Day-19 ()

Saturday, 17 May 2025   5:09 PM

\# [NodeJS] → Run time env
                          ___
                           ↓
                          JS

\# [express] → framework for NodeJS
                          ↓
                       [server]

(JS) [api's] → methods
                 [GET, POST,
                  PUT, PATCH
                  DELETE]

middlewares → [fn's] → 3 args

req , res , next

(obj) (obj) (fn)

req . params → dynamic routing

∞ ||

req . query → [?]

req . body → POST data

By default = undefined

⇒ middlewarec

(prebuilt)
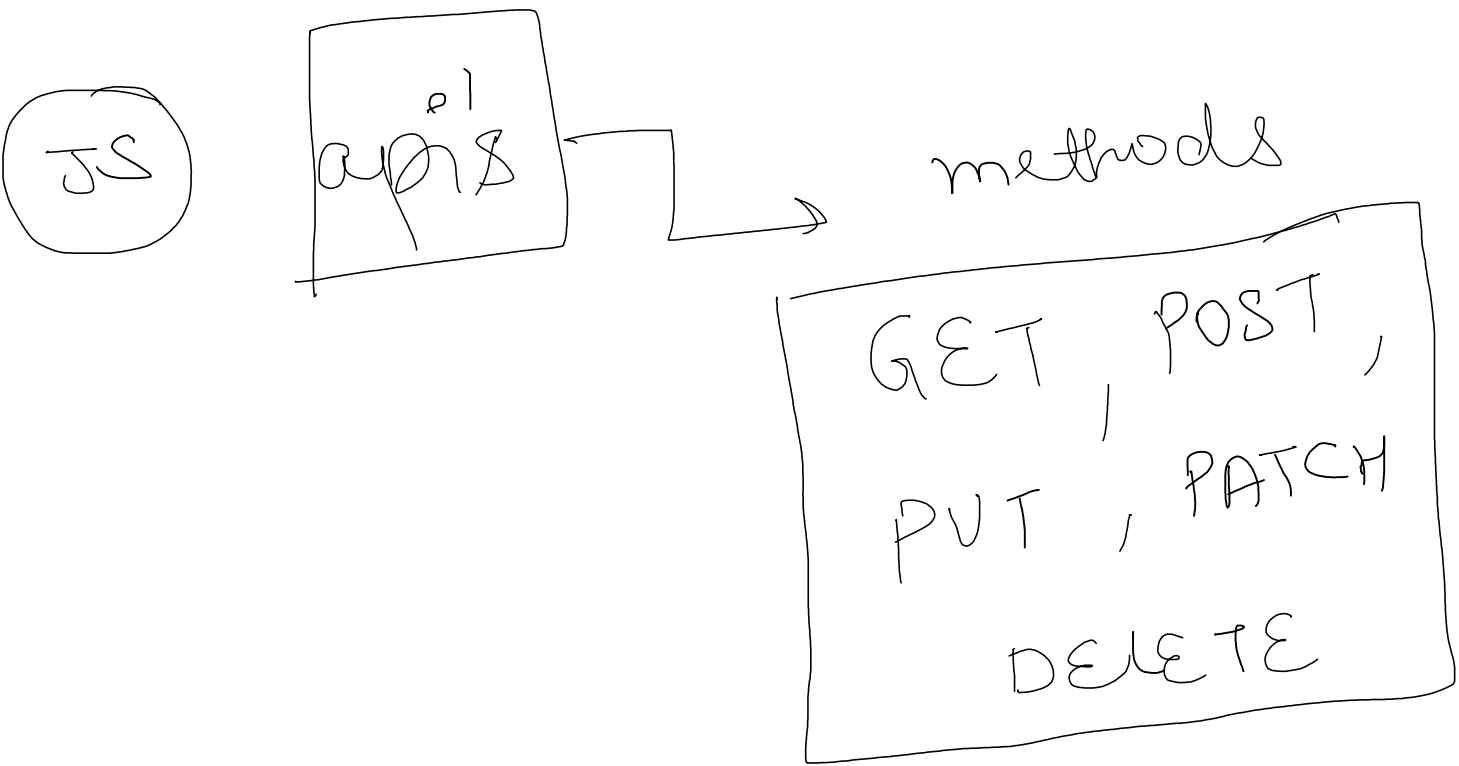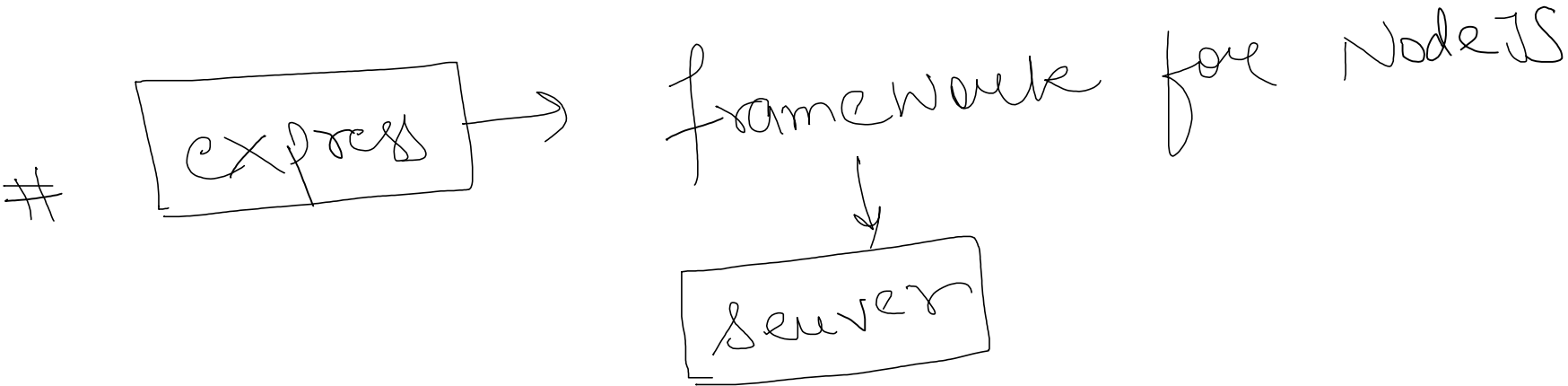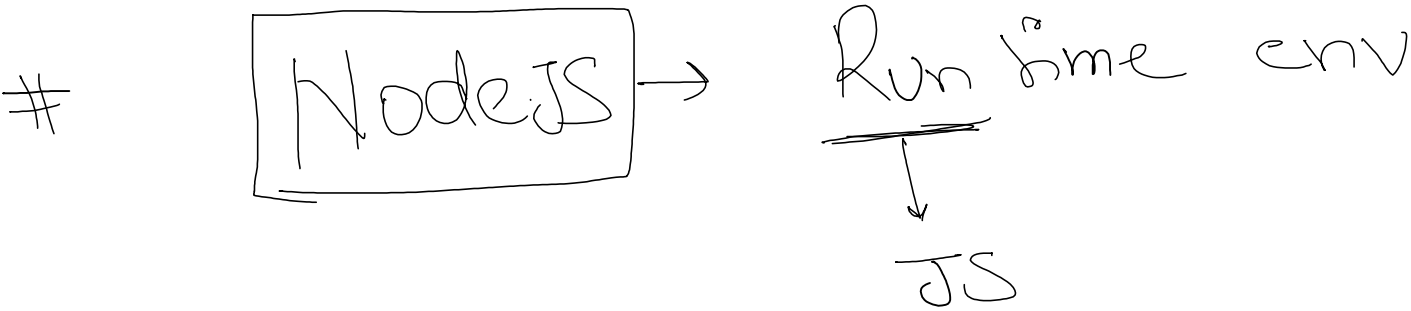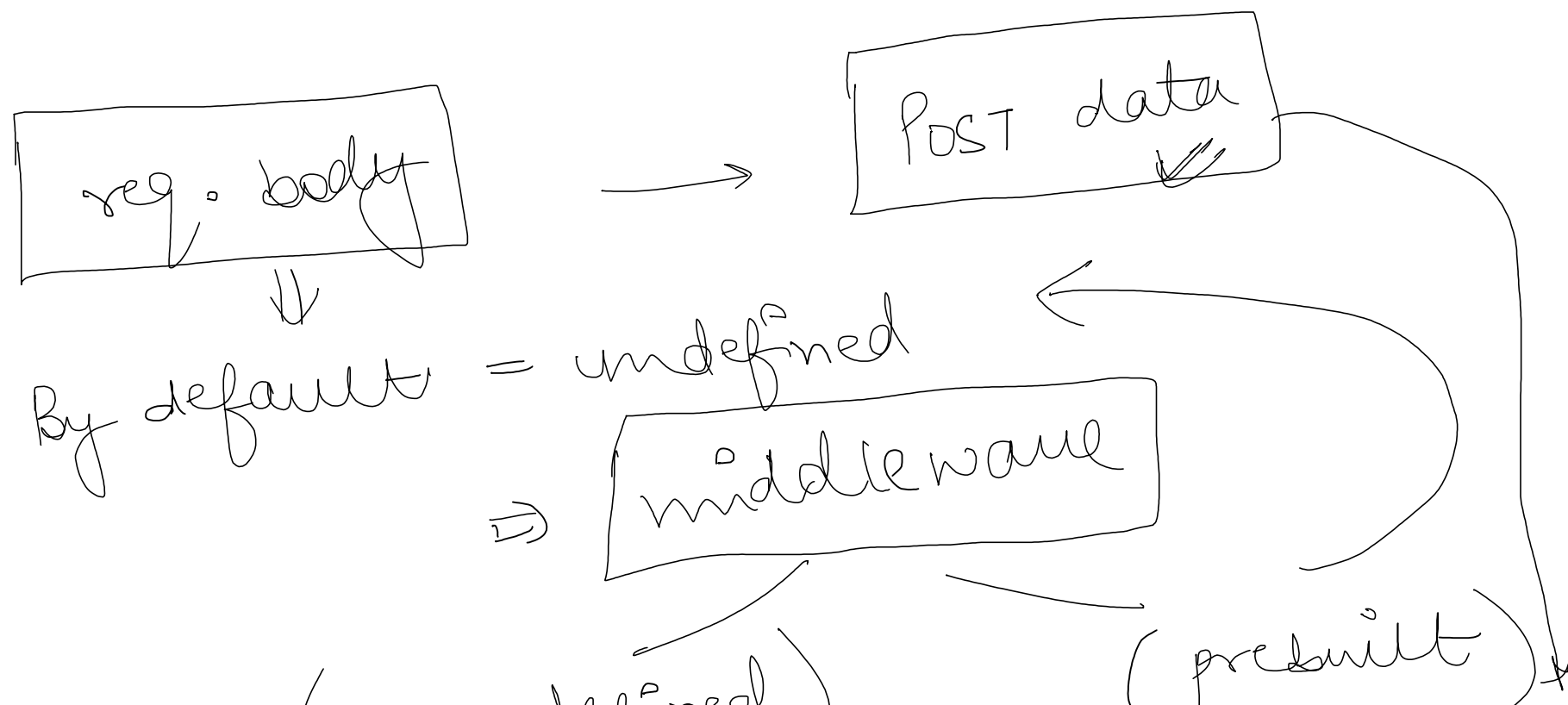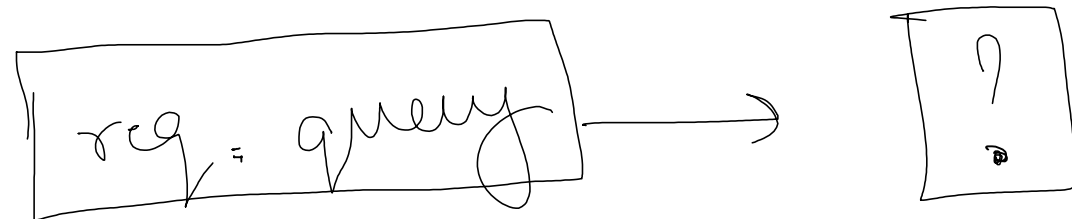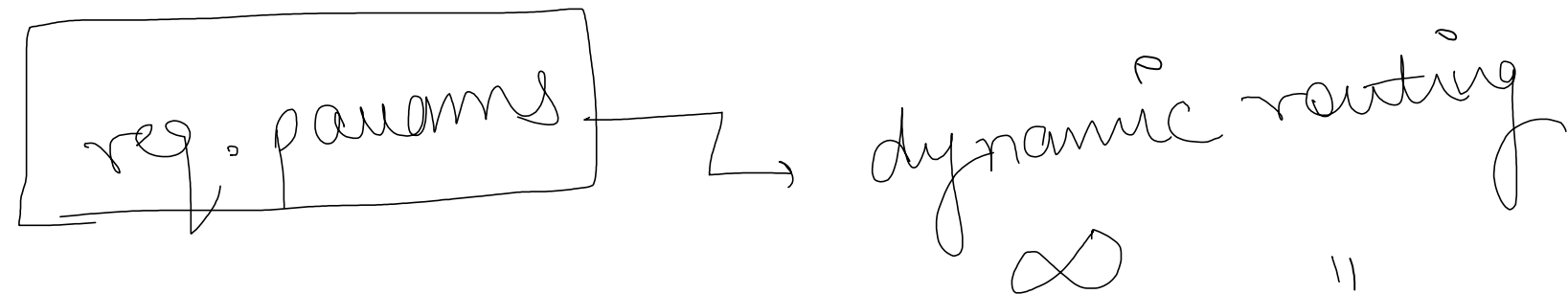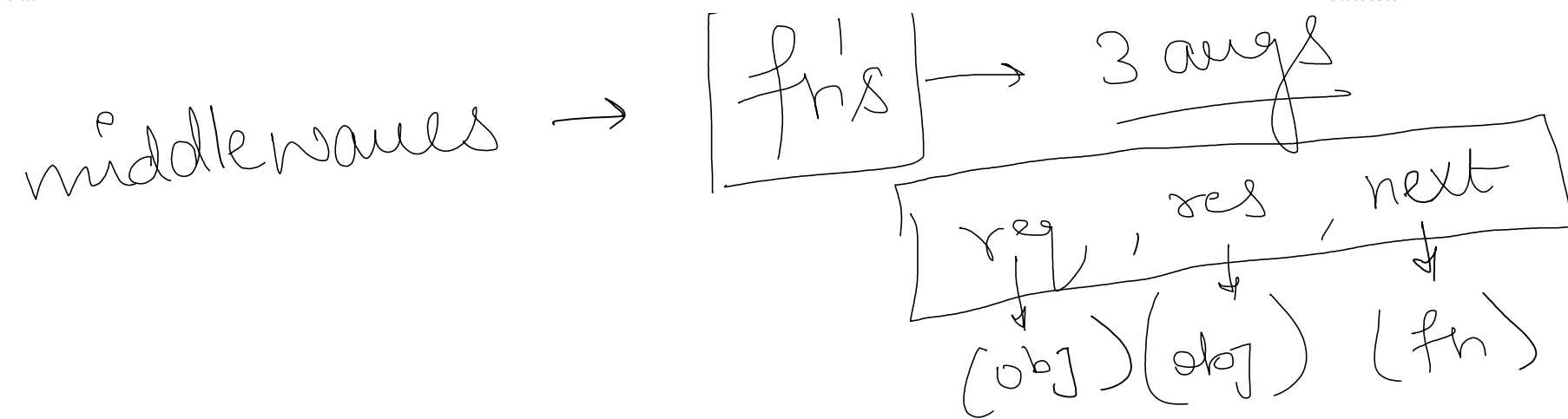
(user defined)

* | app.use (express.json()) |

X → X → X

| RESTful | architecture

{ REpresentational
  State
  Transfer }

{ Way of
   writing code

{ | REST api | → (data) } (No
  RESTful route → pages   use)

task / note

path + method ⇒ (api)

pages

L ←———————————— R
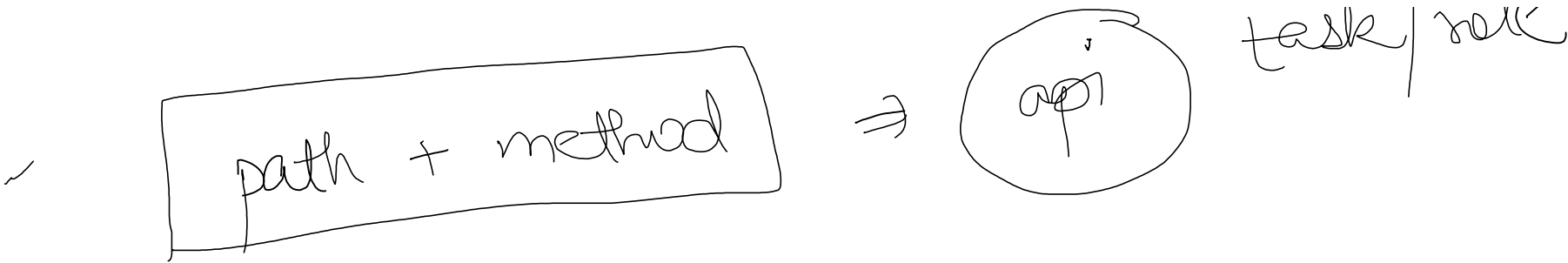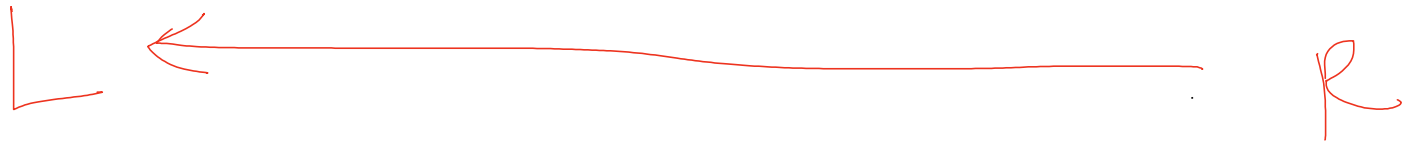
Route name	URL	HTTP Verb	Description
Index	/blogs	GET	Display a list of all blogs
New	/blog/new	GET	Show form to make new blogs
Create	/blogs	+ POST	Add new blog to database, then redirect (api)
Show	/blogs/:id	GET	Show info about one blogs
Edit	/blogs/:id/edit	GET	Show edit form of one blog
Update	/blogs/:id	PUT	Update a particular blog, then redirect
Destroy	/blogs/:id	DELETE	Delete a particular blog, then redirect

(req . params)

DB → NOSQL → MongoDB   (DB)
                              ↓
                         collection   (table)
                              ↓
                         document
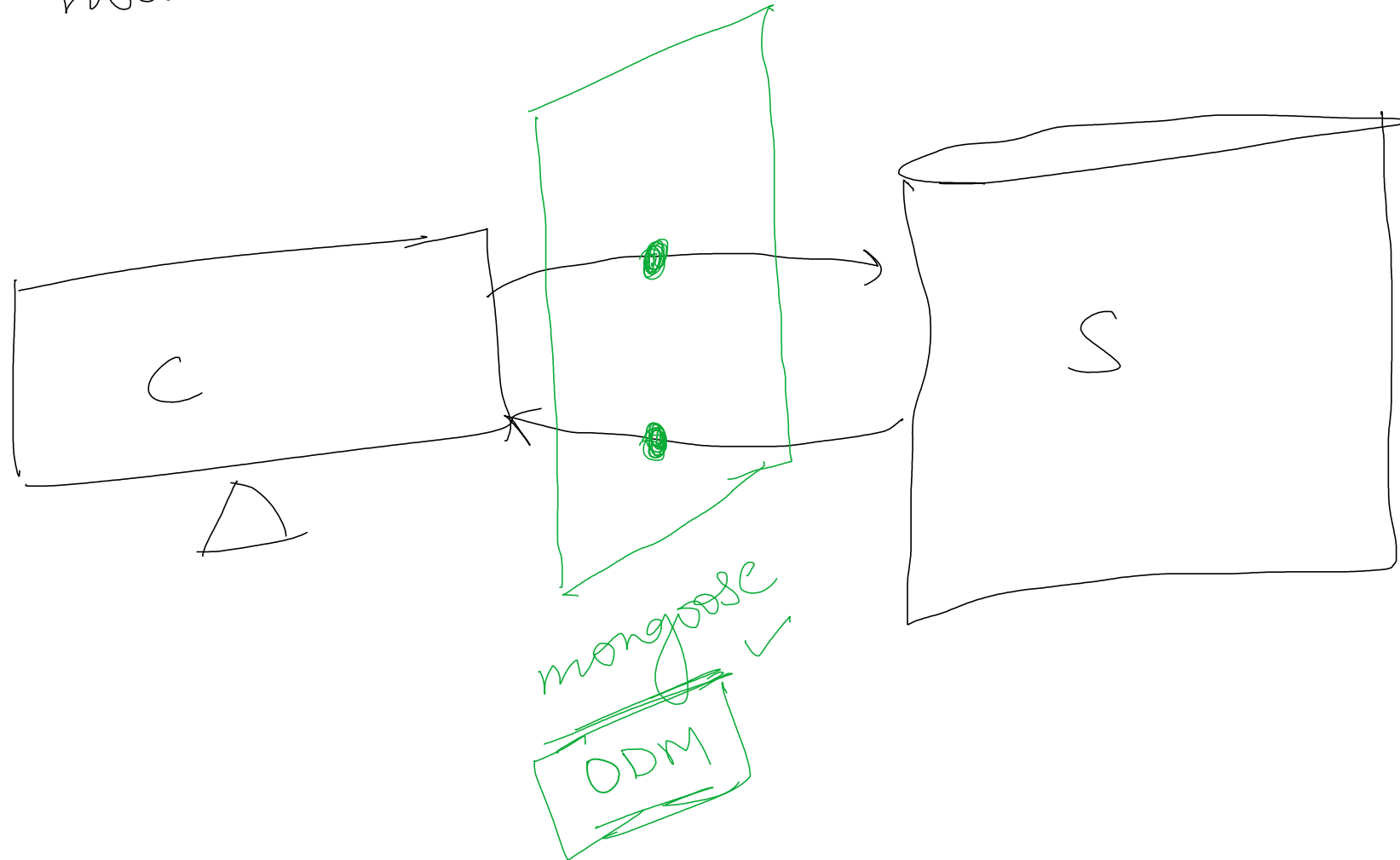
All the DB methods returns you a
✓ PROMISE

——— X ——— X ——— X ———

mongosh X ✓   { RESTapi
                    +
                  Express   } → Rest fastly
VS code X   +
                 MongoDB

ODM (object)

express

nodemon

$\rightarrow$ ODM (data mapper)

mongoose ✓

ODM

C

S

document (mongosh)

DB $\longrightarrow$ Collection $\longrightarrow$

DB $\longrightarrow$ Model $\longrightarrow$ documents

CRUD
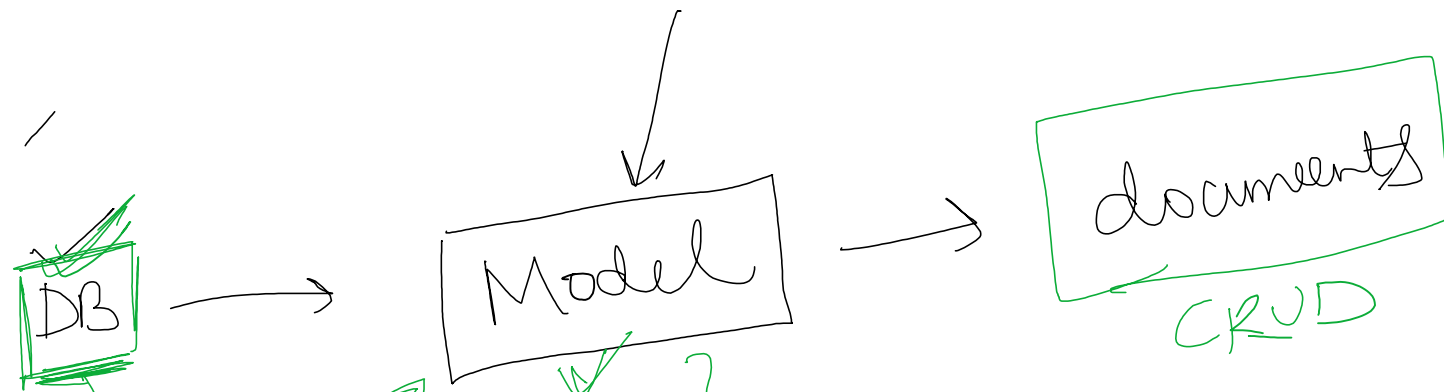
(VS)

Schema $\Rightarrow$ **Blueprint** of how the
Model /collection will look,

X — — VScode — X

MVC → Model → Collection
architecture → Views → pages
↓ Controllers → logic
way of writing
code

{ auth
  login
  Rest }

routes

path

{ auth login Rest }

runs
↓
middlewares → every req.

index.js

routes
/login , /register

Controller
(logic)

DB

Model ⟹ finally DB updates

Schema → Model

Always deploy github → { node_modules
                            package-lock.json

git ignore
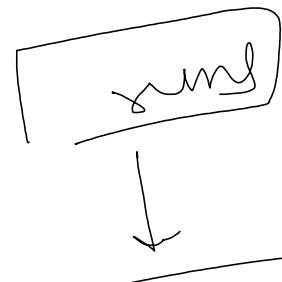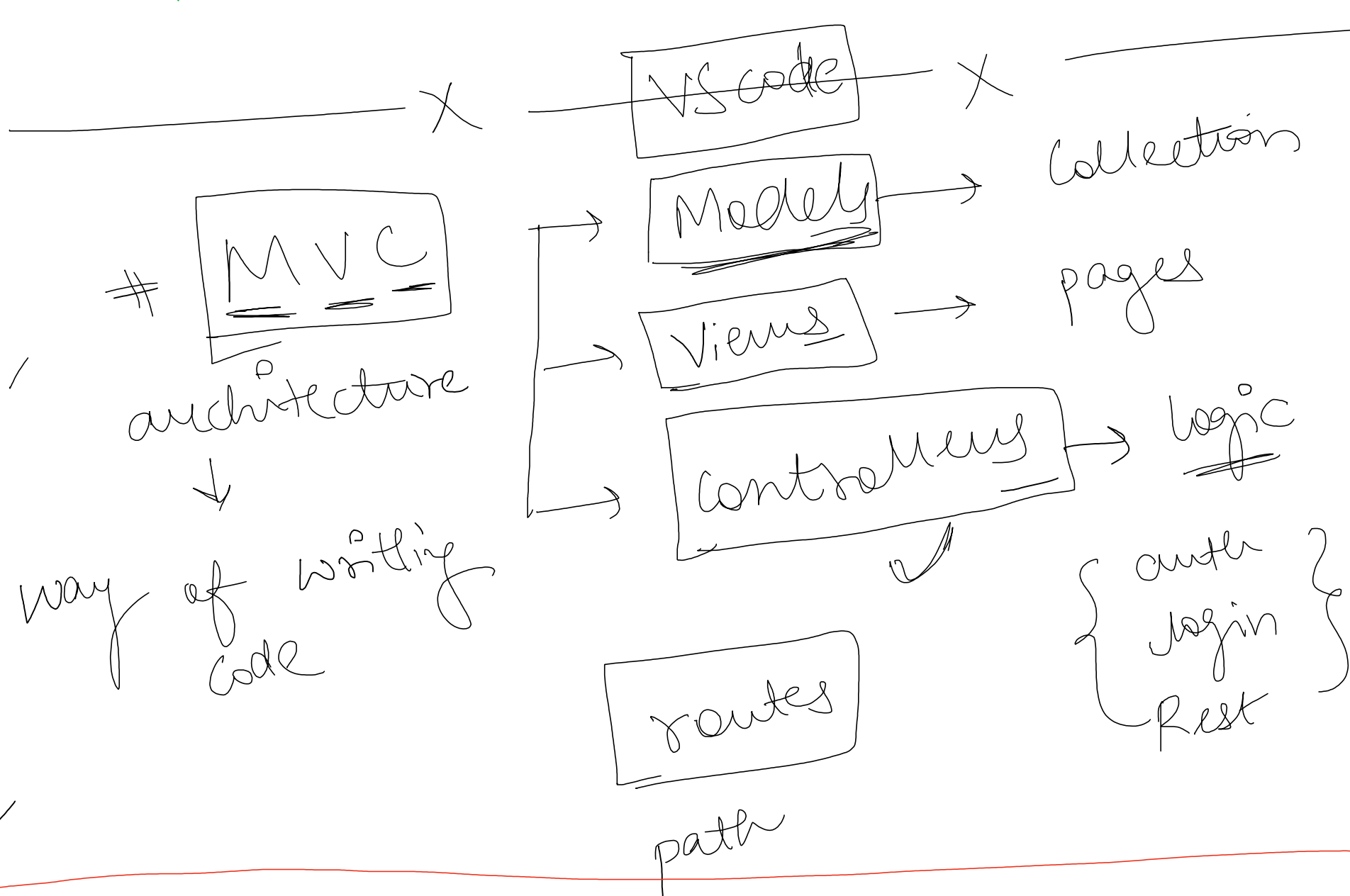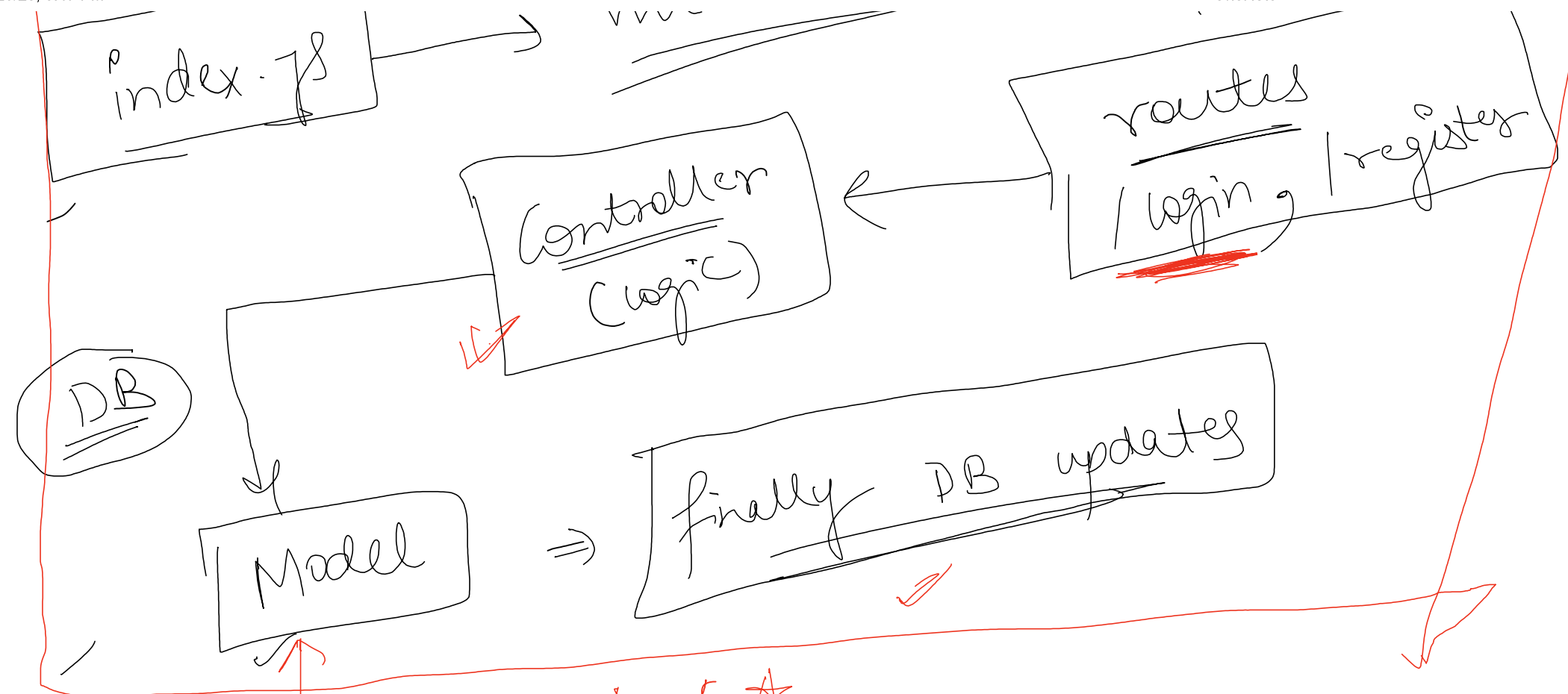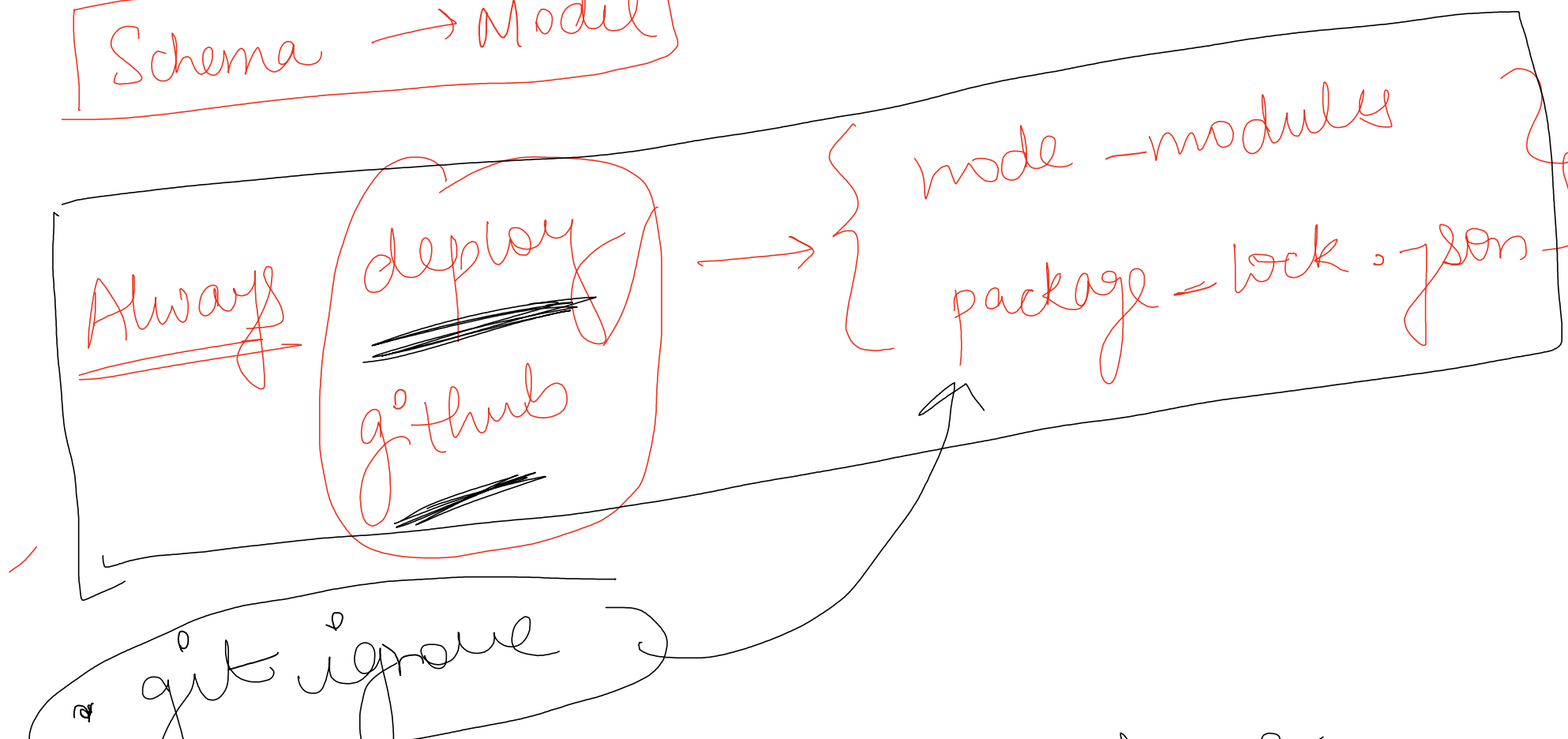
tables = Collection = model = Schema

" Comment . js "

" Comment . model . js "   (Readability)

"model" is a JS class → new ✓
                        → Capital ✓

(T2) Array

Populate → ref .

SQL

n default → GET
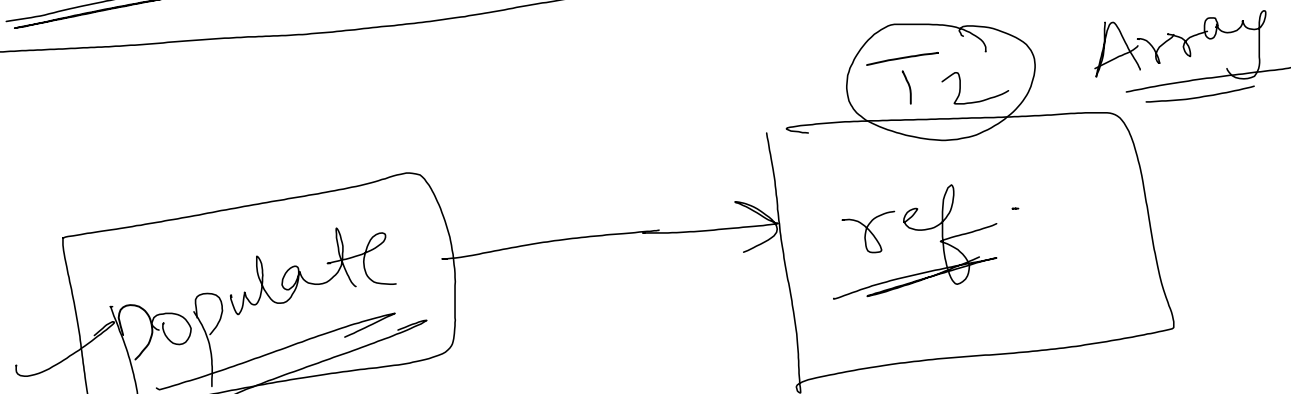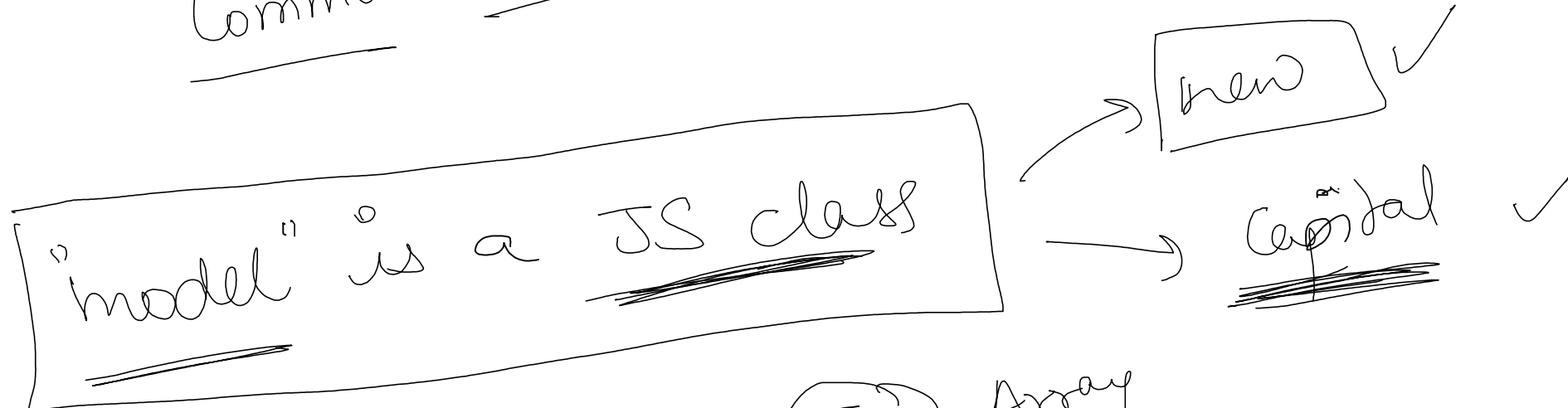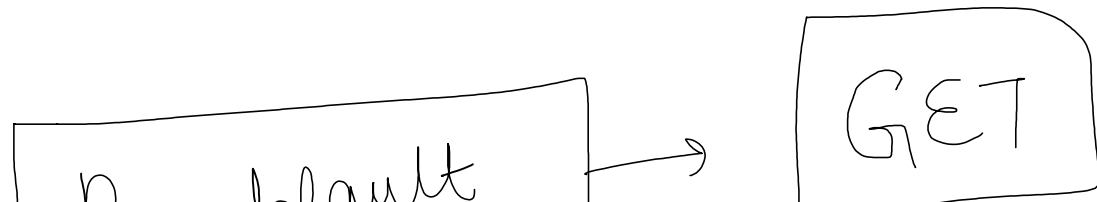
By default

Browser

POST PUT — — .

thunderclient