

The image shows a mobile application interface. At the top, there are two icons: a white circle with a blue arrow pointing left on the left, and a dark grey square with three horizontal white lines on the right. Below these, the main content area has a light grey header bar. The main text in the header reads "Vil du booke et ophold? Eller har du blot et spørgsmål?" (Do you want to book a stay? Or do you just have a question?). Underneath this, smaller text states: "Så tøv ikke med at tage kontakt til os herunder. Vi bestræber os på at svare på henvendelser indenfor 24 timer, men op til ferier kan der være travlt, og svartiden kan derfor være op til 48 timer." (Don't hesitate to contact us. We strive to respond to inquiries within 24 hours, but during holidays, it may take up to 48 hours due to high volume.)

Below the text is a circular image showing a white tent with an orange door set up in a lush green field with a small wooden table in the foreground.

On the left side of the form, there is a green circular icon containing a white letter "G". To its right, the text reads: "Skriv til os - Gitte svarer typisk inden for 24-48 timer." (Write to us - Gitte typically responds within 24-48 hours.)

Below this, the text "Kontakt Gitte" (Contact Gitte) is followed by a form with two input fields: "Navn \*"  
[Input Field]  
"Email \*"  
[Input Field]

Console Issues Autofill +

✖️ ⚡ top 🔍 Filter

Default levels ▾ 6 2 hidden ⚙️

▶ Array(0)

[MyList] /activities myList.jsx:52  
count = 5

[MyList] matched = myList.jsx:70  
▶ Array(0)

[MyList] favs = myList.jsx:31  
▼ Array(0) i

- length: 0
- ▼ [[Prototype]]: Array(0)
  - ▶ at: f at()
  - ▶ concat: f concat()
  - ▶ constructor: f Array()
  - ▶ copyWithin: f copyWithin()
  - ▶ entries: f entries()
  - ▶ every: f every()
  - ▶ fill: f fill()
  - ▶ filter: f filter()
  - ▶ find: f find()
  - ▶ findIndex: f findIndex()
  - ▶ findLast: f findLast()
  - ▶ findLastIndex: f findLastIndex()
  - ▶ flat: f flat()
  - ▶ flatMap: f flatMap()



# Vores ophold

Vores glampingophold kombinerer naturens ro med komfort. Vælg den pakke, der passer jer - og oplev luksus i det fri.



## Weekendtur

2 personer

Fra 4.200,-

[Book / Kontakt](#)

Læs mere →

## Familiepakken



Media College Denmark  
/ Medieskolerne

Ionut Catalin Belu



```
[plugin:vite:css] Failed to load PostCSS config (searchPath: C:/Users/bc/Desktop/glampsemopg/glampingopgave-main/sites/www): [ReferenceError] module is not defined in ES module scope
This file is being treated as an ES module because it has a '.js' file extension and 'C:\Users\bc\Desktop\glampsemopg\glampingopgave-main\sites\www\package.json' contains "type": "module". To treat it as a CommonJS script, rename it to use the '.cjs' file extension.
ReferenceError: module is not defined in ES module scope
This file is being treated as an ES module because it has a '.js' file extension and 'C:\Users\bc\Desktop\glampsemopg\glampingopgave-main\sites\www\package.json' contains "type": "module". To treat it as a CommonJS script, rename it to use the '.cjs' file extension.
    at file:///C:/Users/bc/Desktop/glampsemopg/glampingopgave-main/sites/www/postcss.config.js?t=1762939516130:1:1
    at ModuleJob.run (node:internal/modules/esm/module_job:271:25)
    at async importTracePromise._proto_ (node:internal/modules/esm/loader:578:26)
    at async req$3 (file:///C:/Users/bc/Desktop/glampsemopg/glampingopgave-main/sites/www/node_modules/vite/dist/node/chunks/config.js:8426:12)
    at async Object.search (file:///C:/Users/bc/Desktop/glampsemopg/glampingopgave-main/sites/www/node_modules/vite/dist/node/chunks/config.js:8243:24)

C:/Users/bc/Desktop/glampsemopg/glampingopgave-main/sites/www/src/styles/Fonts.css
`
```

Click outside, press `Esc` key, or fix the code to dismiss.  
You can also disable this overlay by setting `server.hmr.overlay` to false in `vite.config.js`.

The screenshot shows a browser developer tools window with the following details:

- Elements Tab:** Displays the DOM tree. The root node is `<div id="root">`. The tree includes various components like `MuiBox-root`, `MuiPaper-root`, `MuiPaper-elevation`, `MuiAppBar-root`, `MuiAppBar-colorPrimary`, `MuiAppBar-positionSticky`, `MuiPaper-root`, `MuiPaper-root`, `MuiPaper-root`, `MuiContainer-root`, `MuiContainer-maxWidthLg`, and `section` with `Toastify` content.
- DOM Breakpoints Tab:** Shows a list of breakpoints. The list is titled "No DOM breakpoints" and includes a note: "DOM breakpoints pause on the code that changes a DOM node or its children. [Learn more](#)".
- Sidebar:** A sidebar on the right lists several breakpoint locations, all pointing to `MyList.jsx` at line 27, line 22, or line 44. The entries include:
  - `[MyList] favs = ▶ (2) ['kano', 'vandring'] MyList.jsx:27`
  - `[MyList] favs = ▶ (2) ['kano', 'vandring'] MyList.jsx:22`
  - `[MyList] /activities MyList.jsx:44`
  - `count = 5`
  - `[MyList] matched = ▶ [] MyList.jsx:53`
  - `[MyList] /activities MyList.jsx:44`
  - `count = 5`
  - `[MyList] matched = ▶ [] MyList.jsx:53`
  - `[MyList] favs = ▶ (2) ['yoga', 'vandring'] MyList.jsx:27`
  - `[MyList] favs = ▶ (2) ['yoga', 'vandring'] MyList.jsx:22`
  - `[MyList] /activities MyList.jsx:44`
  - `count = 5`
  - `[MyList] matched = ▶ [] MyList.jsx:53`
  - `[MyList] /activities MyList.jsx:44`
  - `count = 5`
  - `[MyList] matched = ▶ [] MyList.jsx:53`
  - `[MyList] favs = ▶ [] MyList.jsx:27`
  - `[MyList] favs = ▶ (2) ['yoga', 'vin'] MyList.jsx:27`
  - `[MyList] favs = ▶ (2) ['yoga', 'vin'] MyList.jsx:22`
  - `[MyList] /activities MyList.jsx:44`

 r/reactjs • 4y ago  
Upbeat\_Combination74

## How the F do I style Material UI components ?

**Needs Help**

I am new to this project and i am pretty good with vanilla css. But i cant even seem to find the classes to style on MUI components ? Its pretty frustrating I am losing my mornings and evenings on trivial tasks.

 0   8  Share

 appwrite • Promoted

Meet Appwrite Sites: the open source Vercel alternative.

 [appwrite.io](#) 

Add your reply

Sort by: Best  Search Comments

 [deleted] • 4y ago

 SwitchOnTheNiteLite • 4y ago

I would suggest to maybe start here and see if that solves any of your issues?

<https://mui.com/customization/how-to-customize/>

 3   Reply  Award  Share ...

 Many\_Application7106 • 4y ago

Have a look @ my starter project <https://mansi1.github.io/material-ui-v5-starter/> There 8s an example for Theming and global styles

 2   Reply  Award  Share ...

 charlesdevin • 4y ago

I know the feeling. It's quite hardcore although it was worse not so long ago, and the documentation is not that friendly. It's full of small mistakes (like wrong imports) and it's not designed with really immersive examples that incentive you to experiment. Everything is very much hidden inside APIs and "themes" which you need to either try and guess or memorize how each component work.

In my opinion, material UI should be more simple. Examples need 123214129283 de imports to work and each API is different. If material UI components could be 100% affected by tailwindcss classes that would already ... Unfortunately it's not.

  Share ...

**Gitte's  
Glamping**

**BOOK NU**

Kom og prøv  
glamping hos Gitte!  
Vi er stolte af at byde dig velkommen til  
Gittes Glamping, hvor hjertevarme og  
omsorg møder naturens skønhed og  
eventyr. Vores dedikerede team, anført af  
Gitte selv, er her for at skabe den perfekte  
ramme om din luksuriøse  
udendørsoplevelse. Vi stræber efter at  
skabe minder og fordybelse, uanset om  
du besøger os som par, familie eller



Media College Denmark  
/ Medieskolerne

Ionut Catalin Belu

So you prefer to use no component library at all?

Θ Up 1 Down Reply Award Share ...



[deleted] • 2y ago

I don't.

Of course, I've build my own micro-libraries and components over time, I don't recode everything from scratch if I have something easy to recycle (*example: form validation, login validation, etc*). But I personally hate the idea of being forced to study a specific framework/system that basically does the same thing, with a different set of rules, a different "grammar" and so on.

Mastering vanilla css/JS lets you work in total freedom, with no fear of vendor lock-in and stuff like that.

Up 1 Down Reply Award Share ...



Mysterious-Meat4872 • 1y ago

bro chakra ui is the best, its highly customizable, you can do anything with that

Up 1 Down Reply Award Share ...



Media College Denmark  
/ Medieskolerne

Ionut Catalin Belu



r/Frontend • 2y ago  
raykage

...

## Do you think Material UI is hard to customize?



I think it can be really frustrating in the beginning (and sometimes even later on). However I haven't used other component libraries much.

What's your experience? Do you think it's harder than other libraries?

(If you have trouble with this, here's a video that goes over all the styling techniques and tips)



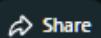
1



4



0



Share

Add your reply



Sort by: Best

Search Comments



[deleted] • 2y ago

Honestly, this whole "*check the component options first, then mess with the css, but maybe not, whoops let me have a look at the docs*" feels like a true nightmare.

I am a vanilla css/js/advocate, these kind of changes really don't take more than a few seconds. But maybe I am too old for this shit... !



2



0



Reply



Award



Share

...



r/reactjs • 2y ago  
sovereignman\_

...

## Styling MUI components is very ugly

Needs Help

I want to use MUI v5 for my web application. Obviously, I need to style some of the components. Normally, I am a React Native developer and love using

```
const styles = StyleSheet.create({})
```

Here is what I don't like in regular React with MUI

1. Using styled. I think it looks super cumbersome and wordy.

```
const Search = styled('div')(({ theme }) => {})
```

2. I also don't like using inline styles, as the component gets very messy

```
<Box sx={{ ...bla bla bla }}>
```

Is there a better way to approach MUI styling? I know I can use a regular object and pass it into the sx property, but that seems a bit unprofessional.

What is your approach to styling MUI components?

↑ 16 ↓

🕒 20



Share



Media College Denmark  
/ Medieskolerne

Ionut Catalin Belu

 UnknownWon • 2y ago

I'm slowly migrating away from MUI

This is enough of a reason for our project, as I work on a really data heavy tool with data streaming in to tables. My team now writes new components using PandaCSS

[↑ 2](#) [↓](#) [Reply](#) [Award](#) [Share](#) [...](#)

 Alternator24 • 2y ago

Oh, dear god. I feel your pain. I had to style lots of components for the company. I usually use `sx` prop. sometimes you need to go deeper and you should use `classes`. like to access `paper` class of drawer in MUI. and sometimes you should F\*\* everything and forget it about it. just use `createStyles` or `makeStyles`. do it from scratch.

I heard a lot about Ant Design and Chakra UI. idk if they are better than MUI in customization or not.

[↑ 2](#) [↓](#) [Reply](#) [Award](#) [Share](#) [...](#)

 MUI System • v7.3.5 •

[Getting started](#)  
Overview  
Installation  
**Usage**  
The sx prop  
Custom components  
Support

[Style utilities](#)  
[Components](#)  
[Migration](#)  
[APIs](#)  
[Experimental APIs](#)  
[Styles LEGACY](#)

**Performance tradeoffs**  
MUI System relies on CSS-in-JS. It works with both Emotion and styled-components.

**Pros**

- The sx prop uses a superset of CSS, so the syntax will be immediately familiar to you if you know CSS already. It also offers (optional) shorthand definitions that can save you time if you put in a little work to learn them upfront. These are documented in the [Style utilities](#) section of the primary navigation to the left.
- The System auto-purges, so that only the CSS that's used on the page is sent to the client. The initial bundle size cost is fixed—it doesn't get any larger as you add more CSS properties. You pay the cost of @emotion/react and @mui/system. The total size is ~15 kB zipped. But if you are already using an MUI Core component library like Material UI, then it comes with no extra overhead.

**Cons**  
Runtime performance takes a hit.

Benchmark case	Code snippet	Time normalized
a. Render 1,000 primitives	<div className="...">	100ms
b. Render 1,000 components	<Div>	112ms
c. Render 1,000 styled components	<StyledDiv>	181ms
d. Render 1,000 Box	<Box sx={...}>	296ms

Visit the [benchmark folder](#) for a reproduction of the metrics above.

We believe that for most use cases it's fast enough, but there are simple workarounds when performance becomes critical. For instance, when rendering a list with many items, you can use a CSS child selector to have a single "style injection" point (using d. for the wrapper and a. for each item).

**API tradeoff**



Media College Denmark  
/ Medieskolerne

Ionut Catalin Belu