

2024

RAPPORT DE STAGE

ADEISSE



»» Pecquenard Antoine

SOMMAIRE

01 Introduction

02 Etude de l'existant

03 Cahier des charges

04 Travail accompli

05 Conclusion

06 Annexe

INTRODUCTION

At Adeisse, I had the opportunity to do a five-week internship focused on learning programming languages. This subject allowed me to discover the field across the entire company, which allowed me to understand and exchange between the different actors in relation to the different interfaces.

During this period, I was able to immerse myself in this universe, explore the mechanisms of language learning and discover the different methods and approaches used. This report will detail the key stages of my internship, focusing on the most relevant aspects and the results obtained.

Firstly, I carried out a study of what existed in the field of learning programming languages. Through my analysis, I will present the work and research that I was able to carry out during my internship to learn more about all aspects of programming languages.

Secondly, we will examine the specifications established for my project. Which will precisely define the objectives that have been set for me, the constraints that I have taken into account and the resources available.

And thirdly, it will be devoted to the work carried out during my internship. I will present in detail the different activities in which I participated, the tools and methods I used as well as the results I obtained. I will also highlight the challenges encountered and the solutions implemented to overcome them. This section will illustrate how I was able to apply my theoretical knowledge to concrete situations within Adeisse.

Finally, I will conclude this report by summarizing the points that I addressed and providing an overall assessment of my internship. I will evaluate the results obtained in relation to the objectives initially set. In addition, I will also discuss future prospects in the field of learning programming languages.

ETUDE DE L'EXISTANT



Dans ce rapport de stage, je vais vous présenter la société Adeisse. C'est une entreprise de conseil et de Sécurité en Services Informatiques. Elle va auprès de ses clients pour définir leurs besoins et pour y répondre. De plus, la société Adeisse peut-être sollicités par des clients potentiels pour des demandes spécifiques.

Avec son expérience solide dans le domaine de la sécurité en services informatiques depuis 2005, année de sa création. Adeisse s'efforce d'offrir des services de qualité à ses clients.

L'Activité d'Adeisse est la sécurité en services informatiques. Son objectif principal est de garantir à ses clients la même prestation et de les accompagner dans leur demande durant toute la prestation. Pour atteindre cet objectif, Adeisse propose une large gamme de services adaptés aux besoins des entreprises.

Produits et services d'Adeisse :

I)-Support et gestion du parc informatique :

Adeisse propose un support ainsi que la gestion d'un parc informatique réunissant des activités visant à maintenir, optimiser et sécuriser les ressources informatiques d'une organisation.

Cela inclut le support technique pour résoudre les problèmes des utilisateurs, la gestion des actifs informatiques, la sécurité, la gestion des tickets et incidents, ainsi que la mise à jour et la maintenance des logiciels et matériels.

L'objectif global est d'assurer le bon fonctionnement des systèmes, de maximiser l'efficacité des ressources et de garantir la conformité aux normes de sécurité et aux objectifs stratégiques de l'entreprise.

ETUDE DE L'EXISTANT

II)-Assistance à maîtrise d'ouvrage :

En plus du support et de la gestion du parc informatique, Adeisse propose une Assistance à la maîtrise d'ouvrage qui consiste à fournir des services par prestation externe pour aider une entreprise cliente à gérer et réaliser efficacement un projet. Ces services incluent la définition des besoins, la planification, la coordination des parties prenantes, le contrôle de la qualité, la gestion des risques et la communication. L'objectif est d'apporter une expertise externe pour l'entreprise pour garantir le succès du projet et la satisfaction du maître d'ouvrage.

III)-Formation (organisme agréé et certifié) :

Adeisse met à disposition des formations dispensées par un organisme agréé et certifié qui permet de garantir une qualité et une conformité. Ces formations, visent à développer des compétences spécifiques et sont reconnues par les employeurs et les institutions. Ce qui renforce la crédibilité et la valeur de la formation sur le marché du travail.

Conclusion :

En résumé, Adeisse est une entreprise de prestation de services de qualité pour répondre aux besoins de chaque client. Que ce soit à travers le support et la gestion du parc informatique, une assistance à la maîtrise d'ouvrage ou encore par rapport à des formations qui sont proposés, Adeisse accompagne ses clients durant la prestation de service pour pouvoir leur offrir la meilleure des prestations et la plus adaptée.

Contexte de travail :

En tant que stagiaire au sein de l'entreprise, j'ai eu la tâche de créer une interface pour un client, entre le client et le fournisseur. Cela m'a permis de tester des méthodes dans les différents langages de programmation. Le html, le css mais encore le php, à travers les différentes tâches que j'ai pu effectuer pour arriver à réaliser ce projet. Ce qui m'a permis d'en apprendre plus sur ces langages mais également de tester de nouvelles choses.

Le directeur Christophe de l'entreprise m'a donnée pour projet de créer une interface pour que le client et le fournisseur ait à chaque instant accès aux informations qu'ils peuvent consulter. À travers les différents langages de programmation que j'ai pu utiliser. Ce projet m'a permis d'en apprendre d'avantage et de pouvoir réaliser ce projet à bien.

CAHIER DES CHARGES

L'objectif de mon stage au sein d'Adeisse est d'approfondir mes connaissances dans les langages de programmation (html,css,php et javascript).

Ce stage me permettra à court et à long terme de connaitre l'environnement de création de page web ainsi que la logistique des bases de données. Voici donc les objectifs que je me suis fixer pour mon stage :



N°1 : **HTML**

Approfondir mes compétences en HTML : Me concentrer sur les balises HTML de base, la structuration du contenu, la création de formulaires, l'intégration d'images mais également de vidéos. Ce qui m'a permis d'avancer dans le HTML5 et de pouvoir l'utiliser et de mieux le connaître.



N°2 : **CSS**

Approfondir mes compétences en CSS : Me concentrer sur les différents styles des sélecteurs CSS, la mise en page des éléments HTML via le CSS. La création des styles attrayants ainsi que l'optimisation du *design responsive* pour que les tailles soient responsives pour les différents écrans de différentes tailles.



N°3 **PHP**

Approfondir mes connaissances en PHP : Me concentrer sur l'affichage des tableaux avec différents contraintes à mettre dans les cases du tableau. Cela m'a permis d'en voir plus sur comment manipuler les différents fichiers text pour pouvoir limiter et ordonner les données que je voulais afficher dans le tableau en question.



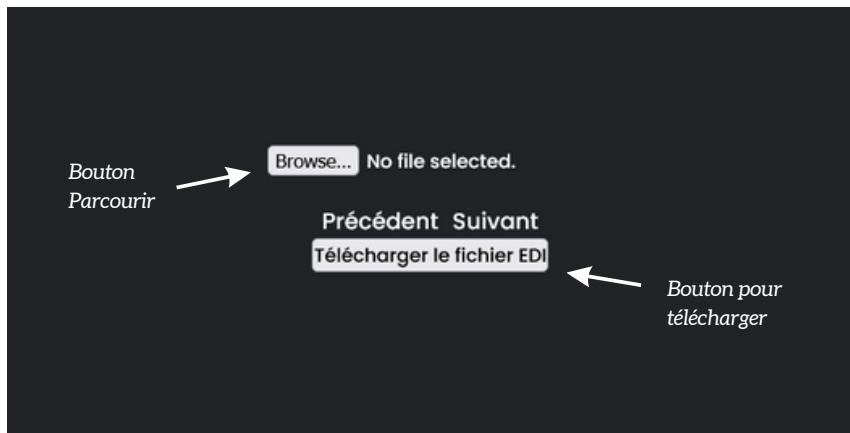
N°4 **PROJET**

Réalisation d'un projet : La création d'un site web avec une page qui comporte un tableau avec les différents langages de programmation, le HTML, le CSS et aussi du JavaScript. Ce qui m'a permis de mettre en page un tableau qui permettra d'afficher les données d'un fichier texte. Ce qui m'a permis de pouvoir rajouter des conditions dans le code pour sélectionner les bonnes informations demandées durant le projet.

Le système pourra donc servir à l'entreprise pour un client, ce qui permettra de lire des fichiers texte qui ne sont pas très lisibles si on ne connaît pas ce que cela veut dire.

TRAVAIL ACCOMPLI

Le projet consiste à travailler sur une page web qui peut accueillir plusieurs boutons, qui permette de choisir un fichier text. Ce qui permet ensuite via le code de le charger sur la page en question pour obtenir un tableau. Il y a également le bouton de téléchargement qui permet donc de télécharger les données présente dans le tableau qui son traduit du fichier text de base.



Voici comme ce présente la page en question, il y a le bouton Parcourir qui permet d'aller sélectionner le fichier en question

Et il y a le bouton de Télécharger le fichier EDI, qui permet de télécharger le nouveau fichier text que l'on souhaite avoir.

Les Boutons Précédent et Suivant sont présents au cas où.

Lorsque le fichier texte est sélectionné cela permet d'afficher les différentes notions, tel que la Référence article, la Désignation, le Numéro de Commande, la Date et la Quantité.

| Browse... test.txt | | | | |
|---|-------------|--------------------|------------|----------|
| Référence article | Désignation | Numéro de commande | Date | Quantité |
| 0060060000101 | BOBINE | 5500018321 | 2023-12-14 | 50 |
| 0060060000101 | BOBINE | 5500018321 | 2023-12-21 | 70 |
| Précédent Suivant Télécharger le fichier EDI | | | | |

```
UNB+UNOA:1+SENDER + RECEIVER +071223:1715+892111'
UNH+1+DELFOR:D:96A:UN:A09041'
LIN++3+0060060000101:IN'
IMD++:::BOBINE'
RFF+ON:5500018321'
DTM+171:20231214:102'
QTY+113:50:PCE'
SCC+1+Y'
DTM+171:20231221:102'
QTY+113:70:PCE'
SCC+1+Y'
UNT+34+1'
UNZ+1+892111'
```

Les Données qui sont traités dans ce tableau, sont de bases composées comme cela, ils sont dans un langage de communication de chaîne de production d'article.

Le tableau ci-dessus est codé pour que les informations de ce fichier soient en quelque sorte traduit de façon lisible.

TRAVAIL ACCOMPLI

Pour ce projet j'ai créé bien évidemment une navbar avec différentes pages qui ont pu servir pour la majorité, j'en ai créé cinq, 3 pages seulement m'on servis.

Nous allons dès à présent voir le début du code pour la sélection du fichier texte, que j'ai expliqué précédemment et le moyen de le télécharger.

```
101 <body>
102     <!-- Sélection de fichier -->
103     <input type="file" id="fileInput" accept=".txt" onchange="handleFile()"> ←
104
105     <!-- Le tableau sera ajouté dynamiquement ici -->
106     <table id="ediTable"></table>
107
108     <!-- Pagination -->
109     <div id="pagination">
110         <span id="prevPage" onclick="changePage(-1)">Précédent</span> ←
111         <span id="nextPage" onclick="changePage(1)">Suivant</span>
112     </div>
113     <!-- Bouton de téléchargement -->
114     <button onclick="downloadEDI()">Télécharger le fichier EDI</button> ←
115
116     <script>
117         // Initialisation des variables de pagination
118         let currentPage = 1;
119         const itemsPerPage = 10;
120
121         function isEDIFile(fileContent) {
122             const ediHeader = "UNB"; // En-tête caractéristique des fichiers EDI
123             // Convertir les premiers caractères du fichier en majuscules pour rendre la comparaison insensible à la casse
124             const fileStart = fileContent.substring(0, ediHeader.length).toUpperCase();
125             return fileStart === ediHeader;
126     }
```

Alors voyons de plus près le code, la flèche rouge montre que j'ai ajouté une condition qui ne prend que les fichiers txt pour le tableau pour éviter qu'il soit une erreur dans le code ou que l'on mette n'importe quoi dedans.

Ensuite la flèche bleue qui montre les bouton Précédent et Suivant qui sont présents pour changer de page du tableau en question. Ce qui facilite la taille du tableau sur la page en question. Et il y a le bouton de Téléchargement qui permet donc de télécharger le nouveau fichier EDI. Pour avoir les données du tableau.

```
127     // Gestion du fichier sélectionné
128     function handleFile() {
129         const fileInput = document.getElementById('fileInput');
130         const file = fileInput.files[0];
131         if (file) {
132             const reader = new FileReader();
133             reader.onload = function (e) {
134                 const content = e.target.result;
135                 // Vérifier si le fichier semble être un fichier EDI
136                 if (isEDIFile(content)) {
137                     processFileContent(content);
138                 } else {
139                     alert("Le fichier sélectionné n'est pas un fichier EDI valide.");
140                 }
141             };
142             reader.readAsText(file);
143     }
144 }
```

On a donc ensuite une fonction que j'ai appelé handleFile qui permet de gérer la gestion du fichier sélectionné qui empêche de prendre n'importe quel type de fichier.

TRAVAIL ACCOMPLI

Je vais donc ensuite expliquer comment j'ai pu passer d'un fichier text avec les données qui sont présentes dedans à un tableau avec les différents caractères dans les cases du tableau. Avec la fonction parseEDI j'ai pu ajuster les délimitations pour les différentes cases.

```
146 // Analyse du contenu du fichier EDI
147 function parseEDI(fileContent) {
148     const lines = fileContent.split('\n');
149     let reference, designation, numeroCommande, date, qty;
150     const data = [];
151
152     for (let i = 0; i < lines.length; i++) {
153         const line = lines[i];
154         const segments = line.split('+');
155
156         // Traitement des différents segments du fichier EDI
157         if (segments[0] === 'LIN') { ←
158             reference = extraireLimiteCaracteres(extrairAvantCaracteres(segments[3], 0), 13); // Supprimer
159
160         } else if (segments[0] === 'IMD') { ←
161             designation = extraireLimiteCaracteres(extrairAvantCaracteres(segments[3], 3), 6); // Supprimer
162
163         } else if (segments[0] === 'RFF') { ←
164             numeroCommande = extraireLimiteCaracteres(extrairAvantCaracteres(segments[1], 3), 10); // Supprimer
165
166         } else if (segments[0] === 'DTM') { ←
167             date = extraireLimiteCaracteres(extrairAvantCaracteres(segments[1], 4), 8); // Supprimer 4 caractères
168             date = formatDate(date); // Formater la date
169
170             const qtySegment = lines[i + 1];
171             if (qtySegment && qtySegment.startsWith('QTY')) { ←
172                 qty = extraireLimiteCaracteres(extrairAvantCaracteres(qtySegment.split(':')[1], 0), 4);
173             }
174
175             // Stockage des données dans un tableau
176             data.push({ reference, designation, numeroCommande, date, qty });
177         }
178     }
179     // Mise à jour du tableau HTML avec les données extraites
180     updateTable(data);
181 }
```

La fonction parseEDI que j'ai pu créer a permis comme je l'ai expliqué de sélectionner les infos nécessaires pour le tableau vue précédemment.

J'ai donc créé plusieurs variables : reference, designation, numeroCommande, date et qty (quantité).

Pour ce qui est de la date j'ai voulu rendre sa plus lisible, j'ai donc séparé l'année, le mois et le jour avec des tirets. Ce qui me permet la fonction formatDate que j'ai créé.

```
182 // Formatage de la date au format YYYY-MM-DD
183 function formatDate(date) { ←
184     let dateString = date.toString(); // Convertir la date en chaîne de caractères
185
186     // Utiliser split() pour séparer la date en trois parties
187     let year = dateString.substring(0, 4);
188     let month = dateString.substring(4, 6);
189     let day = dateString.substring(6, 8);
190     let formattedDate = [year, month, day].join('-'); // Utiliser join() pour réassembler les parties de la date avec des tirets
191
192     return formattedDate;
193 }
```

TRAVAIL ACCOMPLI

Ensuite nous allons voir la *function updateFileContent* que j'ai créé pour pouvoir mettre les informations traiter dans le fichier text qui ont été traiter dans le tableau, donc le fichier texte que l'on peut télécharger en appuyant sur le bouton Télécharger le fichier EDI.

```
248 // Fonction pour mettre à jour le contenu du fichier avec les données actuelles du tableau
249 function updateFileContent() {
250     const data = getDataFromTable();
251     const ediContent = generateEDIContent(data);
252 
253     // Mettre à jour le contenu du fichier actuellement chargé
254     const fileInput = document.getElementById('fileInput');
255     if (fileInput.files.length > 0) {
256         const loadedFile = fileInput.files[0];
257         const reader = new FileReader();
258 
259         reader.onload = function (e) {
260             const loadedContent = e.target.result;
261 
262             // Remplacer le contenu actuel du fichier par le nouveau contenu généré
263             const updatedContent = loadedContent.replace(/\n|\w+\s*IMD\+\w+\s*RFF\+\w+\s*DTM\+\w+\s*QTY\+\d+\:\PCE/g, ediContent);
264 
265             // Réinitialiser le contenu du fichier
266             const updatedBlob = new Blob([updatedContent], { type: 'text/plain' });
267             const updatedUrl = URL.createObjectURL(updatedBlob);
268             fileInput.value = ''; // Effacer le fichier actuel
269             fileInput.value = updatedUrl; // Charger le fichier mis à jour
270         };
271 
272         reader.readAsText(loadedFile);
273     }
274 }
```

La première flèche montre la fonction *updateFileContent*, la flèche bleue montre les conditions que j'ai ajouté pour que cela prenne bien les données du tableau en compte. Et la flèche verte montre le chargement et la réinitialisation le contenu du fichier.

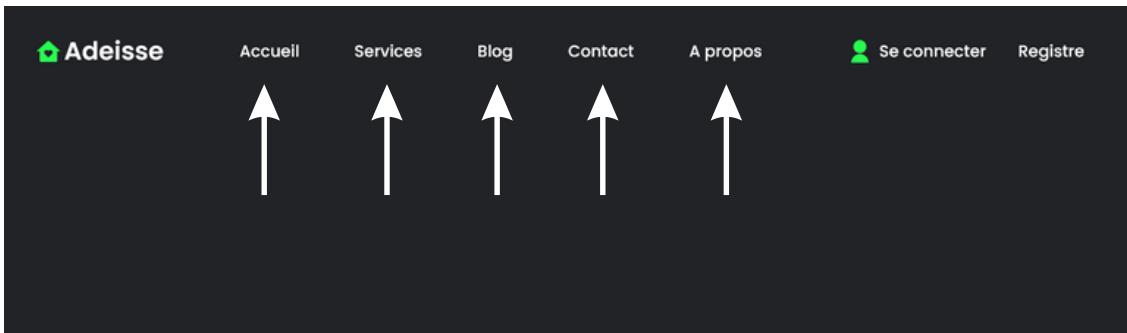
```
275 // Fonction pour télécharger le fichier EDI
276 function downloadEDI() {
277     updateFileContent(); // Mettre à jour le contenu du fichier
278     const data = getDataFromTable();
279     const ediContent = generateEDIContent(data);
280     downloadFile(ediContent, 'fichier.txt');
281 }
```

La fonction suivante est *function dowloadEDI* qui permet à l'utilisateur de cliqué sur le bouton "Télécharger le fichier EDI" afin qu'il puisse télécharger le nouveau fichier qui s'appellera "fichier.txt". Il aura donc ce nouveau fichier texte dans ces téléchargements.

Concernant les différentes fonctions que j'ai pu créer afin que le tableau soit fonctionnel, j'ai montré une partie des fonctions qui était pour moi les plus pertinentes lors de la réalisation du projet.

TRAVAIL ACCOMPLI

Comme énoncé plutôt dans les pages précédentes, j'ai pu créer une navbar pour que l'utilisateur puisse de déplacer entre les pages de façon fluide afin qu'il puisse accéder aux différents outils créer et mise à sa disposition.



J'ai donc créer une page principale qui est l'Accueil du site web en question, on y retrouve plusieurs pages. La page Accueil, Services, Blog, Contact et A propos. Les pages créer son Accueil, Services, Blog et Contact. Les pages Accueil et Services ont juste le menu disponible dessus. La page Blog à un tableau que je vais vous présenter par la suite. Et la page contact à le tableau par rapport au fichier text expliqué précédemment. (Les noms dans la navbar peuvent changer en fonction de la demande de l'utilisateur. Cela n'est qu'une phase de test.

```
index.html > html > head
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <!-- Déclaration de la version HTML utilisée -->
5      <meta charset="utf-8">
6      <!-- Configuration de l'affichage sur les différents dispositifs -->
7      <meta name="viewport" content="width=device-width, initial-scale=1">
8
9      <!-- Titre de la page affiché dans l'onglet du navigateur -->
10     <title>Accueil du Site</title> ←
11
12     <!-- Liaison vers une feuille de style externe -->
13     <link rel="stylesheet" type="text/css" href="style.css">
14
15     <!-- Liaison vers une icône pack (Remix Icon) depuis un CDN -->
16     <link href="https://cdn.jsdelivr.net/npm/remixicon@4.0.0/fonts/remixicon.css" rel="stylesheet"/>
17
18     <!-- Liaison vers une autre bibliothèque d'icônes (Boxicons) depuis un CDN -->
19     <link rel="stylesheet" href="https://unpkg.com/boxicons@latest/css/boxicons.min.css">
20
21     <!-- Préconnexion aux serveurs de Google Fonts pour améliorer le chargement des polices -->
22     <link rel="preload" href="https://fonts.googleapis.com">
23     <link rel="preload" href="https://fonts.gstatic.com" crossorigin> ←
24
25     <!-- Liaison vers une police Google (Poppins) avec des styles spécifiques -->
26     <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@500;600&display=swap" rel="stylesheet">
27 </head>
```

Voici donc les premières lignes de code, avec la balise "head" où j'ai pu mettre le titre mais également un lien vers un fichier "style.css" qui sera en partie Annexe.

Également il y a ensuite des liens pour rendre le style et la mise en page plus belle et plus propre vers google fonts et autres liens que j'ai utilisé pour les pages.

TRAVAIL ACCOMPLI

Il y a la partie "header" qui est l'en-tête de la page avec le logo maison Adeisse, mais également une navbar avec le lien vers les autres pages webs. Les flèches en blanc montrent les pages créés et les flèches en rouge montrent les pages non créés pour le moment.

```
29  <!-- En-tête de la page -->
30  <header>
31      <!-- Logo de la page avec une icône Remix Icon -->
32      <a href="#" class="logo"><i class="ri-home-heart-fill"></i><span>Adeisse</span>
33      </a>
34      <!-- Barre de navigation avec des liens vers différentes pages -->
35      <ul class="navbar">
36          <li><a href="index.html" class="active">Accueil</a></li> ←
37          <li><a href="services.html">Services</a></li> ←
38          <li><a href="#">Blog</a></li> ←
39          <li><a href="contact.html">Contact</a></li> ←
40          <li><a href="#">A propos</a></li> ←
41      </ul>
42      <!-- Section principale de l'en-tête avec des liens d'utilisateur et un bouton de menu -->
43      <div class="main">
44          <a href="#" class="user"><i class="ri-user-fill"></i>Se connecter</a>
45          <a href="#">Registre</a>
46          <div class="bx bx-menu" id="menu-icon"></div>
47      </div>
48  </header>
49  <!-- Liaison vers un fichier JavaScript externe -->
50  <script type="text/javascript" src="script.js"></script> ←
51 </body>
52 </html>
```

Ensuite il y a une div qui permet d'afficher "Se connecter" et Registre qui sont des liens que je n'ai pas développés mais qui pourrait l'être par la suite pour faire une connexion sécuriser.

Et la flèche verte désigne le lien vers la page de script, "script.js" qui est elle alimentée par du code qui permet de gérer la navbar.

```
28  <body>
29      <!-- En-tête de la page -->
30      <header>
31          <!-- Logo de la page avec une icône Remix Icon -->
32          <a href="#" class="logo"><i class="ri-home-heart-fill"></i><span>Adeisse</span>
33          </a>
34          <!-- Barre de navigation avec des liens vers différentes pages -->
35          <ul class="navbar">
36              <li><a href="index.html" class="active">Accueil</a></li>
37              <li><a href="services.html">Services</a></li>
38              <li><a href="blog.php">Blog</a></li> ←
39              <li><a href="contact.html">Contact</a></li>
40              <li><a href="#">A propos</a></li>
41          </ul>
```

Il y a donc ensuite la création de la page blog.php qui est donc une nouvelle page qui amène à un nouveau tableau qui est relié à la base de données de Microsoft SQL Server Management Studio. Les données afficher sur cette page sont donc les données que j'ai pu afficher grâce à une requête SQL qui m'a permis de sélectionner la base Campus de la table dbo.Client pour ensuite afficher une partie du tableau que l'utilisateur pourra donc demander.

TRAVAIL ACCOMPLI

Et pour finir voici le dernier outil que j'ai pu créer sur une page. Un tableau qui charge quand on lui met une requête SQL bien définie dans le code pour afficher ce que l'on veut sur la page dans le tableau. Il y a donc la partie de connexion et la partie de l'appel de la requête SQL.

```
1 <?php
2 function OpenConnection()
3 {
4     $serverName = "tcp:DESKTOP-PMSR7M2\SQLEXPRESS,1433";
5     $connectionOptions = array("Database" => "Campus", "Uid" => "userphp", "Pwd" => "1234");
6     $conn = sqlsrv_connect($serverName, $connectionOptions);
7     if($conn === false) {
8         // En cas d'erreur lors de la connexion, récupérer les erreurs et les afficher
9         $errors = sqlsrv_errors();
10        foreach ($errors as $error) {
11            echo "SQLSTATE: " . $error['SQLSTATE'] . "<br />";
12            echo "Code d'erreur: " . $error['code'] . "<br />";
13            echo "Message d'erreur: " . $error['message'] . "<br />";
14        }
15        die();
16    }
17    return $conn;
18 }
```

```
19 function ReadData()
20 {
21     try
22     {
23         $conn = OpenConnection();
24         $tsql = "SELECT CL_NOM, CL_DT_CREAT FROM dbo.CLIENT"; ←
25
26         $getProducts = sqlsrv_query($conn, $tsql);
27         if ($getProducts === false) {
28             die(formatErrors(sqlsrv_errors()));
29         }
30
31         echo "<table border='1' align='center'>"; // Début du tableau
32         echo "<tr><th>CL_NOM</th><th>CL_DT_CREAT</th></tr>"; // En-têtes des colonnes
33
34         while($row = sqlsrv_fetch_array($getProducts, SQLSRV_FETCH_ASSOC))
35         {
36             echo "<tr>"; // Début de la ligne
37             echo "<td>".$row["CL_NOM"]."</td>"; // Cellule pour CL_NOM
38             echo "<td>".$row["CL_DT_CREAT"]."</td>"; // Cellule pour CL_DT_CREAT
39             echo "</tr>"; // Fin de la ligne
40         }
41
42         echo "</table>"; // Fin du tableau
43
44         sqlsrv_free_stmt($getProducts);
45         sqlsrv_close($conn);
46     }
47     catch(Exception $e)
48     {
49         echo("Error!");
50     }
51 }
52
53 ReadData();
54
55 echo "Bonjour";
56
57 ?>
```

J'ai donc connecté le nom du serveur Microsoft SQL Management Studio. J'ai ensuite entré les identifiants et mot de passe de l'utilisateur que j'avais créé pour la face de test. J'ai également rajouté un foreach au cas où il y aurait une erreur.

J'ai ensuite créé une fonction ReadData qui permet donc la connexion grâce à la fonction précédente, j'ai ensuite mis la requête SQL que l'on souhaitait. J'ai donc ensuite rajouté des éléments pour que ces données soient mises dans un tableau.

Et donc un "echo" pour afficher le début du tableau, un autre pour les en-têtes des colonnes et d'autre pour les lignes et un dernier pour la fin du tableau. J'ai mis un catch qui est une Exception si cela ne fonctionne pas il y aura un message d'erreur qui s'affichera et écrira "Error".

Je ne montre pas les données présentes dans le tableau car ces données sont des données confidentielles d'une base de données qui est fonctionnelle, cela retourne d'informations gardées confidentielles par l'entreprise Adeisse.

CONCLUSION

Pendant mon stage, j'ai eu l'opportunité de travailler sur plusieurs pages web comprenant différents outils que j'ai pu créer. J'ai pu faire évoluer ce projet durant ces cinq semaines de stages que j'ai pu mener à bien. Cette expérience m'a permis d'approfondir mes connaissances dans les langages de programmation tels que HTML, CSS et PHP.



Travailler au sein d'une entreprise familiale cela m'a permis de développer mon autonomie face aux objectifs qui m'ont été donnés. J'ai apprécié travailler dans ce contexte qui était pour moi une opportunité de mettre en pratique mes connaissances.

En ce qui concerne la mise en production de mon travail en entreprise, je me positionne favorablement. Les pages web que j'ai développées ont subi plusieurs changements et améliorations, mais cela m'a permis d'apprendre de mes erreurs et d'en faire des pages fonctionnelles.



En conclusion, mon stage m'a apporté une nouvelle expérience, tant sur le plan humain que technique. J'ai pu approfondir mes compétences en programmation web et découvrir le travail du quotidien d'une petite entreprise. J'ai également développé ma capacité à être autonome sur le projet que l'on m'a confié, même si parfois je faisais le point pour avoir plus d'explication. Ce stage a renforcé ma passion pour le développement web et m'a donné la confiance nécessaire pour continuer à progresser dans ce domaine.



ANNEXE

Voici la page de script.js dont j'ai pu parler durant la partie "Travail Accompli" par rapport à l'affichage de la navbar des pages web que j'ai pu développer lors de mon stage :

```
1 // Sélection des éléments HTML avec les classes et les identifiants appropriés
2 let menu = document.querySelector('#menu-icon');
3 let navbar = document.querySelector('.navbar');
4
5 // Ajout d'un gestionnaire d'événement pour le clic sur l'icône du menu
6 menu.onclick = () => {
7     // Basculement de la classe 'bx-x' pour l'icône du menu
8     menu.classList.toggle('bx-x');
9
10    // Basculement de la classe 'open' pour la barre de navigation
11    navbar.classList.toggle('open');
12 }
```

```
1 /* Réinitialisation des styles de base pour assurer une base propre */
2 {
3     padding: 0;
4     margin: 0;
5     box-sizing: border-box;
6     font-family: 'Poppins', sans-serif;
7     text-decoration: none;
8     list-style: none;
9 }
10 /* Définition des variables CSS personnalisées */
11 :root{
12     --bg-color: #222327;
13     --text-color : #fff;
14     --main-color : #29fd53;
15 }
16 /* Styles généraux du corps de la page */
17 body{
18     min-height: 100vh;
19     background: var(--bg-color);
20     color: var(--text-color);
21 }
22 /* Styles pour l'en-tête de la page */
23 header{
24     position: fixed;
25     width: 100%;
26     top: 0;
27     right: 0;
28     z-index: 1000;
29     display: flex;
30     align-items: center;
31     justify-content: space-between;
32     background: transparent;
33     padding: 28px 12%;
34     transition: all .50s ease;
35 }
```

Les couleurs de la navbar

Les différents styles de la pages

La disposition du corps de la page par rapport au header

```
103 /* Styles pour les médias query responsives */
104 @media (max-width: 1280px){
105     header{
106         padding: 14px 2%;
107         transition: .2s;
108     }
109     .navbar a{
110         padding: 5px 0;
111         margin: 0px 20px;
112     }
113 }
114 @media (max-width: 1090px){
115     /* Affichage de l'icône du menu pour les écrans de petite taille */
116     #menu-icon{
117         display: block;
118     }
119     .navbar{
120         /* Styles pour la version mobile de la barre de navigation */
121         position: absolute;
122         top: 100%;
123         right: -100%;
124         width: 270px;
125         height: 29vh;
126         background: var(--main-color);
127         display: flex;
128         flex-direction: column;
129         justify-content: flex-start;
130         border-radius: 10px;
131         transition: all .50s ease;
```

Responsive navbar

L'icone pour le menu

Styles pour une version mobile de la barre de navigation

Voici donc la page style.css qui a permis de rendre plus lisible et esthétique les différentes pages et les tableaux que j'ai pu créer :

```
37 .logo{
38     display: flex;
39     align-items: center;
40 }
41 .logo i{
42     color: var(--main-color);
43     font-size: 28px;
44     margin-right: 3px;
45 }
46 .logo span{
47     color: var(--text-color);
48     font-size: 1.7rem;
49     font-weight: 600;
50 }
51 /* Styles pour la barre de navigation */
52 .navbar{
53     display: flex;
54 }
55 .navbar a{
56     color: var(--text-color);
57     font-size: 1.1rem;
58     font-weight: 500;
59     padding: 5px 0;
60     margin: 0px 30px;
61     transition: all .50s ease;
62 }
```

Voici également le style pour le logo mais également pour la navbar. Les différents styles sont pour la taille mais également pour la couleur du logo mais de la disposition de la navbar.

Et voici les différents @media qui permette de rendre responsives les pages web sur les différents appareils.

Que cela soit pour la navbar, l'icône pour le menu ou encore pour le changement de format pour les mobiles cela est prévue pour être responsives.