PREDICTION PRIX CRYPTO

YNOV RENNES CAMPUS

Maxime Lombard González
Yoann

I. UI / UX

- Landing Page: Présentation du site, fonctionnement et “get started”
- Sign Up / Login
- Main Page: Tableau présentant, le logo de la devise, le nom du crypto, son volume et son évolution où on pourra voir un graphique qui montre le réel et la prédiction.
- Paramètres: On aura le temps de prédire et le pourcentage de prédiction et / où intervalle de confiance.
- Shop: On donnera à l’utilisateur la possibilité d’acheter un bot de financement automatique qui pourra l’assister pour pouvoir acheter ou vendre.

II. FONCTIONNEMENT

- Mettre en place de bot de financement automatique :
    - Trois personnalités :
        - Le calme : couleur bleu, visage mignon. Ira pour un résultat “safe”.
        - Le téméraire : couleur rouge, visage energétique type Colère dans vice versa. Ira pour un résultat “high risk high reward”.
        - Le guide : couleur verte, agit comme une newsletter sur tout l’univers crypto et sur les meilleurs résultats de prédiction.

III. COMPOSANTS ET MÉTHODES ESSENTIELLES

- LoginComponent
  `initLoginForm()`: Dès le chargement de la page on initialisera la fonction qui créera un nouveau formulaire avec qui demandera deux validateurs: un mail et un mot de passe.
  `setLoginErrorDebounce()`: Fonction qui en cas d'erreur enverra un erreur au backend et aura un debounce time d’environ 5 secondes.
  `handlePasswordError()`: Fonction qui vérifiera que la chaîne de caractères ne soit pas vide et qui essaiera de trouver un match avec le mot de passe actuel. Sinon, cela retournera un message d’erreur.
  `handleEmailError()`: Vérifiera que le mail ait un bon format et fera un match avec un mail dans la base de données.

`handleLoginError()`: Vérification d'erreurs générales.
`handleLoginSuccess()`: Créera une connexion en cas d'absence d'erreurs.

- SignInComponent
  `initRegisterForm()`: On initialisera la fonction qui créera un nouveau formulaire pour l'enregistrement d'un nouveau mail.
  `emailValidator()`: Validateur pour un mail.
  `passwordValidator()`: Validateur pour un mot de passe suivant un RegEx pattern.
  `passwordConfirming()`: Validateur par comparaison du mot de passe inséré avant avec l'actuel.
  `loadUserData()`: Envoie les informations enregistrées.

- Shop | PaymentServices
  `requestPayment()`: En prenant un token comme paramètre, il utilise un service tiers pour effectuer un paiement.
  `checkPromoCode()`: En prenant un token comme paramètre, il utilise un service tiers pour effectuer un paiement il match le code promo inséré avec un code pas utilisé dans la base de données.

- ShopComponent
  `requestPayment()`: En utilisant le token par le service du composant on fera des requêtes au backend par des api's lors pour confirmer un échec, erreur ou succès.
  `updatePrice()`: Permet de mettre à jour le prix du chariot dépendant du pack selectionné.

- SellTradeComponent
  `requestTrade()`: En utilisant le token par le service du composant on fera des requêtes au backend par des api's lors pour confirmer un échec, erreur ou succès.
  `confirmTrade()`: Permet d'envoyer une confirmation pour un échange de devis.

IV. DIAGRAMME FONCTIONNEL POUR LE BOT