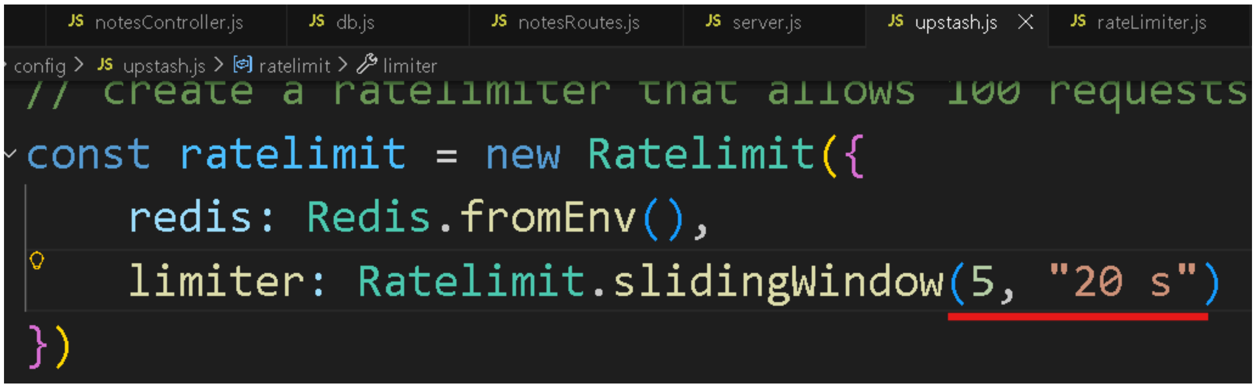
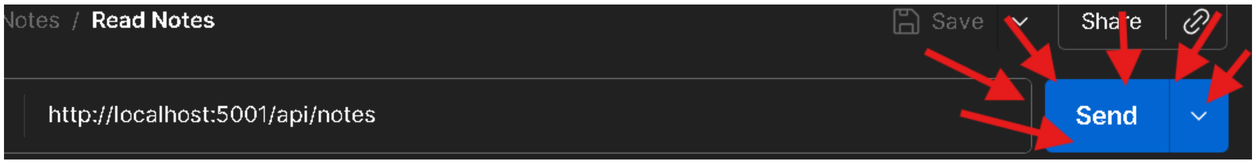
For test:
We get an error if we hit 5 time, send button within 20 seconds.

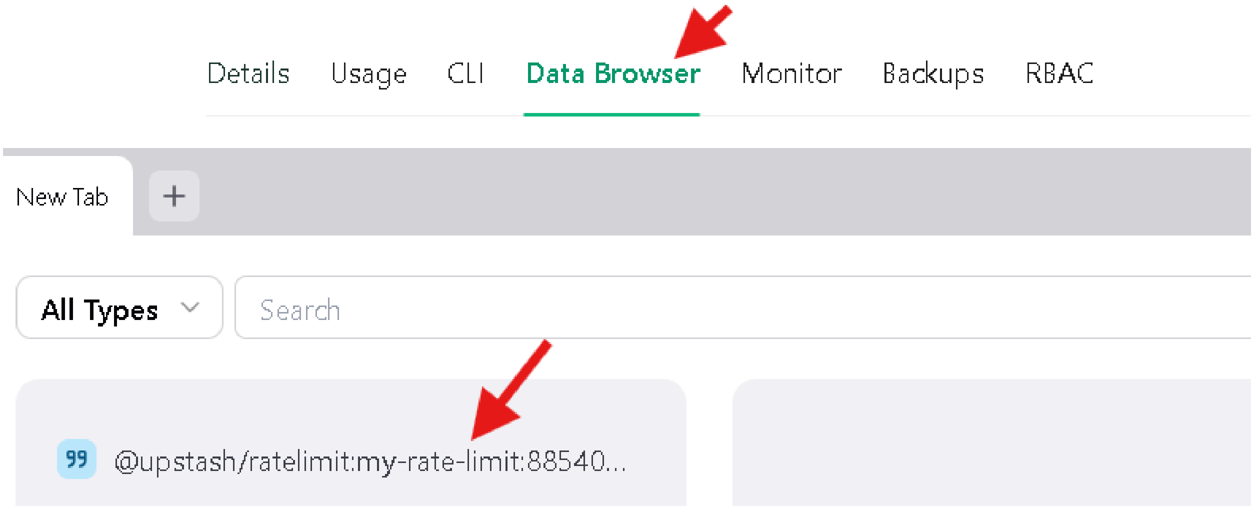


```
// create a ratelimiter that allows 100 requests
const ratelimit = new Ratelimit({
    redis: Redis.fromEnv(),
    limiter: Ratelimit.slidingWindow(5, "20 s")
})
```

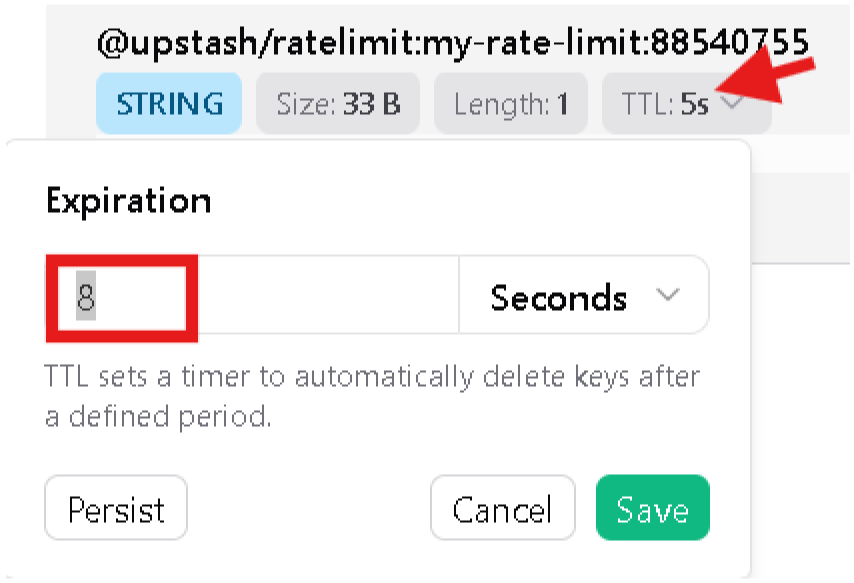
Click send button more than 5 times, in postman:



http://localhost:5001/api/notes



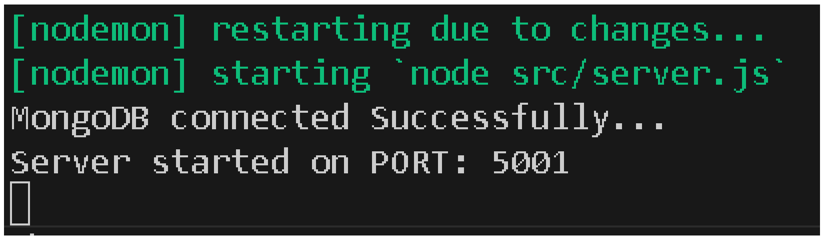
@upstash/ratelimit:my-rate-limit:88540...


@upstash/ratelimit:my-rate-limit:88540755

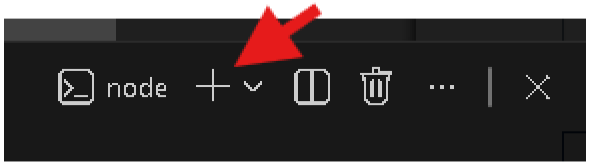
STRING
Size: 33 B
Length: 1
TTL: 5s
Expiration
8
Seconds
TTL sets a timer to automatically delete keys after a defined period.
Persist
Cancel
Save

TTL: reduces its seconds.

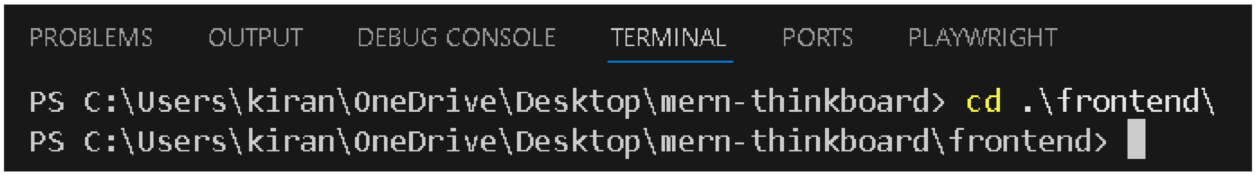
In terminal, keep backend running:


```
[nodemon] restarting due to changes...
[nodemon] starting `node src/server.js`
MongoDB connected Successfully...
Server started on PORT: 5001
```

Now lets start creating frontend, create new terminal: Click plus sign

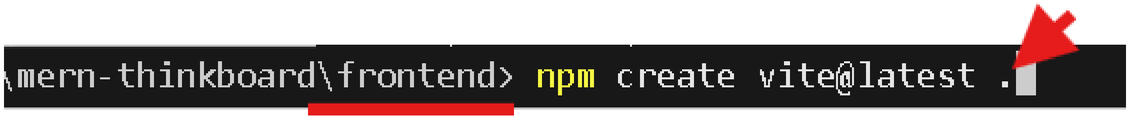


Move to the frontend:

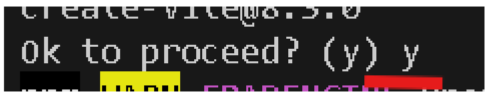


```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard> cd .\frontend\
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend>
```

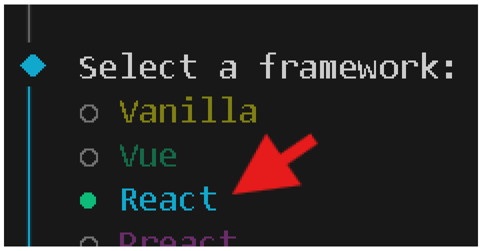
Now let's create a react application, using vite:
Initializing in the current folder(frontend folder): so we use dot at the end.
Command used: **npm create vite@latest .**


```
\mern-thinkboard\frontend> npm create vite@latest .
```

Click **y:**


```
Ok to proceed? (y) y
```

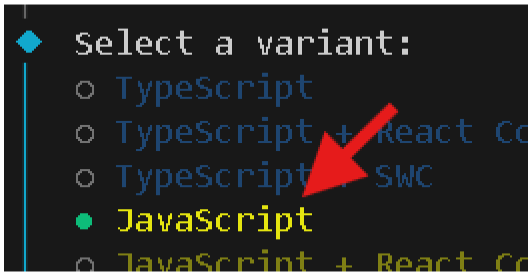
Then choose:



```
◆  Select a framework:
   ○ Vanilla
   ○ Vue
   ● React
   ○ Preact
```

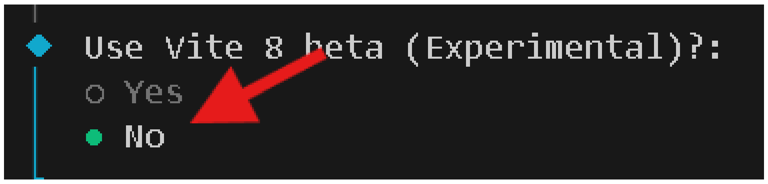
then click enter

Choose JavaScript:



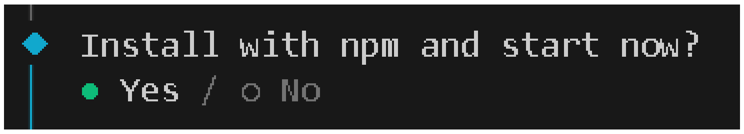
```
◆  Select a variant:
   ○ TypeScript
   ○ TypeScript + React Co
   ○ TypeScript + SWC
   ● JavaScript
   ○ JavaScript + React Co
```

then click enter



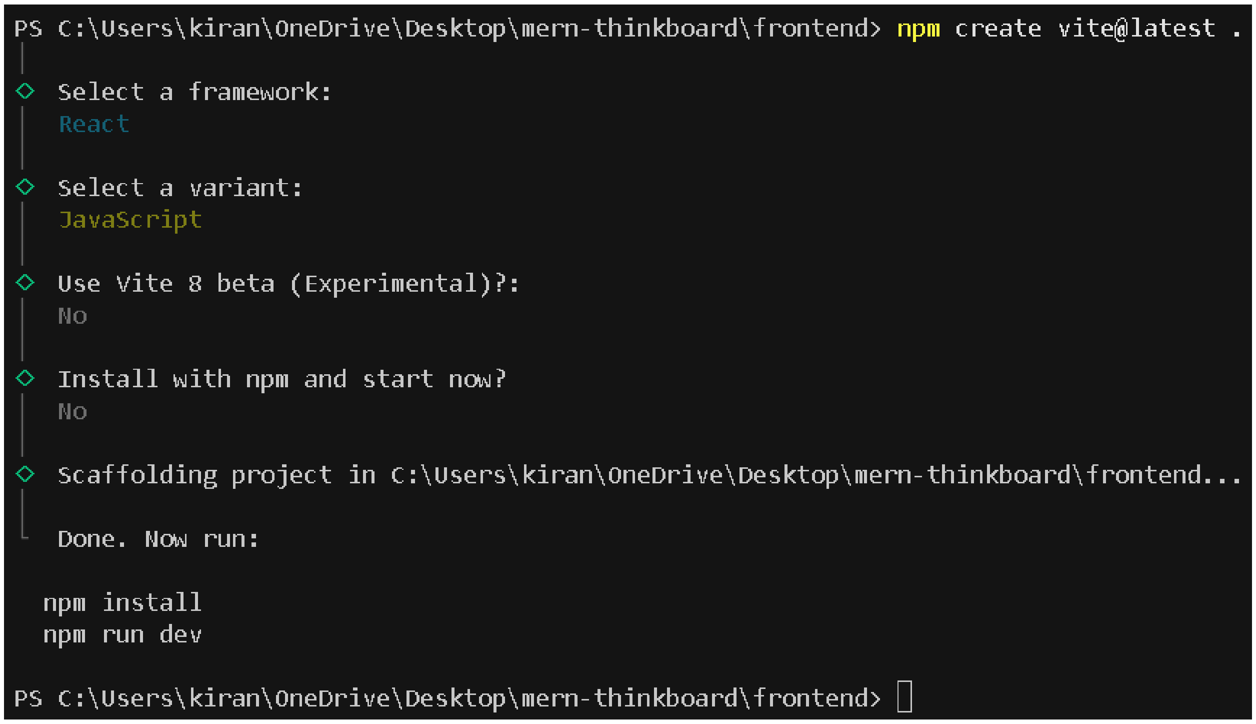
```
◆  Use Vite 8 beta (Experimental)?:
   ○ Yes
   ● No
```

choose **No,** then click enter



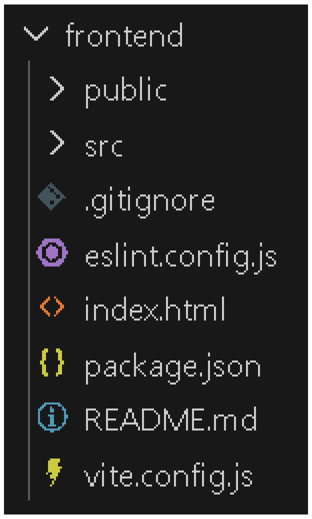
```
◆  Install with npm and start now?
   ● Yes / ○ No
```

choose **No,** then click enter



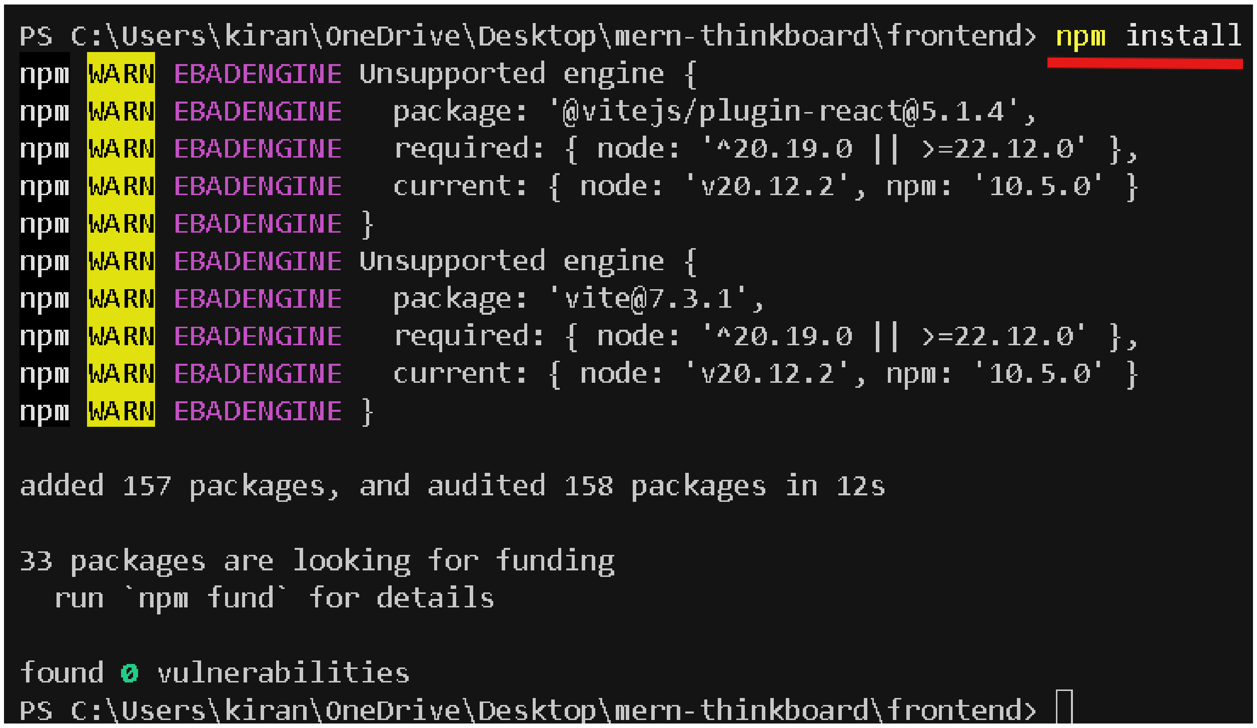
```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm create vite@latest .
|
◇  Select a framework:
|  React
|
◇  Select a variant:
|  JavaScript
|
◇  Use Vite 8 beta (Experimental)?:
|  No
|
◇  Install with npm and start now?
|  No
|
◇  Scaffolding project in C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend...
|
└  Done. Now run:

  npm install
  npm run dev

PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> 
```

Then frontend folder has:



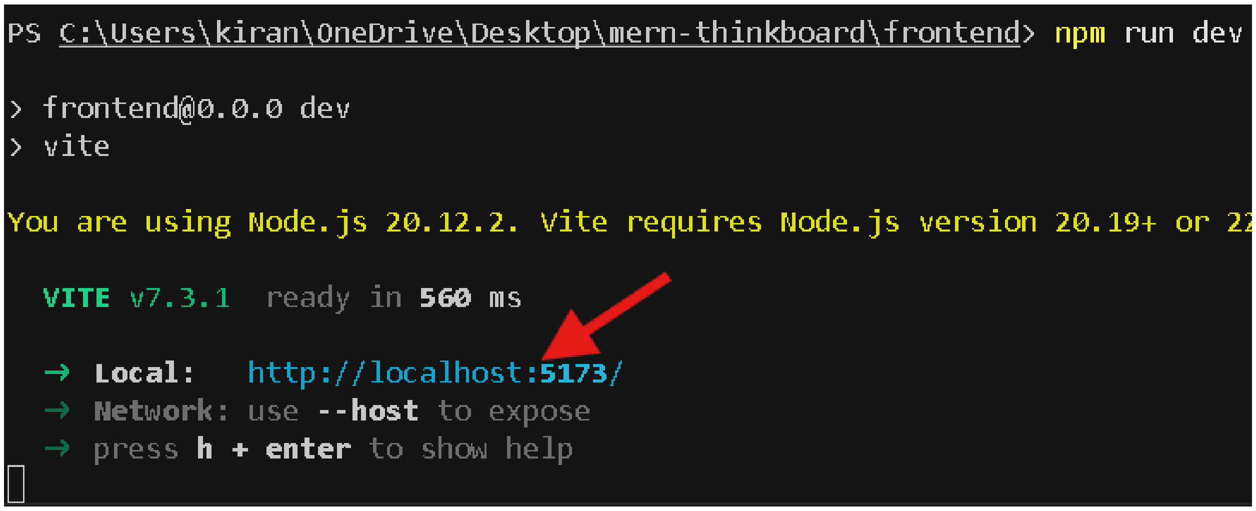
```
frontend
  public
  src
  .gitignore
  eslint.config.js
  index.html
  package.json
  README.md
  vite.config.js
```

To install all the dependencies: npm install



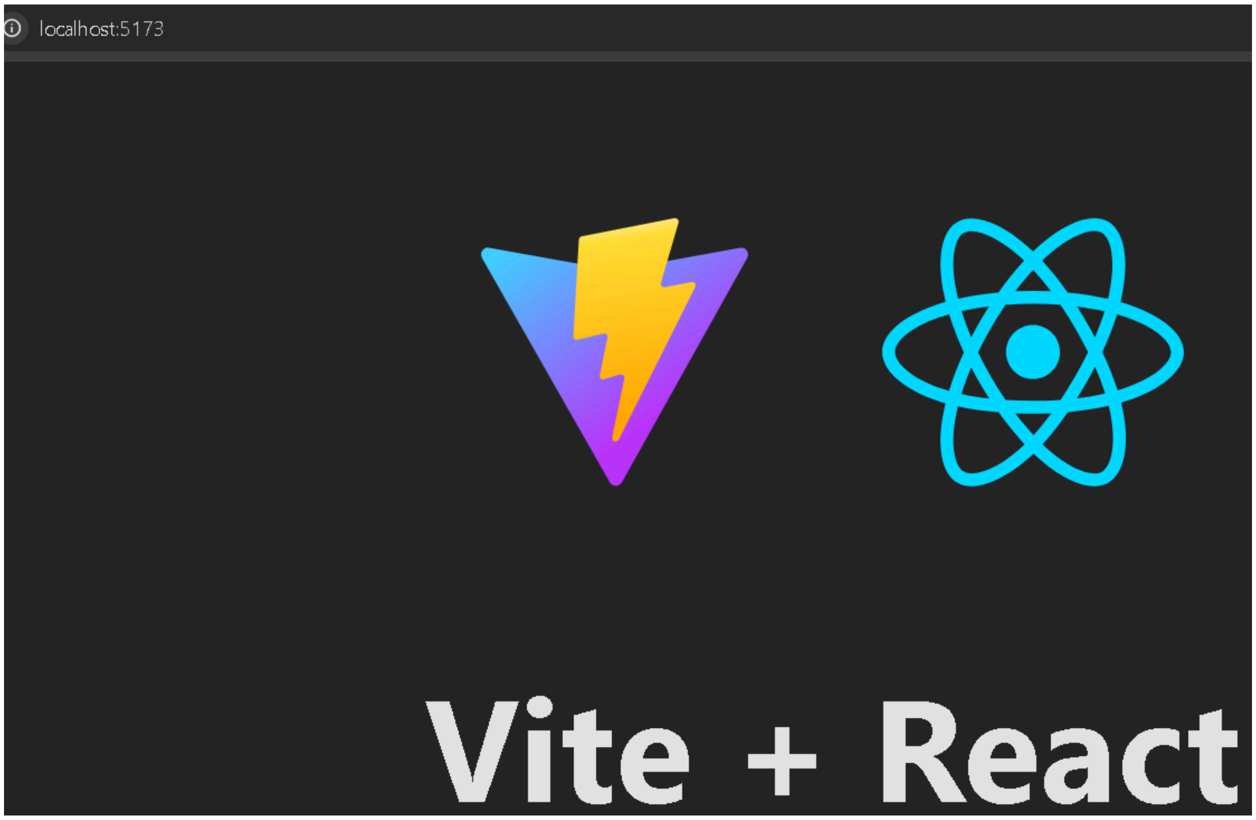
```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm install
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@vitejs/plugin-react@5.1.4',
npm WARN EBADENGINE   required: { node: '^20.19.0 || >=22.12.0' },
npm WARN EBADENGINE   current: { node: 'v20.12.2', npm: '10.5.0' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'vite@7.3.1',
npm WARN EBADENGINE   required: { node: '^20.19.0 || >=22.12.0' },
npm WARN EBADENGINE   current: { node: 'v20.12.2', npm: '10.5.0' }
npm WARN EBADENGINE }

added 157 packages, and audited 158 packages in 12s

33 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend>
```

Then we can run the app using: npm run dev



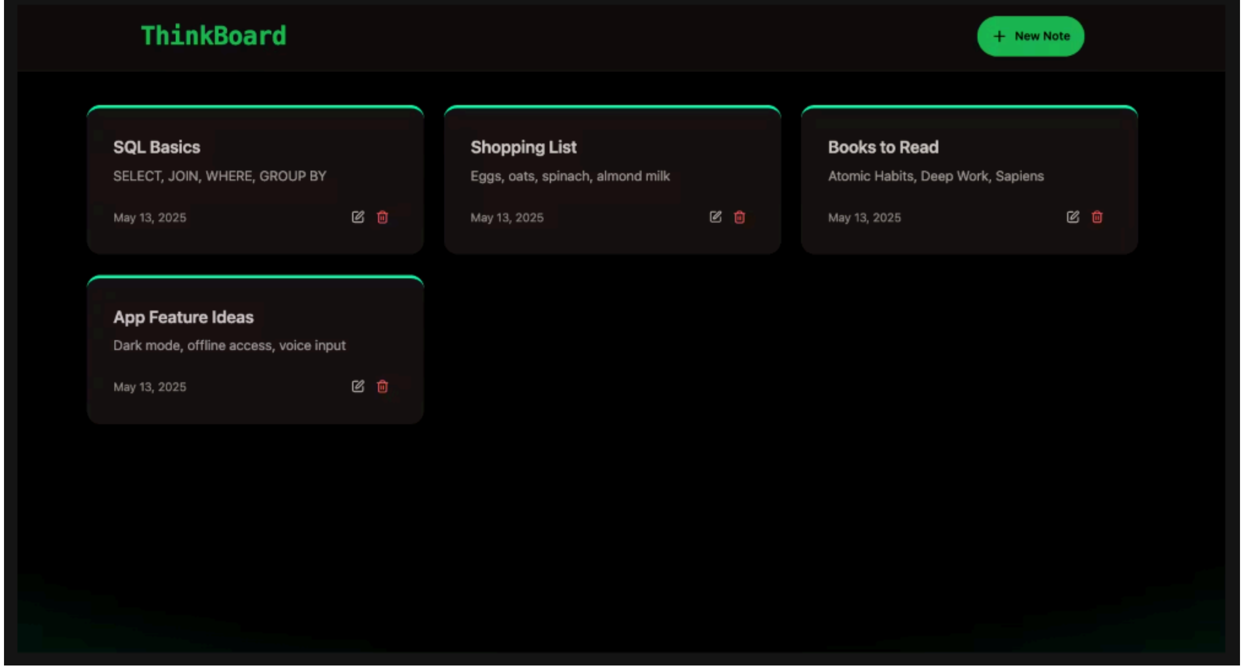
```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm run dev

> frontend@0.0.0 dev
> vite

You are using Node.js 20.12.2. Vite requires Node.js version 20.19+ or 22

  VITE v7.3.1  ready in 560 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Ctrl + Click on the link as shown in above image.
Open the link in : http://localhost:5173/



There are three type of pages:

HOME PAGE 🏠



ThinkBoard

+ New Note

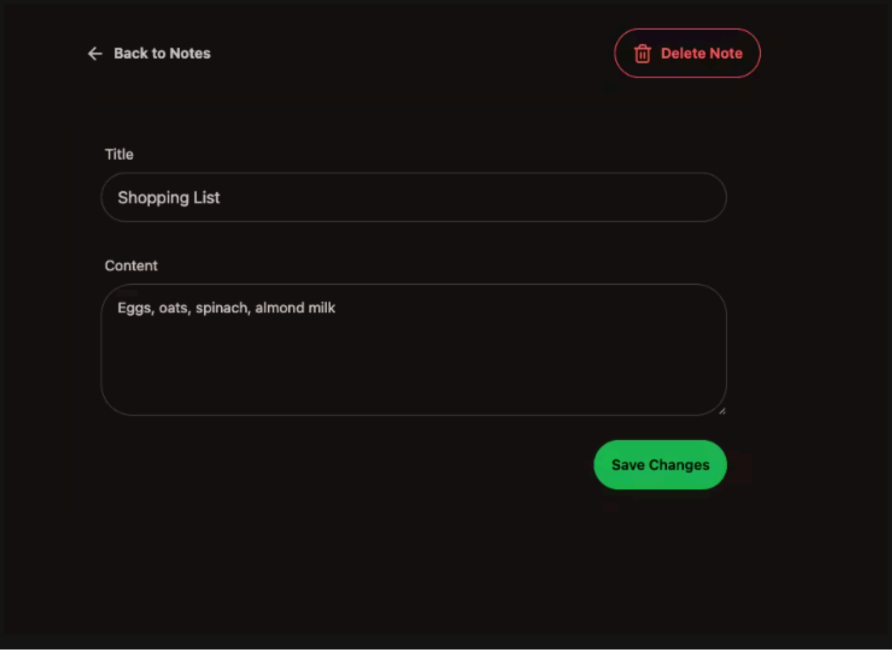
SQL Basics
SELECT, JOIN, WHERE, GROUP BY
May 13, 2025

Shopping List
Eggs, oats, spinach, almond milk
May 13, 2025

Books to Read
Atomic Habits, Deep Work, Sapiens
May 13, 2025

App Feature Ideas
Dark mode, offline access, voice input
May 13, 2025

NOTE DETAIL PAGE ✨



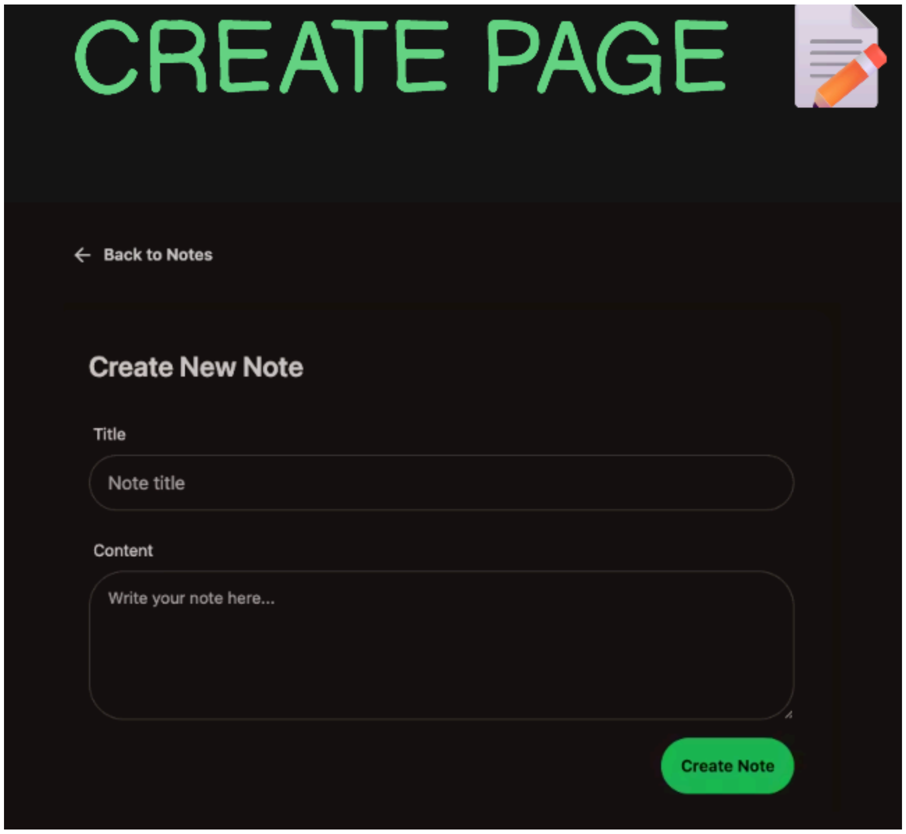
← Back to Notes

Delete Note

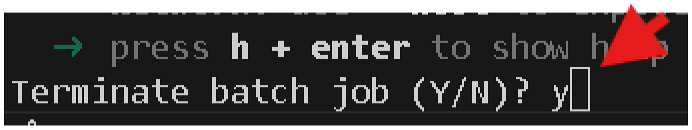
Title: Shopping List

Content: Eggs, oats, spinach, almond milk

Save Changes



To create these pages, we use react routes.
Stop the frontend:
Click: Ctrl + c
Click: y

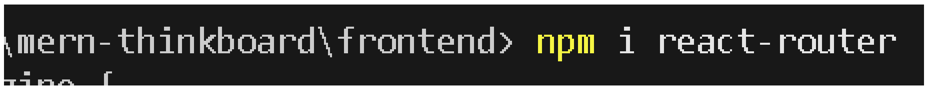


```
  → press h + enter to show help
Terminate batch job (Y/N)? y
```

Click: enter

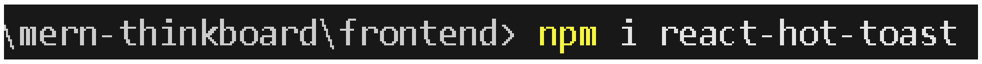
Install Dependencies:

1. Install: **npm i react-router**



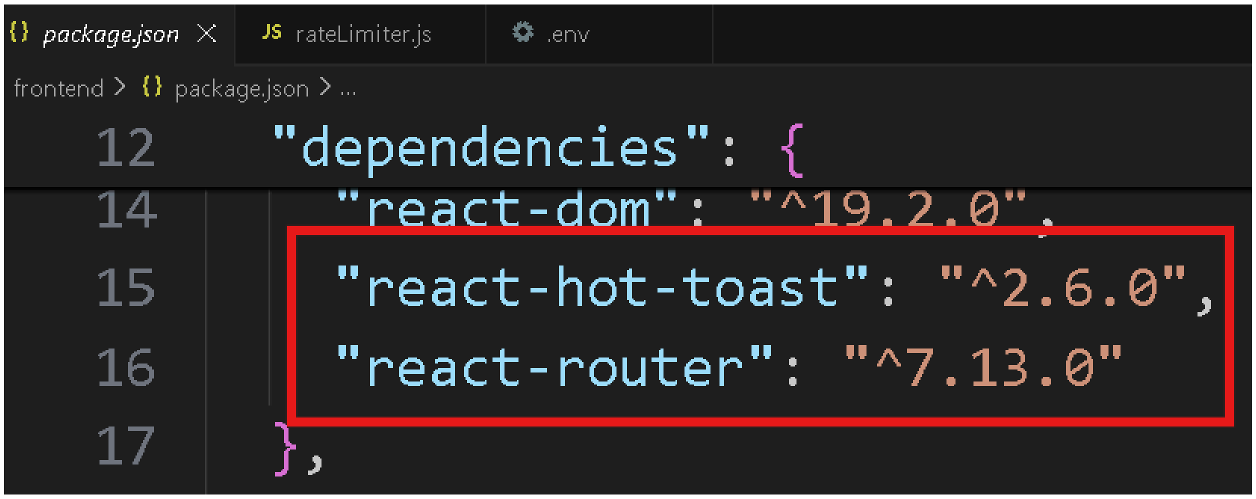
```
\mern-thinkboard\frontend> npm i react-router
```

2. **For notification:** npm i react-hot-toast



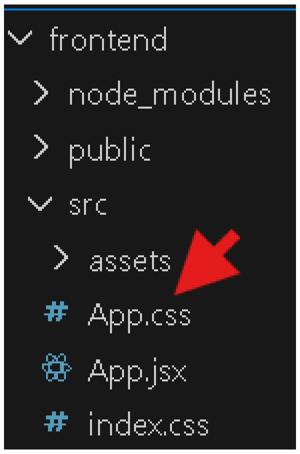
```
\mern-thinkboard\frontend> npm i react-hot-toast
```

In frontend, package.json:

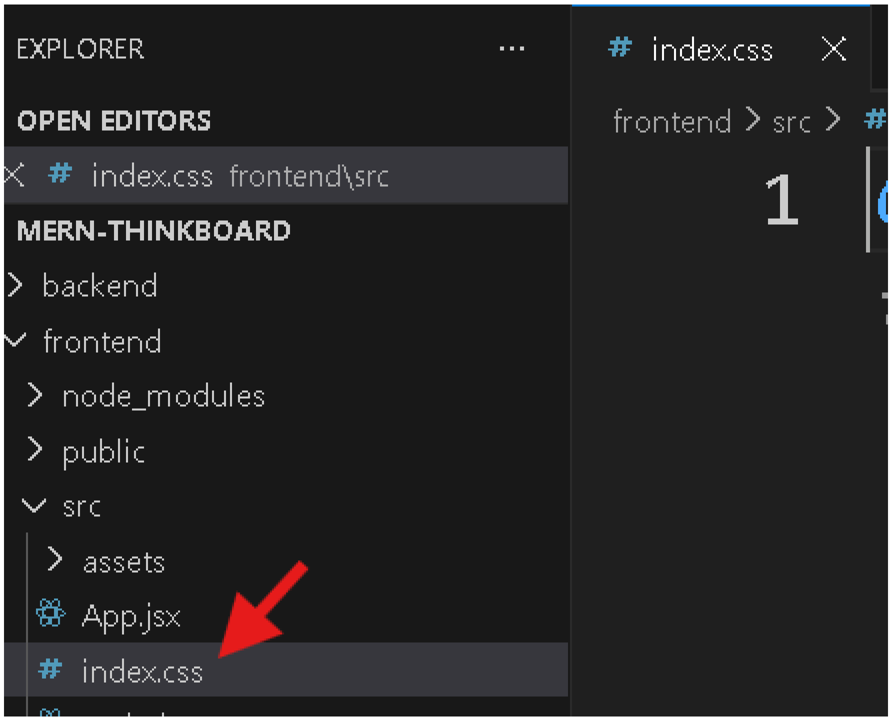


```json
"dependencies": {
  "react-dom": "^19.2.0",
  "react-hot-toast": "^2.6.0",
  "react-router": "^7.13.0"
},
```

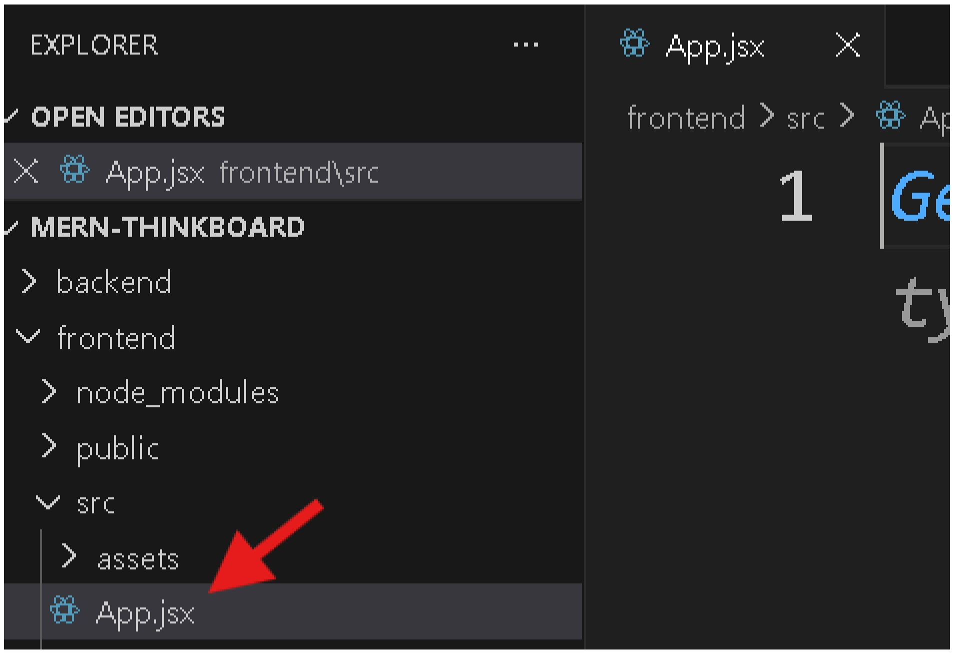
Delete App.css:



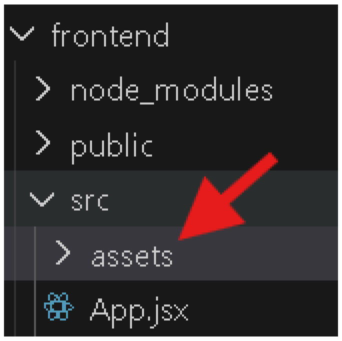
Empty index.css:



Empty App.jsx:

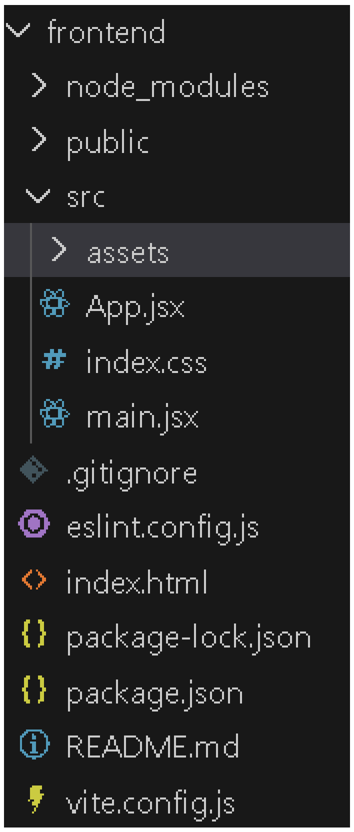


Delete assets:



```
frontend
  node_modules
  public
  src
    assets
    App.jsx
```

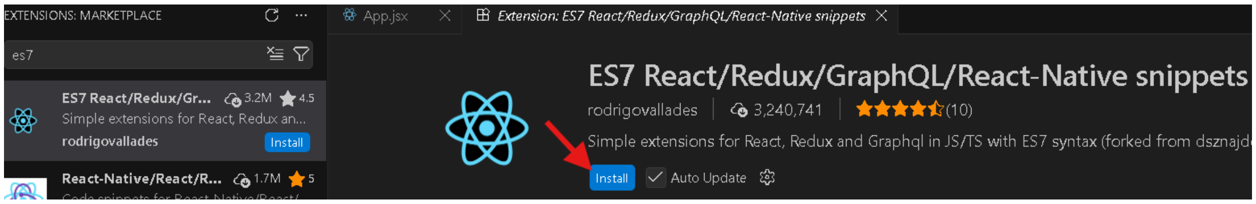
Finally , folder and file structure:



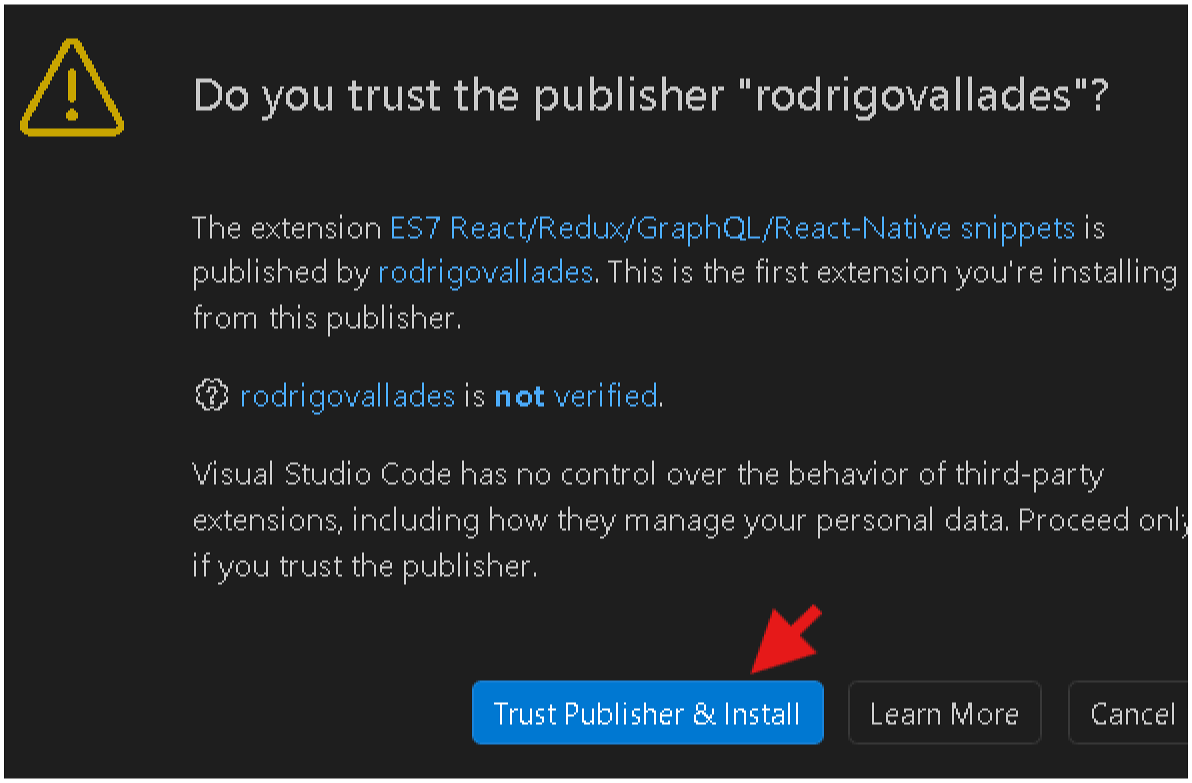
```
frontend
  node_modules
  public
  src
    assets
    App.jsx
    index.css
    main.jsx
  .gitignore
  eslint.config.js
  index.html
  package-lock.json
  package.json
  README.md
  vite.config.js
```

Search: es7

And install it



EXTENSIONS: MARKETPLACE
es7
ES7 React/Redux/Gr... 3.2M 4.5
Simple extensions for React, Redux an...
rodrigovallades
Install
React-Native/React/R... 1.7M 5
App.jsx
Extension: ES7 React/Redux/GraphQL/React-Native snippets
ES7 React/Redux/GraphQL/React-Native snippets
rodrigovallades | 3,240,741 | (10)
Simple extensions for React, Redux and Graphql in JS/TS with ES7 syntax (forked from dsznajd
Install
Auto Update



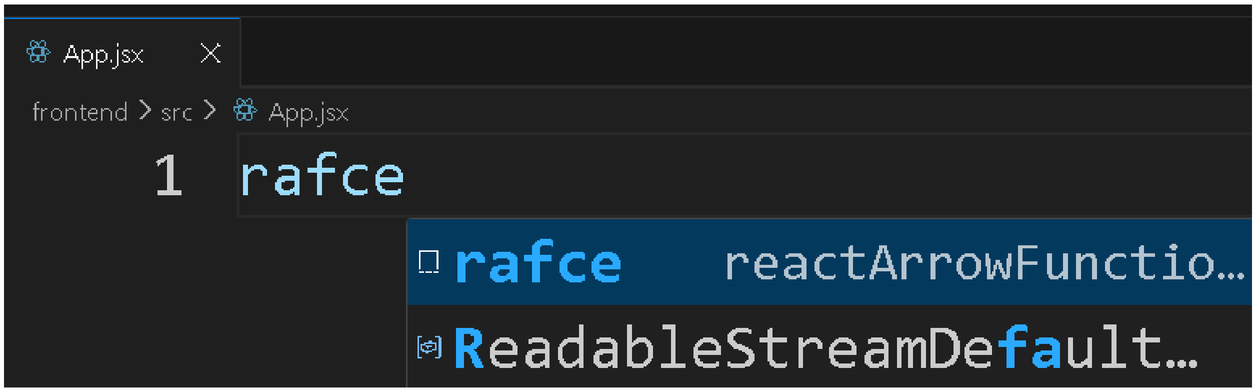
Do you trust the publisher "rodrigovallades"?

The extension ES7 React/Redux/GraphQL/React-Native snippets is published by rodrigovallades. This is the first extension you're installing from this publisher.

rodrigovallades is **not** verified.

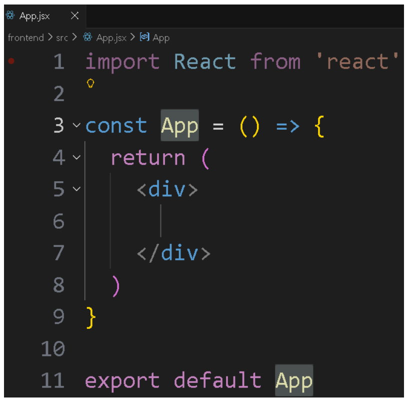
Visual Studio Code has no control over the behavior of third-party extensions, including how they manage your personal data. Proceed onl if you trust the publisher.

Then in App.jsx:
Type: rafce



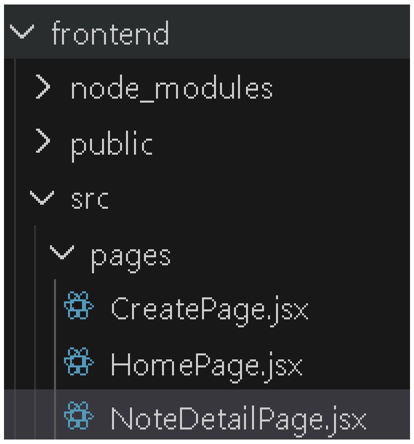
```
rafce
```

Then click enter: Which automatically creates the code for us.



```
import React from 'react'

const App = () => {
  return (
    <div>

    </div>
  )
}

export default App
```

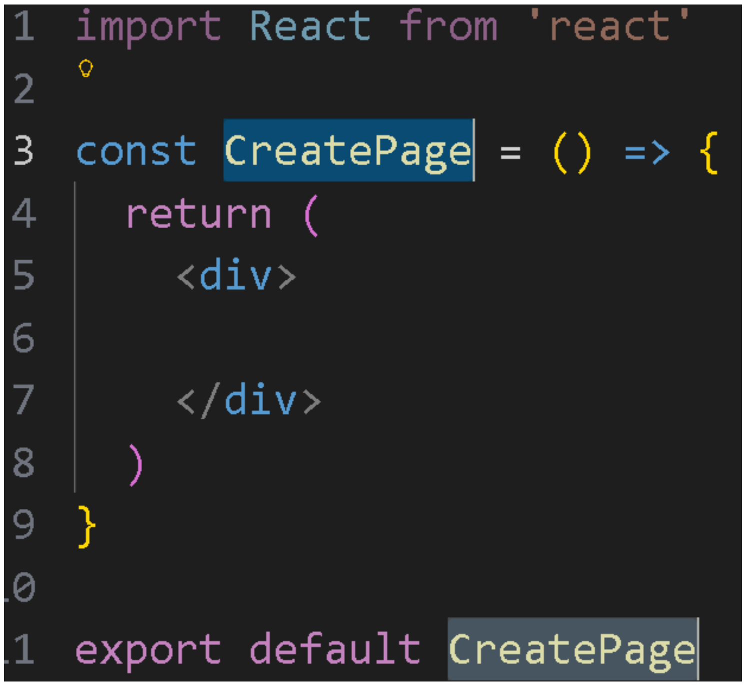
Now create pages, create HomePage.jsx in pages folder which is within src folder:



```
frontend
  node_modules
  public
  src
    pages
      CreatePage.jsx
      HomePage.jsx
      NoteDetailPage.jsx
```

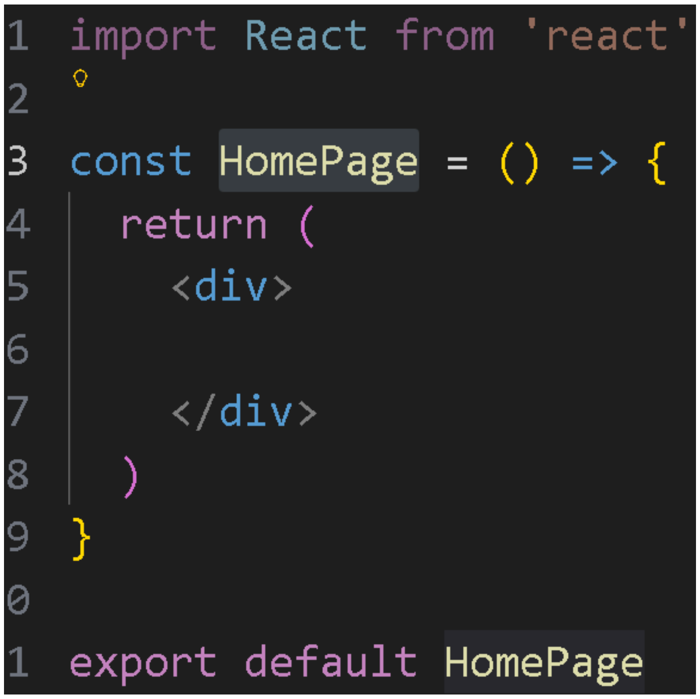
Type: rafce in all three (**.jsx**) file: then click enter.

Initial Code in CreatePage.jsx: type **rafce** then click enter



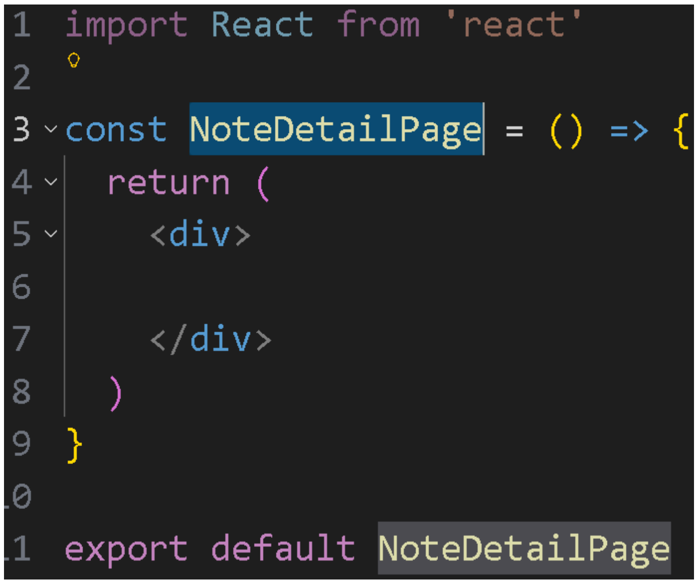
```jsx
import React from 'react'

const CreatePage = () => {
  return (
    <div>

    </div>
  )
}

export default CreatePage
```

Initial Code in HomePage.jsx: type **rafce** then click enter



```jsx
import React from 'react'

const HomePage = () => {
  return (
    <div>

    </div>
  )
}

export default HomePage
```

Initial Code in NoteDetailPage.jsx: type **rafce** then click enter



```jsx
import React from 'react'

const NoteDetailPage = () => {
  return (
    <div>

    </div>
  )
}

export default NoteDetailPage
```

Code in App.jsx:

```
import React from 'react'
import { Route, Routes } from 'react-router'
import HomePage from './pages/HomePage'
import CreatePage from './pages/CreatePage'
import NoteDetailPage from './pages/NoteDetailPage'

const App = () => {
  return (
    <div>

    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/create" element={<CreatePage />} />
      {/* :id <- is dynamic */}
      <Route path="/note/:id" element={<NoteDetailPage />} />
    </Routes>

    </div>
  )
}

export default App
```

Code in HomePage.jsx:

```
import React from 'react'

const HomePage = () => {
  return (
    <div>
      HomePage
    </div>
  )
}

export default HomePage
```

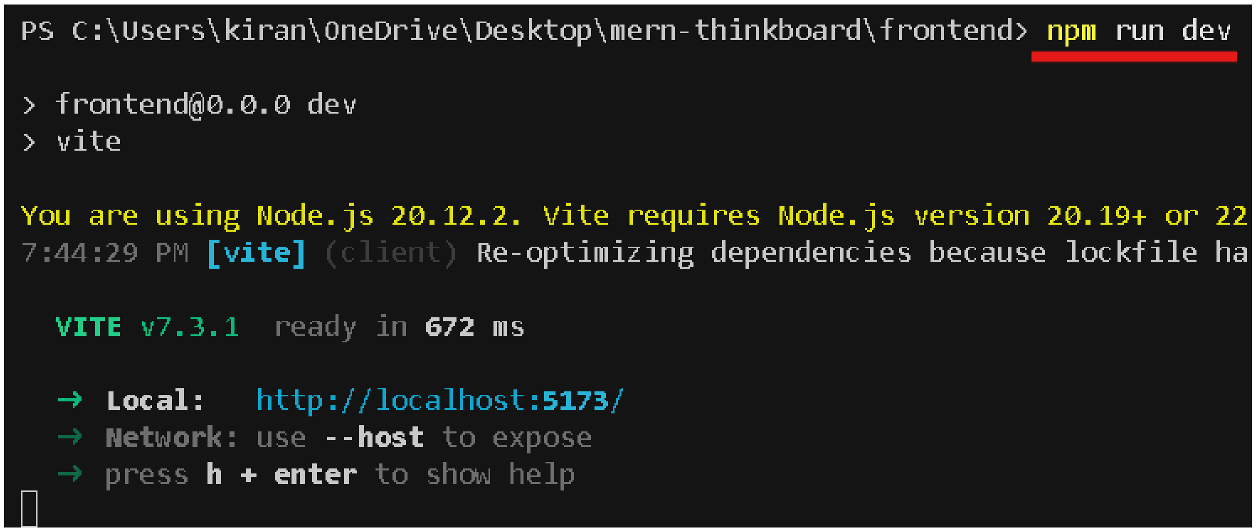
Code in CreatePage.jsx:

```
import React from 'react'

const CreatePage = () => {
  return (
    <div>
      CreatePage
    </div>
  )
}

export default CreatePage
```

Code in NoteDetailPage.jsx:

```
import React from 'react'

const NoteDetailPage = () => {
  return (
    <div>
      NoteDetailPage
    </div>
  )
}

export default NoteDetailPage
```

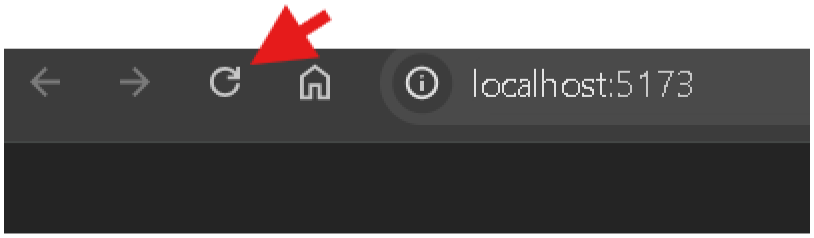
Run the server:

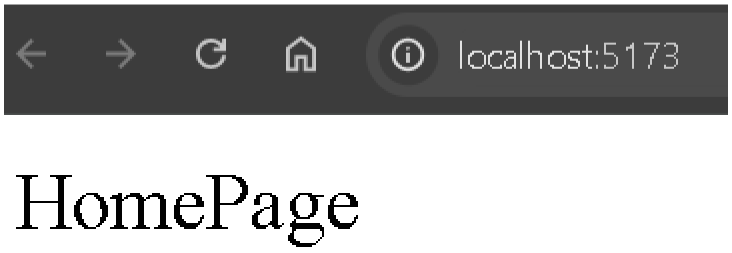


```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm run dev

> frontend@0.0.0 dev
> vite

You are using Node.js 20.12.2. Vite requires Node.js version 20.19+ or 22
7:44:29 PM [vite] (client) Re-optimizing dependencies because lockfile ha

  VITE v7.3.1  ready in 672 ms

  → Local:   http://localhost:5173/
  → Network: use --host to expose
  → press h + enter to show help
```

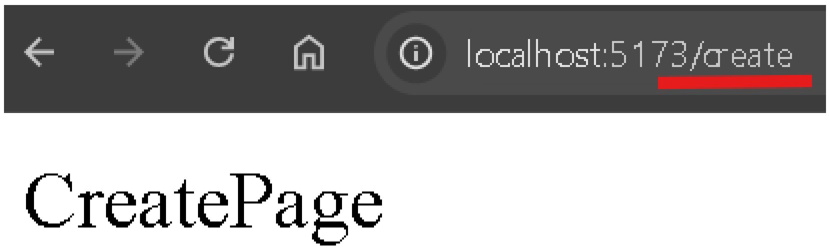
Then in browser:

Refresh it.

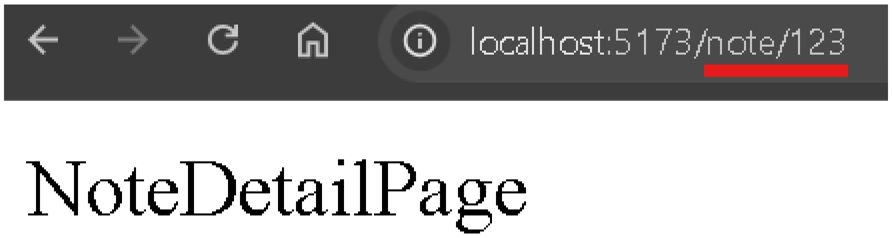



localhost:5173

HomePage


localhost:5173/create

CreatePage

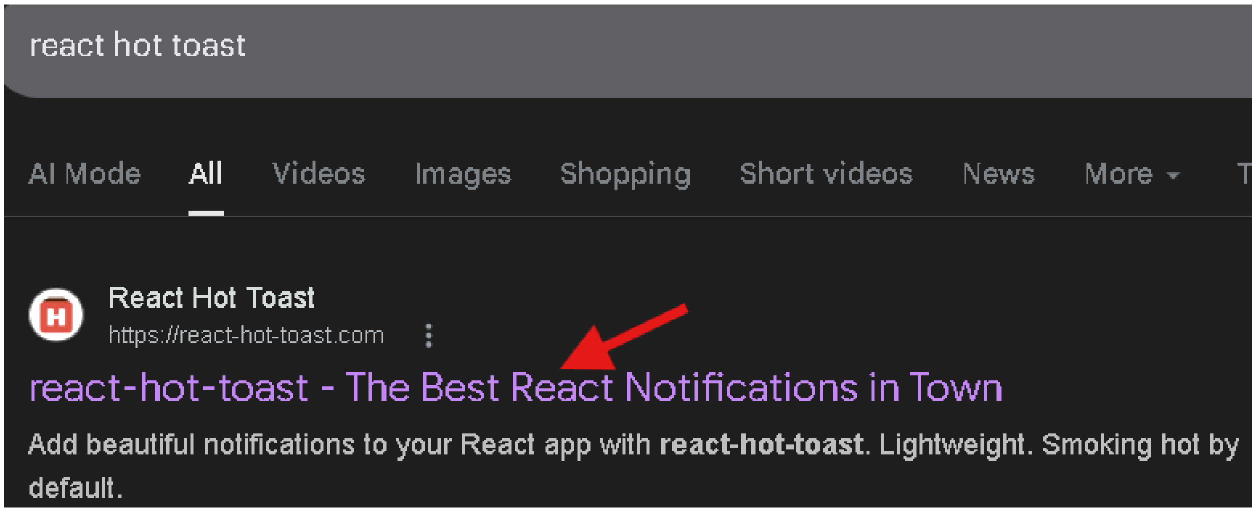

localhost:5173/note/123

NoteDetailPage

Now pages working correctly.

For notifications:

Search: **react hot toast**

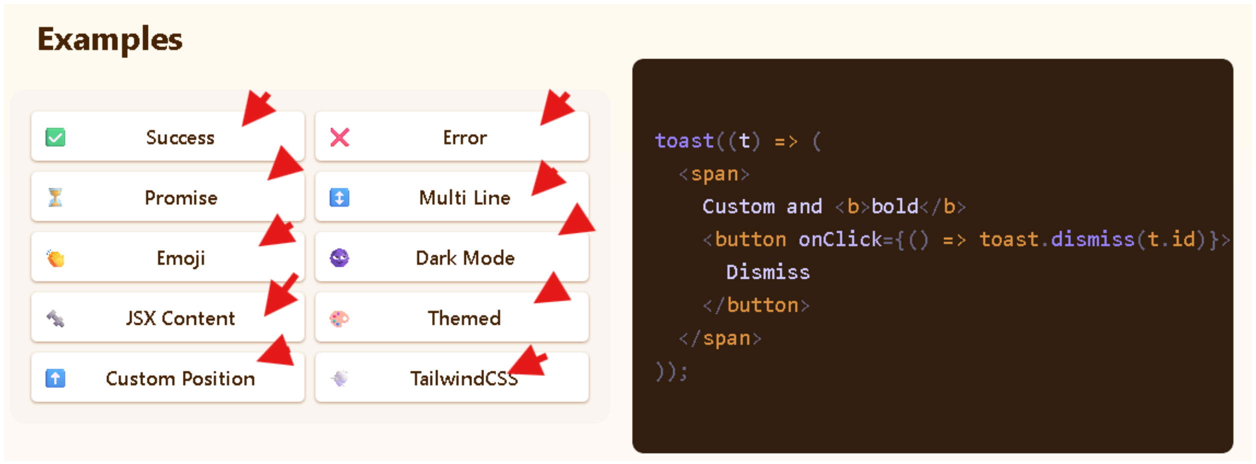

react hot toast

React Hot Toast
https://react-hot-toast.com

react-hot-toast - The Best React Notifications in Town

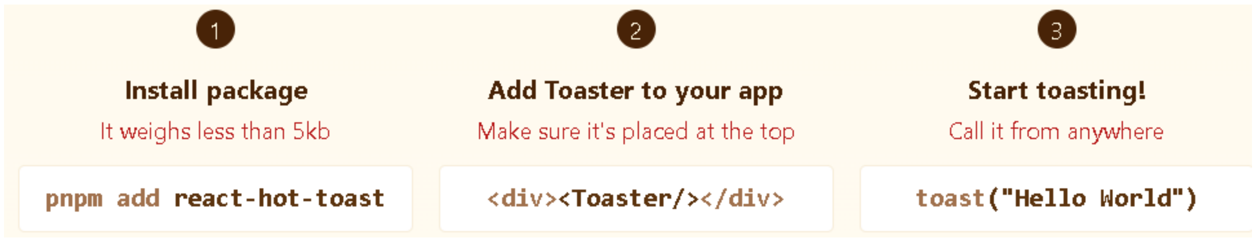
Add beautiful notifications to your React app with **react-hot-toast**. Lightweight. Smoking hot by default.

Use examples:



```
toast((t) => (
  <span>
    Custom and <b>bold</b>
    <button onClick={() => toast.dismiss(t.id)}>
      Dismiss
    </button>
  </span>
));
```

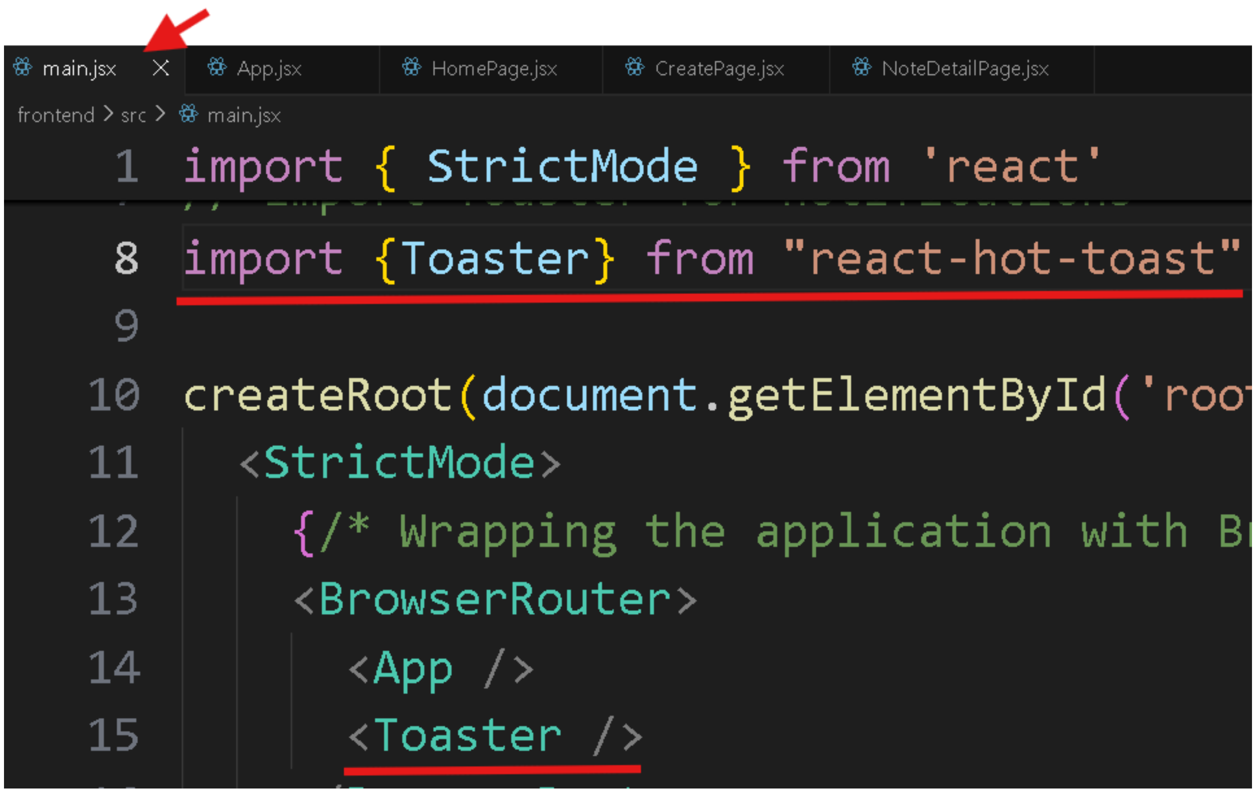
For start using it in project:



```
pnpm add react-hot-toast
```

```
<div><Toaster/></div>
```

```
toast("Hello World")
```



```
import { StrictMode } from 'react'
import {Toaster} from "react-hot-toast"

createRoot(document.getElementById('roo
  <StrictMode>
    {/* Wrapping the application with Br
    <BrowserRouter>
      <App />
      <Toaster />
```

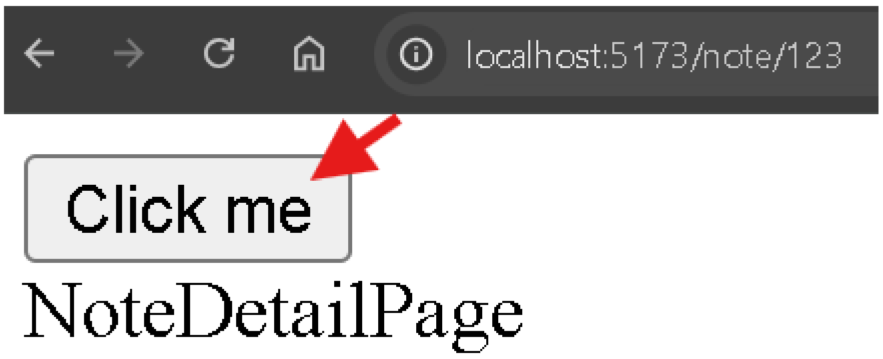
Code in main.jsx:

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.jsx'
// importing the BrowserRouter
import {BrowserRouter} from "react-router"
// import Toaster for notifications
import {Toaster} from "react-hot-toast"

createRoot(document.getElementById('root')).render(
  <StrictMode>
    {/* Wrapping the application with BrowserRouter */}
    <BrowserRouter>
      <App />
      <Toaster />
    </BrowserRouter>
  </StrictMode>,
)
```

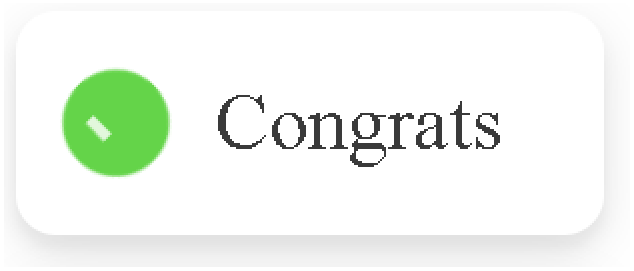
Code in App.jsx:

```
import React from 'react'
import { Route, Routes } from 'react-router'
import HomePage from './pages/HomePage'
import CreatePage from './pages/CreatePage'
import NoteDetailPage from './pages/NoteDetailPage'
// import toast
import toast from "react-hot-toast"

const App = () => {
  return (
    <div>
    {/* For testing using button */}
    <button onClick={() => toast.success("Congrats")}>Click me</button>
    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/create" element={<CreatePage />} />
      {/* :id <- is dynamic */}
      <Route path="/note/:id" element={<NoteDetailPage />} />
    </Routes>

    </div>
  )
}

export default App
```

In browser:

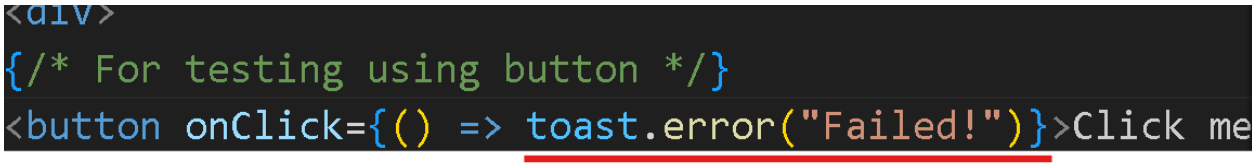


Gets notification:

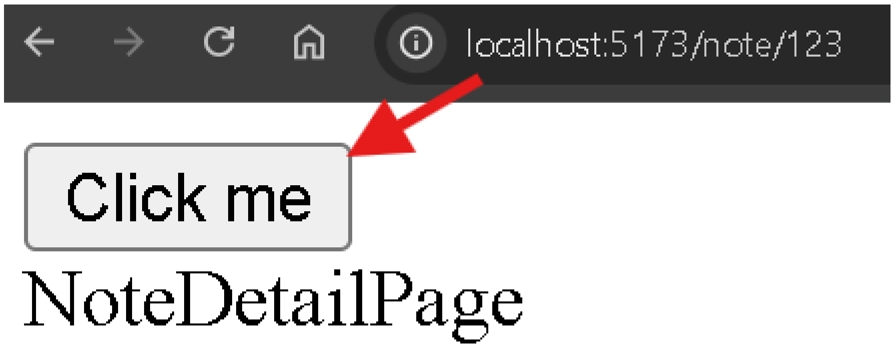


Congrats

To get error:



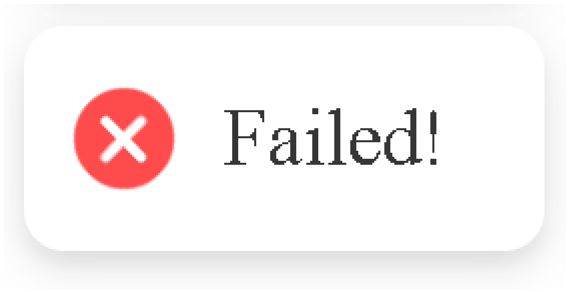
```
<div>
{/* For testing using button */}
<button onClick={() => toast.error("Failed!")}>Click me
```



localhost:5173/note/123

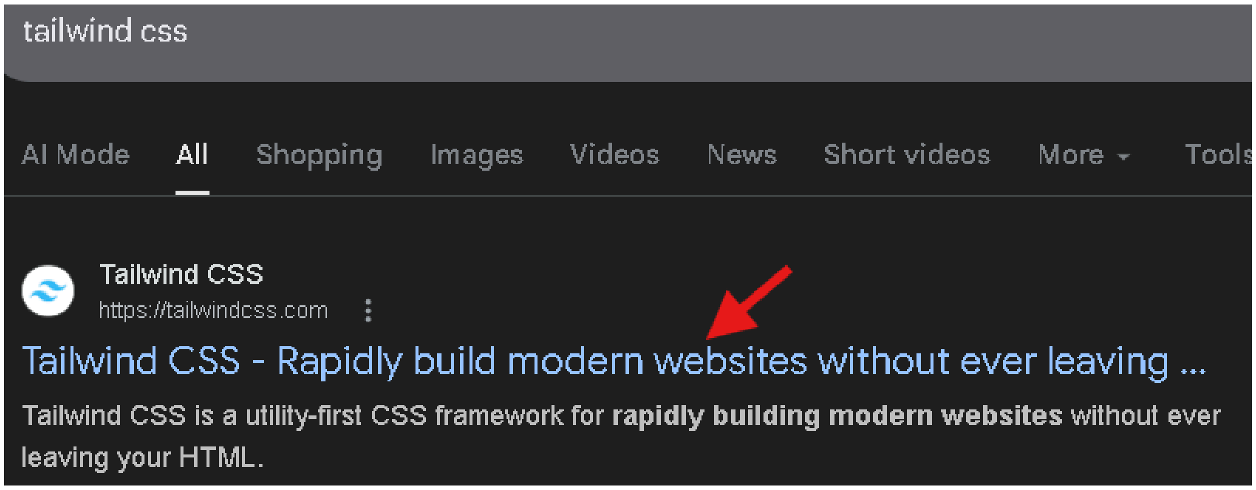
NoteDetailPage

Gets Error:



Failed!

Tailwindcss:

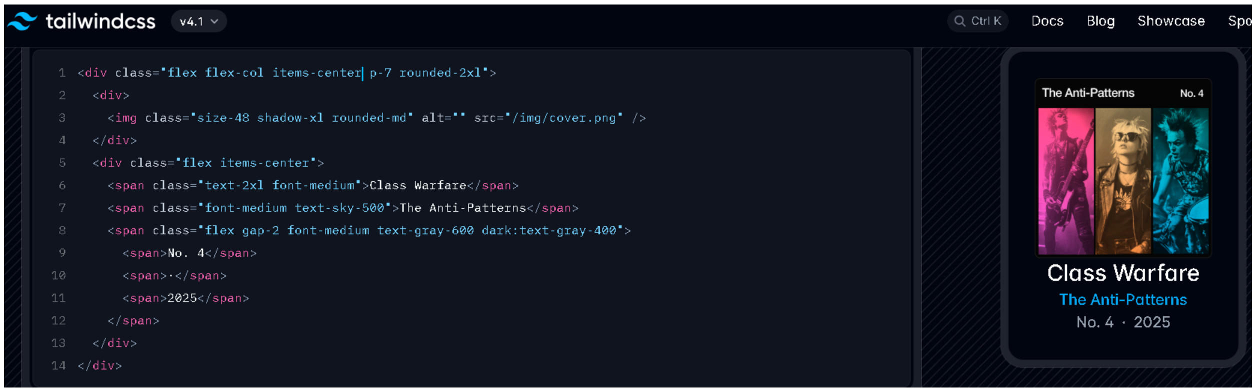


Tailwind CSS
https://tailwindcss.com

Tailwind CSS - Rapidly build modern websites without ever leaving ...

Tailwind CSS is a utility-first CSS framework for **rapidly building modern websites** without ever leaving your HTML.

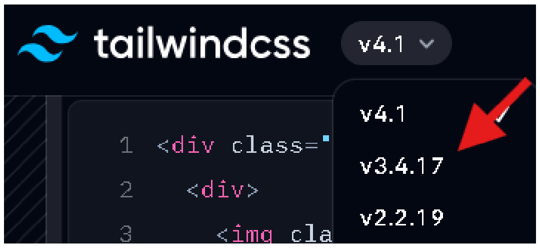
Example shown in site:



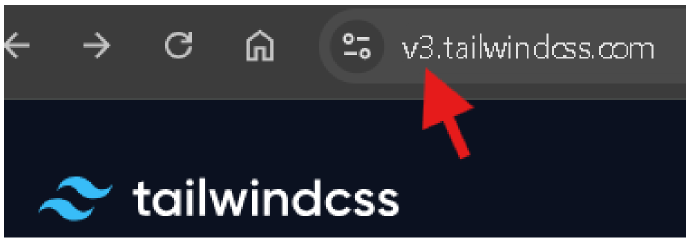
```
<div class="flex flex-col items-center p-7 rounded-2xl">
  <div>
    <img class="size-48 shadow-xl rounded-md" alt="" src="/img/cover.png" />
  </div>
  <div class="flex items-center">
    <span class="text-2xl font-medium">Class Warfare</span>
    <span class="font-medium text-sky-500">The Anti-Patterns</span>
    <span class="flex gap-2 font-medium text-gray-600 dark:text-gray-400">
      <span>No. 4</span>
      <span>·</span>
      <span>2025</span>
    </span>
  </div>
</div>
```

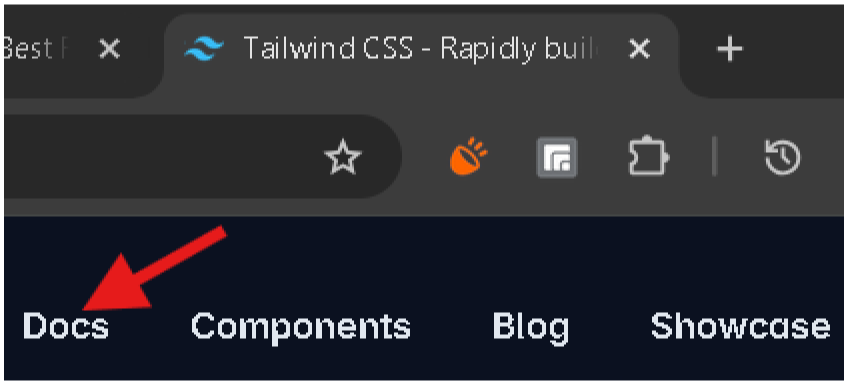
It is the best way of writing css, where we have many utility classes, instead of creating them in css file, we can directly put in (**.jsx**) file. Meaning we can directly put it in html elements.

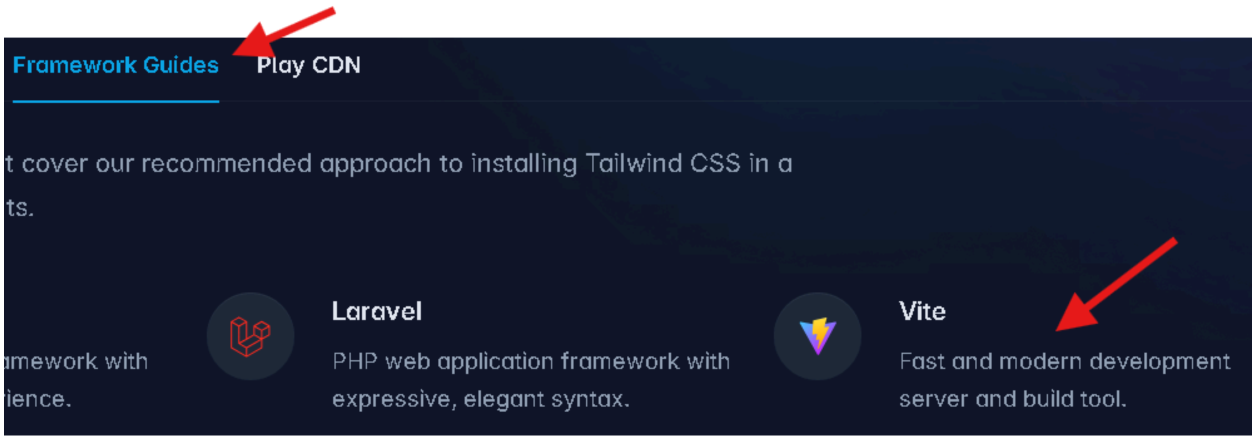
Use 3rd version, for stable:



```
1 <div class="
2   <div>
3     <img cla
```







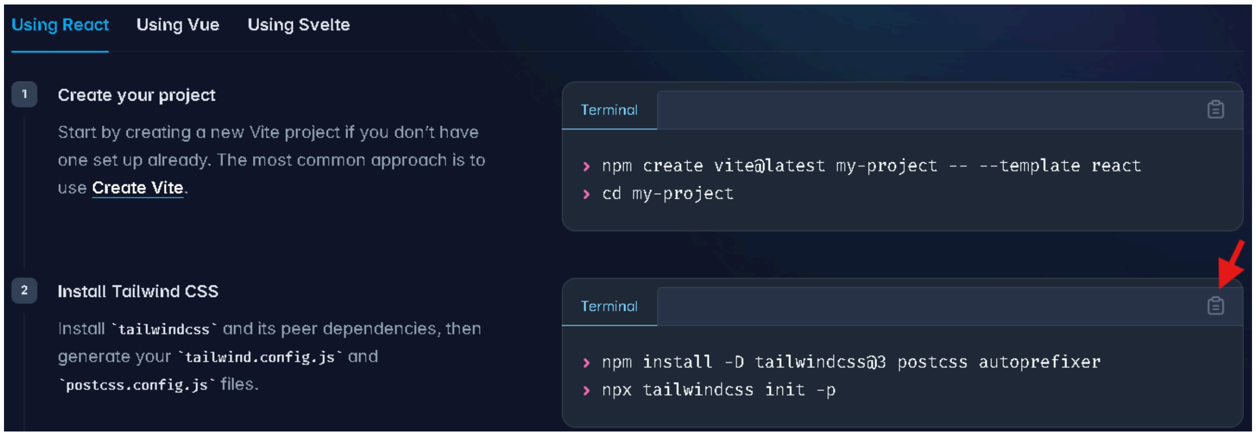
t cover our recommended approach to installing Tailwind CSS in a
ts.

amework with
ience.

Laravel
PHP web application framework with expressive, elegant syntax.

Vite
Fast and modern development server and build tool.

Copy the code:



1 Create your project

Start by creating a new Vite project if you don't have one set up already. The most common approach is to use Create Vite.

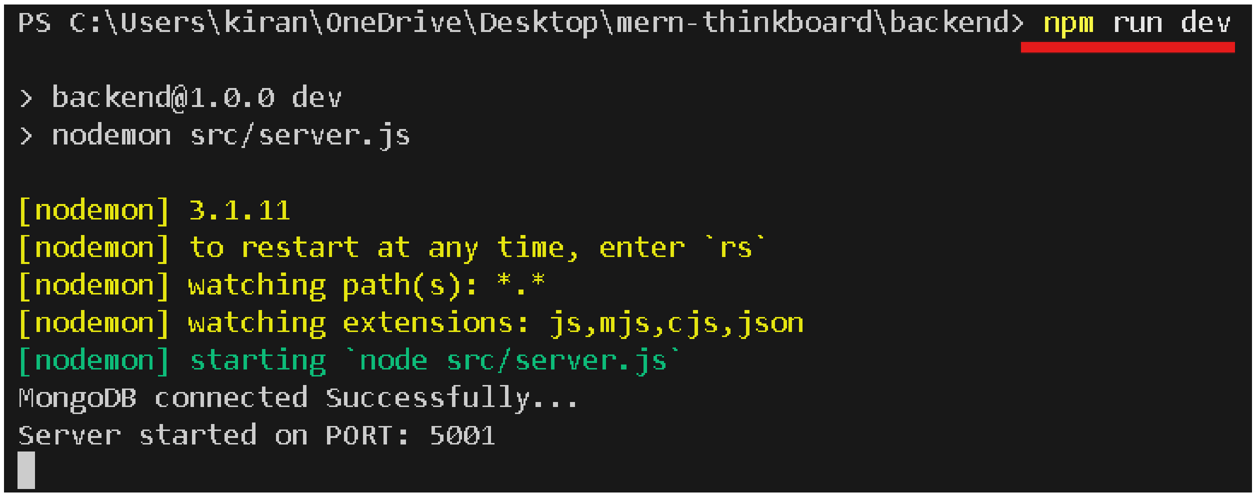
```
> npm create vite@latest my-project -- --template react
> cd my-project
```

2 Install Tailwind CSS

Install `tailwindcss` and its peer dependencies, then generate your `tailwind.config.js` and `postcss.config.js` files.

```
> npm install -D tailwindcss@3 postcss autoprefixer
> npx tailwindcss init -p
```

Code is:
npm install -D tailwindcss@3 postcss autoprefixer
npx tailwindcss init -p
In backend:



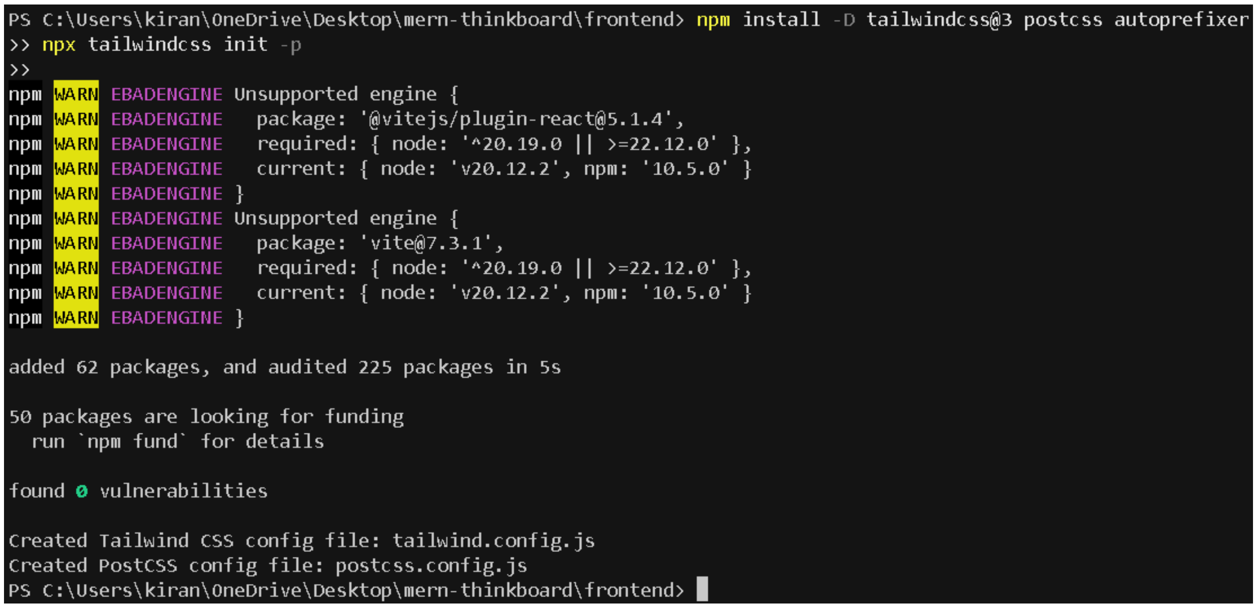
```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\backend> npm run dev

> backend@1.0.0 dev
> nodemon src/server.js

[nodemon] 3.1.11
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node src/server.js`
MongoDB connected Successfully...
Server started on PORT: 5001
```

In the frontend:

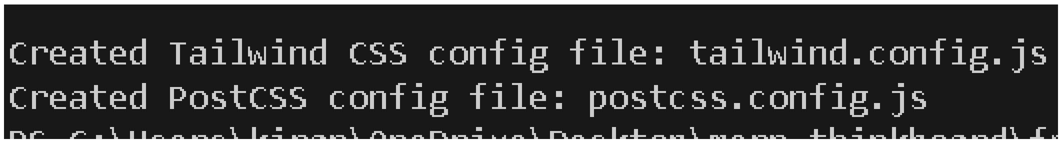
1. Stop the server.
2. Paste the code:

   npm install -D tailwindcss@3 postcss autoprefixer
   npx tailwindcss init -p

In terminal:

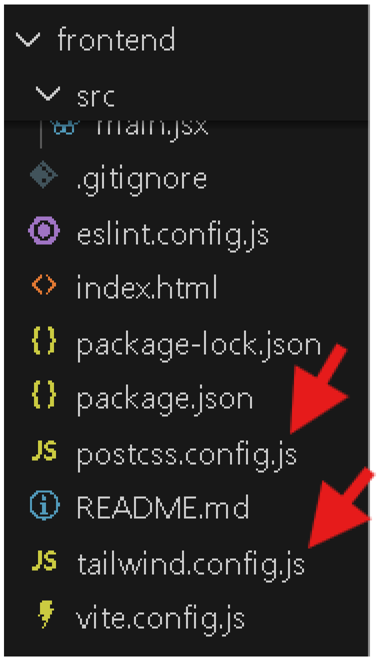


```
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm install -D tailwindcss@3 postcss autoprefixer
>> npx tailwindcss init -p
>>
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: '@vitejs/plugin-react@5.1.4',
npm WARN EBADENGINE   required: { node: '^20.19.0 || >=22.12.0' },
npm WARN EBADENGINE   current: { node: 'v20.12.2', npm: '10.5.0' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'vite@7.3.1',
npm WARN EBADENGINE   required: { node: '^20.19.0 || >=22.12.0' },
npm WARN EBADENGINE   current: { node: 'v20.12.2', npm: '10.5.0' }
npm WARN EBADENGINE }

added 62 packages, and audited 225 packages in 5s

50 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Created Tailwind CSS config file: tailwind.config.js
Created PostCSS config file: postcss.config.js
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend>
```

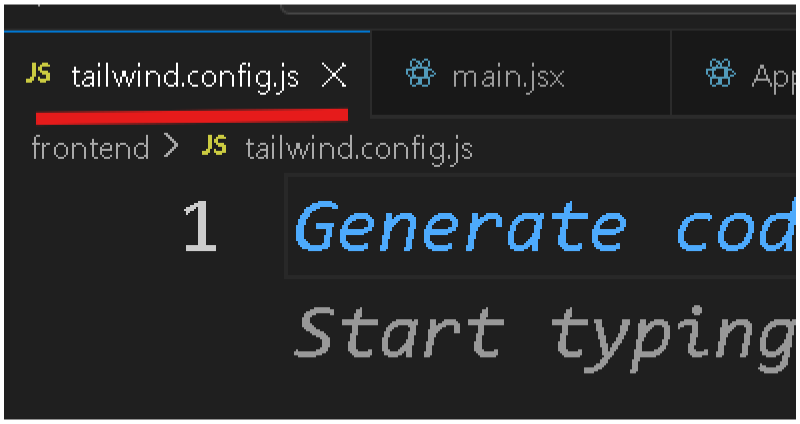


```
Created Tailwind CSS config file: tailwind.config.js
Created PostCSS config file: postcss.config.js
```

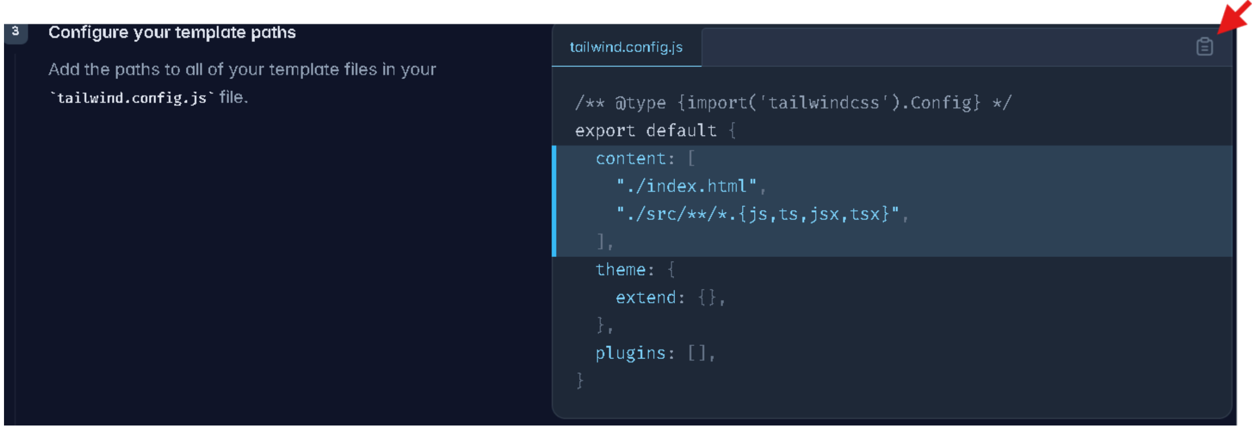
So it has created these files:



Clear the code in the tailwind.config.js:



Get the code from documentation and paste it in above tailwind.config.js:



3 Configure your template paths

Add the paths to all of your template files in your `tailwind.config.js` file.

tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Code is:

/** @type {import('tailwindcss').Config} */
export default {

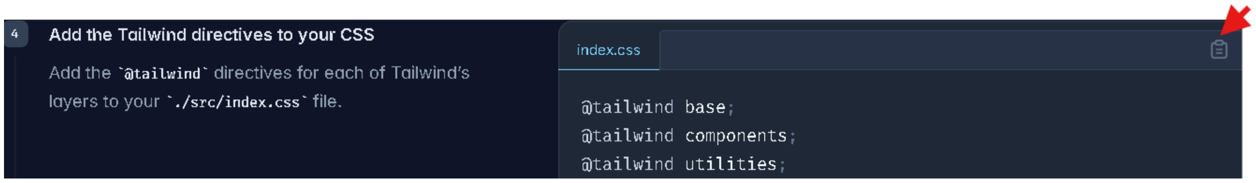
```
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Code in tailwind.config.js:

```
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Updating the index.css file:

From documentation: Copy the code


4 Add the Tailwind directives to your CSS

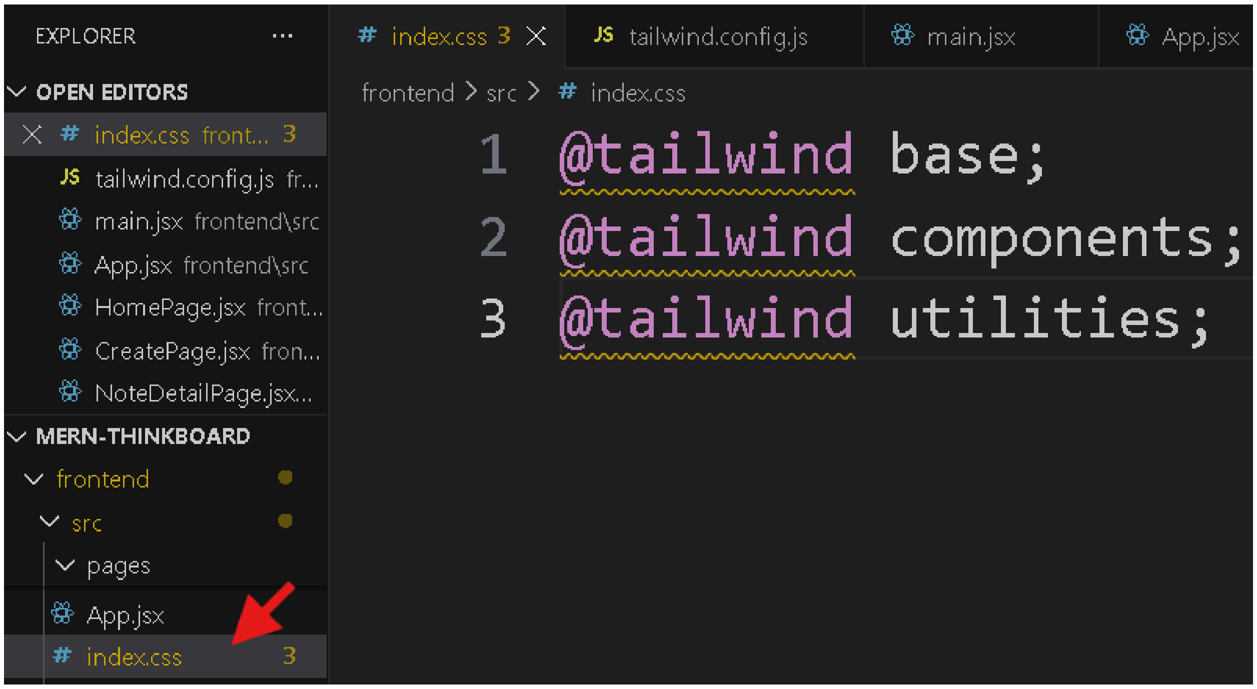
Add the `@tailwind` directives for each of Tailwind's layers to your `./src/index.css` file.

index.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Code is:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```


EXPLORER ···

OPEN EDITORS
- index.css front... 3
- tailwind.config.js fr...
- main.jsx frontend\src
- App.jsx frontend\src
- HomePage.jsx front...
- CreatePage.jsx fron...
- NoteDetailPage.jsx...

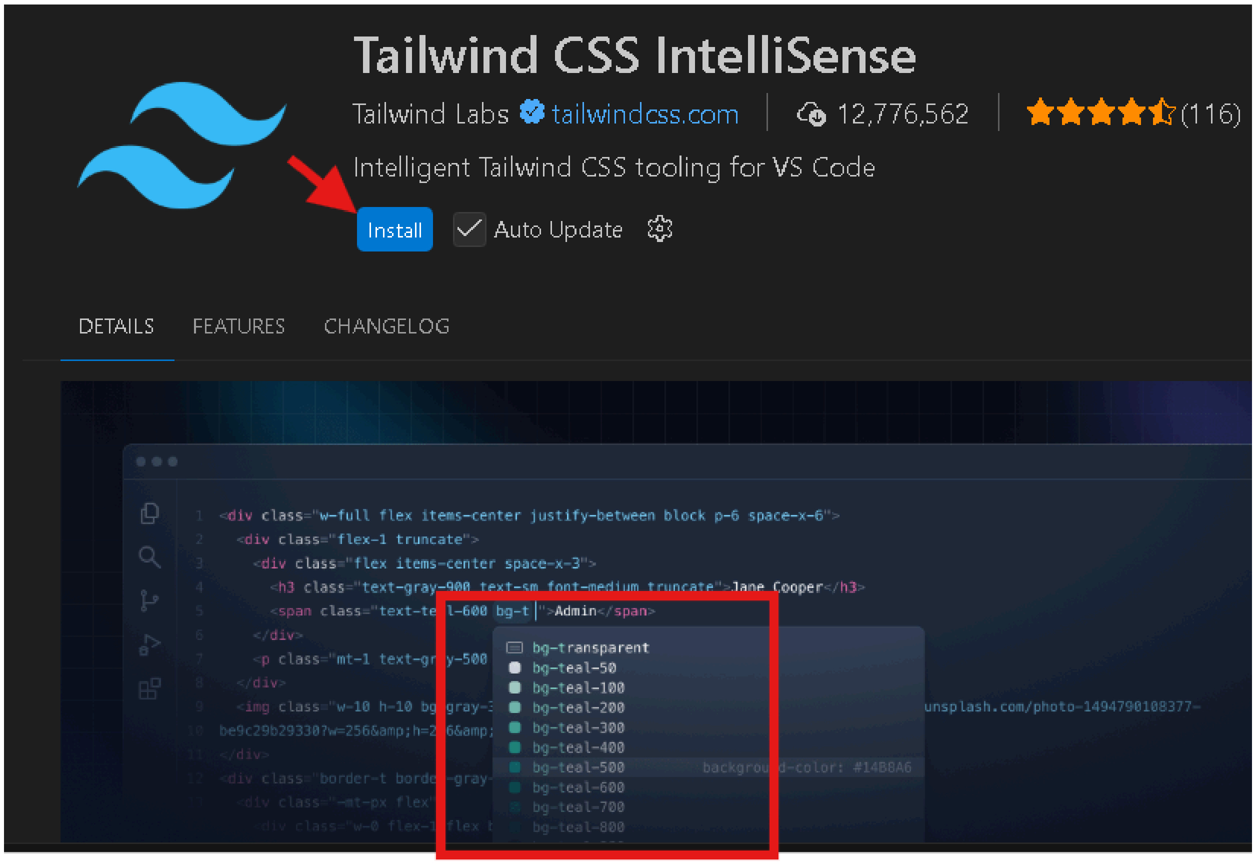
MERN-THINKBOARD
- frontend
  - src
    - pages
    - App.jsx
    - index.css 3

index.css 3 | tailwind.config.js | main.jsx | App.jsx

frontend > src > index.css

```css
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Extension installation: *for auto compilation of CSS*


Tailwind CSS IntelliSense

Tailwind Labs tailwindcss.com | 12,776,562 | (116)

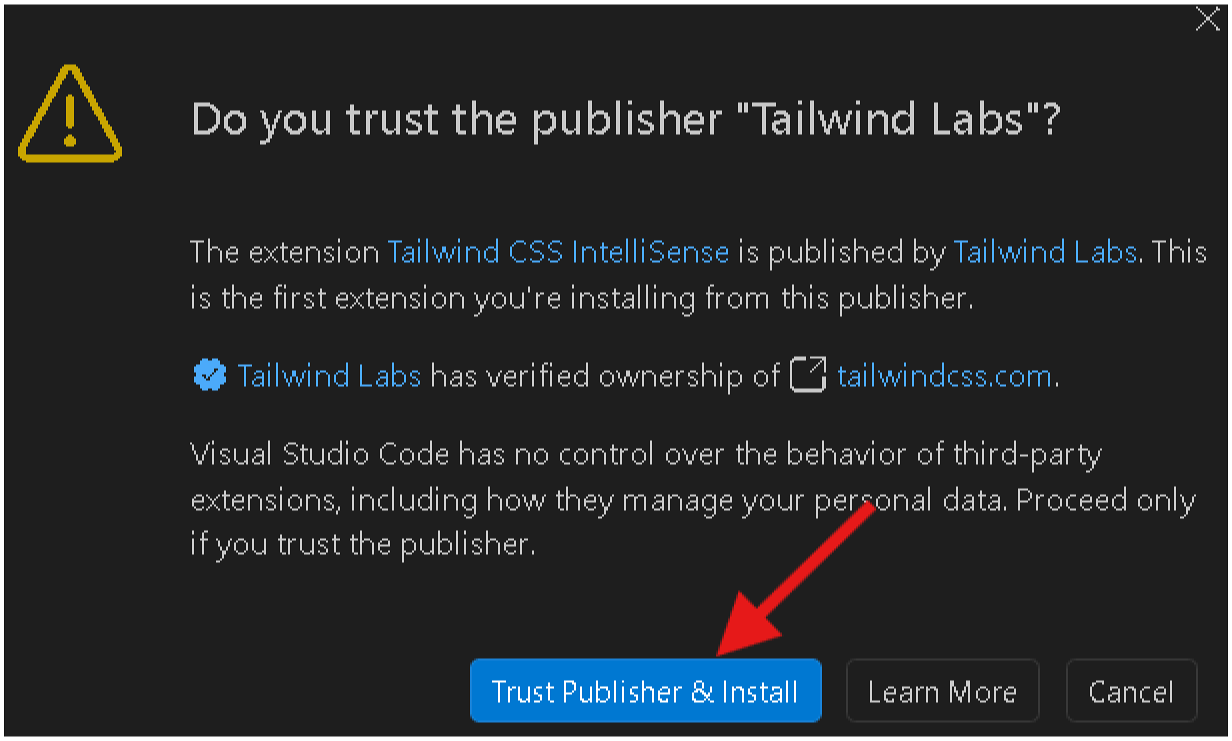
Intelligent Tailwind CSS tooling for VS Code

Install | Auto Update

DETAILS FEATURES CHANGELOG

```html
<div class="w-full flex items-center justify-between block p-6 space-x-6">
  <div class="flex-1 truncate">
    <div class="flex items-center space-x-3">
      <h3 class="text-gray-900 text-sm font-medium truncate">Jane Cooper</h3>
      <span class="text-te l-600 bg-t |">Admin</span>
    </div>
    <p class="mt-1 text-gr y-500
  </div>
  <img class="w-10 h-10 bg gray-3                         unsplash.com/photo-1494790108377-
be9c29b29330?w=256&amp;h=2 6&amp;
</div>
<div class="border-t borde -gray-
  <div class="-mt-px flex"
    <div class="w-0 flex-1 flex
```

```
bg-transparent
bg-teal-50
bg-teal-100
bg-teal-200
bg-teal-300
bg-teal-400
bg-teal-500          backgrou d-color: #14B8A6
bg-teal-600
bg-teal-700
bg-teal-800
```

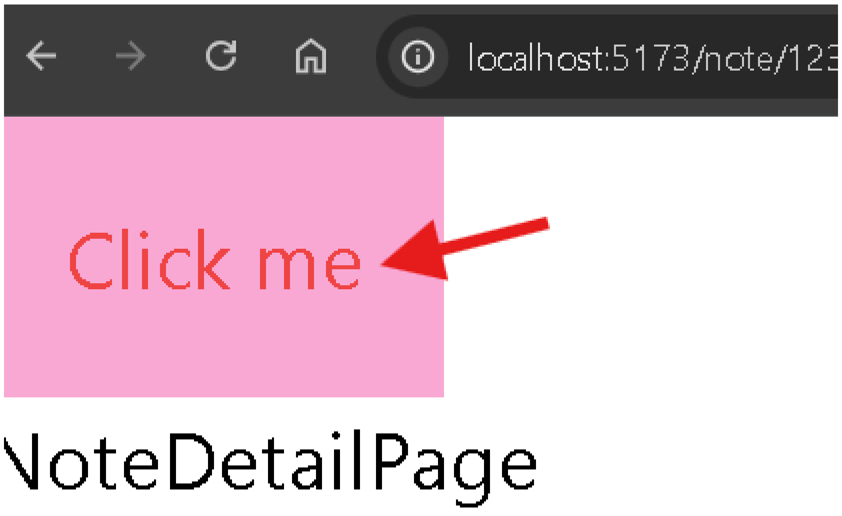

Do you trust the publisher "Tailwind Labs"?

The extension Tailwind CSS IntelliSense is published by Tailwind Labs. This is the first extension you're installing from this publisher.

Tailwind Labs has verified ownership of tailwindcss.com.

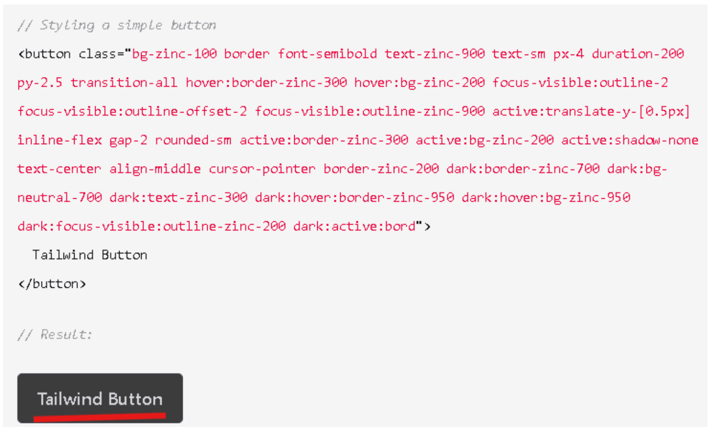
Visual Studio Code has no control over the behavior of third-party extensions, including how they manage your personal data. Proceed only if you trust the publisher.

In App.jsx:
{/* For testing using button */}
  <button onClick={() => toast.error("Failed!")} className='**text-red-500 p-4 bg-pink-300 bg**'>Click me</button>
Then button looks like:

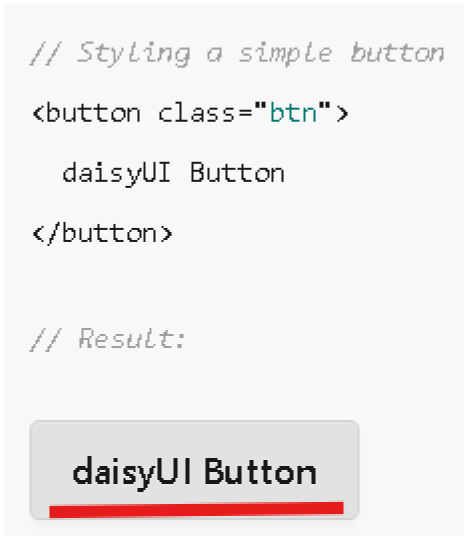


DaisyUI:

Reduces complex codes, class,etc. Visit: https://daisyui.com/ then,
Example:

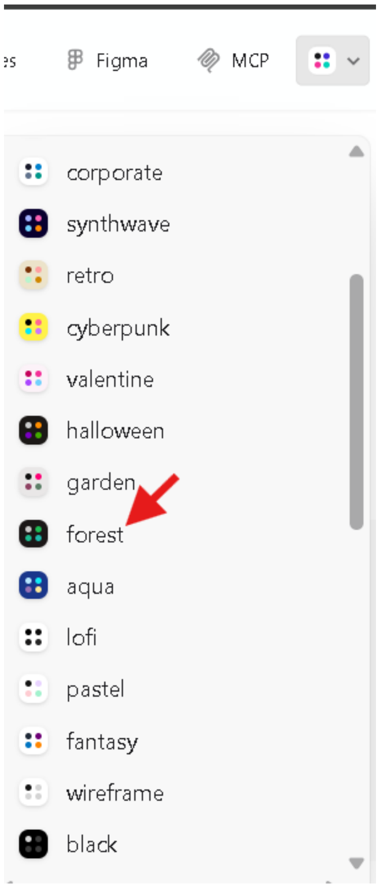


```
// Styling a simple button
<button class="bg-zinc-100 border font-semibold text-zinc-900 text-sm px-4 duration-200
py-2.5 transition-all hover:border-zinc-300 hover:bg-zinc-200 focus-visible:outline-2
focus-visible:outline-offset-2 focus-visible:outline-zinc-900 active:translate-y-[0.5px]
inline-flex gap-2 rounded-sm active:border-zinc-300 active:bg-zinc-200 active:shadow-none
text-center align-middle cursor-pointer border-zinc-200 dark:border-zinc-700 dark:bg-
neutral-700 dark:text-zinc-300 dark:hover:border-zinc-950 dark:hover:bg-zinc-950
dark:focus-visible:outline-zinc-200 dark:active:bord">
  Tailwind Button
</button>

// Result:
```

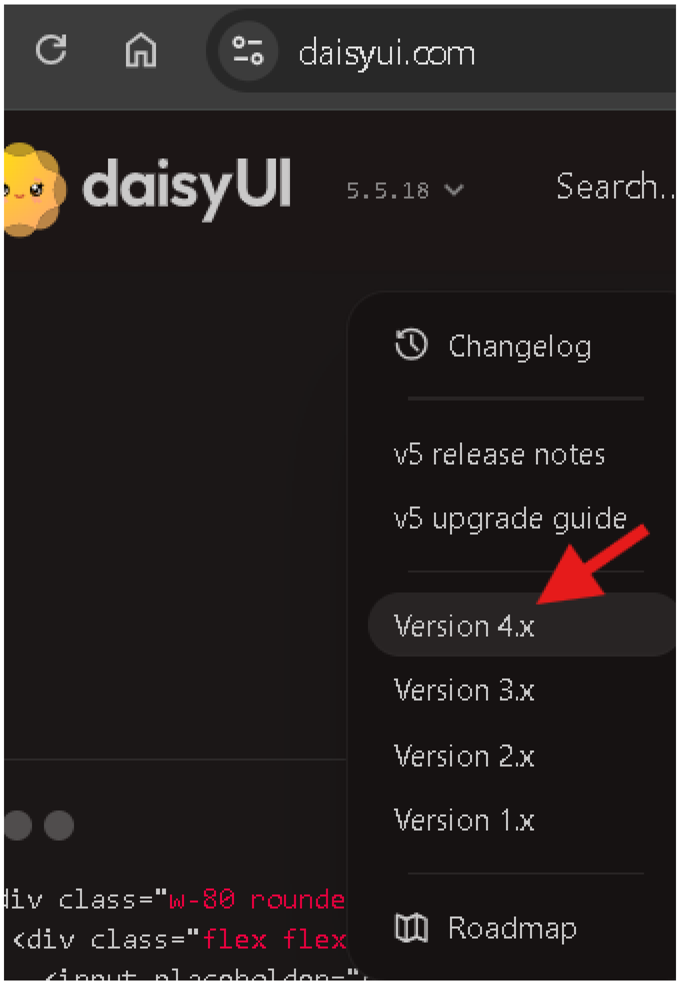


```
// Styling a simple button
<button class="btn">
  daisyUI Button
</button>

// Result:
```

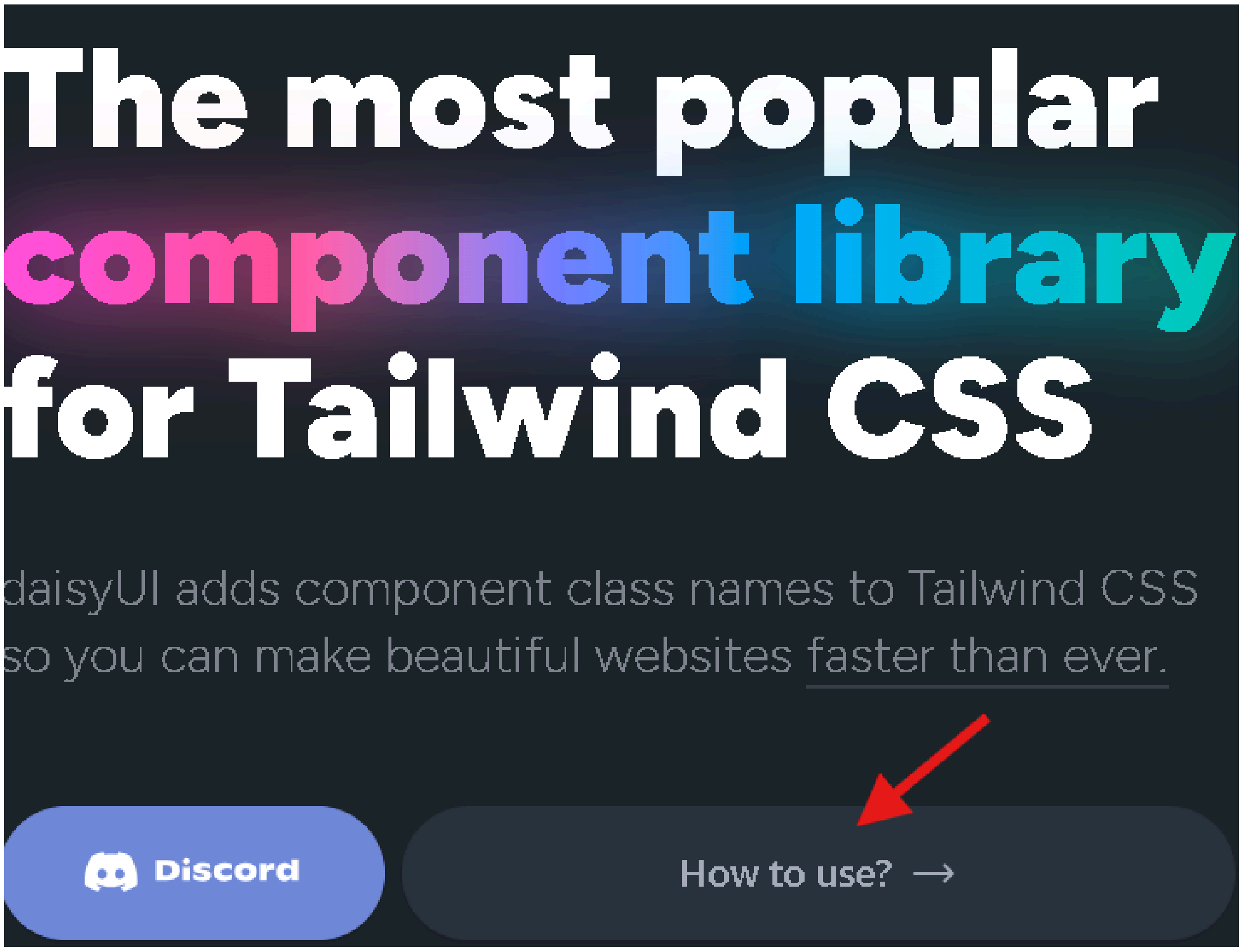
They have different themes:
Choosing forest theme.



Choose 4th version:

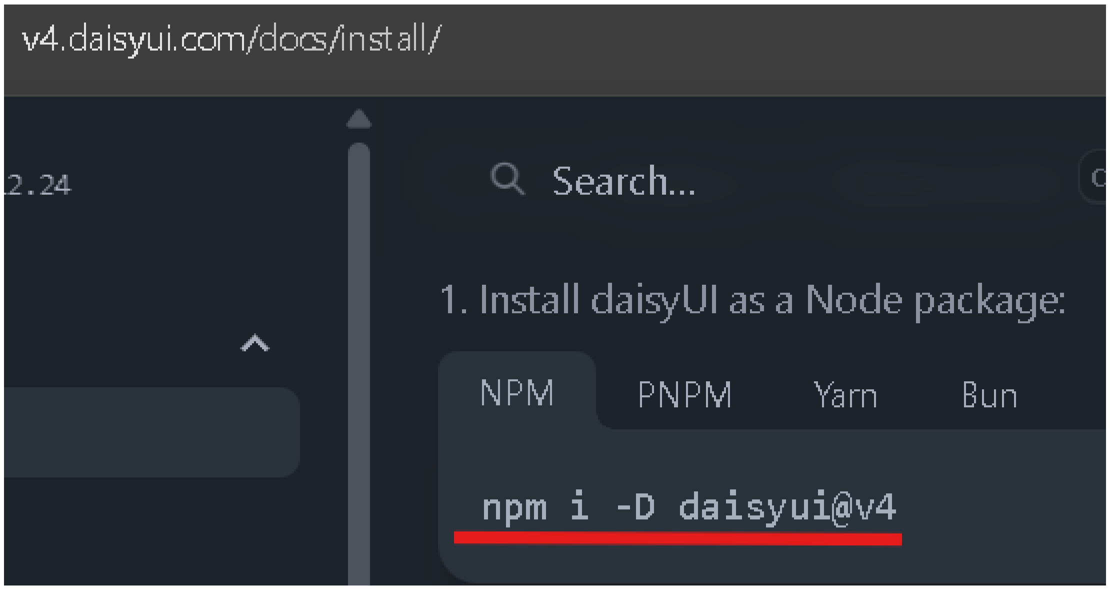


```
div class="w-80 rounde
 <div class="flex flex
```


The most popular component library for Tailwind CSS

daisyUI adds component class names to Tailwind CSS so you can make beautiful websites faster than ever.

For installing daisyUI:

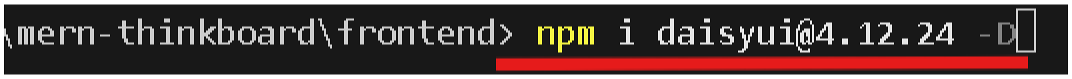

v4.daisyui.com/docs/install/

1. Install daisyUI as a Node package:

```
npm i -D daisyui@v4
```

In terminal:

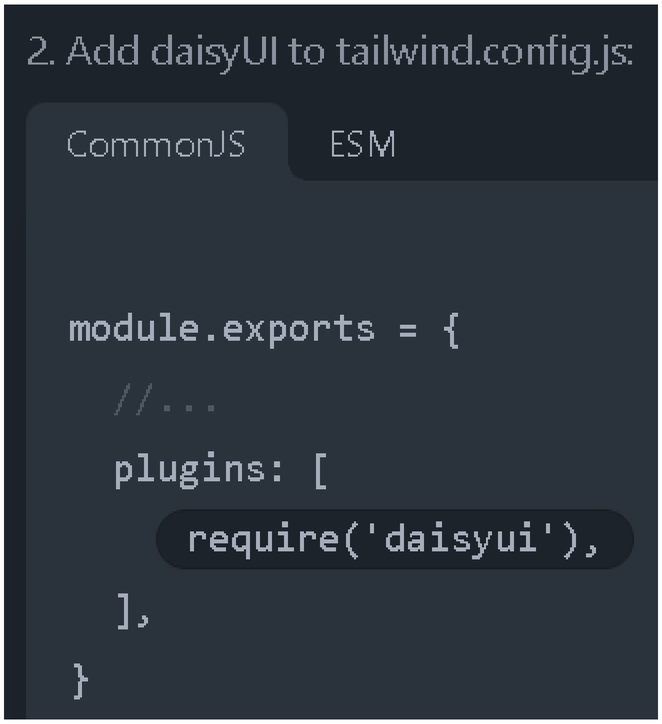
1. Stop the server.
2. Use command: **npm i daisyui@4.12.24 -D**



```
\mern-thinkboard\frontend> npm i daisyui@4.12.24 -D
```

3. Then run the server.

Next step:



2. Add daisyUI to tailwind.config.js:

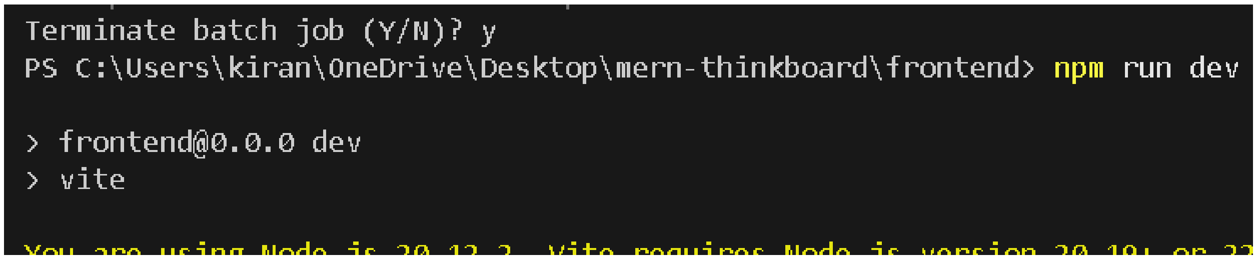
```
module.exports = {
  //...
  plugins: [
    require('daisyui'),
  ],
}
```

We are using the import syntax.

Code in tailwind.config.js:

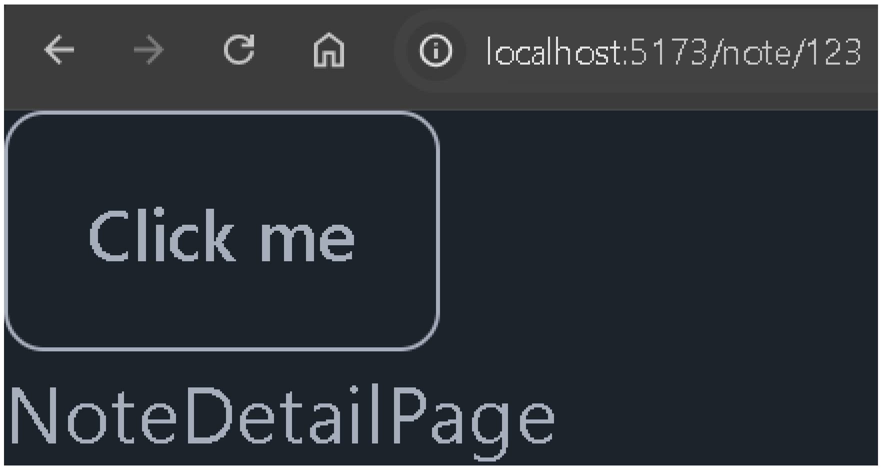
```
import daisyui from 'daisyui';

/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [daisyui],
}
```

Stop the frontend server and rerun it, in terminal:



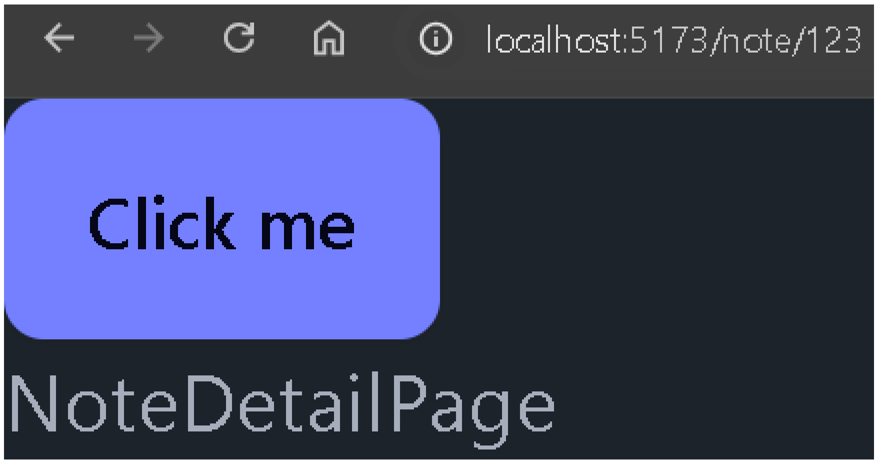
```
Terminate batch job (Y/N)? y
PS C:\Users\kiran\OneDrive\Desktop\mern-thinkboard\frontend> npm run dev

> frontend@0.0.0 dev
> vite
```

In App.jsx:
{/* For testing using button */}
    <button className="btn btn-outline">Click me</button>
In Browser: it also changes the theme.



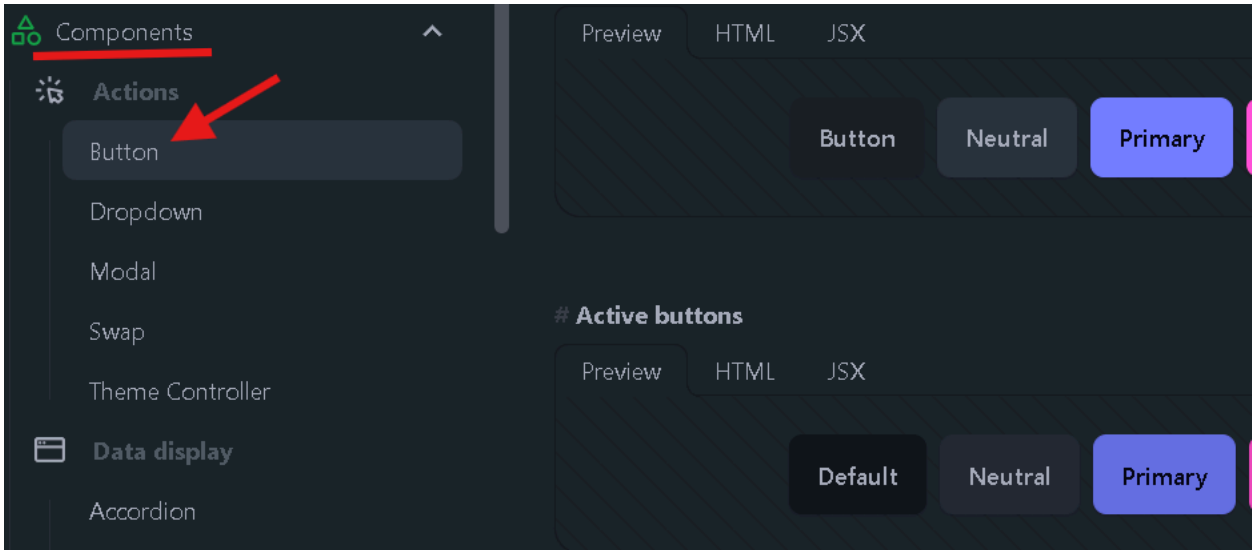
localhost:5173/note/123

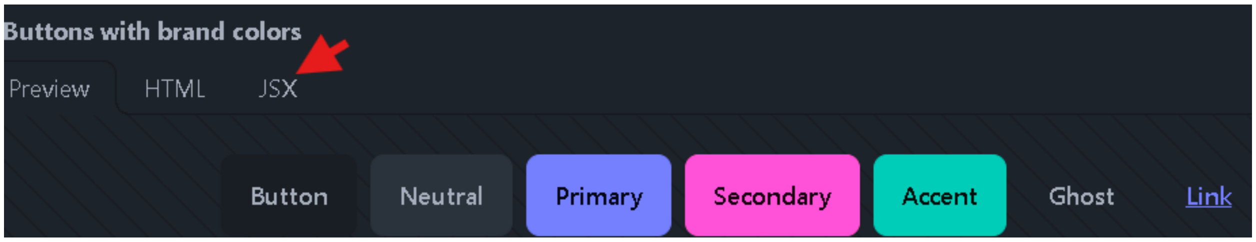
In App.jsx:
{/* For testing using button */}
    <button className="btn btn-primary">Click me</button>
In Browser: it also changes the theme.



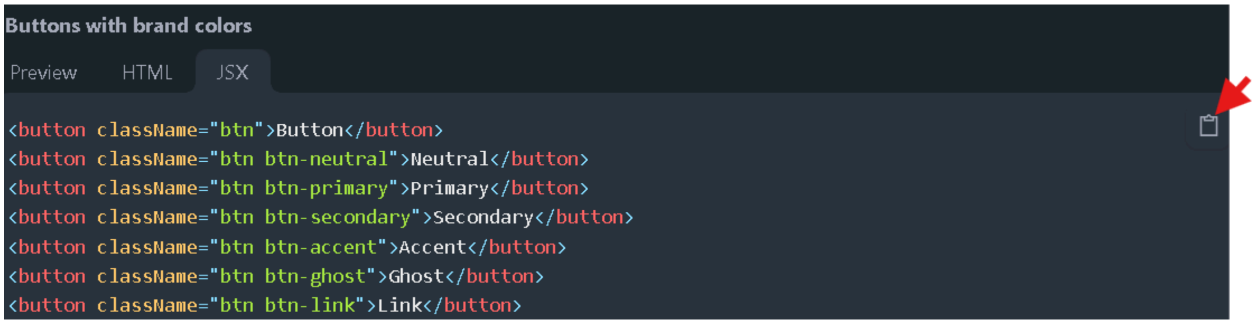
localhost:5173/note/123

Under the component we have different buttons.





Copy the code:

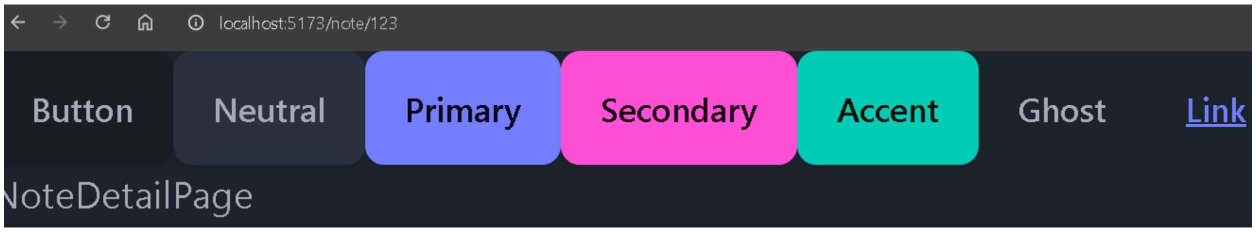


```
<button className="btn">Button</button>
<button className="btn btn-neutral">Neutral</button>
<button className="btn btn-primary">Primary</button>
<button className="btn btn-secondary">Secondary</button>
<button className="btn btn-accent">Accent</button>
<button className="btn btn-ghost">Ghost</button>
<button className="btn btn-link">Link</button>
```

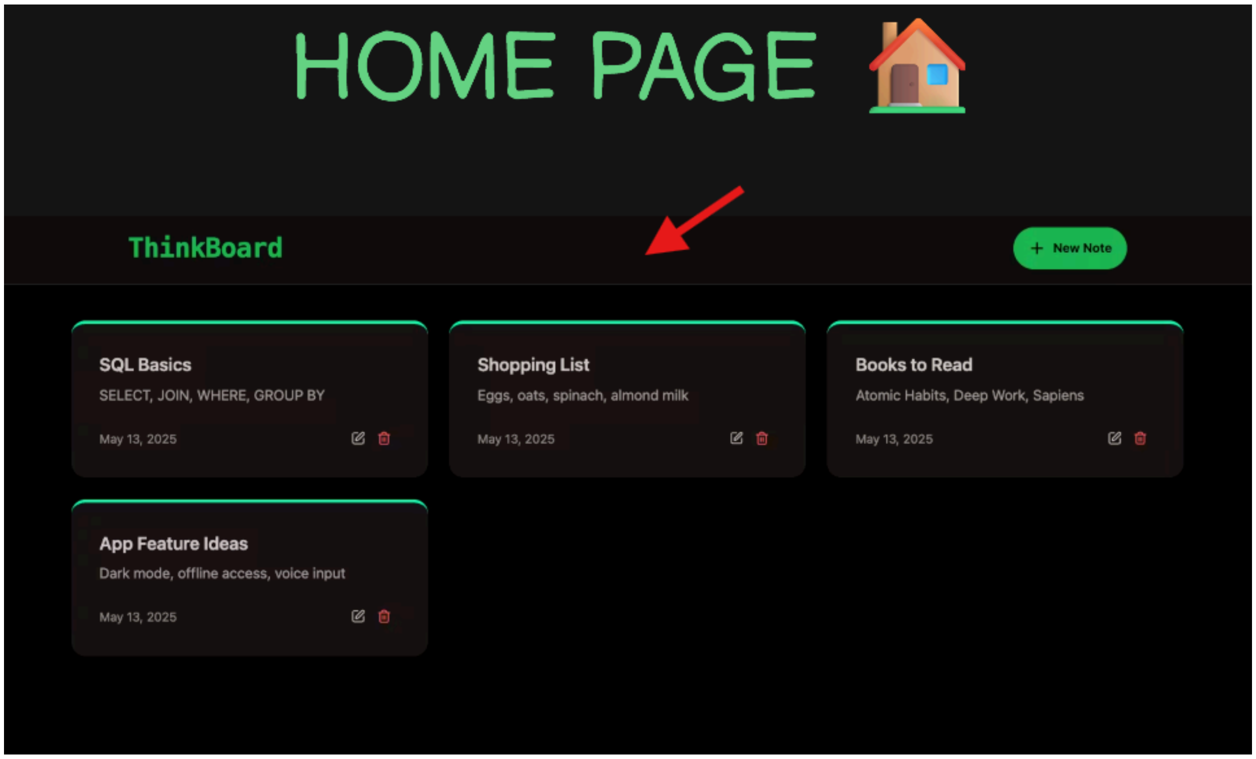
Code in App.jsx:

```
import React from 'react'
import { Route, Routes } from 'react-router'
import HomePage from './pages/HomePage'
import CreatePage from './pages/CreatePage'
import NoteDetailPage from './pages/NoteDetailPage'
// import toast
import toast from "react-hot-toast"

const App = () => {
  return (
    <div>
    <button className="btn">Button</button>
    <button className="btn btn-neutral">Neutral</button>
    <button className="btn btn-primary">Primary</button>
    <button className="btn btn-secondary">Secondary</button>
    <button className="btn btn-accent">Accent</button>
    <button className="btn btn-ghost">Ghost</button>
    <button className="btn btn-link">Link</button>
   <Routes>
    <Route path="/" element={<HomePage />} />
    <Route path="/create" element={<CreatePage />} />
    {/* :id <- is dynamic */}
    <Route path="/note/:id" element={<NoteDetailPage />} />
   </Routes>

   </div>
  )
}

export default App
```

In browser:



For the home page, we can use forest theme:


HOME PAGE 🏠

ThinkBoard

+ New Note

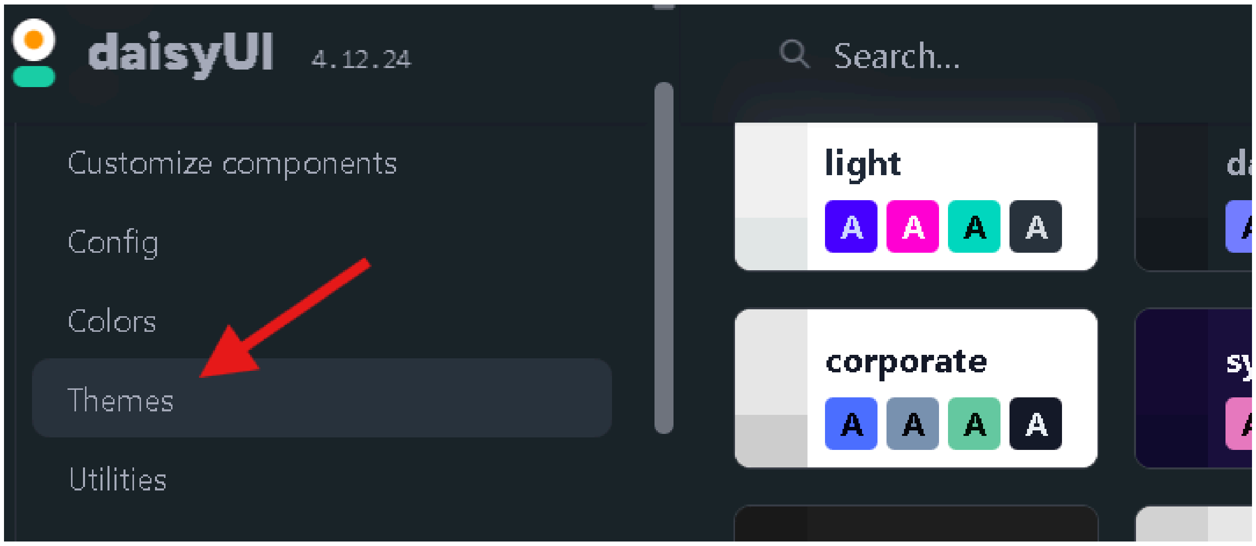
SQL Basics
SELECT, JOIN, WHERE, GROUP BY
May 13, 2025

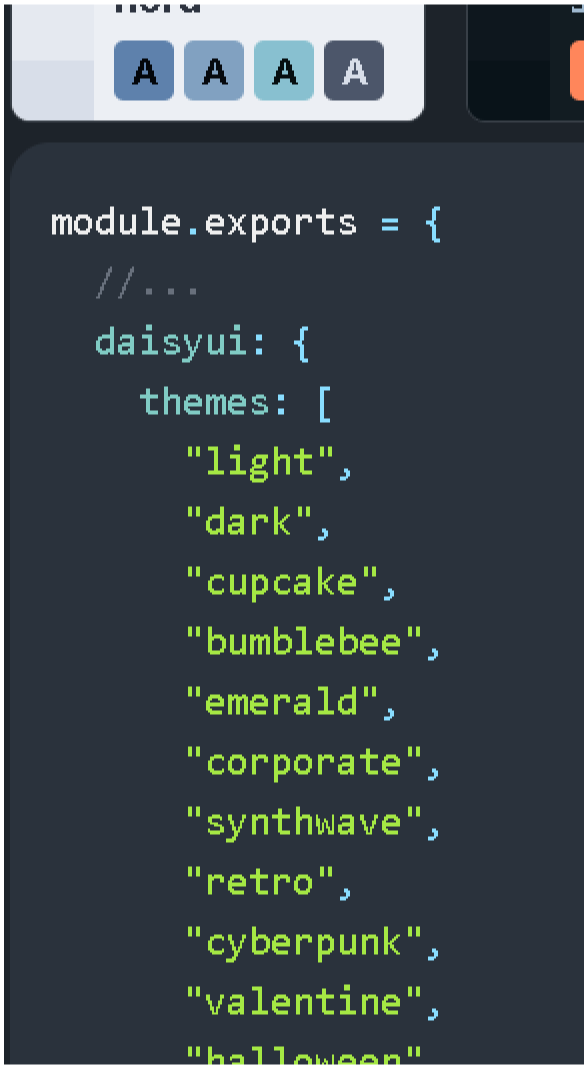
Shopping List
Eggs, oats, spinach, almond milk
May 13, 2025

Books to Read
Atomic Habits, Deep Work, Sapiens
May 13, 2025

App Feature Ideas
Dark mode, offline access, voice input
May 13, 2025

DaisyUI, has the themes:





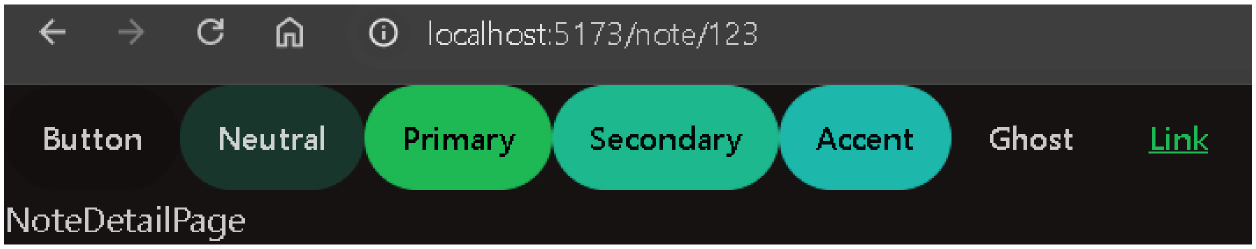
```
module.exports = {
  //...
  daisyui: {
    themes: [
      "light",
      "dark",
      "cupcake",
      "bumblebee",
      "emerald",
      "corporate",
      "synthwave",
      "retro",
      "cyberpunk",
      "valentine",
      "halloween"
```

Code in tailwind.config.js:

```
import daisyui from 'daisyui';

/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [daisyui],
  daisyui:{
    // You can provide themes you like
    // themes: ["light", "dark", "forest"]
    themes: ["forest"]
  },
}
```

Code in App.jsx:

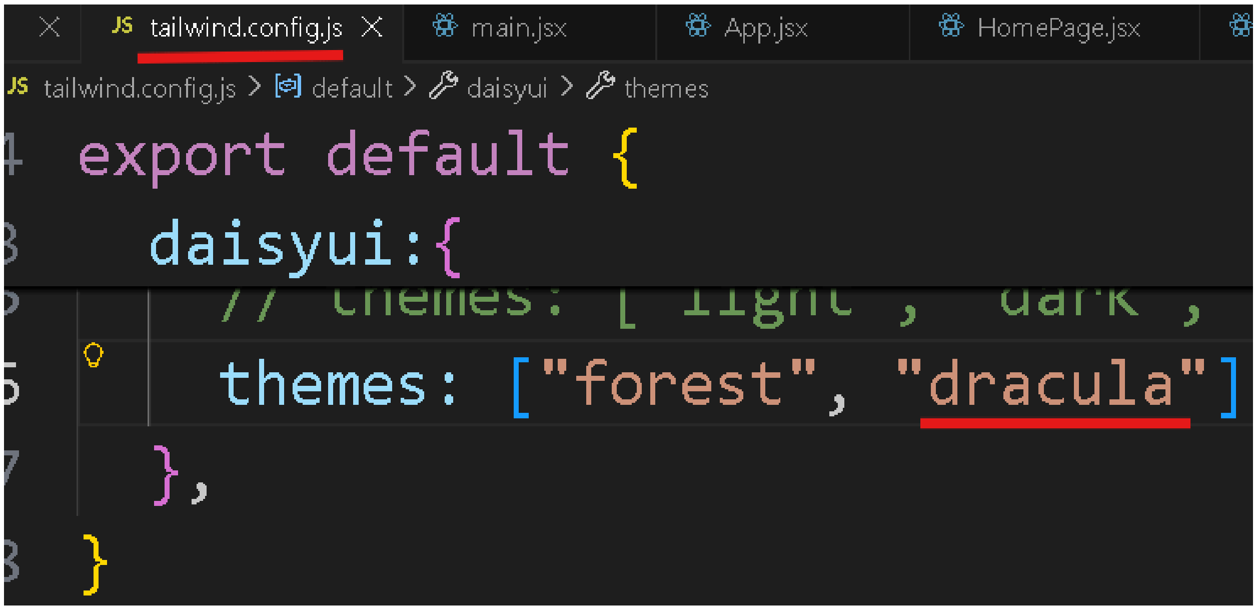
```
import React from 'react'
import { Route, Routes } from 'react-router'
import HomePage from './pages/HomePage'
import CreatePage from './pages/CreatePage'
import NoteDetailPage from './pages/NoteDetailPage'
// import toast
import toast from "react-hot-toast"

const App = () => {
  return (
    <div data-theme="forest">
      <button className="btn">Button</button>
      <button className="btn btn-neutral">Neutral</button>
      <button className="btn btn-primary">Primary</button>
      <button className="btn btn-secondary">Secondary</button>
      <button className="btn btn-accent">Accent</button>
      <button className="btn btn-ghost">Ghost</button>
      <button className="btn btn-link">Link</button>
    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/create" element={<CreatePage />} />
      {/* :id <- is dynamic */}
      <Route path="/note/:id" element={<NoteDetailPage />} />
    </Routes>

    </div>
  )
}

export default App
```
In Browser:



NoteDetailPage

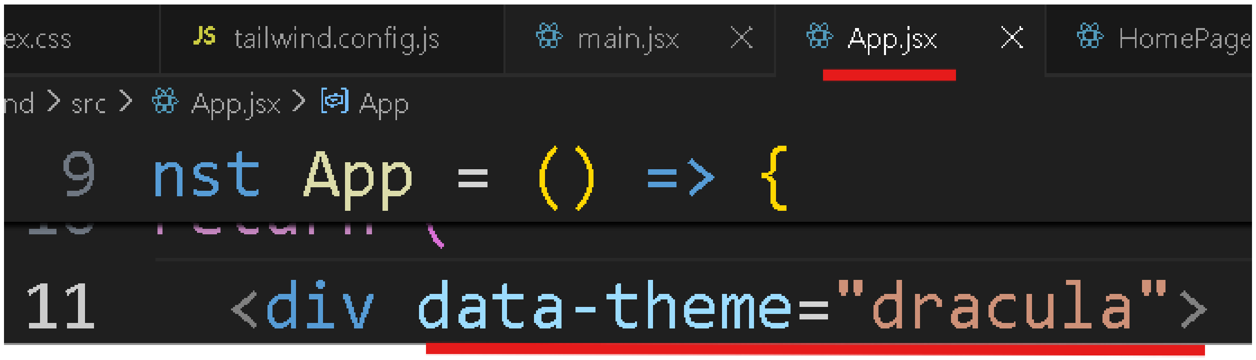
Changing to dracula theme:

In tailwind.config.js:



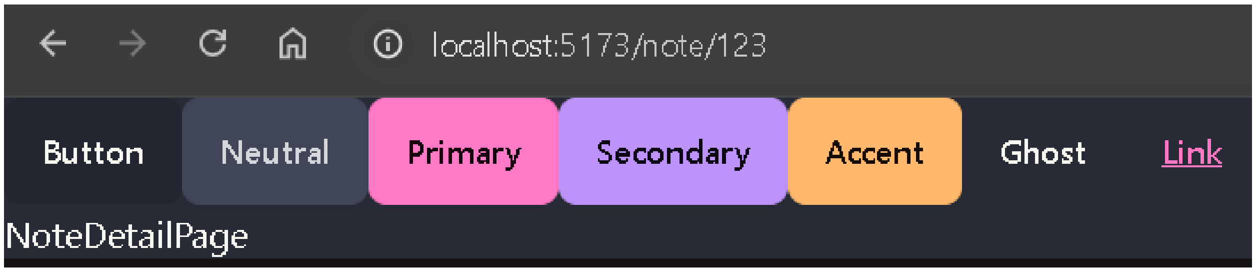
```
export default {
  daisyui:{
    // themes: ["light", "dark",
    themes: ["forest", "dracula"]
  },
}
```

In App.jsx:



```
nst App = () => {
  return (
    <div data-theme="dracula">
```

In browser:



NoteDetailPage

Code in index.css:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Code in tailwind.config.js:

```
import daisyui from 'daisyui';

/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [daisyui],
  daisyui:{
    // You can provide themes you like
    // themes: ["light", "dark", "forest"]
    themes: ["forest"]
  },
}
```

Code in main.jsx:

```
import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.jsx'
// importing the BrowserRouter
import {BrowserRouter} from "react-router"
// import Toaster for notifications
import {Toaster} from "react-hot-toast"

createRoot(document.getElementById('root')).render(
  <StrictMode>
    {/* Wrapping the application with BrowserRouter */}
    <BrowserRouter>
      <App />
      <Toaster />
    </BrowserRouter>
  </StrictMode>,
)
```

Code in App.jsx:

```
import React from 'react'
import { Route, Routes } from 'react-router'
import HomePage from './pages/HomePage'
import CreatePage from './pages/CreatePage'
import NoteDetailPage from './pages/NoteDetailPage'
// import toast
import toast from "react-hot-toast"

const App = () => {
  return (
    <div data-theme="forest">
      {/* <button className="btn">Button</button>
      <button className="btn btn-neutral">Neutral</button>
      <button className="btn btn-primary">Primary</button>
      <button className="btn btn-secondary">Secondary</button>
      <button className="btn btn-accent">Accent</button>
      <button className="btn btn-ghost">Ghost</button>
      <button className="btn btn-link">Link</button> */}
    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/create" element={<CreatePage />} />
      {/* :id <- is dynamic */}
      <Route path="/note/:id" element={<NoteDetailPage />} />
    </Routes>

    </div>
  )
}

export default App
```

Code in HomePage.jsx:

```
import React from 'react'

const HomePage = () => {
  return (
    <div>
      HomePage
    </div>
  )
}

export default HomePage
```

Code in CreatePage.jsx:

```
import React from 'react'

const CreatePage = () => {
  return (
    <div>
      CreatePage
    </div>
  )
}

export default CreatePage
```

Code in NoteDetailPage.jsx:

```
import React from 'react'

const NoteDetailPage = () => {
  return (
    <div>
      NoteDetailPage
    </div>
  )
}

export default NoteDetailPage
```