

Tme Solo

Introduction

Pour ce tme solo nous avons besoin d'implémenter une fonction Fake News.

Classe modifier

- Serveur : nous avons modifier les classes `apimessage` et `message` dans entities
 - dans `message` nous avons ajouté un nouvelle fonction `isFakenews` de dans. cette fonction permet à l'appel d'incrémenter le nombre de personne qui a dit que le message est un Fake news, et met le username de l'utilisateur dans une liste `votedUser`. Pour faire cela nous avons également modifié la fonction `create` pour mettre les ajouters les 2 nouvelles attribut

```

async create(message, login, images) {
  return new Promise((resolve, reject) => {
    this.db.messages.insert({ message: message, login: login, date:
Date.now(), comments: [], images: images , fakenummer : 0, votedUser : []},
function (err, docs) {
    let messageid = docs._id
    if (!messageid) {
      reject();
    } else {
      resolve(messageid);
    }
  })
});
}

async isFakeNews(isFakeNews , userid) {
  return new Promise((resolve, reject) => {
    console.log("isFakeNews",isFakeNews)
    let mess = this.getMessage(isFakeNews)
    let fakenummer = mess.isFakeNews + 1;
    let votedUser = mess.votedUser;
    votedUser.push(userid)

    console.log(fakenummer, votedUser)

    this.db.messages.update({ _id: isFakeNews }, { $set: {
fakenummer: fakenummer , votedUser : votedUser} }, function (err, docs) {
      resolve()
    });
  });
}

```

- Dans `apimessage` nous avons ajouté une nouvelle api de path `isFakeNews` qui permet de récupérer l'id du message qu'on veut modifier et d'appeler la fonction ci-dessus

```
router.post("/isFakeNews", async (req, res) => {
  const { isFakeNews } = req.body;
  console.log("id : " , isFakeNews)
  if (!isFakeNews) {
    res.status(401).send("Missing fields");
  } else {
    try {
      console.log("=====: " , isFakeNews)
      await messages.isFakeNews(isFakeNews ,
req.session.username);
      res.status(201).send("isFakeNews added");
    } catch (e) {
      res.status(500).send(e);
    }
  }
});
```

- Client : dans le côté client nous avons seulement modifié le composant `Message`. On a ajouté un bouton lorsque ce bouton est cliqué une nouvelle fonction qu'on a implémenté, cette fonction fait une requête HTTP POST vers le serveur passant par API implémenté ci-dessus en utilisant axios. Pour gérer cela dans le state nous avons ajouté des nouvelles état `nbFakeNews: 0`, `votedUser: []`, le bouton s'affiche si et seulement si le username de user n'est pas dans la liste, et si le `nbFakeNews` >10 nous allons supprimer ce message pour qu'elle ne soit plus afficher

```
addFakeNews() {
  console.log("addFakeNews : ", this.state.id);
  apimessages.post('/isFakenews', {isFakenwes :
this.state.id}).then(res => {
    console.log("addFakeNews : ", res.data);
    this.state.votedUser.push(sessionStorage.getItem('userinfo'));
    this.setState({ nbFakeNews: this.state.nbFakeNews + 1 })
    window.location.reload();
  })
}

// code button render
{sessionStorage.getItem('userinfo') in this.state.votedUser ? <p></p>: <div
id="fakenewsButton">
  <button id="fakenewsButton" onClick={() => {
this.addFakeNews()}}>Fake News</button>
  </div>}
{this.state.nbFakeNews > 10 ? this.deleteMessage: <p></p>}
```

