Web site and server programming

Coding club, term 2, 2020/21

Week 1

- Install Python and VS Code as per instructions at https://pbaumgarten.com/codingclub/python_setup_instructions.pdf

main.py

```
from flask import Flask, send_file

app = Flask(__name__)

@app.route('/') # Signifies page
def index(): # '/' needs function 'index', otherwise name function same as route
    return "Hi there"

@app.route("/page2") # Type this next to 'localhost' to go to the page
def page2():
    return "This is page 2. This is fun"

@app.route("/page1")
def page1():
    return send_file('page1.html')

app.run(host='0.0.0.0', port=80, debug=True)
```

page1.html

```
<!DOCTYPE html>
<html>
    <body>
        <h1>Page 1</h1>
        <p>This is a paragraph. We're learning HTML!</p>
    </body>
</html>
```

Week 2

- Download files from here https://pbaumgarten.com/codingclub/202021-t2/week02files.zip

Update your `main.py` to the following

```python
from flask import Flask, send_file, request

app = Flask(__name__)

@app.route('/') # Signifies page
def index(): # '/' needs function 'index', otherwise name function same as route
    return "Hi there"

@app.route("/page2") # Type this next to 'localhost' to go to the page
def page2():
    return "This is page 2. This is fun"

@app.route("/page1")
def page1():
    return send_file('page1.html')

@app.route("/register", methods=['POST'])
def register():
    userid = request.values['userid']
    displayName = request.values['displayName']
    email = request.values['email']
    password = request.values['password']
    print(userid)
    print(displayName)
    print(email)
    print(password)
    return "All done"

app.run(host='0.0.0.0', port=80, debug=True)
```

Create `static/register.js` as follows

```
function runChecks(event) {
    let userName = document.querySelector("[name='userid']").value;
    let displayName = document.querySelector("[name='displayName']").value;
    let email = document.querySelector("[name='email']").value;
    let password = document.querySelector("[name='password']").value;
    let password2 = document.querySelector("[name='password2']").value;
    if (userName.length == 0) {
        alert("Username can not be empty! You numpty!");
        event.preventDefault()
    }
    if (displayName.length == 0) {
        alert("Display name can not be empty! You numpty!");
        event.preventDefault()
    }
    if (email.length == 0) {
        alert("Email address can not be empty! You numpty!");
        event.preventDefault()
    }
    if (password.length == 0) {
        alert("Password can not be empty! You numpty!");
        event.preventDefault()
    }
    if (password != password2) {
        alert("Passwords do not match! You numpty!");
        event.preventDefault()
    }
}

document.querySelector('#registrationForm').onsubmit = runChecks;
```

Week 3

Download `models.py` and `whatsthat.db` from coding club website

In your terminal: `pip install -- user flask-sqlalchemy flask_session`

Setup Python to receive registration information and save it to the database by editing the top of your `main.py`...

```
from flask import *
from flask_session import Session
from flask_sqlalchemy import SQLAlchemy
import sqlalchemy

# Create and configure Flask object
app = Flask(__name__)
app.config['SECRET_KEY'] = 'some random code here'
app.config['MAX_CONTENT_LENGTH'] = 16 * 2**20 # 16 MB maximum upload
app.config['ALLOWED_EXTENSIONS'] = set(['pdf', 'png', 'jpg', 'jpeg', 'gif'])
app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_FILE_DIR'] = os.path.join(app.root_path, "cache")
app.config['SESSION_FILE_THRESHOLD'] = 1000 # Allow up to 1000 sessions
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///whatsthat.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
Session(app)
db = SQLAlchemy(app)
from models import *    # Import models.py
```

New routes to add...

```
@app.route('/register', methods=["POST"])
def user_register():
    error = ""
    # Was an avatar uploaded?
    if request.files['avatar'].filename != '':
        f = request.files['avatar']
        dot = f.filename.rfind(".")
        extension = f.filename[dot:].lower()
        avatar = request.values['email'].replace("@","_at_") + extension
        f.save(app.root_path + '/photos/' + avatar)
    else:
        avatar = ""
    user = User(userid=request.values['userid'],
        displayName=request.values['displayName'],
        password=request.values['password'], # WARNING THIS IS NOT SECURE
        email=request.values['email'],
        avatar=avatar)
    db.session.add(user)
    db.session.commit()
    return redirect("/static/login.html")
```

Week 4

Create new routes in `main.py`...

```
@app.route('/login', methods=["POST"])
def user_login():
    print("/login: "+request.values['userid'])
    uid = User.query.filter_by(userid = request.values['userid']).one_or_none()
    # Did we get a user?
    if uid is None:
        return "Invalid login - No userid provided"
    # Does the password match?
    pw_attempt = request.values['password']
    if uid.password != pw_attempt:
        return "Invalid login - Password incorrect"
    # Create the session
    session['userid'] = uid.userid
    session['displayName'] = uid.displayName
    session['avatar'] = uid.avatar
    print("/login: USER LOGGED IN: ",uid)
    return redirect("/static/main.html")

@app.route("/whoami", methods=['GET', 'POST'])
def who_am_i():
    if 'userid' in session:
        return jsonify({
            'displayName': session['displayName'],
            'userid': session['userid'],
            'avatar': session['avatar']
        })
    else:
        return "error - not logged in"
```

Edit `static/main.js`...

```
function getDisplayName() {
    fetch("/whoami").then(function(response) {
        response.json().then(function(data) {
            document.querySelector(".hello").innerHTML = "Hello, "+data.displayName+'.';
        })
    });
}

// Once everything has finished loading, run the main() function
window.onload=getDisplayName;
```

Week 5

Save new messages to database

```
# Add to your imports...
from datetime import datetime

# New route to add...
@app.route('/newmessage', methods=["POST","GET"])
def new_message():
    if 'message' in request.values and 'name' in session:
        new_message = Message(
            userid=session['userid'],
            displayName=session['displayName'],
            ip=request.remote_addr,
            message=request.values['message'],
            dateTime=datetime.utcnow())
        db.session.add(new_message)
        db.session.commit()
        print("Message from "+session['userid']+" at ip address
"+request.remote_addr+" saved")
        print(request.values['message'])
        return "ok"
    else:
        return "Error: Not logged in, or no message received"
```

Edit `static/index.js` to send messages from the browser

```
function sendMessage(event) {
    let form = new FormData();
    let mesgBox = document.querySelector("#messagetext").value;
    if (mesgBox.length > 0) {
        form.append("message", mesgBox);
        fetch("/newmessage", {method: "POST", body: form}).then(response => {
            response.text().then(data => {
                console.log(data);
            })
        }).catch(err => {
            console.log(err);
        });
        document.querySelector("#messagetext").value = '';
        document.querySelector("#messagetext").focus();
    } else {
        alert("Can't send empty message");
    }
}

// Create function main...
function main() {
    getDisplayName();
    document.querySelector("#messagebutton").addEventListener("click",
sendMessage);
}

// Replace the existing window.onload with this...
window.onload = main;
```

Edit `static/main.js` to allow the enter key to send messages (instead of having to use the button)

```
// New function...
function checkForEnter(event) {
    if (event.keyCode == 13) {
        sendMessage(null);
    }
}

function main() {
    getDisplayName();
    document.querySelector("#messagebutton").addEventListener("click",
sendMessage);
    // Add this line...
    document.querySelector('#messagetext').addEventListener('keydown',
checkForEnter);
}
```

Week 6

Get messages from Python

```
@app.route('/getmessages',methods=['GET'])
def get_messages():
    if 'since' in request.args:
        since = int(request.args['since'])
    else:
        since = 0
    msgs = Message.query.filter(Message.id > since).order_by(Message.dateTime.desc()).limit(100).all()
    reply = []
    for msg in msgs:
        reply.insert(0,msg.as_dict())
    return jsonify(reply)
```

Edit `static/main.js` to display messages on your HTML

```
// Add this variable declaration to the top of your Javascript...
var mostRecentMessage = 0;

// New function...
function getNewMessages() {
    fetch("/getmessages?since="+mostRecent).then( response => {
        response.json().then( data => {
            console.log(data);
            if (data.length > 0) {
                let html = "";
                for (let message of data) {
                    let dt = new Date(message.dateTime * 1000);
                    let dt_display = dt.toLocaleDateString() + " " +
dt.toLocaleTimeString();
                    let item = `
                    <div class='messageGrid'>
                        <div class='messageAvatar'><img
src='/getavatar/${message.userid}'></div>
                        <div class='messageName'>${message.displayName}
${dt_display}</div>
                        <div class='messageContent'>${message.message}</div>
                    </div>
                    `;
                    html = html + item;
                    if (message.id > mostRecent) {
                        mostRecent = message.id;
                    }
                }
                document.querySelector("#chatcontent").innerHTML += html;
                document.querySelector(".container").scrollTop =
document.querySelector(".container").scrollHeight;
            }
        });
    });
}

function main() {
    getDisplayName();
    document.querySelector("#messagebutton").addEventListener("click",
sendMessage);
    document.querySelector('#messagetext').addEventListener('keydown',
checkForEnter);
    // Add this line...
    setInterval(getNewMessages, 1000);
}
```

Get the avatar by updating `main.py`

- Download the `avatar_placeholder.png` and place it in your project folder.

```
@app.route('/getavatar/<userid>', methods=['GET'])
def get_avatar(userid):
    user = User.query.filter(User.userid == userid).one_or_none()
    if user is not None and user.avatar != "":
        return send_file(app.root_path + '/photos/' + user.avatar)
    else:
        return send_file(app.root_path + '/static/avatar_placeholder.png')
```

- Edit your `static/index.css` file by adding the following

```
.messageAvatar img {
    width: 48px;
    height: 48px;
}
```