    @classmethod

    def get\_one( cls, data\_row):

        query = """

        SELECT \* FROM users WHERE id = %(id)s;

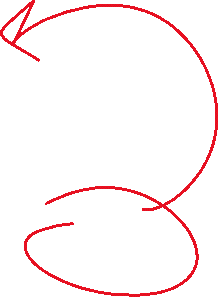
        """

        results = connectToMySQL( *cls*.db ).query\_db( query, data\_row )

        if results:

            return *cls*(results[0])

        return False



@app.route('/dashboard')

def dashboard():

    if "user\_id" in session:

        return render\_template("dashboard.html", user = User.get\_one({"id": session['user\_id']}),

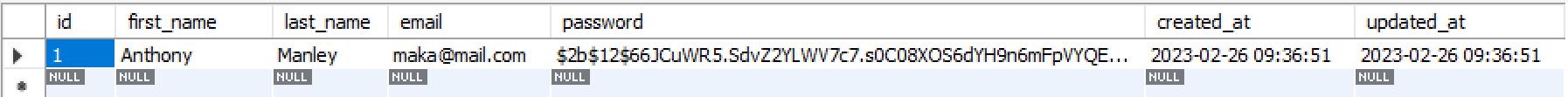
        all\_recipes = Recipe.get\_all\_recipes\_with\_creator())

    else:

        return redirect('/')

    <h1>Welcome, {{ user.first\_name }} {{ user.last\_name }}</h1>

This gives the opportunity to use all below:





This key above in tandem with the SELECT \* lets you use all of this.

@classmethod

    def get\_all\_recipes\_with\_creator(cls):

        query = "SELECT \* FROM recipes JOIN users ON recipes.user\_id = users.id;"

        results = connectToMySQL(*cls*.db).query\_db(query)

        all\_recipes = []



        for data\_row in results:

            single\_recipe = *cls*(data\_row)

            single\_recipe\_user\_info = {

                'id' : data\_row['users.id'],

                'name' : data\_row['name'],

                'description' : data\_row['description'],

                'first\_name' : data\_row['first\_name'],

                'last\_name' : data\_row['last\_name'],

                'email' : data\_row['email'],

                'password' : data\_row['password'],

                'created\_at' : data\_row['users.created\_at'],

                'updated\_at' : data\_row['users.updated\_at']

            }

            recipe\_creator = user\_model.User(single\_recipe\_user\_info)

            single\_recipe.creator = recipe\_creator

            all\_recipes.append(single\_recipe)

            print("RECIPES ARE BELOW.")

            for recipe in all\_recipes:

                print(recipe)

        return all\_recipes

{% if all\_recipes %}

                {% for single\_recipe in all\_recipes %}

                <tr>

                    <td>{{ single\_recipe.name }}</td>

                    <td>{{ single\_recipe.under\_30 }}</td>

                    <td>{{ single\_recipe.creator.first\_name }} {{ single\_recipe.creator.last\_name }}</td>

                    <td>{{ single\_recipe.description }}</td>

                    <td>{{ single\_recipe.instructions }}</td>

                    <td><a href="/recipe\_show/{{ single\_recipe.id }}">View Recipe</a>

                    {% if session['user\_id'] == single\_recipe.creator.id %}

                        <a href="/edit\_recipe/{{ single\_recipe.id }}">Edit</a>

                        <a href="/delete/{{ single\_recipe.id }}">Delete</a>

                    {% endif %}

                {% endfor %}

                {% else %}

                {% endif %}

all\_recipes = Recipe.get\_all\_recipes\_with\_creator())

REVELATION:

You need to add as many

SELECT \* FROM recipes JOIN users ON recipes.user\_id = users.id WHERE users.id = 1;



But make it return just the first inquiry back (this isn’t the end of the problem but a good start)

This could also be the current session ID

FOCUSES:

-Retrieve a list of all recipes from the user that is currently logged in on its own page (My Recipes)

-Display on a single page ALL of the information about a single recipe (including the first and last name of the person who made it)

\*Best to create the code from scratch instead of adjusting what I have made. I am misunderstanding correct query creation processes and creation of lists within class methods.

\*Currently have a working method for getting all recipes for all users. A quick run-through of HOW this works, though, will be very helpful.

RETURN TO ABORTRARY ASSIGNMENT

* ~~Create the My Trees section based on how you did it for Recipes~~
* ~~Confirm validations for character lengths~~
* ~~Create validations for date (absent in exam submission)~~
* ~~Create validation for working reason (absent in exam submission)~~
* ~~Edit page is not showing at all~~