

**2025**



# RAPPORT DE PROJET E6

# Projet : Borne Orientation

Ce projet, réalisé sur une durée de 10 semaines, est développé par notre équipe TEMA (Tony BARELLI, Evan BONNAL, Matthieu CAVIGGIA, Alex NGUYEN)

# Sommaire

I.	<i>Remerciements</i>	2
II.	<i>Introduction</i>	3
III.	<i>Cahier Des Charges</i>	4
IV.	<i>Ressources mises à disposition</i>	5
V.	<i>Répartition des rôles et tâches dans le projet</i>	6
VI.	<i>Diagramme cas d'utilisation UML</i>	7
VII.	<i>Diagramme de séquence UML</i>	8
VIII.	<i>Synoptique du projet</i>	9
IX.	<i>Diagramme de Gantt</i>	10
X.	<i>Diagramme d'exigence</i>	11
XI.	<i>Etudiant 3 (Alex NGUYEN)</i>	12
XII.	<i>Incrustation des images</i>	13
	1 <sup>er</sup> scénario : Les photos du parcours et des flèches ont été mis en œuvre.	14
	2 <sup>ème</sup> scénario : Ajout d'une image de fond et d'une flèche en superposé	15
	3 <sup>e</sup> scénario : Positionnement de la flèche et génération de l'image finale	17
	4 <sup>e</sup> scénario : Tester le fonctionnement du site web	21
XIII.	<i>Implémentation de l'envoi d'e-mails</i>	22
	Chargement de la bibliothèque PHPMailer et traiter des données du formulaire	23
	Configuration du serveur SMTP via IONOS et générer l'envoi de mails en PHP	24
	Envoie du mail	26
XIV.	<i>Sécurisation de l'accès par authentification administrateur</i>	28
	Contrôle d'accès par session en PHP	29
	Ajout d'un bouton de déconnexion	30
	Validation du fonctionnement du contrôle d'accès et de la déconnexion	31
XV.	<i>Problème rencontré</i>	32
XVI.	<i>Mise en commun</i>	33
XVII.	<i>Conclusion</i>	34
XVIII.	<i>Annexes</i>	35

## I. Remerciements

Nous tenons à remercier M. NARELLI, notre enseignant informatique et réseaux, pour son aide dans la compréhension du projet et du cahier des charges, ainsi que pour nous avoir prêté tout le matériel nécessaire à la réalisation de ce projet.

Enfin merci à l'ensemble du corps enseignant pour l'encadrement des projets de deuxième année, qui nous permettent de découvrir la gestion de projets, le travail en groupe et les moyens de communications mis en œuvre.



## II. Introduction

---

Mon équipe constituée de Tony BARELLI, Evan BONNAL, Matthieu CAVIGGIA, et moi-même, élèves de 2<sup>e</sup> année BTS CIEL (Cybersécurité Informatique et réseaux, Electronique), spécialisation Informatique et Réseaux, travaille sur la réalisation d'une borne d'orientation. Ce dispositif permettra d'orienter les visiteurs lors des « journées portes ouvertes » et de les informer sur la nature du lieu visité, avec pour objectif d'aider le département SAMSAH TSA84 (Vaucluse) de l'association « La Bourguette ». Cette dernière, reconnue d'utilité publique, gère sept établissements et propose des services de soins, d'adaptation et de formation aux personnes différentes.

Ce projet vise à développer un système d'orientation pour les usagers au sein de leurs établissements, dont l'architecture peut parfois être labyrinthique. Dans un premier temps, la borne facilitera la navigation dans ces structures, mais l'objectif à long terme, cette solution pourrait être étendue aux espaces publics pour améliorer l'accessibilité des personnes en situation de handicap.

Ainsi, la personne et sa famille, une fois arrivées à destination, pourront, si elles le souhaitent, s'informer sur la nature et la mission du département.

## III. Cahier Des Charges

### Fonction d'usage

Lors de la visite, une personne ou un membre du groupe disposant d'un smartphone pourra sélectionner parmi plusieurs formations de BTS (CIEL A, CIEL B, ATI, ET, CRSA, CSST) une destination sur une borne équipée d'un écran tactile. En fonction de ce choix, un QR code sera généré automatiquement. En le scannant avec son smartphone, l'utilisateur sera redirigé vers un site web de guidance géographique, qui lui fournira un itinéraire adapté.

### Contraintes de performances

Une fois l'utilisateur arrivé à destination, un fichier audio et/ou une vidéo explicative sur le fonctionnement du lieu pourra être déclenché via un menu simple et intuitif. De plus, une proposition de newsletter par mail sera faite aux visiteurs intéressés, leur permettant de recevoir des informations complémentaires sur les formations et les événements à venir.

### Contraintes techniques

- Génération automatique de QR codes en fonction de l'URL pour permettre aux visiteurs d'accéder facilement à leur itinéraire personnalisé.
- Développement d'un générateur d'itinéraires basé sur l'incrustation dynamique de symboles et de textes sur des images via PHP, facilitant la gestion des trajets par les administrateurs du lieu.
- Choix d'un hébergeur compatible avec l'envoi de mails.
- Intégration d'un lecteur de carte RFID connecté au RPI4 pour renforcer la sécurité de la borne d'orientation. En cas de panne ou d'intervention malveillante, l'accès à la borne pourra être restreint et réactivé uniquement via un badge RFID dédié.

Délai de réalisation : 10 semaines

## IV. Ressources mises à disposition

Référence	Caractéristique	Site	Quantité	Prix
<b>RPI 4 Model B Kit</b>	KITPI44GB	KUBII	1	99,00 €
<b>Support d'écran BONTEC</b>	BONTEC Support TV	Support TV Trépied pour Écrans Plasma/LCD/LED Plats 32-60 Pouces	Amazon	1 51,43 €
<b>Écran tactile</b>	HP Écran tactile FHD E24t G5 82999392	Full HD de 1920 x 1080 1x DisplayPort, 1x HDMI	Bureau.fr	1 282,00 €
<b>Lecteur de carte RFID</b>	Module WMHHA179	Lecteur de carte RFID Avec 2 cartes	GoTronic	1 39,90 €
<b>Adaptateur lecteur RFID vers PC</b>	AltonaLab LTDUGS RM-ALTO-007	Module GPIO USB à 8 canaux avec entrées analogiques	GoTronic	1 51,16 €
<b>IONOS</b>	Hébergement du site web	Hébergement web Sécurisé+mailing	IONOS	1 3 € / mois
	TOTAL :			526,49 €

## V. Répartition des rôles et tâches dans le projet

Dans le cadre de notre projet, nous sommes une équipe de quatre étudiants, et chacun d'entre nous a un rôle spécifique afin de mener à bien la réalisation d'une borne d'orientation. La répartition des tâches est la suivante :

### Etudiant 1 (Matthieu CAVIGGIA) :

**Rôle :** Développement front-end avec Framework Bootstrap V4

- Conception site web responsive par bootstrap4
- Génération QR-code pour sélectionner une formation avec langage JavaScript
- Remédiation bug rapporté par client
- Conception et Installation d'un ordinateur RPI4 doté d'un écran tactile
- Mise en place lecteur RFID

### Etudiant 2 (Tony BARELLI) :

**Rôle :** Développement full-stack avec Framework Bootstrap V4

- Concevoir le simulateur de déplacement de chaque formation du BTS
- Ajouter les contacts des utilisateurs
- Remédiation bug rapporté par client
- Concevoir la table « itinéraire » en BDD

### Etudiant 3 (Alex NGUYEN) :

**Rôle :** Développement full-stack

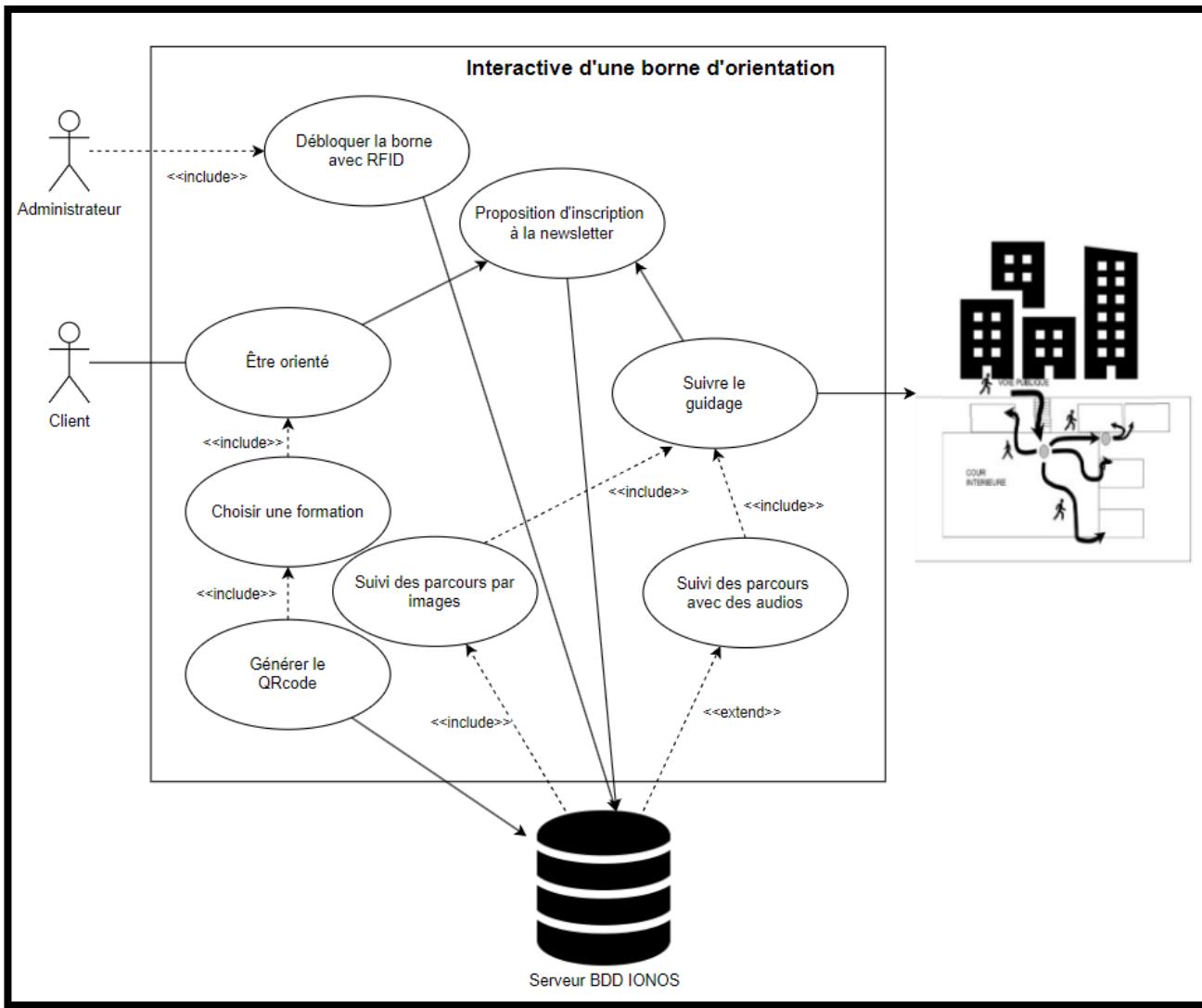
- Conception site web générateur d'itinéraire
- Configuration du mailing et certification SSL
- Utilisation du composant logiciel PHP pour l'envoi des mails
- Remédiation bug rapporté par client
- Configurer l'hébergement d'un site web en utilisant le service "IONOS"

### Etudiant 4 (Evan Bonnal) :

**Rôle :** Développement full-stack

- Authentifier l'administrateur du site
- Valorisation : calcul statistiques, courbes, export Excel
- Cybersécurité : rendre impossible la modification de la querystring de l'URL
- Conception et Installation d'un ordinateur RPI4 doté d'un écran tactile

## VI. Diagramme cas d'utilisation UML



### Orientation du client

Selectionner une formation.  
Générer un QR Code pour accéder à un itinéraire personnalisé.  
Suivi du parcours avec des images et/ou des audios.

### Suivi du guidage

Une fois le QR Code scanné, l'utilisateur suit l'itinéraire proposé.

### Acteurs :

**Client** : Utilise la borne pour s'orienter, choisir une formation et obtenir un guidage.

**Administrateur** : Peut débloquer la borne en cas de problème grâce à un système RFID.

### Inscription à la newsletter

Proposition d'abonnement pour recevoir des informations sur les formations.

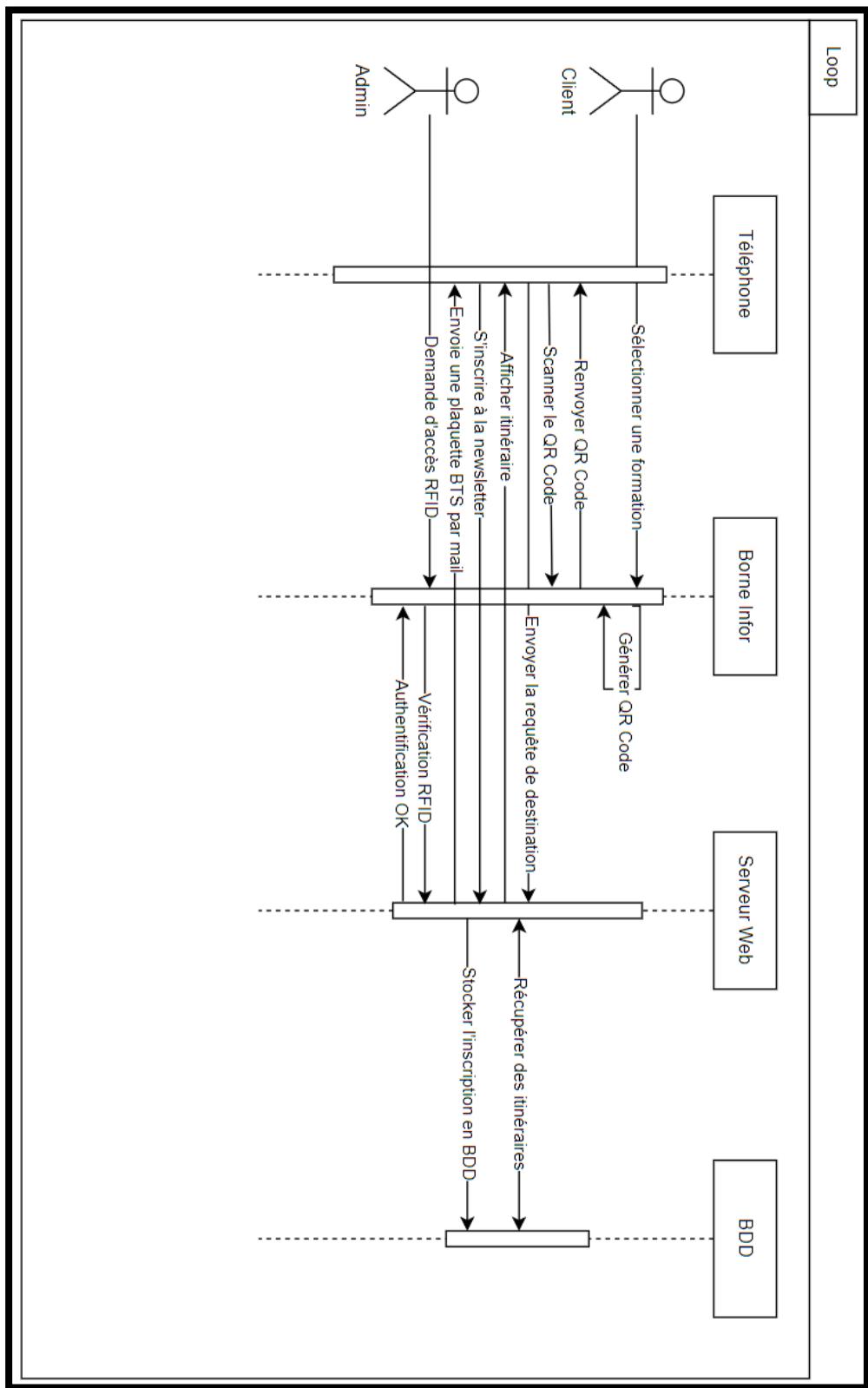
### Sécurisation de la borne

L'administrateur peut la débloquer avec RFID en cas de panne ou d'accès non autorisé.

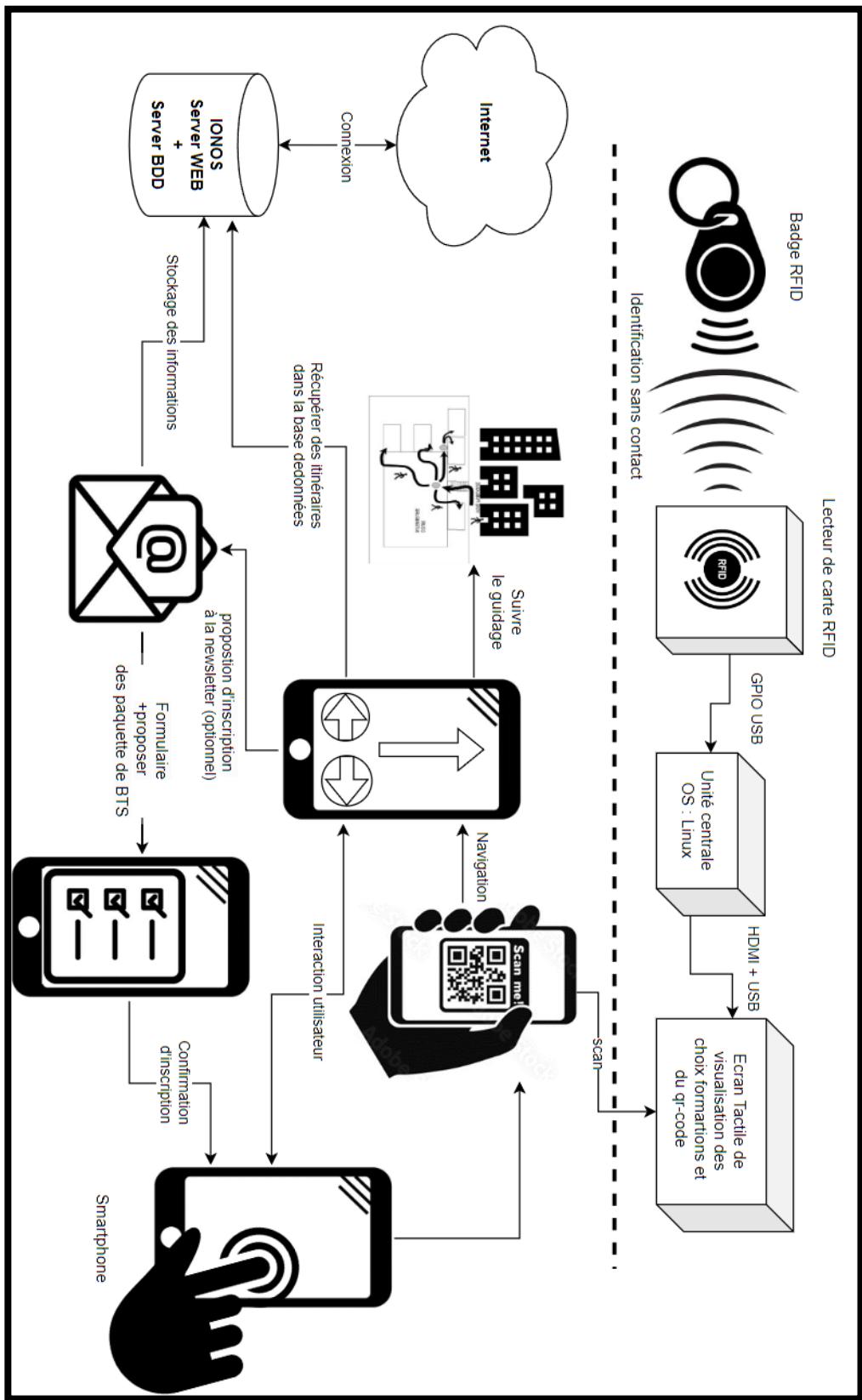
### Base de données (Serveur IONOS)

Stocke les itinéraires, QR Codes et la newsletter

## VII. Diagramme de séquence UML



## VIII. Synoptique du projet

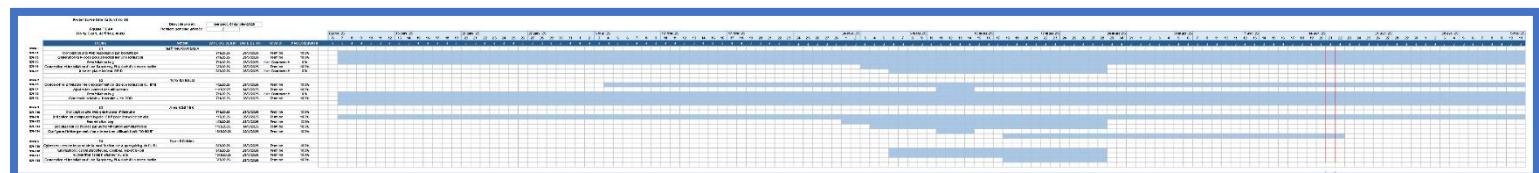


# IX. Diagramme de Gantt

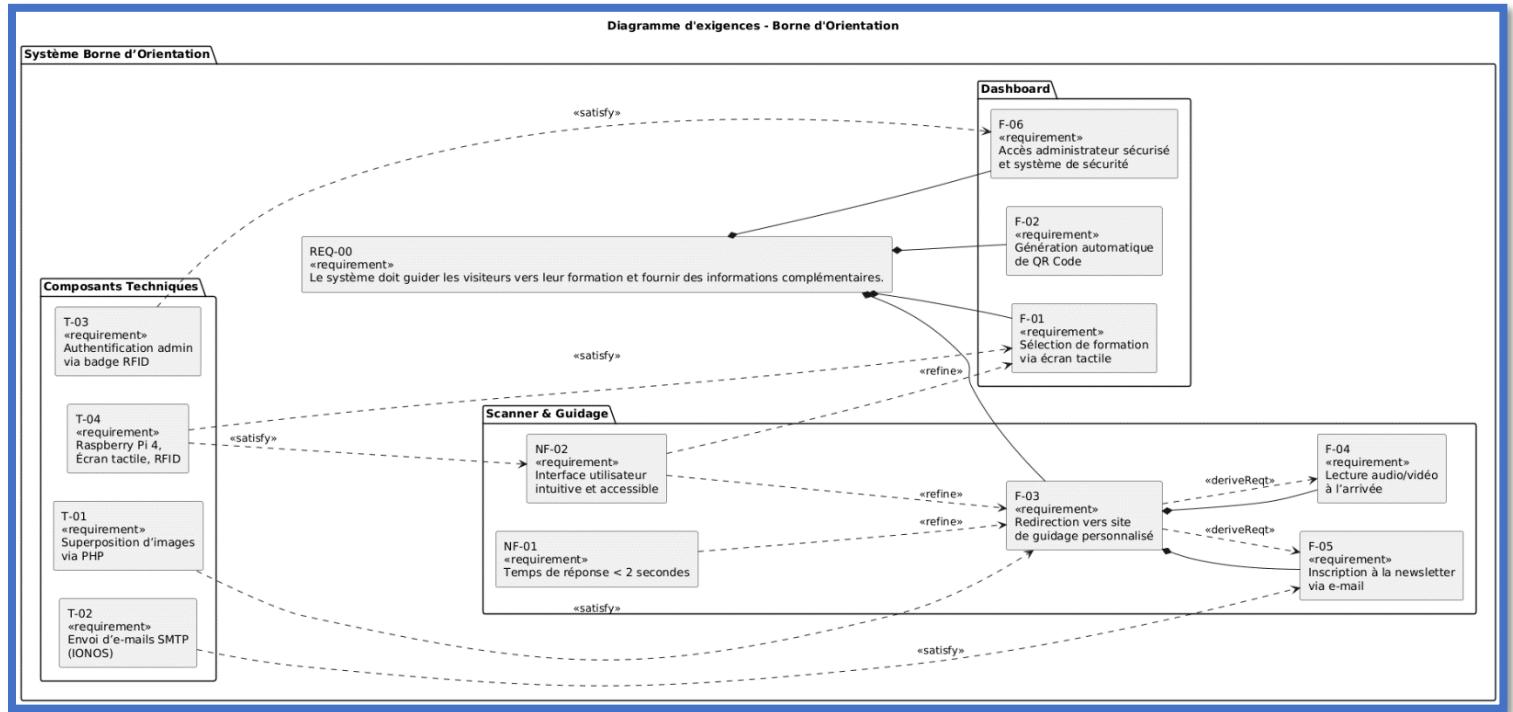
Voici le lien du Diagramme de Gantt pour mieux visualiser :

[https://docs.google.com/spreadsheets/d/15KmgNGNBMsIUy\\_CNxxtfsMC6VmbXhnEo/edit?usp=sharing&ouid=115592501527481610452&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/15KmgNGNBMsIUy_CNxxtfsMC6VmbXhnEo/edit?usp=sharing&ouid=115592501527481610452&rtpof=true&sd=true)

Projet Borne Orientation Info E6		Début du projet :	mercredi, 01-janvier-2025				
		Première semaine affichée :	2				
	TACHE	Attribut	DATE DE DEBUT	DATE DE FIN	STATUT	PROGRESSION	
Phase 1	<b>E1</b>	Matthieu CAVIGGIA					
SD1-T1	Conception site web responsive par bootstrap4		7/1/2025	28/3/2025	Terminé	100%	
SD1-T2	Génération QR-code pour sélectionner une formation		7/1/2025	28/3/2025	Terminé	100%	
SD1-T3	Remédiation bug		7/1/2025	28/3/2025	Non Commencé	0%	
SD1-T4	Conception et Installation d'une Raspberry Pi 4 doté d'un écran tactile		3/3/2025	28/3/2025	Terminé	100%	
SD1-T5	Mise en place lecteur RFID		6/3/2025	28/3/2025	Non Commencé	0%	
Phase 2	<b>E2</b>	Tony BARELLI					
SD1-T6	Concevoir le simulateur de déplacement de chaque formation du BTS		4/2/2025	28/3/2025	Terminé	100%	
SD1-T7	Ajouter les contact des utilisateurs		11/3/2025	14/3/2025	Terminé	100%	
SD1-T8	Remédiation bug		7/1/2025	28/3/2025	Non Commencé	0%	
SD1-T9	Concevoir la table « itinéraire » en BDD		7/1/2025	28/3/2025	Terminé	100%	
Phase 3	<b>E3</b>	Alex NGUYEN					
SD1-T10	Conception site web générateur d'itinéraire		7/1/2025	28/2/2025	Terminé	100%	
SD1-T11	Utilisation du composant logiciel PHP pour l'envoi des mails		1/3/2025	28/3/2025	Terminé	100%	
SD1-T12	Remédiation bug		4/3/2025	28/3/2025	Terminé	100%	
SD1-T13	Sécurisation de l'accès par authentification administrateur		11/3/2025	14/3/2025	Terminé	100%	
SD1-T14	Configurer l'hébergement d'un site web en utilisant l'outil "IONOS"		18/3/2025	22/4/2025	Terminé	100%	
Phase 4	<b>E4</b>	Evan BONNAL					
SD1-T15	Cybersec : rendre impossible la modification de la querystring de l'URL		6/3/2025	28/3/2025	Terminé	100%	
SD1-T16	Valorisation : calcul statistiques, courbes, export Excel		6/3/2025	28/3/2025	Terminé	100%	
SD1-T17	Authentifier l'administrateur du site		18/3/2025	28/3/2025	Terminé	100%	
SD1-T18	Conception et Installation d'une Raspberry Pi 4 doté d'un écran tactile		3/3/2025	28/3/2025	Terminé	100%	



# X. Diagramme d'exigence



- Dans ce diagramme d'exigences, nous décrivons les besoins liés à la borne d'orientation mise en place pour guider les visiteurs d'un établissement.
- Le système permet à l'utilisateur de sélectionner une formation sur un écran tactile, de générer un QR code, puis de suivre un parcours personnalisé vers sa destination.
- Des exigences non-fonctionnelles, comme la rapidité d'affichage ou la simplicité d'utilisation, viennent raffiner les attentes utilisateurs.
- Enfin, les exigences techniques comme l'utilisation du Raspberry Pi, du module RFID ou de PHP permettent de satisfaire les fonctionnalités prévues.

## XI. Etudiant 3 (Alex NGUYEN)

---

# Partie individuelle

## XII. Incrustation des images

Dans le cadre de ce projet, nous avons développé une site web interactive intégrée à une borne d'orientation. Celle-ci repose sur **l'incrustation dynamique d'images**, de symboles ou de textes sur un fond cartographique via une interface web développée en **HTML, CSS, JavaScript** et **PHP**. L'objectif est de permettre aux administrateurs d'un établissement de créer et de générer facilement des itinéraires personnalisés, en superposant des repères visuels (flèches, pictogrammes) sur un plan. Cette solution rend la gestion des trajets plus intuitive et améliore l'expérience de navigation pour les visiteurs.

Pour mener à bien ce projet, plusieurs outils logiciels et ressources en ligne ont été mobilisés. Le développement a principalement été réalisé avec Visual Studio Code (**VSCode**) pour l'édition du code, et WampServer pour simuler un environnement serveur local compatible avec PHP V8.2.18 et MySQL V8.3.0. Afin de préparer les images nécessaires à l'incrustation (suppression de fond, conversion de formats, compression), des sites spécialisés tels que **remove.bg**, **jpg2png.com**, et **compresspng.com** ont été utilisés. Enfin, un appareil photo a permis de capturer les images de base servant de fond pour la création des itinéraires personnalisés.

Voici une proposition de recette avec ses différents scénarios :

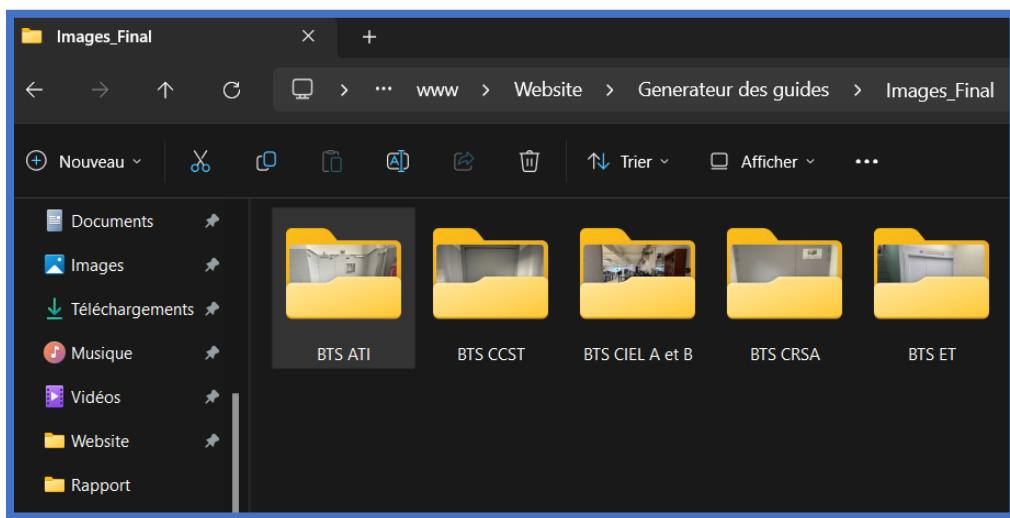
N° de scénario	Description
1	Des photos du parcours ont bien été prises et que les flèches nécessaires ont été téléchargées.
2	Ajout d'une image de fond et d'une flèche en superposé
3	Positionnement de la flèche et génération de l'image finale
4	Tester le fonctionnement du site web

Ce cahier des charges définit les différentes étapes à suivre pour garantir la bonne mise en œuvre des scénarios. Dans cette section, nous allons présenter les parties de code qui composent notre site web, en les associant à des scénarios concrets d'utilisation. Chaque scénario sera illustré par un extrait de code accompagné d'une explication de son rôle dans l'interface. Ces instructions permettent d'obtenir un processus simple et efficace lors de l'intégration des images.

## 1<sup>er</sup> scénario : Les photos du parcours et des flèches ont été mis en œuvre.

Avec l'aide de mes collègues, nous avons pris plusieurs photos des différents parcours de BTS (CIEL, ET, ATI, CRSA, CSST) et créé des dossiers afin d'éviter tout mélange et faciliter l'identification des parcours de BTS. Par ailleurs, la version PHP 8.2.18 impose une limite de taille de fichier photo à 8 Mo, ce que nous avons constaté lors du test d'affichage des images sur le site, exécuté localement via WampServer :

```
[04-Feb-2025 13:58:20 UTC] PHP Warning: POST Content-Length of 14404117 bytes exceeds the limit of 8388608 bytes in Unknown on line 0
```



Donc, j'ai décidé d'utiliser des outils en ligne pour les convertir en PNG, afin de réduire leur poids et compresser les formats. Pour cela, j'ai fait appel à deux sites open source, [jpg2png.com](http://jpg2png.com) et [compresspng.com](http://compresspng.com). Une fois cette opération achevée, il suffit de télécharger des flèches au format image avec un arrière-plan transparent grâce l'outil [remove.bg](http://remove.bg).

Détails	
Type	Fichier PNG
Taille	1,03 Mo
Emplacement d...	C:\wamp64\www\Website\Gene...
Modifié le	21/03/2025 11:40
Dimensions	800 x 800

## 2<sup>ème</sup> scénario : Ajout d'une image de fond et d'une flèche en superposé

```

18     .image-front {
19         position: relative;
20         z-index: 1;
21         width: 800px;
22         height: 800px;
23     }
24     .image-absolute {
25         position: absolute;
26         z-index: 2;
27         top: 130px;
28         left: 600px;
29         width: 180px;
30         height: auto;
31     }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49     <form action="" method="post" enctype="multipart/form-data">
50         <input type="file" name="image" accept="image/*" required>
51         <input type="submit" value="Image de fond">
52     </form>
53
54     <form action="" method="post" enctype="multipart/form-data">
55         <input type="file" name="image2" accept="image/*" required>
56         <input type="submit" value="Image superposée">
57     </form>

```

### Explication :

Les deux formulaires (L49-L57) permettent à l'utilisateur de sélectionner et de télécharger deux images. L'une sera utilisée comme image de fond et l'autre comme image superposée. Les deux formulaires utilisent `enctype="multipart/form-data"` pour gérer les fichiers téléchargés. Lors de la soumission du formulaire, le fichier sera transféré vers le serveur local.

La classe CSS « `.image-front` » (L18) est appliquée à l'image de fond. La propriété « **position relative** ; » permet de placer cette image comme référence de positionnement pour des éléments sans le sortir de sa position d'origine. Le « **z-index: 1;** »(L20) signifie que cette image sera affichée en dessous de l'image superposée (qui a un z-index plus élevé). Enfin, la taille de l'image est fixée à 800x800 pixels, assurant une base stable pour l'incrustation d'autres éléments.

La classe « `.image-absolute` » (L24) est appliquée à l'image superposée. La propriété « **position: absolute;** » permet de positionner l'image librement par rapport à son conteneur parent. Grâce à « **z-index: 2;** », cette image est placée au-dessus de l'image de fond, assurant qu'elle soit bien visible. Les propriétés « `top: 130px;` » et « `left: 600px;` » définissent la position initiale de l'image superposée sur la page. Enfin, « `width: 180px;` » fixe la largeur de l'image superposée, tandis que « `height: auto;` » permet de conserver ses proportions d'origine.

```

73 <?php
74 session_start();
75 $Image = "Images/";
76 if ($_SERVER['REQUEST_METHOD'] == 'POST') {
77     if (isset($_FILES['image1'])) {
78         $Image_Fichier = $Image . basename($_FILES["image1"]["name"]);
79         $_SESSION['image1'] = $Image_Fichier;
80     }
81     if (isset($_FILES['image2'])) {
82         $Image_Fichier2 = $Image . basename($_FILES["image2"]["name"]);
83         $_SESSION['image2'] = $Image_Fichier2;
84     }
85 }
86 if (isset($_SESSION['image1'])) {
87     echo "<img id='image-front' class='image-front' src='{$SESSION['image1']}' alt='Image de fond>";
88 }
89 if (isset($_SESSION['image2'])) {
90     echo "<img id='image-absolute' class='image-absolute' src='{$SESSION['image2']}' alt='Image superposée>";
91 }
92 ?>

```

### Explication :

Cette ligne (L74) est initialisée une session PHP. Elle permet de stocker temporairement des données côté serveur. C'est ce qui permet de retenir les images sélectionnées même après un rechargeement de page.

Cette ligne (L75) est définie ici le dossier de destination où seront stockées les images téléchargées par l'utilisateur. Ce dossier s'appelle simplement "Images/".

Ce morceau de code (L76-L85) vérifie si la méthode HTTP est bien un POST (donc un envoi de formulaire). Ensuite, il regarde si un fichier a été uploadé via les champs « **image1** » ou « **image2** ». Pour chaque image :

- Le chemin du fichier est généré sous la forme « Images/nom\_fichier.png ».

- Ce chemin est ensuite stocké dans la session avec « `$_SESSION['image1']` » ou « `$_SESSION['image2']` ».

Une fois que les chemins des images sont enregistrés en session, ces conditions permettent de les afficher dans la page HTML (L86-L91).

Aux lignes de code (L86-L88) montre si l'image de fond (**image1**) est disponible, elle est affichée avec la classe **.image-front**.

Aux lignes de code (L89-L91) montre si l'image superposée (**image2**) est aussi présente, elle est affichée par-dessus avec **.image-absolute**.

### 3<sup>e</sup> scénario : Positionnement de la flèche et génération de l'image finale

```

66   <div class="controls">
67     <button onclick="moveImage('haut')>Haut</button>
68     <button onclick="moveImage('bas')>Bas</button>
69     <button onclick="moveImage('gauche')>Gauche</button>
70     <button onclick="moveImage('droite')>Droite</button>
71   </div>

96   function moveImage(direction) {
97     const image = document.getElementById('image-absolute');
98     let top = parseInt(image.style.top) || 130;
99     let left = parseInt(image.style.left) || 600;
100
101    switch (direction) {
102      case 'haut': top -= 10; break;
103      case 'bas': top += 10; break;
104      case 'gauche': left -= 10; break;
105      case 'droite': left += 10; break;
106    }
107
108    image.style.top = top + 'px';
109    image.style.left = left + 'px';
110  }

```

#### Explication :

La balise « `<div class="controls">` » contient (L66-L71) quatre boutons, chacun correspondant à une direction : haut, bas, gauche, droite. Chaque bouton (balise html) intègre un attribut « `onclick` » qui représente un événement DOM JavaScript, qui déclenche une fonction JavaScript « `moveImage()` » en lui passant comme paramètre la direction souhaitée. Cela permet à l'utilisateur de déplacer l'image superposée par simples clics successifs.

La fonction `moveImage(direction)` (L96-L110) a pour rôle de modifier dynamiquement la position de l'image superposée dans le navigateur. Elle commence par cibler l'élément HTML correspondant à l'image en superposition, identifié par son ID `image-absolute`. Ensuite, elle récupère ses coordonnées actuelles à l'aide de `parseInt()` sur les propriétés CSS `top` et `left`. Si aucune position initiale n'est définie, les valeurs par défaut sont **130px** et **600px**, comme spécifié dans la feuille de style CSS.

La fonction utilise une structure `switch` pour déterminer quelle opération effectuer en fonction de la direction choisie. À chaque appel, elle incrémente ou décrémente la valeur de `top` ou `left` de 10 pixels, ce qui déplace l'image légèrement dans la direction indiquée. Enfin, ces nouvelles valeurs sont réassignées aux propriétés `style.top` et `style.left` de l'élément, provoquant une mise à jour visuelle instantanée de sa position sur l'écran.

```

112 function downloadMergedImage() {
113     const mainImage = document.getElementById('image-front');
114     const movableImage = document.getElementById('image-absolute');
115     const nomFichier = document.getElementById('nomFichier').value;
116
117     if (!nomFichier) {
118         alert("Veuillez entrer un nom pour votre fichier");
119         return;
120     }
121
122     if (!mainImage || !movableImage) {
123         alert("Veuillez charger les deux images avant de télécharger");
124         return;
125     }
126     let fileName = nomFichier + '.png';
127
128     const img = new Image();
129     img.onload = function () {
130         const canvas = document.createElement('canvas');
131         const ctx = canvas.getContext('2d');
132         canvas.width = 800;
133         canvas.height = 800;
134
135         ctx.drawImage(mainImage, 0, 0, canvas.width, canvas.height);

```

### Explication :

La ligne 112, c'est une déclaration d'une fonction nommée **downloadMergedImage**.

Aux lignes 113 à 115, On utilise **document.getElementById()** pour récupérer trois éléments HTML : **mainImage** représente l'image de fond, **movableImage** est l'image superposée et **nomFichier** correspond au texte saisi par l'utilisateur pour nommer le fichier final à générer.

Aux lignes 117-120, Cette condition vérifie si le champ de nom est vide. Si c'est le cas, une alerte est affichée à l'utilisateur grâce **alert()**. Cette deuxième condition vérifie si les deux images sont bien présentes dans le DOM. Si l'une ou l'autre est absente, une alerte s'affiche, et l'action est annulée pour éviter des erreurs.

On va déclarer une variable **fileName** contenant le **nom final du fichier**, en ajoutant l'extension **.png** au nom personnalisé saisi par l'utilisateur. À la ligne 128, On va instancier un objet **Image()** vide. Cela va servir à attendre le chargement complet de l'image de fond avant d'exécuter le reste. À la ligne 129, on définit une fonction de rappel (callback) à exécuter quand l'image **img** est entièrement chargée (événement **onload**). À la ligne 130, On crée dynamiquement un élément **<canvas>** qui servira de support pour dessiner et fusionner les images dans un espace graphique en 2D directement dans le navigateur. Aux lignes 131-133, On récupère le contexte de dessin avec **getContext('2d')**, et on fixe une résolution de 800x800 pixels, ce qui correspond à la taille des images utilisées. À la ligne 135, on dessine l'image de fond (**mainImage**) sur toute la surface du canvas, à partir du coin supérieur gauche (0, 0), en l'ajustant aux dimensions complètes du canvas.

```

137 const rect = movableImage.getBoundingClientRect();
138 const mainRect = mainImage.getBoundingClientRect();
139 const relativeLeft = rect.left - mainRect.left;
140 const relativeTop = rect.top - mainRect.top;
141
142 ctx.drawImage(movableImage, relativeLeft, relativeTop, movableImage.width, movableImage.height);

```

Aux lignes 137-138, On récupère les coordonnées absolues dans la page (en pixels) des deux images, grâce à **getBoundingClientRect()**. Cela retourne un objet avec les dimensions et la position réelle d'un élément affiché à l'écran.

Aux lignes 139-140, on fait un calcul de la position relative de l'image superposée par rapport à l'image de fond. Cela permet de savoir où dessiner la deuxième image sur le canvas pour qu'elle soit parfaitement alignée avec ce que l'utilisateur voit à l'écran.

Voici une illustration pour mieux visualiser ces positions :

**Écran (viewport)**

top: 130px  
left: 100px

image de fond

image sup

left: 600px

Image de fond (**mainImage**) en position CSS :  
**left = 100px**  
**top = 80px**  
Donc son coin supérieur gauche est situé à (100, 80) **par rapport à la fenêtre**.

Image superposée (**movableImage**) position CSS :  
**left = 600px**  
**top = 130px**  
Son coin supérieur gauche est à (600, 130) par rapport à la fenêtre.

```

relativeLeft = rect.left - mainRect.left;
500          =   600      -     100

```

L'image superposée est 500 pixels à droite de l'image de fond.

```

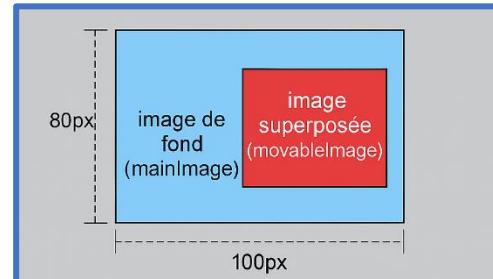
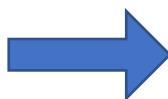
relativeTop = rect.top - mainRect.top;
50          =   130      -     80

```

L'image superposée est 50 pixels plus bas que le coin supérieur de l'image de fond.

A la ligne 142, On dessine maintenant l'image superposée (flèche), au bon endroit et avec la bonne taille. Elle sera fusionnée sur le canvas par-dessus l'image de fond, dans sa position actuelle à l'écran.

Voici un canevas final en exemple :  
Une simulation du rendu fusionné



Enfin, voici la dernière partie de code :

```
144     const link = document.createElement('a');
145     link.download = fileName;
146     link.href = canvas.toDataURL('image/png');
147     document.body.appendChild(link);
148     link.click();
149     document.body.removeChild(link);
150   };
151   img.src = mainImage.src;
152 }
153 </script>
```

Aux lignes 144-146, On prépare un élément **<a>** (lien HTML) en lui attribuant :

- Une valeur **download** : ce sera le nom du fichier une fois téléchargé ;
- Un lien de téléchargement href, obtenu à partir du canvas converti en image au format PNG (**toDataURL()**).

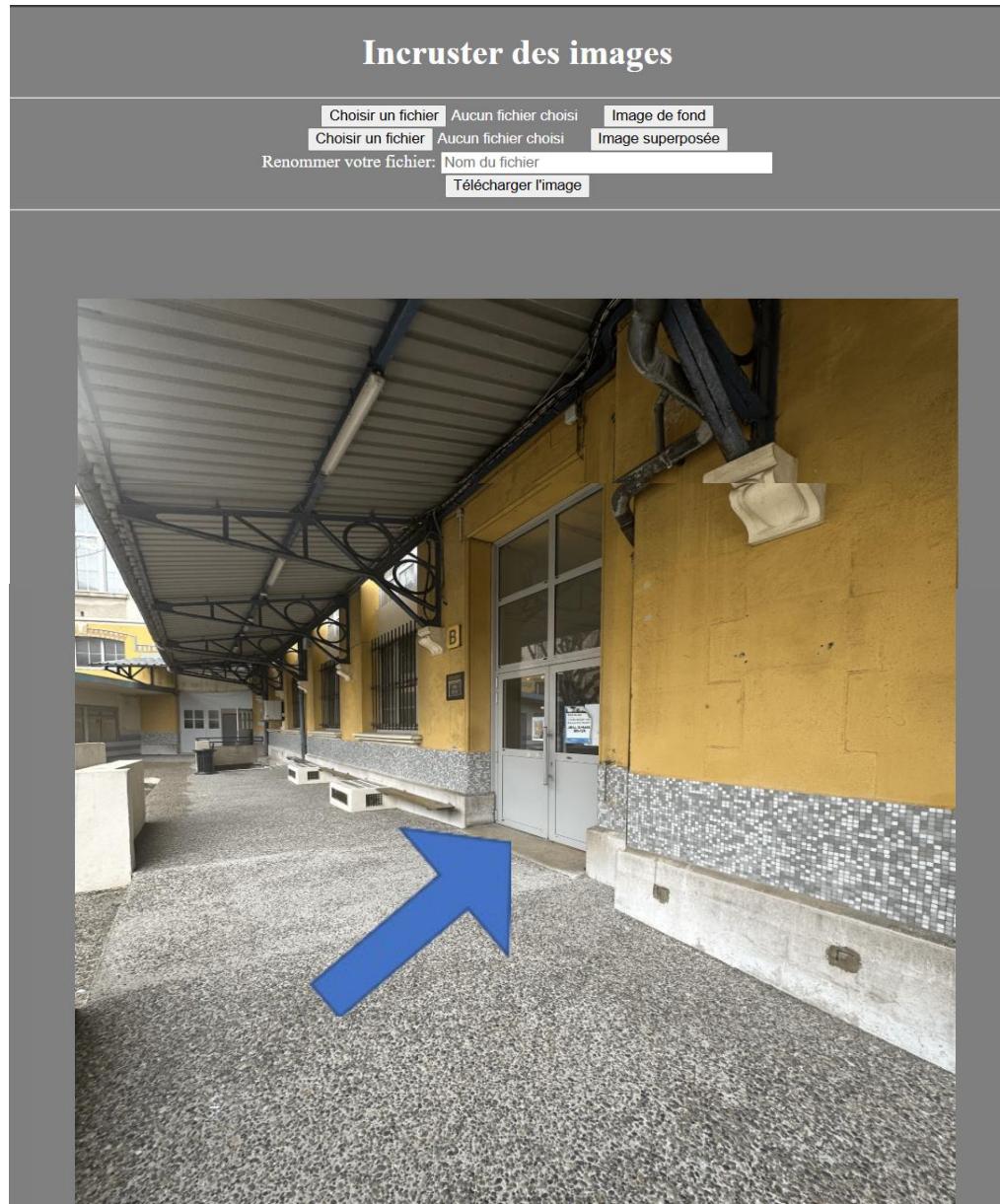
Aux lignes 147-149, on ajoute le lien est ajouté temporairement au DOM, cliqué automatiquement avec **link.click()** pour lancer le téléchargement, puis supprimé immédiatement pour ne pas encombrer la page.

A la ligne 151, On lance le chargement de l'image **img** en lui assignant le même **src** que l'image de fond. Cela déclenche automatiquement le **onload**, une fois que l'image est disponible, et le processus commence.

Enfin, on peut passer à l'étape supérieure, c'est-à-dire insérer des images du parcours du BTS tout en ajoutant des flèches superposées pour indiquer la direction. Si vous souhaitez plus de détails sur mon code, veuillez-vous référer à **l'annexe**.

## 4<sup>e</sup> scénario : Tester le fonctionnement du site web

Suite à différentes séries de tests effectués en local, accompagnées de corrections des erreurs identifiées, le site web fonctionne à présent de façon stable et répond pleinement aux objectifs attendus. L'image présentée ci-dessous met en évidence le résultat final, après l'ajout et le placement précis des éléments visuels superposés. J'ai ensuite transmis à Tony Barelli l'ensemble du dossier contenant les images déjà incrustées pour chaque spécialité de BTS, afin qu'il puisse développer le simulateur de déplacement spécifique à chaque formation.



## XIII. Implémentation de l'envoi d'e-mails

### Recherches et récupération de ressources

Dans le cadre de ce projet, je me suis intéressé à plusieurs options permettant l'envoi d'e-mails via PHP, dans l'objectif d'ajouter une fonctionnalité d'abonnement à une newsletter, ou de proposer l'envoi de formations de BTS au format PDF. Au fil de mes recherches, j'ai découvert une bibliothèque open source dédiée au mailing, accessible sur GitHub. Celle-ci propose un ensemble d'outils robustes et efficaces pour gérer l'envoi de messages électroniques.

### Compréhension de la logique d'envoi

Afin de mieux comprendre son fonctionnement, j'ai téléchargé les codes sources d'exemples fournis avec la librairie. Ces exemples m'ont permis d'analyser la structure du code, notamment la manière dont les données issues d'un formulaire HTML sont traitées, puis envoyées sous forme d'e-mail via le script PHP.

### Intégration dans mon projet

J'ai ensuite entrepris une phase d'expérimentation en local. Pour cela, j'ai essayé d'intégrer une partie de cette logique dans un fichier nommé **merci.php**, qui a été initialement développé par Tony Barelli. Mon objectif était de reproduire un envoi de mail fonctionnel en partant d'un formulaire HTML simple.

### Mise en place de l'environnement de test

L'environnement de test se compose :

- D'un formulaire HTML basique permettant à l'utilisateur de saisir ses informations (nom, prénom, e-mail) (**voir en annexes**).
- D'un script PHP de traitement utilisant la librairie récupérée pour générer et envoyer l'e-mail (**voir en annexes**).

Dans cette partie, nous allons examiner en profondeur le mécanisme d'envoi d'e-mails construit à l'aide de la bibliothèque **PHPMailer**. Le code source est décomposé en plusieurs parties, chacune étant accompagnée d'une explication technique mettant en évidence sa fonction spécifique dans le processus d'envoi. Cette approche permet de mieux comprendre le rôle de chaque composant dans l'architecture globale de l'interface.

## Chargement de la bibliothèque PHPMailer et traiter des données du formulaire

```

91      <?php
92      use PHPMailer\PHPMailer\PHPMailer;
93      use PHPMailer\PHPMailer\SMTP;
94      use PHPMailer\PHPMailer\Exception;
95      require './PHPMailer/src/Exception.php';
96      require './PHPMailer/src/PHPMailer.php';
97      require './PHPMailer/src/SMTP.php';
98
99      if ($_SERVER['REQUEST_METHOD'] === 'POST') {
100          $nom = htmlspecialchars($_POST['Nom']);
101          $prenom = htmlspecialchars($_POST['Prenom']);
102          $email = htmlspecialchars($_POST['Email']);
103
104          $filiere_array = [];
105          if (isset($_POST['CIEL'])) $filiere_array[] = 'CIEL';
106          if (isset($_POST['ATI'])) $filiere_array[] = 'ATI';
107          if (isset($_POST['CCST'])) $filiere_array[] = 'CCST';
108          if (isset($_POST['CRSA'])) $filiere_array[] = 'CRSA';
109          if (isset($_POST['ET'])) $filiere_array[] = 'ET';
110          if (isset($_POST['Pour_tous'])) $filiere_array[] = 'Pour tous';
111      }
112

```

### Explication :

Les lignes 92 à 94 déclarent l'utilisation des classes de la bibliothèque **PHPMailer**, ce qui permet d'accéder aux fonctionnalités d'envoi d'e-mail (**PHPMailer**), de protocole SMTP (**SMTP**), et de gestion des erreurs (**Exception**).

Les lignes (L95-L97) incluent les fichiers sources de **PHPMailer**, nécessaires pour que les classes mentionnées ci-dessus puissent fonctionner. C'est une manière d'intégrer le composant externe dans le script PHP.

À la ligne 99, cette ligne vérifie si le formulaire a bien été soumis avec la méthode POST. Aux lignes 100-102, Ces lignes récupèrent les valeurs envoyées par le formulaire (**Nom**, **Prenom**, **Email**) tout en les protégeant contre les injections HTML grâce à **htmlspecialchars()**.

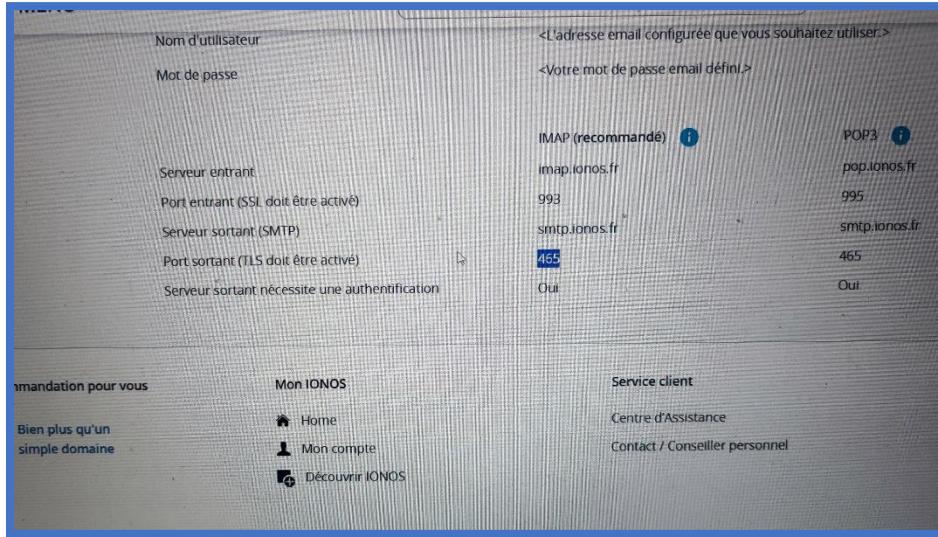
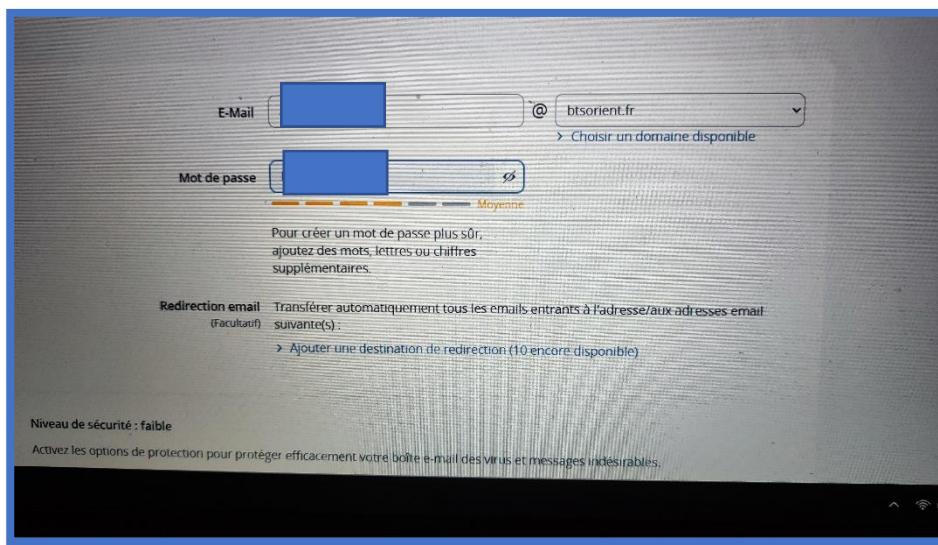
Les lignes 104-110, le script vérifie si certaines cases à cocher ont été sélectionnées dans le formulaire, correspondant aux différents BTS. Chaque choix est ajouté dans un tableau **\$filiere\_array**.

A la ligne 111, Les filières sélectionnées sont ensuite converties en une chaîne de texte séparée par des virgules, pour faciliter l'envoi dans le mail.

## Configuration du serveur SMTP via IONOS et générer l'envoi de mails en PHP

Avant de procéder à la configuration du serveur SMTP, il est nécessaire de se rendre sur le site du fournisseur de services IONOS. Une fois sur la plateforme, la première étape consiste à créer une boîte mail professionnelle. Cela permet d'obtenir une adresse e-mail fonctionnelle, tout en récupérant les informations essentielles à la configuration : le numéro du port à utiliser, ainsi que le protocole de sécurité recommandé pour l'envoi des messages.

Les captures photos ci-dessous illustrent les différentes étapes de la configuration que j'ai suivie sur l'interface IONOS :



```

113     function envoie_mail($from_name, $from_email, $subject, $nom, $prenom, $to_email, $filiere) {
114         $mail = new PHPMailer();
115         $mail->isSMTP();
116         $mail->SMTPDebug = 0;
117         $mail->SMTPSecure = 'tls';
118         $mail->Host = 'smtp.ionos.fr';
119         $mail->SMTPAuth = true;
120         $mail->Username = 'lycee-lgt.rempart@btsorient.fr';
121         $mail->Password = 'XXXXXXXXXX';
122         $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
123         $mail->Port = 465;
124
125         $mail->setFrom('lycee-lgt.rempart@btsorient.fr', 'Lycee-Rempart-Vinci');
126         $mail->addAddress($to_email, $prenom . ' ' . $nom);
127         $mail->isHTML(true);
128         $mail->Subject = "Voici le programme du BTS que vous avez choisi : " . $filiere;
129         $mail->Body = "Bonjour " . $prenom . " " . $nom . ", voici le programme du BTS que vous avez choisi : " . $filiere;
130         $mail->setLanguage('fr', '/optional/path/to/language/directory/');
131
132         $formations = explode(' ', $filiere);
133         for ($i = 0; $i < count($formations); $i++) {
134             $mail->addAttachment(__DIR__ . '/Plaquette_BTS/plaquette_' . $formations[$i] . '.pdf');
135         }
    
```

### Explication :

À la ligne 113, c'est une déclaration d'une fonction nommée **envoie\_mail**.

À la ligne 114, on instancie un nouvel objet **PHPMailer**, en précisant qu'on utilise le protocole **SMTP** (protocole standard pour envoyer des mails).

Les lignes 116-123, On configure la connexion au serveur SMTP de l'hébergeur IONOS :

- **SMTPDebug = 0** désactive l'affichage de débogage.
- **SMTPAuth = true** active l'authentification.
- **Username** et **Password** sont les identifiants du compte qui envoie le mail.
- **Host** est le nom du serveur SMTP (ici, celui de IONOS).
- **Port = 465** est le port sécurisé avec le protocole TLS

Dans les lignes 125-126, Le mail est envoyé depuis l'adresse du lycée et est destiné à l'adresse de l'utilisateur.

Dans les lignes 127-129, l'e-mail est au format HTML, avec un objet personnalisé (le nom des BTS sélectionnés) et un corps de message simple.

À la ligne 130, Cela définit la langue des messages d'erreur en français (utile pour le retour utilisateur).

Cette ligne 132 permet de transformer une chaîne de texte (par exemple : "**CIEL, ATI, ET**") en un tableau contenant chaque élément séparé. Ex : **\$formations = ["CIEL", "ATI", "CRSA"];**

Les lignes 133-135, Pour chaque filière sélectionnée, un fichier PDF correspondant est joint dynamiquement à l'e-mail. Le fichier doit se trouver dans le dossier **/Plaquette\_BTS/** et avoir un nom de type **plaquette\_CIEL.pdf**, **plaquette\_ATI.pdf**, etc.

## Envoi du mail

```

136     return $mail->send();
137 }
138 if (envoie_mail($nom, $email, "Programme BTS", $nom, $prenom, $email, $filiere)) {
139     echo '<h2>Email envoyé avec succès !</h2>';
140 } else {
141     echo '<h2>Une erreur s\'est produite lors de l\'envoi du mail</h2>';
142 }
143 ?>
144

```

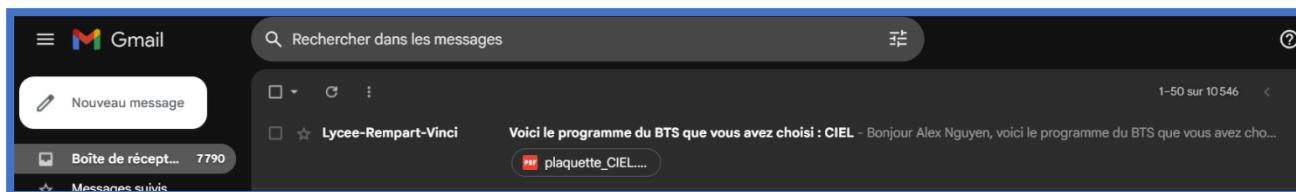
### Explication :

À la ligne, On exécute la fonction **send()** de **PHPMailer** pour envoyer le message. La fonction renvoie **TRUE** ou **FALSE**.

Enfin, après l'appel à **send()**, un message est affiché à l'utilisateur :

- Si l'envoi a réussi : « **Email envoyé avec succès** »,
- Sinon : « **Une erreur s'est produite lors de l'envoi du mail** ».

Voici ci-dessous les captures d'écran montrant le succès de l'opération :



**Merci, Alex Nguyen, pour votre inscription !**

Nous avons bien reçu vos informations.

**Merci !**

**Nous vous contacterons très bientôt !**

Pour plus d'information, vous avez à votre disposition [le site](#) de notre établissement. Merci pour

[Site de l'établissement](#)

- [BTS ATI](#)
- [BTS CIEL](#)
- [BTS CCST](#)
- [BTS CRSAT](#)
- [BTS ET](#)

**Lycée Polyvalent Rempart-Vinci**

8 Rue du Rempart  
04 91 14 01 40

Email envoyé avec succès !

**Voici le programme du BTS que vous avez choisi : CIEL**

**Lycee-Rempart-Vinci <lycee-lgt.rempart@btsoorient.fr>**  
À moi ▾

**Traduire en français** X

Bonjour Alex Nguyen, voici le programme du BTS que vous avez choisi : CIEL

**1 pièce jointe • Analyse effectuée par Gmail** ⓘ

Un autre test consiste à sélectionner toutes les formations BTS ci-dessous :



Une fois la tâche liée à l'envoi des mails terminée, Tony Barelli s'est chargé de l'optimisation responsive du site. Il a stylisé l'ensemble à l'aide de CSS tout en conservant le framework Bootstrap. Ci-dessous, le résultat des pages **contact.html** et **merci.php** :

**Contact**

Si vous êtes intéressé par notre établissement et une de ses filières, veuillez compléter ce formulaire ci-dessous.

**Informations personnelles**

Nom

Prénom

E-Mail

Téléphone

Adresse



## XIV. Sécurisation de l'accès par authentification administrateur

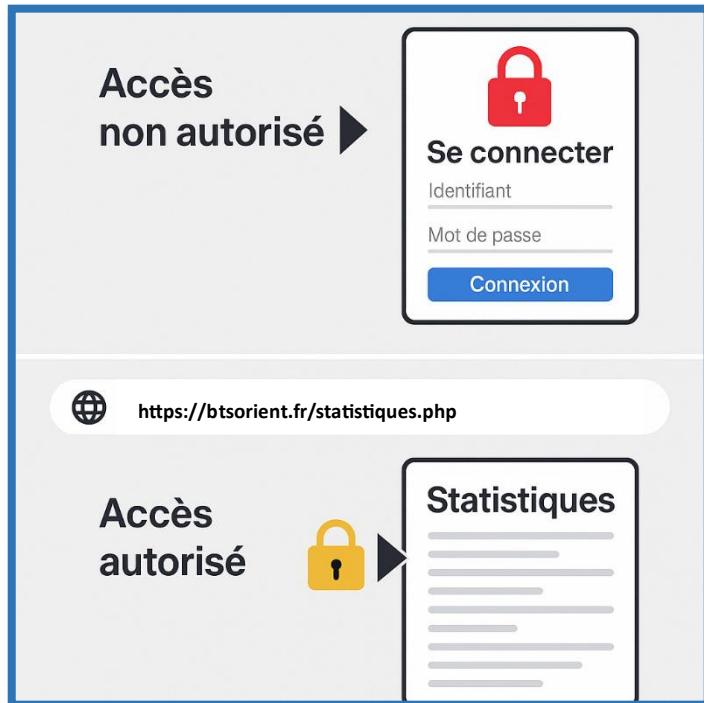
Dans le cadre de la sécurisation du site, j'ai collaboré avec mon coéquipier Evan Bonnal pour empêcher tout accès non autorisé à l'espace administrateur, même en cas de modification manuelle de l'URL.

Lors de nos recherches, nous avons rencontré de nombreuses difficultés pour implémenter un système fiable de contrôle d'accès. Pendant plusieurs semaines, nous avons testé différentes méthodes sans résultat concluant.

Ce n'est qu'à la fin du mois de mars que nous avons trouvé une solution fonctionnelle, grâce à l'aide de notre camarade de classe Rida BENAMAR. Il nous a présenté une partie de son propre code d'authentification, ce qui nous a permis de comprendre le bon fonctionnement de la gestion des sessions et de la redirection conditionnelle. Inspiré par cette démonstration, j'ai ensuite adapté et intégré ce système dans notre projet.

Pour cela, nous avons mis en place un contrôle d'accès basé sur la session PHP. Ce mécanisme permet de vérifier si l'utilisateur est bien authentifié en tant qu'administrateur avant d'accéder aux pages protégées. Si ce n'est pas le cas, l'utilisateur est automatiquement redirigé vers la page de connexion.

Voici l'illustration de mise en place du contrôle d'accès administrateur en simulation :



### Avant la mise en place du contrôle :

L'utilisateur peut taper  
<https://btsorient.fr/statistiques.php>

Et accéder à la page sans avoir saisi aucun identifiant ✗

### Après avoir ajouté la vérification PHP :

Si la session `$_SESSION['login']` n'est pas active :

➡ L'utilisateur est automatiquement redirigé vers `login.php`

✓ Sécurité renforcée, l'accès est protégé ✓

## Contrôle d'accès par session en PHP

Ce bloc PHP est placé en haut de chaque page protégée, comme **statistiques.php** ou une page d'administration. Il permet de vérifier qu'un utilisateur est connecté avant d'autoriser l'accès.

```
1  <?php
2  session_start();
3
4  // Vérifie si l'utilisateur est connecté
5  if (!isset($_SESSION['login']) || empty($_SESSION['login'])) {
6      // Redirection vers la page de connexion s'il n'est pas authentifié
7      header("Location: login.php");
8      exit;
9  }
10 ?>
```

### Explication :

- A la ligne 2, **session\_start()** : initialise ou récupère la session active de l'utilisateur.
- A la ligne 5, cette condition vérifie si l'identifiant de session est bien défini. Cela signifie que l'utilisateur est connecté.
- Les lignes 7-8, si l'utilisateur n'est pas connecté, il est automatiquement redirigé vers la page de connexion (**login.php**) pour bloquer tout accès direct via l'URL.
- En cas de connexion valide, l'utilisateur peut continuer à naviguer sur les pages réservées

## Ajout d'un bouton de déconnexion

Afin de compléter le système d'authentification administrateur, nous avons également ajouté un bouton de déconnexion, indispensable pour terminer proprement une session. Ce bouton permet à l'administrateur de se déconnecter en toute sécurité et d'être redirigé automatiquement vers la page de connexion. Cela permet qu'aucun accès non autorisé ne persiste après une session.

### Fonctionnement technique

Lorsqu'un utilisateur clique sur le bouton de déconnexion, un script PHP est exécuté pour supprimer toutes les données de session, détruire la session en cours, puis rediriger l'utilisateur vers la page de connexion (**login.php**).

```
1  <?php
2
3  // Démarrer la session
4  session_start();
5
6  // Détruire toutes les variables de session
7  $_SESSION = array();
8
9  // Détruire la session
10 session_destroy();
11
12 // Rediriger vers la page de connexion
13 header("Location: login.php");
14 exit;
15 ?>
```

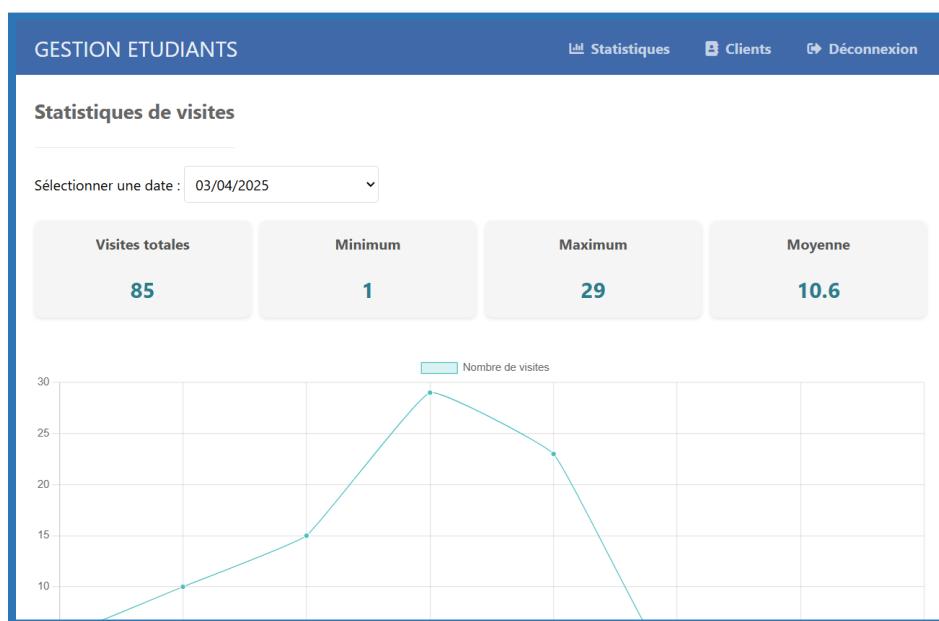
Ce mécanisme renforce la sécurité de l'espace administrateur, en s'assurant que la session est proprement fermée et qu'il n'est plus possible d'accéder aux pages protégées sans authentification.

## Validation du fonctionnement du contrôle d'accès et de la déconnexion

Après avoir effectué plusieurs séries de tests, nous avons pu confirmer le bon fonctionnement à la fois du contrôle d'accès et du système de déconnexion. Les fonctionnalités mises en place répondent parfaitement aux attentes : l'accès aux pages sécurisées est désormais strictement réservé aux utilisateurs administrateurs dûment authentifiés, tandis que la déconnexion permet de mettre fin à la session de manière propre et sécurisée. Les objectifs fixés ont donc été atteints avec succès.

Ci-dessous, les captures des pages login et statistiques illustrent le bon fonctionnement du dispositif mis en place :

The screenshot shows a login form with a logo for 'LPO Rempart - SEP Vinci' at the top. Below the logo, the word 'Connexion' is centered. The form contains two input fields: the first is filled with 'admin' and the second is filled with '.....'. At the bottom is a large blue 'Connexion' button.



## XV. Problème rencontré

### Incrustation des images

Au cours du projet, l'une des principales difficultés que j'ai rencontrées concernait la fusion de deux images, notamment en essayant d'utiliser la fonction **imagecopymerge** en PHP. Bien que cette fonction permette une superposition, elle n'était pas adaptée à notre besoin: l'image à superposer conservait son arrière-plan, ce qui empêchait une intégration visuellement propre.

Pour contourner ce problème, j'ai exploré diverses solutions sur des forums spécialisés et des sites de documentation. J'ai notamment testé **l'API de Remove.bg**, qui est conçue pour supprimer automatiquement les arrière-plans. Bien que cette solution ait donné quelques résultats satisfaisants, elle n'était pas toujours fiable, en raison de limitations techniques ou de la complexité de certaines images.

Devant ces résultats mitigés, j'ai finalement choisi d'adopter une approche alternative en JavaScript. Après plusieurs essais et ajustements, j'ai réussi à mettre en place une solution fonctionnelle. Malgré quelques imprécisions au départ, notamment liées au calcul du décalage en pixels pour positionner correctement l'image superposée, j'ai pu affiner le rendu et obtenir une incrustation fluide et contrôlée, directement dans l'interface du navigateur.

### Implémentation de l'envoi d'e-mails

Comparée à la tâche précédente liée à l'incrustation des images, l'implémentation de l'envoi d'e-mails s'est révélée plus abordable. Après plusieurs semaines de travail, j'ai commis moins d'erreurs, notamment grâce à une lecture attentive de la documentation officielle sur le GitHub de **PHPMailer**. Cette ressource, accompagnée de nombreux exemples de code source, m'a grandement facilité la compréhension du fonctionnement de la bibliothèque.

Concernant l'ajout de pièces jointes, comme les plaquettes au format PDF, la démarche était relativement simple : il suffisait de convertir la sélection de formations en tableau, puis de parcourir ce tableau à l'aide d'une boucle conditionnelle. Cela m'a permis de joindre automatiquement les bons fichiers PDF en fonction des choix de l'utilisateur et de finaliser l'envoi de manière fluide et personnalisée.

## XVI. Mise en commun

Travailler en équipe a vraiment fait toute la différence pour réussir le projet borne d'orientation. Chacun a apporté ses compétences spécifiques pour atteindre nos objectifs communs.

**Matthieu CAVIGGIA** a pris en charge la partie liée aux QR codes, permettant de sélectionner une formation via JavaScript. Il a ensuite conçu le site web en veillant à son adaptabilité sur tous les types d'écrans grâce au framework Bootstrap (version 4). Il a également créé une page d'accueil accueillante, souhaitant la bienvenue aux visiteurs à l'occasion de la journée portes ouvertes de notre établissement, le lycée Rempart. Enfin, il a procédé à l'installation d'un Raspberry Pi 4 équipé d'un écran tactile, monté sur support, pour afficher le site localement sur la borne.

**Tony BARELLI**, de son côté, a développé le système de déplacement interactif permettant de naviguer entre les différentes formations de BTS. Ce module était étroitement lié à ma partie, consacrée à l'incrustation d'images. Il a conçu une base de données contenant les images préalablement superposées à l'aide de symboles, afin d'afficher dynamiquement chaque formation. Il s'est chargé de créer une page de formulaire de contact stylisée en CSS et parfaitement responsive. Toutes les données envoyées via ce formulaire sont automatiquement enregistrées dans la base de données.

**Evan BONNAL** s'est occupé de la partie authentification de l'administrateur du site. Il a veillé à rendre cette section responsive grâce à Bootstrap 4. Par la suite, il a développé une page de statistiques affichant le nombre de visiteurs par heure, avec une visualisation sous forme de courbe en utilisant la bibliothèque Chart.js. Chaque fois qu'un utilisateur scanne un QR code, une valeur est incrémentée dans les statistiques horaires.

Au final, chacun de nous a mené à bien les tâches qui lui étaient attribuées. Après avoir effectué plusieurs tests du simulateur, j'ai procédé au déploiement du site <https://btsoorient.fr> via le service IONOS, après configuration du serveur. Avec mon équipe, nous avons ensuite présenté le projet devant le directeur de l'établissement, le directeur adjoint, ainsi que des représentants du département SAMSAR TSA84 (Vaucluse) de l'association « La Bourguette ». Des photos de cet événement, prises par un étudiant de notre classe, sont disponibles en **annexe**.

## XVII. Conclusion

Ce projet de borne d'orientation a été une expérience très enrichissante, autant sur le plan technique que personnel. Travailler en équipe nous a permis de réunir nos compétences pour créer une solution complète, fonctionnelle et conforme aux objectifs fixés. Chaque membre du groupe a contribué de manière active, que ce soit pour la création du site web, l'ajout d'images interactives, le développement des fonctionnalités ou la mise en place du système de connexion et des statistiques.

Sur le plan personnel, ce projet m'a permis de renforcer mes compétences en développement web, tant côté client que serveur. J'ai été confronté à des défis techniques variés comme la gestion fine de la superposition d'images ou l'envoi d'e-mails avec pièces jointes que j'ai su relever grâce à une combinaison d'efforts personnels, de recherches, de tests répétés, et de l'entraide au sein du groupe.

J'ai également appris à mieux m'organiser, à faire preuve de persévérance face aux bugs et à m'adapter aux imprévus. Ce projet m'a permis de comprendre l'importance d'un code bien structuré, de la documentation claire, et du respect des bonnes pratiques en matière de sécurité, notamment pour la gestion des sessions et le contrôle d'accès.

Même si tout n'est pas encore finalisé, notamment l'intégration d'un lecteur RFID, cette amélioration permettra de sécuriser davantage la borne. En cas de blocage ou de mauvaise utilisation, un badge RFID pourra être utilisé pour la déverrouiller rapidement et en toute sécurité.

Je remercie mon professeur, Monsieur Narelli, d'avoir contribué à la réalisation de mon projet. Je remercie également mes coéquipiers, Matthieu CAVIGGIA, Tony BARELLI et Evan BONNAL, pour leur contribution.

Ce projet pourra aussi servir dans d'autres domaines à l'avenir, comme les musées, les salons professionnels ou les hôpitaux. Il suffira d'adapter les trajets et les contenus affichés selon les besoins de chaque lieu. Cela permettrait d'offrir une aide à l'orientation simple et efficace pour différents types de visiteurs.

En résumé, ce travail d'équipe m'a non seulement permis de mettre en pratique mes connaissances techniques dans un contexte concret, mais aussi de développer des compétences essentielles à toute collaboration professionnelle. Il représente une belle réussite collective, et m'a apporté une réelle valeur ajoutée pour la suite de mon parcours.

# XVIII. Annexes

## Incrustation d'Images : Fonctionnement du Code

- 1 Section HTML :

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Télécharger une image</title>
```

- 2 Section CSS :

```
<style>
    body {
        text-align: center;
        background-color: #808080;
        color: #FFFFFF;
    }
    img {
        margin-top: 80px;
        width: 800px;
        height: 800px;
    }
    .image-front {
        position: relative;
        z-index: 1;
        width: 800px;
        height: 800px;
    }
    .image-absolute {
        position: absolute;
        z-index: 2;
        top: 130px;
        left: 600px;
        width: 180px;
        height: auto;
    }
    .controls {
        position: fixed;
        right: 20px;
```

```
        top: 50%;  
        display: flex;  
        flex-direction: column;  
    }  
.controls button {  
    padding: 10px;  
    font-size: 16px;  
    cursor: pointer;  
}  
</style>  
</head>  
<body>  
    <h1>Incruster des images</h1>  
    <fieldset>  
        <form action="" method="post" enctype="multipart/form-data">  
            <input type="file" name="image1" accept="image/*" required>  
            <input type="submit" value="Image de fond">  
        </form>  
  
        <form action="" method="post" enctype="multipart/form-data">  
            <input type="file" name="image2" accept="image/*" required>  
            <input type="submit" value="Image superposée">  
        </form>  
        <div>  
            <form method="post" id="downloadForm">  
                <label>Renommer votre fichier:</label>  
                <input type="text" name="nomFichier" id="nomFichier" size="40" maxlength="25"  
placeholder="Nom du fichier" required /><br />  
                <button type="button" onclick="downloadMergedImage()">Télécharger l'image</button>  
            </form>  
        </div>  
        <div class="controls">  
            <button onclick="moveImage('haut')">Haut</button>  
            <button onclick="moveImage('bas')">Bas</button>  
            <button onclick="moveImage('gauche')">Gauche</button>  
            <button onclick="moveImage('droite')">Droite</button>  
        </div>  
    </fieldset>  
</body>
```

- **3 Section PHP :**

```
<?php
session_start();
$image = "Images/";
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    if (isset($_FILES['image1'])) {
        $image_Fichier = $image . basename($_FILES["image1"]["name"]);
        $_SESSION['image1'] = $image_Fichier;
    }
    if (isset($_FILES['image2'])) {
        $image_Fichier2 = $image . basename($_FILES["image2"]["name"]);
        $_SESSION['image2'] = $image_Fichier2;
    }
}
if (isset($_SESSION['image1'])) {
    echo "<img id='image-front' class='image-front' src='{$SESSION['image1']}' alt='Image de fond'>";
}
if (isset($_SESSION['image2'])) {
    echo "<img id='image-absolute' class='image-absolute' src='{$SESSION['image2']}' alt='Image superposée'>";
}
?>
</body>
</html>
```

- **4 Section JavaScript :**

```
<script>
function moveImage(direction) {
    const image = document.getElementById('image-absolute');
    let top = parseInt(image.style.top) || 130;
    let left = parseInt(image.style.left) || 600;

    switch (direction) {
        case 'haut': top -= 10; break;
        case 'bas': top += 10; break;
        case 'gauche': left -= 10; break;
        case 'droite': left += 10; break;
    }

    image.style.top = top + 'px';
    image.style.left = left + 'px';
}
```

```
function downloadMergedImage() {
    const mainImage = document.getElementById('image-front');
    const movableImage = document.getElementById('image-absolute');
    const nomFichier = document.getElementById('nomFichier').value;

    if (!nomFichier) {
        alert("Veuillez entrer un nom pour votre fichier");
        return;
    }

    if (!mainImage || !movableImage) {
        alert("Veuillez charger les deux images avant de télécharger");
        return;
    }
    let fileName = nomFichier + '.png';

    const img = new Image();
    img.onload = function () {
        const canvas = document.createElement('canvas');
        const ctx = canvas.getContext('2d');
        canvas.width = 800;
        canvas.height = 800;

        ctx.drawImage(mainImage, 0, 0, canvas.width, canvas.height);

        const rect = movableImage.getBoundingClientRect();
        const mainRect = mainImage.getBoundingClientRect();
        const relativeLeft = rect.left - mainRect.left;
        const relativeTop = rect.top - mainRect.top;

        ctx.drawImage(movableImage, relativeLeft, relativeTop, movableImage.width,
        movableImage.height);

        const link = document.createElement('a');
        link.download = fileName;
        link.href = canvas.toDataURL('image/png');
        document.body.appendChild(link);
        link.click();
        document.body.removeChild(link);
    };
    img.src = mainImage.src;
}
</script>
```

**Page HTML (contact.html) :**

Si vous êtes intéressé par notre établissement et une de ses filières, veuillez compléter ce formulaire ci-dessous.

**Informations personnelles**Nom Prénom E-Mail Téléphone Adresse Département  ▾Code postal **Informations générales**

Quelle(s) filière(s) vous intéresse ?

- BTS CIEL
- BTS ET
- BTS ATI
- BTS CCST
- BTS CRSA
- Pour tous

### Script PHPMailer d'exemple : code source

```
1  <?php
2  use PHPMailer\PHPMailer\PHPMailer;
3  use PHPMailer\PHPMailer\SMTP;
4  use PHPMailer\PHPMailer\Exception;
5  require './PHPMailer/src/Exception.php';
6  require './PHPMailer/src/PHPMailer.php';
7  require './PHPMailer/src/SMTP.php';
8  //Create an instance; passing `true` enables exceptions
9
10 function envoie_mail($from_name,$from_email,$subject,$message){
11     $mail = new PHPMailer();
12     $mail->isSMTP();
13     $mail->SMTPDebug = 0;
14     $mail->SMTPSecure = 'ssl';
15     $mail->Host      = 'smtp.gmail.com';
16     $mail->SMTPAuth   = true;
17     $mail->Username   = 'johnazomahou@gmail.com';           //SMTP username
18     $mail->Password   = 'tonmotdepasse';                   //SMTP password
19     $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;        //Enable implicit TLS end
20     $mail->Port       = 465;
21     $mail->setFrom($from_email, $from_name);
22     $mail->addAddress('johnazomahou@gmail.com','AthenaTech');
23     $mail->isHTML(true);
24     $mail->Subject = $subject;
25     $mail->Body    = $message;
26     $mail->setLanguage('fr', '/optional/path/to/language/directory/');
27     if (!$mail->Send()) {
28         return false;
29     }
30     else {
31         return true;
32     }
33 }
34 if (envoie_mail($_POST['name'],$_POST['email'],$_POST['subject'],$_POST['message'])) {
35     echo 'OK';
36 }
37 }
```

## Implémentation de l'envoi d'e-mails en PHP : Fonctionnement du Code

```

<?php

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;
require './PHPMailer/src/Exception.php';
require './PHPMailer/src/PHPMailer.php';
require './PHPMailer/src/SMTP.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $nom = htmlspecialchars($_POST['Nom']);
    $prenom = htmlspecialchars($_POST['Prenom']);
    $email = htmlspecialchars($_POST['Email']);

    $filiere_array = [];
    if (isset($_POST['CIEL'])) $filiere_array[] = 'CIEL';
    if (isset($_POST['ATI'])) $filiere_array[] = 'ATI';
    if (isset($_POST['CCST'])) $filiere_array[] = 'CCST';
    if (isset($_POST['CRSA'])) $filiere_array[] = 'CRSA';
    if (isset($_POST['ET'])) $filiere_array[] = 'ET';
    if (isset($_POST['Pour_tous'])) $filiere_array[] = 'Pour tous';
    $filiere = implode(' ', $filiere_array);
}

function envoie_mail($from_name, $from_email, $subject, $nom, $prenom, $to_email, $filiere) {
    $mail = new PHPMailer();
    $mail->isSMTP();
    $mail->SMTPDebug = 0;
    $mail->Host = 'smtp.ionos.fr';
    $mail->SMTPAuth = true;
    $mail->Username = 'lycee-lgt.rempart@btsorient.fr';
    $mail->Password = [REDACTED];
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
    $mail->Port = 465;

    $mail->setFrom('lycee-lgt.rempart@btsorient.fr', 'Lycee-Rempart-Vinci');
    $mail->addAddress($to_email, $prenom . ' ' . $nom);
    $mail->isHTML(true);
    $mail->Subject = "Voici le programme du BTS que vous avez choisi : " . $filiere;
    $mail->Body = "Bonjour " . $prenom . " " . $nom . ", voici le programme du BTS que vous avez
choisi : " . $filiere;
    $mail->setLanguage('fr', '/optional/path/to/language/directory/');

    $formations = explode(' ', $filiere);
    for ($i = 0; $i < count($formations); $i++) {
        $mail->addAttachment(__DIR__ . '/Plaquette_BTS/plaquette_' . $formations[$i] . '.pdf');
    }

    return $mail->send();
}
if (envoie_mail($nom, $email, "Programme BTS", $nom, $prenom, $email, $filiere)) {
    echo '<h2>Email envoyé avec succès !</h2>';
} else {
    echo '<h2>Une erreur s\'est produite lors de l\'envoi du mail</h2>';
}
?>
```

**Voici des code source que j'ai trouvés au cours de mes recherches :** **HTML & CSS :**

<https://developer.mozilla.org/fr/docs/Web/HTML/Reference/Elements/form>  
<https://developer.mozilla.org/fr/docs/Web/CSS/position>  
[https://www.w3schools.com/css/css\\_website\\_layout.asp](https://www.w3schools.com/css/css_website_layout.asp)

 **JavaScript :**

<https://developer.mozilla.org/fr/docs/Web/API/Document/getElementById>  
<https://developer.mozilla.org/fr/docs/Web/API/Element/getBoundingClientRect>  
[https://developer.mozilla.org/fr/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/fr/docs/Web/API/Canvas_API)  
[https://www.w3schools.com/jsref/canvas\\_drawimage.asp#:~:text=The%20drawImage\(\)%20method%20draws,increase%2Freduce%20the%20image%20size](https://www.w3schools.com/jsref/canvas_drawimage.asp#:~:text=The%20drawImage()%20method%20draws,increase%2Freduce%20the%20image%20size)

 **Manipulation d'images et téléchargement :**

→ Exemple de fusion d'images avec canvas et téléchargement avec un lien dynamique

<https://stackoverflow.com/questions/29187029/save-multiple-canvas-as-one-image-make-website-like-picframe>  
<https://stackoverflow.com/questions/38851963/how-to-combine-3-canvas-html-elements-into-1-image-file-using-javascript-jquery>

→ Pour convertir une image en base64 et la transformer en lien de téléchargement.

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement/toDataURL>

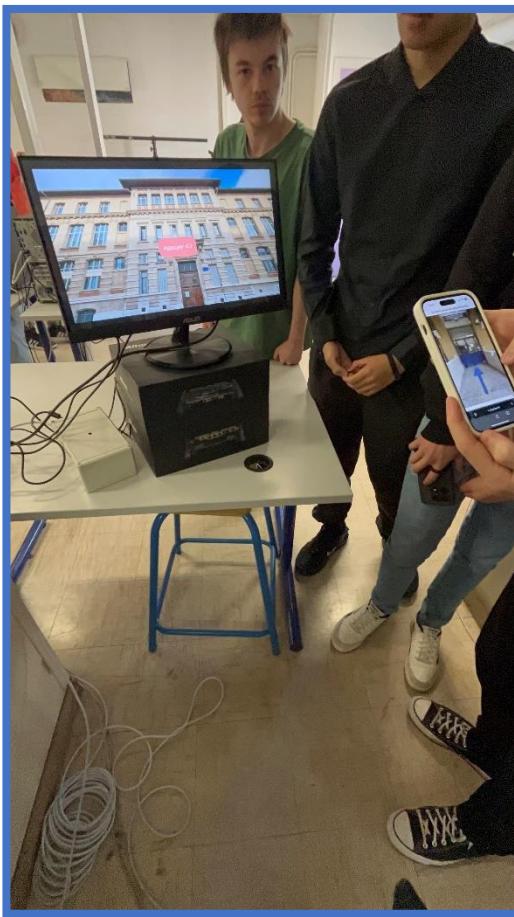
 **PHP :**

<https://www.php.net/manual/fr/reserved.variables.files.php>  
<https://www.php.net/manual/fr/function.session-start.php>  
[https://www.w3schools.com/php/php\\_file\\_upload.asp](https://www.w3schools.com/php/php_file_upload.asp)

→ Exemple de téléchargement de fichier

<https://formeezy.com/examples/file-upload-example>  
<https://uploadcare.com/blog/file-upload-php/>  
[https://www.w3schools.com/HOWTO/howto\\_html\\_file\\_upload\\_button.asp](https://www.w3schools.com/HOWTO/howto_html_file_upload_button.asp)

**Voici quelques photos de l'événement, accompagnées de témoignages :**



Bonjour Madame **BARNOUIN**,

votre idée de projet se concrétise.

Il ne s'agit que d'une version intra muros à un de vos établissement (adaptable à SAMSAT84).

Le projet est composé d'un logiciel fabriquant le parcours puis d'une borne proposant les QRcodes de chaque parcours. Un de vos protégés ou un visiteur occasionnel pourra scanner un des QRcodes afin d'être guidé à l'intérieur de vos structures, bâtiments etc... pour un rendez-vous ou lors de journées portes ouvertes...

Le GPS ne pouvant être actif dans des locaux, il s'agit de la meilleure solution technologique actuellement.

Bien sûr, la version "street view" de guidage dans un environnement inconnu et stressant, est en cours de codage, nous reviendrons vers vous plus tard.

Pour l'instant, afin de prévisualiser ce projet, 3 petits films (couper le son car le bruit est gênant) sont téléchargeables ou cliquables depuis le présent mail.

(les choix tactiles proposent des guidages vers des structures de formations, un smartphone suffit)  
(on peut laisser un avis, recevoir par mail de la documentation pour mieux connaître le lieu d'arrivée)

dans l'attente de vous lire et peut-être de vous présenter notre travail, nous vous remercions d'avoir instillé cette magnifique idée de réalisation.

cordialement  
M NARELLI  
BTS CIELA

Super vous avez bien travaillé !  
J'ai tout compris avec les vidéos !

Tu remercieras bien tes élèves pour leur persévérance dans ce projet inclusif, de la part de toute la communauté des Atypiques.

Nous n'avons pas de locaux pour l'application du logiciel en tant qu'équipe mobile mais :

L'idéal serait de l'installer dans toutes les universités et centres de formation !  
J'ai une convention avec l'université d'Avignon dont l'IUT et les beaux-arts. On pourrait commencer l'expérimentation par-là ?

Deux possibilités :

On peut venir vous rencontrer début juin avec trois personnes TSA accompagnées, douées et très intéressées avec les outils numériques qui pourraient apprendre la mise en place concrète de votre logiciel, dans des lieux inclusifs ? Pour les apprenants, cela serait intéressant pour eux aussi car ils devraient transférer leurs compétences acquises cette année, de manière très pédagogique...  
On peut les faire intervenir auprès des étudiants de l'IUT aussi mais je ne sais pas s'il y'a une sorte de concurrence entre vos établissements ? Ou si une synergie est possible entre étudiants ?

Ou

Est-ce que tes apprentis pourraient venir en "sortie de classe" à Avignon pour créer et installer les QRCodes dans les locaux, etc.? Comme un "stage collectif optionnel" ? On pourrait alors convenir pour que cette journée leur soit comptée dans leur cursus de professionnalisation ?

À peine ton projet numérique terminé, je t'en redemande, un peu plus encore ! J'suis insatiable !

Merci pour tout !

On s'en reparle quand on se revoit ?

Passe une bonne fin de semaine !

Cordialement,

Marion **BARNOUIN**

Neuropsychologue

Directrice Adjointe

SAMSAH TSA 84

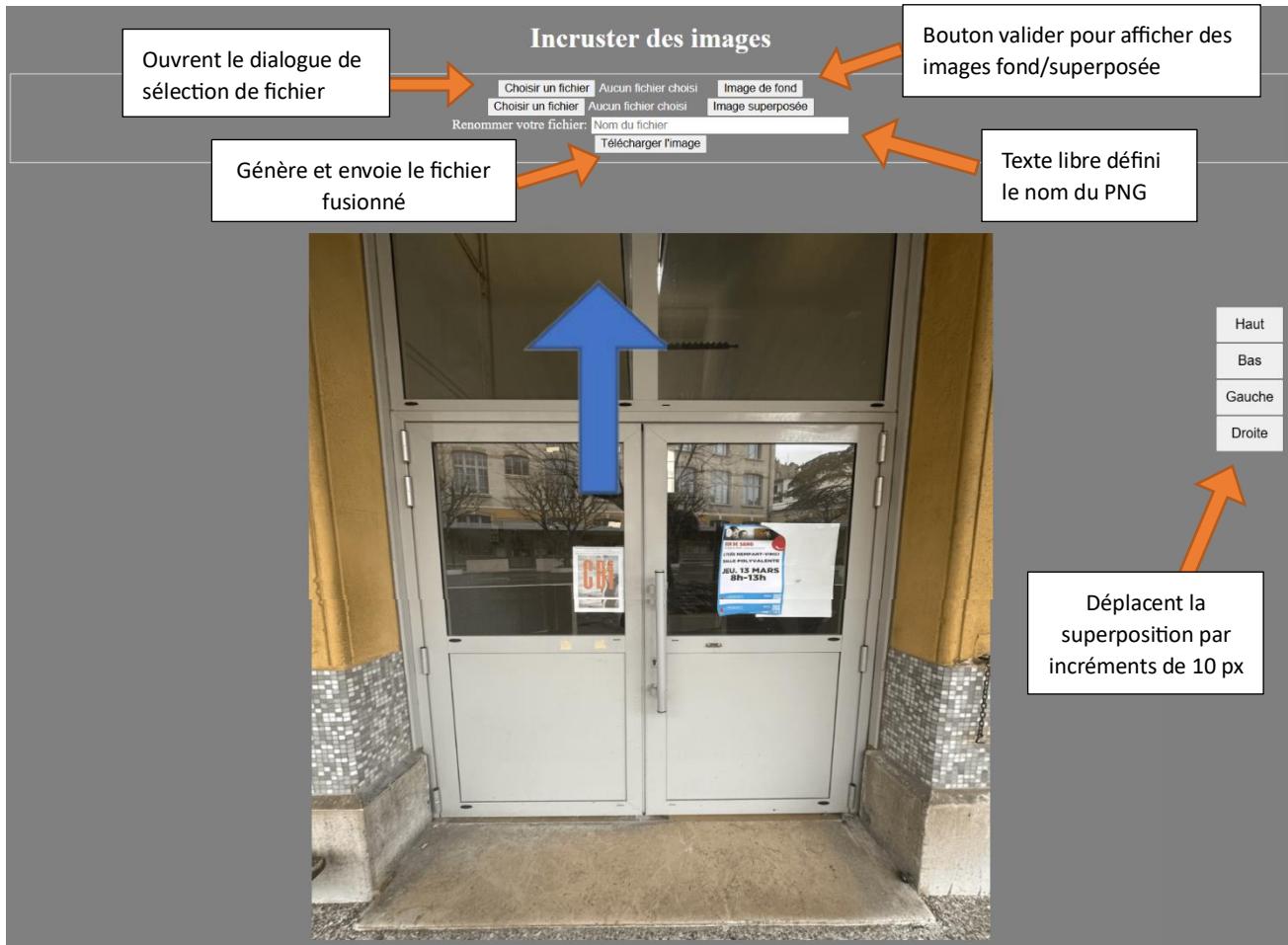
## Schéma de la base de données – Borne d'orientation



## Manuel d'utilisation de l'outil « Incruster des images »

### Prérequis

- Un navigateur moderne (Chrome, Firefox, Edge...)
- Un serveur PHP (avec sessions activées) et un répertoire Images/ accessible en écriture
- Des fichiers images légers (PNG, JPG < 8 Mo)



### Outils :

- **jpg2png.com** : permet de convertir une image JPG en PNG pour garder la transparence.
- **compresspng.com** : compresse les images PNG pour réduire leur taille sans perdre en qualité.
- **remove.bg** : enlève automatiquement l'arrière-plan d'une image pour obtenir un fond transparent.

## Étapes d'utilisation :

### 1. Charger l'image de fond

- Cliquez sur “**Image de fond**”, sélectionnez votre fichier, puis validez.
- L’aperçu s’affiche au centre, en 800x800 px.

### 2. Charger l'image superposée

- Cliquez sur “**Image superposée**”, choisissez votre icône/flèche, puis validez.
- L’élément apparaît par-dessus le fond.

### 3. Positionner la superposition

- À l'aide des boutons **Haut, Bas, Gauche, Droite**, déplacez l'image jusqu'à l'emplacement souhaité.
- Vous pouvez prévisualiser instantanément son effet sur le fond.

### 4. Renommer et télécharger

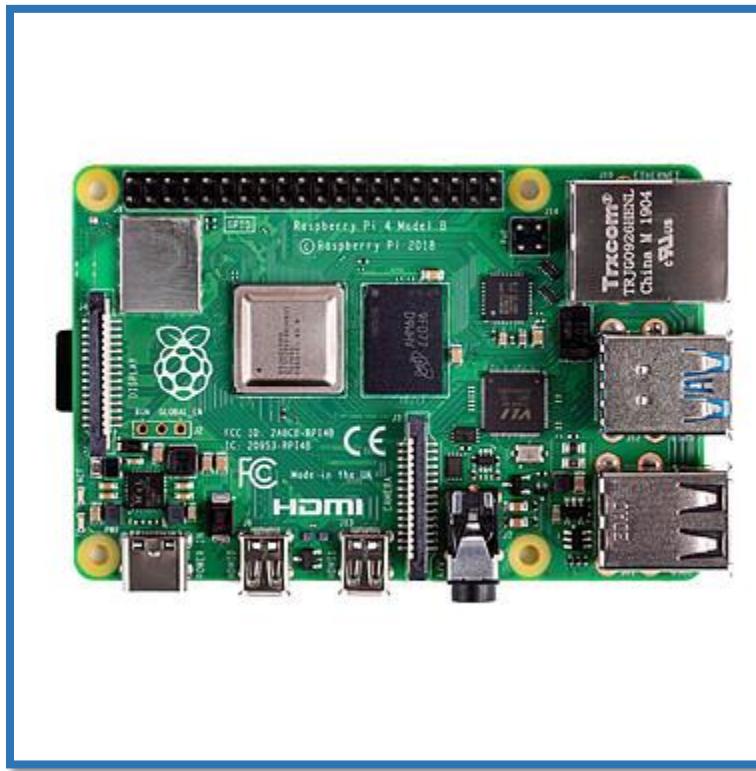
- Dans le champ “**Renommer votre fichier**”, saisissez un nom sans extension (par ex. parcours\_BTS\_1).
- Cliquez sur “**Télécharger l'image**” : un PNG sera généré à partir d'un <canvas> et lancé en téléchargement.

### 5. Recommencer

- Pour créer un autre parcours, rechargez simplement la page et répétez les étapes.

**Voici les caractéristiques principales du Raspberry Pi 4 model b :**

- **Carte mère Raspberry Pi 4**
- **Processeur** : Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- **RAM** : 4 Go LPDDR4
- **GPU** : VideoCore VI prenant en charge OpenGL ES 3.0, décodage HEVC 4K à 60 i/s
- **Connexion sans fil** : Bluetooth 5.0, Wi-Fi 802.11b/g/n/ac
- **Connexion filaire** : Gigabit Ethernet (RJ45)
- Lecteur de carte micro-SD (stockage non fourni)
- Port caméra CSI pour connecter la caméra Raspberry Pi
- Port d'affichage DSI pour connecter l'écran tactile Raspberry Pi
- **Audio** : AV 3.5 mm
- **Ports** : 2 x USB 3.0 / 2 x USB 2.0 / 1 x USB-C (alimentation seulement) / 1 x GPIO 40 pin / 1 x port quadripôle Audio/Vidéo composite / 2 x micro-HDMI
- **Alimentation** : 5V DC via un connecteur USB-C (minimum 3A), 5V DC via un en-tête GPIO (minimum 3A), compatible Power over Ethernet (PoE) (nécessite un HAT pour PoE)



**Structure du Répertoire du Projet Borne d'Orientation :**

```
.                                         # Interface utilisateur locale de la borne
|   └── Dashboard/
|       ├── Images/
|       ├── accueil.html
|       ├── album.css
|       └── borne_v3.php

|   └── Generateur des guides/      # Module de génération des itinéraires en local
|       ├── Images/
|       ├── Images_Final/
|       └── index.php

└── OrienInfor/                  # hébergée sur le site web public (serveur IONOS)
    ├── Images/
    ├── css/
    ├── js/
    ├── PHPMailer/
    ├── phoffice/
    ├── Plaquette_BTS/
    ├── vendor/
    ├── index.php
    ├── schema.sql
    └── ...

└── Rapport de projet/          # Dossier de documentation et illustrations
    ├── Diag cas d'utilisation.png
    ├── Diag d'exigence.png
    ├── Diag de séquence.png
    ├── Présentation_projet.pdf
    ├── Fiche de validation projet.pdf
    ├── Rapport de projet E6.pdf
    └── Synoptique du projet.png
```

## Lien vers le dépôt GitHub du projet

Vous trouverez l'intégralité du code source, de la documentation et des livrables de ce projet sur notre dépôt GitHub.

### Accéder au dépôt :

[https://github.com/AlexaNgder/Projet\\_Borne\\_Orientation\\_E6\\_TEMA](https://github.com/AlexaNgder/Projet_Borne_Orientation_E6_TEMA)

The screenshot shows the GitHub repository page for 'Projet\_Borne\_Orientation\_E6\_TEMA'. The repository is public and has 1 branch and 0 tags. The main file listed is 'README'. The repository was last updated 1 hour ago with 28 commits by AlexaNgder. The commits include updates to the README, a dashboard, guides, orientation files, project reports, a license, and the README itself. The repository page also includes sections for Packages, Contributors (AlexaNgder and Anthony19-20), Languages (PHP 89.5%, CSS 4.8%, HTML 3.4%, Hack 2.3%), and Suggested workflows (SLSA Generic generator, PHP).