REACT & NEXT.JS WORKSHOP DAY 1️⃣

SURVIVE, LEARN, AND BUILD SOMETHING COOL. 😎

JOIN OUR SOCIAL MEDIA !!!

ITCommunity

simitclub

SIM Information Technology Club

Guildlines / Notes





SIM PARTNER

DEGREE YEAR 2 - UNIVERSITY OF LONDON
COMPUTER SCIENCE (AI & ML)

MOE PYE SONE

Instagram
@moecrosoft_

LinkedIn
@moecrosoft

Github
@moecrosoft



SIM PARTNER

DEGREE YEAR 3 - UNIVERSITY OF WOLLONGONG
COMPUTER SCIENCE (AI & BIG DATA)

ALEX VUN

Instagram
@alexvun319

LinkedIn
@alexvun

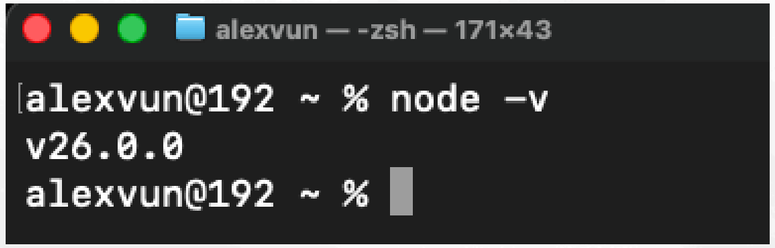
Github
@AlexBearBear0319

PRE-CLASS SETUP

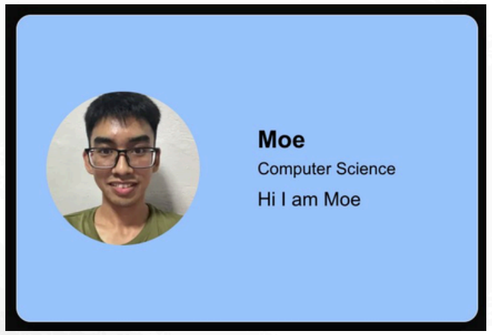
Please ensure you have prepared:

Node.js installed? (The engine)

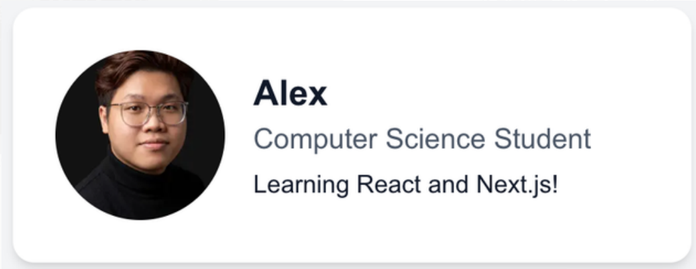
VS Code ready? (The canvas)



Today's Mission: The Personal Profile Card 🪪



- **Forget boring theory. We are building today.**
- **A custom routing system (Home vs. About Me).**
- **Goal: Code your own interactive Student/Personal Profile Card.**
- **We'll learn React and Next.js while we build it.**



LESSON OUTLINE

1. Pre-class setup
2. Introduction to React & Next.js
3. Project Setup
4. File Structure
5. Navigation / Routing
6. Components
7. Variables (Props)
8. Tailwind CSS
9. Hands-on Challenge #00 to #04
10. Wrap Up & Summary

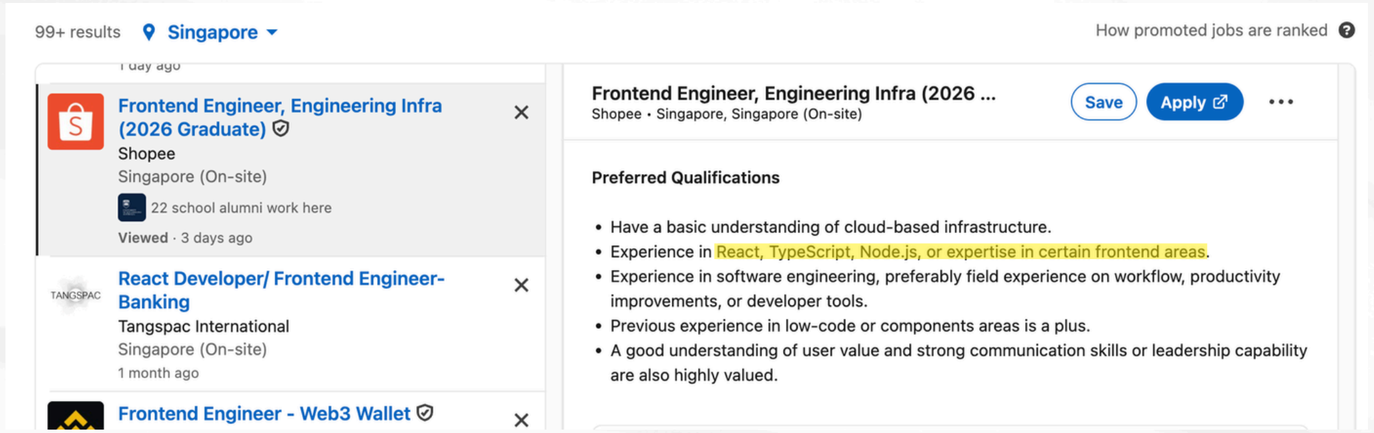
INTRODUCTION

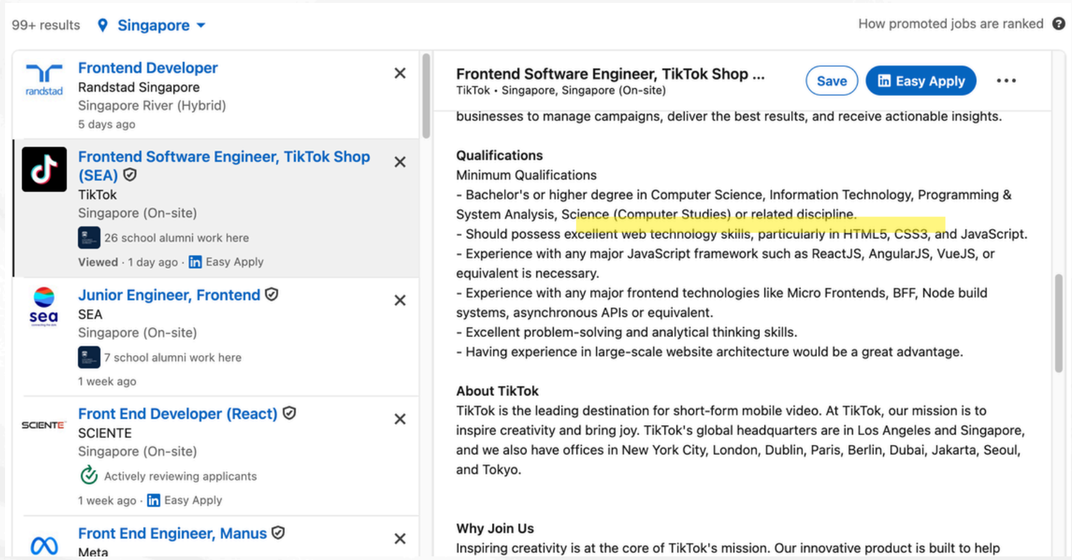
🤔 WHY REACT? WHY NEXT.JS?

Why Are We Even Learning This? 💸

Why can’t we just learn HTML?

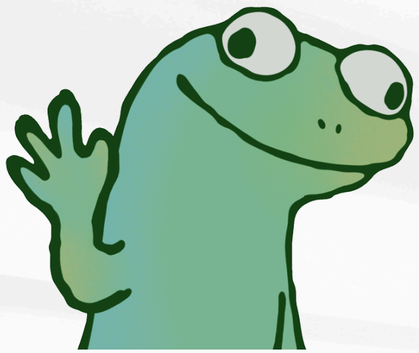
- React & Next.js are the most demanded frontend skills globally.
- Top companies (Shopee, Grab, TikTok, Netflix) actually use them.
- Benefits: High employability, faster building and fewer headaches.





The Pain of Vanilla HTML 😩

Imagine building 50 student cards in raw HTML.
Copy + Paste = Nightmare.
And what if you need to change the logo in all 50 pages?
You have to select all the logos and change the image tag manually one by one?



```html
<div class="card">
  <img src="/old-sim-logo.png" class="logo" />
  <h2>Priya</h2>
  <p>Business</p>
</div>

<div class="card">
  <img src="/old-sim-logo.png" class="logo" />
  <h2>Jun Wei</h2>
  <p>IT</p>
</div>

<div class="card">
  <img src="/old-sim-logo.png" class="logo" />
  <h2>Ahmad</h2>
  <p>Computer Science</p>
</div>
```

need to do ctrl-c and ctrl-v the nav_bar to every page “icon”

script “too much work to do + once need maintanence need to multiple ways

So, does any method make the work easier?
No Ctrl C + Ctrl V?



Solution

Only update 1 component file, done. XD

```
<StudentCard name="Priya" course="Business" />
<StudentCard name="Jun Wei" course="IT" />
<StudentCard name="Ahmad" course="Computer Science" />
```

React

StudentCard.js

```
export default function StudentCard({ name, course }) {
  return (
    <div className="card">
      {/* Change the logo HERE ONCE. It updates everywhere! ✨ */}
      <img src="/new-sim-logo.png" className="logo" />

      <h2>{name}</h2>
      <p>{course}</p>
    </div>
  );
}
```



What’s React and Next.js and WHY do we use them?

- React: A JavaScript library for building user interfaces (The LEGO blocks).
- Next.js: A framework built on top of React (The instruction manual + superglue).
- Why? Because writing pure HTML/JS for complex apps is painful.

Extra knowledge!!!

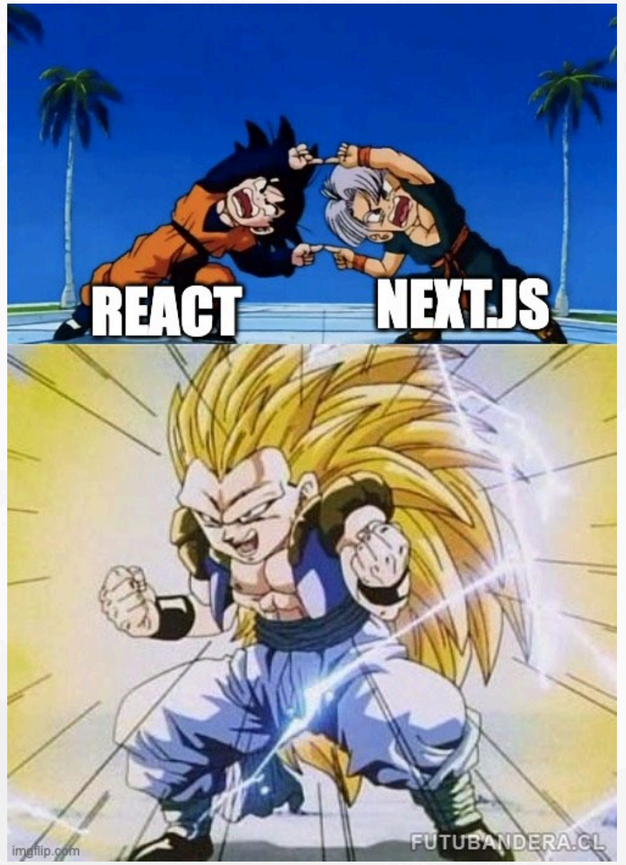


What’s Next.js

React is great! but...

React can ✅	React can’t ❌
Build UI components	Handle page routing
Reuse across pages	Fetch data cleanly
Update efficiently	Optimise for Google

So, with Next.js = React with superpowers



Library vs Framework

Library *(React)* = **Utensils** 🍴

- ✓ A tool you pick up whenever you need it.
- ✓ Such as a knife to cut vegetables.
- ✓ You decide when and how to use it.
- ✓ You are the chef. The library is your tool.



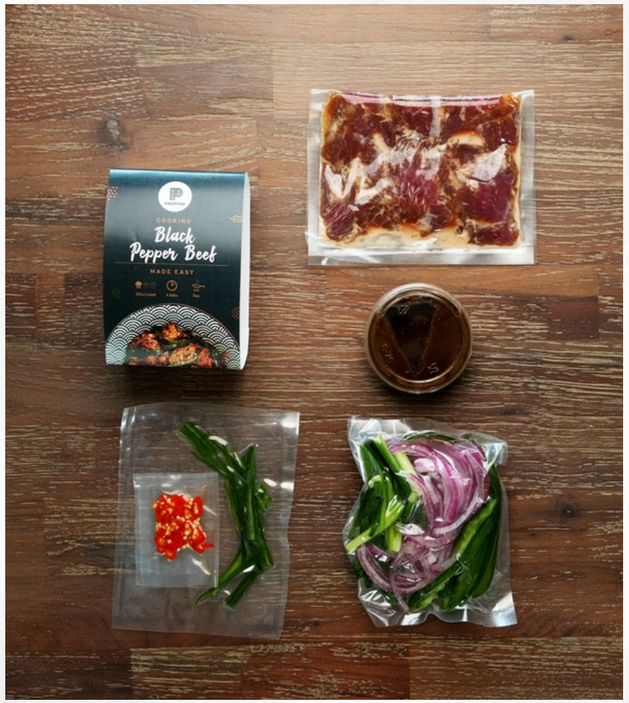


Library vs Framework

Framework *(Next.js)* = **Recipe/ Meal Kits** 🍱

- ✓ A recipe with steps you need to follow.
- ✓ You don’t decide the order, the recipe does.
- ✓ You just have to fill in the steps.
- ✓ It gives you structure.

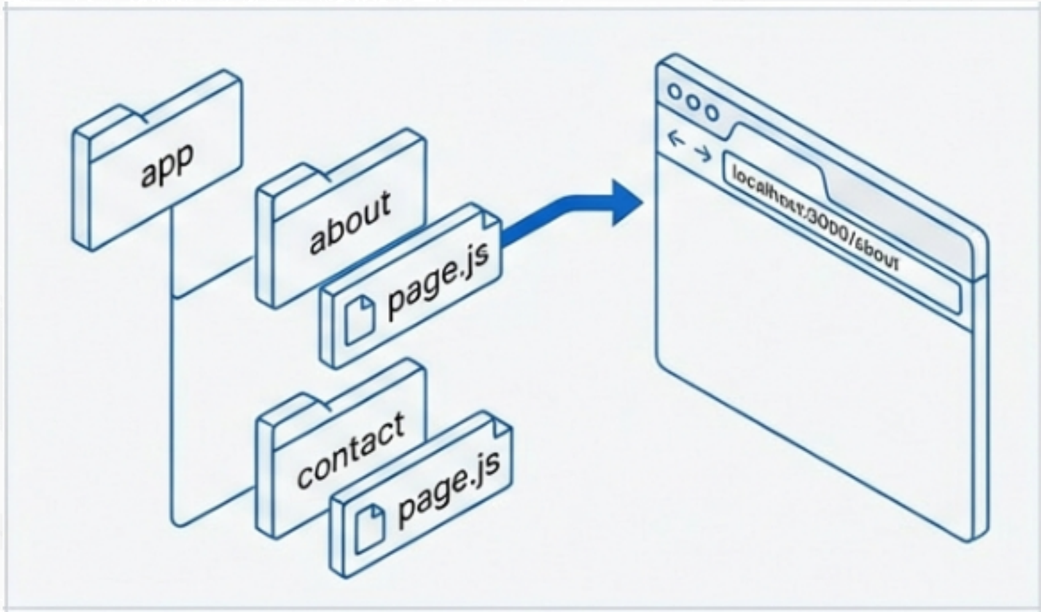




Why Next.js?

🗂️ Routing Magic: Folders = Pages

Create a folder → Get a page. That's it.



Next.js File-based Routing (App router)

Routes are automically generated after simply creating a folder and a file.

No config files. No confusing setup. Just folders.

Why Next.js?

🚀 **Next.js Bonus Features**

(You don't need to memorise these today — just know they exist)

- ✦ Full Stack — Frontend + backend in one project
- ✦ SEO Friendly — Google can actually find your site
- ✦ Image Optimisation — Auto-compressed, auto-resized
- ✦ Built-in Performance — Fast by default, not by accident

⚖️ React vs. Next.js — Quick Summary

	React	**Next.js**
What is it?	Library	Framework
Routing	DIY 😅	Auto 🎉
SEO	Extra work	Built-in
Best for	Internal tools	Public websites

🏗️ PROJECT SETUP TIME

LAPTOPS OPEN. HANDS ON KEYBOARDS. THIS IS WHERE WE ACTUALLY START.

CODE ALONG 💻 WITH ME!

YOUR PERSONAL STUDENT/PROFILE CARD!

We will build:

- A reusable card component.
- Beautiful styling (no ugly 90s websites here).

STEPS

1. Open your `terminal (MacOS)/ cmd (Win)`
2. Type: `npx create-next-app@latest my-profile-card`
3. CRITICAL SETTINGS:
   - `TypeScript? NO (Keep it simple today!, we'll just use JSX)`
   - `Tailwind CSS? YES`
   - `App Router? YES`

CONFIGURE SETTINGS

```
alexvun@192 ~ % npx create-next-app@latest my-profile-card
npm notice run npx
npm notice run 'create-next-app' my-profile-card
? Would you like to use the recommended Next.js defaults? › - Use arrow-keys. Return to submit.
    Yes, use recommended defaults
    No, reuse previous settings
❯   No, customize settings - Choose your own preferences
```

```
✔ Would you like to use TypeScript? … No / Yes
✔ Which linter would you like to use? › ESLint
✔ Would you like to use React Compiler? … No / Yes
✔ Would you like to use Tailwind CSS? … No / Yes
✔ Would you like your code inside a `src/` directory? … No / Yes
✔ Would you like to use App Router? (recommended) … No / Yes
✔ Would you like to customize the import alias (`@/*` by default)? … No / Yes
✔ Would you like to include AGENTS.md to guide coding agents to write up-to-date Next.js code? … No / Yes
Creating a new Next.js app in /Users/alexvun/my-profile-card.

Using npm.
```

WHAT IS JSX?

- JSX stands for JavaScript XML.
- It allows us to write HTML directly inside React.
- Makes it much easier to write and add HTML in React.

STILL DON'T KNOW WHAT'S JSX? WELL...

JSX = HTML + JavaScript had a baby 👶

```
// Normal HTML
<h1>Hello World</h1>

// JSX (inside React)
function Greeting() {
  return <h1>Hello World</h1>
}
```

📁 FILE STRUCTURE (WHERE DOES EVERYTHING LIVE?)

AND WHAT CAN I SAFELY DELETE WITHOUT BREAKING EVERYTHING?

Open your folder in VS Code

- Open your `terminal (MacOS)/ cmd (Win)`

```
cd my-profile-card
```

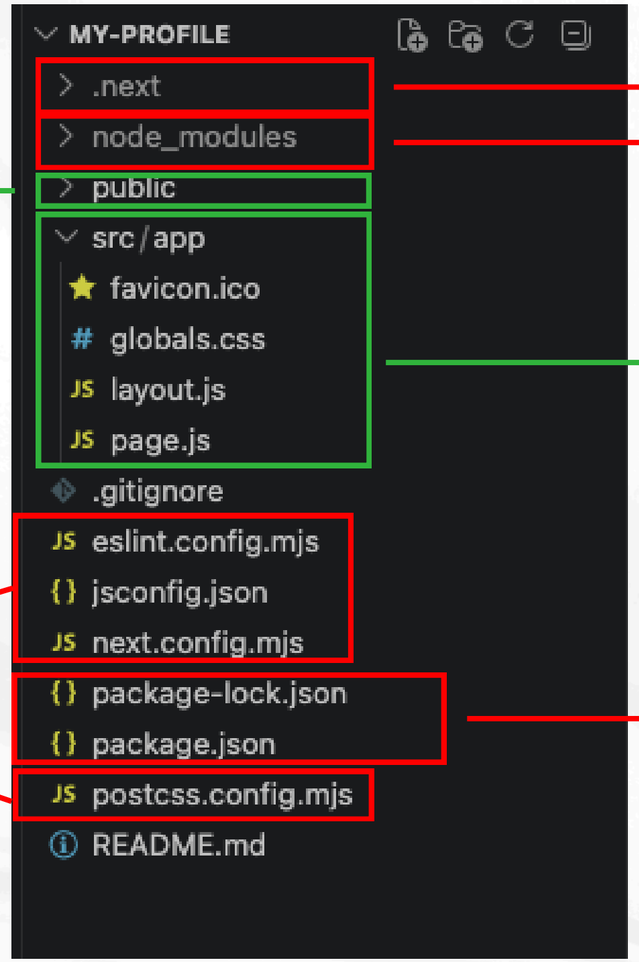
```
code .
```

```
Success! Created my-profile-card at /Users/alexvun/my-profile-card

alexvun@192 ~ % cd my-profile-card
alexvun@192 my-profile-card % code .
alexvun@192 my-profile-card %
```

Project Structure

Today's playground: src/app/ only.



Next.js engine room. Don’t touch!!!

1000 packages. Don't touch!!!

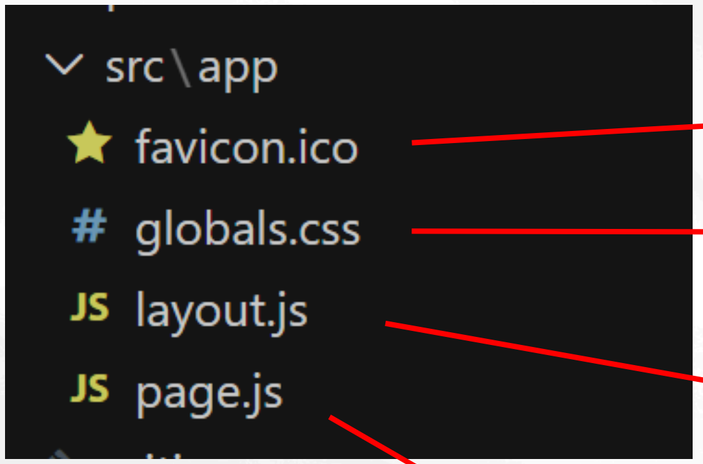
Your images can go here ✅

YOUR MAIN CODE LIVE HERE ✅

Settings and config files.
Just leave these alone for now.

all the tools your project uses

src\app Structure



Site icon

Application wide styles and **tailwind css** imports

shared UI elements that appears on all pages (navbar/footer)

Unique UI for home page (/)
localhost:3000

BUILD & TEST

HOW WE MAKE IT LIVE IN LOCAL?

```
npm run dev
```

DEVELOPMENT VS PRODUCTION

```
# Development mode (what we've been using)
npm run dev
# - Hot reload, detailed error messages
# - Larger bundle size, slower
# - Great for development

# Production build
npm run build
# - Optimizes everything for speed
# - Minifies code, optimizes images
# - Creates static files

# Production preview
npm run start
# - Runs the optimized version
# - What users will actually see
```



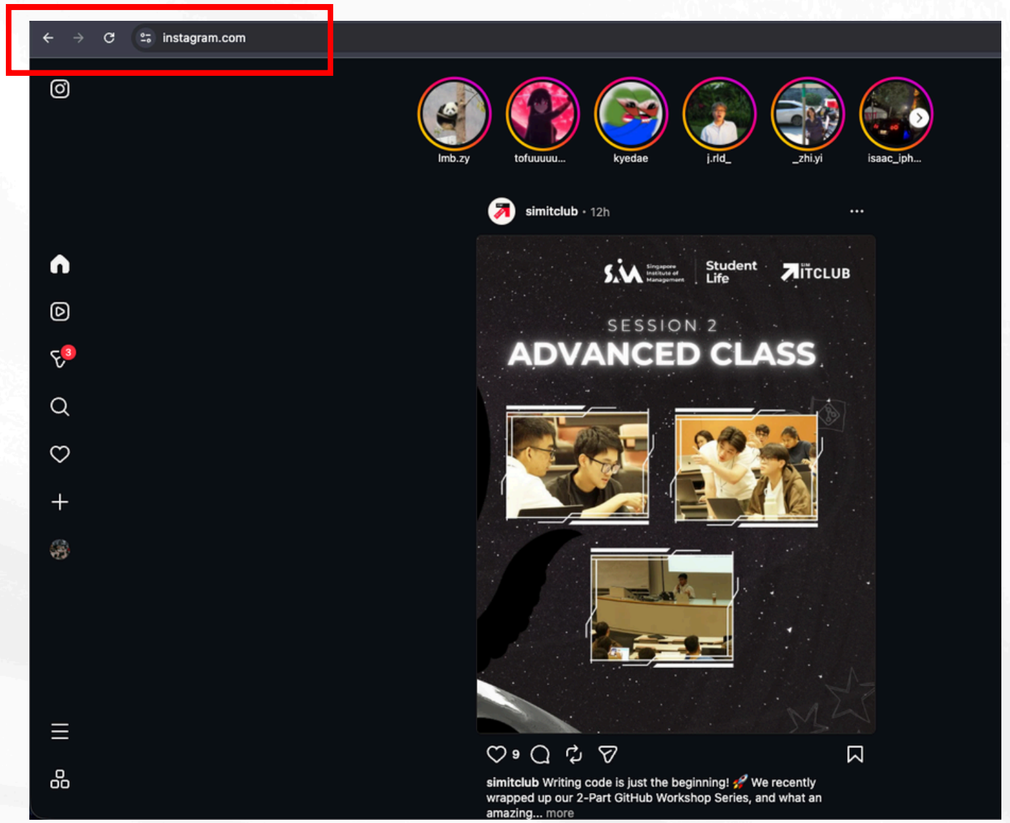


ROUTING

APP ROUTER/ FILE BASED ROUTING

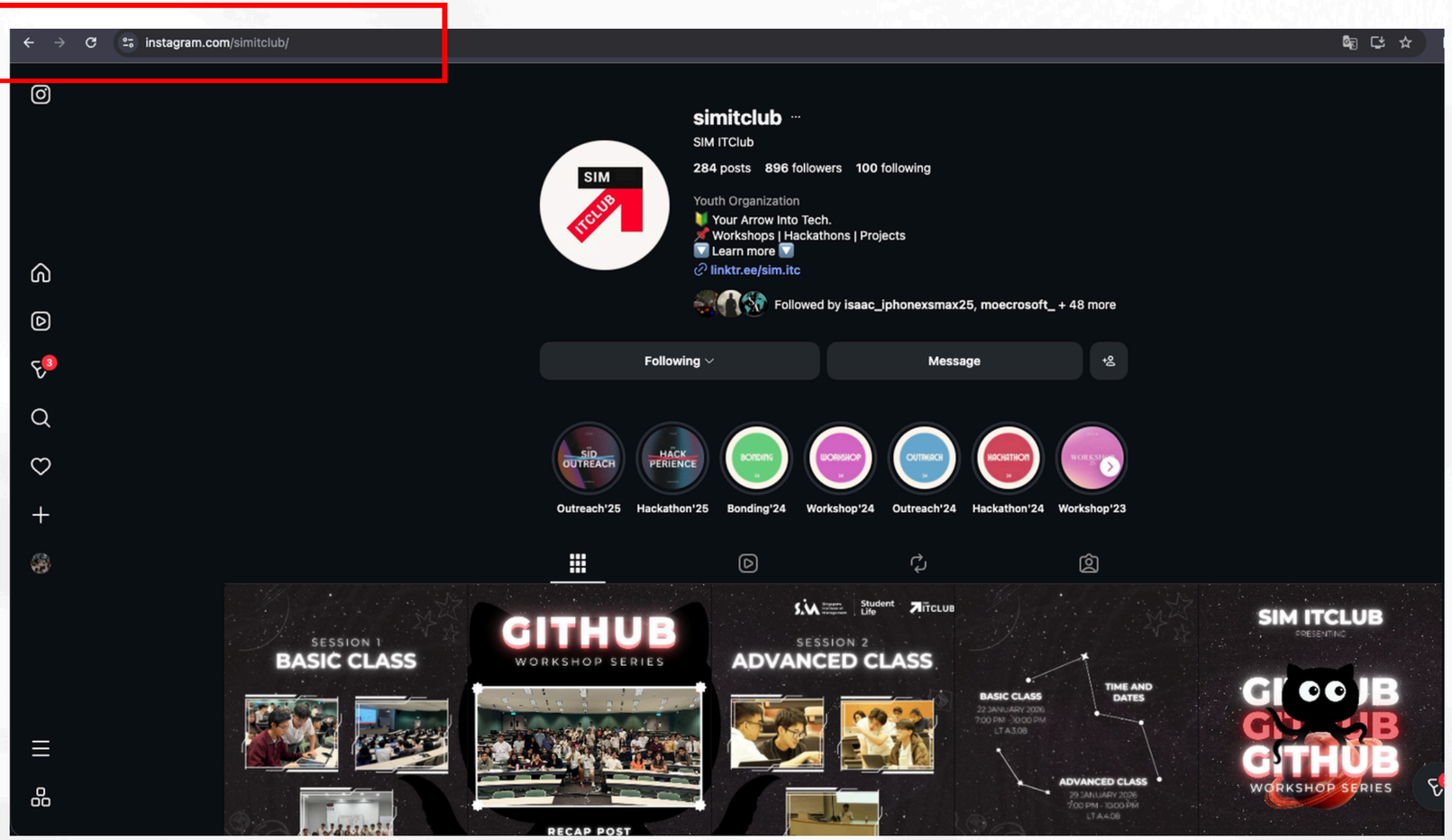
HOW DOES YOUR WEBSITE KNOW WHICH PAGE TO SHOW?

What is routing?



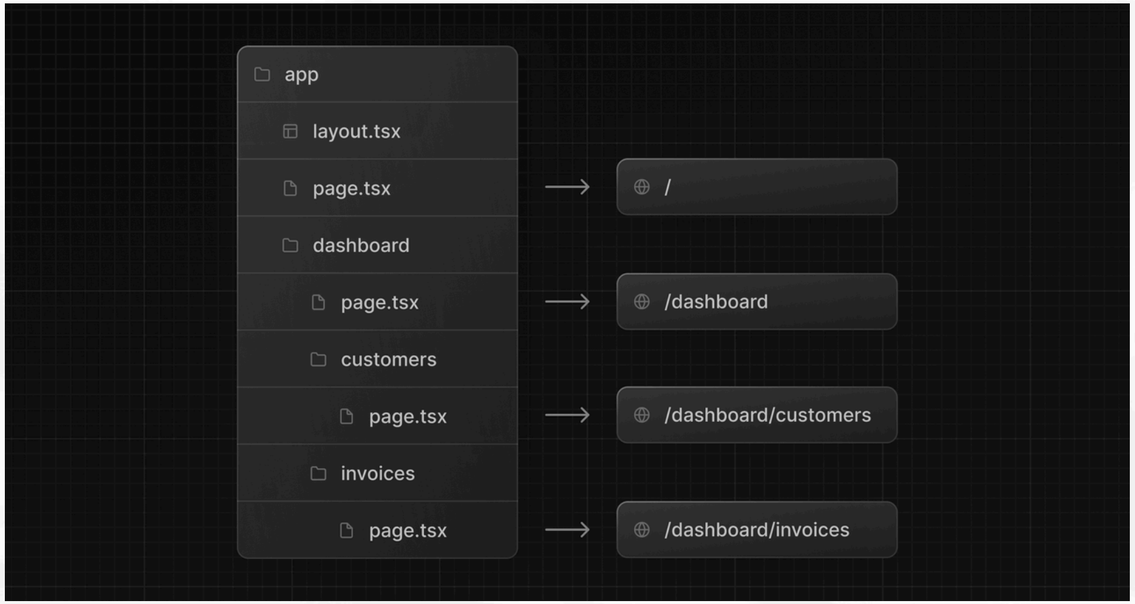
What is routing?

WHY?



🗺️ Routing = GPS for Your Website

- User types **/dashboard** in the URL → Website will determine what user requested and says: "Oh, show the **dashboard** page!"
- Folder name = URL path. No extra config needed.



Old school HTML routing



With plain React

IMPORT MANUALLY

```
const React = require('react');
const Router = require('react-router').Router;
const Route = require('react-router').Route;
const App = require('./components/page/App');
const Nav = require('./components/component/Nav');
const Hero = require('./components/component/Hero');
const About = require('./components/component/About');
const Contact = require('./components/component/Contact');
const Footer = require('./components/component/Footer');

const routes = (
  <Router>
    <Route path="{nav}" component={Nav} />
    <Route path="{hero}" component={Hero} />
    <Route path="{about}" component={About} />
    <Route path="{contact}" component={Contact} />
    <Route path="{footer}" component={Footer} />
  </Router>
);

module.exports = routes;
```

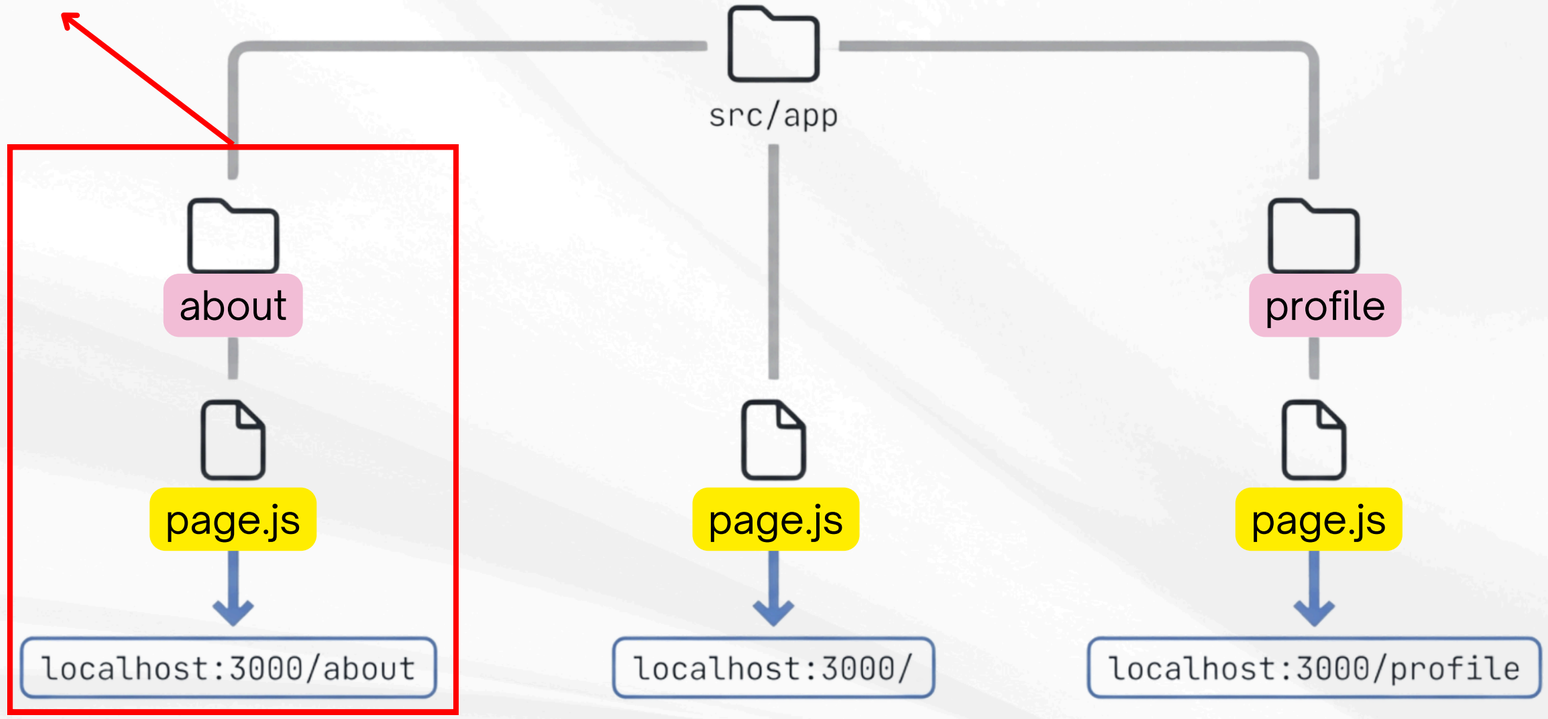


🧭 Next.js App Router

File based routing

new page? new route? just create file and put **page.js**, thats all!!!

folder structure becomes the URL structure



COMPONENTS

(THE BUBBLE TEA THEORY) 🧋



THE BUBBLE TEA ANALOGY

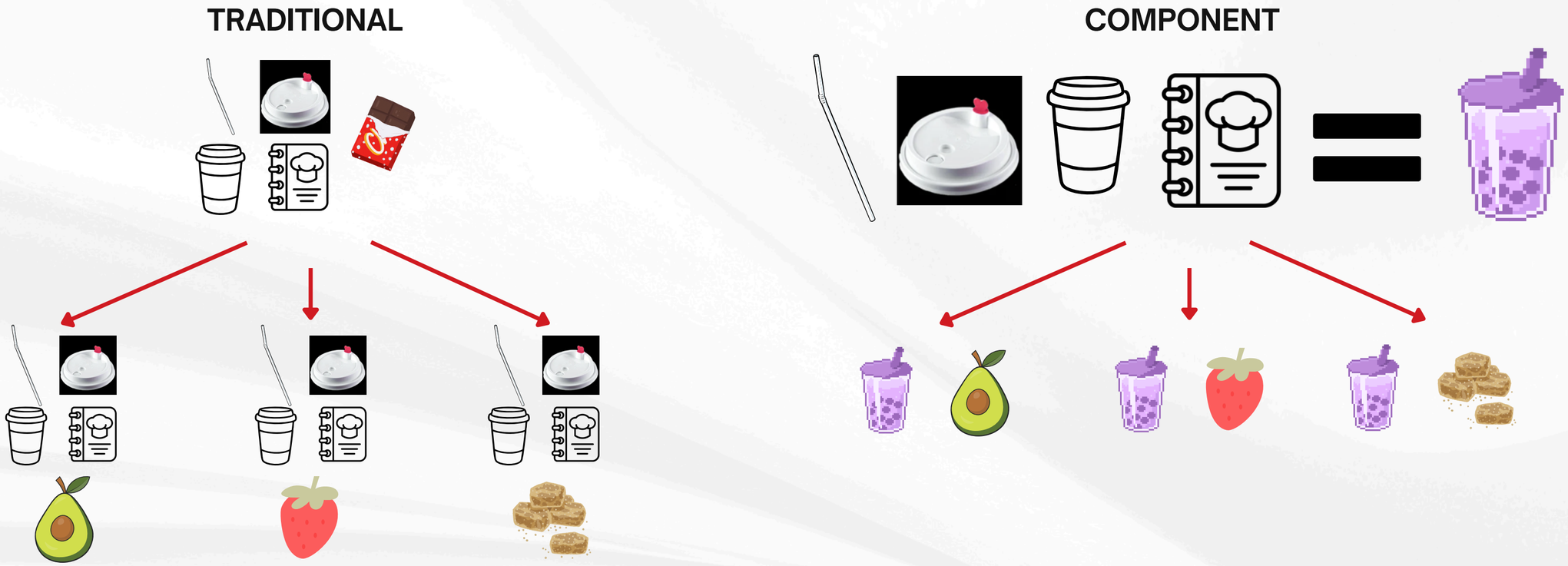
- A component = the standard cup + straw + recipe.
- You make the template once. Every order uses the same template. Each order has different toppings (data).
- Stop building a brand new cup from scratch every time!

```
<BubbleTeaCup flavour="Taro" sugar="50%" />
<BubbleTeaCup flavour="Brown Sugar" sugar="100%" />
<BubbleTeaCup flavour="Matcha" sugar="30%" ice="less" />
```

One component. Three different drinks.

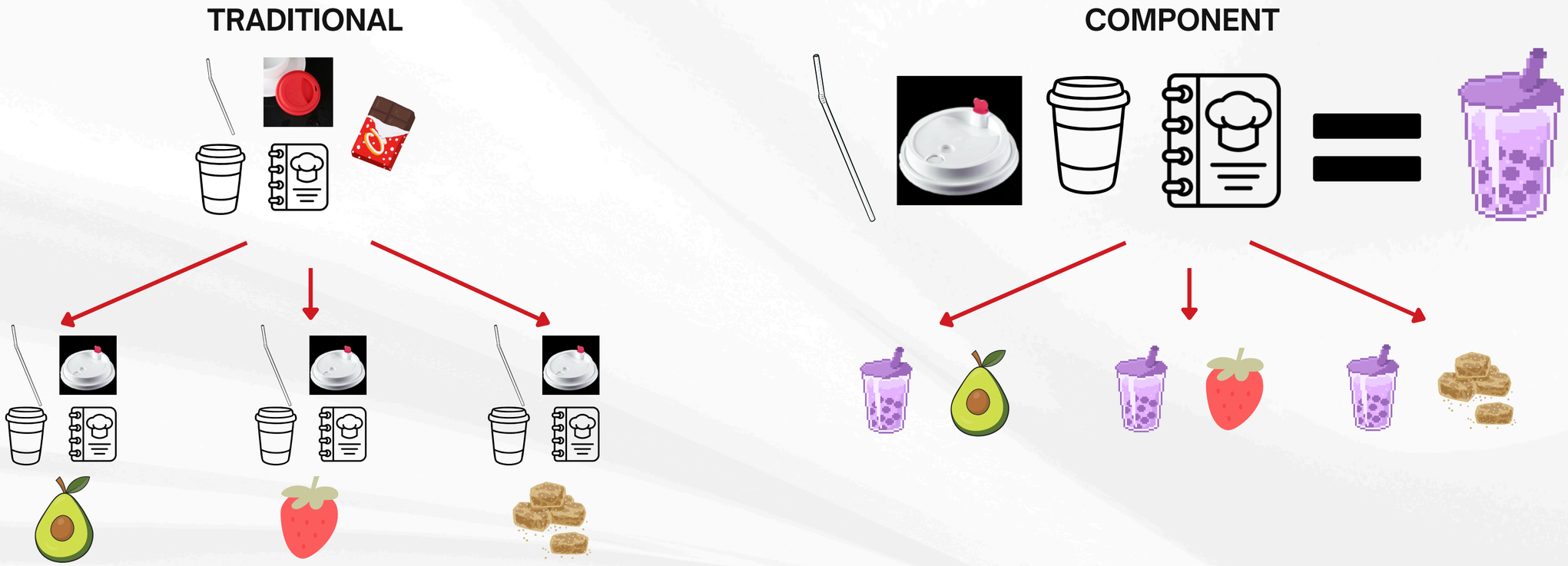
WHY COMPONENT?

CHANGE IT TO BUBBLE TEA CONCEPT



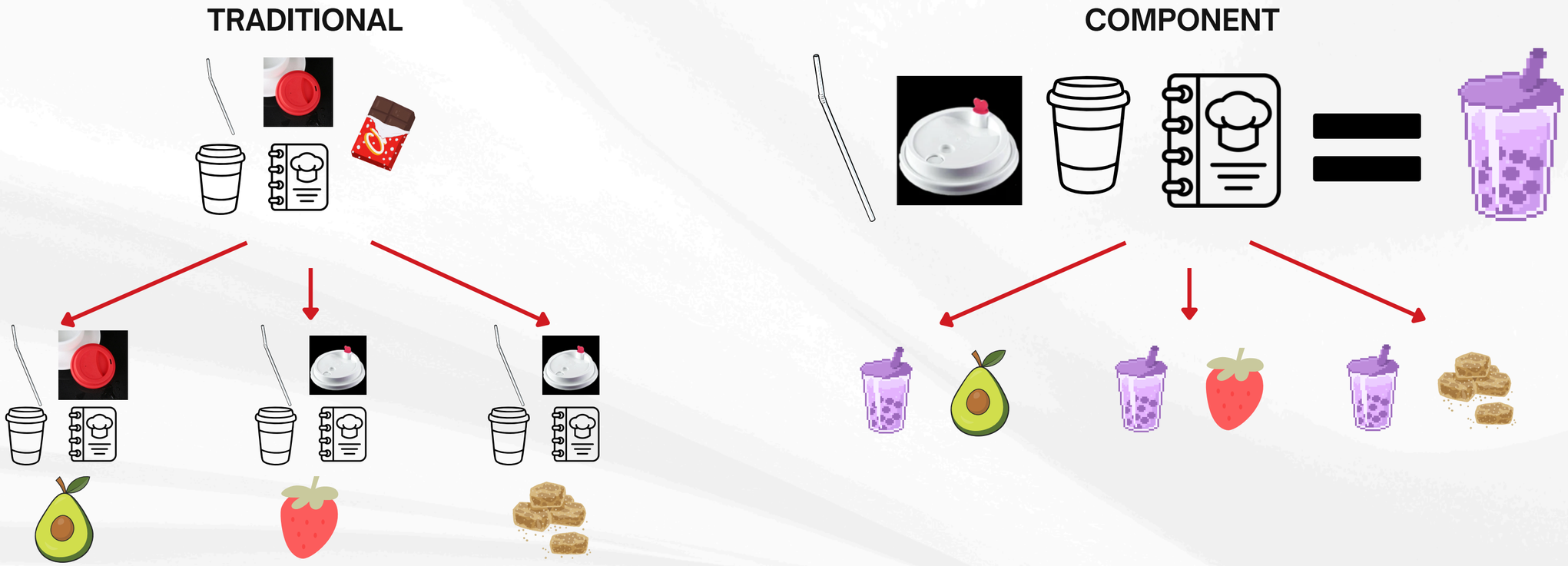
WHY COMPONENT?

CHANGE IT TO BUBBLE TEA CONCEPT



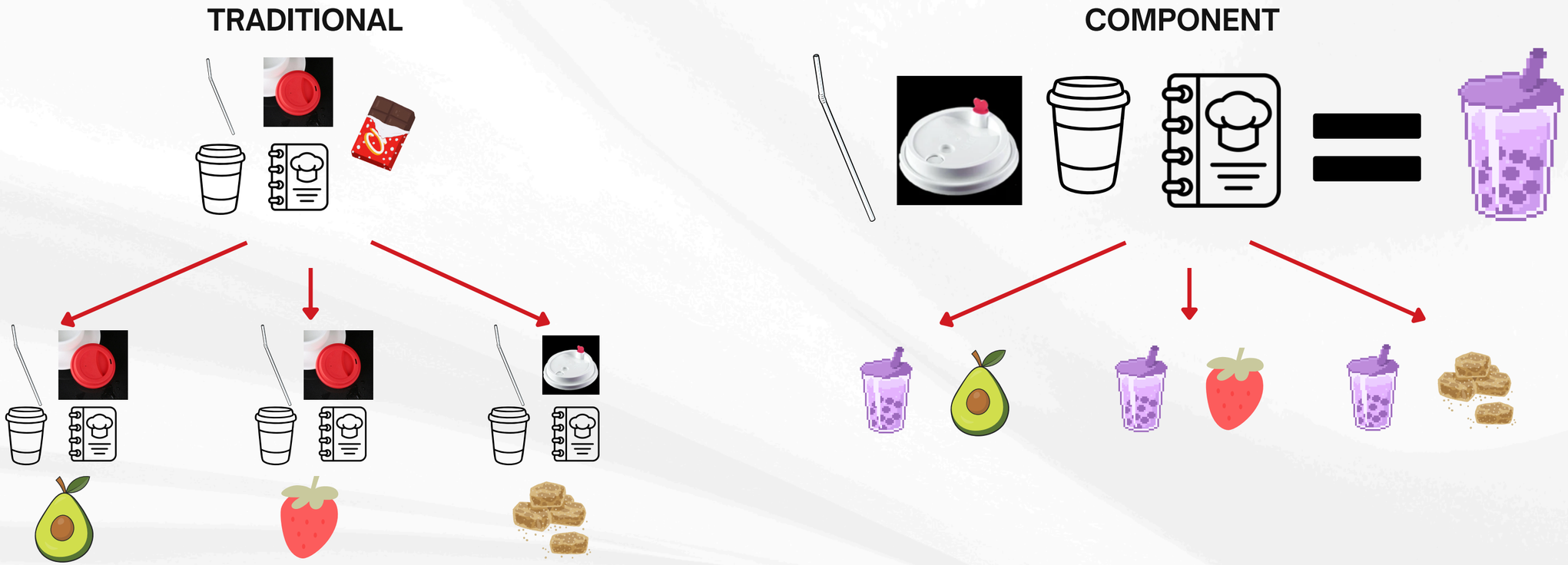
WHY COMPONENT?

CHANGE IT TO BUBBLE TEA CONCEPT



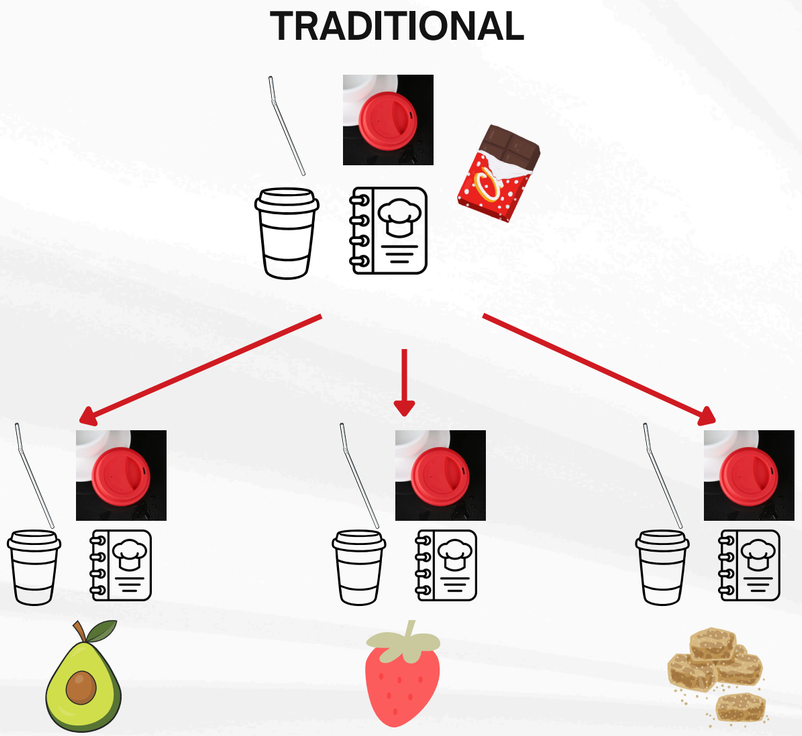
WHY COMPONENT?

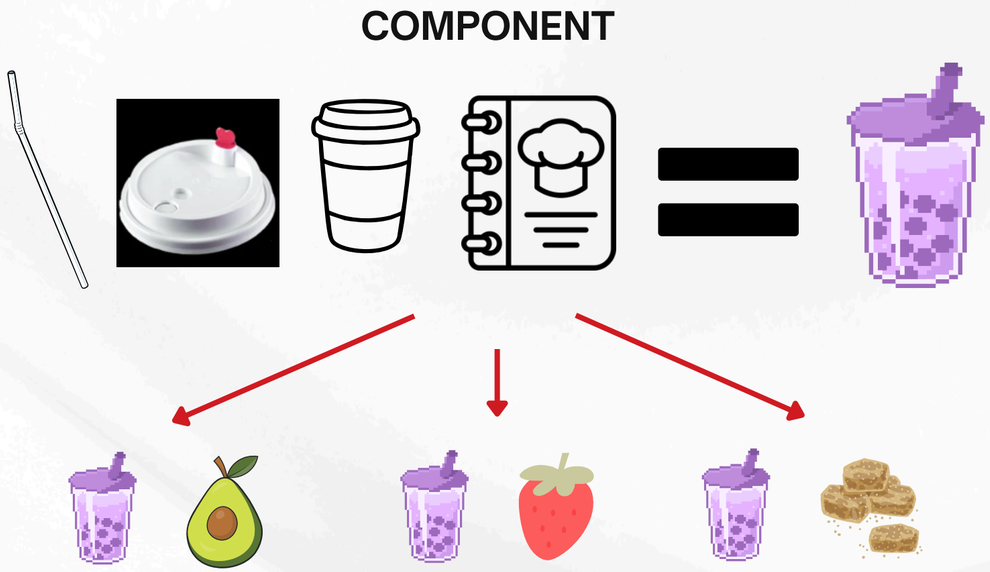
CHANGE IT TO BUBBLE TEA CONCEPT



WHY COMPONENT?

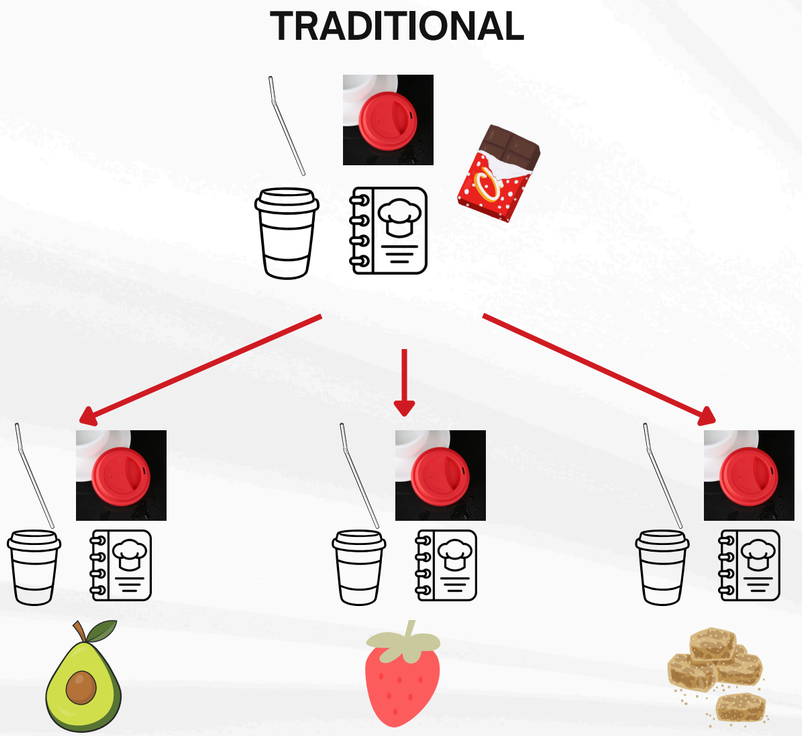
CHANGE IT TO BUBBLE TEA CONCEPT

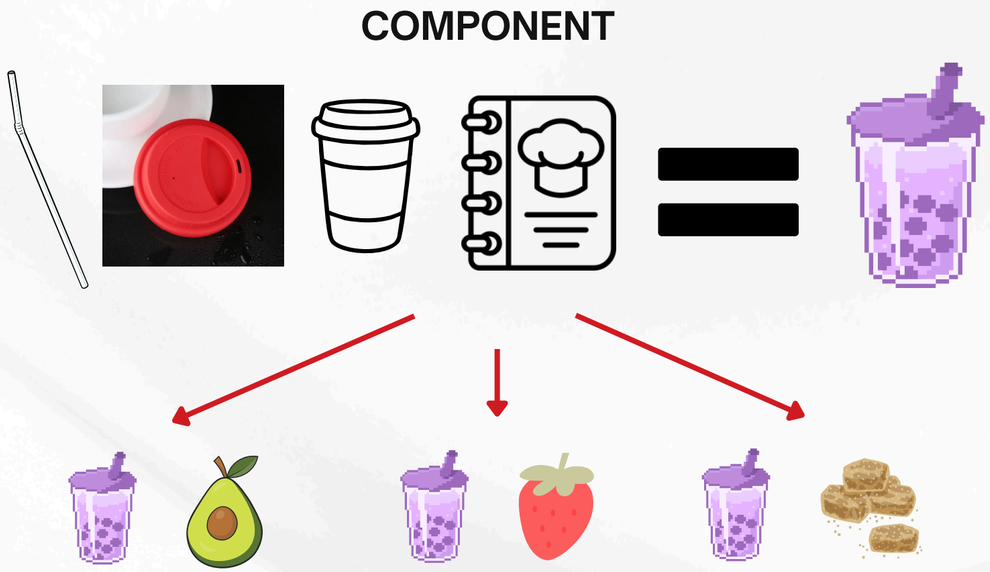




WHY COMPONENT?

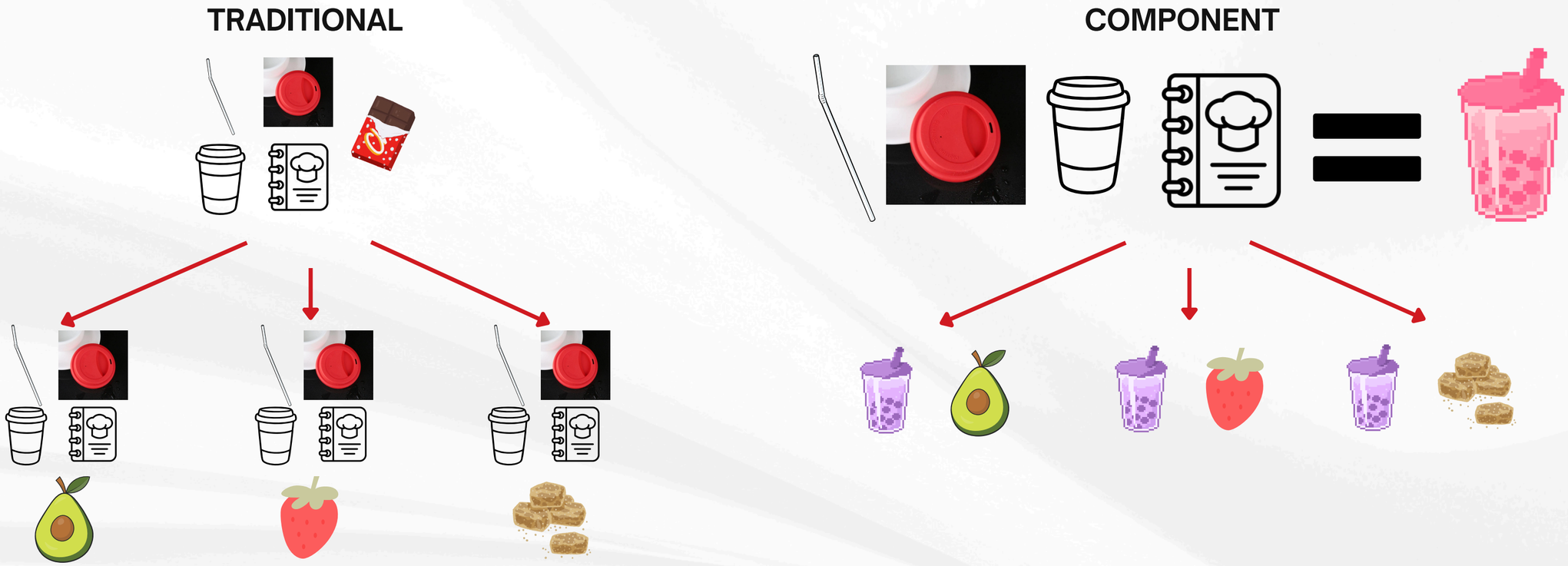
CHANGE IT TO BUBBLE TEA CONCEPT





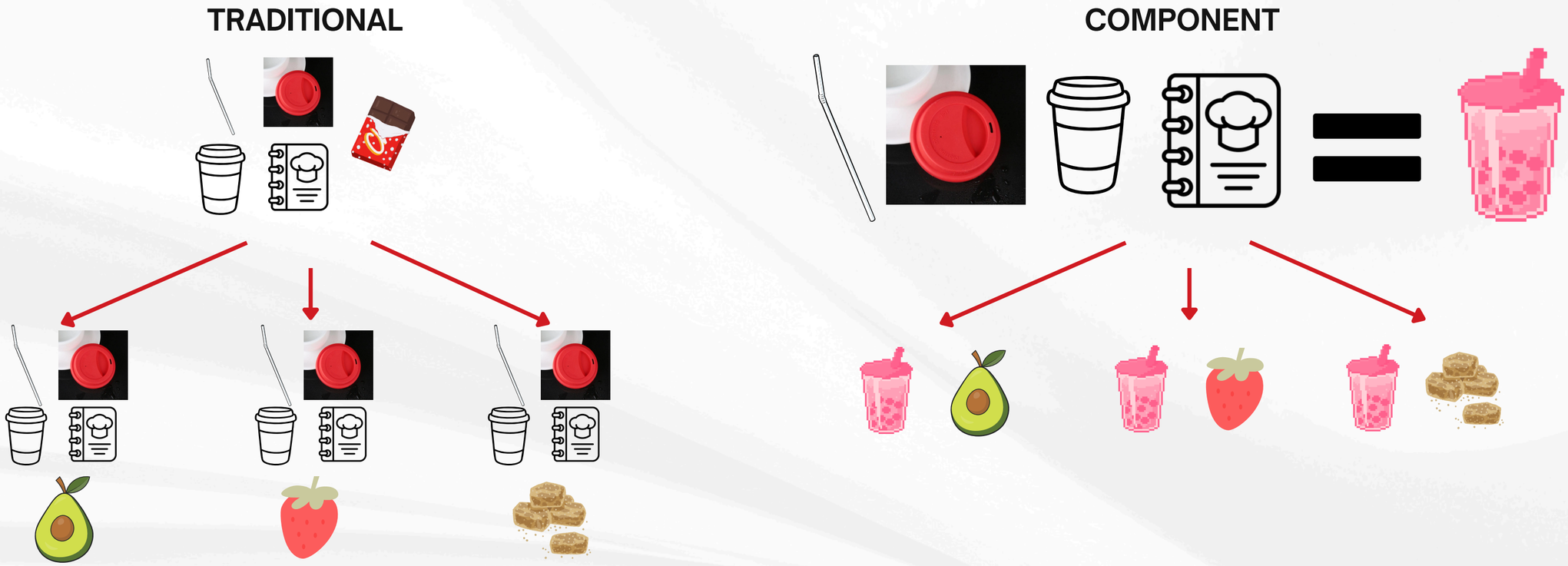
WHY COMPONENT?

CHANGE IT TO BUBBLE TEA CONCEPT



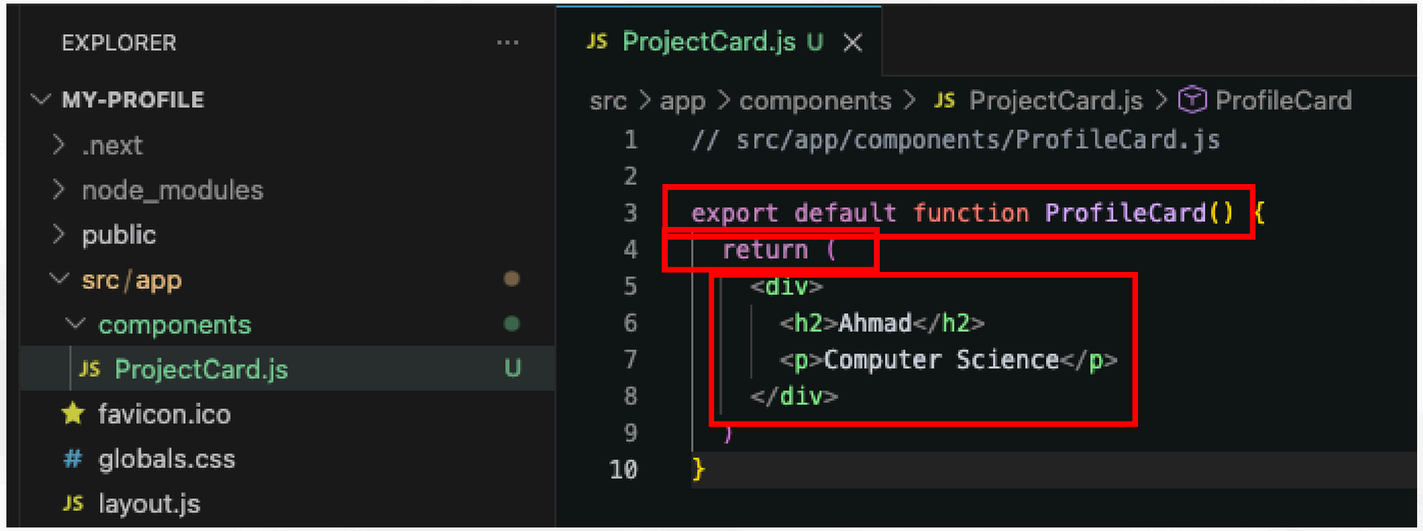
WHY COMPONENT?

CHANGE IT TO BUBBLE TEA CONCEPT



REACT COMPONENT

✍️ What a Component Looks Like



3 things every component needs:

1. A function (capital letter name!)
2. A return statement
3. JSX inside the return

 COMPONENTS COME IN DIFFERENT FORMS

Same concept. Different purposes.

React

Regular/**Custom Component**
[like your your ProfileCard]

Next.js (built-in components)

- **<Link>** — navigation without page reload
- **<Image />** — optimised images automatically
- **layout.js** — shared UI across all pages

For more Next.js components:
https://nextjs.org/docs/app/api-reference/components

Page Component

Individual website page

```
export default function About() {
    return (
        <div>
            <h1>About Page</h1>
        </div>
    )
}
```



Link Component

```tsx
import Link from "next/link";
```

```tsx
<Link href="/about">About</Link>
<Link href="/profile">Profile</Link>
```

<Link>

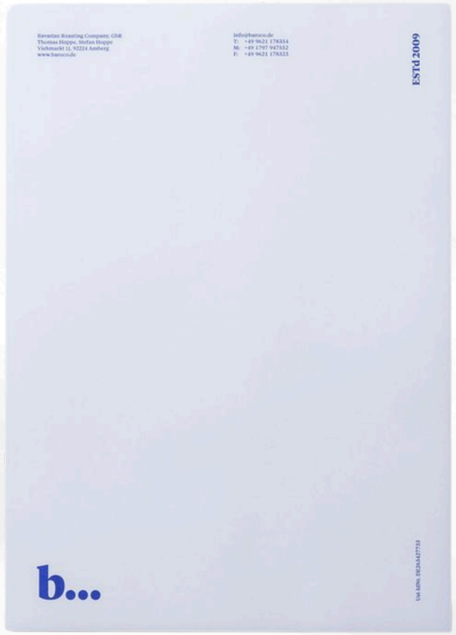
- Instant navigation, next.js loads the next page instantly
- Better UX & smooth transitions
- Works perfectly with server components
- Improves SEO

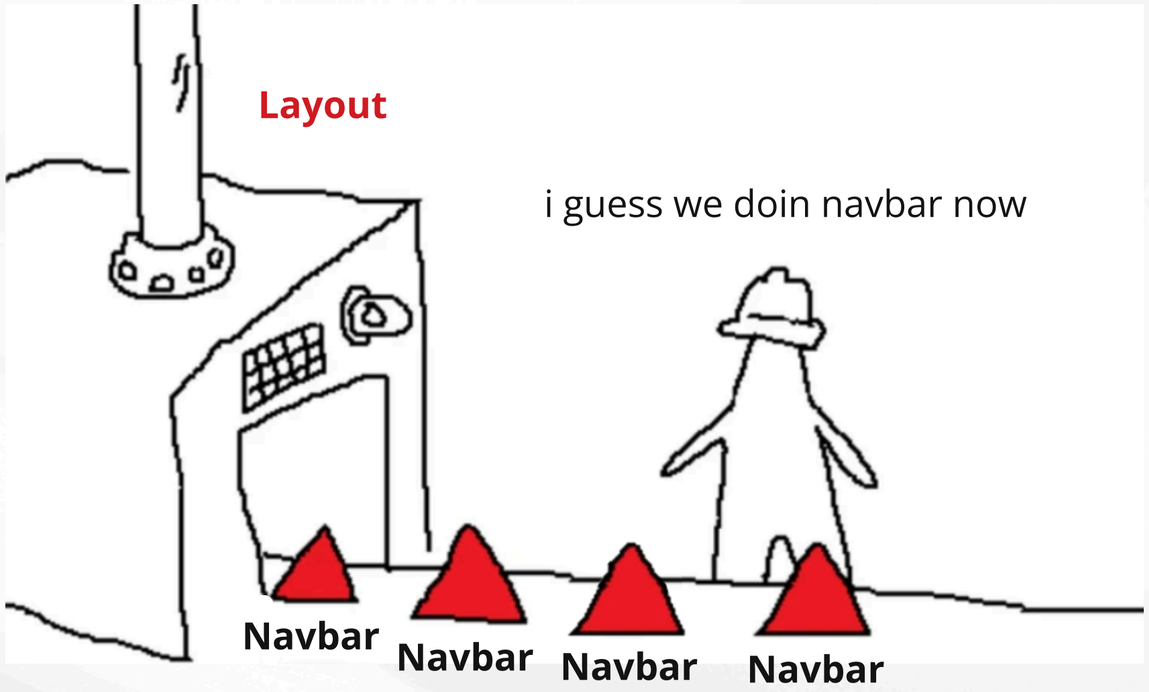


Layout Component

A layout is UI that is shared between multiple pages in your app.



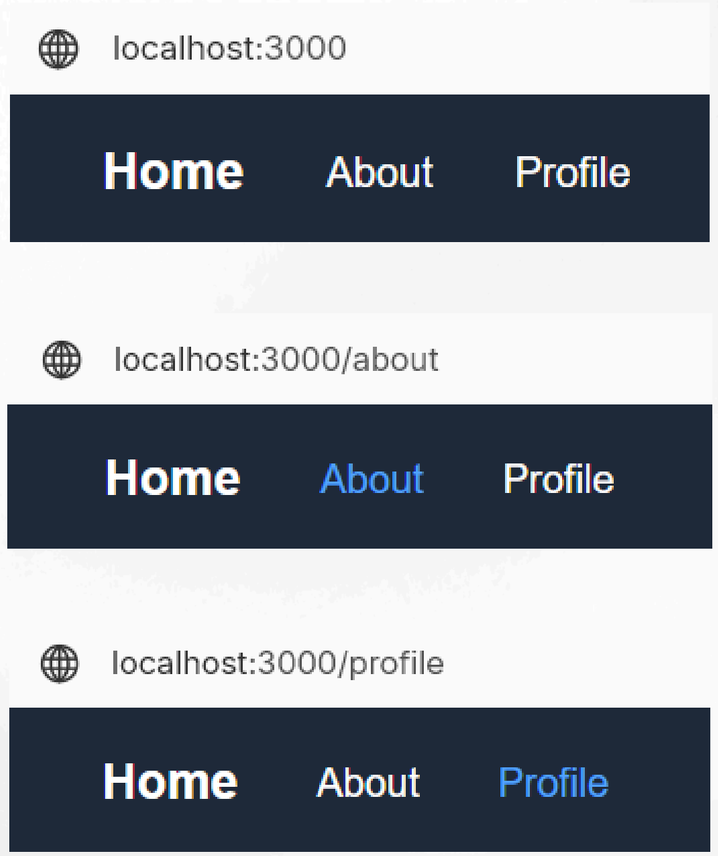




- Making a navbar layout

```tsx
import Navbar from "../component/navbar";
import "./globals.css";

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body>
        <Navbar />
        {children}
      </body>
    </html>
  );
}
```



BREAK ⏱️ ~15 MINS

⏭️ NEXT STOP: PROPS



PROPS

HOW TO CUSTOMISE YOUR COMPONENT

(BACK TO PERSONAL PROFILE INFO SCENARIOS) 🪪

📦 Props: Before vs. After

❌ **Without Props — hardcoded, not reusable:**

```
function ProfileCard() {
  return <h2>Ahmad</h2>  // Always Ahmad. Forever.
}
```

✅ **With Props — flexible, reusable:**

```
function ProfileCard({ name, course }) {
  return <h2>{name} — {course}</h2>
}

// Usage:
<ProfileCard name="Ahmad" course="CS" />
<ProfileCard name="Priya" course="Business" />
```

📦 Props: Different

```
function ProfileCard() {
  return <h2>Ahmad</h2>  // Always Ahmad. Forever.
}
```

It’s hardcoded, even you call it many time,
it’s still Ahmad. That’s mean your component is not dynamic..

```
function ProfileCard({ name, course }) {
  return <h2>{name} — {course}</h2>
}

// Usage:
<ProfileCard name="Ahmad" course="CS" />
<ProfileCard name="Priya" course="Business" />
```

A

B

A - created name and course with curly braces {} as boxes to receive the values.

B - pass the values with name=”Ahmad” and course=”CS” to the function

📦 Props: Parent and Child Components

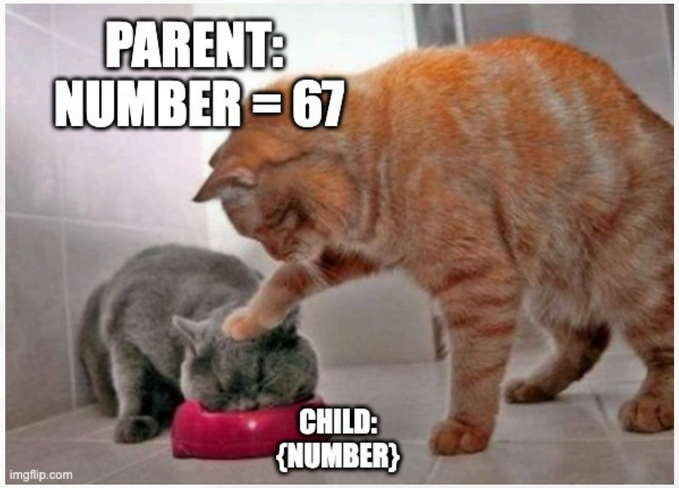
Parent — Uses the component, passes props:

```
// page.js (the parent)
<ProfileCard name="Ahmad" course="CS" />
```

Child — Receives and displays the props:

```
// ProfileCard.js (the child)
function ProfileCard({ name, course }) {
  return <div>{name} – {course}</div>
}
```

It means... Parent feeds data down. Child displays it.
Data only flows ONE way: parent → child.





TAILWIND CSS

MAKING IT LOOK LESS UGLY.

(NO SEPARATE CSS FILES. NO SWITCHING TABS. JUST CLASSES.)

🎨 TAILWIND: STYLING AT THE SPEED OF THOUGHT

No more flipping back and forth between HTML and CSS files

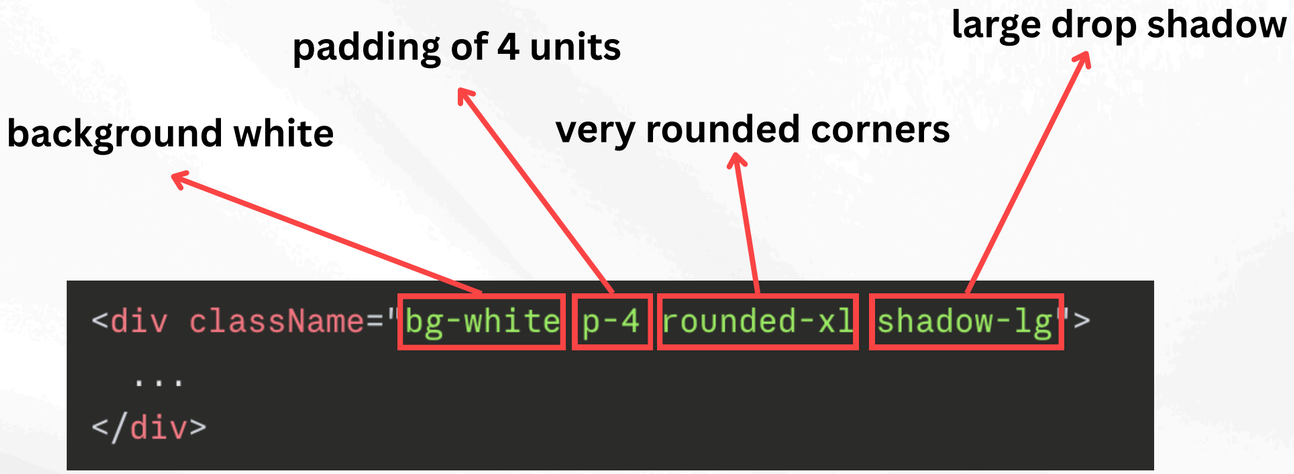
```css
/* styles.css */
.card { background: white; padding: 16px; border-radius: 12px; }
```

```jsx
<div className="card">...</div>
```

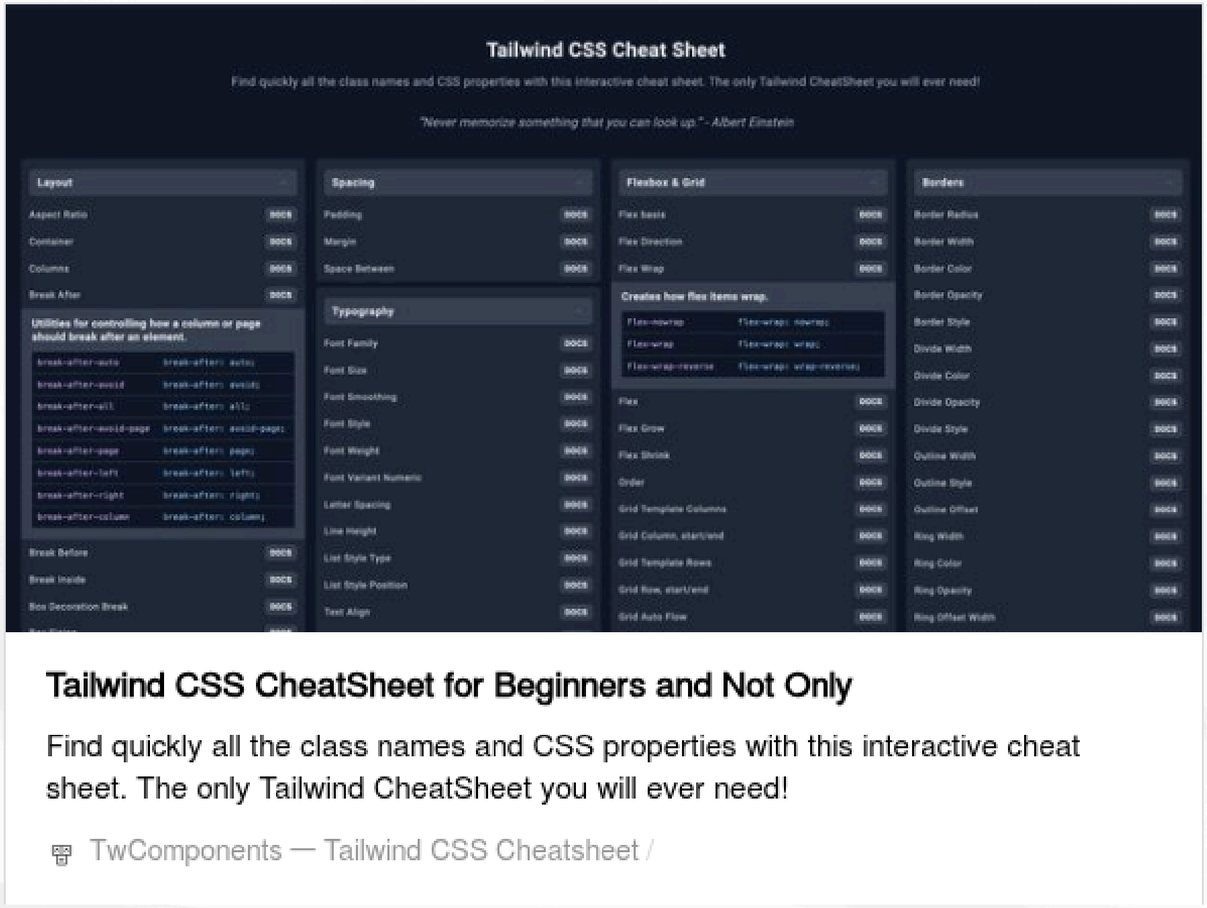
Utility-first: Write styles directly in your class names.

```jsx
<div className="bg-white p-4 rounded-xl shadow-lg">
  ... Read the class names. They say exactly what they do.
</div>
```

🎨 TAILWIND: STYLING AT THE SPEED OF THOUGHT



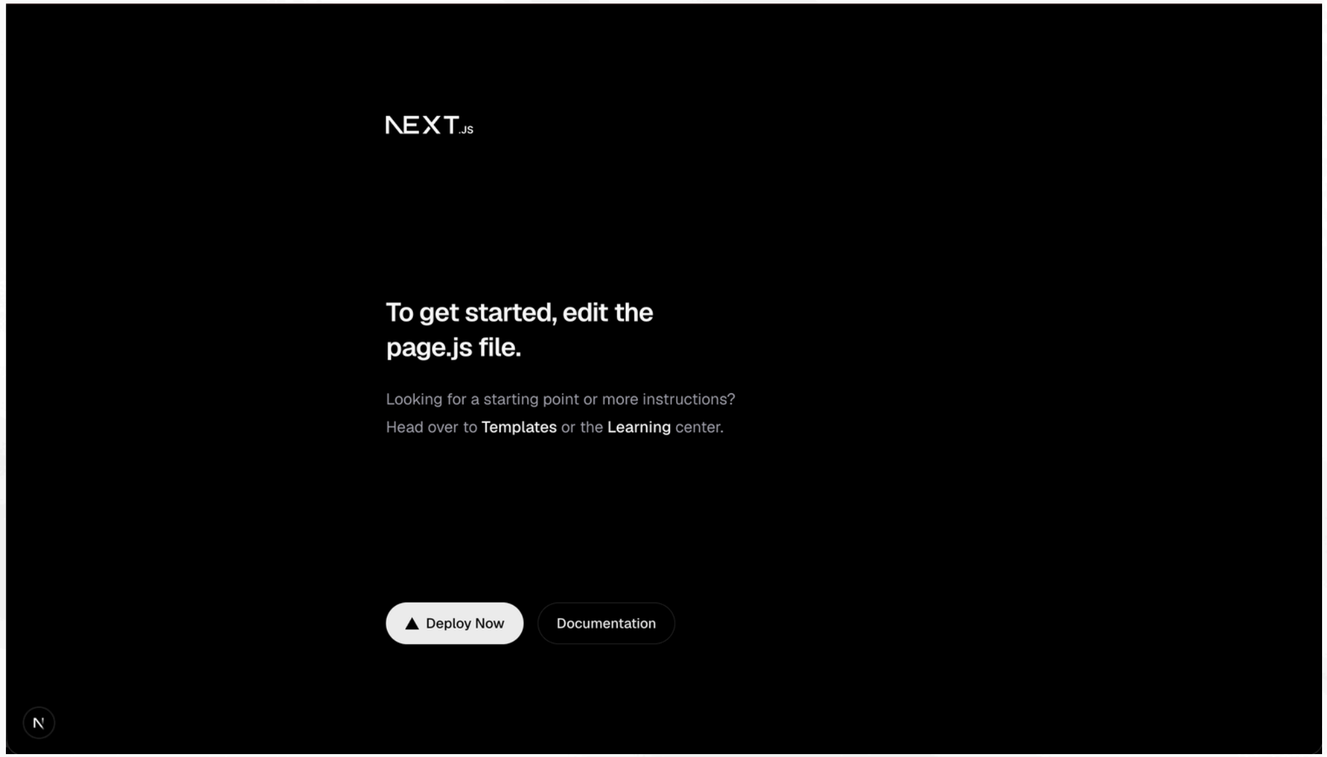
- Tailwind CSS is designed to be readable
- Reads like English
- Tailwind CSS documentation


Tailwind CSS Cheat Sheet
Find quickly all the class names and CSS properties with this interactive cheat sheet. The only Tailwind CheatSheet you will ever need!
"Never memorize something that you can look up." - Albert Einstein
Layout
Spacing
Flexbox & Grid
Borders
Typography
Tailwind CSS CheatSheet for Beginners and Not Only
Find quickly all the class names and CSS properties with this interactive cheat sheet. The only Tailwind CheatSheet you will ever need!
TwComponents — Tailwind CSS Cheatsheet /

HANDS-ON #00

CLEAN UP BEFORE BUILD 🛠️

⏱️ ~2 MINUTES



- This is the default Next.js "Boilerplate".
- It proves your installation worked! 🎉
- Now... we delete it to make room for our code.

EXERCISE

- Open `src/app/page.js` and delete **EVERYTHING**.
- Write the code below:

```
export default function Home() {
  return (
    <main className="min-h-screen bg-gray-100 p-8 text-gray-900">
      <h1>My Workspace</h1>
    </main>
  );
}
```

- **Navigate to `localhost:3000/` on your browser and test it**

HANDS-ON #01

BUILD YOUR PROFILECARD COMPONENT 🛠️

⏱️ ~10 MINUTES

EXERCISE

- Create `src/components/ProfileCard.js`
- Write a function component that accepts props: `name, course, description`
- Display all three props in your JSX

```jsx
export function ProfileCard({ name, title, desc }) {
  return (
    <div className="flex items-center gap-4 text-gray-900">
      <div>
        <h2 className="text-xl font-semibold">{name}</h2>
        <p className="text-gray-600">{title}</p>
        <p className="mt-1 text-sm">{desc}</p>
      </div>
    </div>
  );
}
```

HOW ABOUT IMAGE???

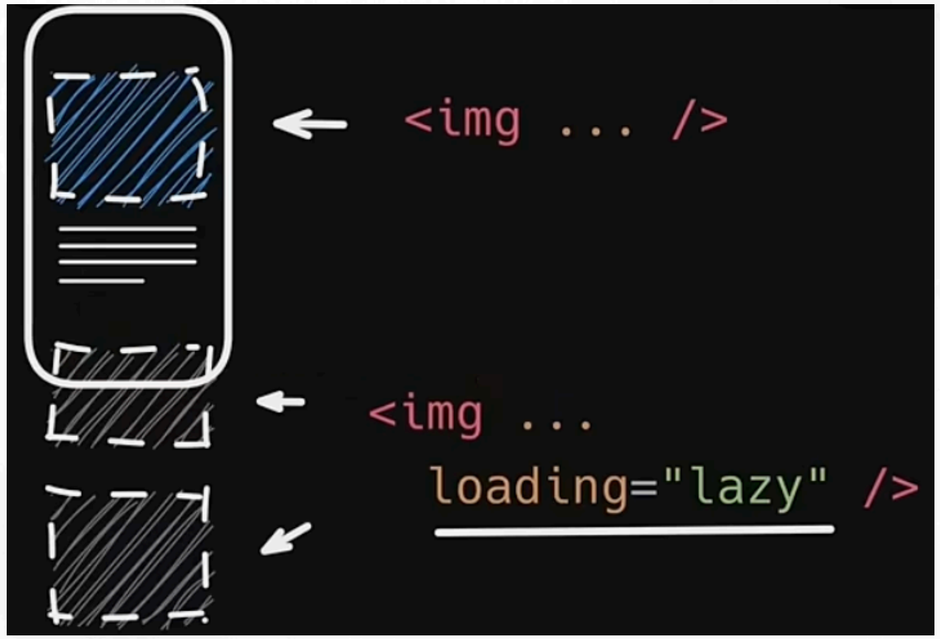
HANDS-ON #02

THE NEXT.JS <IMAGE /> COMPONENT 🖼️

⏱️ ~10 MINUTES

WHY USE <IMAGE />?

- Next.js <Image /> tag = Automatic resizing, lazy loading, WebP format.
- Standard <img> tag = Unoptimized, slow, layout shifts.



EXERCISE

- Drop a photo into the public/ folder
- Import Image component:
  - import Image from "next/image"
- Add parameter into function.
- Then, add to your component

```jsx
import Image from "next/image";

export function ProfileCard({ name, title, desc, imageSrc }) {
  return (
    <div className="flex items-center gap-4 text-gray-900">
      <Image
        src={imageSrc}
        alt={name}
        width={96}
        height={96}
      />

      <div>
        <h2 className="text-xl font-semibold">{name}</h2>
        <p className="text-gray-600">{title}</p>
        <p className="mt-1 text-sm">{desc}</p>
      </div>
    </div>
  );
}
```

HOW I CAN VIEW MY PROFILE CARD ON MY BROWSER???

HANDS-ON #03

ADDING AN "ABOUT ME" ROUTE 🗺️

⏱️ ~10 MINUTES

EXERCISE

- Create folder: `src/app/about/`
- Create file: `src/app/about/page.js`

```
import { ProfileCard } from "@/components/ProfileCard";

export default function About() {
  return (
    <main className="min-h-screen bg-gray-100 flex items-center justify-center p-8">
      <ProfileCard
        name="Alex"
        title="Computer Science Student"
        desc="Learning React and Next.js!"
        imageSrc="/alex.png"
      />
    </main>
  );
}
```

EXERCISE

- Go to: `src/app/page.js`
- Add a navigation button by using `<Link href="/about"> </Link>`
- Then can testing on click the link you created

```
import Link from "next/link";

export default function Home() {
  return (
    <main className="min-h-screen bg-gray-100 p-8 text-gray-900">
      <h1>My Workspace</h1>

      <Link href="/about">Go to About Me</Link>
    </main>
  );
}
```

NOW.

SEE YOURSELF ON YOUR OWN WEBSITE 🤩



HANDS-ON #04

MAKE YOUR CARD LOOK GOOD 💅

⏱️ ~10 MINUTES

EXERCISE

- Must have:
  - **ProfileCard.js**
    - bg-white — clean white card
    - text-gray-800 — readable dark text
    - p-6 — breathing room
    - rounded-xl — soft corners
    - shadow-lg — depth / “card” feel
  - **about/page.js**
    - bg-gradient-to-br — diagonal gradient
    - from-blue-50 — light blue start
    - to-indigo-100 — soft indigo end
- Bonus points!!! 🔥 :
  - **ProfileCard.js**
    - transition-all — animate style changes
    - duration-300 — 300ms smooth feel
    - hover:shadow-2xl — bigger shadow on hover
  - **Image styles (on the <Image /> in ProfileCard.js)**
    - rounded-full — makes the photo a circle
    - object-cover — crops photo nicely, no stretch
    - w-24 h-24 — fixed avatar size
    - shrink-0 — photo won’t get squished by long text
    - ring-4 ring-blue-200 — soft colored ring (extra polish)
    - hover:scale-105 — slight zoom on hover (pair with parent transition)

SUMMARY

WHAT'S NEXT? 🎉

🎉 WHAT YOU JUST DID

✅ Set up a real Next.js project

✅ Understood React components + Next.js pre-built components **[Bubble Analogy]** 

✅ Passed data using props

✅ Created multiple pages with file-based routing

✅ Styled with Tailwind CSS

✅ Used the Next.js **<Image />** component **[Hands-on #02]**

DAY 2 TEASER

- State 👉 Making buttons actually do things (Click → something changes → it's interactive)
- Deploying to Vercel 👉 Your site goes LIVE (Real URL. Show off to your friends. Put it on LinkedIn.)

🚨 HOMEWORK BEFORE DAY 2:

- Create a free account at vercel.com
- Connect it to your GitHub account
- Push your project to GitHub

THAT'S A WRAP

YOU SURVIVED. YOU BUILT. YOU REACTED. (GET IT?)

Share your card with us!

Tag @SIM Information Technology Club,
@alexvun & @moecrosoft
on LinkedIn 🔥

📲 Stay connected:
Telegram: SIM ITCommunity
Email: itc@mymail.sim.edu.sg

THANK YOU