BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2024**

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (recto)

Épreuve E5 - Conception et développement d'applications (option SLAM)

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE	**N° réalisation :**
Nom, prénom : ALEXANDER, Tyrese	**N° candidat : 02342378825**
Épreuve ponctuelle ☒ **Contrôle en cours de formation** ☐	**Date :** 15/04/2024

Organisation support de la réalisation professionnelle :
Association Té Mily's

Intitulé de la réalisation professionnelle : Création d'animation pour alimenter des vidéos de présentations professionnelles

Période de réalisation : Décembre 2023 **Lieu :** Cayenne (télétravail et en entreprise)
Modalité : ☒ **Seul(e)** ☐ **En équipe**

Compétences travaillées

☒ Concevoir et développer une solution applicative
☒ Assurer la maintenance corrective ou évolutive d'une solution applicative
☒ Gérer les données

Conditions de réalisation[1] (ressources fournies, résultats attendus)
Ressources fournies: description du contexte, de l'existant, expression du besoin
Résultats attendus: Création d'animation et de page web à placer en module sur vidéos commerciales, tests fonctionnels, documentation technique, documentation utilisateur, déploiement de l'application

Description des ressources documentaires, matérielles et logicielles utilisées[2]

- Environnement de travail collaboratif (Miro)
- Environnement de développement : Visual Studio Code
- Tests de comportement anormaux: tests unitaires, tests fonctionnels
- Langage: PHP, JavaScript, SSS, HTML, WolframAlpha
- Gestion des versions: Git (GitHub)

Modalités d'accès aux productions[3] et à leur documentation[4]

Page du portfolio concernant la réalisation : whoamitty.github.io/alexander_tyreseportfolio
(liens, document technique, compte rendu, liens liveYouTube, page internet, etc. .)

[1] En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.
[2] Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.
[3] Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.
[4] Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

BTS SERVICES INFORMATIQUES AUX ORGANISATIONS **SESSION 2024**

ANNEXE 9-1-B : Fiche descriptive de réalisation professionnelle (verso, éventuellement pages suivantes)

Épreuve E5 - Conception et développement d'applications (option SLAM)

Descriptif de la réalisation professionnelle, y compris les productions réalisées et schémas explicatifs

Contexte :

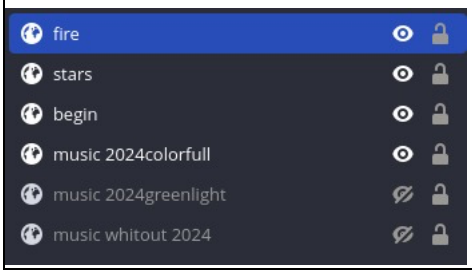
L'Association Té Mily's a décidé de faire des Lives pendant la saison des fêtes. Ces Lives seront largement déployés sur tous les réseaux sociaux (YouTube, Twitch, Instagram, Facebook).
Afin de rendre ces Lives plus ludiques et moins redondant, l'association souhaiterait une solution pour animer les vidéos au début de live (salle d'attente). Ces animations devront représenter les valeurs de l'association.

Solutions proposées :

J'ai proposé à l'Association la création de pages Web qui serviront de module à placer sur la vidéo en début de Live. Logiciel OBS.

Étape 1 :

Étant en période de Noël, j'ai proposé de faire un module d'animation de flocons de neige qui tombe avec un sapin sur la vidéo.



count.js | snow.js

afk > js > snow.js > ...

```js
// Configuration des flocons de neige
const snowflakesContainer = document.body;
const numberOfSnowflakes = 100;

Codeium: Refactor | Explain | Generate JSDoc | ×
function createSnowflake() {
  const snowflakeElement = document.createElement('div');
  snowflakeElement.classList.add('snowflake');
  snowflakeElement.textContent = Math.random() > 0.5 ? '*' : '+';
  snowflakesContainer.appendChild(snowflakeElement);

  snowflakeElement.style.left = Math.random() * 100 + 'vw';
  snowflakeElement.style.opacity = Math.random();
  snowflakeElement.style.transform = 'scale(' + Math.random() + ')';
  snowflakeElement.style.animationDuration = Math.random() * 3 + 2 + 's';
  // snowflakeElement.style.animationDuration = Math.random() + 2 + 's';

  snowflakeElement.style.animationName = 'fall';
  snowflakeElement.style.animationIterationCount = 'infinite';
  snowflakeElement.style.animationTimingFunction = 'linear';

  // Suppression du flocon de neige après qu'il soit tombé
  snowflakeElement.addEventListener('animationend', function() {
    snowflakeElement.remove();
  });
}

// Génération initiale de flocons de neige
for (let i = 0; i < numberOfSnowflakes; i++) {
  createSnowflake();
}

// Génération continue de flocons de neige
setInterval(createSnowflake,1000 );
```

count.js | snow.js | stars.js

afk > js > stars.js > ...

```
const canvas = document.getElementById('starsCanvas');
const ctx = canvas.getContext('2d');

canvas.width = window.innerWidth;
canvas.height = window.innerHeight;

Codeium: Refactor | Explain
class Star {
    Codeium: Refactor | Explain | Generate JSDoc | ×
    constructor() {
        this.reset();
    }

    Codeium: Refactor | Explain | Generate JSDoc | ×
    reset() {
        this.x = Math.random() * canvas.width;
        this.y = Math.random() * canvas.height;
        this.opacity = Math.random();
        this.radius = Math.random() * 2;
    }

    Codeium: Refactor | Explain | Generate JSDoc | ×
    update() {
        // Change the opacity randomly to create a twinkling effect
        this.opacity = Math.random();
        // Randomly reset star to create a new star effect
        if (Math.random() > 0.95) {
            this.reset();
        }
    }

    Codeium: Refactor | Explain | Generate JSDoc | ×
    draw() {
        ctx.beginPath();
        ctx.arc(this.x, this.y, this.radius, 0, Math.PI * 2);
        ctx.fillStyle = `rgba(255, 255, 255, ${this.opacity})`; // white color with variable opacity
        ctx.fill();
    }
}

let stars = [];

Codeium: Refactor | Explain | Generate JSDoc | ×
function createStars(count) {
    for (let i = 0; i < count; i++) {
        stars.push(new Star());
    }
}
```

count.js

afk > js > count.js > ...

```js
// Heure cible sous la forme 'HH:MM'
let heureCible = '22:00';

Codeium: Refactor | Explain | Generate JSDoc | ×
function obtenirHeureDuLive(heure) {
  const maintenant = new Date();
  const [heures, minutes] = heure.split(':').map(Number);
  let cible = new Date(maintenant.getFullYear(), maintenant.getMonth(), maintenant.getDate(), heures, minutes, 0);

  return cible;
  /*   // Si l'heure cible est déjà passée, réglez-la pour le lendemain
  if (maintenant >= cible) {
    cible.setDate(cible.getDate() + 1);
  }
  */
}

var countDownDate = obtenirHeureDuLive(heureCible);

// Mettre à jour le compteur toutes les secondes
var x = setInterval(function() {
  var now = new Date().getTime();
  var distance = countDownDate - now;

  // Calculs du temps pour les heures, minutes et secondes

  var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
  var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
  var seconds = Math.floor((distance % (1000 * 60)) / 1000);

  // Affichage du résultat dans l'élément avec l'id "compteARebours"
  // document.getElementById("compteARebours").innerHTML = hours + "h " + minutes + "m " + seconds + "s";
  document.getElementById("compteARebours").innerHTML = hours + "h " + minutes + "m " + seconds + "s";

  // Si le compte à rebours est terminé, afficher un message
  if (distance < 0) {
    clearInterval(x);
    document.getElementById("compteARebours").innerHTML = "Quelques instants...";
    // Redirection optionnelle ou autres actions ici
  }
}, 1000);
```

Il a fallu générer l'animation d'un sapin avec un logiciel. J'ai dû modifier le code de couleur afin que des boules rouges s'affichent en plus des boules blanches. J'ai dû programmer dans le langage WolframAlpha. J'ai pris en vidéo le sapin animé et intégrer à la page

Étape 2 :

Pour intégrer pour intégrer les pages j'ai dû créer une source pour chacune d'entre elles, et dans la configuration de la source, mettre le lien vers la page.
J'ai également programmé un compteur pour le début de chaque live de l'Association

Étape 3 :
J'ai également ajouté les animations créés (étoiles) sur la page internet du l'association Té Mily's.
J'ai fait ensuite une documentation technique.

Bilan : L'application est opérationnelle et répond aux attentes de l'expression des besoins. Les tests sont réalisés, la documentation technique est en ligne.
Les informations détaillées sont dans le compte rendu, accessible en ligne, via la page du portfolio qui présente cette réalisation professionnelle (voir lien au recto de cette fiche).