

Muvatsi KYAHIMBA MESSENGER
Niveau: 2ème année GENIE INFO
DEVOIR SURVEILLE 1
PROGRAMMATION MOBILE
31/03/2024

Création d'un site Web de Quiz interactif avec utilisation d'un fichier JSON
QUIZ-RDC

I.INTRODUCTION

Dans les pages suivantes, nous vous invitons à découvrir les coulisses du développement du site web Quiz-RDC. Axé sur la République Démocratique du Congo (RDC), ce site offre une expérience interactive unique permettant aux utilisateurs de tester leurs connaissances sur ce magnifique pays d'Afrique centrale.

Plongez dans le processus de création de Quiz-RDC, où chaque ligne de code et chaque choix de conception ont été minutieusement élaborés pour offrir une expérience ludique et éducative. À travers des extraits de code, des captures d'écran et des explications détaillées, vous découvrirez les tenants et aboutissants du développement de ce site web captivant.

Que vous soyez un passionné de la RDC, un amateur de quiz ou un développeur web en herbe, ce document vous offre un aperçu privilégié du processus de création d'un site web interactif. Préparez-vous à plonger dans l'univers coloré et captivant de Quiz-RDC, où la découverte et le divertissement sont au rendez-vous.

II.PROGRAMMATION

II.1. HTML

Ce code HTML établit la structure de base de notre page web interactive pour un quiz sur la République Démocratique du Congo. Il comprend des éléments pour afficher les règles du quiz, les questions, les options de réponse, un chronomètre et les résultats du quiz.

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="MySkyPower / CREATIVE PRODUCTION GROUP
We are the producers of your ideas. The workshop - the craftsmen - the
designers - the engineers. Together we are one. One group of talented people
who harness the power to build big ideas. From handmade to high- tech. With
you we create. Bring us your ideas, designs, and dreams and we will build
them. Whatever the scale. EVERY DAY WE BUILD BIG IDEAS We are MySkyPower." />
```

```

    <meta property="og:title" content="My Sky Power | MSP" />
    <meta property="og:description" content="MySkyPower / CREATIVE PRODUCTION
GROUP We are the producers of your ideas. The workshop – the craftsmen – the
designers – the engineers. Together we are one. One group of talented people
who harness the power to build big ideas. From handmade to high- tech. With
you we create. Bring us your ideas, designs, and dreams and we will build
them. Whatever the scale. EVERY DAY WE BUILD BIG IDEAS We are MySkyPower." />
    <meta property="og:type" content="website" />
    <meta property="og:url" content="https://www.myskypower.net/" />

    <title>Quiz-RDC</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css"/>
</head>

<body>

    <main class="main">
        <div class="start_btn">
            <h1>Quiz-RDC</h1>
            <p>Bienvenue sur votre site de quiz dédié à la République
Démocratique du Congo !
                Préparez-vous à découvrir et à explorer la richesse culturelle,
historique
                et géographique de ce magnifique pays à travers ce jeu et des
questions captivantes.
            <br>Bonne exploration ! 🌍🌟</p>
            <button>Commencer</button>
        </div>
    </main>

    <div class="info_box">
        <div class="info-title"><span>Règles du Quiz</span></div>
        <div class="info-list">
            <div class="info">1. Vous ne disposerez que de <span>10sec</span>
pour chaque question.</div>
            <div class="info">2. Une fois que vous avez sélectionné votre
réponse, elle ne peut plus être annulée.</div>
            <div class="info">3. Vous ne pouvez sélectionner aucune option une
fois le temps écoulé.</div>
            <div class="info">4. Vous ne pouvez pas quitter le Quiz pendant
que vous jouez.</div>
            <div class="info">5. Vous obtiendrez des points sur la base de vos
réponses correctes.</div>
        </div>
        <div class="buttons">
            <button class="quit">Quitter</button>
            <button class="restart">Continue</button>
        </div>
    </div>

```

```

        </div>
    </div>

    <div class="quiz_box">
        <header>
            <div class="title">Quiz-R.D.Congo</div>
            <div class="timer">
                <div class="time_left_txt">chrono</div>
                <div class="timer_sec">10</div>
            </div>
            <div class="time_line"></div>
        </header>
        <section>
            <div class="que_text">

                </div>
            <div class="option_list">

                </div>
        </section>

        <footer>
            <div class="total_que">

                </div>
            <button class="next_btn">Suivant</button>
        </footer>
    </div>

    <div class="result_box">
        <div class="icon">
            <i class="fas fa-crown"></i>
        </div>
        <div class="complete_text">Vous avez terminé le Quiz !</div>
        <div class="score_text">

        </div>
        <div class="buttons">
            <button class="restart">Rejouer</button>
            <button class="quit">Quitter</button>
        </div>
    </div>

    <script src="script.js"></script>

</body>
</html>

```

Explication détaillée des parties principales du code

<head>: Cette balise contient les métadonnées et les liens vers les fichiers externes tels que les feuilles de style CSS et les scripts JavaScript.

Métadonnées: Définissent des informations sur le document HTML, telles que l'encodage, la description et les données Open Graph pour le partage sur les réseaux sociaux.

charset="UTF-8": Définit l'encodage des caractères comme UTF-8.

viewport: Définit comment la page doit être affichée sur différents appareils.

title: Définit le titre de la page qui apparaîtra dans l'onglet du navigateur.

Liens vers les fichiers CSS externes pour la mise en forme de la page et les icônes de police.

<body>: Contient le contenu principal de la page web.

main: Balise contenant le contenu principal de la page, avec une classe "main" utilisée pour le style CSS.

start_btn: Contient le bouton de démarrage du quiz, avec un titre et un paragraphe de bienvenue.

info_box: Contient les règles du quiz, avec des informations détaillées sur la manière de jouer.

quiz_box: Contient les éléments du quiz, y compris le titre, le chronomètre, la question, les options et les boutons de navigation.

result_box: Contient le résultat du quiz, avec un message personnalisé en fonction du score de l'utilisateur.

<script>: Lie les fichiers JavaScript externes pour le fonctionnement interactif du quiz.

II.2. CSS

Ce code CSS définit les styles de base pour la mise en page et l'apparence visuelle des éléments de la page web du quiz sur la République Démocratique du Congo. Il crée une mise en page centrée et utilise des couleurs, des polices et des ombres pour rendre l'interface utilisateur attrayante et conviviale.

Nous avons utilisé les règles **@media** qui permettent d'appliquer des styles spécifiques en fonction de la taille de l'écran, assurant ainsi une expérience utilisateur adaptée aux différents appareils, des ordinateurs de bureau aux smartphones.

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
```

```

    background:#2280e4;
}

::selection{
    color: #fff;
    background: #007bff;
}

.start_btn,
.info_box,
.quiz_box,
.result_box{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2),
                0 6px 20px 0 rgba(0, 0, 0, 0.19);
}

.start_btn {
    max-width: 600px;
    display: flex;
    align-items: center;
    flex-direction: column;
    padding: 15px 30px;
}

.main.active {
    filter:blur(15px);
}

.start_btn h1 {
    font-size: 78px;
    font-weight: 700;
    text-shadow:0 0 10px #c3e6cb;
}

.start_btn p{
    font-size: 16px;
    text-align: center;
    text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
    margin-bottom: 30px;
}

.info_box.activeInfo,
.quiz_box.activeQuiz,
.result_box.activeResult{
    opacity: 1;
}

```

```

    z-index: 5;
    pointer-events: auto;
    transform: translate(-50%, -50%) scale(1);
}

.start_btn button{
    font-size: 25px;
    font-weight: 500;
    color: #007bff;
    padding: 10px 20px;
    outline: none;
    border: none;
    border-radius: 5px;
    background: #fff;
    cursor: pointer;
}

.info_box{
    width: 540px;
    background: #fff;
    border-radius: 5px;
    transform: translate(-50%, -50%) scale(0.9);
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}

.info_box .info-title{
    height: 60px;
    width: 100%;
    border-bottom: 1px solid lightgrey;
    display: flex;
    align-items: center;
    padding: 0 30px;
    border-radius: 5px 5px 0 0;
    font-size: 20px;
    font-weight: 600;
}

.info_box .info-list{
    padding: 15px 30px;
}

.info_box .info-list .info{
    margin: 5px 0;
    font-size: 17px;
}

.info_box .info-list .info span{

```

```

    font-weight: 600;
    color: #007bff;
}
.info_box .buttons{
    height: 60px;
    display: flex;
    align-items: center;
    justify-content: flex-end;
    padding: 0 30px;
    border-top: 1px solid lightgrey;
}

.info_box .buttons button{
    margin: 0 5px;
    height: 40px;
    width: 100px;
    font-size: 16px;
    font-weight: 500;
    cursor: pointer;
    border: none;
    outline: none;
    border-radius: 5px;
    border: 1px solid #007bff;
    transition: all 0.3s ease;
}

.quiz_box{
    width: 550px;
    background: #fff;
    border-radius: 5px;
    transform: translate(-50%, -50%) scale(0.9);
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}

.quiz_box header{
    position: relative;
    z-index: 2;
    height: 70px;
    padding: 0 30px;
    background: #fff;
    border-radius: 5px 5px 0 0;
    display: flex;
    align-items: center;
    justify-content: space-between;
    box-shadow: 0px 3px 5px 1px rgba(0,0,0,0.1);
}

```

```

.quiz_box header .title{
    font-size: 20px;
    font-weight: 600;
}

.quiz_box header .timer{
    color: #004085;
    background: #cce5ff;
    border: 1px solid #b8daff;
    height: 45px;
    padding: 0 8px;
    border-radius: 5px;
    display: flex;
    align-items: center;
    justify-content: space-between;
    width: 145px;
}

.quiz_box header .timer .time_left_txt{
    font-weight: 400;
    font-size: 17px;
    user-select: none;
}

.quiz_box header .timer .timer_sec{
    font-size: 18px;
    font-weight: 500;
    height: 30px;
    width: 45px;
    color: #fff;
    border-radius: 5px;
    line-height: 30px;
    text-align: center;
    background: #343a40;
    border: 1px solid #343a40;
    user-select: none;
}

.quiz_box header .time_line{
    position: absolute;
    bottom: 0px;
    left: 0px;
    height: 3px;
    background: #007bff;
}

section{
    padding: 25px 30px 20px 30px;
    background: #fff;
}

```



```

}

section .que_text{
    font-size: 25px;
    font-weight: 600;
}

section .option_list{
    padding: 20px 0px;
    display: block;
}

section .option_list .option{
    background: aliceblue;
    border: 1px solid #84c5fe;
    border-radius: 5px;
    padding: 8px 15px;
    font-size: 17px;
    margin-bottom: 15px;
    cursor: pointer;
    transition: all 0.3s ease;
    display: flex;
    align-items: center;
    justify-content: space-between;
}

section .option_list .option:last-child{
    margin-bottom: 0px;
}

section .option_list .option:hover{
    color: #004085;
    background: #cce5ff;
    border: 1px solid #b8daff;
}

section .option_list .option.correct{
    color: #155724;
    background: #d4edda;
    border: 1px solid #c3e6cb;
}

section .option_list .option.incorrect{
    color: #721c24;
    background: #f8d7da;
    border: 1px solid #f5c6cb;
}

section .option_list .option.disabled{

```

```

    pointer-events: none;
}

section .option_list .option .icon{
    height: 26px;
    width: 26px;
    border: 2px solid transparent;
    border-radius: 50%;
    text-align: center;
    font-size: 13px;
    pointer-events: none;
    transition: all 0.3s ease;
    line-height: 24px;
}

.option_list .option .icon.tick{
    color: #23903c;
    border-color: #23903c;
    background: #d4edda;
}

.option_list .option .icon.cross{
    color: #a42834;
    background: #f8d7da;
    border-color: #a42834;
}

footer{
    height: 60px;
    padding: 0 30px;
    display: flex;
    align-items: center;
    justify-content: space-between;
    border-top: 1px solid lightgrey;
}

footer .total_que span{
    display: flex;
    user-select: none;
}

footer .total_que span p{
    font-weight: 500;
    padding: 0 5px;
}

footer .total_que span p:first-child{
    padding-left: 0px;
}

```

```

footer button{
    height: 40px;
    padding: 0 13px;
    font-size: 18px;
    font-weight: 400;
    cursor: pointer;
    border: none;
    outline: none;
    color: #fff;
    border-radius: 5px;
    background: #007bff;
    border: 1px solid #007bff;
    line-height: 10px;
    opacity: 0;
    pointer-events: none;
    transform: scale(0.95);
    transition: all 0.3s ease;
}

footer button:hover{
    background: #0263ca;
}

footer button.show{
    opacity: 1;
    pointer-events: auto;
    transform: scale(1);
}

.result_box{
    background: #fff;
    border-radius: 5px;
    display: flex;
    padding: 25px 30px;
    width: 450px;
    align-items: center;
    flex-direction: column;
    justify-content: center;
    transform: translate(-50%, -50%) scale(0.9);
    opacity: 0;
    pointer-events: none;
    transition: all 0.3s ease;
}

.result_box .icon{
    font-size: 100px;
    color: #007bff;
    margin-bottom: 10px;
}

```

```

.result_box .complete_text{
    font-size: 20px;
    font-weight: 500;
}

.result_box .score_text span{
    display: flex;
    margin: 10px 0;
    font-size: 18px;
    font-weight: 500;
}

.result_box .score_text span p{
    padding: 0 4px;
    font-weight: 600;
}

.result_box .buttons{
    display: flex;
    margin: 20px 0;
}

.result_box .buttons button{
    margin: 0 10px;
    height: 45px;
    padding: 0 20px;
    font-size: 18px;
    font-weight: 500;
    cursor: pointer;
    border: none;
    outline: none;
    border-radius: 5px;
    border: 1px solid #007bff;
    transition: all 0.3s ease;
}

.buttons button.restart{
    color: #fff;
    background: #007bff;
}

.buttons button.restart:hover{
    background: #0263ca;
}

.buttons button.quit{
    color: #007bff;
    background: #fff;
}

```

```

}

.buttons button.quit:hover{
    color: #fff;
    background: #007bff;
}

@media (max-width: 768px) {
    .start_btn,
    .info_box,
    .quiz_box,
    .result_box {
        width: 90%;
        max-width: none;
    }
}

@media (max-width: 576px) {
    .start_btn h1 {
        font-size: 48px;
    }
}

```

II.3. JAVASCRIPT

Ce code JavaScript est responsable de la logique interactive du quiz.

```

const start_btn = document.querySelector(".start_btn button");
const info_box = document.querySelector(".info_box");
const exit_btn = info_box.querySelector(".buttons .quit");
const continue_btn = info_box.querySelector(".buttons .restart");
const quiz_box = document.querySelector(".quiz_box");
const result_box = document.querySelector(".result_box");
const option_list = document.querySelector(".option_list");
const time_line = document.querySelector("header .time_line");
const timeText = document.querySelector(".timer .time_left_txt");
const timeCount = document.querySelector(".timer .timer_sec");
const main = document.querySelector(".main");

start_btn.onclick = ()=>{
    info_box.classList.add("activeInfo");
    main.classList.add("active");
}

exit_btn.onclick = ()=>{
    info_box.classList.remove("activeInfo");
    main.classList.remove("active");
}

```

```

}

continue_btn.onclick = ()=>{
    info_box.classList.remove("activeInfo");
    quiz_box.classList.add("activeQuiz");
    showQuestions(0);
    queCounter(1);
    startTimer(10);
    startTimerLine(0);
}

let timeValue = 10;
let que_count = 0;
let que_numb = 1;
let userScore = 0;
let counter;
let counterLine;
let widthValue = 0;

const restart_quiz = result_box.querySelector(".buttons .restart");
const quit_quiz = result_box.querySelector(".buttons .quit");

restart_quiz.onclick = ()=>{
    quiz_box.classList.add("activeQuiz");
    result_box.classList.remove("activeResult");
    timeValue = 10;
    que_count = 0;
    que_numb = 1;
    userScore = 0;
    widthValue = 0;
    showQuestions(que_count);
    queCounter(que_numb);
    clearInterval(counter);
    clearInterval(counterLine);
    startTimer(timeValue);
    startTimerLine(widthValue);
    timeText.textContent = "Chrono";
    next_btn.classList.remove("show");
}

quit_quiz.onclick = ()=>{
    window.location.reload();
}

const next_btn = document.querySelector("footer .next_btn");
const bottom_ques_counter = document.querySelector("footer .total_que");

next_btn.onclick = ()=>{

```

```

        if(que_count < questions.length - 1){
            que_count++;
            que_numb++;
            showQuetions(que_count);
            queCounter(que_numb);
            clearInterval(counter);
            clearInterval(counterLine);
            startTimer(timeValue);
            startTimerLine(widthValue);
            timeText.textContent = "Chrono";
            next_btn.classList.remove("show");
        }else{
            clearInterval(counter);
            clearInterval(counterLine);
            showResult();
        }
    }
}

fetch('questions.json')
    .then(response => response.json())
    .then(data => {
        questions = data;
    })
    .catch(error => {
        console.error('Une erreur s\'est produite lors du chargement des questions:', error);
    });

function showQuetions(index){
    const que_text = document.querySelector(".que_text");

    let que_tag = '<span>' + questions[index].numb + ". " +
questions[index].question + '</span>';
    let option_tag = '<div class="option"><span>' + questions[index].options[0]
+'</span></div>'
    + '<div class="option"><span>' + questions[index].options[1]
+'</span></div>'
    + '<div class="option"><span>' + questions[index].options[2]
+'</span></div>'
    + '<div class="option"><span>' + questions[index].options[3]
+'</span></div>';
    que_text.innerHTML = que_tag;
    option_list.innerHTML = option_tag;

    const option = option_list.querySelectorAll(".option");

    for(i=0; i < option.length; i++){
        option[i].setAttribute("onclick", "optionSelected(this)");
    }
}

```

```

    }
}

let tickIconTag = '<div class="icon tick"><i class="fas fa-check"></i></div>';
let crossIconTag = '<div class="icon cross"><i class="fas fa-times"></i></div>';

function optionSelected(answer){
    clearInterval(counter);
    clearInterval(counterLine);
    let userAns = answer.textContent;
    let correcAns = questions[que_count].answer;
    const allOptions = option_list.children.length;

    if(userAns == correcAns){
        userScore += 1;
        answer.classList.add("correct");
        answer.insertAdjacentHTML("beforeend", tickIconTag);
        console.log("Correct Answer");
        console.log("Your correct answers = " + userScore);
    }else{
        answer.classList.add("incorrect");
        answer.insertAdjacentHTML("beforeend", crossIconTag);
        console.log("Wrong Answer");

        for(i=0; i < allOptions; i++){
            if(option_list.children[i].textContent == correcAns){
                option_list.children[i].setAttribute("class", "option
correct");
                option_list.children[i].insertAdjacentHTML("beforeend",
tickIconTag);
                console.log("Auto selected correct answer.");
            }
        }
    }
    for(i=0; i < allOptions; i++){
        option_list.children[i].classList.add("disabled");
    }
    next_btn.classList.add("show");
}

function showResult(){
    info_box.classList.remove("activeInfo");
    quiz_box.classList.remove("activeQuiz");
    result_box.classList.add("activeResult");
    const scoreText = result_box.querySelector(".score_text");

    if (userScore > 3){

```



```

        let scoreTag = '<span>Très bien! 🎉, votre score est de <p>'+
userScore +'</p> sur <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
    else if(userScore > 1){
        let scoreTag = '<span>Bien 😊, votre score est de <p>'+ userScore
+'</p> sur <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
    else{
        let scoreTag = '<span>Désolé 😞, votre score est de <p>'+ userScore
+'</p> sur <p>'+ questions.length +'</p></span>';
        scoreText.innerHTML = scoreTag;
    }
}

function startTimer(time){
    counter = setInterval(timer, 1000);
    function timer(){
        timeCount.textContent = time;
        time--;
        if(time < 9){
            let addZero = timeCount.textContent;
            timeCount.textContent = "0" + addZero;
        }
        if(time < 0){
            clearInterval(counter);
            timeText.textContent = "Stop";
            const allOptions = option_list.children.length;
            let correcAns = questions[que_count].answer;
            for(i=0; i < allOptions; i++){
                if(option_list.children[i].textContent == correcAns){
                    option_list.children[i].setAttribute("class", "option
correct");
                    option_list.children[i].insertAdjacentHTML("beforeend",
tickIconTag);
                    console.log("Time Off: Auto selected correct answer.");
                }
            }
            for(i=0; i < allOptions; i++){
                option_list.children[i].classList.add("disabled");
            }
            next_btn.classList.add("show");
        }
    }
}

function startTimerLine(time){
    counterLine = setInterval(timer, 19.3333);

```

```

function timer(){
    time += 1;
    time_line.style.width = time + "px";
    if(time > 549){
        clearInterval(counterLine);
    }
}

function queCounter(index){

    let totalQueCounTag = '<span><p>'+ index + '</p> sur <p>'+ questions.length
    + '</p> Questions</span>';
    bottom_ques_counter.innerHTML = totalQueCounTag;
}

```

Explication détaillée des parties principales du code

Sélection des éléments du DOM :

Le code commence par sélectionner différents éléments du DOM tels que les boutons, les boîtes d'information, les boîtes de quiz, etc., à l'aide de la méthode **document.querySelector()**.

Gestion des événements :

Les événements de clic sont attachés aux boutons de démarrage (**start_btn**), de sortie (**exit_btn**), et de continuer (**continue_btn**). Lorsque ces boutons sont cliqués, différentes actions sont déclenchées, telles que l'affichage de la boîte d'information, le démarrage du quiz, etc.

Initialisation des variables :

Diverses variables sont initialisées pour garder une trace de l'état du quiz, telles que le temps restant, le nombre de questions, le score de l'utilisateur, etc.

Fonction showQuestions(index) :

Cette fonction est chargée de présenter une question spécifique au quiz à partir des données stockées dans le fichier JSON. Elle affiche la question ainsi que les options associées.

Gestion de la sélection d'option (optionSelected(answer)) :

Cette fonction est appelée lorsque l'utilisateur sélectionne une option de réponse. Elle vérifie si la réponse est correcte ou non, met à jour le score de l'utilisateur, et affiche les icônes de validation ou d'erreur.

Fonction showResult() :

Une fois que toutes les questions ont été répondues ou que le temps est écoulé, cette fonction est appelée pour afficher le résultat final du quiz en fonction du score de l'utilisateur.

Fonctions startTimer() et startTimerLine() :

Ces fonctions gèrent le chronomètre du quiz et la barre de progression associée.

Fonction `queCounter(index)` :

Cette fonction met à jour l'affichage du compteur de questions, indiquant à l'utilisateur le nombre de questions sur le total.

Chargement des questions à partir d'un fichier JSON :

Les questions sont chargées à partir d'un fichier JSON à l'aide de la méthode `fetch()`.

II.4. FICHIER JSON

Ce code représente un tableau JSON contenant des questions et leurs réponses pour notre Quiz-RDC:

- **numb** : Numéro de la question.
- **question** : La question elle-même.
- **answer** : La réponse correcte à la question.
- **options** : Un tableau contenant quatre options parmi lesquelles l'utilisateur peut choisir la réponse.

Pour notre cas, nous avons créé un tableau JSON de cinq questions illustratives, chacune avec sa propre réponse correcte et ses options de réponse. Ces données sont utilisées dans le code JavaScript pour afficher les questions et gérer les réponses de l'utilisateur lors du quiz.

```
[
  {
    "numb": 1,
    "question": "Quelle est la capitale de la République démocratique du Congo ?",
    "answer": "Kinshasa",
    "options": [
      "Lubumbashi",
      "Kananga",
      "Kinshasa",
      "Goma"
    ]
  },
  {
    "numb": 2,
    "question": "Quelle est la langue officielle de la RDC ?",
    "answer": "français",
    "options": [
      "lingala",
      "swahili",
      "français",
      "kikongo"
    ]
  },
]
```

```

{
  "numb": 3,
  "question": "Quel fleuve traverse la RDC ?",
  "answer": "Fleuve Congo",
  "options": [
    "Fleuve Niger",
    "Fleuve Congo",
    "Fleuve Nil",
    "Fleuve Zambèze"
  ]
},
{
  "numb": 4,
  "question": "Quelle est la devise nationale de la RDC ?",
  "answer": "Justice - Paix - Travail",
  "options": [
    "Unité - Travail - Progrès",
    "Liberté - Travail - Patrie",
    "Justice - Paix - Travail",
    "Unité dans la diversité"
  ]
},
{
  "numb": 5,
  "question": "Quelle est la superficie approximative de la RDC en kilomètres carrés ?",
  "answer": "2 345 409 km²",
  "options": [
    "1 234 567 km²",
    "3 456 789 km²",
    "2 345 409 km²",
    "4 567 890 km²"
  ]
}
]

```

III. TESTS

Consigne : Utilisation du Serveur pour Charger les Contenus du Fichier JSON

Lors de ces essais, nous avons utilisé le serveur HTTP pour charger les contenus du fichier JSON. Le serveur est essentiel pour assurer le bon fonctionnement de l'application et l'accès aux données nécessaires. Nous avons suivi les étapes ci-dessous pour démarrer le serveur :

S'assurer d'avoir Node.js installé sur notre système. Si ce n'est pas le cas, le télécharger et l'installer depuis nodejs.org.

Une fois Node.js installé, ouvrons une fenêtre de terminal pour s'assurer et vérifier la version installée.

```
C:\Users\kyahi> node -v  
v20.11.1
```

Pour notre cas, Node.js était déjà installé et de même le server était aussi installé de manière globale sur notre système. Voici la commande pour installer le server :

```
npm install -g http-server
```

Accédons au **répertoire contenant les fichiers de l'application** pour démarrer le serveur en utilisant la commande :

```
http-server
```

Après avoir exécuté cette commande, nous avons vu une sortie indiquant l'adresse IP du serveur (généralement `http://127.0.0.1`) et le port sur lequel le serveur écoute (généralement 8080).

```
C:\Users\kyahi\Downloads\QUIZ-RDC> http-server  
Starting up http-server, serving ./  
  
http-server version: 14.1.1  
  
http-server settings:  
CORS: disabled  
Cache: 3600 seconds  
Connection Timeout: 120 seconds  
Directory Listings: visible  
AutoIndex: visible  
Serve GZIP Files: false  
Serve Brotli Files: false  
Default File Extension: none  
  
Available on:  
http://192.168.56.1:8080  
http://192.168.1.50:8080  
http://127.0.0.1:8080  
Hit CTRL-C to stop the server
```

Ensuite ouvrir cette adresse dans notre navigateur pour accéder aux fichiers servis par le serveur HTTP.

III.1. Page d'accueil

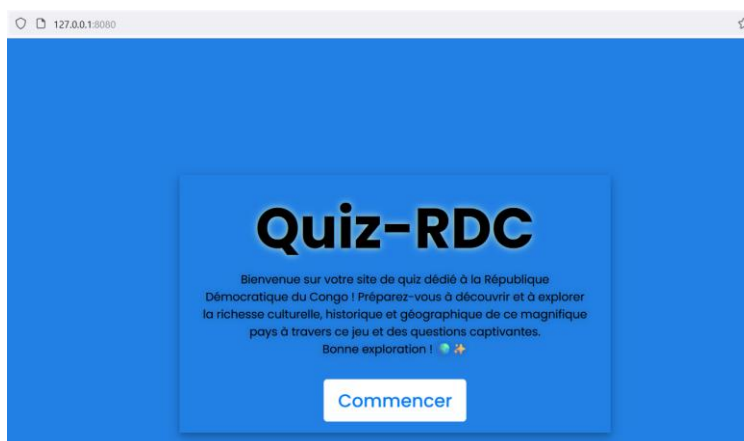


Figure 1. Page d'accueil sur un ordinateur

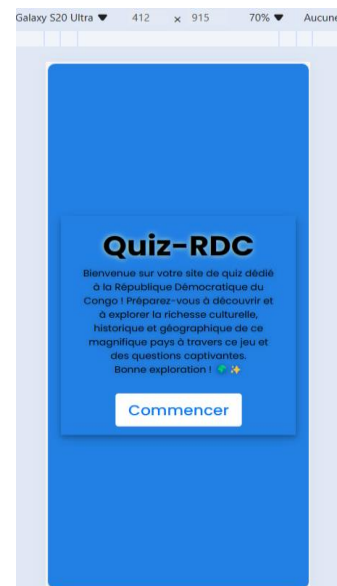


Figure 2. Page d'accueil sur un smartphone

En appuyant sur le bouton « **Commencer** », l'utilisateur est dirigé vers les règles du quiz.

III.2. Page pour les règles du quiz

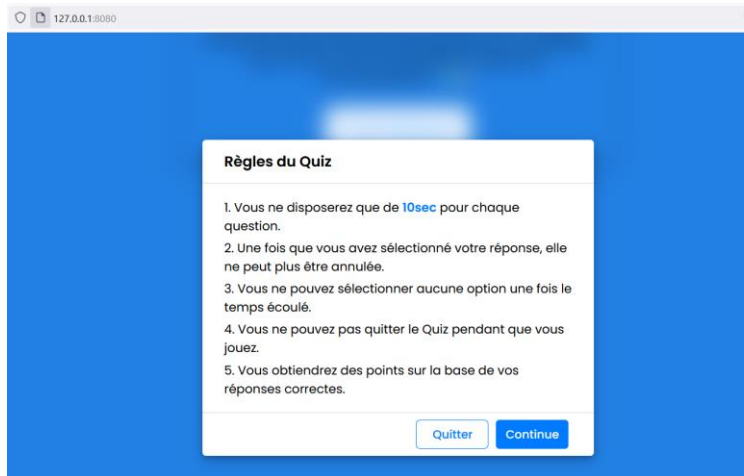


Figure 3. Page pour les règles du quiz sur un ordinateur

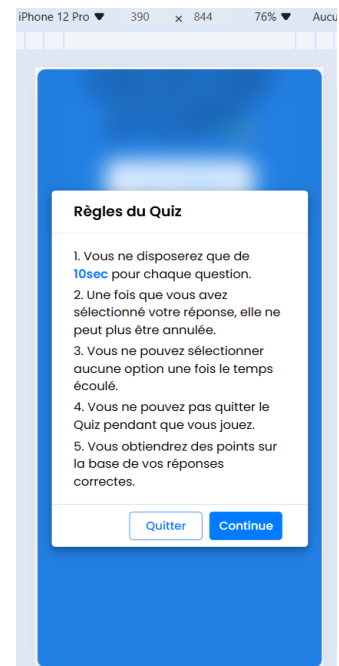


Figure 4. Page pour les règles du quiz sur un smartphone

Le bouton « **Quitter** » fait un retour à la page d'accueil, et le bouton « **Continue** » donne accès à la page des questions.

III.3. Page des questions



Figure 5. Page des questions sur un ordinateur

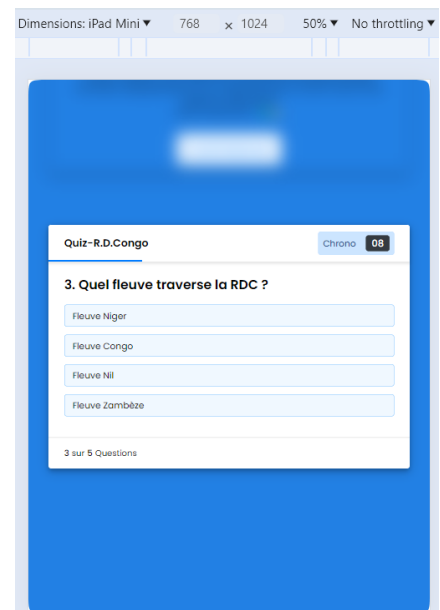


Figure 6. Page des questions sur une tablette

Sur cette page, vous trouverez un compteur à rebours (chrono 10sec) et une barre-time qui avance avec le chrono. Vous trouvez aussi la question et les options de réponse mais aussi une signalisation montra si l'utilisateur est à la quantième question sur un nombre total des questions.

Pour répondre aux questions, l'utilisateur sélectionne la réponse qu'il trouve bonne. Pour une bonne réponse, la section de la réponse devient verte et pour une mauvaise rouge en indiquant aussi la bonne. Si le chrono atteint 10sec sans que l'utilisateur ne réponde, le chrono stop et la bonne réponse est indiqué mais en considérant que l'utilisateur a échoué la question.

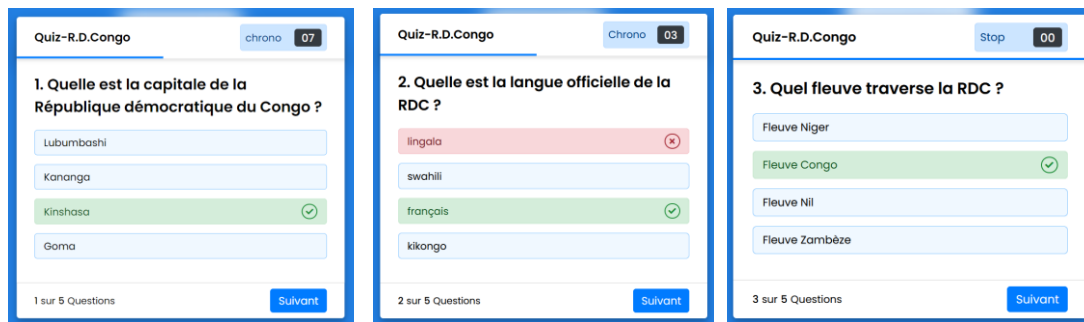


Figure 7. Cas pour la bonne réponse, la mauvaise et lorsque le chrono a atteint 10sec

Pour passer à la question suivante, l'utilisateur doit soit répondre à la question actuelle, soit il doit attendre que le chrono atteinte le 10sec.

Une fois l'utilisateur est à la fin des questions, la page du résultat s'affiche.

III.4. Page du résultat

Un message personnalisé s'affiche selon le score que l'utilisateur a obtenu (très bien, bien ou mauvais).

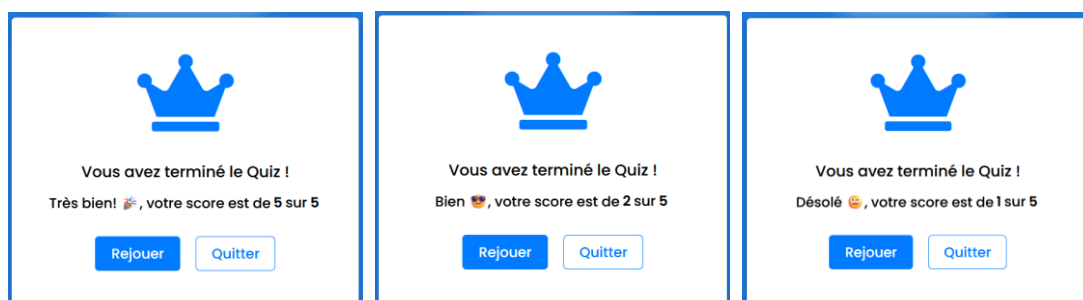


Figure 8. Résultat avec différents scores

A la fin, l'utilisateur choisi entre rejouer le quiz ou quitter le quiz pour revenir à la page d'accueil.

IV. CONCLUSION

Le développement du site web Quiz-RDC a été une expérience enrichissante, nous permettant d'explorer divers aspects de la programmation web, notamment l'utilisation de HTML, CSS, JavaScript et l'intégration de données à partir d'un fichier JSON. À travers ce projet, nous avons pu mettre en pratique nos compétences en programmation pour créer un site interactif offrant une expérience ludique et éducative aux utilisateurs.

En utilisant une approche méthodique, nous avons décomposé le processus de développement en différentes étapes, de la conception initiale à la mise en œuvre fonctionnelle. Chaque aspect du site, de la structure HTML à l'esthétique CSS et à la logique JavaScript, a été soigneusement conçu pour offrir une expérience utilisateur fluide et intuitive.

L'intégration du fichier JSON pour stocker les questions et les réponses a été particulièrement intéressante, car elle nous a permis de séparer les données du code principal, ce qui rend le site plus flexible et évolutif. De plus, l'utilisation du serveur HTTP pour charger dynamiquement les contenus du fichier JSON a renforcé l'interactivité du site en permettant une gestion dynamique des questions et des réponses.

Grâce à ce projet, les utilisateurs peuvent désormais tester leurs connaissances sur la République démocratique du Congo de manière ludique et éducative. Ce document a offert un aperçu détaillé du processus de développement et des fonctionnalités du site web Quiz-RDC, mettant en évidence les différentes étapes et parties impliquées dans sa création.