Get	Post	Put	Delete

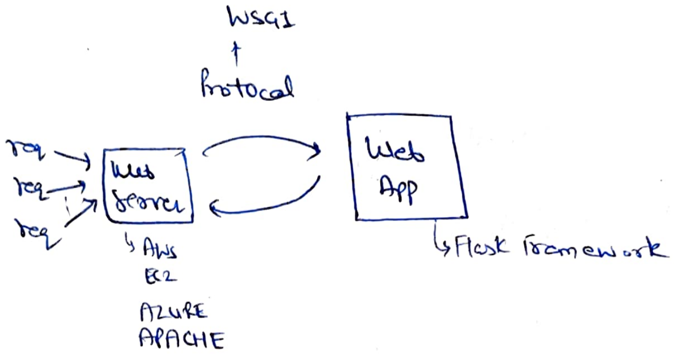
HTTP verbs: 4

[Day-14]

Flask Framework - Web framework which is created with the python program.

① WSGI (Web server Gateway Interface)

② Jinja2 Template Engine — combines web templates with a data source



→ Basic Structure:

```
from flask import Flask          ①

app= Flask(_name_)               ②
------ #code# ---
if _name_ == "main":             ③
    app.run()
          debug=True  ------ #realtime updation after saving
```

It creates an instance of Flask class, which will be your WSGI application

initializing flask

entry point of appl^n & inside running it

```
@app.route("/")
def welcome():
    return "welcome"


@app.route("/")
def welcome():
    return "<html> <H3> Welcome </H3> </html>
```

```
# from flask import Flask, render_template
app = Flask(_name_)

@app.route("/index")
def index():
    return render_template('index.html')
```

(responsible for redirecting to that particular HTML page)

```
@app.route('/about', methods=['GET'])
def about():
    return render_template('about.html')

@app.route('/form', methods = ['GET, 'POST'])
def form():
    if request.~~method~~ ~~form~~ method == 'POST':
        name = request.~~method~~ form ['name']
        return f'Hello {name} !'
    return render_template('form.html')

if _name_ == "main":
    app.run(deby=True)
```

variable rule

```
@app.route('/success/<int:score>')
def success(score):
    return" Marks are" + str(score)
```

Jinja2 Template Engine

{{ }} → expressions to print output in html , {% ... %} → conditions, for loops

{# ... #} → this is for comments.

Streamlit : Open-source app framework for ML & DS projects.

→ It allows to create beautiful web applications for ML & DS project

```
import streamlit as st

#title of application
St.title("Hello Streamlit")

# display simple text
St.write("text")

# display dataframe
St.write("Here is dataframe")
St.write(df)

#  create line chart
Chart_data = pd.DataFrame(
        np.random.randn(20,3), columns = ['a','b','c']
)
St.line_chart(chart_data)

# st.slider("Select age", 0,100, 25)          —— Slider

# options = ["C++", "Java", "Python"]         —— choose from options
choise = st.selectbox("Choose Lang.", options)
St.write(f"You selected {choice}.")

# file upload
st.file_uploader("Choose file", type="csv")
```