## To get started

In a directory you want run command. It will ask you a bunch of questions, just select the default answer

>npx create-next-app@latest

Then install pnpm the npm package manager

> npm install -g pnpm

Run below to install the project’s package

>pnpm i

Run below to start the development server which will be on <http://localhost:3000>

>pnpm dev

## When to use SSR (server side rendering) vs CSR (client side rendering)

Things you want to load quickly that calls data from a database should generally be SSR, page.tsx usually is that

* One can getServerSideProps or getStatic props, but cannot use both together

Things that are interactive to users, like search.tsx should be CSR

* useEffect and useState

Deployment Notes

Need to add the environment variable in authentication

<https://nextjs.org/learn/dashboard-app/adding-authentication>

For auth to work in production, you'll need to update your environment variables in your Vercel project too. Check out this [guide](https://vercel.com/docs/projects/environment-variables) on how to add environment variables on Vercel.

## ------------------------------------

## React foundations

<https://nextjs.org/learn/react-foundations>

To use React in your newly created project, load two React scripts from an external website called unpkg.com:

* react is the core React library.
* react-dom provides DOM-specific methods that enable you to use React with the DOM.

Resources for many java script syntax

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object>

There are three core concepts of React that you'll need to be familiar with to start building React applications. These are:

Components

Props

State

To make the button do something when clicked, you can use the onClick event:

In React, event names are camelCased. The onClick event is one of many possible events you can use to respond to user interaction. For example, you can use onChange for input fields or onSubmit for forms.

Handling events

In React, event handler functions are typically defined inside your component.

function HomePage() {

// ...

function handleClick() {

console.log('increment like count');

}

return (

<div>

{/\* ... \*/}

<button onClick={handleClick}>Like</button>

</div>

);

## You can use state to store and increment the number of times a user has clicked the "Like" button. In fact, the React hook used to manage state is called: useState()

The first item in the array is the state value, which you can name anything.

The second item in the array is a function to update the value.

You can also take the opportunity to add the initial value of your likes state to 0:

## const [likes, setLikes] = React.useState(0);

When you use Next.js in your project, you do not need to load the react and react-dom scripts from unpkg.com anymore. Instead, you can install these packages locally using npm or your preferred package manager.

Once you install React with npm install react@latest react-dom@latest next@latest

You put your indext.html file into folder /app/page.js

--modify the code –

In terminal run >npm run dev

In browser go to localhost:3000

<https://react.dev/>

Dashboard App course

<https://nextjs.org/learn/dashboard-app>

npm install -g pnpm

You want to add global css at the top layer of the app. So in the layout.tsx file:

Import ‘@/app/ui/global.css’;

Tailwind, and its utility classes

<https://tailwindcss.com/docs/utility-first>

## how did that just installed all the tailwindscss?? And got that tailwinds.config.ts is in the root folder nextjs-dashboard

## in the app/ui/folder, that’s where you keep the global.css file

database name

neon-lightBlue-school

Finally, run **pnpm i @vercel/postgres** in your terminal to install the Vercel Postgres SDK.

How to fetch data

API layer

APIs are an intermediary layer between your application code and database. There are a few cases where you might use an API:

If you're using 3rd party services that provide an API.

If you're fetching data from the client, you want to have an API layer that runs on the server to avoid exposing your database secrets to the client.

In Next.js, you can create API endpoints using [**Route Handlers**](https://nextjs.org/docs/app/building-your-application/routing/route-handlers).

Route.ts file can be nested anywhere inside /app directory, but can’t be in the samd route segment level as page.js

## Mutating data

In React, you can use the action attribute in the <form> element to invoke actions. The action will automatically receive the native FormData object, containing the captured data.

Good to know: In HTML, you'd pass a URL to the action attribute. This URL would be the destination where your form data should be submitted (usually an API endpoint).

However, in React, the action attribute is considered a special prop - meaning React builds on top of it to allow actions to be invoked.

Behind the scenes, Server Actions create a POST API endpoint. This is why you don't need to create API endpoints manually when using Server Actions.

## Data Validation

To handle type validation, you have a few options. While you can manually validate types, using a type validation library can save you time and effort. For your example, we'll use Zod, a TypeScript-first validation library that can simplify this task for you.

**Hierarchy**