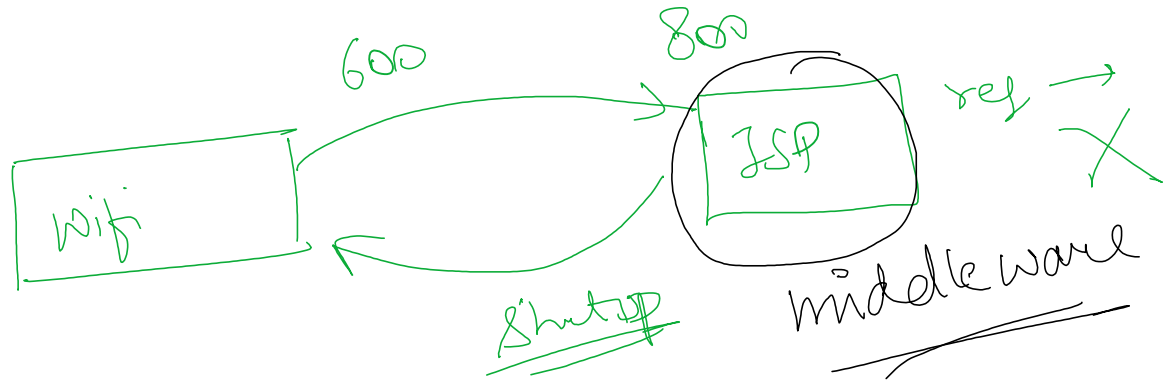
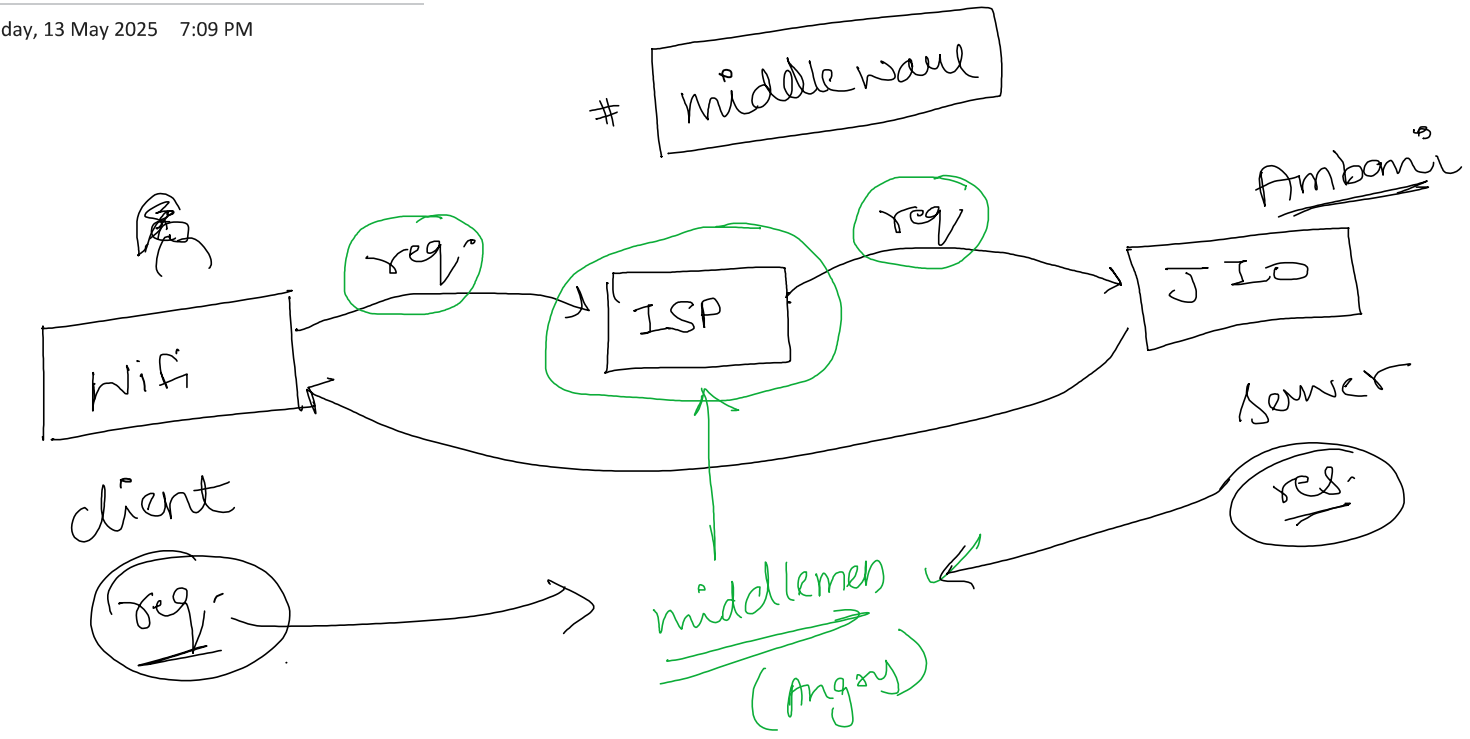
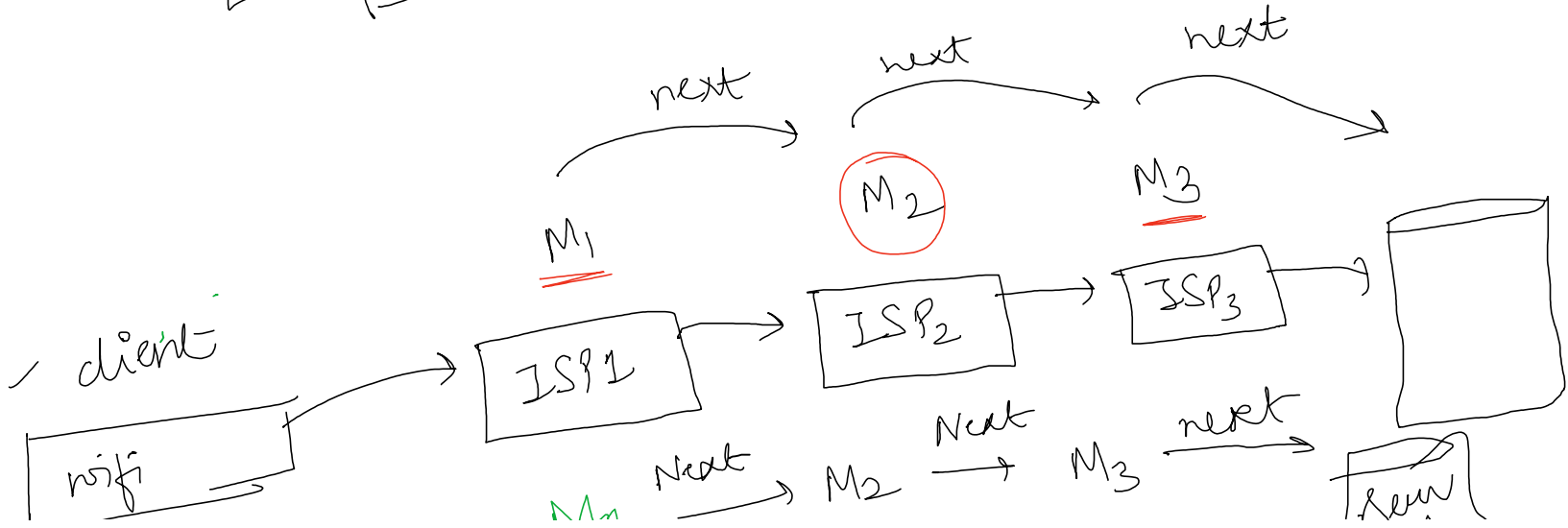


Day-18 (middleware + db)

Tuesday, 13 May 2025 7:09 PM



middlewares?





middleware = ~~fn~~ (4) X  
3 req, res, next

const verify = () => { }

middleware { = (req, res, next) => {  
}

Types use defined => login  
prebuilt => express.json()

✓ req. params ⇒ http://localhost:5050/products/3?id=420

↳ proto    domain    path

req. query ← id

✓ req. body ⇒ POST method (data)

↳ By default = undefined

↳ data → middleware → express.json()

↳ see X

app.use(express.json()) ✓

↳ real data ✓

① fn

② it runs in b/w 2 things

✓ ③ it always runs on every incoming (method)

request ( path )

{ app.get ('/som') ✓  
 post ('/baru') ✓  
 delete ('/bechout') ✓ }

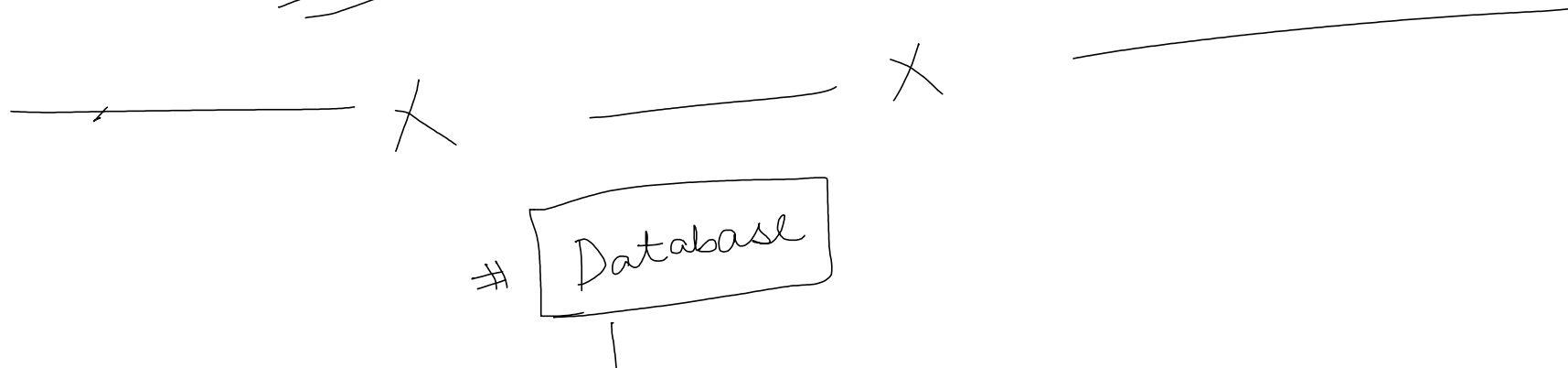
http://localhost:5050 /som ? user = samarth URL

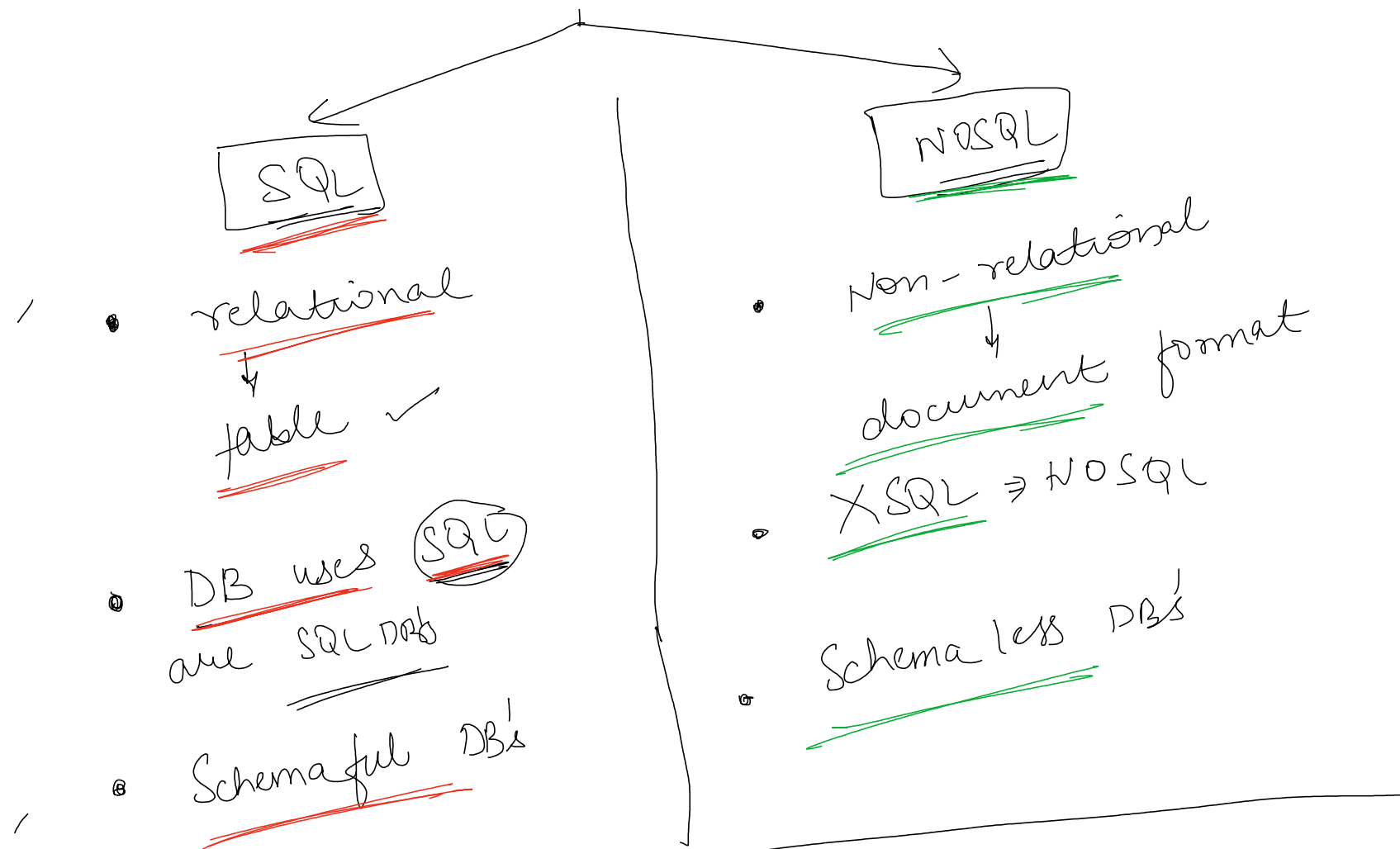
middleware

next() ⇒ user = samarth  
 Break ⇒ res ⇒ Invalid user

req. query ✓

req. params



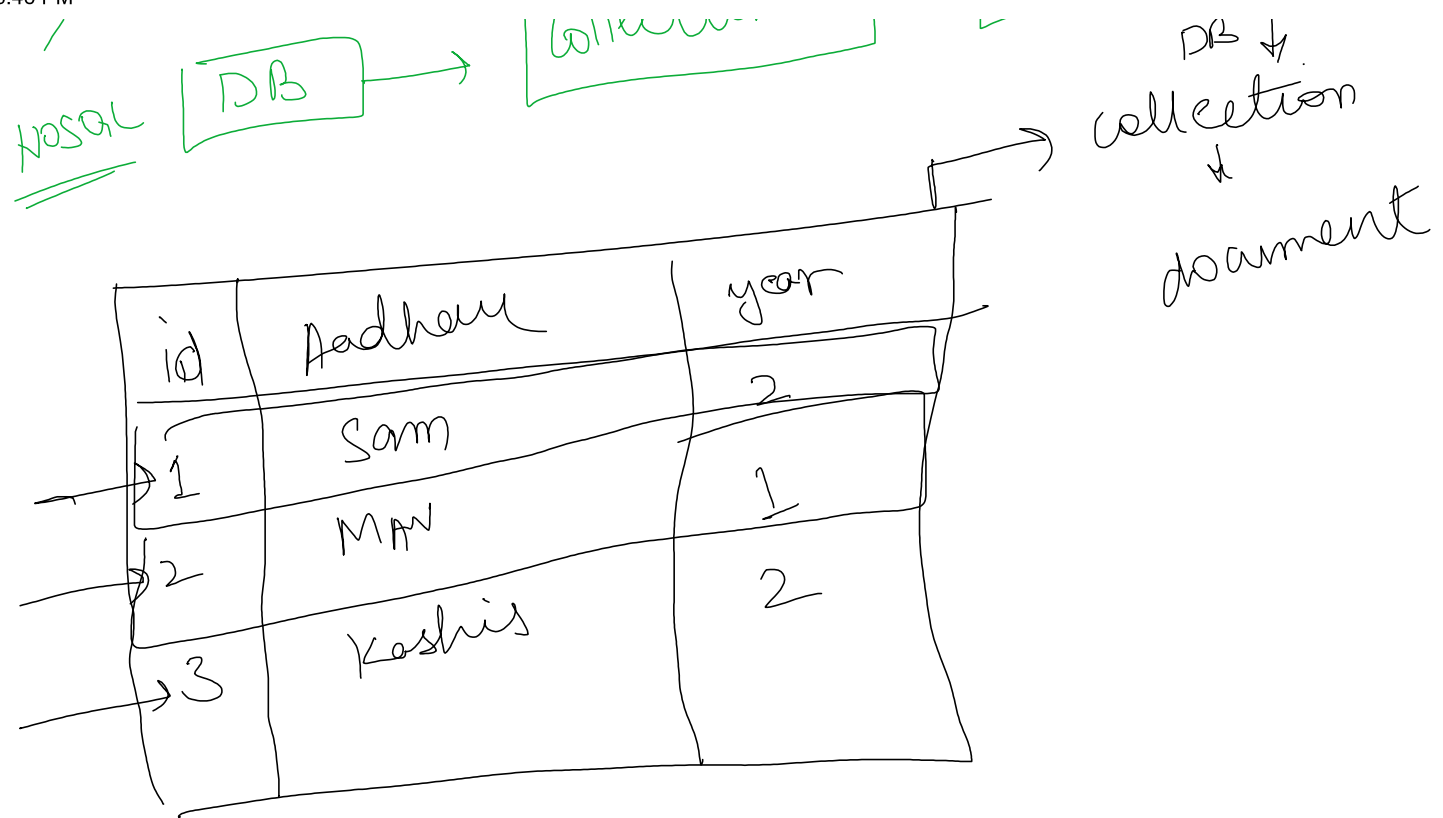


SQL → structured  
Query  
Language

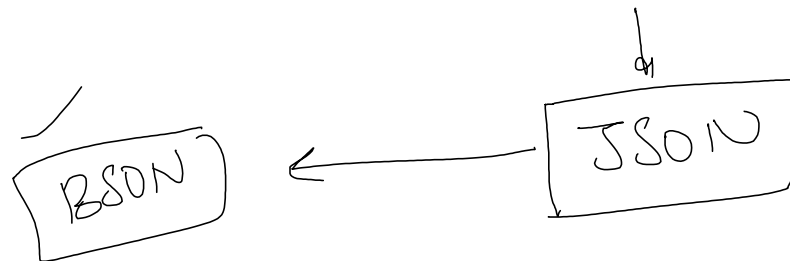
NoSQL → MongoDB (Schemaless)

SQL DB → tables → row & col

NoSQL → documents → document



NOSQL  
↳ MongoDB  
↳ document



① Show dbs ↔ databases  
(na)

- ✓
- ②

use 'phone (10)
- ③

show collections
- ④

db. createCollection('samsung')
- ⑤

db. samsung. drop()
- ⑥

db. dropDatabase()
- ⑦



Laptops

✓

└─ Apple = 1/2

└─ Windows = 1/2

update  
delete

→ use laptops

//create

```
db.apple.insertOne({name:"macbook pro1" , price:200000, year: 2018, isBought:true})
```

```
db.apple.insertMany([ {name:"macbook pro2" , price:300000, year: 2022, isBought:false} , {name:"macbook air" , price:90000 , year:2022, isBought:true}])
```

```
//read
db.apple.find()
```

```
db.apple.find({year:2022 , isBought:false})
```

```
//update
db.apple.updateOne( {price:200000} , { $set : {year:2020 , name:"Macbook-pro1"} } )
```

```
db.apple.updateMany( {year:2022} , { $set : {year:2024} } )
```

*shell => mongosh*

```
//delete
db.apple.deleteOne({isBought: true})
```

*\* DB methods they all return a Promise \**

*VScode handle*



