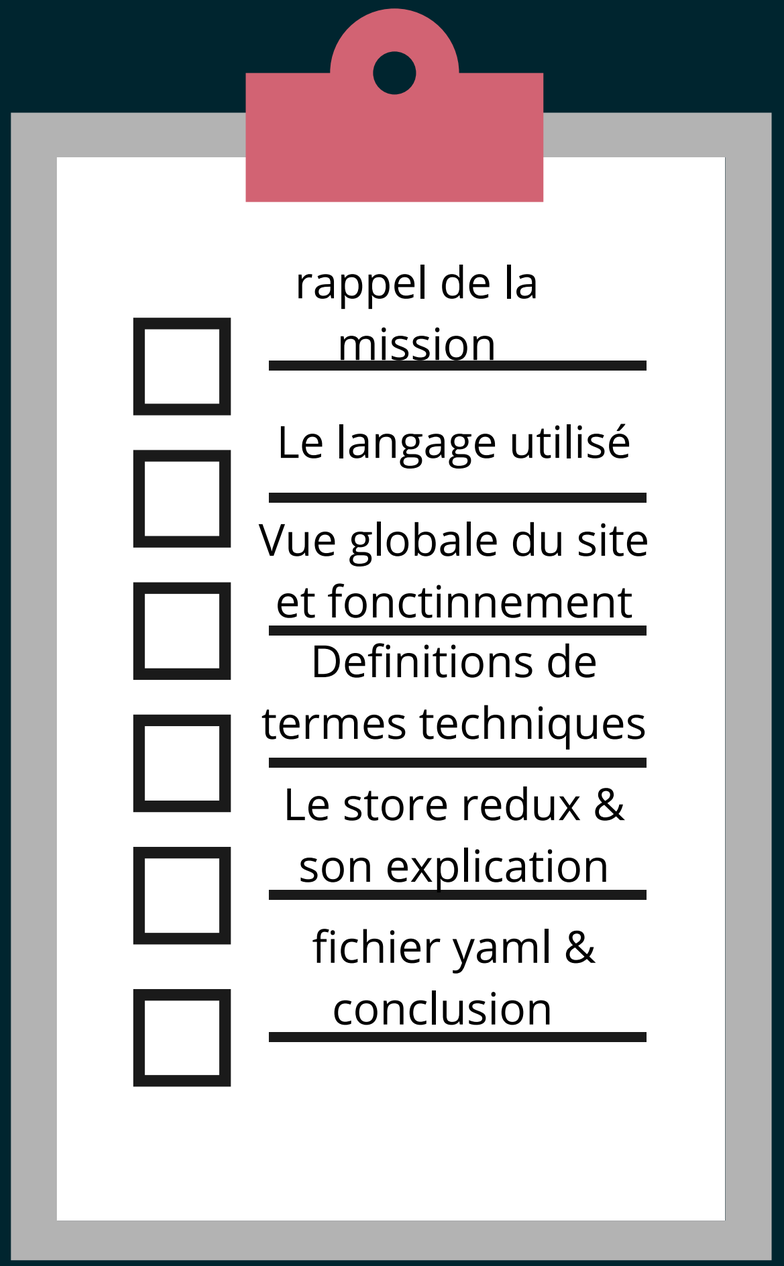
Implémentez le front-end d'une application bancaire avec React


rappel de la mission
Le langage utilisé
Vue globale du site et fonctinnement
Definitions de termes techniques
Le store redux & son explication
fichier yaml & conclusion

RAPPEL DE LA MISSION

-> implémentation du appli bancaire
-> creations des pages en react
-> gerer les authentifications
-> le store toolkit

Le langage utilisé

Redux Toolkit simplifie la configuration et la gestion de Redux en regroupant de nombreuses fonctionnalités couramment utilisées, ce qui rend le développement d'applications web plus efficace et moins sujet aux erreurs.

Reducers

les réducteurs sont les travailleurs de Redux qui prennent des actions comme instructions pour mettre à jour l'état de votre application de manière organisée et prévisible.

store

le store dans Redux Toolkit est le gardien central des données de votre application, rendant la gestion de l'état plus simple et plus prévisible.

payload

données d'une action qui transporte les informations spécifiques nécessaires pour effectuer une tâche ou une modification particulière dans votre application.

dispatch :

moyen d'initier des changements d'état dans votre application en envoyant des actions au magasin Redux, ce qui déclenche les réducteurs pour effectuer les mises à jour nécessaires. C'est un élément clé pour interagir avec l'état géré par Redux.

LE STORE

```
import { configureStore } from "@reduxjs/toolkit";
import userReducer from './reducers/userSlice';

const mainStore = configureStore({
    reducer: {
        user: userReducer,
    }
});

export default mainStore;
```

User slice

```
import { createSlice } from "@reduxjs/toolkit";

// On initialise d'abord l'état de base
const initialState = {
    userName: "null",
    firstName: "null",
    lastName: "null",
    token: null
}

const userSlice = createSlice({
    name: 'user',
    initialState,
    reducers: {
        setUserName: (state, { payload }) => {
            state.userName = payload;
        },
        setFirstName: (state, { payload }) => {
            state.firstName = payload;
        },
        setLastName: (state, { payload }) => {
            state.lastName = payload;
        },
        setToken: (state, { payload }) => {
            state.token = payload;
        },
    }
});

export const { setUserName, setFirstName, setLastName,
export default userSlice.reducer
```

intitialisation des états

on definit les reducers pour gerer et mettre les états a jour

À l'intérieur de la tranche, nous définissons des réducteurs pour gérer les mises à jour de l'état. Chaque réducteur, comme setUserName, prend deux arguments : l'état actuel (ici, state) et une action contenant des données (ici, { payload }). Le réducteur met à jour la partie de l'état appropriée (par exemple, state.userName) avec la valeur passée dans le payload de l'action.

Selector

```
1   export  const selectUserName = (state) => state.user.userName
2   export  const selectFirstName = (state) => state.user.firstName
3   export const selectLastName = (state) => state.user.lastName
4   export  const selectToken= (state) => state.user.token
5
```

MERCI DE M’AVOIR CONFIE CETTE MISSION