

Lundi 7 Novembre 2022

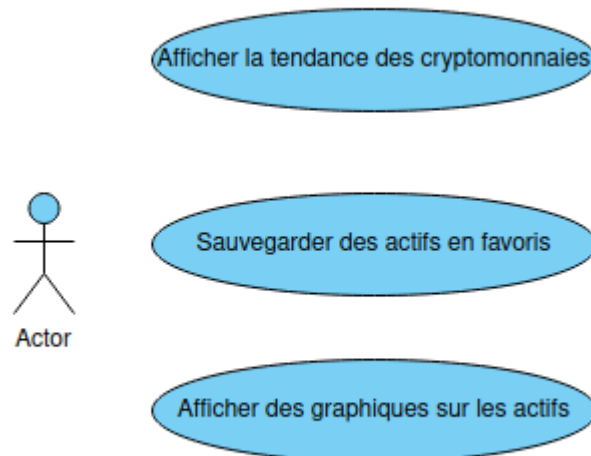
BLOC 3

Le développement d'une solution digitale

A. Presentation des fonctionnalités	2
I. Diagramme Use Case	2
II. User Storie	2
III. Arborescence fonctionnelle	3
B. Explications sur les fonctionnalités principales	4
I. Quelques exemples des composants :	4
II. Organisation des tests :	4
III. Organisation de travail et technique	6
VI. Point d'amélioration	7

A. Presentation des fonctionnalités

I. Diagramme Use Case



Pour le moment, le site crypto-tracker permet à l'utilisateur de visualiser les actifs de cryptomonnaie sur une liste, un graphique et de sauvegarder les actifs les plus intéressants pour lui. Grâce à cela il pourra se donner une idée des bénéfices grâce à l'historique de l'actif disponible sur le site. Pour le projet nous décidons de retenir l'API coin gecko proposant des data plus facile à mettre en place sur notre site, elle nous facilitera le travail pour la récupération d'information et la réalisation de chaque graphique pour les cryptomonnaies.

II. User Storie



Intitulé de poste
Chef de projet IT

Âge
Entre 35 et 44 ans

Réseaux sociaux

Secteur d'activité
Sport

Hugo Salies

Moyen de communication préféré
Ordinateur + portable

Objectifs
Realiser une veille sur les cryptos monnaies

Sources d'information
Internet
Journal
Radio

Principaux défis
Obtenir de bonne informations fiable a jour et facile a comprendre

Prenons l'exemple d'un utilisateur lambda pour notre site :

- En tant que user, Hugo souhaite obtenir la tendance des différentes cryptomonnaies afin de pouvoir acheter au meilleur prix.
- Souhaite obtenir un affichage dynamique, rapide et compréhensible.
- Il aimerait également acheter, vendre, échanger pour réaliser des bénéfices.

III. Arborescence fonctionnelle



Sur la page d'accueil nous afficherons une liste avec toutes les différentes cryptomonnaies, l'utilisateur aura la possibilité de choisir le nombre de crypto-monnaies qu'il souhaite visualiser et d'ajouter cette cryptomonnaie dans ses favoris.

Pour afficher le graphique il suffira de diriger la souris sur le graphique présent à côté du nom de la crypto-monnaie.

Sur la page Exchange l'utilisateur pourra vendre, acheter et échanger ces cryptomonnaies.

Sur la page Graphique rentabilité, compte rendu de l'activité de financement de l'utilisateur.

B. Explications sur les fonctionnalités principales

I. Quelques exemples des composants :

- Header Infos
- Percent Change
- TableLine

II. Organisation des tests :

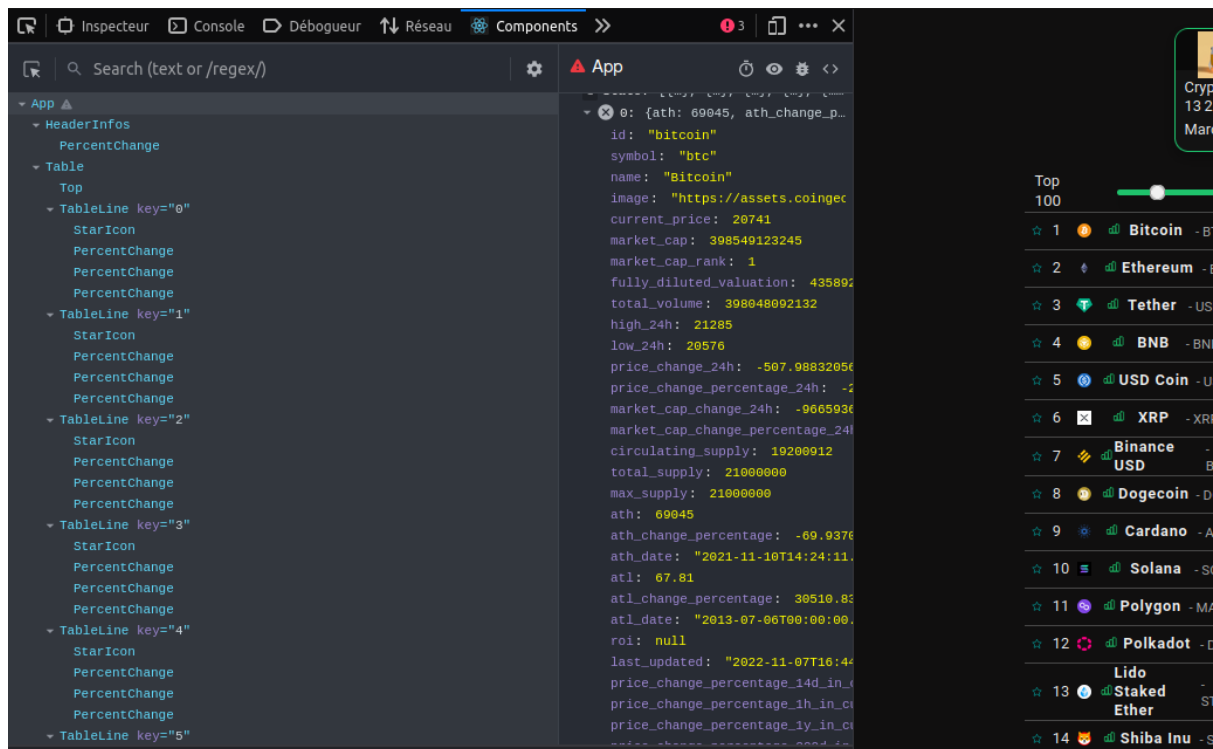
Tout d'abord, je décide de m'intéresser à la technologie Javascript et de voir les différentes possibilités pour la sauvegarde de la data dans une base de données. Ayant une grosse sensibilité pour React et voulant vraiment améliorer mes compétences dans cette technologie, je décide de démarrer mon projet sur React afin de maîtriser l'environnement de données. Par ailleurs, je décide de récupérer les données sur l'API sur coingecko qui propose un environnement plus fiable et maintenable que sur coin market cap. Proposant de travailler directement sur la partie du client et non du côté serveur, cela permettra d'accélérer la puissance de calcul et la réactivité du site.

Un exemple d'appel de l'API coingecko :

useEffect permet d'effectuer des effets : cela permet à notre composant d'exécuter des actions après l'affichage, en choisissant à quel moment cette action doit être exécutée. Le hook useEffect est appelé après chaque rendu de notre composant. Il est possible de préciser quelle modification de donnée déclenche les effets exécutés dans useEffect, avec le tableau de dépendances. Un tableau de dépendances vide permet d'exécuter un effet uniquement au premier rendu de notre composant.

```
useEffect(() => {  
  axios  
    .get(  
      `https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&`  
    )  
    .then((res) => setCoinsData(res.data));  
}, []);
```

Nous pouvons visualiser dans un second temps les éléments grâce à l'extension React component disponible sur plusieurs navigateurs web.



- Header Infos

Composant d'en tête qui permet d'afficher les éléments en première ligne du site, on réalise par la suite une fonction PercentChange afin de catégoriser la couleur des chiffres sur le site ≥ 0 green else red.

```
// props.percent
const PercentChange = ({ percent }) => {
  const [color, SetColor] = useState();

  useEffect(() => {
    if (percent) {
      if (percent >= 0) {
        SetColor(colors.green1);
      } else {
        SetColor(colors.red1);
      }
    } else {
      SetColor(colors.whitel);
    }
  }, [percent]);
}
```

- TableLine

Ce composant nous permet de fractionner les nombres que nous récupérons via un map sur l'élément parent Table. À partir d'une liste de données, map() permet de créer une liste de composants React. La prop key est indispensable dans les listes de composants. Elle doit être unique au sein de la liste, perdurer dans le temps.

```
    .map((coin, index) => (
      <TableLine coin={coin} index={index} key={index} />
    ))
  </div>
);
};

export default Table;
```

Filtre des nombres au format max et min afin d'améliorer la visualisation:

```
// Price crypto format : if lenght number < 4 function
const priceFormater = (num) => {
  if (Math.round(num).toString().length < 3) {
    return new Intl.NumberFormat("us-Us", {
      minimumFractionDigits: 3,
      maximumFractionDigits: 3,
    }).format(num);
  } else {
    return num;
  }
};
```

Filtre des nombres supérieurs à 6 caractères pour également une meilleur compréhension :

```
// MarketCap format price to Million
const mktCapFormater = (num) => {
  let newNum = String(num).split("").slice(0, -6);
  return Number(newNum.join(""));
};
```

III. Organisation de travail et technique

Tout d'abord, je me suis intéressé à l'API coingecko après avoir lu en long en large la documentation. Je me suis penché également à l'API recharts me permettant de créer un graphique pour me permettre de partager l'historique des cryptomonnaies.

Magalie a créé la maquette avec une charte graphique, au cours du développement nous avons changé un peu l'ergonomie du site tout en respectant les codes couleurs.

Nous nous sommes orientés sur la technologie React version 18. Bien entendu par manque de temps je n'ai pu créer une base de donnée pour ainsi sauvegarder les choix des utilisateurs, nous utilisons le localStorage donc si l'utilisateur rafraîchit la page, les informations sont perdues définitivement. Il a pas été simple de prendre cette décision, par manque de temps j'ai préféré avancer sur l'affichage des graphiques qui n'a pas été simple également, enfin pour la partie css on a utilisé sass pour rendre notre application responsive sur un écran de 480px.

VI. Point d'amélioration

Il manque l'achat de crypto, la vente, l'utilisation de l'API coingecko.

Il est vrai que celle-ci n'est pas recommandée dans les consignes mais elle m'a permis d'avancer dans le projet et de proposer une solution plus ergonomique car elle m'a permis de gagner du temps sur la base de données et donc de me pencher plus sur l'API recharts que propose npm.