

# Foo Festival

## Team KLM

Katrine Bang Nielsen - [katr164m@stud.kea.dk](mailto:katr164m@stud.kea.dk)

Lasse Vestergaard - [lass315s@stud.kea.dk](mailto:lass315s@stud.kea.dk)

Mark Hjort - [mark204f@stud.kea.dk](mailto:mark204f@stud.kea.dk)

Antal anslag: 11.692



# Links

Link til Festivalsiden

<https://milk-foofest.vercel.app/>

Link til Ticket Flow

<https://milk-foofest-tickets.vercel.app/>

Link til Github Repository (festivalside)

<https://github.com/marhjo/nextjs-foofighter>

Link til Github Repository (ticket flow)

<https://github.com/marhjo/astro-solid-foofighter>

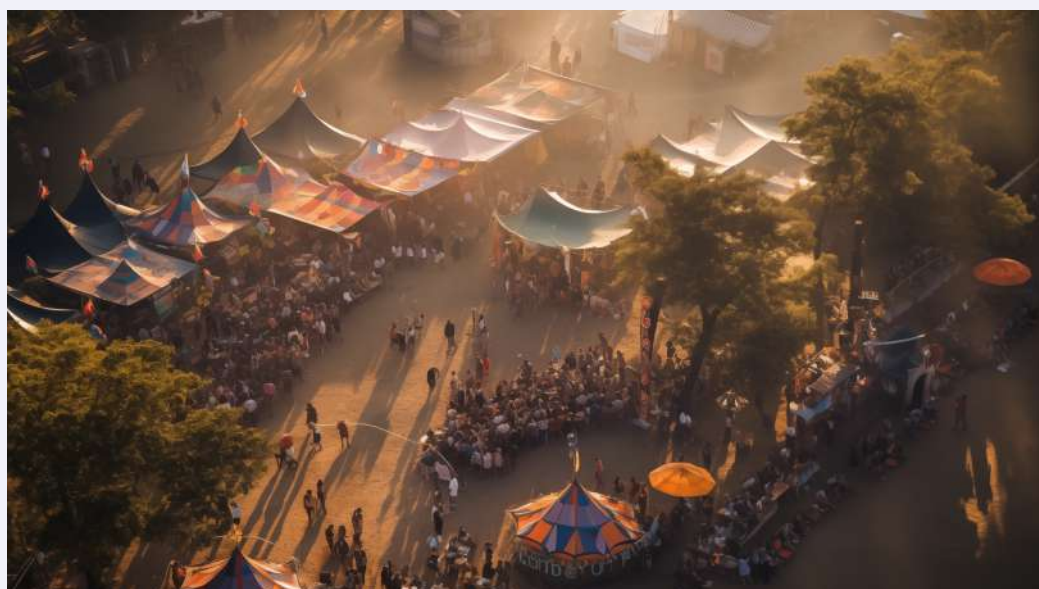
Link til Figma

<https://www.figma.com/file/SXrNJUJhuDw6eUIQkLYCpU/Design?type=design&node-id=0-1&mode=design&t=b6xj3MatFF4JeFny-0>

Link til videoer

[https://youtube.com/playlist?](https://youtube.com/playlist?list=PL8BRfgsw1EGQyoBoppJw6jysNuCXncr9y&si=VxJDu5RmXWFwJv6W)

[list=PL8BRfgsw1EGQyoBoppJw6jysNuCXncr9y&si=VxJDu5RmXWFwJv6W](https://youtube.com/playlist?list=PL8BRfgsw1EGQyoBoppJw6jysNuCXncr9y&si=VxJDu5RmXWFwJv6W)



# Introduktion

Vores løsning for Foo Festival er udviklet med et stærkt fokus på inklusivitet og mangfoldighed, hvilket afspejles i udbuddet af musik, der dækker et bredt spektrum af genrer. Konceptet er designet til at være enkelt og brugervenligt, således at festivalgæster nemt kan få et overblik over festivalens tilbud samt søge information om kunstnere og det planlagte program.

I vores arbejde med Foo Festival har vi anvendt vores ekspertise inden for design og programmering til at skabe en omfattende festival hjemmeside. Denne hjemmeside inkluderer en indbydende frontpage, der introducerer festivalen, en dedikeret kunstnerside, der fremhæver de optrædende kunstnere, en programside som detaljeret beskriver festivalens tidsplan, samt en billetsalgs side med en effektiv Ticket Flow-proces.



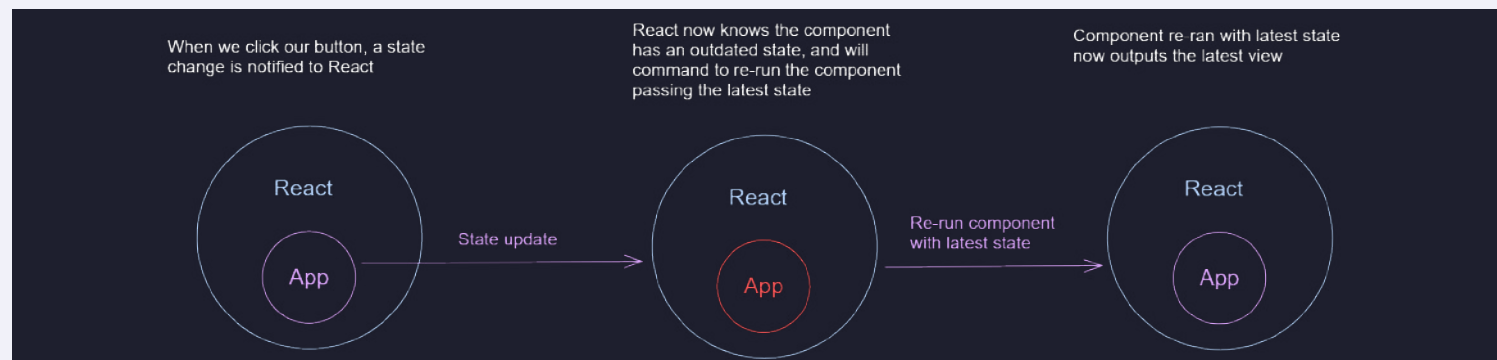
# Tech Stack

I starten af vores projekt tog vi en vigtig beslutning omkring tech stack og struktureringen af løsningen. Vi blev hurtigt enige om at opdele projektet i to separate løsninger, men som for brugeren ville fremstå som én integreret side.

Til den primære side valgte vi Next.js, da vi lige havde brugt det til Matas opgaven, og vi havde derfor god erfaring med dette framework. Next.js er også oplagt at bruge, da den tilbyder at bygge mere end én side. Til vores ticket flow valgte vi en kombination af Astro og SolidJS, da det skinner med sin reactivity og overlegne ydelse, især i scenarier med høje belastninger som en billetsalgs webside vil have.

React og SolidJS er begge moderne JavaScript-frameworks, der bruges til at bygge interaktive webapps, men de adskiller sig væsentligt i deres tilgang til opdatering og rendering af brugergrænseflader, som er grunden til vi brugte to forskellige frameworks gennem vores arbejdsprocess.

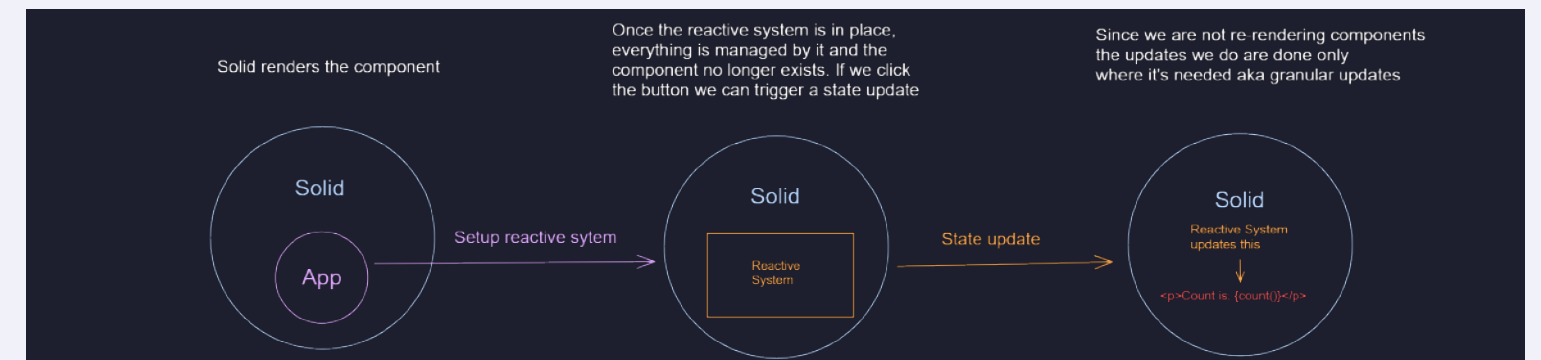
Under hele processen benyttede vi hovedsageligt Tailwind CSS til styling, hvilket sikrede at begge løsninger holdte samme stil.



Til kontrolstyring brugte vi Git, hvilket fungerede godt, da vi skulle samarbejde om at kode løsningen. Hvis der undervejs skulle ske fejl, kunne vi altid gå tilbage til de tidligere versioner. På Github ligger vores to repositories, hvor alle gruppemedlemmer har været collaborators. Via Github kunne vi linke til Vercel, som har hostet vores løsninger så den er offentlig tilgængelig.

I den sidste del af projektet, har vi benyttet Lighthouse testing for løbende at kunne få fokuspunkter til forbedringer. Dette hjalp os med løbende at identificere og rette problemer, såsom at tilføje alt-tekst til svg-filer, forbedre vores to løsningers ydeevne og accessibility.

Endelig valgte vi Supabase som vores database for Ticket Flow, hvor vi gemmer informationer om køb, såsom kunders navne, ordre-id'er, email-adresser og billettyper.



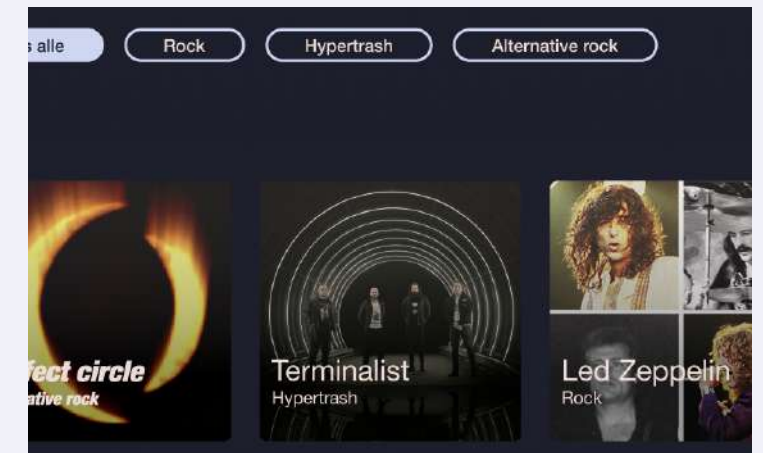
# Design- valg & proces

Som det første i designprocessen, snakkede vi sammen om, hvad vores ønsker var til festivalens look and feel. Efter enighed i teamet, valgte vi at arbejde med mørkere farver, nærmest som et dark theme, da vi syntes at det gav den rigtige stemning. Til farvevalget fandt vi inspiration fra fem udvalgte farver på hjemmesiden <https://tailwindcss.catppuccin.com/>

Med dette tema som grundlag begyndte vi vores design som hovedsageligt er opbygget af komponenter, som kan have enkelte variationer efter behov. Vi har taget udgangspunkt i Foo Festival som en kunde, der selv ville skulle opdatere siden løbende, efter vi har afleveret det færdige resultat. Et godt eksempel på dette kunne være billedet med tekst ved siden af som man finder på forsiden. Her kan kunden selv vælge billede og tekst samt hvilken rækkefølge det skal vises i.

Vi begyndte at lave et mock-up af vores løsning i Figma for at se, hvordan designet ville se ud som helhed. Vi designede løbende nye komponenter ud fra vores egne vurderinger af, hvad brugeren ville få gavn af. Et eksempel på dette var vores arbejde med designet af program siden, hvor vi løbende diskuterede, hvad man som festivalgæst havde brug for at vide. Da vi til sidst var tilfredse med sidernes komponenter og den overordnede opbygning, gik vi over til kodningen.

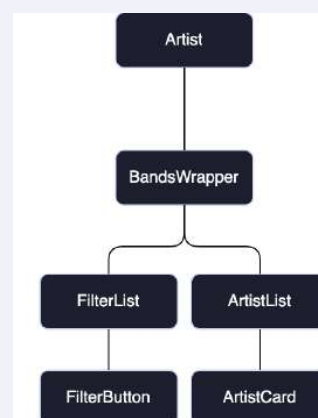
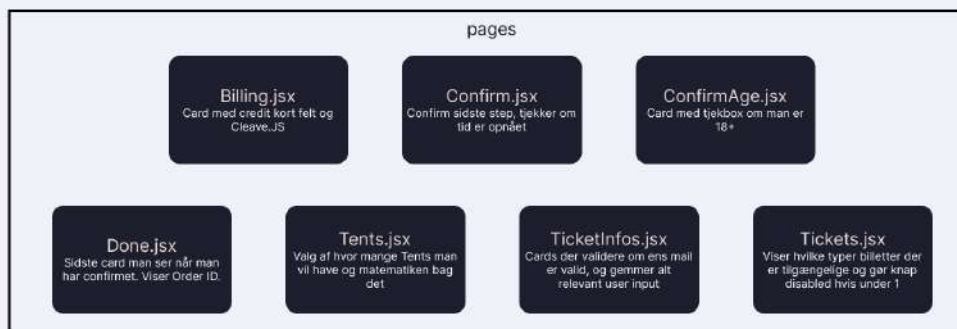
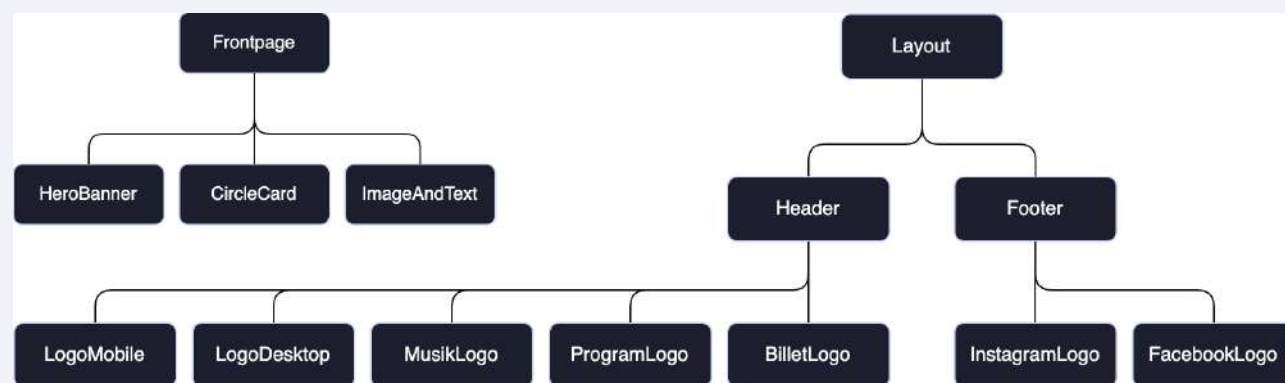
En ting, vi kunne have gjort bedre i vores designprocess, var at færdiggøre designet af vores løsning, inden vi begyndte at kode. Da vi ikke havde et færdigt design, skulle vi bruge mere tid på at tage nogle løbende designvalg, som vi ellers kunne have taget tidligere i processen. Selvom man skulle bruge mere tid i starten, ville man kunne spare tid, når der skulle kodes, da man ville være 100% klar over, hvordan designet skulle se ud.



# Git Branching Strategier

Til hovedsiden var vi to om at kode, så det var vigtigt, at vi arbejdede i separate branches, således at vi ikke kodede på det samme. På den måde undgik vi en del mergekonflikter. Der blev arbejdet i flere forskellige branches alt efter, hvilke sektioner der skulle kodes. Vi har dog ikke altid efterlevet en korrekt navngivning af alle branches. Her kunne vi sagtens have lavet flere branches, som var navngivet mere fyldestgørende, for at gøre det nemmere for udefrakommende at finde rundt i projektet.

Inden vi hver især fik merget vores rettelser fra en branch til main, sikrede vi os med hinanden, at det ikke ville konflikte med vores indbyrdes rettelser.



# Components & overview

Hver page på hovedsiden er opbygget af et fast layout som indeholder et header komponent og et footer komponent. Mellem disse to komponenter har Next.js automatisk placeret et {children} prop, hvor indholdet fra de enkelte page.jsx sider indsættes. Denne fungerer på samme måde som <Slot/> i Astro. Nedenfor har vi samlet et overblik over vores komponenter for hver enkelte page.

På vores Ticket Flow side har vi lavet syv sider, som bliver udskiftet via en state change ved brug af SolidJS CreateSignal. Disse sider bliver håndteret som komponenter, som vises på siden og udskiftes, når der trykkes "Next", efter man har udfyldt informationerne på siden.

Breakdown komponentet (Side.jsx) viser prisen for billetter, telte, og andre fees. Denne bliver vist på flere af de syv sider, og det er derfor et komponent for sig selv, da den bliver brugt oftere, hvorimod hver side kun er på siden en gang eller på et specifikt tidspunkt. Det er også her hvor noget af matematikken, som bruges til at beregne prisen, ligger.

FilterButton er et af de komponenter, som indgår på flere af vores sider. Vi benytter os af komponentet på Artists- og Program-siden, når vi skal filtrere på de forskellige kunstnere. Knappen skal dog opføre sig på som henholdsvis en radiobutton og en checkbox alt afhængig af, hvilken side man befinder sig på. Vi har derfor sendt en prop ned til komponentet, som angiver, hvilken type knap der er tale om.

# Vores arbejdsfordeling

## Lasse

### Generelt fokus

Kodning af [Slug]-siden, Header, Footer, og forsiden.

### Yderligere bidrag

Samarbejde med Katrine i design og kodning af hovedsiders komponenter og funktionaliteter.

### Design og funktionalitet valg

Bidrog til beslutningstagning om design og funktionaliteter på Ticket Flow-siden og andre sider, men med primært fokus på de sider, han kodede.

## Katrine

### Generelt fokus

Ledende rolle i kodningen af Program- og Kunstner-siderne samt andre kode intensive dele af den generelle side.

### Yderligere bidrag

Samarbejde med Lasse på hovedsiders design og komponenter.

### Designbeslutninger

Involveret i design- og funktionalitets valg for Ticket Flow-siden og andre sider.

## Mark

### Generelt fokus

Hovedansvarlig for kodning af Ticket Flow-siden.

### Yderligere bidrag

Lidt bugfixing og optimering på de andre sider.

### Design og farvevalg

Aktiv deltagelse i gruppearbejde med Figma, mock-up og farvevalg, og andre beslutninger om design i gruppen.



# Vores valg af “Ekstras”

## SolidJS

Til Foo Festivalens Ticket Flow-side har vi valgt SolidJS grundet dets høje ydeevne. SolidJS sikrer, at vores billetsalgssystem fungerer effektivt og problemfrit, selv under meget høj belastning, hvilket er afgørende for en god brugeroplevelse.

## Container queries plugin til Tailwind CSS

Til stylingen af vores ArtistCard komponent på Artists-siden har vi installeret et container query plugin. Vi har tilføjet klassen @container og givet vores “h2”- og “p”-tags klassen @[15rem]. Skriftstørrelsen vil således ændre sig, når containeren får en bredde på 15rem.

## NextJS local fonts

Importer af fonte i Next.js fungerer på en anden måde, end vi har været vant til. Vi har downloadet to eksterne fonts og importeret dem som local fonts. Inde i tailwind.config har vi oprettet dem som custom variable, så vi nemt har kunnet anvende dem i projektet.

## Cleave / NanoID

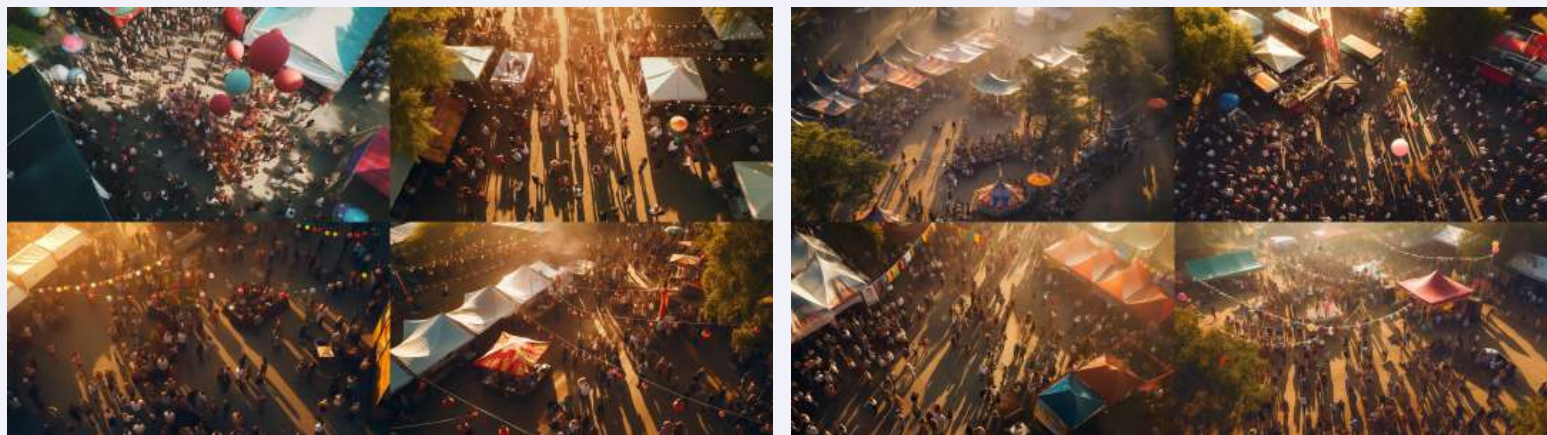
På Ticket Flow-siden af Foo Festival hjemmesiden har vi specifikt anvendt Cleave.js og NanoID for at gøre brugeroplevelsen bedre. Cleave.js forbedrer brugerinput i billetkøb formularen ved at formatere input felter såsom kreditkortnumre på en intuitiv måde. Samtidigt bruger vi NanoID til at tildele hvert billetkøb et unikt, kort ID, hvilket giver en fordel frem for længere UUID'er.



# Vores brug af AI

Da vi skulle finde billeder til forsiden, anvendte vi Midjourney. Dette var et godt redskab til at skabe passende billeder, når Unsplash og andre sider ikke havde brugbare billeder. Midjourney blev instrueret med specifikke prompts for at generere billeder i den ønskede stilart og indhold.

Til tekstproduktion brugte vi hovedsageligt ChatGPT til at generere indhold til tekstfelter på forsiden og undersider. Vi har også i nogle tilfælde brugt det til oversættelse fra Dansk til Engelsk og andet copy på websiden. ChatGPT bidrog også i kodningsprocessen, især med rådgivning og fejlrettelse ved implementering af funktioner som filtrering og slug-sider for kunstnere.



Festival from above with people partying , Photojournalism, 4k, Natural Lighting --ar 16:9

You

I want to make this a dynamic component I can use in my main component. I want to be able to flip around the text so the text and image switch side



```
export default function ImageAndText(props) {  
  const { image, overskrift, text, layout } = props;  
  
  const isReversed = layout === "reversed"; // Check if layout is reversed  
  
  return (  
    <article className={isReversed ? "flex flex-direction-reverse" : "flex flex-direction-normal"}>
```

# Potentielle videreudvikling muligheder

## Forbedre Accessibility

- Vi vil gerne forbedre accessibility på specielt vores ticket flow, da den lige nu ikke er fyldestgørende. For at forbedre dette kan man tilføje ARIA-labels for bedre skærmlæser kompatibilitet, forbedre 'tab index' for nemmere tastaturnavigation, og anvende mere semantisk kode for at gøre siden mere tilgængelig for alle brugere.

## Scroll på Program & Artist Page

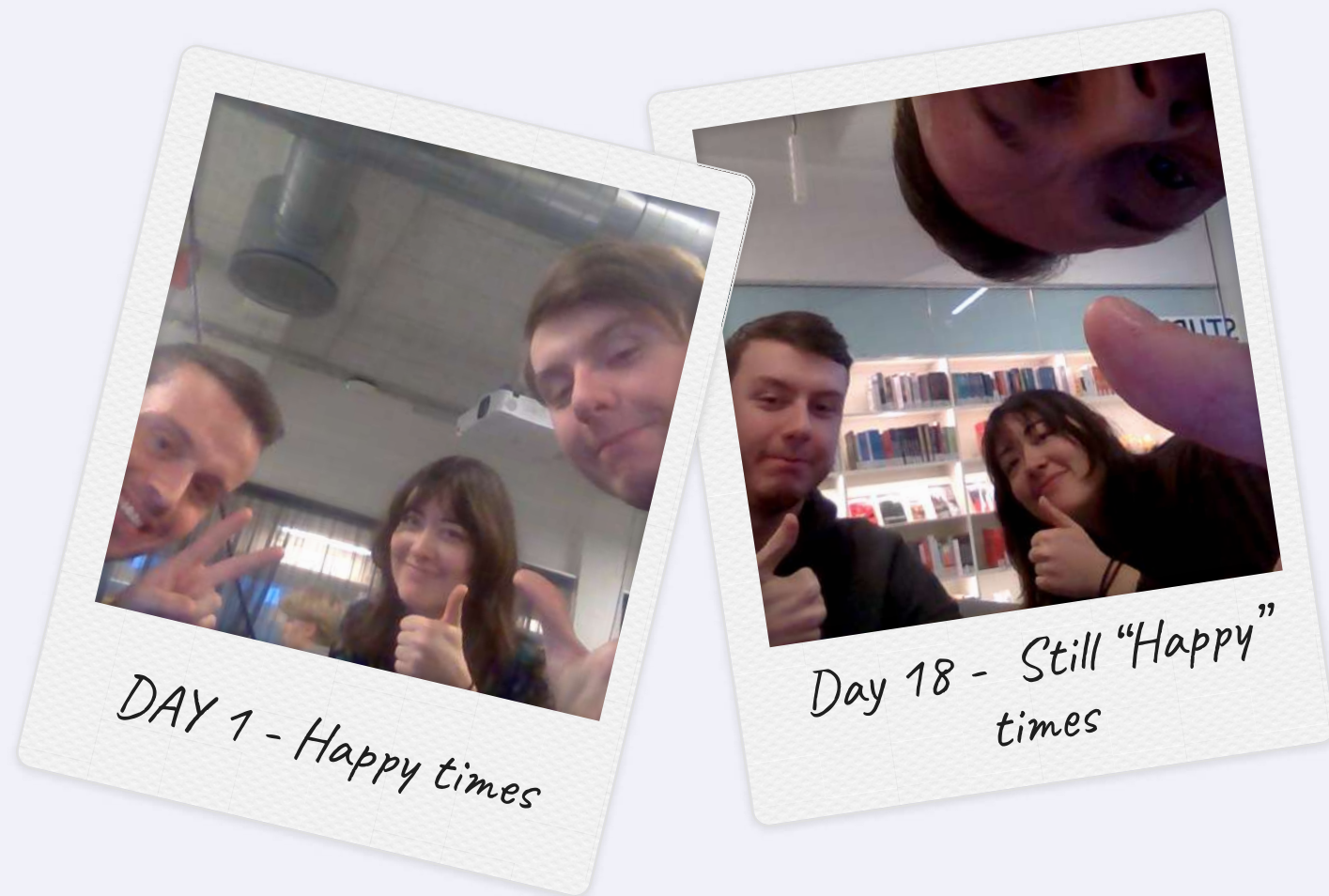
- En anden måde, hvorpå man kan scrolle gennem de forskellige genrer på Artists og de forskellige dage på Programsiden. Dette kunne eksempelvis være en knap, som scroller hver gang man trykker.

## Bedre Kunstner information / navigation

- Tilføje scenenavn og dag på hver kunstner på slug siden, og gøre bandnavnet klikbare på Programsiden. Det kan forbedre brugeroplevelsen, at man kan klikke sig ind på en kunstner og se hvilken dag de spiller samt fra hvilken scene.'

# Konklusion og fokuspunkter frem til eksamen

I gennem vores arbejde med Foo Festival websiden er vi kommet frem til en potentiel løsning, som vi mener er et godt bud med et stærkt fokus på inklusivitet og mangfoldighed. Vi mener at designet og de funktionaliteter vi har tilføjet, giver en god brugeroplevelse, som er med til at skabe et overblik for den besøgende. Op til den mundtlige eksamen vil vores fokuspunkter være at lave relevant user testing for at finde mulige forbedringspunkter, som vi vil videreudvikle på.



## Tænke-Højt-Tests

Afprøvning af begge løsninger med en bred målgruppe for at evaluere brugeroplevelsen og interaktionen. Her forventer vi at få inspiration til eventuelle rettelser til vores design.

## Bipolar Emotional Response Tests (BERT)

Udførelse af BERT for at vurdere løsningernes følelsesmæssige interaktioner med brugere, og på den måde sikre, at design og brugervenlighed er godt.

## General Bug Testing og Videreudvikling/Optimering

Vi vil foretage en omhyggelig gennemgang og rettelser af tekniske fejl for at sikre stabilitet og pålidelighed, samtidig med at vi vurderer mulighederne for yderligere forbedringer eller tilføjelse af nye funktioner, der kan øge effektiviteten af vores løsninger.