Form Projects

Techstack

- Express JS
- Next JS
- Sqlite
- DaisyUI
- Tailwind CSS

How would you approach designing a layout that maintains consistency and usability across different screen sizes, from desktops to smartphones?

Designing a responsive layout using Tailwind CSS involves leveraging its utility-first approach and responsive design features. Here's how I can approach designing a layout that maintains consistency and usability across different screen sizes:

- Start with a Mobile-First Approach: Tailwind CSS is mobile-first by default, meaning you design for small screens first and then progressively enhance for larger screens. Begin by structuring layout to work well on smartphones.
- Use Responsive Design Utilities: Tailwind provides responsive design utilities that allow me to specify different styles based on screen sizes. For example, i can use classes like sm:, md:, lg:, and xl: to apply styles at specific breakpoints.
- Grid Layout: Tailwind offers a responsive grid system that allows I can create layout structures that adapt to different screen sizes. Utilize grid classes like grid-cols-{number}, col-span-{number}, and row-span-{number} to create flexible grids.
- Responsive Typography: Tailwind allows me to create responsive typography using its text utilities. i can adjust font sizes, line heights, and other typographic properties based on screen sizes using responsive classes.
- Flexbox and Grid for Layout: Tailwind provides utilities for Flexbox and CSS Grid layout, allowing you to create responsive and flexible layouts easily. Use classes like flex, flex-wrap, justify-{value}, items-{value}, and grid to create layouts that adapt to different screen sizes.
- Conditional Rendering: Tailwind does not include conditional rendering features like CSS media queries. However, i can use JavaScript frameworks like Vue.js or React to conditionally render components based on screen size or other factors.
- Media Queries: Utilize CSS media queries to apply different styles based on screen size, orientation, and other device characteristics. Define breakpoints where the layout changes to accommodate smaller or larger screens.

Backend:

- Project : https://cool-ionized-skink.glitch.me
- Repository : https://github.com/Kigamekun/TalentGrowthBE

Frontend:

- Project : https://dainty-pastelito-332b52.netlify.app
- Repository : https://github.com/Kigamekun/TalentGrowthFE

I deployed these 2 applications to 2 different places, namely glitch and also netlifly, for the front end I deployed it to netlifly by building next js, then pushed it to a repository then took the repository and collected it on netlifly, for the backend I used glitch.com which gives me the ability to import my own node project then run it on the demo server, they both have handled the problem of server settings if they don't exist for next JS, you can use PM2 for deployment, for express, you can use a server like nginx, etc.