

Projet AWA1 Notus Dashboard

Ce projet a pour but d'apprendre à intégrer des librairies et outils existants au sein du framework NextJS. Mon objectif a également été de réaliser des animations en utilisant différentes techniques tel que le JavaScript avec Framer Motion pour modifier les propriétés de style sur l'élément HTML. J'ai également pu tester les animations dans des Canvas à l'aide de particles.js ou encore Chart.js. Pour finir, j'ai implémenter des animations à l'aide de transitions CSS. Ces différentes techniques m'ont permis d'observer les avantages et désavantages de chacune d'entre elles.

Template de départ pour le projet : <https://www.creative-tim.com/product/notus-nextjs>

Le site est déployé sur Netlify à l'adresse : <https://boring-davinci-9df57f.netlify.app/>

Librairies et Outils

- [Framer Motion](#)
- [ParticlesJS](#)
- [TailwindCSS](#)
- [ChartJS](#)
- [SVGMap](#)
- [react-countup](#)

La liste complète des librairies peut être retrouvée dans le [package.json](#)

Technologies

NextJS

NextJS est un framework autour de la librairie React qui permet de créer des sites web en utilisant une approche de rendu hybride. Il est en effet possible avec NextJS de faire du rendu côté serveur, mais il est également possible de faire du rendu côté client ou de mélanger les deux. Ceci permet aux utilisateurs d'obtenir une meilleure expérience d'utilisation. Cela permet, par exemple, de faire un rendu côté serveur du layout de la page, puis de charger du contenu dynamique sur le client et d'animer le rendu à cet instant.

TailwindCSS

TailwindCSS est un framework CSS "utility-first", il met à disposition des classes utilitaires, contrairement à d'autres outils qui vont proposer des classes orienté sur une approche "component". La version 3 de TailwindCSS est proposée avec un compilateur JIT, cela permet de ne pas avoir à définir quelles seront les classes utilisées dans le projet, c'est le compilateur qui va inspecter le code source et déterminer les classes à ajouter à la feuille de style.

Fonctionnalités

- Carte interactive avec affichage de statistiques
- Chargement des données de la carte à partir d'un endpoint d'API NextJS
- Progressive Web App (à l'aide du package next-pwa)
- Fond animé sur le layout d'authentification (à l'aide de particles.js)

- Migration du template de Javascript à Typescript
- Animations sur la page d'accueil (stagger, translate on-show, etc) avec Framer Motion
- Animations CSS sur changement d'état des éléments (hover par ex.)
- Déploiement automatique sur Netlify

Exemple de code

Exemple de changement d'état lorsque 50% de la div se trouve dans le viewport :

```
<motion.div
  variants={{
    hidden: {
      opacity: 0,
    },
    visible: {
      opacity: 1,
    }
  }}
  initial="hidden"
  whileInView="visible"
  viewport={{ once: true, amount: 0.5 }}
></motion.div>
```

Exemple de stagger :

```
const container = {
  visible: {
    transition: {
      staggerChildren: 0.15,
      delayChildren: 0,
    },
  },
};

const item = {
  hidden: {
    opacity: 0,
    top: 100,
  },
  visible: {
    opacity: 1,
    top: 0,
  },
};

<motion.div initial="hidden" whileInView="visible" viewport={{ once: true
}}>
  <motion.div variants={item}></motion.div>
  <motion.div variants={item}></motion.div>
```

```
<motion.div variants={item}></motion.div>  
</motion.div>;
```

API

L'API permet de récupérer les statistiques de chaque État. Elle est accessible à l'adresse </api/stats?state=ny>. Les données sont chargées à partir d'un fichier JSON ayant comme structure :

```
[  
  {  
    "id": "ak",  
    "data": {  
      // Statistiques globales de l'État  
      "target": [76, 24],  
      // Statistiques des villes  
      "cities": [  
        { "name": "Anchorage", "value": 75 },  
        { "name": "Juneau", "value": 42 },  
        { "name": "Fairbanks", "value": 78 },  
        { "name": "Wasilla", "value": 12 }  
      ]  
    }  
  }  
]
```

Sources

- <https://www.freefavicon.com/freefavicons/icons/iconinfo/simple-water-icon-3-152-231714.html>
- <https://javascript.plainenglish.io/animate-when-element-is-in-view-with-framer-motion-63b254403bf>
- <https://www.framer.com/docs/animation/>
- <https://www.creative-tim.com/product/notus-nextjs>
- <https://codepen.io/VincentGarreau/pen/pnlso>
- <https://www.chartjs.org/>
- <https://react-chartjs-2.netlify.app/>