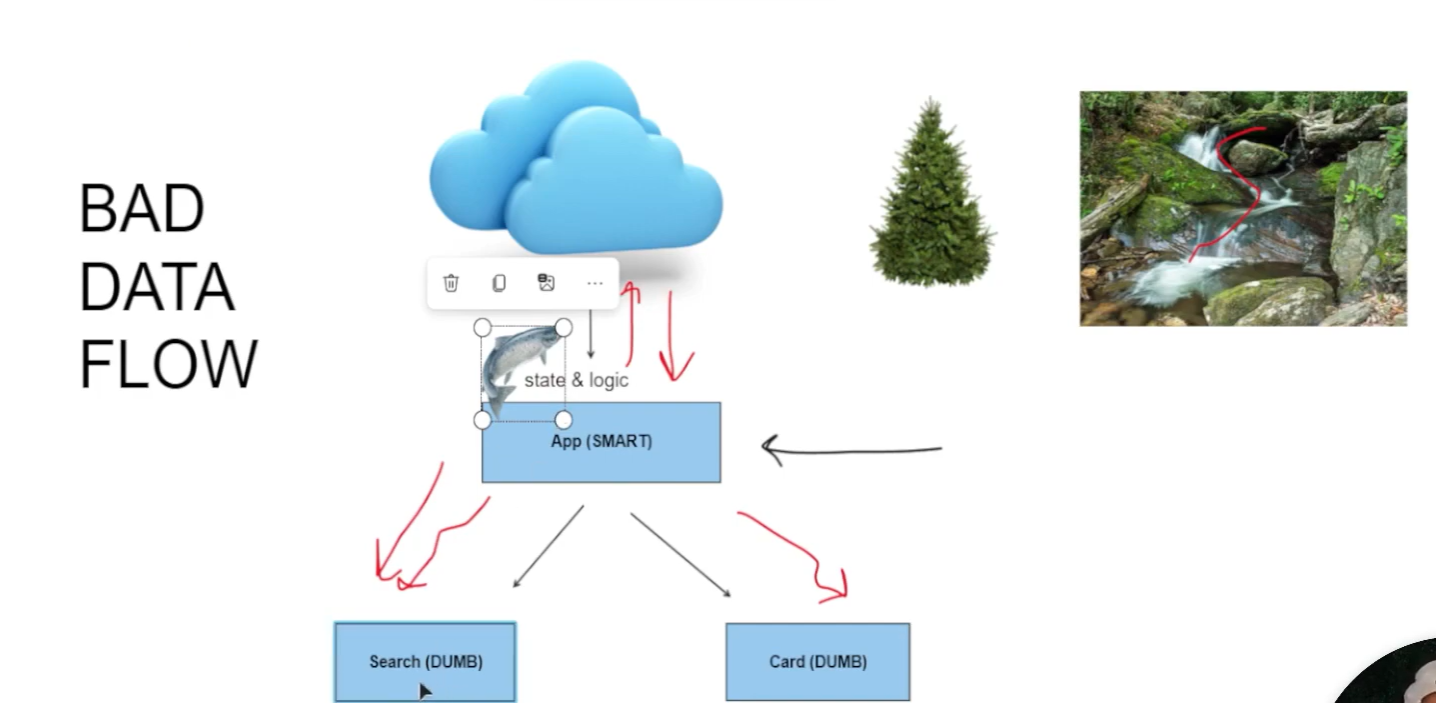
**3 props:  
we use interface since interfaces will give us a label for our data, also add data check, putting checks and balances about our data we are passing, (typescript says no like since companyName is string)**

**7. Financial Modeling Prep API**npm install axios --save

npm install –save-dev @types/axios

npm install dotenv –save (for the .env)  
  
so li sar eno bel company.d.tsx htyna kel shi entities (interfaces fiha attributes)  
then bel api 3mlna interface lal method and then we called the api  
  
then bel index we just tested in a simple console.log(searchcompanies(“tsla”) and added tesla

**8. Data Flow**

****

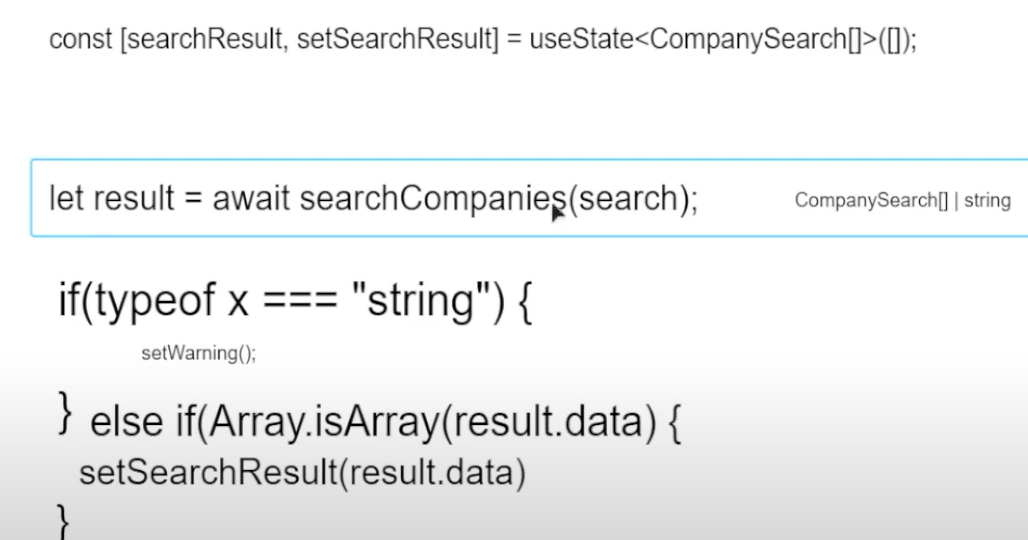
A good data architecture is a cross betweena christimas tree and a river:  
  
good react architecture is to be as tree, (fo2 smart and tahet dumb) so state and logic should be fo2  
  
so this will allow our app to act similar as data river: we cann call api and get our data  
Data will flow down and , and the dumb components will house the events and the events, can   
go up into smart comp -> trigger the logic, go get the data and this components can trigger the data river once again  
  
so search comp is dumb  
  
so the logic in it is to remove the logic code from search and them to app:  
  
const [search, setSearch] = useState<string>(“”);

Const handleChange = (e: ChangeEvent<HTMLInputEelement>) => { setSearch (e.target.value)]

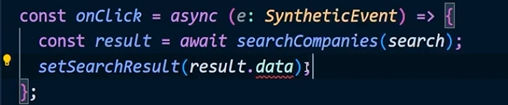
And since it is in app compo so we need pass data so we make props -> props pass thing it is the tube data is communicating in it  
  
Final step

**9. Type Narrowing**

Type narrowing is a technique in TypeScript that helps to refine a variable’s type based on certain conditions. It allows TypeScript to infer a more specific type for a variable, which helps in reducing errors and makes the code more predictable.



We should be aware about what u are passing as param and aware what is returned



This is wrong

**Why is Type Narrowing Useful?**

* **Improves Type Safety**: Type narrowing ensures you handle the right type and avoid errors that can occur when performing operations on the wrong type.
* **Helps With Union Types**: When you have union types like string | number | null | undefined, type narrowing ensures that the correct type is used in each part of the code.
* **Improves Developer Experience**: TypeScript gives you autocompletion and error checking based on the narrowed types, making development more efficient.

**10. Conditional Rendering**

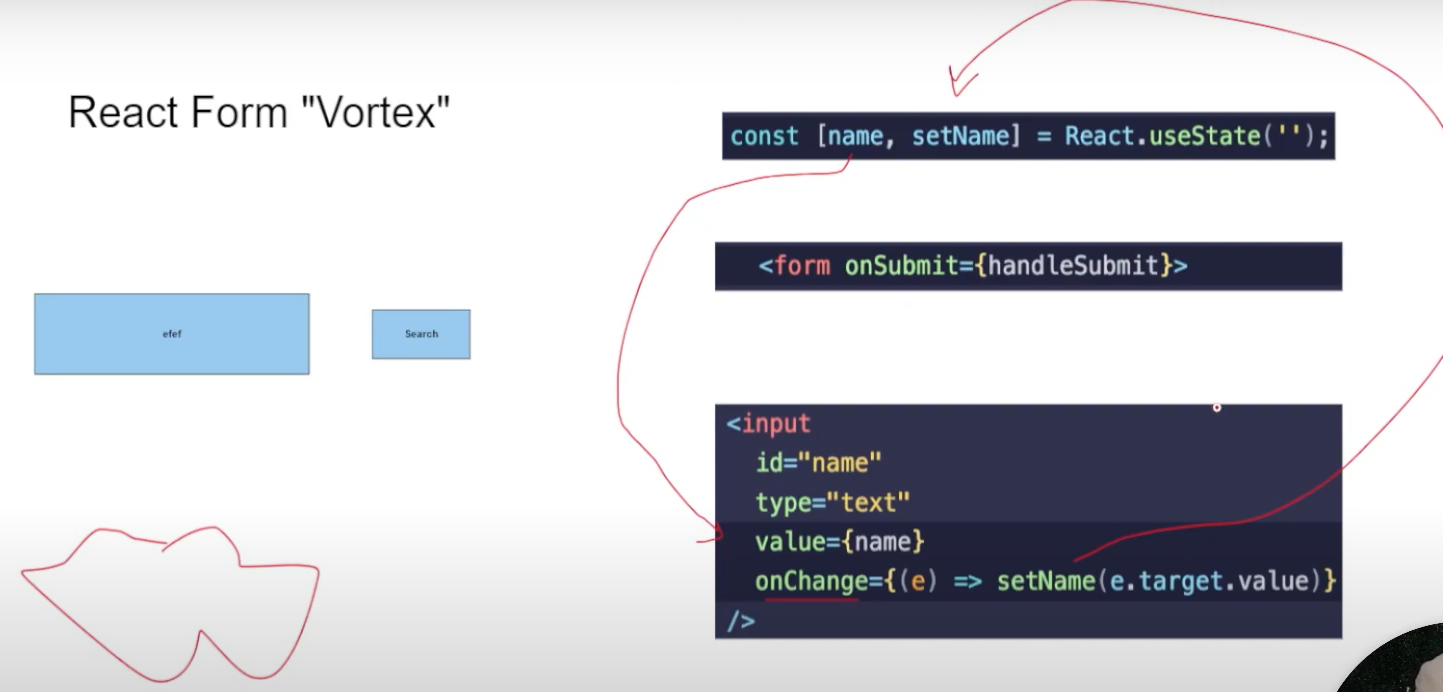
Conditional Rendering in React is a way to render different UI elements or components based on certain conditions or states. It allows you to display different content or components depending on the state of your application or the values of certain variables.

****

**11. Lists**

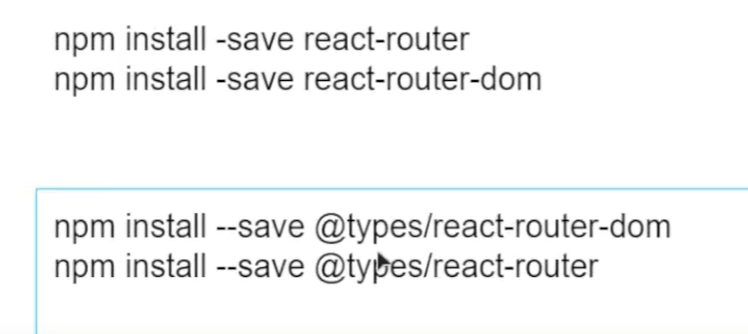
One of the most part of being se is the ability to render lists

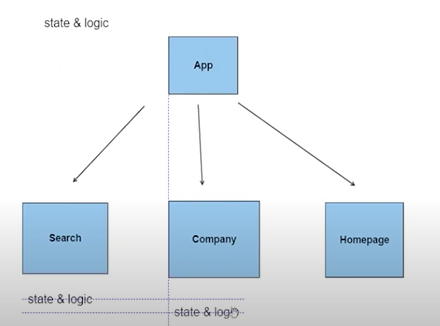
**L api ha tsir tbayen bel 3 cards…**

**12: Forms:  
**

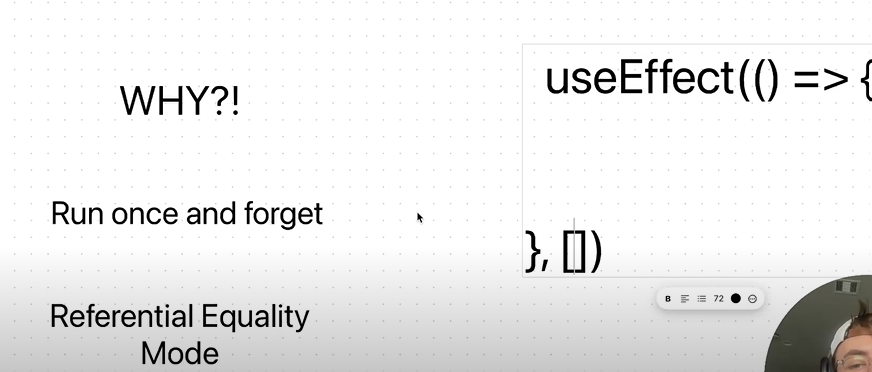
**In react, handling forms is distinct:**bs ektob bel search ef.. everytime on change is triggered and it is changing setName , and then name will be give to value (vortex) (3m ndour)

So what happens when we submit the form? We trigger another function (onSubmit)

**React router:**  
  


 sah good eno app.tsx ykoun fi logic but our app.tsx is going to be master

**17. useEffect**



Why mode is when we use effect in this form above,   
  
run once and forget is when we add an empty array metel bel soura, so the code runs one and don’t run anymore  
  
Referential equality mode, renders as the [] renders [0] changed to [1] for example

But it is Referential equality so it needs to change actual place in memory,

