Engineer

Found the light theme missing a whole layer

Found that the light theme had been missing a full-page overlay layer since the day it was written, by asserting that both themes paint every layered surface with the same number of layers.

September 2026

the overlay layer the light theme never had

Situation. The site ships two colour themes. The dark one is the default and gets looked at constantly; the light one is an override that appears only under a system preference, which means it is seen far less often and by nobody who is checking it. Theme parity is a class of defect where a surface is built once and restated once, and the restatement quietly drops a layer.

Task. Parity needed an assertion, because the only alternative is a person remembering to switch preference and look.

Action. The check is small and structural: for every layered surface, count the layers each theme paints, and fail when the counts differ. It does not compare appearances — themes are supposed to look different — it compares composition, which is the thing that should be identical. It found the defect on its first run. The site's full-page cover is three layers in the dark theme, a facet pattern over a scrim over the wall, and the light theme restated it with two. The facet overlay had been absent from the light theme from the day it was written, and nothing had ever said so, because a page missing one of three background layers looks like a design decision rather than a bug. The fix was one rule; the write-up afterwards recorded what the overlay costs in contrast, so the addition is measured rather than assumed to be free.

Result. A layer missing from one theme now fails a commit, and one that had been missing since it was written is painted. This is the shape of defect the whole checking approach is aimed at — nothing was broken, nothing errored, the page rendered correctly, and it had been wrong for months.

Bring this experience to your project

Use “Found the light theme missing a whole layer” as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Found the light theme missing a whole layer — the overlay layer the light theme never had

Explore our services Get in touch →

Does your dark mode hide bugs your light mode has?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Design Systems Engineer. It draws on Frontend Engineering, Automation & CI/CD, Testing & QA, Design Systems & UI and UX / UI Design. And while you are here, we are rather good at Frontend Development and UI/UX Design & Design Systems.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Does your dark mode hide bugs your light mode has?

Ask about the decisions behind “Found the light theme missing a whole layer”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/portfolio/found-the-light-theme-missing-a-whole-layer/