**Flash tutorial video notes**

* Flask is the module in python used for handling HTML files.
* The render template function returns the HTML file in the templates folder as the route given to it.
* app.run(debug = True) is used to automatically restart .py file after editing over manual way
* Jinga Templating:
  + <https://jinja.palletsprojects.com/en/3.1.x/>
  + {{ name }}
  + {{ url\_for(‘static’ , filename=’path’) }}
* Template inheritance
  + {% block *<name>* %}
  + {% endblock *<name if more than one>* }
  + {% extends “*<name of base HTML file>*” %}

Video

* XAMPP
  + Create a new database.
  + Create tables in it as required.
* Flask SQLAlchemy
  + It is used to link the python app to the database
  + Inherited Class comprising of columns to the database is made
  + Refer to <https://flask-sqlalchemy.palletsprojects.com/en/2.x/quickstart/?highlight=quick%20start> for more details
  + MySQL URI for SQLAlchemy
  + Inherited Class with the name of the table is created of class SQLAlchemy and the object of SQLAlchemy is used to link columns of the database with variables used to access and modify data in the database.
* Configuring .json file
  + Create a config.json file.
  + The elements in this file can be used in python by importing json module in main file.
  + The contents from json file are read in the python file using with open() file handling technique.

with open('config.json','r') as c:

    params= json.load(c)['parameters'] #referred params

* + For accessing items in json file to html file jinja templating is used.
* Automated mailing when contact form is filled
  + From the flask\_mail module Mail method is used to send automated emails with the app.
  + smtp.gmail.com (gmail smtp) server is used here to send the mail.
  + Allow less secure app to send emails (2-factor authentication, app permission)
  + Flask Mail Documentation: <https://pythonhosted.org/Flask-Mail/>

#Automated Mail

app.config.update(

    MAIL\_SERVER = 'smtp.gmail.com',

    MAIL\_PORT = '465',

    MAIL\_USE\_SSL = True,

    MAIL\_USERNAME = params['gmail-user'],

    MAIL\_PASSWORD = params['gmail-password']

)

mail = Mail(app)

        mail.send\_message(

            "New message from blog",

            sender=email,

            recipients= [params['gmail-user']],

            body= f"New Message from {name}.\n{message}\nSender phone: {phone}",

        )