FastAPI INTRODUCTION

'What is FasAPI

- FastAPI is a moder, high-performance web framework for building APIs with python.
- it is built with starlette and pydantic

(i)Starlette:
it manages how your API receives requests and sends back responses

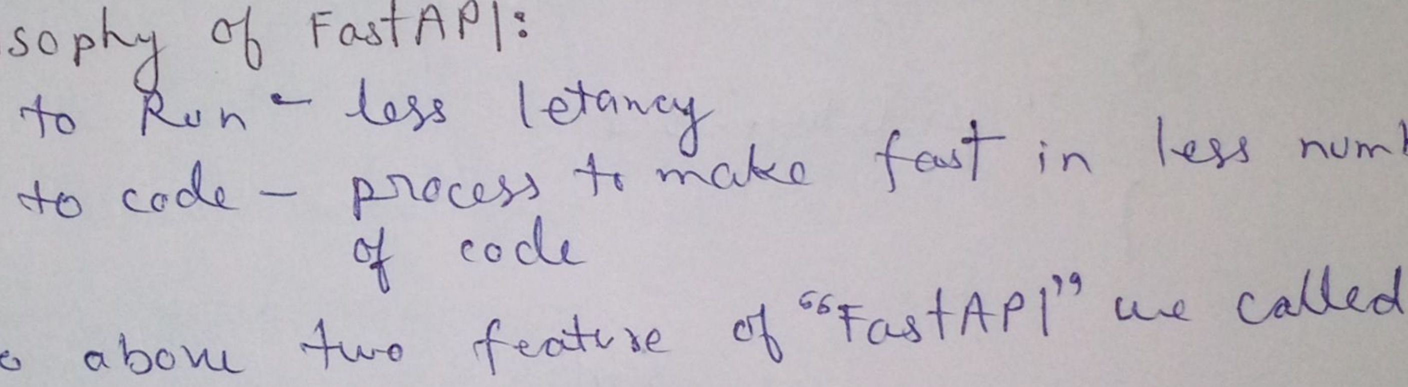
(ii)pydantic:
pydantic is used to check if the data coming into your API is correct and in the right format

~~phy~~ Philosophy of FastAPI:

{Fast to Run - less letancy
Fast to code - process to make fast in less number of code

due to above two feature of "FastAPI" we called it FastAPI

(I) Why FastAPI is Fast to run?
let's take an example. here we make a ML model'
API's end point(predict), we provide the ~~$~~ input f1 and f2 then he predict and give me an answer



① http request
② convert the http request into python understadable
③ convert python code into http understadable

①
```
POST/predic HTTP/1.1
Host: api.example.com
Content-Type: application/json
Content-Length: 45
{
    "feature1": 5.2,
    "feature2": 3.1
}
```

②
```
Request.method --> "POST"
Request.url --> "/predict"
Request.json() --> {"feature1":5.2, "feature2":3.1}
```

③
```
HTTP/1.1 200 Ok
Content-Type: application/json
{
    "prediction": 8.3
}
```

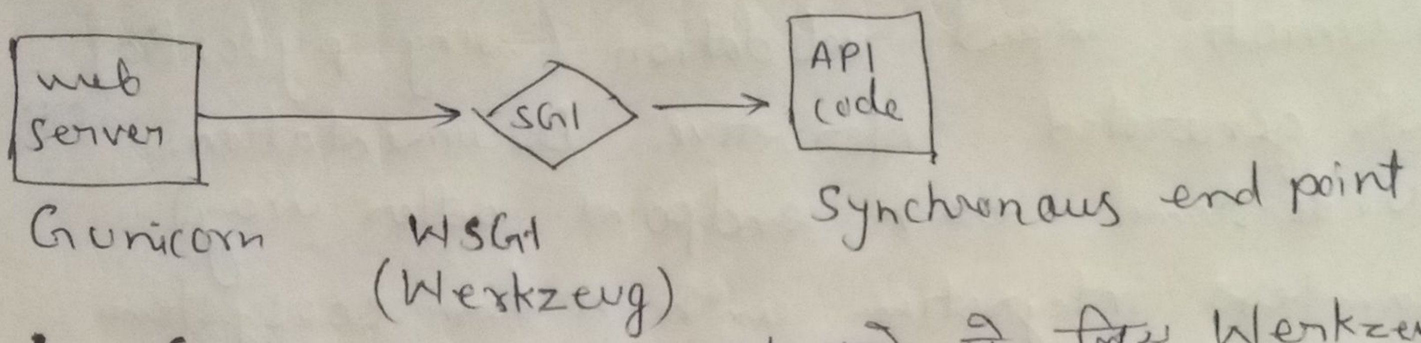
SGI
(Servier Gateway Interface)

WSGI
Web Server Gateway interface
W → Werkzeug
used in ~~FastAPI~~ flask

ASGI
used in FastAPI

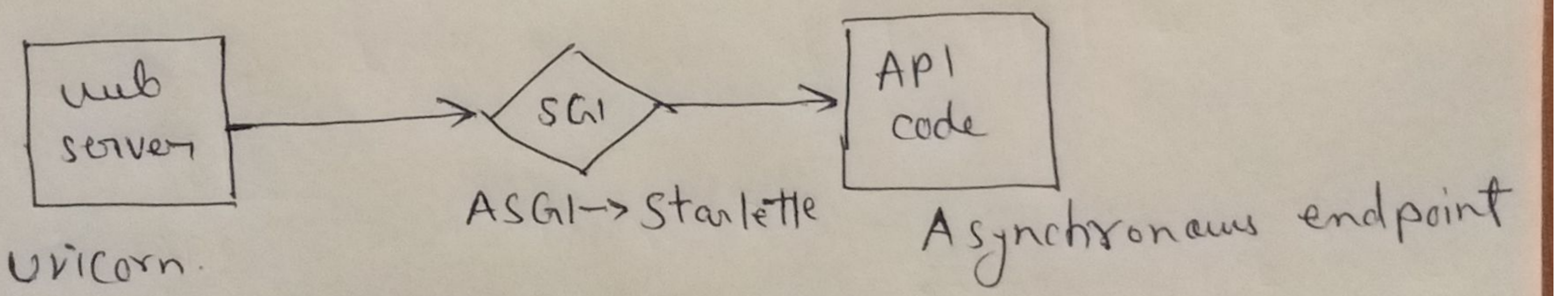
WSGI Disadvantages
- has some limitations, particularly when dealing with high accuracy or I/O bound applications.
- it synchronous nature and blocking architecture can lead to slower request processing and scalability challenges
इसका मतलब हम एक बार मे एक ही request process कर सकते हैं।

- Same abone architecture in flask



- in flask WSGI user ~~करत~~ करने के लिए Werkzeug use करता है
- Werkzeug: it is comprehensive WSGI web application library
  Werkzeug German noun: "tool". Etymology: Werk ("Work"). zeug ("stuff")
- Gunicorn: it is a WSGI HTTP server, here performance issues, including I/O wait time and high latency, can arise from misconfiguration and debugging these can be challenging

- Same abone architecture in FastAPI



- ASGI is Asynchronous.
- web we used the starlette library
- here webserver used is uvicorn (handle multiple parallel request)
- FastAPI supports "async" and 'await" feature of a python (used thus we can perform parallel processing)

(II) Why FastAPI is fast to code?

- Automatic input validation (using pydantic)
- Auto generated interactive Documentation. (to understand each endpoint better way)
- Seamless integration with modern ecosystem (ML/DL libraries, OAuth, JWT, SQL Alchemy, Docker, kubernetes etc.)