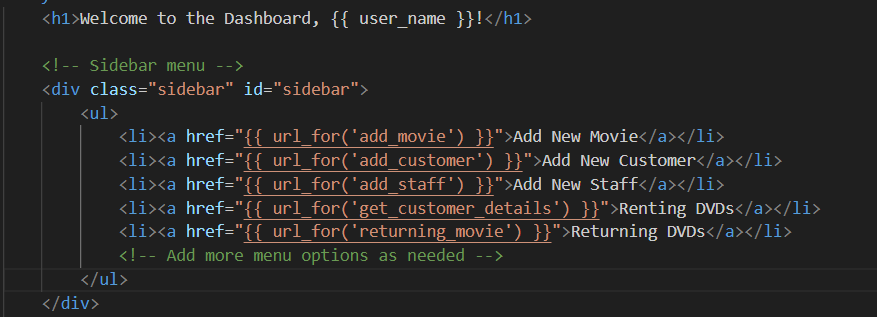
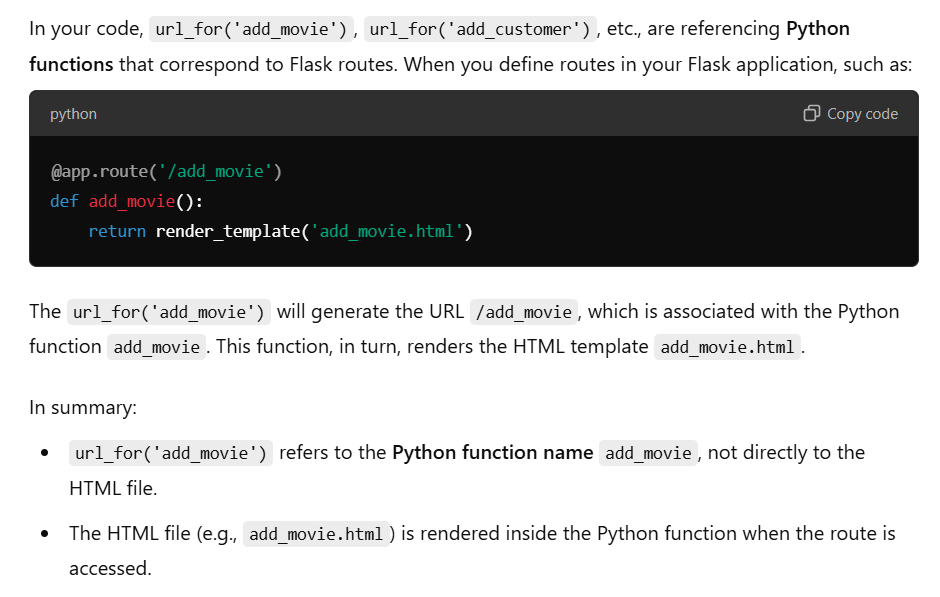
**Notes for Learning**

**Accessing the side bar:**

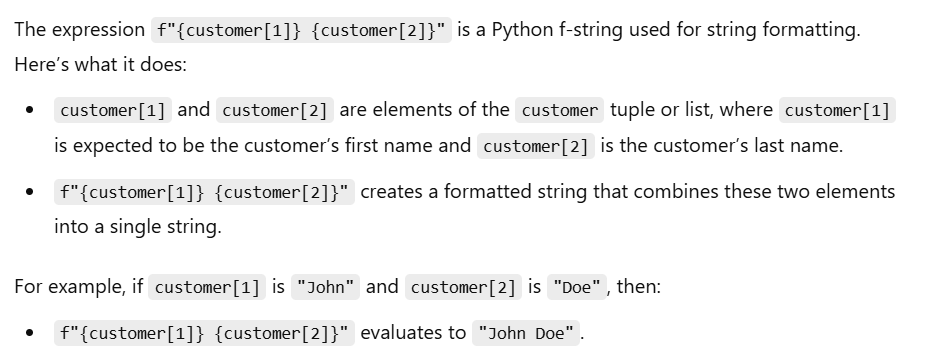
****

The url\_for(‘add\_movie’) actually is used to generate URLs for Python functions that handle the routes. (Not directly for the HTML files)



The Python Function Gathers the data and loads the template (Render Template)

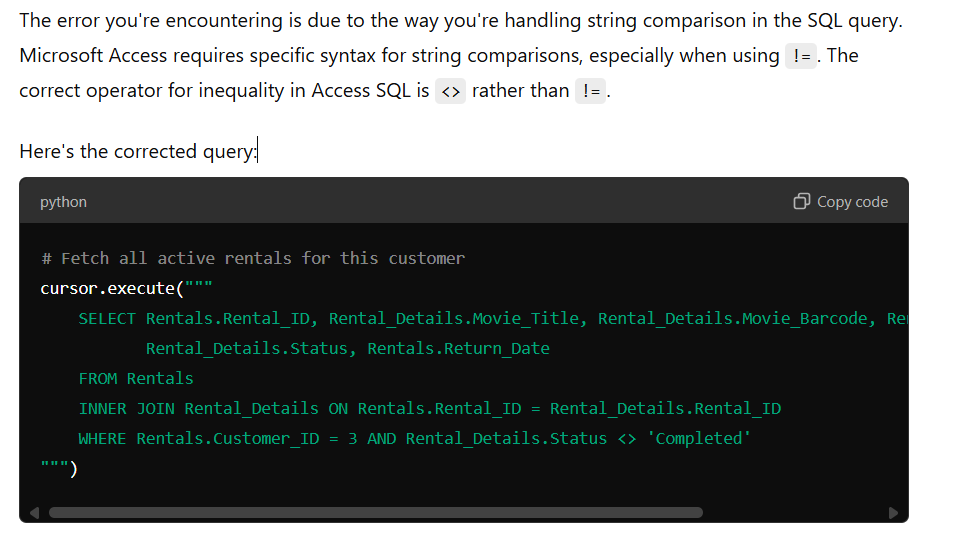
**Combining 2 elements into 1 string**



**Error when Processing code:**

* Ensure that the correct variable name is being used at all times
* Put print statement at every chunk of code when you expect an output, thus you can check if the error actually happens.

**NOT Equals to Operator in Access SQL**

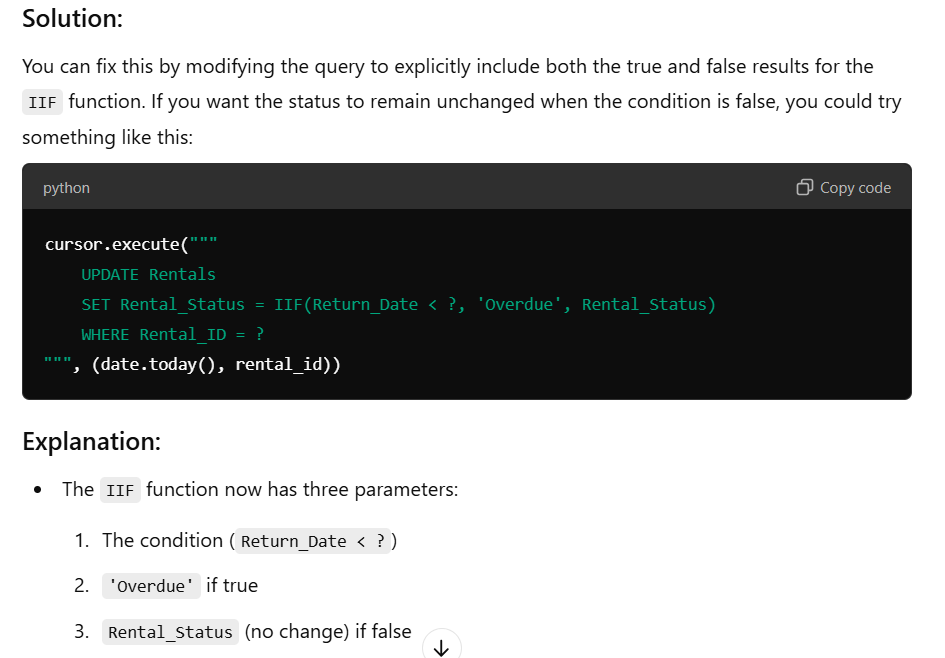


Access SQL uses “<>” for NOT EQUALS rather than “!=”

**Date Comparions**

Later Dates are greater than earlier dates (2024 is greater than 2023)

**CASE VS IIF**



In Microsoft access SQL “IIF” is being used instead of CASE.

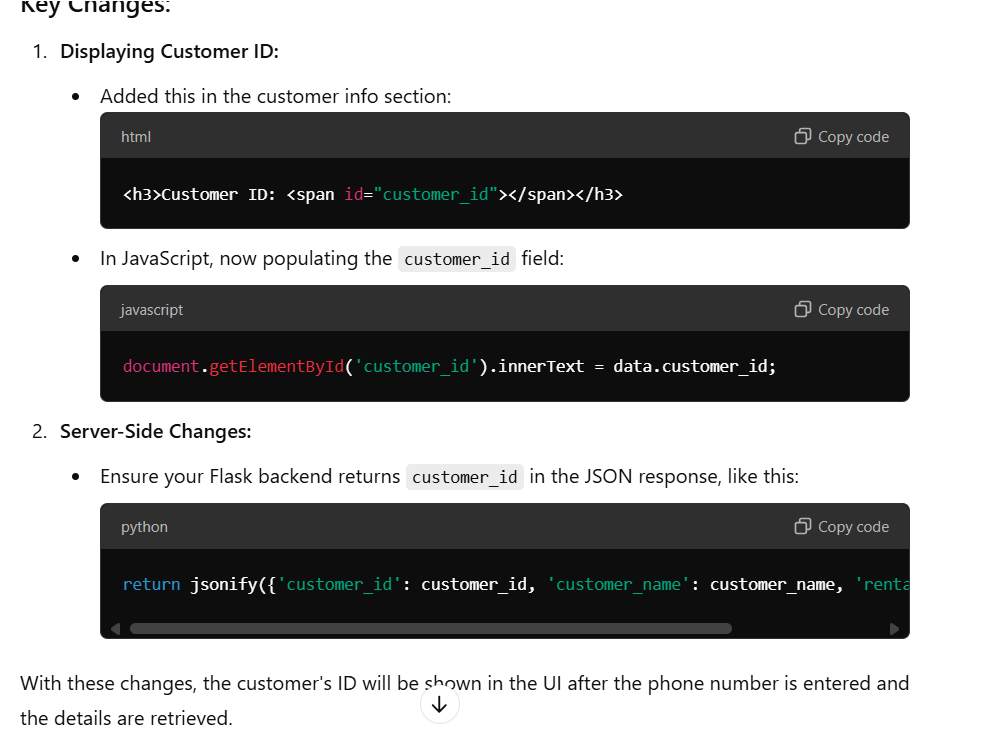
**Too Few Parameters error**



Things to look into:

* Whether the columns names are correct (align with the database)
* Whether all the placeholders (“?”) have a corresponding value being assigned. (make sure all placeholders are assigned a value)

**Displaying new things on a UI**

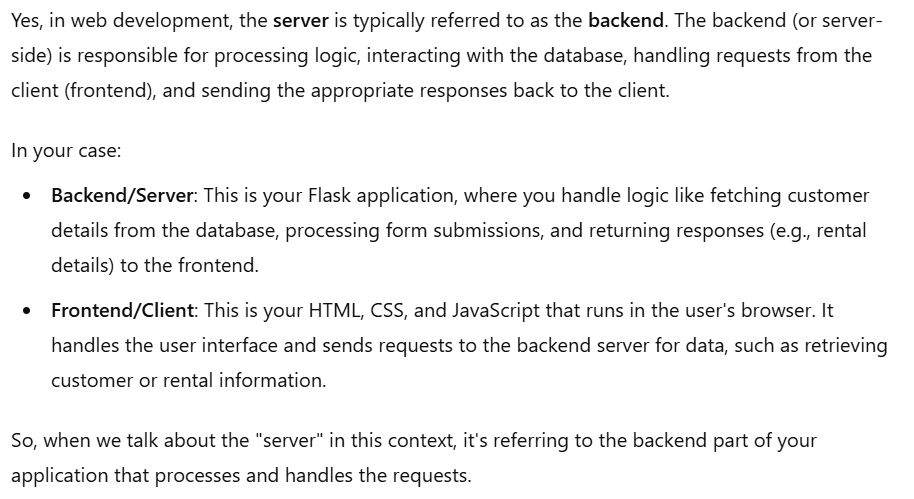


**Html Section**: Make sure that you create a placeholder id = “customer\_id” and the bone of the UI.

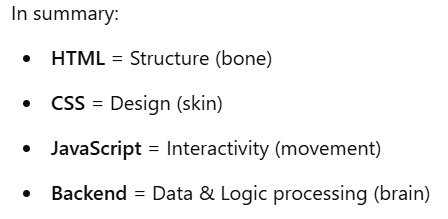
“Customer ID” will be shown on the HTML but the actual will not be shown.

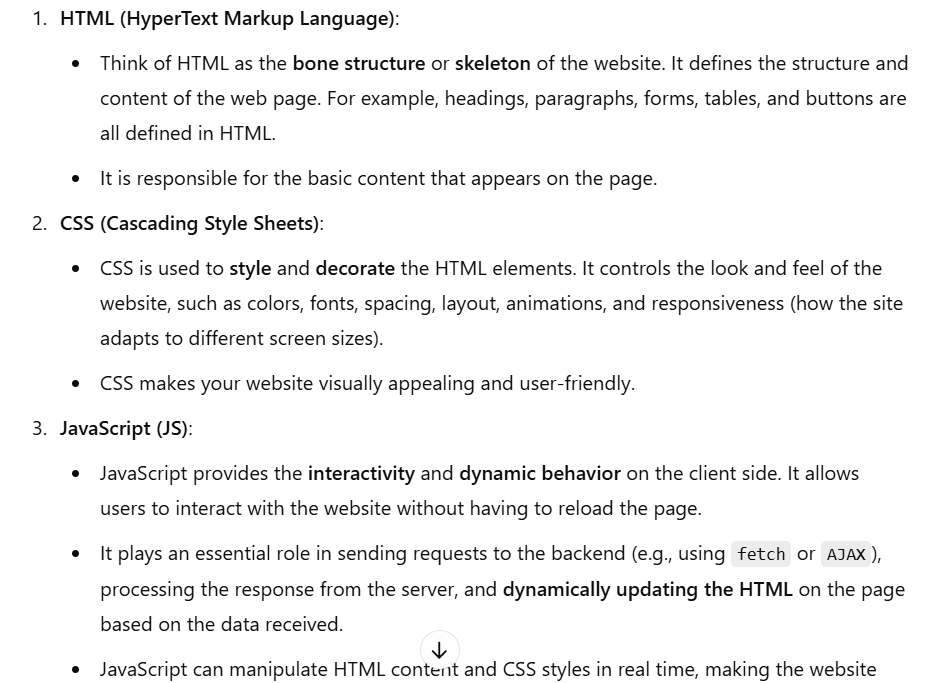
**Javascript Section:** Create a Request to the Server for a specific thing (asking for the customer name or customer ID)

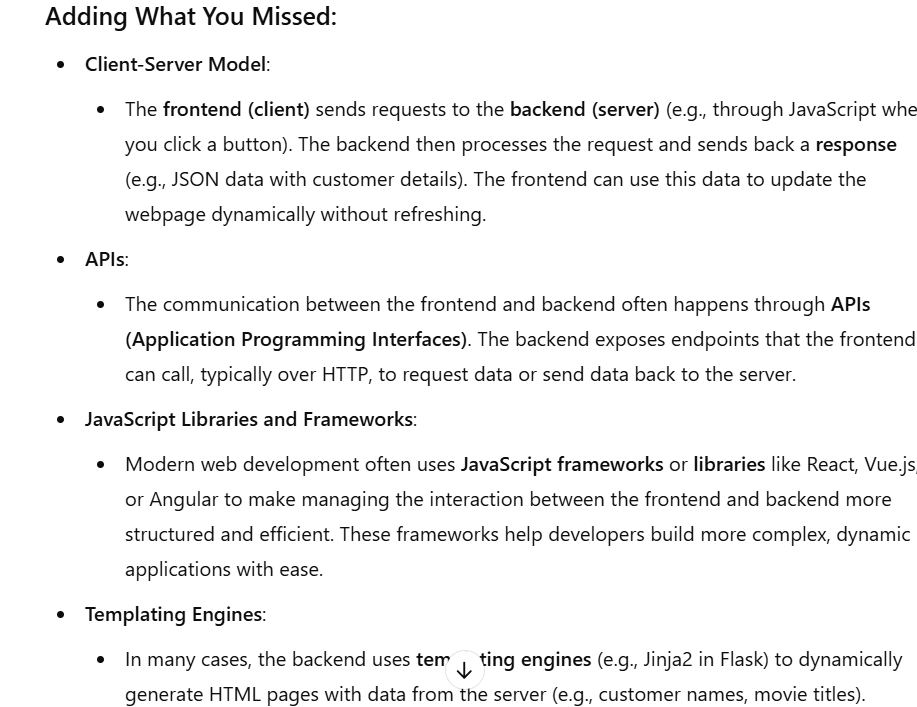
**Backend Section (Server):** Receive request and return values being requested from the client (communicating through Javascript)



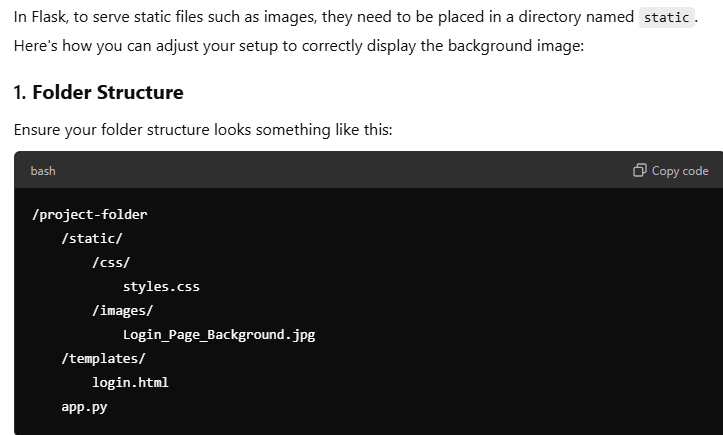
**A quick summary of HTML, CSS and Javascript**

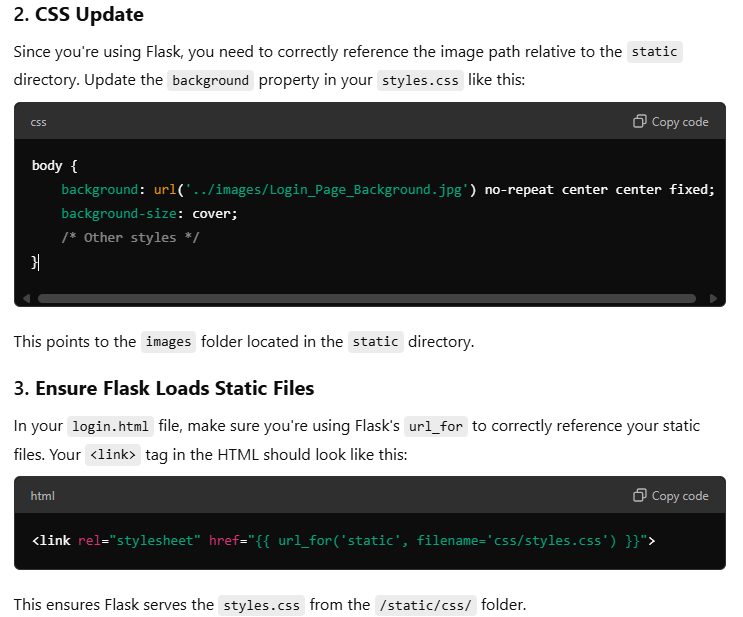






**Flask Folder Structure**

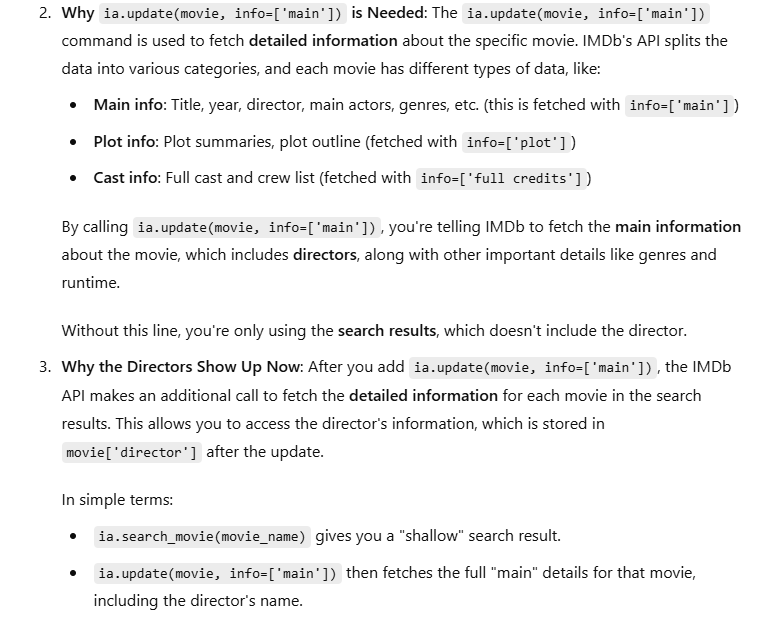
****

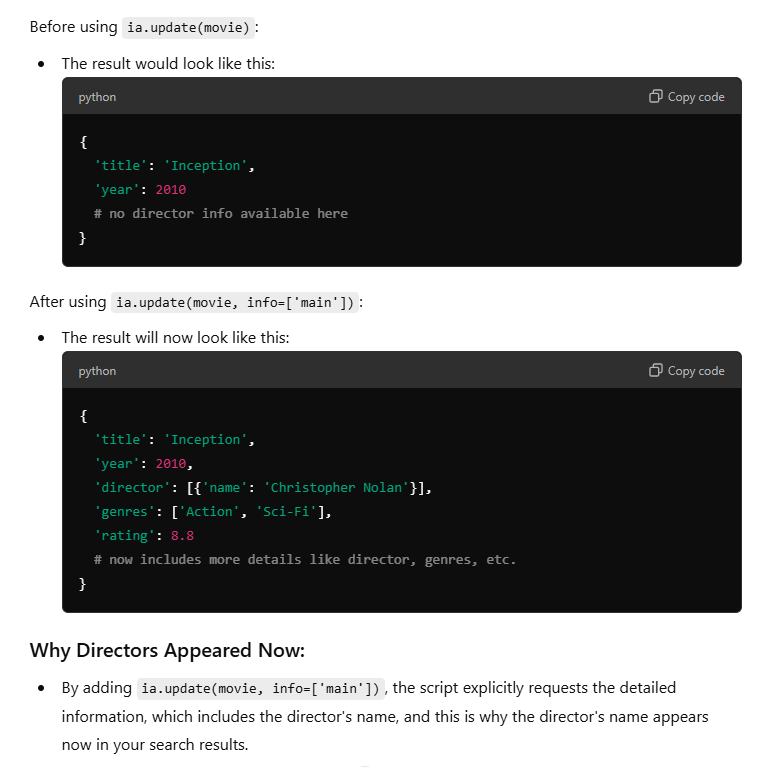


If the styles.css file does not seem to be integrated into the html file. Do the following:

1. Check if the correct name and path is being referenced
2. Check for the file structure

**What does info=[‘main’] means?**





The IMDB API splits the data already into different categories. (essentially I am extracting the information I need from the category that has the data I need)

**How the Flask routes and the html files work together**

/add\_staff is an example of the flask route

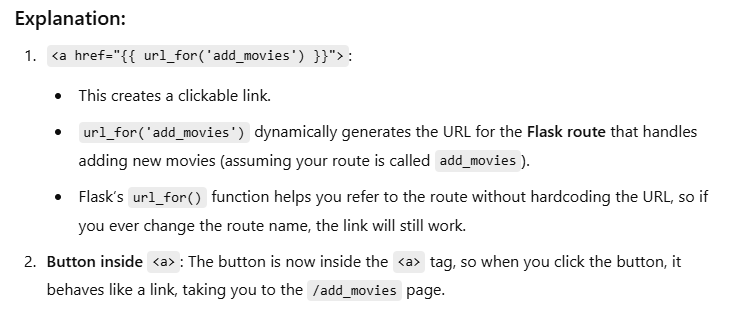
Behind each interactive button on the dashboard. (It triggers the flask route to do something)

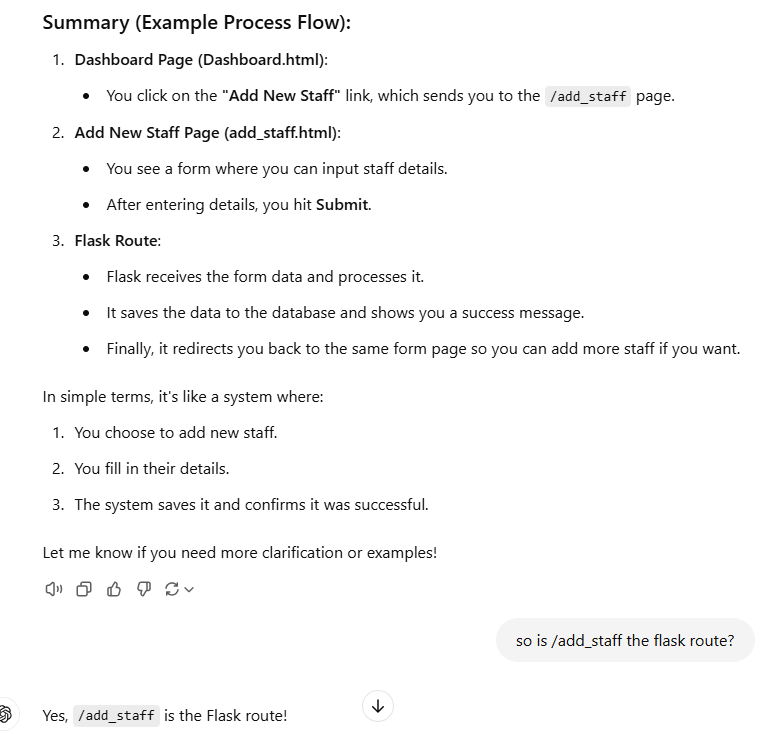
In this case, pressing the Add New Staff Button will trigger/send request to the /add staff route to load the add\_staff.html.

Once the form (add\_staff.html) is being loaded and submit button has been pressed.

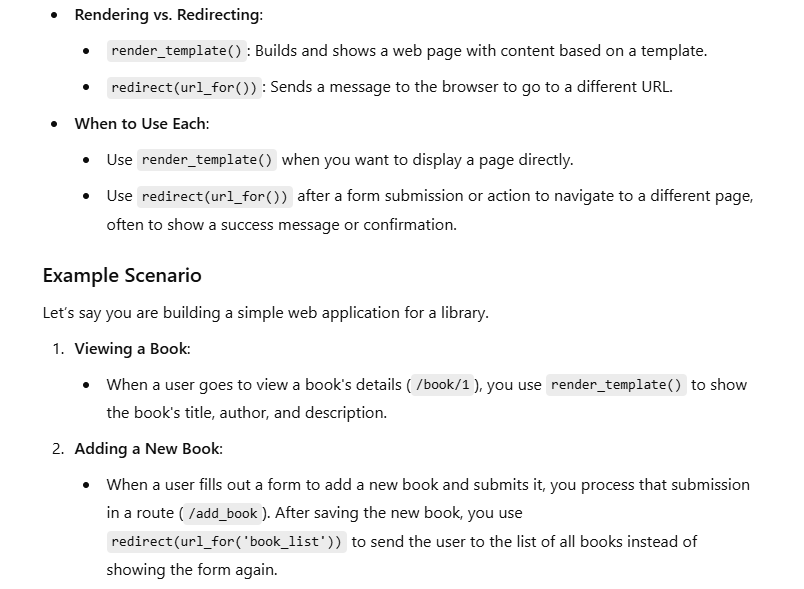
It POSTS the data back to the flask route /add\_staff and it starts extracting the information being inputted into the form and added it into the database.

The following is an example of how the button works:





**Rendering Vs Redirecting**

****

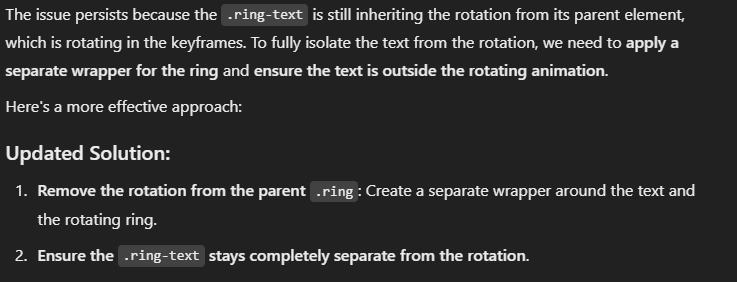
**Template Inheritance**

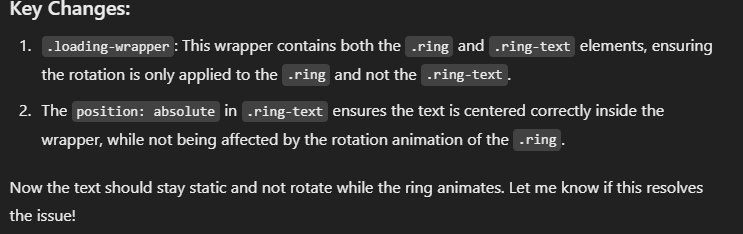
**Purpose: No need to write duplicated code, if many pages requires the same feature of exist, for example: Sidebar and etc. We can just create a sidebar.html and**



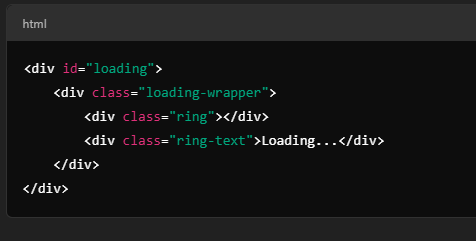
Call the sidebar.html under the body section of the html code.

**How to ensure the loading Ring and the text within it have separate animation**



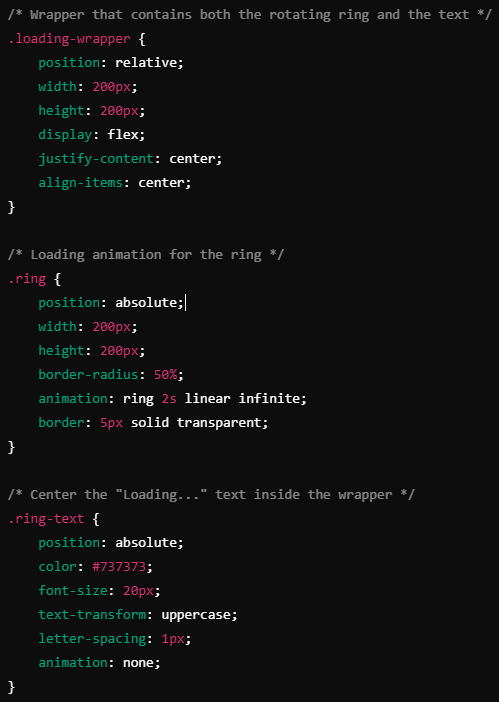


**HTML Code:**



HTML Creates the Structure

CSS Deals with styling and its appearance.



Inside a loading wrapper