MATHIEU DUBART

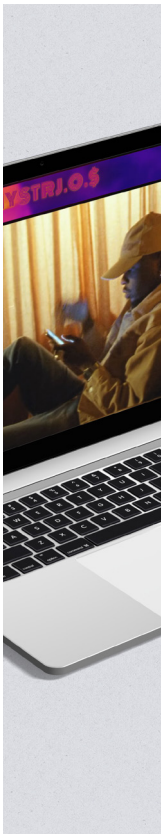
Book Technique

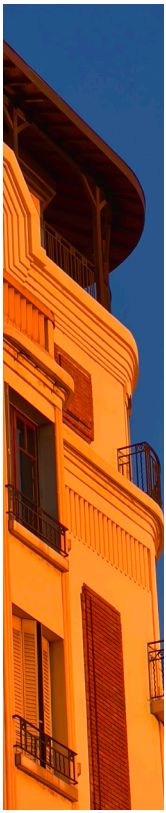
Gobelins

SOMMAIRE

SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE
SOMMAIRE

SOMMAIRE





PRESENTATION
PRESENTATION
PRESENTATION

Je m'appelle Mathieu DUBART, et je suis étudiant en première année d'un bachelor de développement web et d'applications mobiles à l'école des Gobelins sur Annecy. Créatif, j'aime découvrir de nouvelles choses et apprendre de nouvelles techniques.

1 Présentation

Pour le premier projet de l'année, nous devions créer un site internet sans utiliser de JavaScript.

Le mien est un site nommé « Rap Gallery », site sur lequel on retrouve plusieurs artistes urbains français ainsi que leurs projets, triés dans l'ordre alphabétique.

2

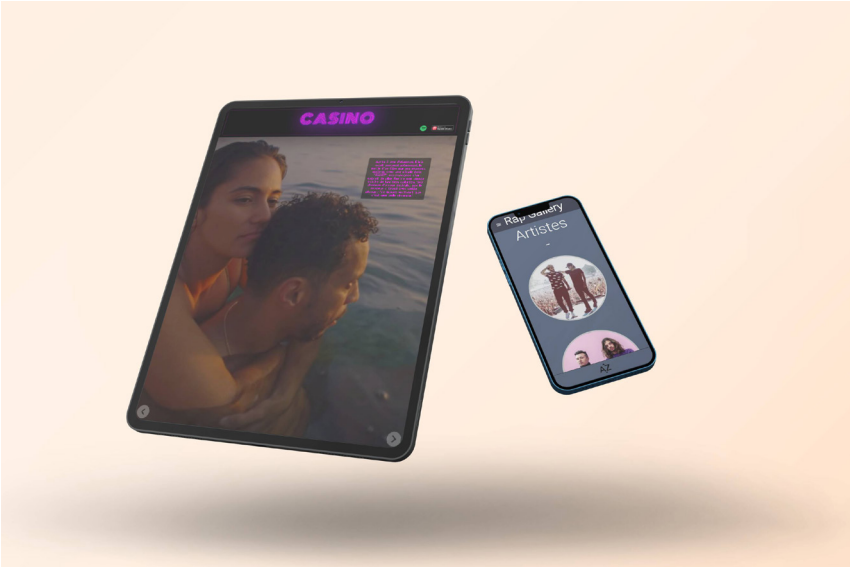
HTML & CSS

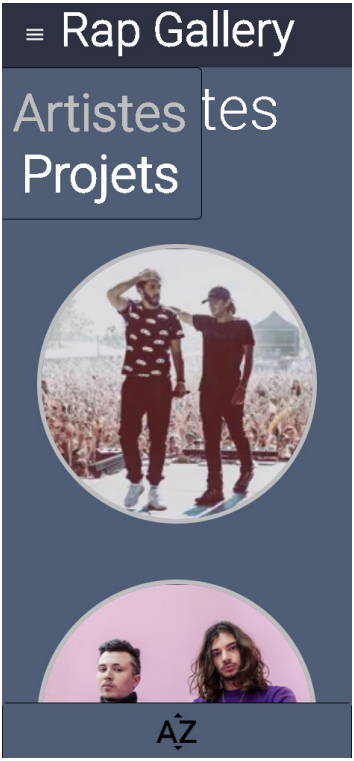
Pour naviguer sur le site, il suffit de choisir la catégorie dans la barre supérieure puis une lettre dans la barre latérale droite. La version mobile dispose de deux menus, un «hamburger» et un déroulant.

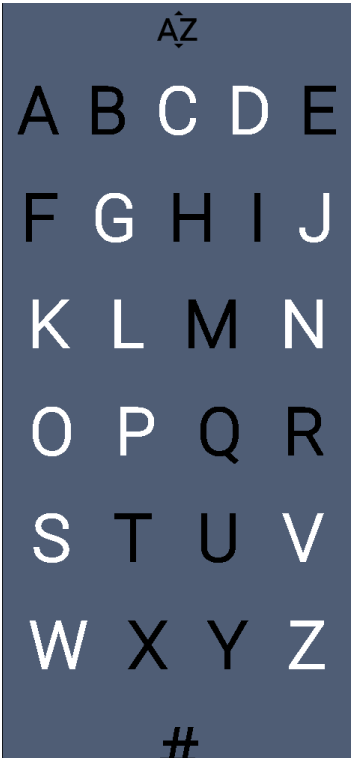
RAP

Pour la partie «Artistes», lorsque l'on survole la photo d'un artiste ou d'un groupe avec la souris, son nom s'affiche et, lorsque l'on clique dessus, nous sommes alors renvoyés sur le profil Apple Music de ce dernier.

Quant à la partie «Projets», le même effet est appliqué, renvoyant alors vers la page Apple Music de l'album en question. De plus, trois projets disposent d'un page plus personnalisée, avec des visuels en fond ainsi que des petits textes à propos du projet, page personnalisée qui peut facilement être étendu à tous les projets.








```
<div id="menuAlphabet">
        <div id="imageMenuAlphabet">
            <a href="#menuAlphabet" id="lienMenuAlphabet"> <img src="img/alphamenu.svg" id="imgMenuAlphabet"/> </a>
            <a href="javascript:history.back()" id="lienMenuAlphabet2"> <img src="img/alphamenu.svg" id="imgMenuAlphabet"/> </a>
        </div>
    <div id="navAlphabet">
        <a class="lienAlphabetNone"> A </a>
        <a class="lienAlphabetNone"> B </a>
        <a class="lienAlphabet" href="#C"> C </a>
        <a class="lienAlphabet" href="#D"> D </a>
        <a class="lienAlphabetNone"> E </a>
        <a class="lienAlphabetNone"> F </a>
        <a class="lienAlphabet" href="#G"> G </a>
        <a class="lienAlphabetNone"> H </a>
        <a class="lienAlphabetNone"> I </a>
        <a class="lienAlphabet" href="#J"> J </a>
        <a class="lienAlphabet" href="#K"> K </a>
        <a class="lienAlphabet" href="#L"> L </a>
        <a class="lienAlphabetNone"> M </a>
        <a class="lienAlphabet" href="#N"> N </a>
        <a class="lienAlphabet" href="#O"> O </a>
        <a class="lienAlphabet" href="#P"> P </a>
        <a class="lienAlphabetNone"> Q </a>
        <a class="lienAlphabetNone"> R </a>
        <a class="lienAlphabet" href="#S"> S </a>
        <a class="lienAlphabetNone"> T </a>
        <a class="lienAlphabetNone"> U </a>
        <a class="lienAlphabet" href="#V"> V </a>
        <a class="lienAlphabet" href="#W"> W </a>
        <a class="lienAlphabetNone"> X </a>
        <a class="lienAlphabetNone"> Y </a>
        <a class="lienAlphabet" href="#Z"> Z </a>
        <a class="lienAlphabetNone"> # </a>
    </div>
</div>
```

CODE

Pour le menu burger, présent uniquement sur la version mobile, j'ai utilisé un input «checkbox» que j'ai relié à l'image correspondante, afin d'afficher le menu de navigation quand celui-ci est coché, et de le cacher dans le cas contraire.

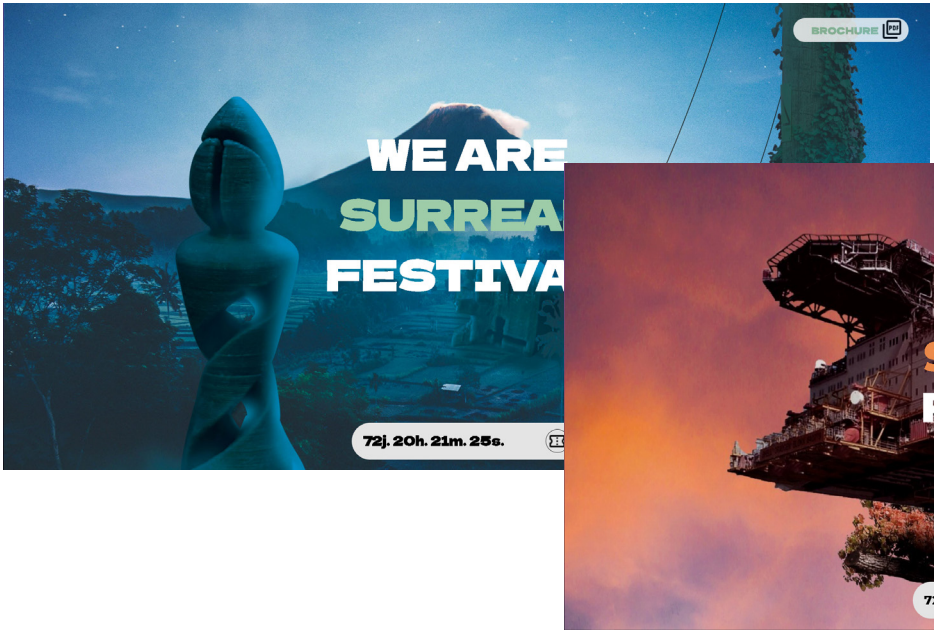
Pour le menu alphabétique, c'est une simple div qui apparait lorsque l'on clique sur l'image, et qui disparait lorsque l'on reclique.

```css
#imageMenuAlphabet {
    background-color: #4f5d75;
    width: 100%;
    height: 60px;
    display: flex;
    justify-content: center;
    align-items: center;
}

#imgMenuAlphabet{
    background-color: #4f5d75;
    height: 50px;
    width: 50px;
    display: flex;
    justify-content: center;
    align-items: center;
}

#menuAlphabet {
    height: 100vh;
    top: calc(100% - 60px);
    position:  fixed;
    z-index: 11;
    border:  black 1px solid;
    border-radius: 5px;
    transition: all 0.5s;
}

#menuAlphabet:target {
    top: 0px;
}


#navAlphabet{
    width: 100%;
    left: 0px;
    flex-direction: row;
    justify-content: space-evenly;
    flex-wrap: wrap;
    position: relative;
    top:  0px;
    border: none;
}
```

Pour les titres des projets sur les pages personnalisées de ceux-ci, j'ai utilisé une typographie importée de Google Fonts et rappelant les néons (la Monoton), afin de faire écho au coté urbain des clips de rap, et j'ai animé le tout à l'aide de keyframes pour faire clignoter certaines lettres à intervalle régulier et en créant des décalages entre certaines.

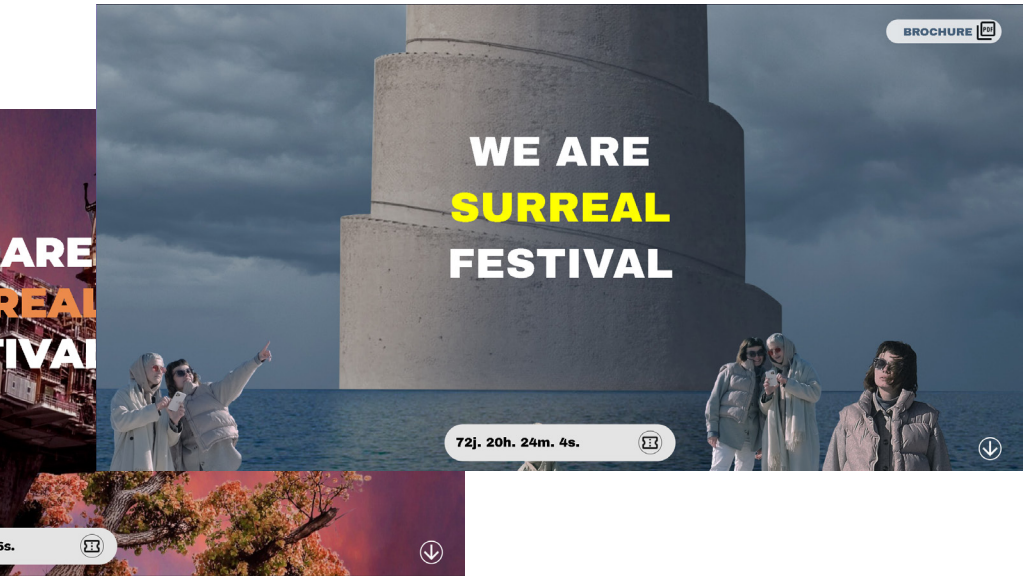
Pour garder l'ambiance propre à chaque projets, j'ai également choisi des couleurs ainsi que des visuels tirés directement des clips issus de ces derniers.

PROJETS



3

Workshop



WE ARE

URREAL

```
let str = window.location.href;
let url = new URL(str);
let search_params = new URLSearchParams(url.search);
if(search_params.has('version')) {
  let version = search_params.get('version');
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            let response_V1=this.responseText;
            let infos = JSON.parse(response_V1);
            pageContainer.style.backgroundColor = infos.colorOne
        }
    }
}
```

Pour le premier workshop de l'année, nous avons travaillé en collaboration avec les designers, et nous devions alors réaliser trois versions différentes d'un même site internet, le tout pour un festival surréaliste.

Afin d'avoir trois versions du site sur un seul fichier HTML, j'ai créé, dans mon DOM, des blocs communs pour chaque élément puis, à l'aide du paramètre récupéré dans l'url avec la fonction ci-dessus, j'ai pu ajuster le contenu et le css en fonction de la version demandée.

```
{
    "fontFamily": "'Archivo Black', sans-serif",
    "fontSize": "32px",
    "fontSizePDF": "24px",
    "colorOne":"white",
    "colorTwo":"#405d79",
    "colorThree": "#262b32",
    "surrealColor": "yellow",
    "backgroundImage": "url('version_1/img/backgroundImage.jpg')",
    "countdownTop": "calc(50% - 20px)",
    "pdf":"version_2/brochure.pdf",
    "border": "2px solid #405d79",
    "fontForBilling": "'Archivo Black', sans-serif",
    "concert1_2": "version_1/img/concerts/ccrt1_2.png",
    "concert3": "version_1/img/concerts/ccrt3.png",
    "concert4": "version_1/img/concerts/ccrt4.png",
    "conference1": "version_1/img/conferences/cfr1.png",
    "conference2": "version_1/img/conferences/cfr2.png",
    "conference3": "version_1/img/conferences/cfr3.png",
    "expo1": "version_1/img/expos/exp1.png",
    "expo2": "version_1/img/expos/exp2.png",
    "performance1": "version_1/img/performances/prf1.png",
    "performance2": "version_1/img/performances/prf2.png",
    "workshop1": "version_1/img/workshop/wrk1.png",
    "workshop2": "version_1/img/workshop/wrk2.png",
    "instagramLink": "https://instagram.com/surreal.festival",
    "instagram1": "version_1/img/instagram/instagram1.png",
    "instagram2": "version_1/img/instagram/instagram2.png",
    "instagram3": "version_1/img/instagram/instagram3.png",
    "instagram4": "version_1/img/instagram/instagram4.png"
}
```

Pour les informations et photos relatives à chaque version, je les ai rangées dans des fichiers json, afin de pouvoir les injecter facilement dans le dom en fonction du paramètre passé dans l'url.

CODE

```
// Update the count down every 1 second
let x = setInterval(function() {

  // Get today's date and time
  let now = new Date().getTime();

  // Find the distance between now and the count down date
  let distance = countDownDate - now;

  // Time calculations for days, hours, minutes and seconds
  let days = Math.floor(distance / (1000 * 60 * 60 * 24));
  let hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
  let minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
  let seconds = Math.floor((distance % (1000 * 60)) / 1000);

  // Display the result in the element with id="demo"
  document.getElementById("countdown").innerHTML = days + "j. " + hours + "h. "
  + minutes + "m. " + seconds + "s. ";

  // If the count down is finished, write some text
  if (distance < 0) {
    clearInterval(x);
    document.getElementById("demo").innerHTML = "Le festival a commencé";
    bookingButton.style.display = "none"
```

Il y a également un compteur (en position sticky, afin de toujours l'avoir affiché, peu importe la position de l'utilisateur sur la page), qui se met à jour dynamiquement chaque seconde.

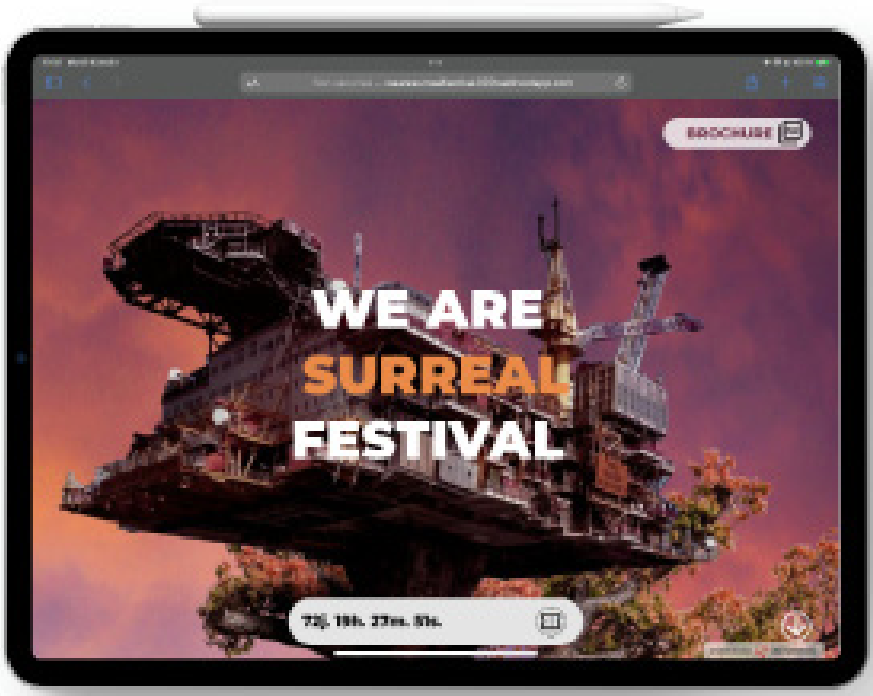
```
// ------------------ Swiper  ------------------//

let swiper = new Swiper(".mySwiper", {
  effect: "coverflow",
  grabCursor: true,
  centeredSlides: true,
  slidesPerView: "auto",
  coverflowEffect: {
    rotate: 0,
    stretch: -80,
    depth: 100,
    modifier: 1,
    slideShadows: false,
  },
  autoplay: {
    delay: 5000,
  },
});
```

Pour les carousels des évènements, j'ai utilisé l'API Swiper, ce qui m'a fait gagner un temps non négligeable et m'a beaucoup facilité la vie.

RESPON

Pour essayer par vous-même,
scannez ce QR Code

NSIVE



Pour essayer par vous-même,
scannez ce QR Code

REMERCIEMENTS
REMERCIEMENTS
REMERCIEMENTS

Je remercie Maddy MISTO, la designeuse sans qui ce book n'aurait pas été possible, ainsi que toutes les personnes m'ayant toujours soutenu et m'ayant permis d'arriver jusqu'ici.

4

Remerciements

github.com/MathieuDubart

06.67.19.29.99
mathieu.dubart@icloud.com