

LY Energy

Luis Pedro Pinheiro & Yann Menoud

1 Description

Notre site consistera à présenter un nouveau produit d'une entreprise fictive. Nos sources d'inspirations sont les différentes pages de vente d'Apple où le produit est présenté dans plusieurs sections qui s'affichent à l'aide du scroll.

Elles contiennent plusieurs types d'animations, des images qui s'affichent peu à peu, l'animation de texte et des graphiques animés pour présenter les spécificités de notre produit.

Notre site contiendra trois pages, soit :

- Accueil
- Formulaire de contact
- À propos de nous

2 Produit

Nous avons choisi de partir sur la présentation de la nouvelle boisson énergisante de l'entreprise LY ENERGY.

3 Sources d'inspiration

- Apple Vision Pro
<https://www.apple.com/apple-vision-pro/>
- Apple iPad Pro
<https://www.apple.com/ipad-pro/>

4 Technologies utilisées

N'ayant jamais utilisé VueJS, nous avons décidé d'explorer cette technologie. Il était intéressant pour nous de pouvoir tester et implémenter toutes sortes de composants VueJS.

Voici la liste complète des technologies utilisées :

- Vue.js
- Tailwind CSS
- Git
- GitHub Project

5 Bibliothèques utilisées

- Vueuse/motion (les composants qui arrivent de la gauche et de la droite)
- Chart.js (graphiques)
- Flowbite (menus et éléments graphiques)
- Flowbite-vue (menus et éléments graphiques)
- Scrolly-video (Vidéo sur le scroll)

6 Points difficiles

6.1 Vidéo

Selon la documentation, l'installation et l'implémentation d'une bannière vidéo fonctionnant en fonction du scroll est relativement simple, cependant il y a pas mal de petits imprévus à régler qui nous ont pris du temps.

Voici comment nous avons ajouté la vidéo à notre composant.

```
<div class="scrolly-video-container">
  <ScrollyVideo src=/video.mp4 frameThreshold=0.001 sticky=true />
</div>
```

Après ceci, nous avons donc dû tester beaucoup de méthodes différentes en CSS afin que cela colle parfaitement avec ce que nous voulions faire.

```
.scrolly-video-container {
  height: 300vh;
}

.scrolly-video-container video {
  object-fit: cover !important;
  object-position: 20% !important;
}
```

Comme nous pouvons le voir, `object-position` est crucial dans notre code CSS, car sans celui-ci, sur mobile, le focus n'était pas fait sur l'une des cannettes. Cela rendait donc pas très bien.

6.2 Graphiques

Les graphiques ont été l'étape la plus compliquée de notre projet. Il a fallu comprendre la syntaxe de vue en profondeur puis d'implémenter une première version simplifiée d'un graphique. Celui-ci a finalement évolué pour devenir les deux graphiques qui sont actuellement affichés sur notre site.

Ce qui nous a donné beaucoup de fil à retordre était que l'animation des graphiques se passait qu'une seule et unique fois, c'est-à-dire quand nous arrivions en fond de page. Cependant, si nous remontions et que nous redescendions, l'animation ne se reproduisait pas.

Nous avons donc dû implémenter un observer par graphique afin de les détruire lorsque nous remontions, et les recréés quand nous redescendions.

Voici le code nécessaire à cette tâche.

```
onMounted(() => {
  const observerPie = new IntersectionObserver(
    (entries) => {
      entries.forEach((entry) => {
        if (barP !== null && !entry.isIntersecting) {
          barP.destroy();
          barP = null;
        } else if (entry.isIntersecting && barP === null) {
          createPieChart();
        }
      });
    },
    { threshold: 0.2 }
  );
  const observerBar = new IntersectionObserver(
    (entries) => {
      entries.forEach((entry) => {
        if (barC !== null && !entry.isIntersecting) {
          barC.destroy();
          barC = null;
        } else if (entry.isIntersecting && barC === null) {
          createBarChart();
        }
      });
    },
    { threshold: 0.2 }
  );

  observerPie.observe(pieChart.value);
  observerBar.observe(barChart.value);
});
```

7 Installation

Il est **requis d'avoir Node.js** installé sur sa machine.

Installer les dépendances

npm install

Lancer le serveur de développement

npx vite

Ouvrir le projet dans le navigateur

http://localhost:5173 ou le lien indiqué dans la console
