

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 : 1		
Nom, prénom : BOUHAJJA Mariem		N° candidat : 01950008457		
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>	Date : 14 / 06 /2024		
Organisation support de la réalisation professionnelle La société StreamIO. est une jeune startup de 5 personnes créée en 2020 qui innove dans le domaine de la production audio-visuelle sur Internet. Elle propose des services d'édition vidéo, de diffusion en direct sur Internet, de création de contenu numérique, ainsi que la gestion d'identité numérique pour les vidéastes du web (Youtuber/Streamer), elle propose aussi des salles pour diffusion en direct. Comment faciliter la réservation des salles ?				
Intitulé de la réalisation professionnelle Mise en place d'une application web de réservation de salle (CRS)				
Période de réalisation : 11/09/23 au 18/12/23 Lieu : ISCIO – CFA EVE Modalité : <input checked="" type="checkbox"/> Seul(e) <input type="checkbox"/> En équipe				
Compétences travaillées <input checked="" type="checkbox"/> Concevoir et développer une solution applicative <input checked="" type="checkbox"/> Assurer la maintenance corrective ou évolutive d'une solution applicative <input checked="" type="checkbox"/> Gérer les données				
Conditions de réalisation (ressources fournies, résultats attendus) <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top; border: none;"> <u>Ressources fournies :</u> - Cahier des charges StreamIO - Environnement de développement </td> <td style="width: 50%; vertical-align: top; border: none;"> <u>Résultats attendus :</u> - Un site en HTML, CSS , JS - Une API REST pour la gestion des données en Node.js - Une base données MySQL - Documentation d'utilisation - Documentation technique </td> </tr> </table>			<u>Ressources fournies :</u> - Cahier des charges StreamIO - Environnement de développement	<u>Résultats attendus :</u> - Un site en HTML, CSS , JS - Une API REST pour la gestion des données en Node.js - Une base données MySQL - Documentation d'utilisation - Documentation technique
<u>Ressources fournies :</u> - Cahier des charges StreamIO - Environnement de développement	<u>Résultats attendus :</u> - Un site en HTML, CSS , JS - Une API REST pour la gestion des données en Node.js - Une base données MySQL - Documentation d'utilisation - Documentation technique			
Description des ressources documentaires, matérielles et logicielles utilisées - Système Windows pour l'exécution du code source du site Web - Système Linux pour l'exécution du code source de l'API - Documentation d'utilisation du site web - Documentation de déploiement du code source et intégration des données				
Modalités d'accès aux productions et à leur documentation URL : https://tinyurl.com/iscio-bts-sio-2024 Mot de passe : 1SCIO@91				

**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 :**

La société StreamIO est une jeune startup de 5 personnes créée en 2020 qui innove dans le domaine de la production audio-visuelle sur internet.

Elle propose des services d'édition vidéo, de diffusion directe sur internet, de création de contenu numérique, mais aussi de la gestion d'identité numérique pour les vidéastes du web (Youtubeurs/Streamers...).

Cette société propose aussi des salles à réserver pour filmer/ produire des vidéos accompagnées ou pas par un équipement de tournage selon le choix du client.

M. Miller, directeur de StreamIO, nous a contacté suite à une hausse de demandes concernant la réservation de ces salles.

Après nous avoir expliqué que les demandes de réservations des clients sont envoyées par mail et traitées par l'équipe commerciale, ce dernier a exprimé que cette solution n'est pas optimale : les clients ne voient pas les salles réservées, ils ont tendance à toujours demander les mêmes dates, et l'équipe perd du temps à leur refuser leurs demandes et à leur proposer de nouveaux créneaux. De plus, les membres de l'équipe commerciale communiquent les salles qui sont déjà réservées entre eux par un tableau dessiné dans la salle de repos : parfois, une personne peut oublier de remplir le tableau, ce qui peut conduire à la réservation de la même date pour deux clients différents, créant ainsi des désorganisations.

Le directeur estime même que la réservation des salles des salles devrait se faire automatiquement sans être gérée par l'équipe commerciale.

M. Miller nous a demandé alors une solution sous forme d'application qui permet de mieux organiser ces réservations et les communiquer aux clients d'une façon simple et claire.

Le besoin :

Le besoin principal ici semble être d'avoir la possibilité d'afficher les dates où les salles sont réservées aux clients ainsi que de permettre la réservation de ces salles sans devoir à passer par l'équipe commerciale.

Après discussion avec M. Miller, nous avons décidé que la solution optimale sera la mise en place d'un calendrier de réservation qui affiche les disponibilités des salles et qui fournit un formulaire pour les réserver.

Analyse du besoin :

Après entretien avec M. Miller, nous nous sommes mis d'accord sur les fonctionnalités suivantes :

Fonction principale :

- Afficher les salles disponibles et fournir un formulaire pour les réserver.

Fonctions secondaires :

- Avoir dans le formulaire la possibilité d'ajouter les coordonnées de la personne et les détails.
- Affichage des salles sous forme de calendrier.
- Avoir dans le calendrier des couleurs pour pouvoir distinguer facilement les disponibilités.

- Afficher une option : matériels inclus (l'utilisateur sélectionnera le matériel dont il a besoin et qui sera réservé alors par l'équipe des gestionnaires).
- Afficher une liste des matériels qu'on peut réserver.
- Pour faciliter la gestion des réservations, la durée minimale d'une réservation est d'une heure et les réservations sont indivisibles.

Suite a nos entretiens nous nous somme mis d'accord sur la maquette suivante :

February 2024

Dim	Lun	Mar	Mer	Jeu	Ven	Sam
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	1	2

Time Select
Pick and reserve a time slot with me for a variety of services!

9:00 – 10:00 10:00 – 11:00
11:00 – 12:00 12:00 – 13:00
13:00 – 14:00 14:00 – 15:00
15:00 – 16:00 16:00 – 17:00

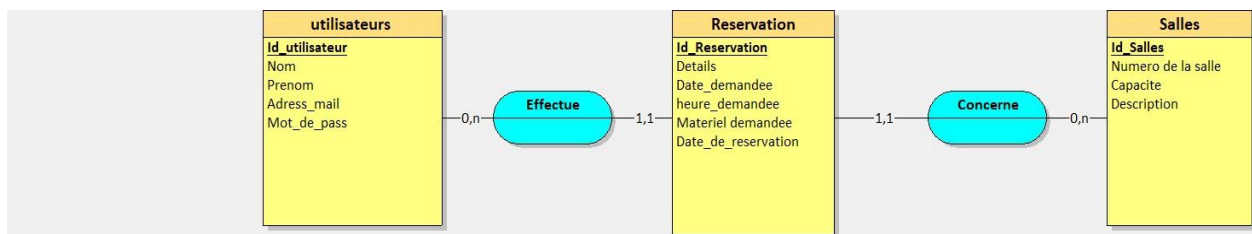
Booking Form
Reservez votre salle!

Equipement: Choisissez votre equipement:

Détails:
Ajoutez vos détails ici

Envoyer

Après étude de besoin nous avons conçu la base de données MySQL :



Technologies utilisées :

Langages de programmation utilisées :

- JavaScript ES6
- Node.js v18.17.1
- HTML 5
- CSS 4
- MySQL

Librairies/framework utilisées :

- Bootstrap :
<https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/css/bootstrap.min.css>
- Google Fonts API : <https://fonts.googleapis.com/css?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200>
- Express : <https://expressjs.com/>
- Jest : <https://jestjs.io/>
- Nodemon : <https://nodemon.io/>

Les logiciels utilisés :

- WAMP : wamp server 64 3.3.2
- LucidChart : pour la conception des diagrammes d'utilisation
- Looping: pour la conceptualisation de la base de données
- phpmyadmin qui est une application web de gestion pour la gestion de SGBD (systèmes de gestion de la base de données)
- VisualStudio Code version : x64

Ordinateur :

- Windows 10

Procédure de la réalisation :

1. Création de la base de donnée : le logiciel Looping m'a permis de générer automatiquement le fichier SQL pour la création de la base de donnée, j'ai importer ce fichier dans phpmyadmin et donc ma base correspond au schéma relationnel

2. Développement du Frontend : mon Frontend est composé de deux parties ; une page de connexion/inscription et une deuxième page a laquelle on accède une fois authentifiés et qui comporte le calendrier de réservation

3. Développement de l'API : mon API comporte 5 routes

=>/disponibilité : cette route permet de récupérer les réservations déjà effectués pour une salle et une date précise

=>/réservation : cette route permet d'effectuer une réservation

=>/salles : cette route permet de récupérer les noms des salles stockées dans la base de donnée

=>/signin : cette route permet d'ajouter un utilisateur a la base de donnée

=>/login : cette route permet de récupérer un utilisateur de la bdd pour un mail et un mot de passe donnés

4. Mise en relation entre le Frontend et l'API : après la création de l'API avec node.js et le frontend avec HTML, CSS et JS, j'ai relié les deux applications en utilisant des fonctions de type Fetch pour récupérer et/ou ajouter des données dans la bdd.

Exemple : j'utilise une fonction fetchdata qui va récupérer la date et la salle que l'utilisateur veut consulter et qui va faire appel à la route « /disponibilité ». Je récupère les réservations pour cette date et je les affiche dans le calendrier.

Testes du bon fonctionnement :

Pour m'assurer du bon fonctionnement de mon application j'ai effectué plusieurs types de tests :

- des tests unitaires : en utilisant Jest, Jest.Mock(pour simuler ma base de données) et supertest (pour pouvoir tester les routes d'une API)
- des tests en utilisant Postman : pour pouvoir voir les réponses des requêtes
- des test manuels : en effectuant une action dans le Front et voir si il y a les changements attendus dans la base de données