PROJET FINAL

**Réalisé par:**

Outamha Reda

Al Anbar Jawad

**Introduction:**

Ce projet, développé par Outamha Reda et Al Anbar Jaouad, est structuré en trois parties distinctes. Il comprend une application web interactive regroupant une lampe virtuelle, un portail dynamique et un jeu de deviner nombre. Ce rapport détaillera notre parcours de développement, mettant en lumière les défis rencontrés et les solutions apportées. Nous présenterons également nos réalisations finales, soulignant ainsi l'impact et la pertinence de notre travail dans le contexte contemporain.

**Partie 1 : Lampe Digitale**

La première partie de notre projet se concentre principalement sur la création d'une lampe digitale, offrant aux utilisateurs une expérience interactive en ligne. Cette section met en œuvre une combinaison de HTML, CSS et JavaScript pour concevoir et animer une représentation visuelle d'une lampe virtuelle.

**Description de la partie :**

Dans cette partie, nous avons conçu une interface conviviale permettant aux utilisateurs d'interagir avec une lampe virtuelle en un seul clic. L'objectif était de créer une expérience intuitive où les utilisateurs peuvent allumer et éteindre la lampe en cliquant sur un bouton virtuel, tout en observant des changements d'état visuel en temps réel.

**Implémentation HTML :**

La structure HTML de la lampe digitale comprend les éléments suivants :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>LAMPE</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <script src="script.js"></script>

    <div id="divlamp">

        <img id="oflampe" src="lamp.png" >

        <button id="BouttoN" onclick="Lamp()">click</button>

    </div>

    <audio id="audio" src="click.mp3"></audio>

</body>

</html>

Dans cette structure, nous avons un conteneur pour la lampe et un bouton pour allumer/éteindre la lampe.

**Implémentation CSS :**

La structure CSS de la lampe digitale comprend les éléments suivants :

body {

    font-family: Arial, sans-serif;

    background-color: #f2f2f2;

    margin: 0;

    padding: 0;

}

#divlamp {

    text-align: center;

    margin-top: 100px;

}

#oflampe {

    width: 200px;

    height: auto;

    margin-bottom: 20px;

}

#BouttoN {

    width: 120px;

    height: 50px;

    background-color: #4CAF50;

    color: white;

    border: none;

    border-radius: 25px;

    cursor: pointer;

    font-size: 18px;

    transition: background-color 0.3s;

    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

}

#BouttoN:hover {

    background-color: #45a049;

    box-shadow: 0 6px 8px rgba(0, 0, 0, 0.3);

}

La partie CSS (styles.css) est responsable de la mise en forme visuelle de la lampe

**Implémentation Java Script :**

var lalampeOff = false; //execution du 2eme evenemnt

function Lamp() {

    var btn = document.getElementById('BouttoN');

    var lmp = document.getElementById('oflampe');

    lalampeOff =! lalampeOff;  //permet le changement des photos de boutton et de lampe

    if (lalampeOff) {

        lmp.src = 'lamp.png';

    }

    else {

        lmp.src = 'lampon.png';

    }

    var audio = document.getElementById("audio");

    audio.play();

}

**Partie 2 : Portail dynamique**

Dans cette partie de notre projet, nous avons développé un portail interactif permettant aux utilisateurs de découvrir différentes sections en cliquant sur des boutons, sans que la page entière ne se recharge. Le portail dynamique offre une expérience fluide et intuitive, où les utilisateurs peuvent accéder à divers contenus sans quitter la page principale.

**Description de la partie :**

Le portail dynamique se compose de plusieurs sections, chacune associée à un bouton spécifique. Lorsque l'utilisateur clique sur l'un des boutons, le contenu de la section correspondante s'affiche instantanément, sans recharger la page. Les sections comprennent un formulaire de contact (popup), une vidéo de présentation, une photo de vous et de votre binôme, ainsi qu'un bouton pour afficher votre CV. Cette approche permet une navigation efficace et engageante, offrant aux utilisateurs un accès rapide aux différentes informations tout en maintenant une expérience utilisateur fluide et agréable. En utilisant uniquement HTML, CSS et JavaScript, nous avons développé un portail dynamique qui répond aux besoins actuels en matière d'interface utilisateur web.

**Implémentation HTML :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="styles.css">

    <title>EX2</title>

</head>

<body>

    <button id="imgCon">Image</button>

    <button id="cv">CV</button>

    <button id="VideoCon">Video</button>

    <button id="formulaireCon">Formulaire</button>

    <div id="divv"></div>

    <script src="java.js">

    </script>

</body>

</html>

**Implémentation CSS :**

button {

    padding: 10px 20px;

    font-size: 16px;

    background-color: #4caf50;

    color: white;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin: 10px;

    transition: background-color 0.3s;

}

#divv {

    margin-top: 20px;

    text-align: center;

}

#imgCon {

    background-color: #1e90ff;

}

#cv {

    background-color: #ff4500;

}

#VideoCon {

    background-color: #9400d3;

}

#formulaireCon {

    background-color: #ff69b4; /\* Hot pink \*/

}

**Implémentation Java Script :**

var imageButton = document.getElementById('imgCon');

imageButton.addEventListener('click', function() {

    var contentDiv = document.getElementById('divv');

    contentDiv.innerHTML = ''; // Vide le contenu précédent

    var image = document.createElement('img');

    image.src = "basso.jpg";

    contentDiv.appendChild(image);

});

var cvButton = document.getElementById('cv');

cvButton.addEventListener('click', function() {

    var contentDiv = document.getElementById('divv');

    contentDiv.innerHTML = ''; // Vide le contenu précédent

    var image = document.createElement('img');

    image.src = "cvvv.png";

    contentDiv.appendChild(image);

});

var videoButton = document.getElementById('VideoCon');

videoButton.addEventListener('click', function() {

    var contentDiv = document.getElementById('divv');

    contentDiv.innerHTML = ''; // Vide le contenu précédent

    var video = document.createElement('video');

    video.src = "silkala.mp4";

    video.setAttribute('controls', true);

    contentDiv.appendChild(video);

});

var formButton = document.getElementById('formulaireCon');

formButton.addEventListener('click', function() {

    var contentDiv = document.getElementById('divv');

    contentDiv.innerHTML=''; //pour eviter la repitition

    var a = document.createElement('h2');

    a.innerHTML ="<legend>Mon Formulaire</legend>";

    var b = document.createElement('h2');

    b.innerHTML="<label>Prénom:</label>  <input placeholder=Required>";

    var c = document.createElement('h2');

    c.innerHTML="<label>Nom:</label>  <input placeholder=Required>";

    var d = document.createElement('h2');

    d.innerHTML="<label>Courriel:</label>  <input placeholder=Required>";

    var e = document.createElement('h2');

    e.innerHTML="<label>Telephone:</label>  <input placeholder=Required>";

    var f = document.createElement('h2');

    f.innerHTML = "<input type=submit value=Envoyer>";

    contentDiv.appendChild(a);

    contentDiv.appendChild(b);

    contentDiv.appendChild(c);

    contentDiv.appendChild(d);

    contentDiv.appendChild(e);

    contentDiv.appendChild(f);

});

**Partie 3 : Jeu de devinez nombre**

Dans cette section de notre projet, nous avons développé un jeu interactif en JavaScript où l'utilisateur est invité à deviner un nombre généré aléatoirement par l'ordinateur. Cette partie du projet met en œuvre la manipulation d'événements en JavaScript pour détecter les actions de l'utilisateur et fournir des retours en temps réel.

**Description de la partie :**

Le jeu de devinez nombre consiste en une interface utilisateur simple comprenant un champ de texte pour que l'utilisateur saisisse ses propositions, un bouton pour vérifier la proposition, et une zone d'affichage pour donner le feedback à l'utilisateur. L'objectif est de créer une expérience ludique où l'utilisateur peut tester ses compétences en devinette tout en interagissant avec l'application de manière intuitive.

**Implémentation HTML :**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="./style.css">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Jeu</title>

</head>

<body>

    <h3>Mon Jeu</h3>

    <p>  Devinez le Nombre (entre 1 et 100) </p>

    <div class="container">

        <div class="test">

                <h4>Essayer :</h4>

                <input type="number" id="userGuess" placeholder="saisir un nombre .." id="essayer">

            <button id="btn" onclick="checkGuess()" value="">Verification</button>

        </div>

        <div class="affichage">

            <h3> feedback:</h3>

            <p id="feedback"> </p>

        </div>

    </div>

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

</body>

</html>

**Implémentation CSS:**

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body{

    width: 100%;

    height: 100vh;

    background-color: rgba(241, 207, 13, 0.993);

}

h3,h3+p{

    text-align: center;

}

h3{

    font-size: 1.5rem;

    margin: 10px 0;

}

p{

    line-height: 1.3;

    font-size: 1.5rem;

}

/\* content\*/

.container{

    width: 70%;

    height: 80%;

    margin: 0 auto;

    display: flex;

    justify-content: space-around;

    align-items: center;

}

.test{

    width: 50%;

    display: flex;

    flex-direction: column;

}

.test h4{

    font-size: 2rem;

    text-align: center;

}

.test button{

    padding: 10px 32px;

    border-radius: 10px;

    border: none;

    outline: none;

    background: #555;

    color: #eee;

    font-weight: 800;

    font-size: 1rem;

    cursor: pointer;

    transition: .3s;

}

.test button:hover{

    color: #fff;

    background-color: black;

}

.test input{

    background: #fff;

    width: 100%;

    height: 50px;

    margin:  2rem 0;

    border: none;

    outline: none;

    padding: 0 8px;

    font-size: 1.3rem;

}

.test input::placeholder{

    font-size: 20px;

}

.affichage{

    box-shadow: 2px 2px 2px ;

}

**Implémentation Java Script:**

       // Génération aléatoire du nombre à deviner entre 1 et 100

       const randomNumber = Math.floor(Math.random() \* 100) + 1;

       // Fonction pour vérifier la proposition de l'utilisateur

       function checkGuess() {

           // Récupérer la proposition de l'utilisateur depuis le champ de texte

           const userGuess = parseInt(document.getElementById('userGuess').value);

           // Vérifier si la proposition est un nombre valide

           if (isNaN(userGuess) || userGuess < 1 || userGuess > 100) {

               document.getElementById('feedback').innerText = "Veuillez entrer un nombre entre 1 et 100.";

               return;

           }

           // Comparer la proposition de l'utilisateur avec le nombre généré aléatoirement

           if (userGuess === randomNumber) {

               document.getElementById('feedback').innerText = "Bravo, vous avez deviné le nombre correctement !";

           } else if (userGuess < randomNumber) {

               document.getElementById('feedback').innerText = "Trop petit. Essayez à nouveau !";

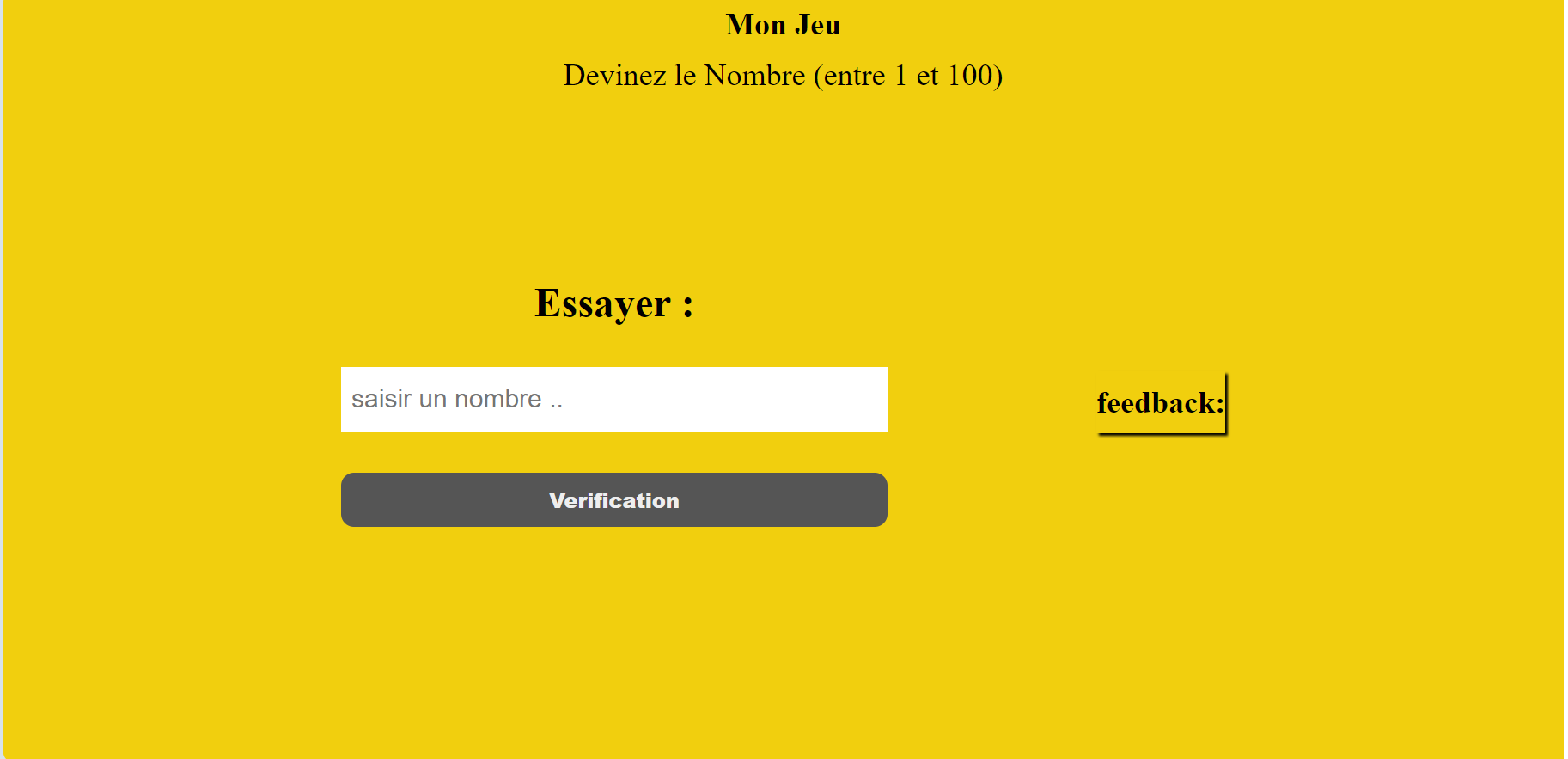
           } else {

               document.getElementById('feedback').innerText = "Trop grand. Essayez à nouveau !";

           }

       }

**Captures d'écran de l'application :**

****