



2021-2022

Rapport final de projet WEB

HADDADI Mazigh

OTMANE CHERIF Mohammed Islem



Comme convenu dans la partie 1 et 2, Notre projet consiste de créer une application Quiz qui permet aux utilisateurs de tester leurs connaissances dans des domaines différents (Mathématiques, Technologie, Géographie, Science, Sport et musique), 5 questions par domaine et 15 secondes pour répondre à chaque question.

Côté Client :

Après la connexion à son compte, Une page contenant deux boutons s'affiche, le premier bouton permet de commencer le jeu, le deuxième sert à se déconnecter et sortir du quiz (retourner vers la page du login). Quand l'utilisateur clique sur le premier bouton "Start Quiz", une page contenant plusieurs domaines s'affiche. Ces domaines représentent les différents thèmes du quiz (Tech, Sports, Science, Music, Maths, Geography). Chaque thème, qui est un bouton, mène aux règles du jeu suivi d'un quiz contenant une série de 5 questions à choix multiples portant sur le domaine en question. Quand le quiz commence, le joueur aura 15 secondes pour répondre à la question. Si le temps est écoulé, le joueur ne pourra plus répondre ; il devra donc passer à la prochaine question. Quand le joueur choisit sa réponse, il ne peut plus modifier son choix ; il connaîtra directement si sa réponse est correcte ou pas. (Si elle est correcte, elle prend la couleur verte avec une icône check, sinon elle prend la couleur rouge avec une icône cross et la bonne réponse prend la couleur verte). Une fois le choix effectué, un bouton "Next" apparaît en bas du quiz permettant de passer à la question suivante. Si le joueur arrive au bout de la cinquième question, le bouton "Next" va afficher le résultat du quiz, par exemple : Perfect ! You got 5 out of 5.

Ou encore : You got 1 out of 5. Better Luck next time !

Deux autres boutons s'affichent, l'un permet de refaire une nouvelle partie en nous redirigeant vers la page des choix, l'autre permet de se déconnecter.

Pour implémenter tout cela, du code HTML, CSS et JavaScript sont utilisés.

Les questions sont stockées sous formes d'objets JavaScript comme illustré ci-dessous :

```
// Sports questions
let questions = [
  {
    number: 1,
    question: "As of April 2022, which male tennis player has won the most Grand Sla",
    answer: "Rafael Nadal",
    options: [
      "Novak Djokovic",
      "Rafael Nadal",
      "Roger Federer",
      "Pete Sampras"
    ]
  },
  {
    number: 2,
    question: "Which footballer has scored the most goals in World Cup history?",
    answer: "Miroslav Klose",
    options: [
      "Ronaldo Nazário",
      "Pelé",
      "Miroslav Klose",
      "Gerd Müller"
    ]
  },
  {
    number: 3,
    question: "In American Football, how many points do you score for a touchdown?"
  }
]
```

Voici quelques captures d'écran du coté client de l'application :

Login :



Log In

Sign Up

Login

Une fois connecté :

Start Quiz

Log Out

Start Quiz :

Choose one theme to start the quiz 🧙

Music 🎵

Sports ⚽

Tech 💻

Geography 🌍

Maths 📐

Science 🔬

On choisit un thème :

Here are the rules

1. You will have **15 seconds** per each question.
2. Once you select your answer, you can not change it.
3. You can not select any option once time goes off.
4. You can not exit the game while you're playing.
5. The number of points depends on how many questions you answer correctly.

Log Out

Continue

On clique sur continue :

quizApp

Time Left

11

1. Which former member of One Direction left the band first?

Harry Styles

Zayn Malik

Niall Horan

Liam Payne

Question 1 of 5

2. Which of the following was the lead singer for Linkin Park?

Elvis Presley

Freddie Mercury



Prince

Chester Bennington

Question 2 of 5

Next

quizApp

Time Off

0

3. Which famous rapper starred in the movie 8 Mile?

50 Cent

Snoop Dogg

Eminem

Biggie

Question 3 of 5

Next



You have completed the Quiz!

Good job! You got 3 out of 5

Replay

Log Out

Côté Serveur :

Tout d'abord pour la base de données, on a utilisé une base de données SQLite3 (comme vue en TP8) avec une table "user" contenant quatre champs (id auto-incrémente, email, username, password)

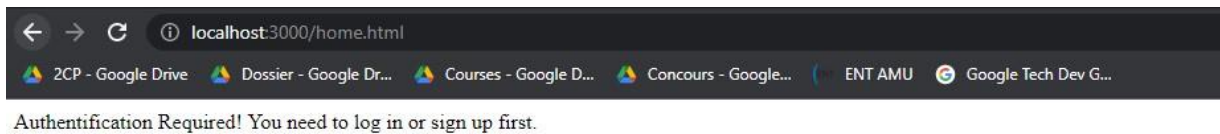
On a utilisé un serveur Node.js (localhost) comme vue en TP8.

Ensuite pour la gestion des sessions des utilisateurs, chaque utilisateur doit se connecter s'il possède déjà un compte ou créer un compte sinon pour pouvoir jouer. La création du compte nécessite une adresse mail, un nom d'utilisateur et un mot de passe (il ne peut pas y'avoir deux compte avec la même adresse mail et le même nom d'utilisateur). Une fois l'utilisateur connecté, il peut commencer à jouer.

L'utilisateur ne pourra pas jouer sauf s'il se connecte ou il crée un compte. Si on essaie par exemple de saisir l'url suivante sans se connecter :

(<http://localhost:3000/home.html>)

On aura le résultat suivant :



Donc on peut se connecter ou créer un compte grâce à des fonctions login et signup et se déconnecter grâce à la fonction logout.

Changements par rapport à la conception:

Il n'y a pas eu de changements par rapport à la partie 2. On a réussi à implémenter tout ce qui était prévu.

Les difficultés rencontrées :

Au début du projet on a partagé le travail entre nous et chacun de nous a commencé à implémenter les pages nécessaires (HTML -CSS - JS). Ensuite, quand on a voulu intégrer la partie serveur à celle du Client, les pages s'affichent sans CSS et JS (affichage HTML seulement sans intégrer les fichiers externes de CSS et JS). Après maintes recherches, nous étions obligés d'intégrer le code CSS et le code JavaScript qui contrôle le jeu dans le fichier HTML. Donc, en gros on a des fichiers HTML qui contiennent du code CSS

dans une balise `<style></style>` et du code JS dans une balise `<script></script>`.

Originalité et Effort :

Notre projet ne ressemble pas trop au TP8 (Mis à part la gestion des utilisateurs et le coté serveur) car on a dû implémenter plusieurs trucs de zéro, à savoir le coté Client. On a codé plusieurs fonctionnalités, notamment :

- Le compte à rebours qui doit s'arrêter quand l'utilisateur choisit sa réponse.
- Passer d'une question à une autre.
- Informé le joueur du résultat de son choix dès qu'il ait choisi sa réponse.
- Mettre à jour le nombre de bonnes réponses et afficher le score final.
- Cacher des pages et des boutons et en afficher d'autres en fonction de l'état de la partie.
- Alerter l'utilisateur si ce dernier essaie de créer un compte déjà existant.

Instruction pour lancer le projet :

Il suffit d'installer les modules (express, mustache-express, better-sqlite3, alert, cookie-session) avec la commande `npm install ...`

Par exemple : `npm install better-sqlite3`

`npm install cookie-session`

`npm install alert`

Après, exécutez la commande : `node create-db.js`

Et finalement lancez le serveur (`node server.js`)

L'application sera (Si tout s'est bien installé) à l'URL :

`http://localhost:3000`