

Static



dynamic → CFB
↓ JS



JSX → "React"

ejs, pug, jade, django

Angular
Template

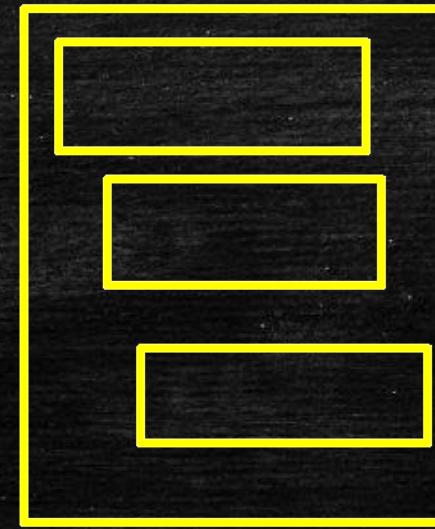
home

L → P ←

dynamis

index

about





~~App.js~~

index.html

HomePage

Comment Code

function App() {

return (

<div><HomePage />

<NavBar />

<Banner />

<Formdata />

</div>

);

}

HomePage

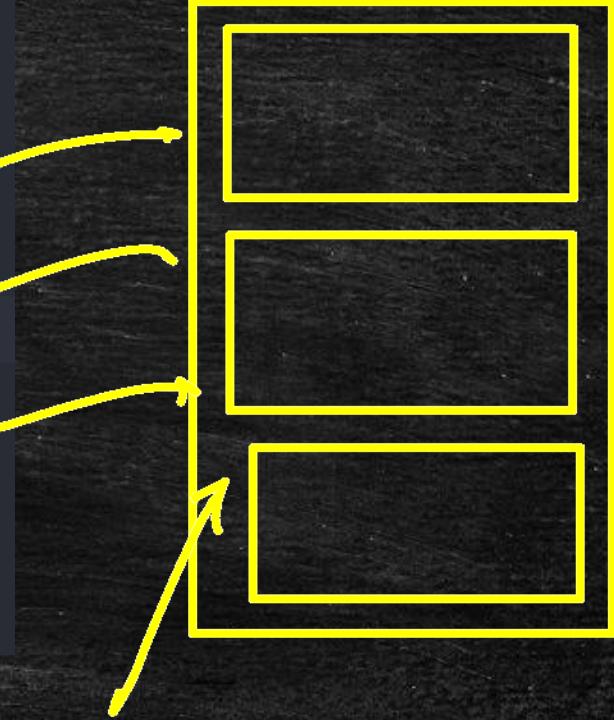
<div><HomePage />

<NavBar />

<Banner />

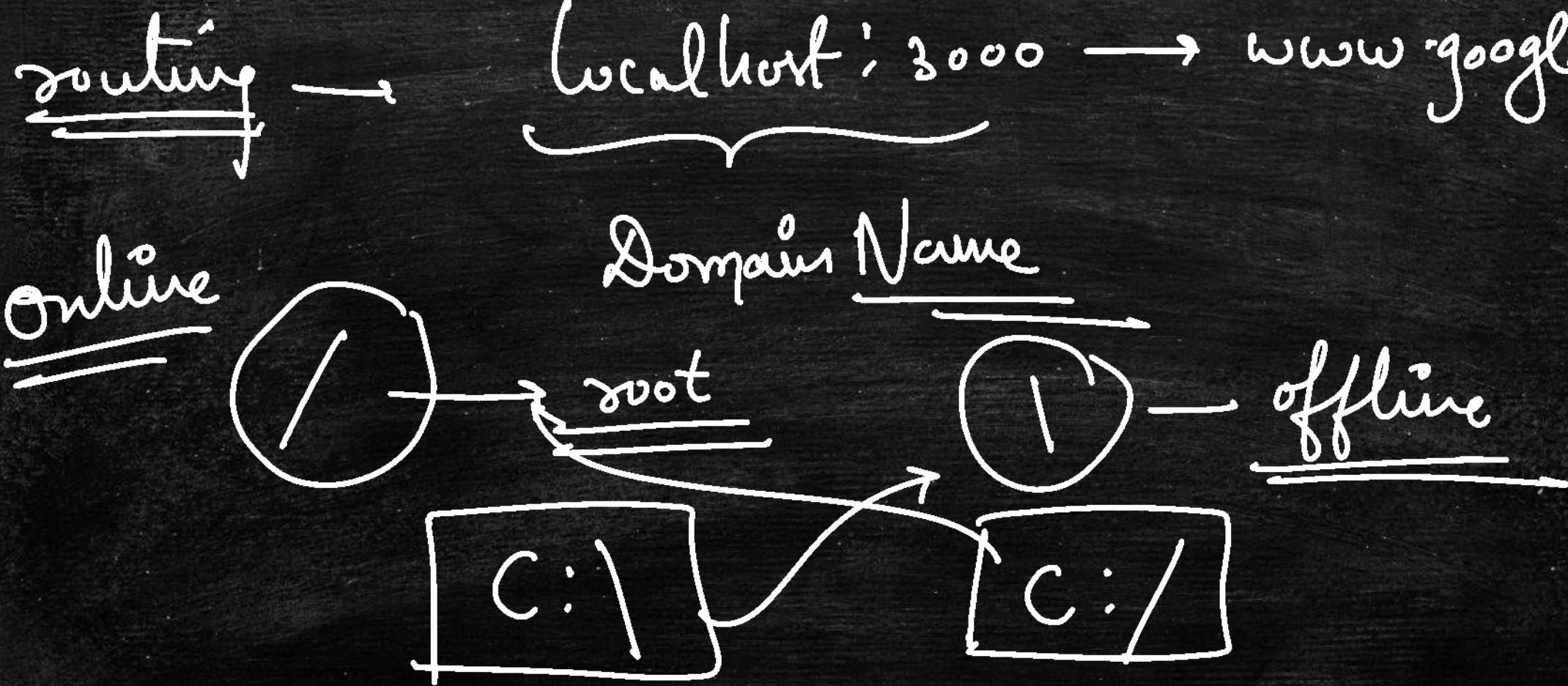
<Formdata />

</div>



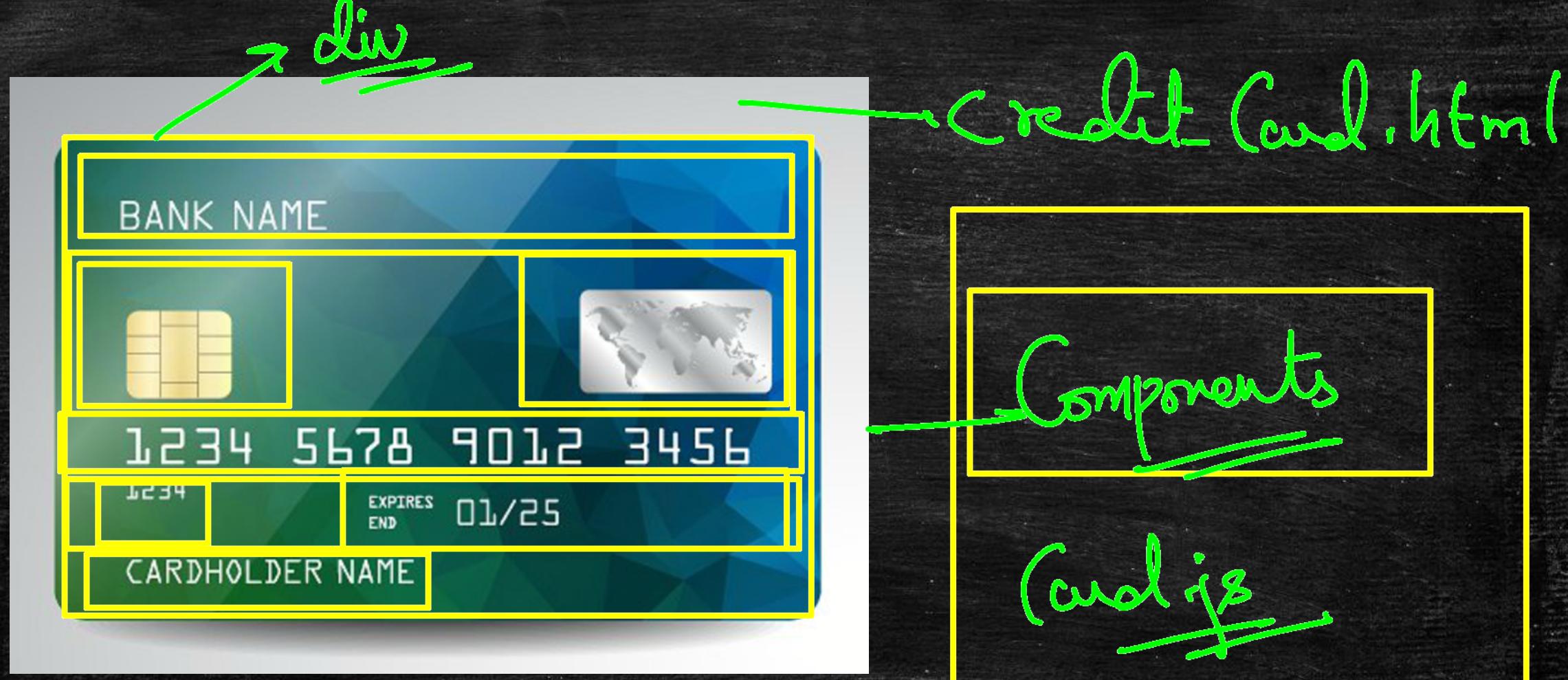
react - router - dom

index.html



C:\ document

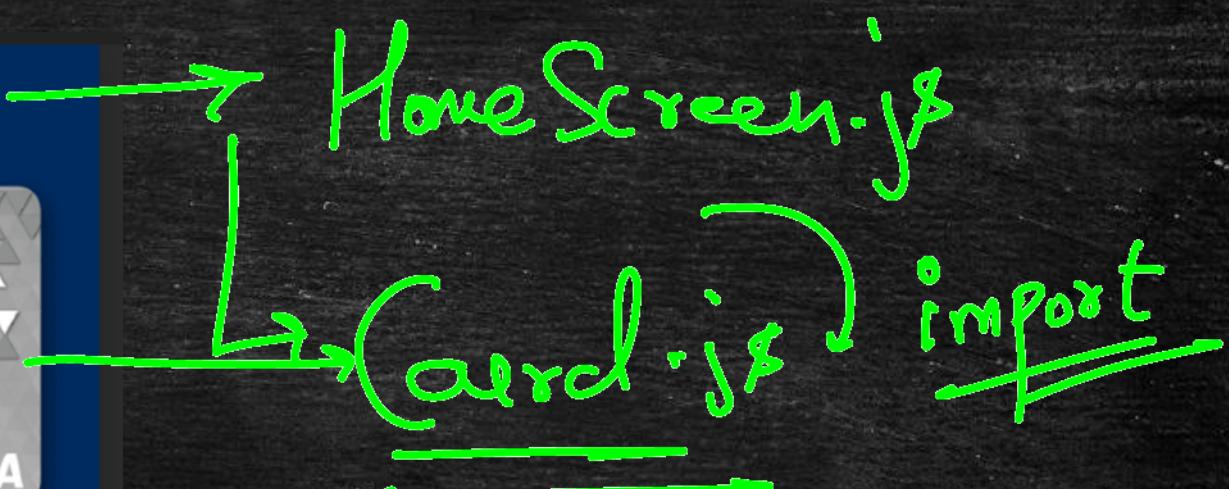
C:\

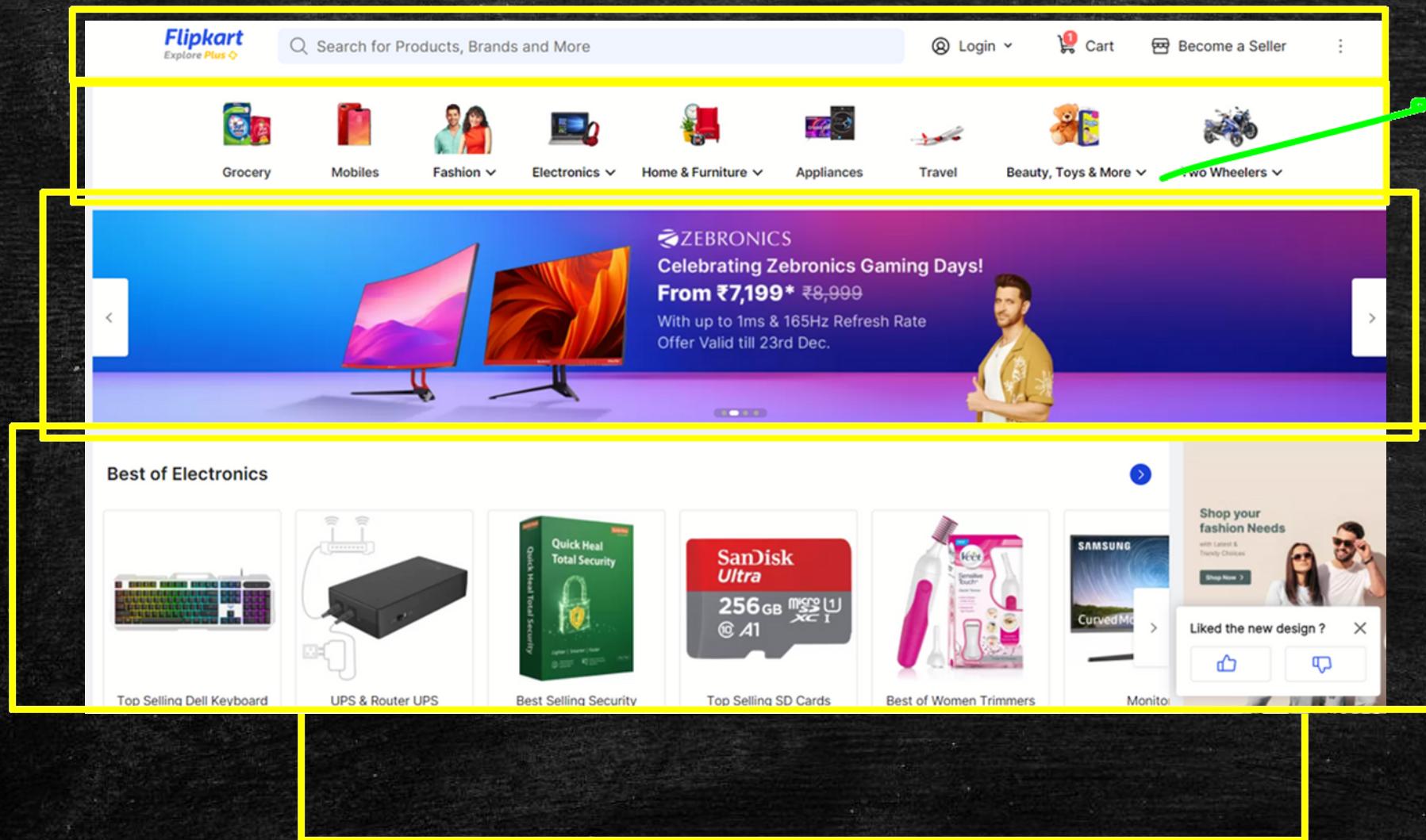


SILVER CREDIT CARD



Check out its **features**,
benefits and **how you can**





Home Screen

NavBar

Category Barriers

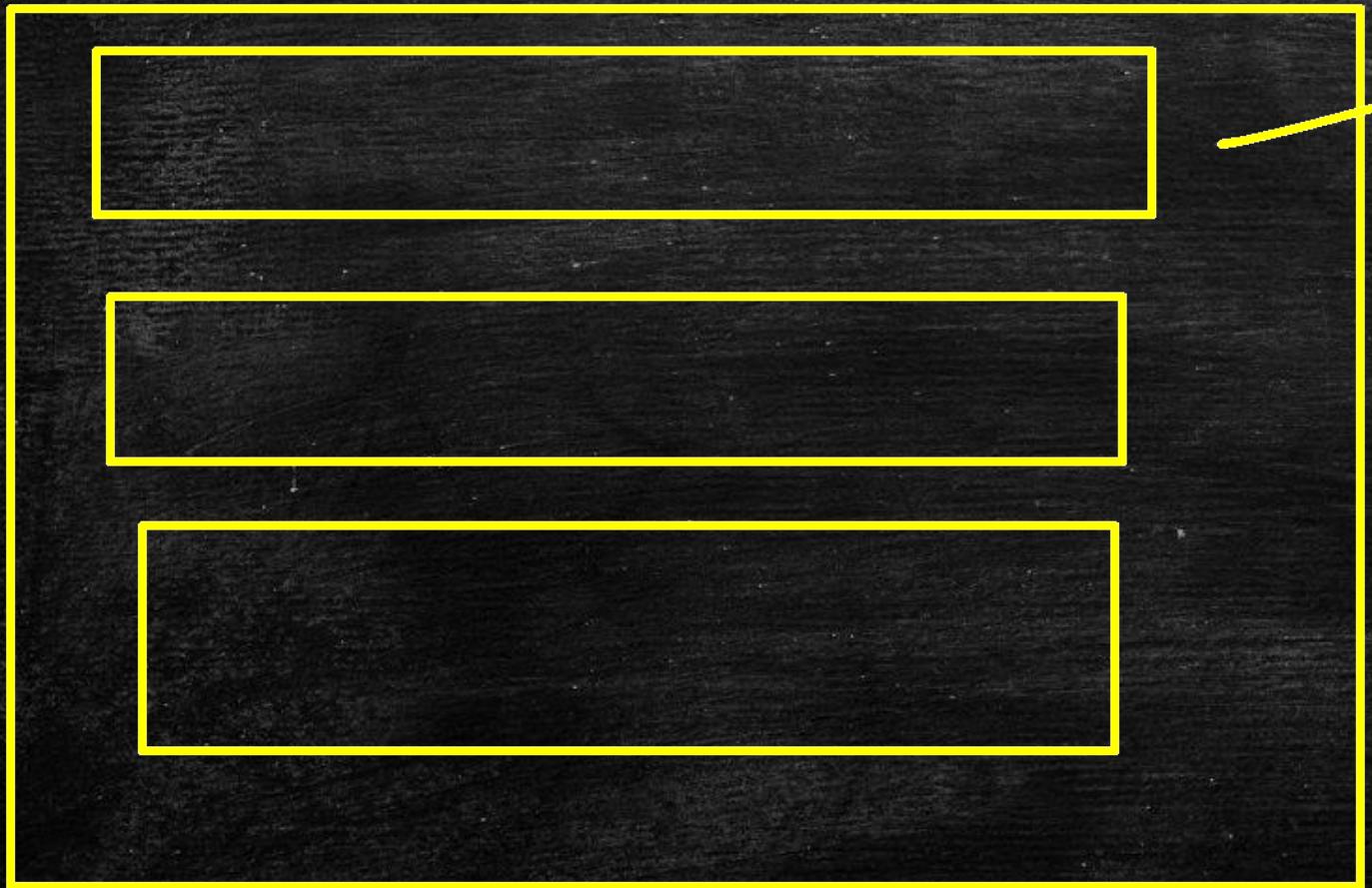
+ Product

4 div → 4 Comp

function HomeScreen() {

< NavBar />
< Banner />

}



only 1 parent
div
↓
App.js

App.js



ContactPage

NavBar
Banner
Prod
Footer

NavBar
LoginForm
Footer



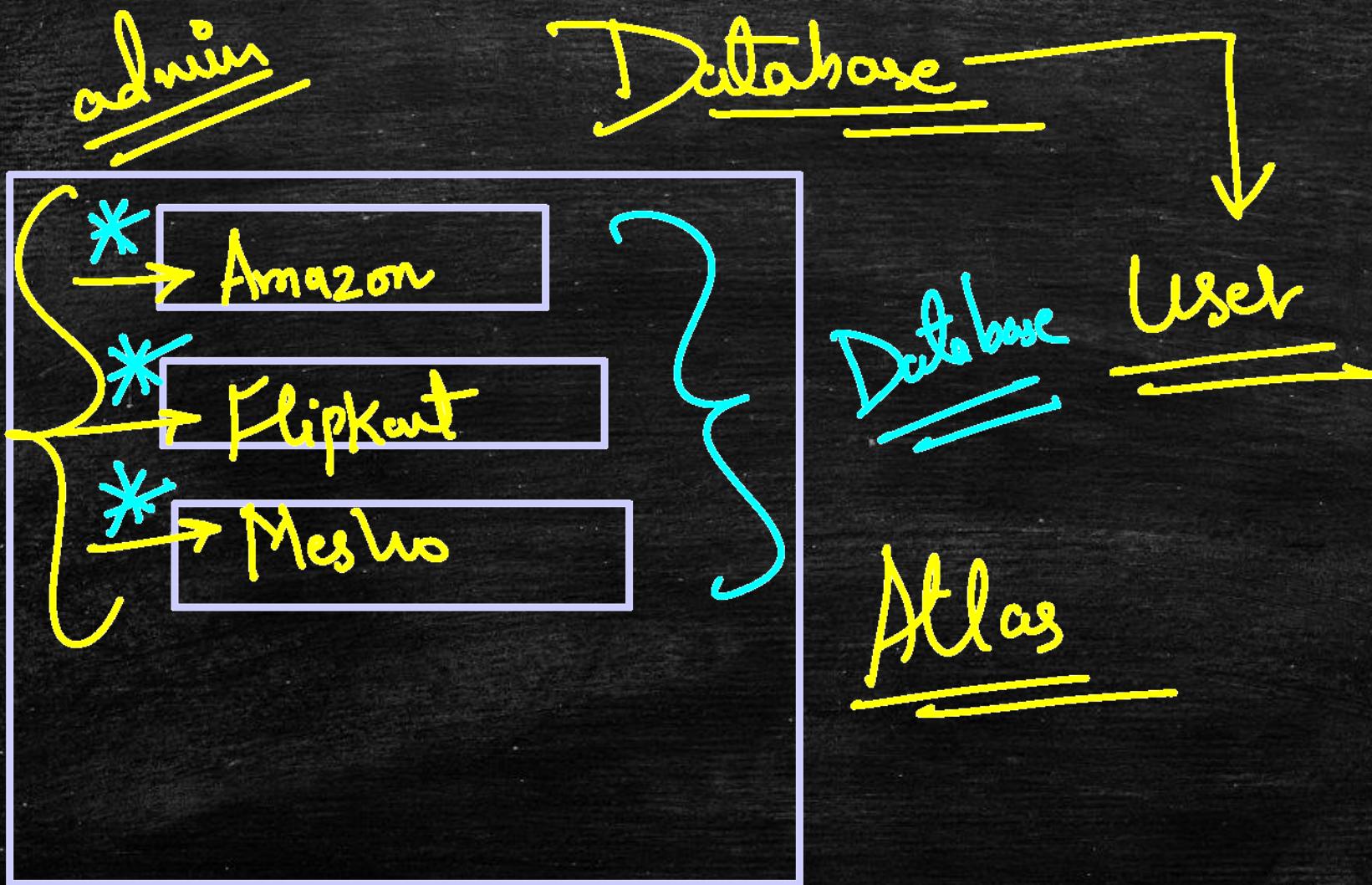
/ → root page

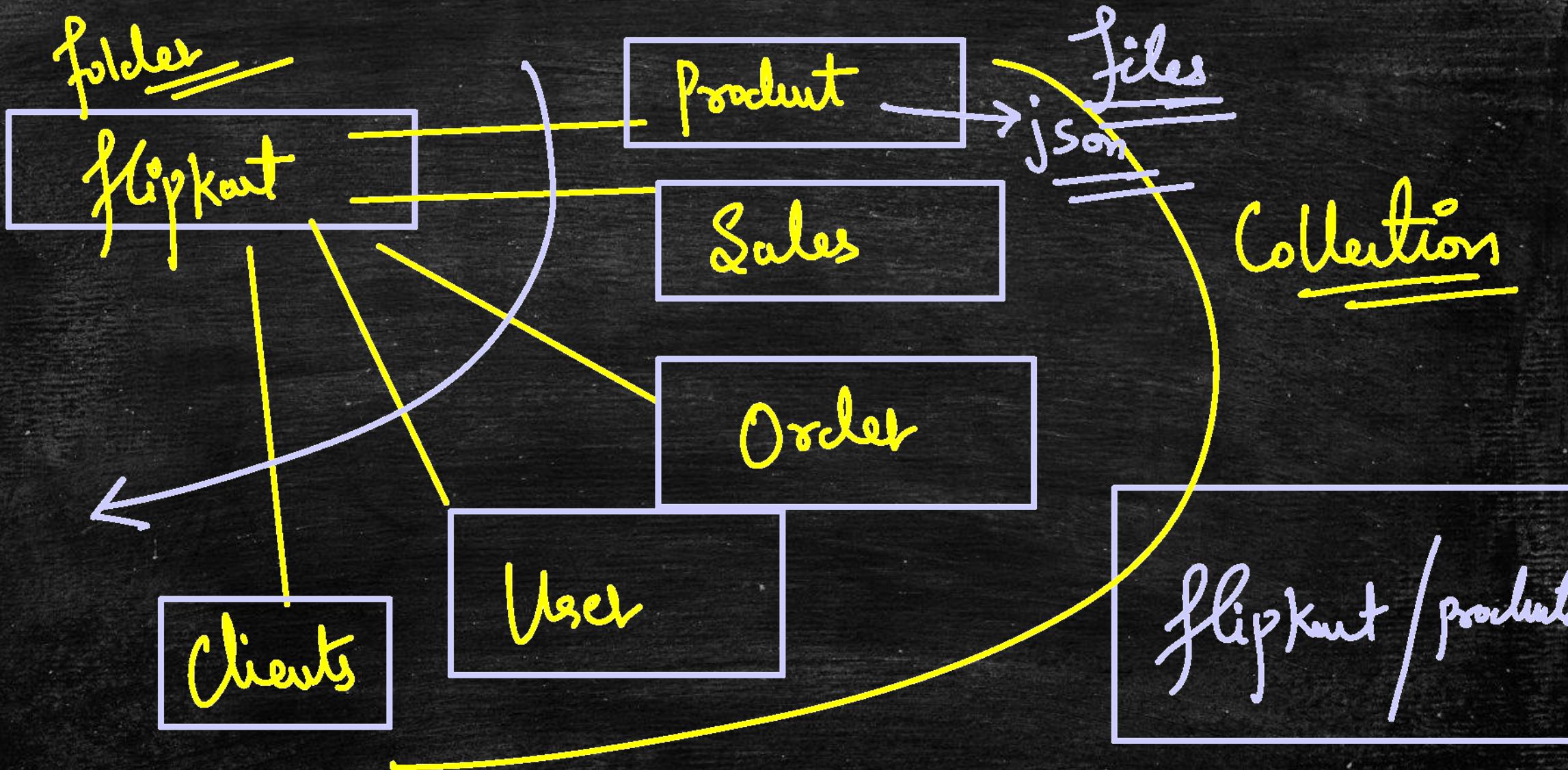
/product → product page

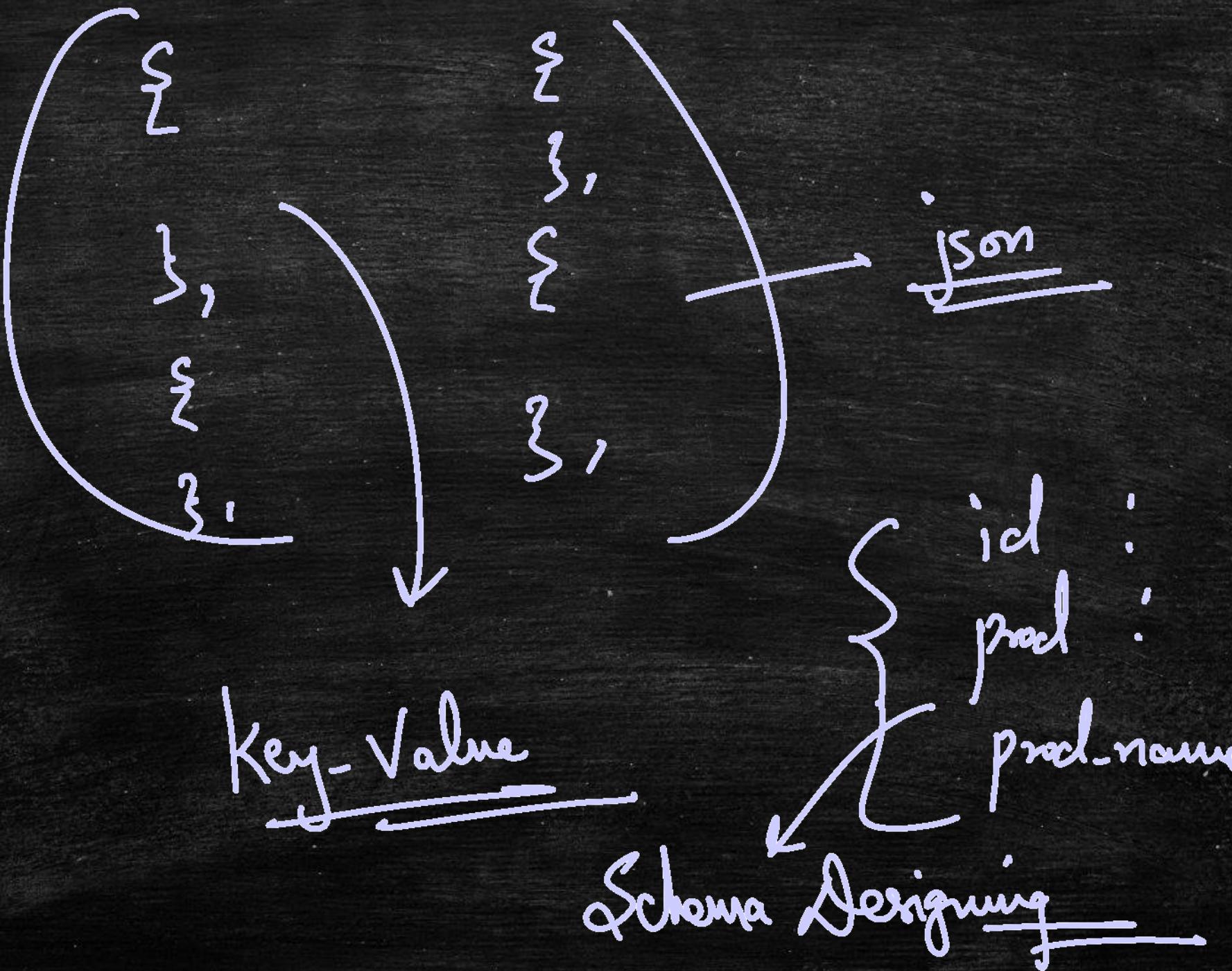
/contact → contact

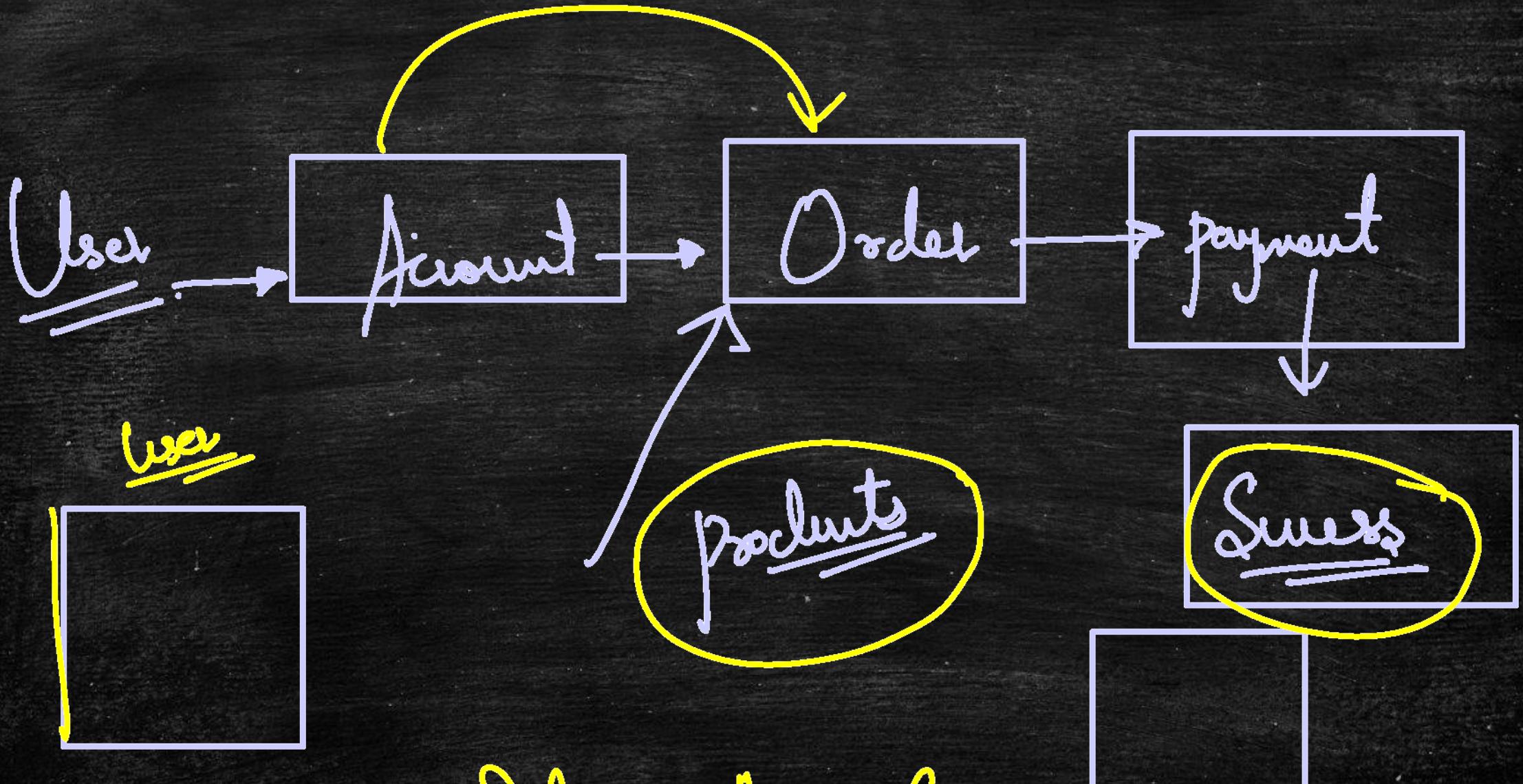
/about → about

* npm i react-router-dom





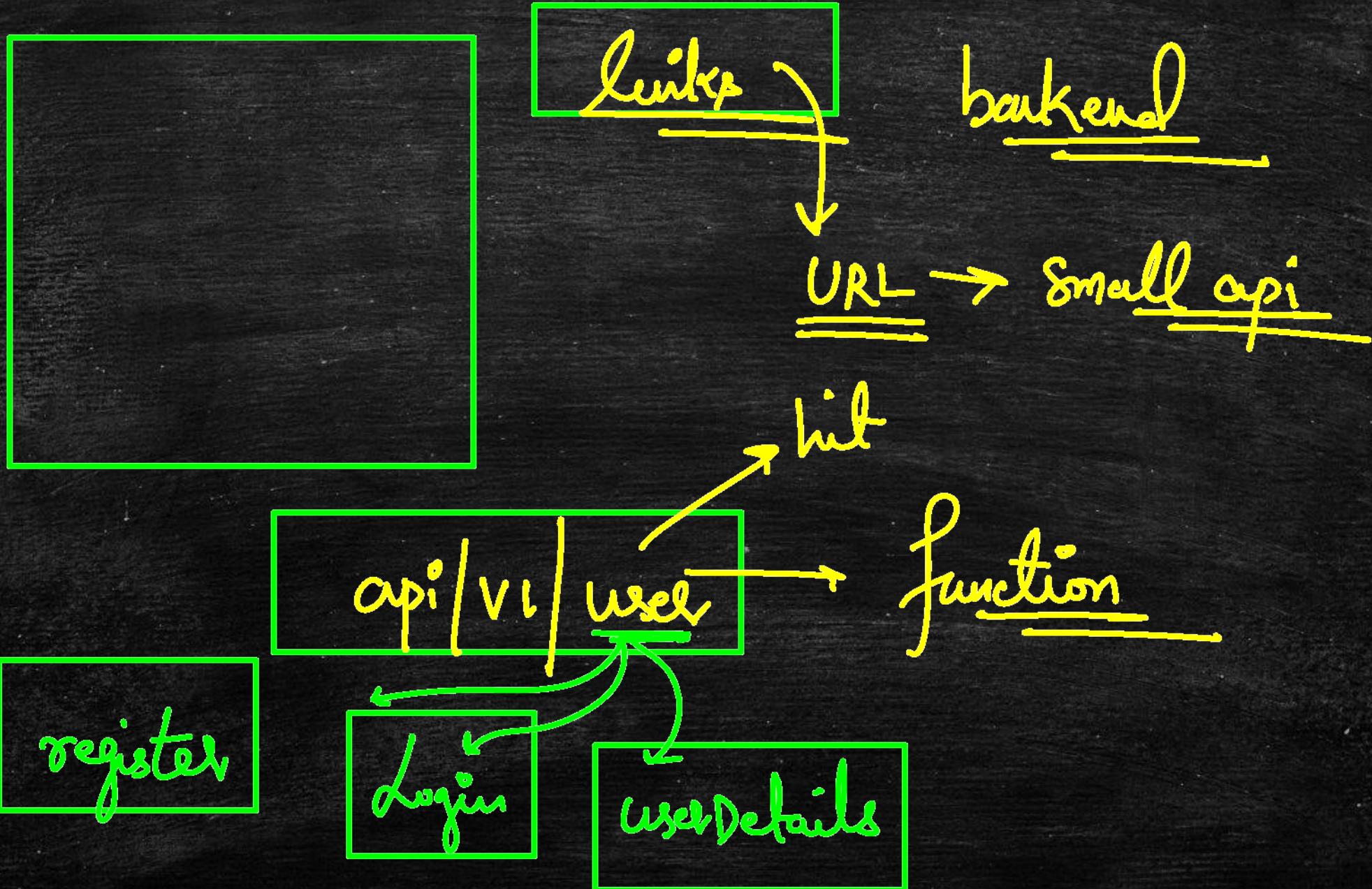




Data - Acquisitions

→ Connection

→ Model — Schema Designing



Intermediate



middlewares

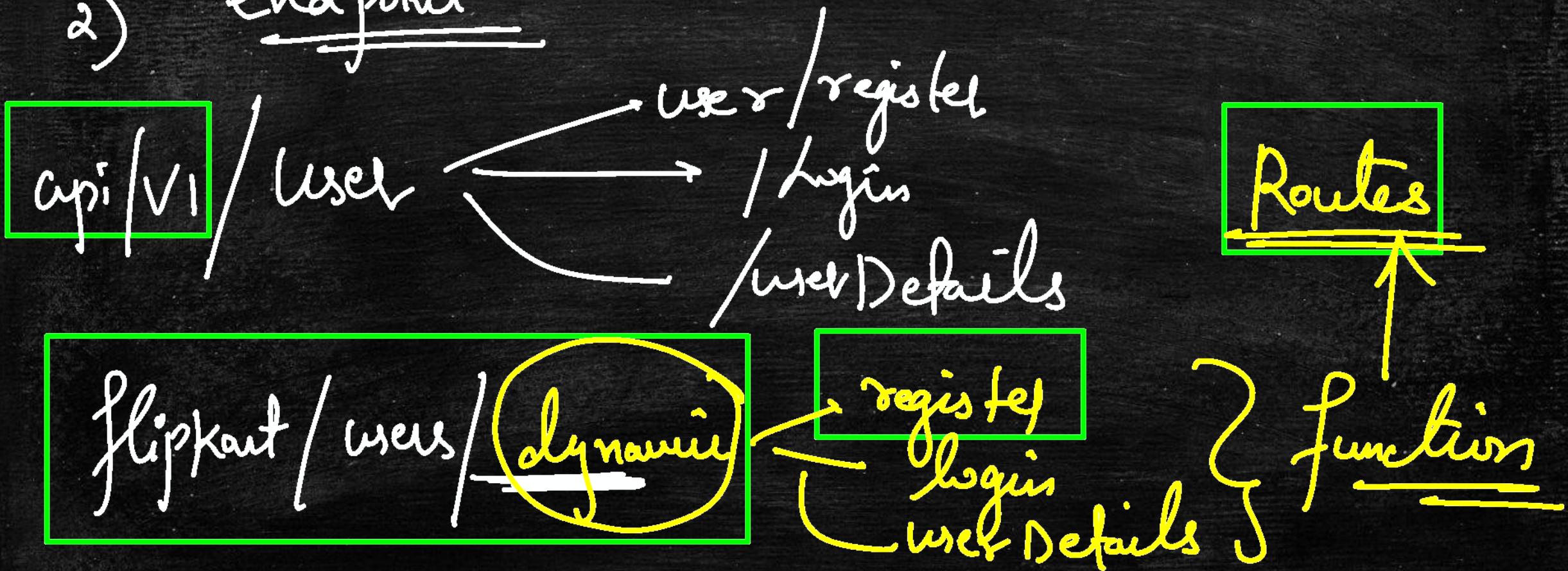


User koi exist karta hai ?
↓
Database — final → user

1) Server → Create

URLS → active

2) end point



URL



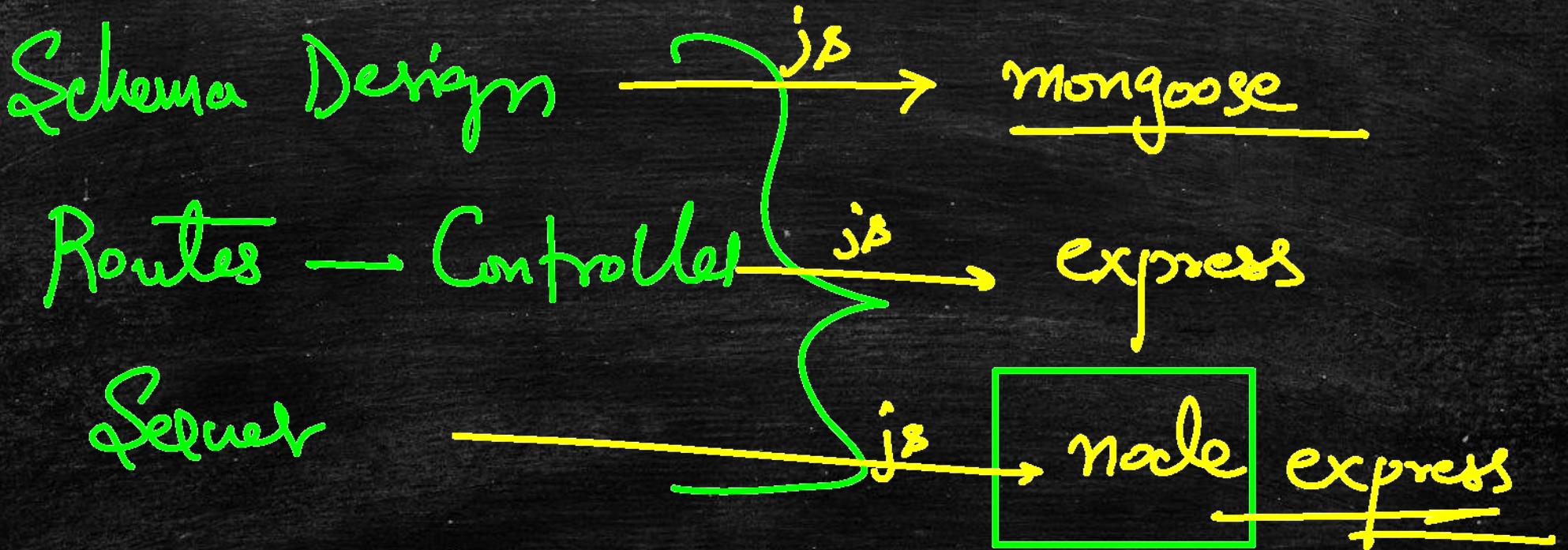
"backend"

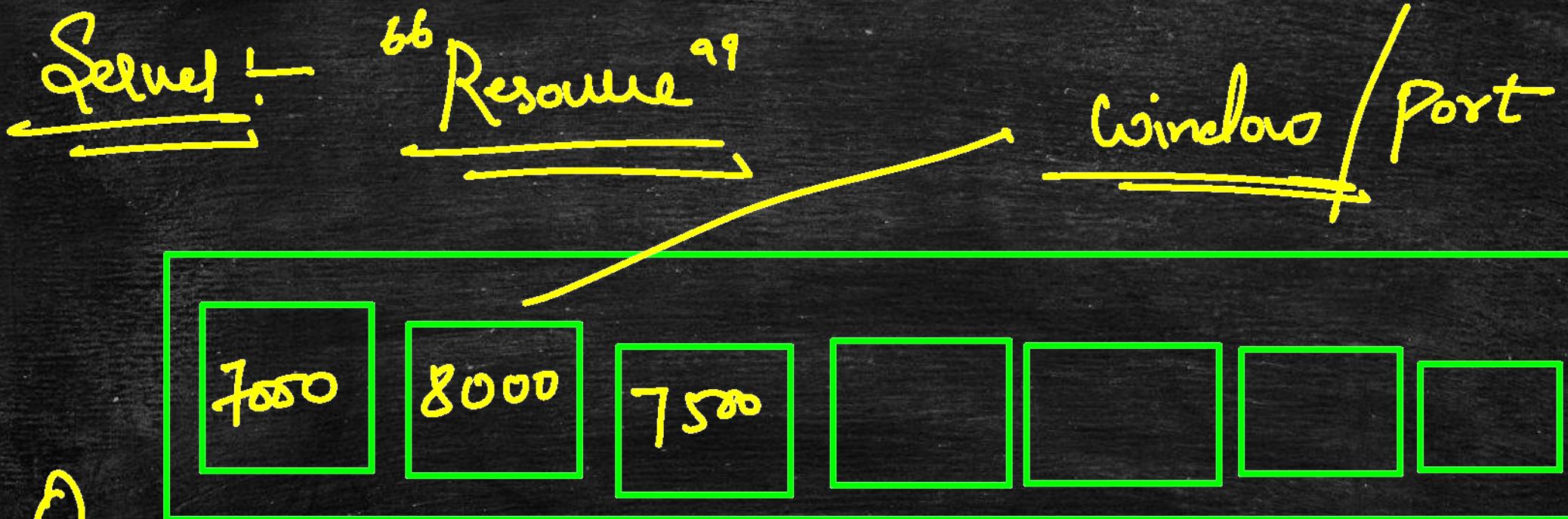


Controller

UserController

4) Models → Schema Design

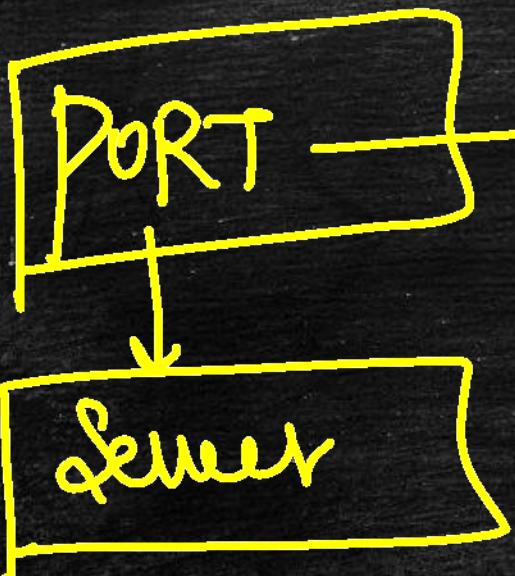




0 →

6080 | 7500 1-5 Windows
 3012 |
 3500 5000 6000

Phase



platform

A'

10

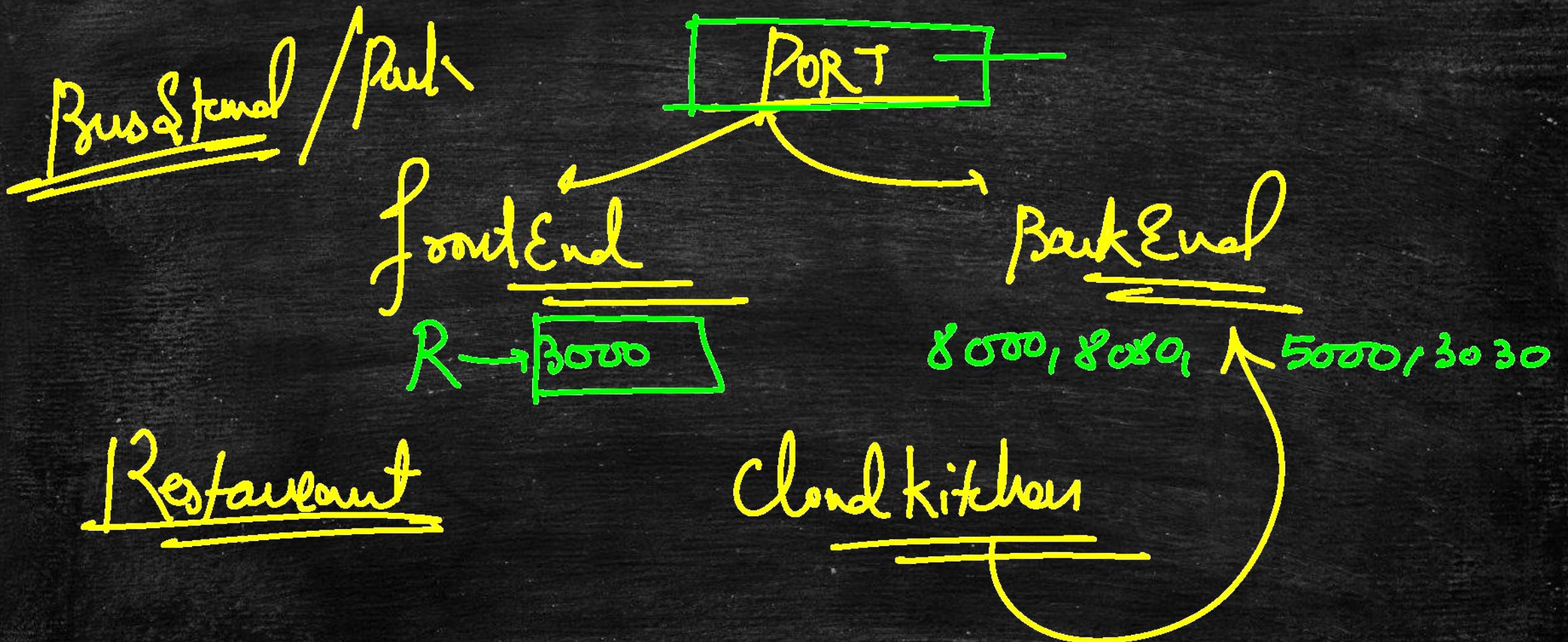
Ab

24 07

Brakes

phone

33 06
33 07



1)

npm init -y

2)

npm i express

nodemon

morgan

mongoose

eslint

dotenv

color

cors

bcryptjs

Axios

jsonwebtoken

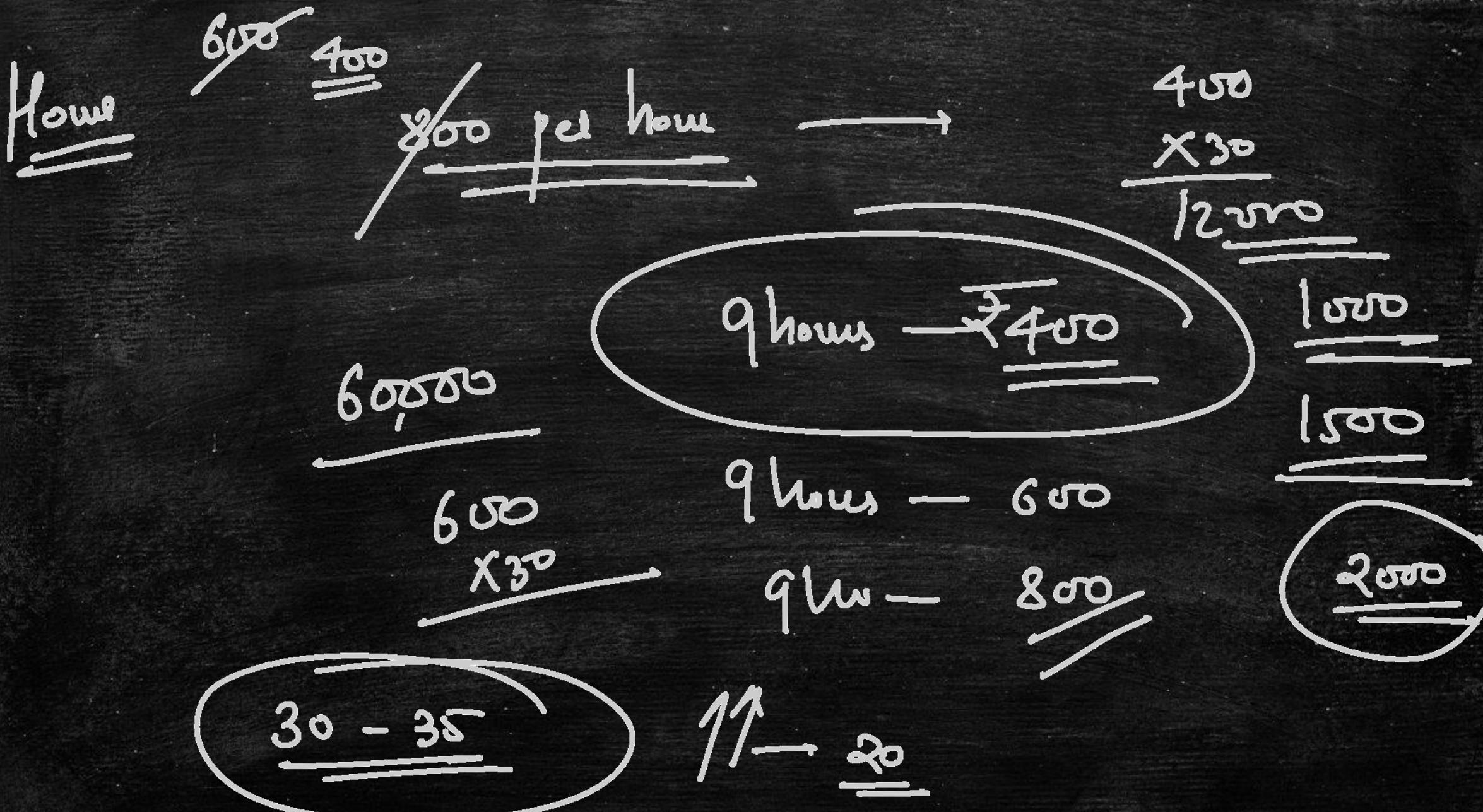
moment

Concurrently

3

13

≈ 20



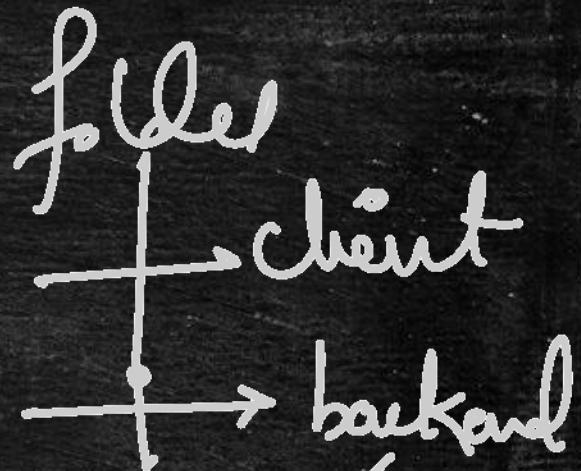
1) Appname → client

2) cd .. → main folder

3) mkdir backend

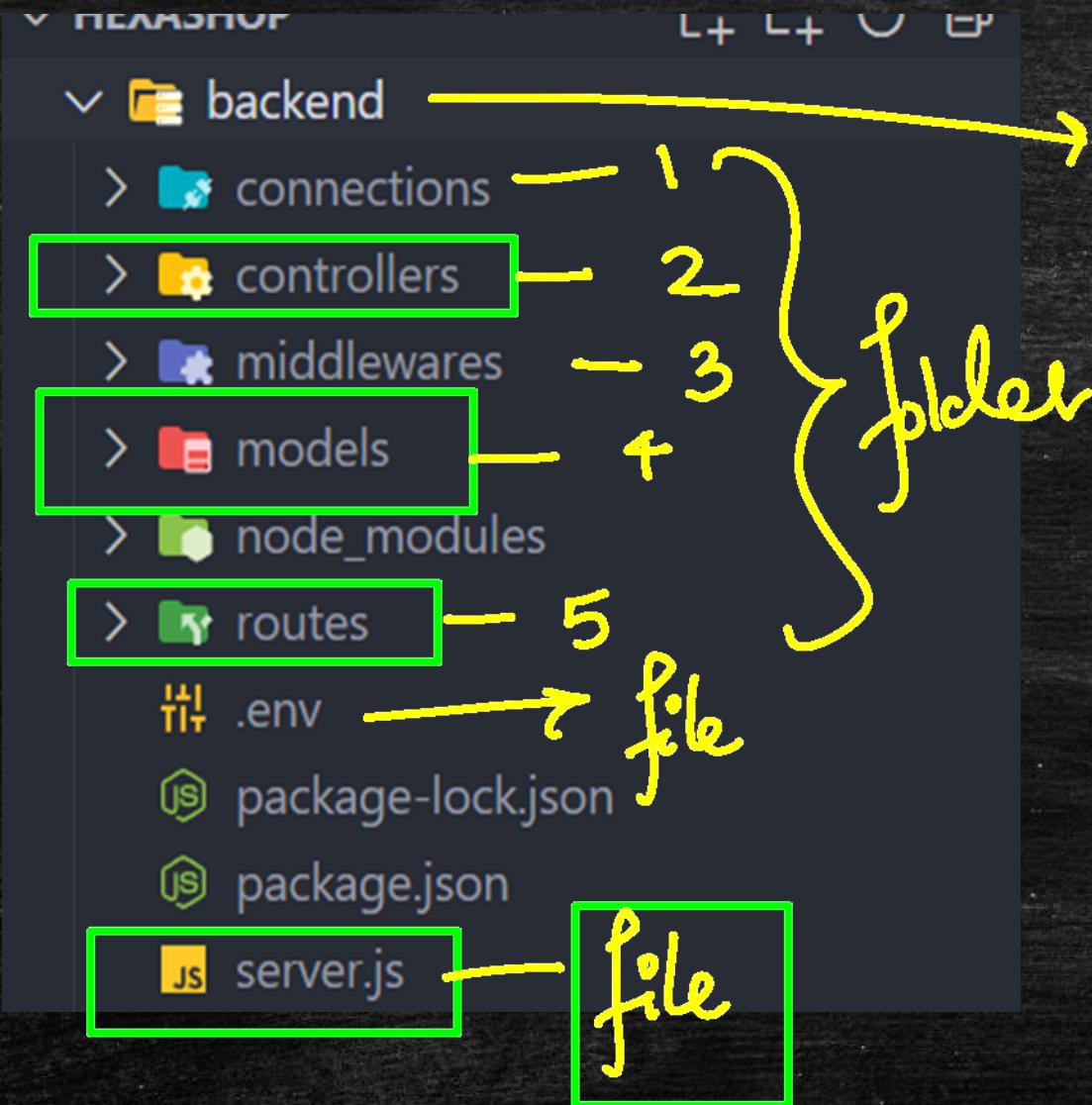
4) cd backend

5) npm init -y



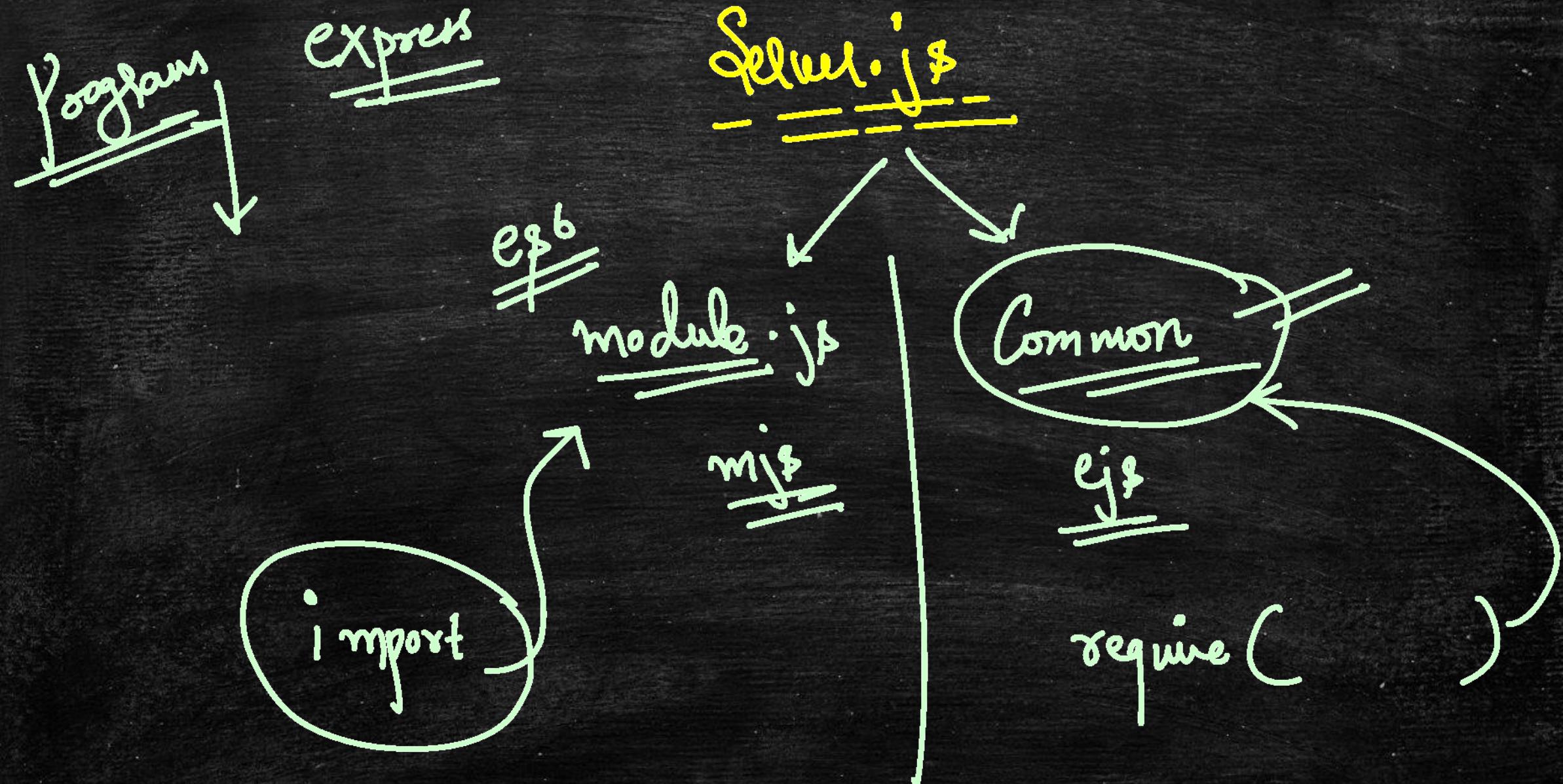
1) npm i -g express

2) npm i express-generator + folders



folder Structure

1) Connections



`import {React} from "React";` → mjs

`const React = require("react");` → ejs

* `import express from 'express'` → mjs

`const express = require("express");` → ejs

```
! CLICK HERE TO ASK BLOCKBOX TO HELP YOU CODE FASTER  
const express = require("express");  
const colors = require("colors");  
const morgan = require("morgan");  
const cors = require("cors");  
const dotenv = require("dotenv");
```

```
const connectDB = require("./connections/db");
```

Connect

```
const app = express();  
dotenv.config();
```

fun Call

Rest API



Rest

CREATE → post

READ → get

UPDATE → put / patch

DELETE → delete

```
//rest api  
Comment Code  
app.get("/", (req, res) => {  
  res.send({  
    message: "welcome to express",  
  });  
});
```

Syntax:

app.get

app.post

use this API
read from database
responses.

use this API
create item

in database

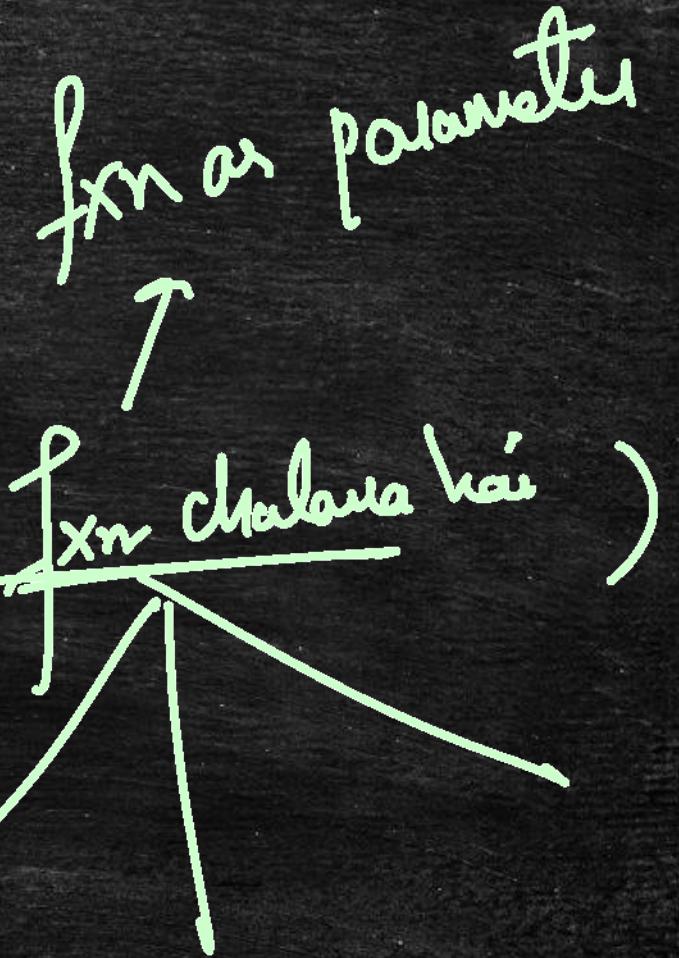
request.

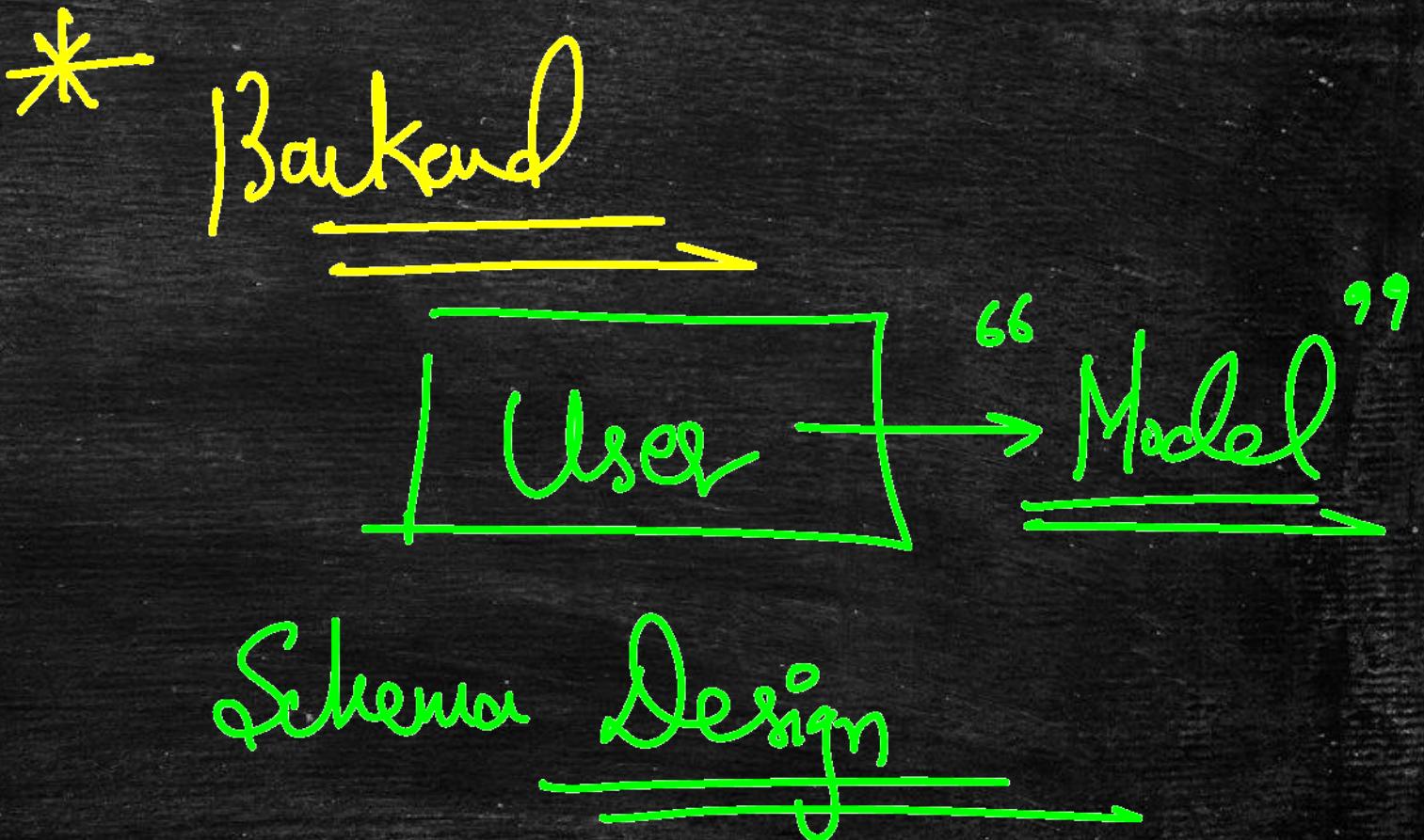
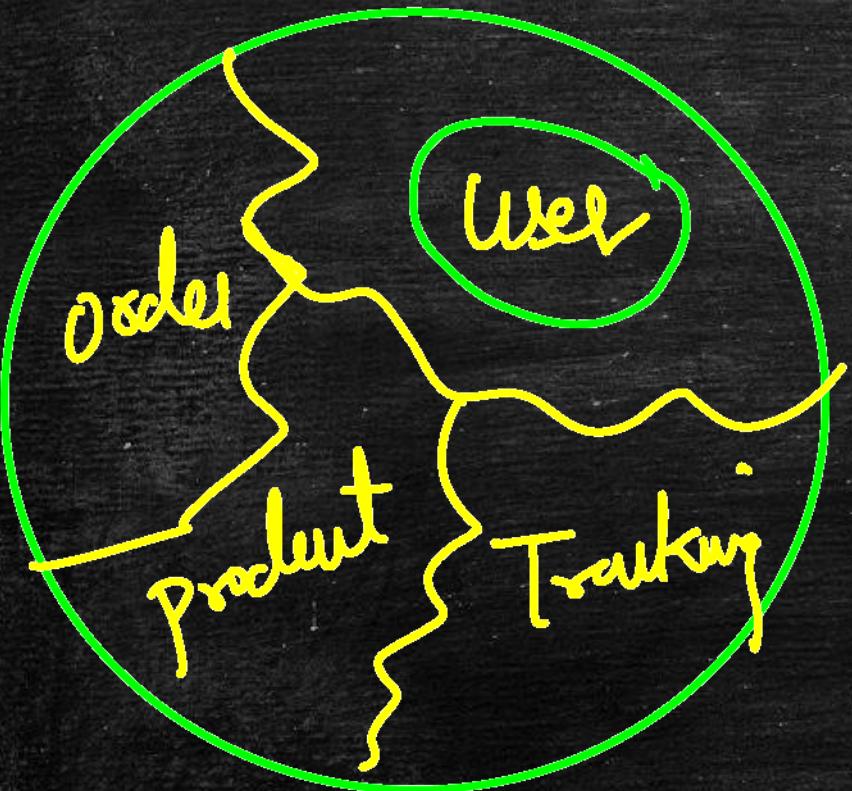
```
//rest api  
Comment Code  
app.get("/", (req, res) => {  
  res.send({  
    message: "welcome to express",  
  });  
});
```

app.get(" / ")

folder

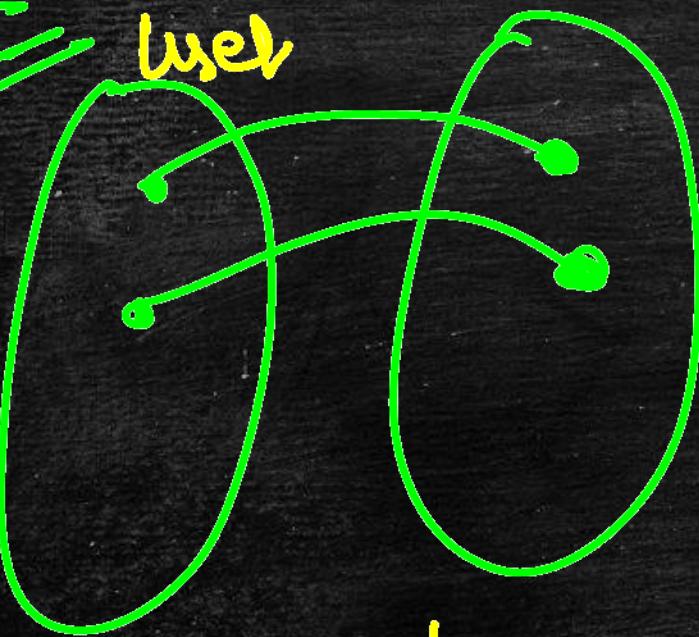
Router
Controllers





Connections, Sequelize, .env ✓
 Model → web Model / web Schema

66 EP 99



useSchema →

-id:

name

email

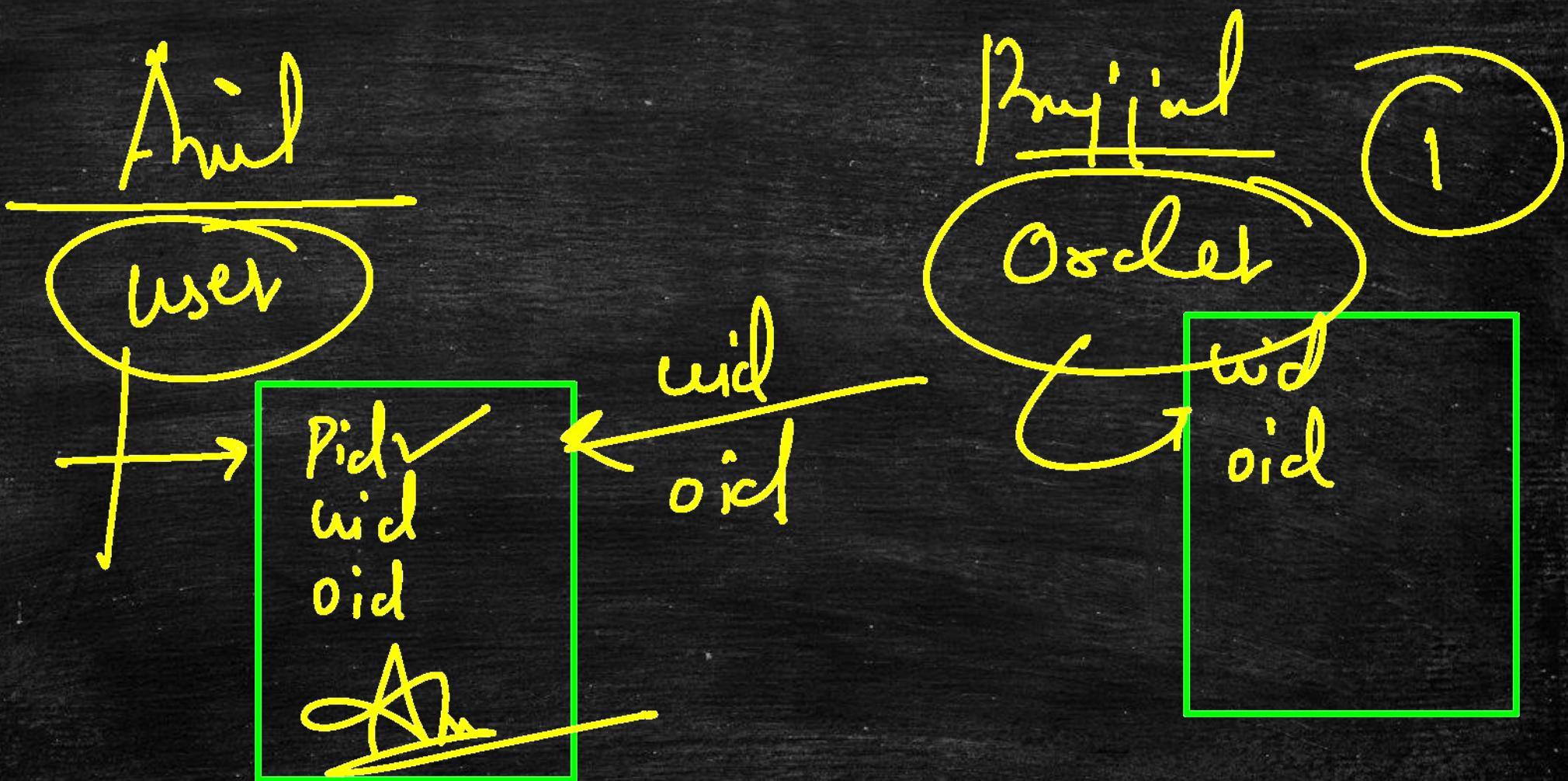
Pwwood

Post

Order

wishlist

Abaus: — user — id



```
const UserSchema = new Mongoose.Schema({  
    name: { type: String },  
    email:  
    pass  
})
```