MAKE SCHOOL

Authentication & Authorization

BEW 1.2

Agenda

MAKE SCHOOL

- Learning Outcomes
- Warm-Up: Forms Review
- What is Authentication & Authorization?
- Lab Activity: Add auth to the Books application
- **BREAK**
- Continue Lab Activity
- What is a Login Manager?
- Bcrypt & Password Hashing
- Wrap-Up

Learning Outcomes

MAKE SCHOOL

By the end of today, you should be able to...

1. **Describe** the difference between Authentication & Authorization.
2. **Use** the Flask-Login library to implement sign-up & login pages in a Flask application.
3. **Use** the Flask-Login library to display user info, and restrict certain routes to only logged-in users.

MAKE SCHOOL

Warm-Up: Forms Review

Warm-Up (5 minutes)

MAKE SCHOOL

In a group of 3, go over the **Homework 3: Grocery Store (Forms)** assignment.

- How far did you get?
- What questions do you still have?

MAKE SCHOOL

What is Authentication & Authorization?

Why Authentication?

Share in the chat - Name a website where the ability to log in is an important/significant part of the site's functionality.

- YouTube
- Make School
- Amazon
- Banks

Authentication

MAKE SCHOOL

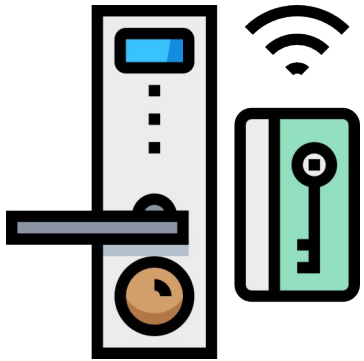
When you visit a website for the first time, you usually need to complete the following:

- Sign up for an account
  - Choose a username - can't be the same as anyone else's
  - Choose a password - must be (somewhat) secure
  - (Maybe) Give more info about yourself
- Log in
  - Your password must match what you picked previously
  - (Maybe) "Remember Me" - Stay logged in even if you closed the window

Authentication

Authentication asks the question, *"Are you who you say you are?"*. We can **authenticate** a user by checking that their password matches what is stored in the database.

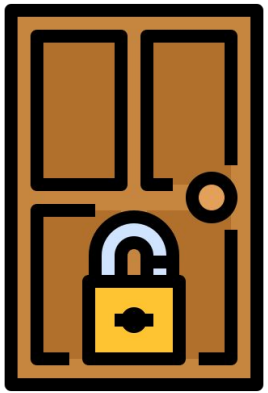
It's kind of like swiping your key-card at the door to enter a building.



Authorization

Authorization asks the question, *"What are you able to access?"*. A user is **authorized** to access certain areas of a site if they have certain credentials.

For example, all users (staff & students) can enter the building, but only certain users (e.g. only staff) can access certain rooms.



Flask-Login library

MAKE SCHOOL

To implement these features, we'll be using the **Flask-Login** library.

Let's take a look at the documentation together.

MAKE SCHOOL

Lab Activity

Lab Activity (25 minutes)

MAKE SCHOOL

With a partner, complete the Books Lab Part 3 (Auth) activity. Follow the steps to complete the code, and make sure you run your server as you go through the steps.

See if you can get up to the "Challenge" part. We'll have more time to work on this after the break.

MAKE SCHOOL

Break - 10 min

MAKE SCHOOL

The Login Manager

Login Manager

MAKE SCHOOL

The **Flask-Login** library uses a **"login manager"** to keep track of who is currently logged in. Behind the scenes, it uses cookies to store this info.

For this to work, we need to tell Flask-Login how to look up a user by their id.

```
# books_app/__init__.py
from flask_login import LoginManager
login_manager = LoginManager()

from .models import User

@login_manager.user_loader
def load_user(user_id):
   return User.query.get(user_id)
```

Login Manager

MAKE SCHOOL

We also need our **User** model to inherit from **flask_login.UserMixin**, so that it will have all of the functionality needed to support logins.

```
# books_app/models.py
from flask_login import UserMixin


class User(UserMixin, db.Model):
    id = db.Column(db.Integer, primary_key=True)
    # ...
```

current_user

The **flask_login** library gives us access to a **current_user** object, which will tell you who is logged in at any given time:

```
from flask_login import current_user


print(current_user) # <User: meredith>
print(current_user.is_authenticated) # True
```

If no one is logged in, the **current_user** object will store an **AnonymousUser**, and **current_user.is_authenticated** will evaluate to **False**.

current_user

We can even use the current_user object within templates, without having to pass it from a route. Pretty cool!

```
<!-- books_app/templates/base.html -->
{% if current_user.is_authenticated %}
  <p>Hello {{ current_user.username }}!!</p>
{% endif %}
```

bcrypt & Password Hashing

Password Hashing

MAKE SCHOOL

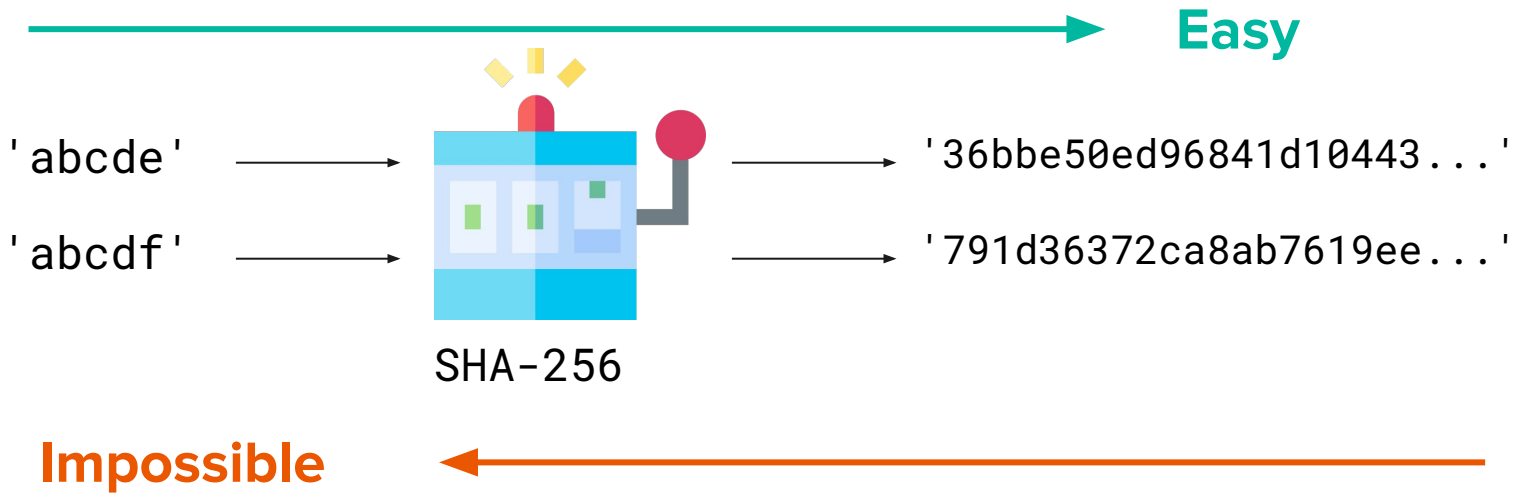
It is generally a bad idea to store **"plain text"** passwords in a database. (Why?)

Answer: If a hacker gains access to your database, they could steal everyone's passwords (and you could be sued for a lot of money).

So we use a **hash function** to disguise passwords so that a hacker can't steal them.

Password Hashing

A **hash function** is a function that is easy to compute, but (mathematically speaking) "impossible" to reverse. This means that if a hacker steals a password hash, they have no way of knowing what the original password was.



Password Hashing & bcrypt

MAKE SCHOOL

We'll be using a library called **Bcrypt** to do our password hashing.

```
password = 'thisisasecret'
hashed_password = bcrypt.generate_password_hash(password)
    .decode('utf-8')
# ... store hashed_password in database
```

Password Hashing & bcrypt

MAKE SCHOOL

So, how do we know if a password is valid?

We hash it, then compare it to the stored hash. If they match, then the password is valid.

Bcrypt can do this for us:

```
if bcrypt.check_password_hash(stored_password_hash, password):
    # passwords match, can log the user in
```

Lab Activity

Lab Activity

MAKE SCHOOL

Continue working on the lab activity. See if you can finish the **"Challenge"** section to add a **"Favorite/Unfavorite"** feature.

MAKE SCHOOL

Final Project Proposal

Final Project Proposal

Go over the Final Project requirements and answer any questions.

The proposal will require you to:

- Summarize your project idea
- Create an Entity Relationship Diagram demonstrating the relationships between database models
- List the routes you will implement for the project

MAKE SCHOOL

Wrap-Up

Wrap-Up

MAKE SCHOOL

- Homework 3 (Grocery Store): Tuesday 4/27
- Final Project Proposal: Tuesday 4/27
- Homework 4 (Grocery Store Pt. 2): Friday 4/30

***** If you receive an email that you are At Risk, please set up a meeting with me on my calendar. (Alternately, we can meet during lab time.)**