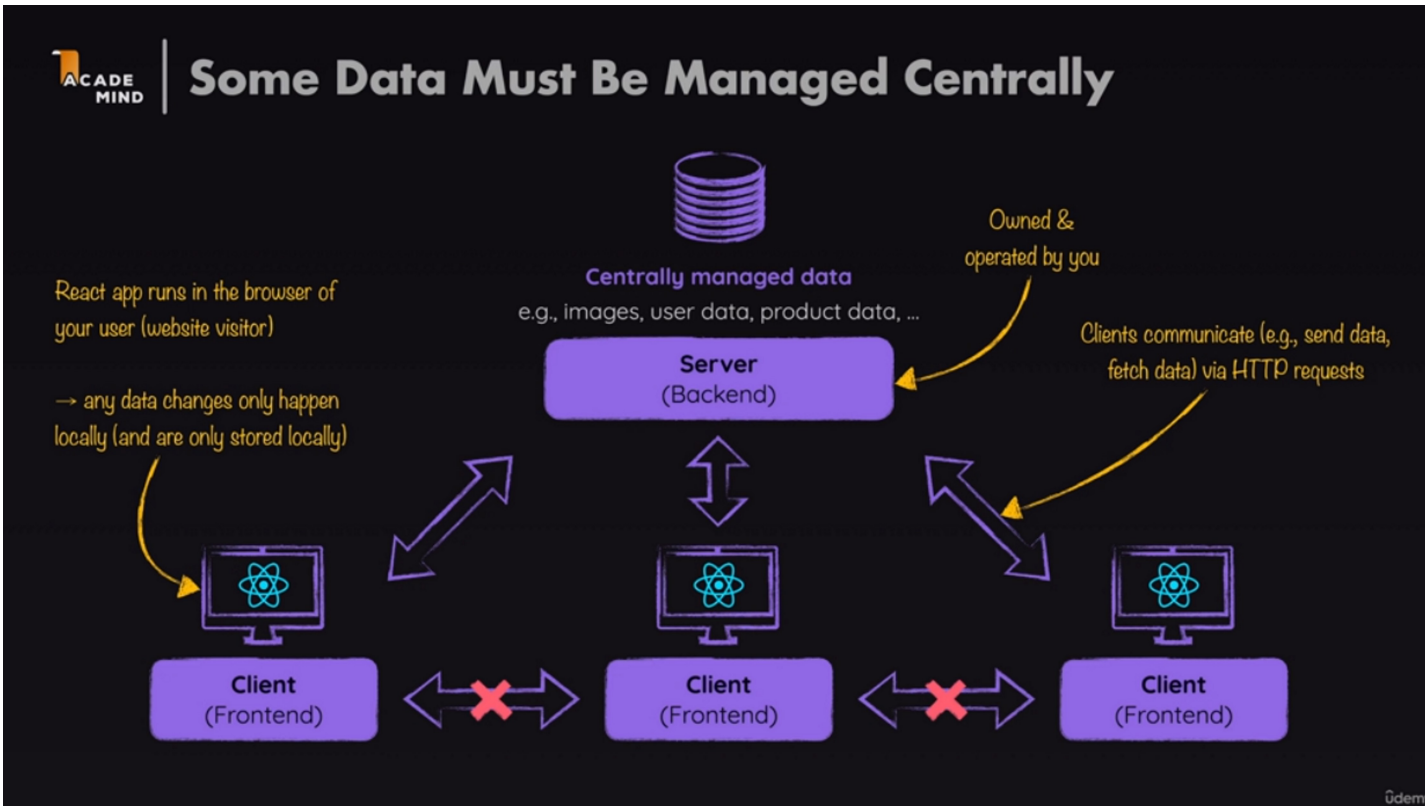
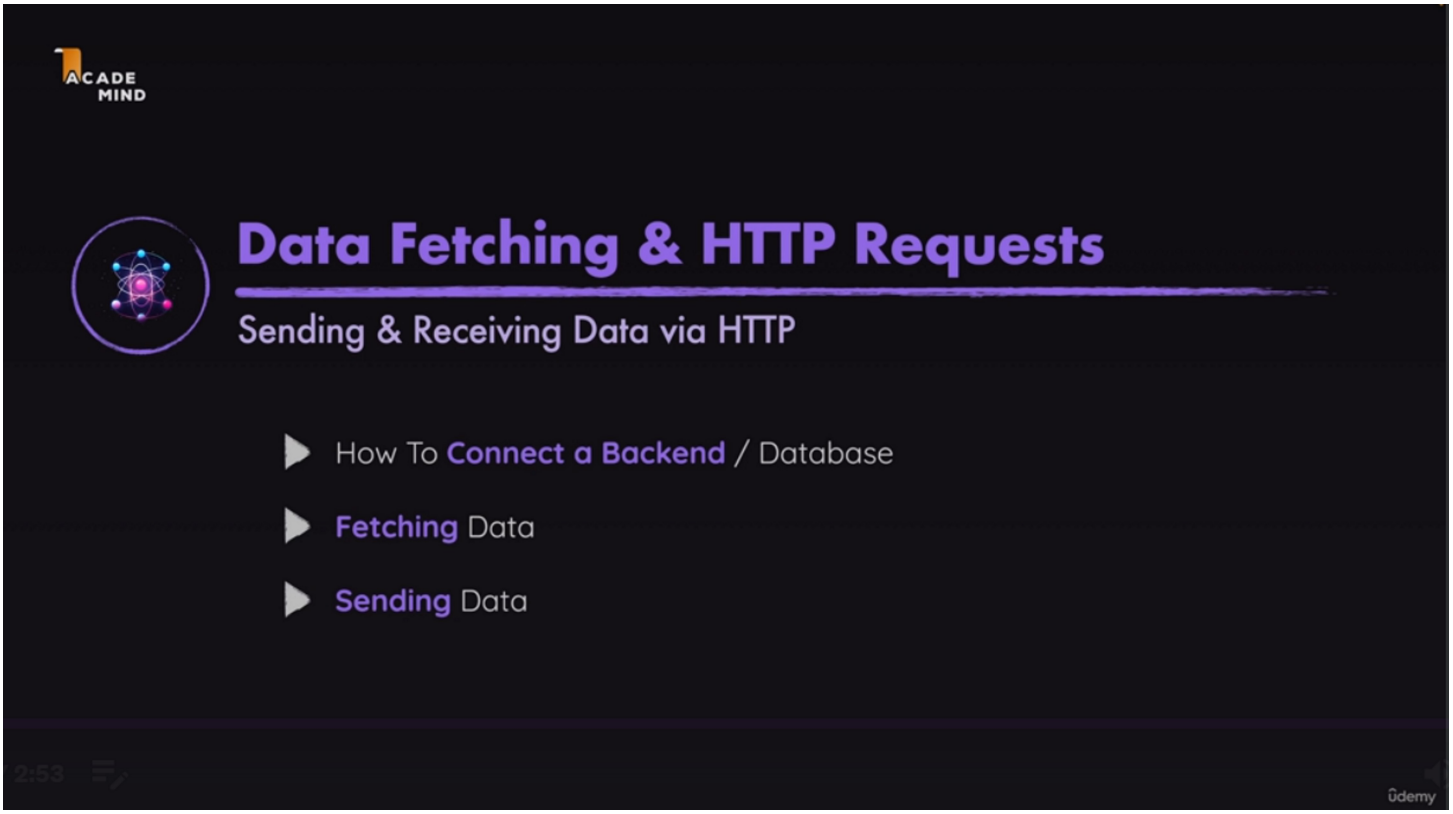
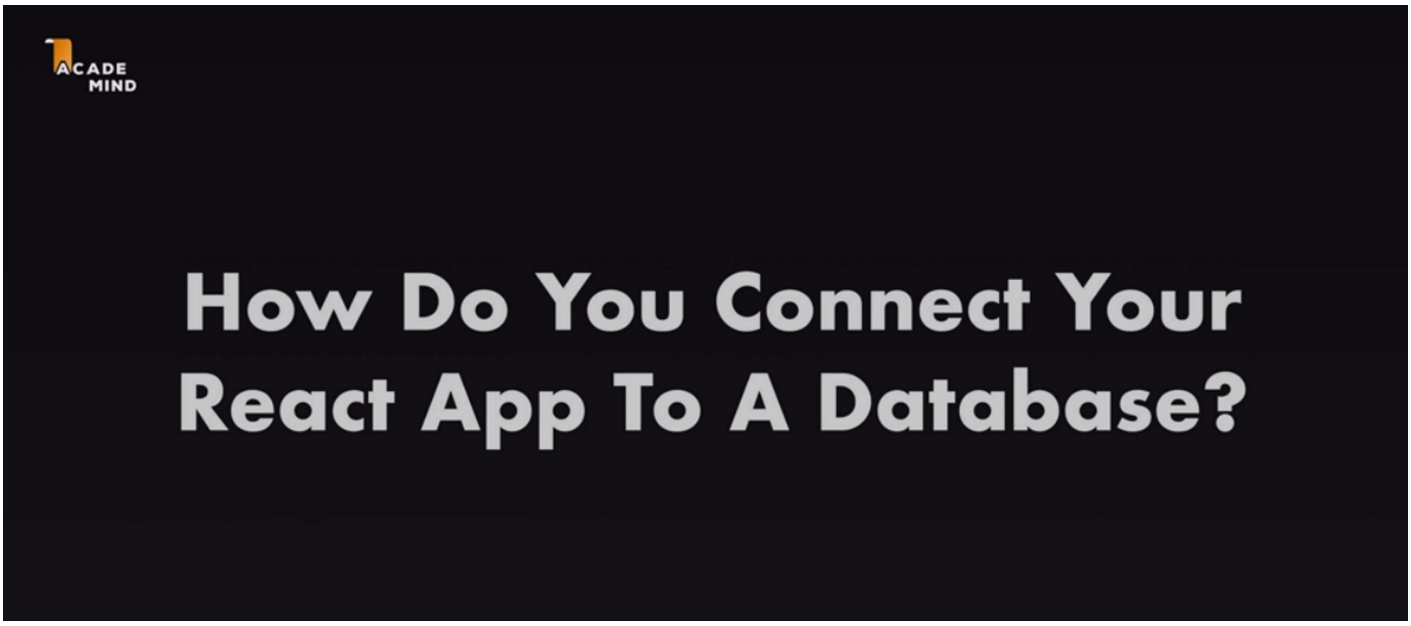
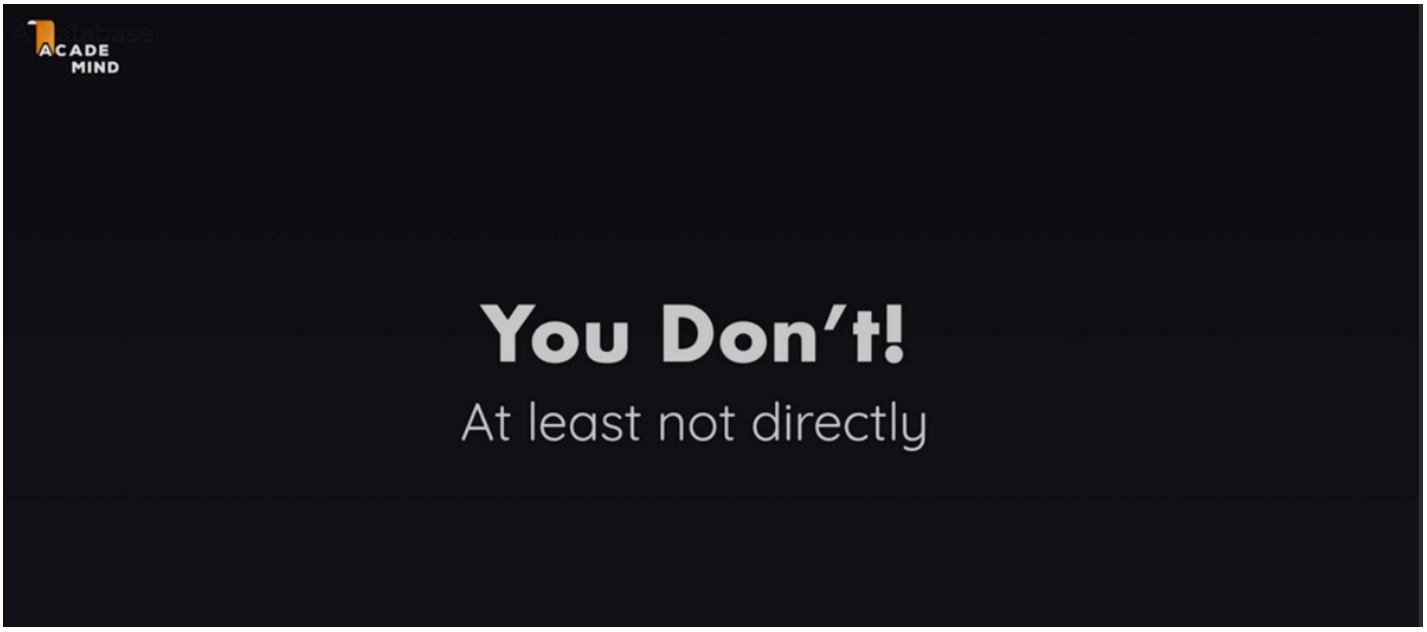
Section 15: Sending Http Requests (Connecting to a Database)

Sunday, September 29, 2024 3:01 PM






ACADEMIND
How Do You Connect Your React App To A Database?


ACADEMIND
You Don't!
At least not directly

ACADEMIND

Always keep in mind

Your React code runs in the browsers of your users

They can view that code (via the browser developer tools) if they want to!

Insecure Shop

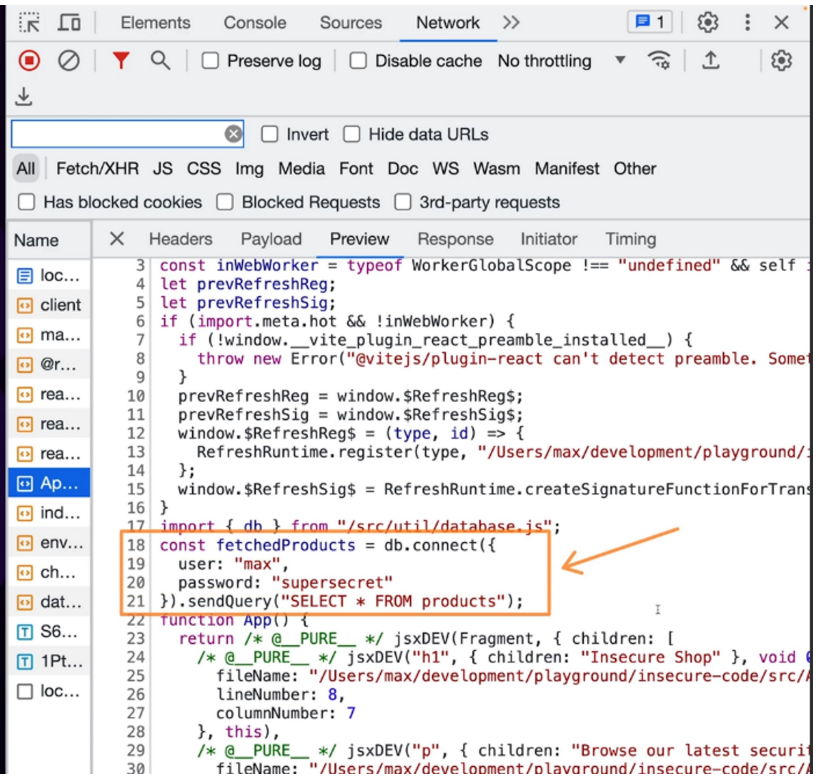
Browse our latest security-compromised products

Insecure Computer

All our PCs come with malware pre-installed!

A Guide to an Insecure Personality

This bestselling book is there to crush your self-worth and make you leave comments on the internet.

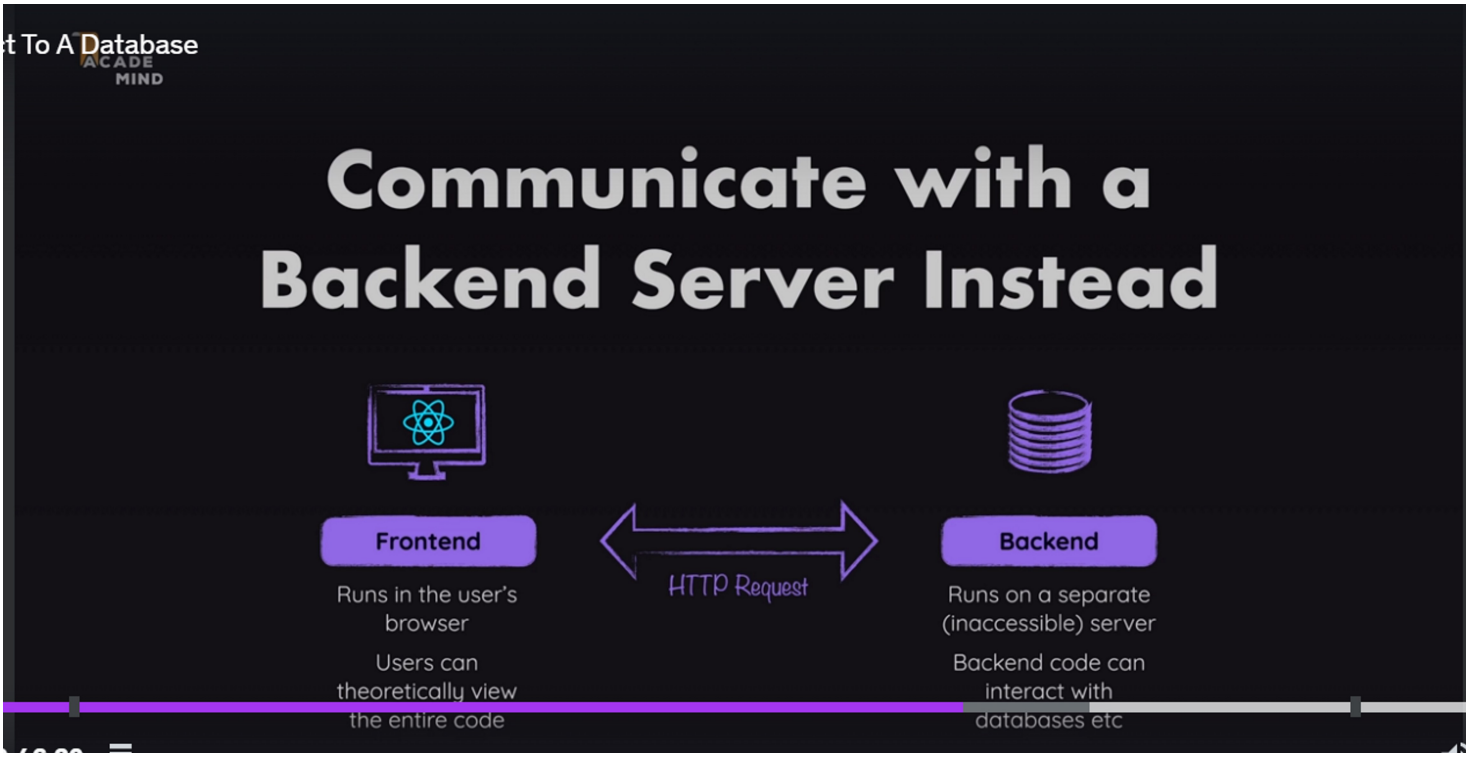


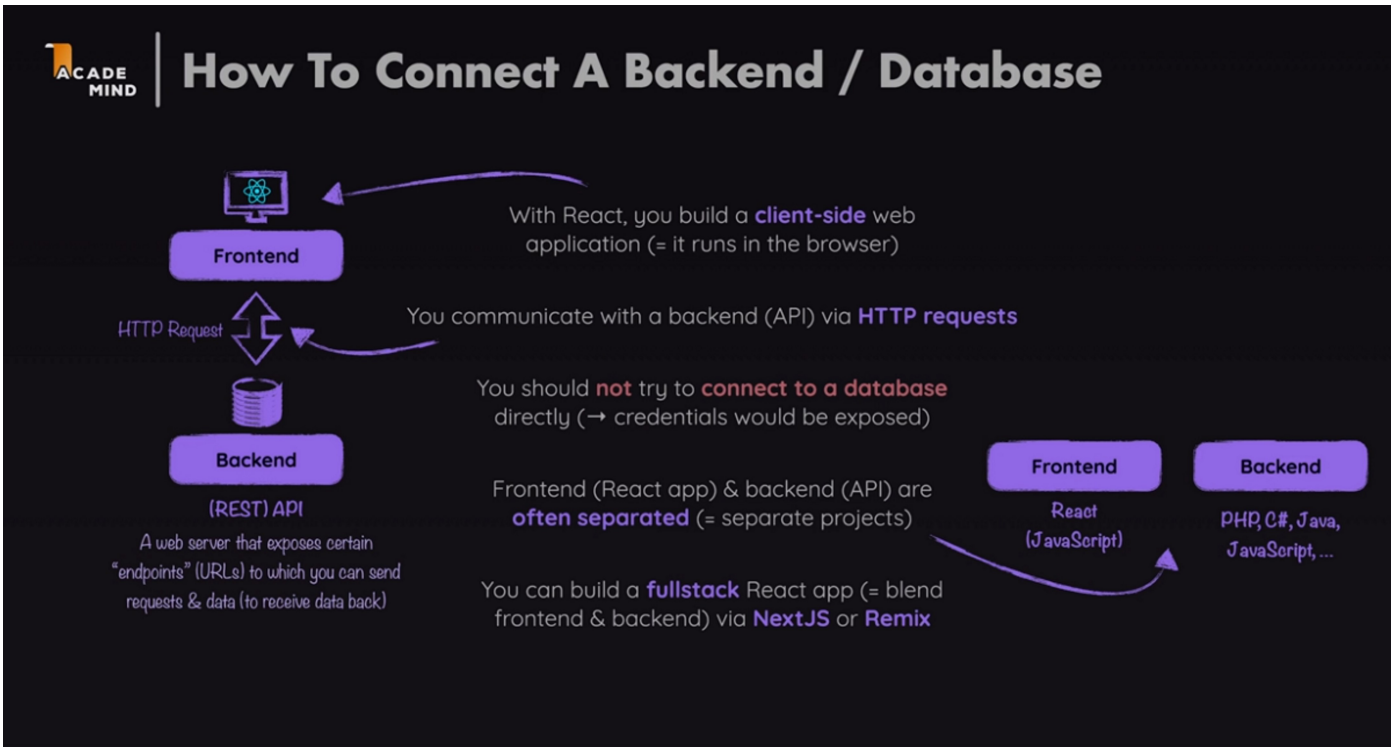
There also are technical restrictions & constraints

Not all operations can be performed in the browser

E.g, you can't access a (centrally managed) file system

Communicate with a Backend Server Instead





Now, as I already mentioned in the previous lecture, there often are good reasons for having a central server, a central backend or database to which multiple instances of the React app connect I order to share data across users.

If you are building an online store, you wanna make sure that different users from different parts of the world can access the same data and can of course also send data, can place orders, for example. And you as an owner of the shop, of course then also need access to those orders so that you can fulfill them.

Therefore, you often need such a central backend or database. And therefore, an important question is, how do you connect your React app to such a database? How can you send data to such a database and get data out of such a central database?

And the short answer is, you don't, at least not directly. You don't directly connect your React app to a database because if you were to try that, you would run into security issues. Because what's important to understand, as a web developer in general, is that when writing code that runs in the browser, which is exactly what you're doing with React, all your React application code runs in the browsers of your visitors.

If you are doing that, your visitors can access that code, they can view that code.
And if that code then, for example, contains the credentials to access a database, your database can get compromised because users can retrieve those credentials and access your database to destroy data, insert incorrect data, or do all kinds of bad things.

In addition, you also have certain restrictions when writing front-end code. So code that runs in a browser. For example, you can't easily access a file system. Especially not a shared file system that lives on some central server or computer.

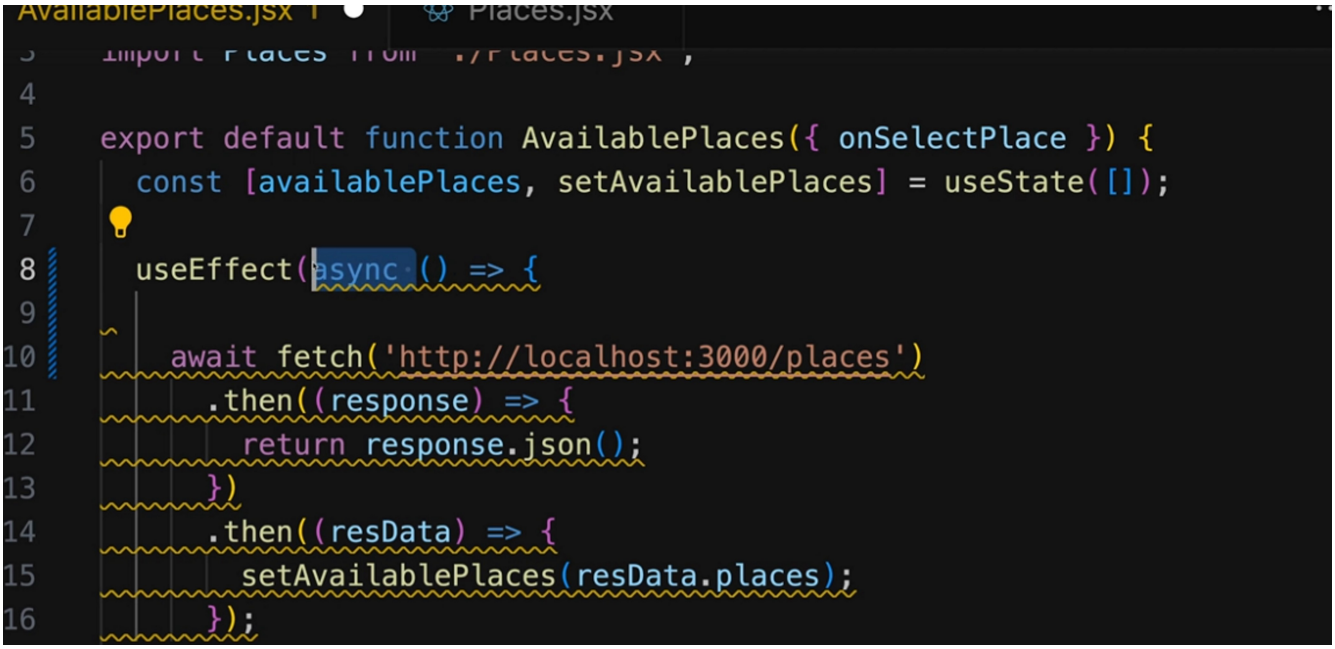
You don't really have those capabilities in the browser. And therefore, instead of directly accessing a database or some file system that may hold shared data, you instead communicate with a backend server, which kind of acts as a middleman, you could say.

So that your front-end React application, which runs on the browser, and where users can theoretically view the entire code, that application can reach out to a backend. So a separate project running on a separate server, a computer owned and operated by you, the developer.
And it's then on that backend where you can interact with databases, etcetera because that backend code is inaccessible by the users of your website because only you, the owner and developer, only you have access to the code on such a backend server.

And to connect the frontend and the backend, you use HTTP requests so that from inside your React application you, for example, send an HTTP request to the backend to request some data or to request a change to some data. But what's super important, and a safety feature here, is that you can only send HTTP requests that are allowed and accepted by this backend.

If a backend doesn't want to accept a certain request, it's simply not possible for the client to force it. This allows you to control what users can and cannot do.
To summarize: if you have a React app that needs to interact with a database, you must understand that React is used for writing **client-side code**. You are building a web application that runs in the browser, and the code can be viewed by visitors of your website.
Therefore, you communicate with a backend through HTTP requests. The backend code may interact with a database because it is not exposed to your users. **You control** what types of requests are allowed by exposing endpoints on your backend API—essentially, URLs that accept specific requests.
You **should never** try to connect directly to a database from your React code, as this would expose your database credentials. That's why it's common to have separate **frontend** and **backend** projects, which don’t need to use the same programming language.
When building a React app, you're typically using **JavaScript** for the frontend, but your backend could be built in **any language**, like PHP, C#, or JavaScript with Node.js, for example.
You can also build **full-stack React apps**, blending frontend and backend functionality while still ensuring that the backend code remains secure. Frameworks like **Next.js** or **Remix** help facilitate this.
Later in the course, there will be a thorough introduction to **Next.js**, which provides a slightly different approach to building backend-powered web applications with React.
For now, we’ll explore how to connect to a standalone backend from your React app.

NOT ALLOWED



In javascript, you can create function inside of a function.