

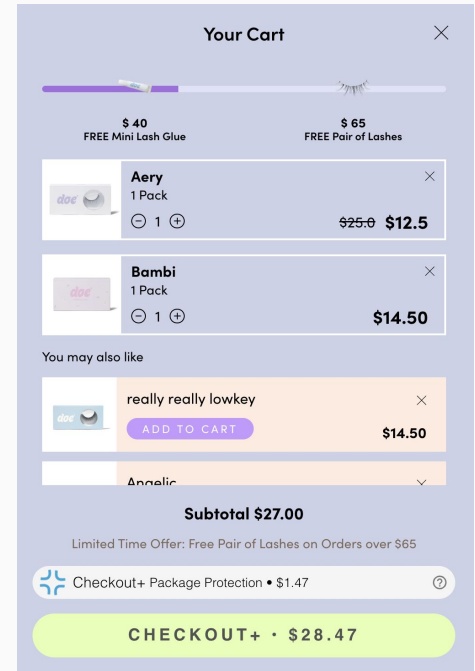
Lab 6: Session Data in Flask

GW CS 2541: Database Systems and Team Projects -
Prof. Gabe Parmer, Isabella Dayrit, Max Eichholz, Sydney Berritt, Aditya Arjun, Emil
Abbasov



How would you implement a shopping cart on a website?

- **Eg. How would you implement a shopping cart on a website that could hold your items over time? How could we store this context?**
- How could we manage this state?
- Storing these items in the database?
- What if we had a data structure that could hold this stateful information from the user?



Has this ever happened to you?

Session Timed Out Your session timed out due to inactivity. Please log in again.



Email Address

E-Mail Address

Password

Password

Login >

I Forgot My Password!

Why do you think we need this feature?

Session Data

- “Session” refers to the time between a client logging in to the server and logging out of the server
- With Flask, Session data is stored in the client’s browser on top of cookies
- Each client has their own session that is assigned a **Session ID**
- Use Cases
 - Remember a user when they log in
 - Store items in a cart while shopping online
- Sessions last for 31 days unless `SESSION_PERMANENT` is set to false (in which case they last until the browser or tab is closed)

Using Session with Flask

- The Session object is a dictionary object with key-value pairs of session variables and associated values
 - Similar syntax to Python dictionaries
- For session data to be encrypted, also set a SECRET_KEY

To set a **'username'** session variable:

```
session['username'] = "admin"
```

To release a session variable:

```
session.pop('username', None)
```

To set the session secret key:

```
app.secret_key = "any string"
```

To clear all session variables:

```
session.clear()
```

More Examples of Using Sessions

- Session data allows us to temporarily store data that we want to preserve across different pages (i.e. a logged in user or a shopping cart of products)
- **What kind of features could these properties be especially useful for?**

Checking if a session variable is set:

```
if 'username' in session:
```

Using session variables in templates:

```
<p> Hello, {{ session['username'] }} </p>
```

Redirecting in Flask: Example - Logging Out

```
from flask import Flask, redirect, url_for, session
app = Flask('app')

@app.route('/')
def login():
    ...

@app.route('/logout')
def logout():
    session.clear()
    return redirect('/')

app.run(host='0.0.0.0', port=8080)
```

- The `redirect()` function allows us to redirect users to a URL that we specify
- Instead of specifying a URL, we can also redirect to a function using `url_for()`
- For example, the following lines would be equivalent for our code example:

```
redirect('/')
```

```
redirect(url_for('login'))
```

Example: Using Sessions in Templates

```
from flask import Flask, session, redirect
app = Flask('app')
app.secret_key = "secret"
```

...

```
@app.route('/home')
```

```
def home():
```

```
    if 'name' in session:
```

```
        return render_template("home.html")
```

```
    return redirect('/')
```

```
app.run(host='0.0.0.0', port=8080)
```

Why do we check the session to make sure a user is logged in?

home.html

```
<html>
```

```
<body>
```

```
    <h1> Welcome, {{ session['name'] }} </h1>
```

```
</body>
```

```
</html>
```

We can access our session variables in templates, too!

Session variables are globally accessed

How to access session variables in Flask templates

Accessing session variables inside Flask templates is just like in Python!

Traditional for loop (using an index) in Flask templates

Breaking a string into a list of characters

Python

```
session['intro'] = ['hello', 'i', 'am', 'tim']
```

...

HTML

```
<b>{% session['intro'][0] %}</b> {% session['intro'][3][0] %}  
{% for i in range(0, session['intro']|length) %}  
    {% session['intro'][i] %}  
{% endfor %}
```

...

Python

```
word = "hello"  
letters = list(word)  
print(letters)
```



What will these print?

What other data can be stored in session?

- We can also store objects, lists, tuples, dictionaries etc.! Not just key value pairs! How?
 - `session ['this_is_list'] = [1, 2, 3, 4]`
- **How do we interact with these variables ?**
 - When storing a list in a session, you can't just append to it!
 - Store the session value in a variable, append to it locally, then replace it in the session

```
session['names'].append("Ethan")
```

...

```
names = session['names']
names.append("Ethan")
session['names'] = names
```

This does NOT work!

Do this instead!

Refresher: Flask and Database Mistakes & Tips!

1. You must set up your database connection and create a cursor object **within each function** in your Flask app
2. If you are getting a Python indentation / tab error but everything looks aligned on your screen, this is likely due to a collaboration lag in Repl. Have every group member check the spacing on their own screen and adjust!
3. If you want styling tips or aren't sure about syntax for HTML / CSS, [w3schools.com](https://www.w3schools.com) is a great resource!
4. If you need to reset your database, run the following command in the **Shell**:

```
sqlite3 <db file name> ".read <sql file name>"
```

Refresher: Using Forms in Flask

```
from flask import Flask, render_template, request
app = Flask('app')

@app.route('/', methods=['GET', 'POST'])
def get_username():
    if request.method == 'POST':
        uname = request.form["username"]
        return render_template('simple_form.html')
app.run(host='0.0.0.0', port=8080)
```

```
<body>
  <form action="/" method="POST">
    <input type="text" name="username">
    <input type="submit" name="submit">
  </form>
</body>
```

Activity 1: Login Page

1. Look at Activity 1 in the GitClassroom for the instructions. This is to be submitted for credit. Due date is on the class webpage.

Activity 2: User Login

1. Look at Activity 2 GitClassroom for the instructions. This will be submitted for credit. Due date is on the class webpage.