Midterm Review

Jinja Tags

{{ … }} for variables inserting them into <p> tags for example to print stuff out like <p> Hello {{name}} </p>

{{% … %}} This is used for basic python like code within the HTML template.

i.e. {{% if name %}} <p> Print this </p> {{% else %}} <p> Print that </p> {{%endif%}}

{{# … #}} Comment, Use this as HTML comment syntax will be read in as an error while using Jinja.

Temple Inheritance

With the models,

What is ACID? Memorize it = 2 points

Know Relationships (fill in the blank questions) **Example from: Flask Models (one to many relationships)**

class Person(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(50))

addresses = db.relationship('\_\_1\_\_\_', backref='\_\_\_\_\_2\_\_',

lazy='\_\_3\_\_')

class Address(db.Model):

id = db.Column(db.Integer, primary\_key=True)

email = db.Column(db.String(50))

\_\_\_\_4\_\_\_\_

1 = **Address**, table name to enter in

2 = **Person**, if you put some kind of garbage there, but just know that whenever you call it you do addresses.backref. so the backreef is just the name used to disambiguate two relationship with the same lazy type and tables they are related to. Backref has nothing to do with SQL, just a naming convention

3 = needs to be a relationship that is accessed by a join call. Three lazy parameters to know and use: joined, dynamic, select. **Professor will tell you which relationship it is on the exam, just enter it in the lazy parameter.**

4 = person\_id = (db.Integer, db.ForeignKey(‘person.id’)), This defines what the foreign key

Difference between one to many and one to one relationship is that a one to one relationship has a uselist=false in the 4th parameter db.relationship(), so i.e.

addresses = db.relationship(‘Address’, backref=”person”, lazy=”dynamic”, useList=false)

One to Many does not have this extra parameter

AJAX

You do not send a request with xml.open()

You can send a request with xml.send(data) or xml.send()

CIA (confidentiality, integrity, availability)

Don’t really need to memorize terms too much just know what they are.

Don’t worry about tools for identity

Know how to talk about the DDOS attack, and all other attacks. Persistent attack, reflected attack.

Never implement your own crypto = +1 point

Know difference between access control and authentication

Authentication is all about identifying who you are. If you tell me who you are, can you prove to me who you are? Access control is what we do after we know who you are, and it relies on how much access you can be given to a particular system. (Are you an admin or a normal employee)

Know how basic auth works. How does it attach it to a header?

The biggest shortcoming with basic auth is the fact that it is not encrypted. It is encoded however. With http you have the option to add a header that says that you have the basic authentication. Once you are authenticated, the browser ass this authentication Authorization: Basic Laha9aDS8n3q8bv

On the server side, it checks if auth = request.authorization. If auth does not match then send a 401 response, which is a code that sends out the message unauthorized.

{'WWW-Authenticate': 'Basic realm="Login Required"'})

^^ Most important part of basic http authentication.

Overall, just focus a bit more on HTTP authentication and learn how it works inside and out.

Functional Programming

Map(item, index, array)

Filter(item, index, array)

Reduce(Aggregator, index, array)

Map vs. for each.

For each is like an iterator. Map does what “for each” does but in constant time. Map will not modify anything. For each will look through each one with the ability to modify.