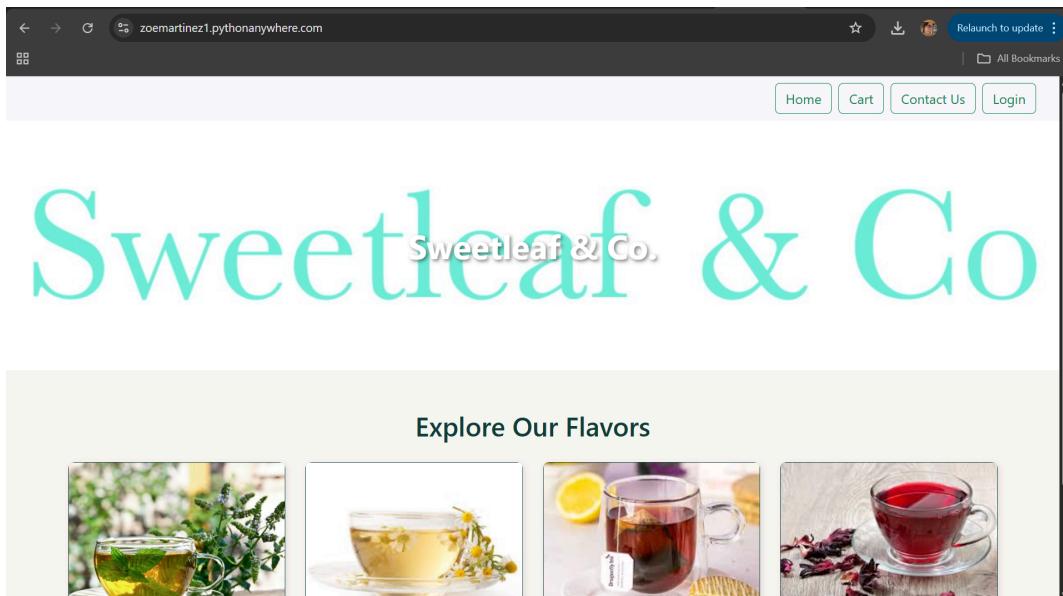
Github Page: <https://zoemtz.github.io/sweetleaf-co/>

Github Repository: <https://github.com/ZoeMtz/sweetleaf-co/tree/main/part4>

Pythonanywhere: <https://zoemartinez1.pythonanywhere.com>

Public API Endpoint: <https://zoemartinez1.pythonanywhere.com/api/products>

Index part 1:



zoemartinez1.pythonanywhere.com

Sweetleaf & Co.

Explore Our Flavors

Mint Meadow

Calm Chamomile

Sir Earl Grey

Radiant Hibiscus

Mint Meador Page:

zoemartinez1.pythonanywhere.com/product_mint

[Back to Home](#)

Mint Meadow

A refreshing mint blend that cools and calms with every sip.

Price: \$5.99

Add to Cart

Calm Chamomile Page:

[← Back to Home](#)



Calm Chamomile

Gentle chamomile blossoms brewed to help you unwind and relax.

Price: \$4.99

[Add to Cart](#)

Sir Earl Grey Page:

[← Back to Home](#)



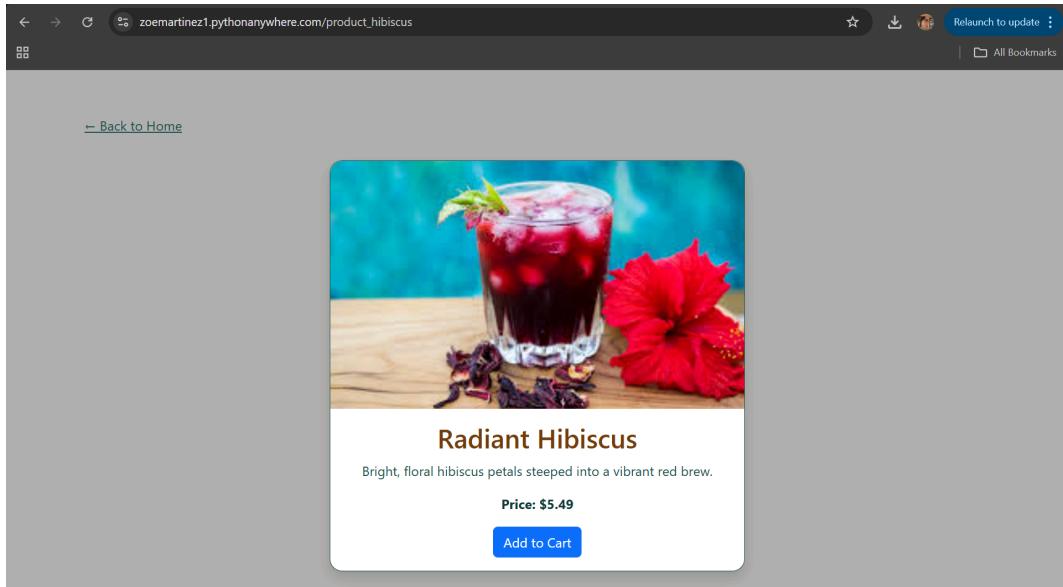
Sir Earl Grey

A bold, classic black tea with citrusy notes and timeless flavor.

Price: \$6.49

[Add to Cart](#)

Radiant Hibiscus Page:



Cart:

A screenshot of a web browser displaying a cart summary page. The URL in the address bar is `zoemartinez1.pythonanywhere.com/cart`. The page title is "Your Cart". It shows three items in the cart: "Mint Meadow" (Quantity 1, Price \$5.99), "Mint Meadow" (Quantity 1, Price \$5.99), and "Sir Earl Grey" (Quantity 1, Price \$6.49). The total amount is "Total: \$18.47". At the bottom, there are two buttons: "Proceed to Checkout" and "Clear Cart".

Checkout:

Full Name

Email

Shipping Address

Payment Method

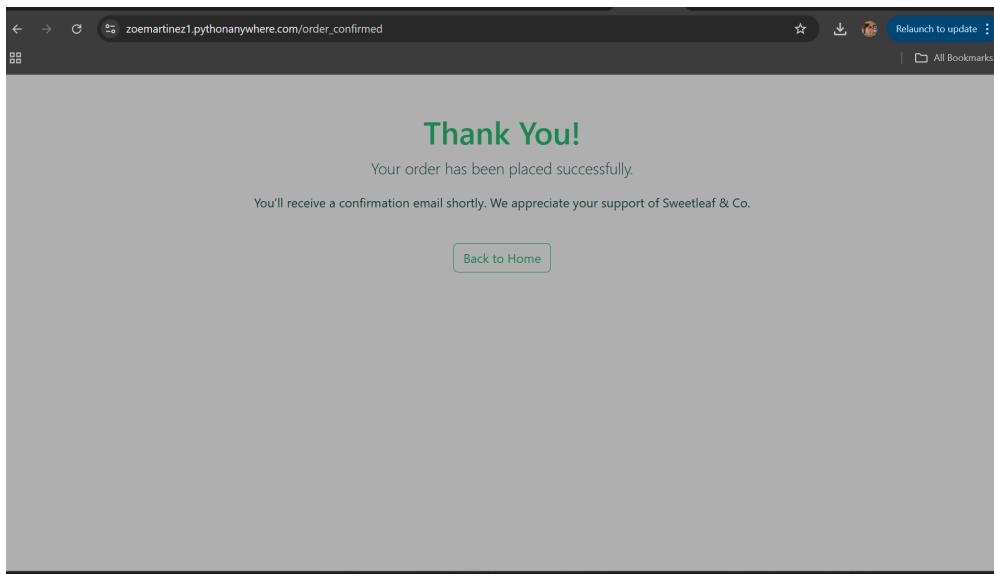
Select

Total

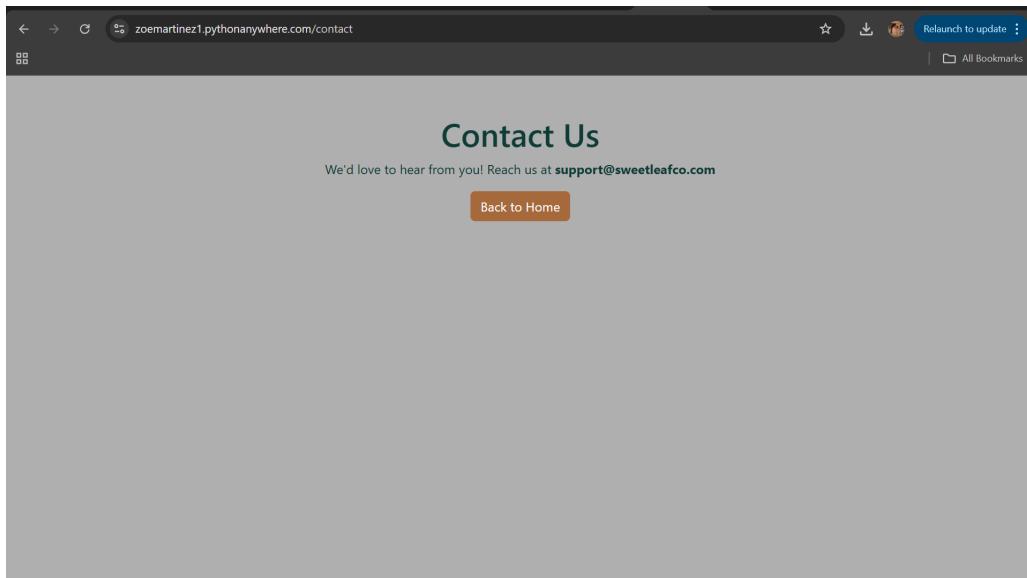
12.97

Place Order

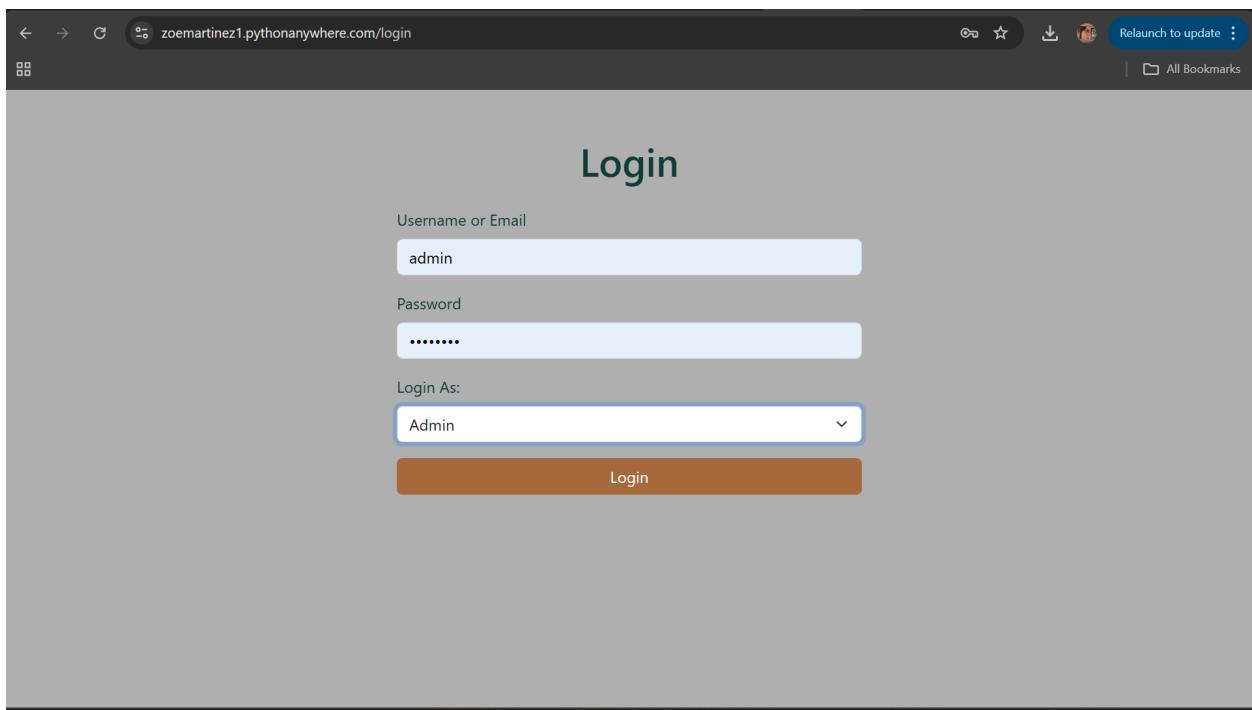
Checkout Confirmation:



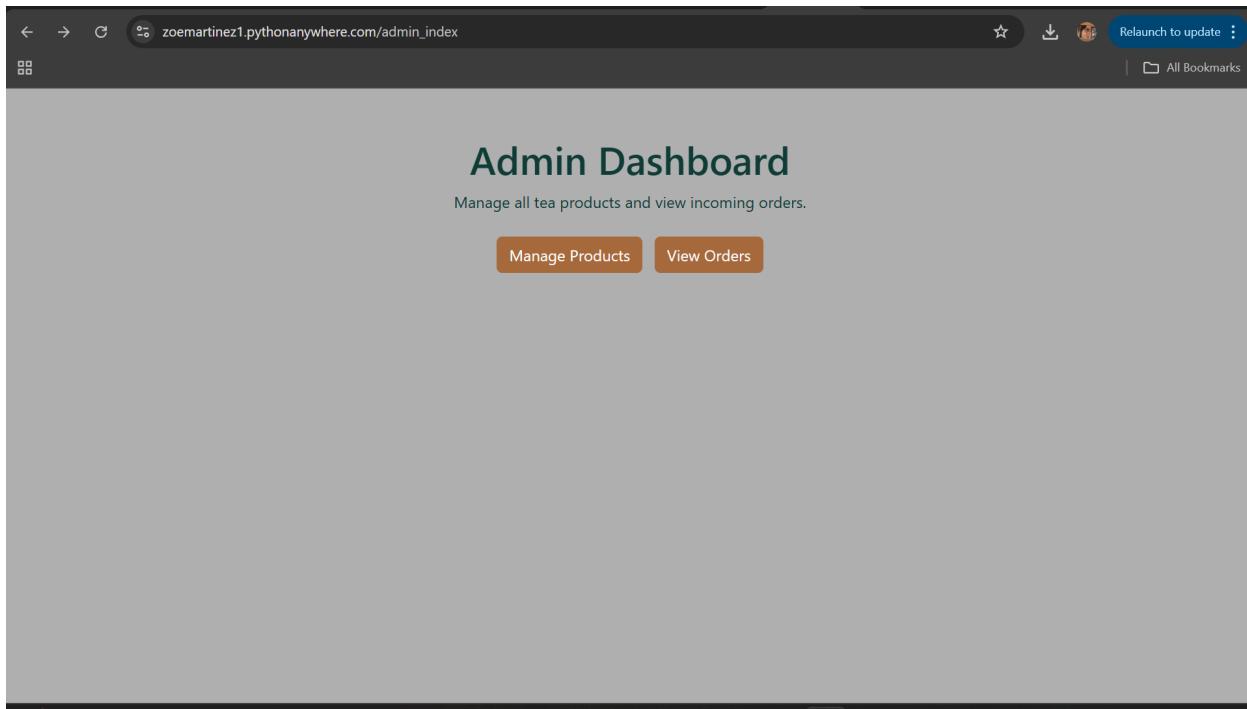
Contact:



Log in:



Admin Dashboard:



Admin Manage Products:

A screenshot of a web browser window titled "Manage Tea Products". The URL in the address bar is "zoemartinez1.pythonanywhere.com/admin_product". The page content includes the title "Manage Tea Products", a subtitle "Add, edit, or remove teas from your shop.", and a table with four columns: Name, Price, Stock, and Actions. The table has one row with the data: Mint Meadow, \$5.99, 20, and two buttons: "Edit" and "Delete".

Admin Manage Customer Orders:

Customer Orders

View recent orders placed by customers.

Order #	Name	Total	Status
#1	Jane Doe	\$11.98	Pending
#2	Carlos Rivera	\$8.49	Pending
#3	Nina Lopez	\$14.99	Pending
#4	Chloe Connors	\$12.97	Pending
#5	Chloe Connors	\$12.97	Pending
#6	Chloe K	\$12.97	Pending
#7	Example	\$12.97	Pending
#8	Example	\$12.97	Pending
#9	Ex Ex	\$12.97	Pending
#10	Example	\$12.97	Pending
#11	Zoe M	\$12.97	Pending

API Products, Screenshot of /api/products JSON response on PythonAnywhere:

```
[{"id":1,"name":"Mint Meador","price":3.99},{"id":2,"name":"Calm Chamomile","price":4.49},{"id":3,"name":"Sir Earl Grey","price":4.25},{"id":4,"name":"Radiant Hibiscus","price":4.75}]
```

Working manifest.json loaded successfully in Chrome DevTools:

The screenshot shows the Chrome DevTools Application tab with the 'Manifest' section selected. The manifest file content is displayed as follows:

```
manifest.json
```

Errors and warnings

- Richer PWA Install UI won't be available on desktop. Please add at least one screenshot with the form_factor set to wide.
- Richer PWA Install UI won't be available on mobile. Please add at least one screenshot for which form_factor is not set or set to a value other than wide.

Identity

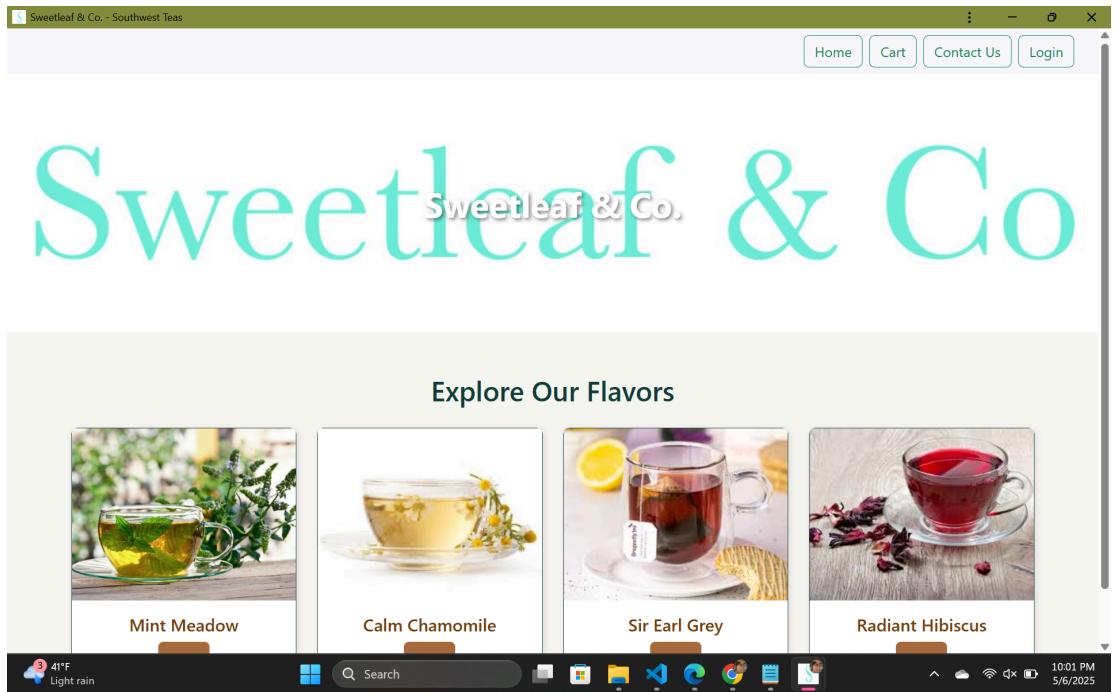
Name: Sweetleaf & Co.
Short name: Sweetleaf
Description:
Computed App ID: https://zoemartinez1.pythonanywhere.com/ [Learn more](#)

Note: id is not specified in the manifest, start_url is used instead. To specify an App ID that matches the current identity, set the id field to /.

Presentation

Start URL: / [Learn more](#)
Theme color: #838e3c

Installed prompt or app window:



Flask running in terminal:

```
PS C:\Users\zoe_M\Downloads\mysite> python flask_app.py
Sample products added.
Sample orders added.
* Serving Flask app 'flask_app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Products already exist.
Orders already exist.
* Debugger is active!
* Debugger PIN: 121-972-311
```