

TP Développeur Web / Web Mobile

GIL RIVAS Jorge

Dossier Projet

Réalisation d'un site web de gestion de
relation client

ONLINE
FORMAPRO

< ACCESS **CODE** >
S C H O O L

Stage du 14/11/22 au 13/11/23

Sommaire

1. Introduction	3
2. Liste des compétences du référentiel	4
3. Le Projet	
3.1 Résumé du projet	5
3.2 Cahier des charges, expression des besoins	6
3.3 Spécifications techniques du projet	7
4. Réalisations	9
4.1 base de données	9
4.2 page de connexion	9
4.3 page client	15
4.4 page services	17
4.5 Pages devis, facture et avoir	18
4.6 FPDF	21
5. Description de la veille	28
6. Description d'une situation ayant nécessité une recherche anglophone	30
7. Conclusion	33
8. Annexes	34

Introduction

L'entreprise

Le labo est une entreprise de communication située au 122 RUE GEORGES TROUILLOT, 39000, LONS-LE-SAUNIER.

L'entreprise travaille avec des campagnes publicitaires, des campagnes de référencement payantes, gérer les réseaux sociaux et la création de sites internet

Le stage

La durée du stage était de 2 mois, du 14 novembre au 13 janvier, la première semaine j'ai découvert le fonctionnement d'elementor. Elementor est un constructeur de pages WordPress par le système **Drag and Drop**. C'est-à-dire que pour créer une page Web via WordPress, il vous suffit de faire glisser et de déposer les différents éléments qu'Elementor met à votre disposition depuis son panneau de contrôle.

Après avoir découvert certaines fonctionnalités d'elementor, on m'a confié la tâche de créer un site d'administration pour l'entreprise

Liste des compétences du référentiel

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité:

1. Réaliser une interface utilisateur web statique et adaptable

J'ai intégré le code qui m'a été donné par un autre stagiaire à mon code, j'ai dû adapter ce code à la structure Modèle-Vue-Contrôleur.

2. Développer une interface utilisateur web dynamiques

Les interfaces créées varient selon le comportement de l'utilisateur, qu'il s'agisse d'appuyer sur un bouton dans le menu pour naviguer vers une autre page avec un contenu différent ou d'ouvrir un modal, ou de devoir remplir un formulaire.

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité:

1. Créer une base de données

J'ai créé la base de données à l'aide de MySQL, car l'entreprise travaille avec ce système de gestion de base de données.

2. Développer les composants d'accès aux données

J'ai utilisé l'objet « PHP Data Objects » (PDO) pour créer l'accès à la base de données

3. Développer la partie back-end d'une application web ou web mobile

J'ai utilisé le back-end pour gérer les utilisateurs, les clients, les devis, les factures, les services et les avoirs.

Résumé du projet

Lelabocrm.com est un site web de gestion de clients, le site se concentre sur la création automatisée de PDF.

La liste des PDF pouvant être créés comprend des:

- Devis.
- Factures.
- Avoirs.

Le site a été créé à partir de zéro en utilisant le modèle **MVC** (Model-View-Controller ou Modèle-Vue-Contrôleur) et la librairie **FPDF** pour la génération de PDF.

Le site web est utilisé pour se connecter avec un compte administrateur avec des fonctionnalités exclusives ou en tant qu'utilisateur normal, en tant qu'administrateur, vous avez le droit d'ajouter des services, d'ajouter des clients, de créer des PDF, de créer des utilisateurs, de tout supprimer ou modifier les clients ou les services.

L'utilisateur normal n'a le droit de visualiser que les PDF et les clients.

Les idées du site ont été prises d'un autre site web existant de l'entreprise appelé **lelabolons.fr** et des idées de mon tuteur au fur et à mesure de l'avancement du projet.

La structure « **front-end** » du site a été faite par un autre stagiaire de l'entreprise, le reste j'ai travaillé sur la partie « **back-end** » et autres fonctionnalités « **front-end** ».

Cahier des charges

L'objectif du stage était de créer un site web **CRM** (Customer Relationship Management) ou **GRC** (Gestion de la Relation Client) pour permettre à l'entreprise de gérer la comptabilité de ses clients, et surtout de créer des PDF automatiques.

Les étapes de ce projet sont les suivantes:

- Créer une base de données contenant les tables suivantes:
 - users
 - client
 - services
 - devis
 - facture
 - avoir
- Intégrer un système de connexion.
- Faire des interfaces de gestion pour chaque type de PDF (Devis, Facture, Avoir)
- Le site doit avoir 2 types de rôles d'utilisateur (administrateur et utilisateur)
- Utiliser un outil de génération de PDF
- La structure du site web doit être faite dans le modèle MVC.
- Le développement du projet devra être fait en Objet.

Spécifications techniques du projet

Outils de versioning

L'entreprise n'utilise aucun outil de version, donc le choix d'en utiliser un était le mien et en même temps, étant un projet individuel, j'ai choisi d'utiliser **Git** et **GitHub** pour avoir une trace de mon travail et aussi avoir une copie de sauvegarde.

Environnement de développement

Ordinateur

J'ai utilisé l'ordinateur Windows qui m'a été donné par l'Access Code School.

Éditeur de code

J'ai utilisé Visual Studio Code, Créé par Microsoft, utilisable sur Windows et Mac

Type d'architecture

Le type d'architecture que j'ai utilisé est de type MVC, la lettre **M** signifie «Modèle», la lettre **V** pour «Vue» et **C** pour «Contrôleur».

- **Modèle:** Représente la structure logique cachée des données dans une application logicielle, le modèle ne contient aucune information d'interface utilisateur.
- **Vue:** C'est un ensemble de classes qui représentent les éléments de l'interface utilisateur (tout ce que l'utilisateur voit à l'écran et avec lequel il peut interagir : boutons, boîtes de dialogue, etc.).

- Contrôleur: Ce sont les classes qui se connectent au modèle et à la vue, et servent à la communication entre les classes du modèle et la vue.

Pour créer le modèle MVC, je l'ai créé «From Scratch» (à partir de zéro).

Navigateur web

J'ai codé le site exclusivement sur le navigateur Google chrome.

Serveur de développement

J'ai utilisé le serveur local par défaut «Localhost» et le système de gestion de base de données «MySQL».

Pour pouvoir utiliser ces deux il faut installer **WampServer** une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2.

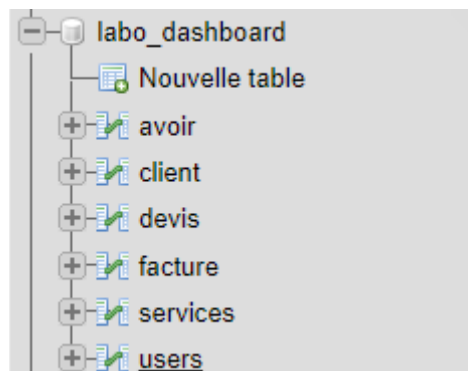
Réalisations

Pour la création du site lelabocrm.com, on m'a demandé de travailler avec le modèle MVC, la première chose que j'ai faite a été de chercher à comprendre le fonctionnement de cette structure.

Après avoir compris la structure MVC j'ai commencé à créer la base de données puis la page de connexion.

Base de données

Au départ, la base de données ne comprenait qu'une table "users", puis au fur et à mesure de l'avancement du projet et à la fin de la page de connexion j'ai ajouté de nouvelles tables.



Page de connexion

La première étape de l'intégration du projet a consisté à pouvoir se connecter à la base de données.

Le premier fichier à créer était "Connection.php" dans le dossier "Models", après avoir vérifié que cela fonctionnait j'ai commencé la création de la page de connexion.

J'ai donc commencé à créer 3 fichiers, "Controller.php" dans le dossier "Controllers", "User.php" dans le dossier Models et "login.php" dans le dossier Views.

Controller.php

Dans ce fichier, j'ai utilisé la fonction «spl_autoload_register()» qui sert à appeler mes classes automatiquement sans avoir à utiliser un include() pour chaque classe créée dans le dossier «Models»

```
spl_autoload_register(function ($class_name) {  
    include './Models/' . $class_name . '.php';  
});
```

Ensuite, j'ai créé la fonction **loginPage()** où j'appellerai ma classe User et les fonctions pour communiquer avec la base de données et invoquer ma vue.

```
public function loginPage(){  
  
    if(isset($_POST['login']) && isset($_POST['email']) && !empty($_POST['email']) && isset($_POST['password']) && !empty($_POST['password'])){  
        $username = htmlspecialchars($_POST['email']);  
        $password = htmlspecialchars($_POST['password']);  
        $user = new User();  
        $action = $user->login($username, $password);  
    }  
    else if(isset($_SESSION['nom'])){  
        header('location: ./');  
    }  
  
    require(__DIR__."/../views/login.php");  
}
```

Et pour finir la fonction urlManager() qui sert à gérer l'adresse de mon site avec un autre fichier appelé «.htaccess».

```
if(empty($_GET['page'])){  
    $this->mainPage();  
}else if($_GET['page'] === 'login'){  
    $this->loginPage();  
}
```

Dans cette fonction je crée une condition qui consiste à lire mon lien, la version textuelle de cette condition serait, « Si mon lien est vide envoie moi sur la page d'accueil, mais s'il contient "login" envoie moi sur la page de connexion. ».

User.php

Dans ce fichier vous trouverez toutes les fonctions spécifiques pour se connecter en tant qu'utilisateur, en même temps des fonctions pour ajouter un utilisateur, afficher tous les utilisateurs et se déconnecter

- login() :

Cette fonction sert à se connecter, on vérifie si l'utilisateur saisi et le mot de passe correspondent à ce qu'il y a dans la base de données

```
public function login($email, $password) {
    $login = $this->connect()->prepare("SELECT * FROM users WHERE user_email = :email");
    $login->bindValue(':email', $email);
    $login->execute();
    $logData = $login->fetch();

    if ($logData && password_verify($password, $logData[2])) {
        $_SESSION['nom'] = $logData['user_email'];
        header('Location: ./');
    }
    else {
        $loginError = "Nom d'utilisateur ou mot de passe incorrect !";
        return $loginError;
    }
}
```

- register() :

Cette fonction sert à ajouter un utilisateur, dans cette fonction nous devons indiquer si l'utilisateur est administrateur ou non, en même temps nous utilisons la fonction «**password_hash()**» qui sert à chiffrer

notre mot de passe et à le rendre plus sécurisé dans cas où la base de données serait attaquée.

```
public function register($email, $password, $isAdmin) {
    $available = $this->connect()->prepare("SELECT * FROM users WHERE user_email = :email");
    $available->bindValue(':email', $email);
    $available->execute();
    $emailExist = $available->fetch();
    if ($emailExist != false) {
        $msg = "Ce nom d'utilisateur est déjà utilisé.";
        session_destroy();
        return $msg;
    }
    else {
        $insert = $this->connect()->prepare("INSERT INTO users (user_email, user_password, is_Admin) VALUES (:email, :pwd, :isAdmin)");
        $insert->bindParam(':email', $email, PDO::PARAM_STR);
        $insert->bindParam(':pwd', $password, PDO::PARAM_STR);
        $insert->bindParam(':isAdmin', $isAdmin, PDO::PARAM_STR);
        $insert->execute();
        $_SESSION['nom'] = $email;
    }
}
```

- show() et deleteUser()

La fonction show() va chercher tous les utilisateurs dans la base de données puis va créer une boucle pour naviguer dans la liste récupérée par la requête faite, puis dans la boucle on va générer la liste à afficher dans l'interface utilisateur.

```

public function show(){
    $findData = $this->connect()->prepare("SELECT * FROM users");
    $findData->execute();
    $data = $findData->fetchAll();

    foreach ($data as $key => $value) {
        echo
        '<tr>',
        '<td style="text-align: center;">'. $value['user_email']. '</td>';

        if($value['is_Admin'] != 0){
            echo '<td style="text-align: center; text-transform: uppercase;">oui</td>';
        }else{
            echo '<td style="text-align: center; text-transform: uppercase;">non</td>';
        }

        echo
        '<td style="text-align: center;"><form method="POST"><input type="hidden" name="id" value="'. $value['id_user']. '"><input
        type="submit" name="delete" value="Delete"></form></td>',
        '<td style="text-align: center;"><form method="GET"><input type="hidden" name="id" value="'. $value['id_user']. '"><input
        type="submit" name="modifier" value="modifier"></form></td>',
        '</tr>';
    }
}

```

Toutes les utilisateurs

+ Créer un utilisateur

User	Admin		
contact@agencelelabo.fr	OUI	Delete	modifier

La fonction deleteUser() lorsqu'un bouton est pressé supprimera l'utilisateur sélectionné de la base de données.

```

public function deleteUser($id){
    $deleteUser = $this->connect()->prepare("DELETE FROM `users` WHERE id_user = :id");
    $deleteUser->bindParam(':id', $id, PDO::PARAM_STR);
    $deleteUser->execute();
}

```

- isAdmin()

La fonction isAdmin() est utilisée pour renvoyer une donnée booléenne, ce qui signifie qu'elle renverra vrai ou faux si l'utilisateur est un administrateur.

```

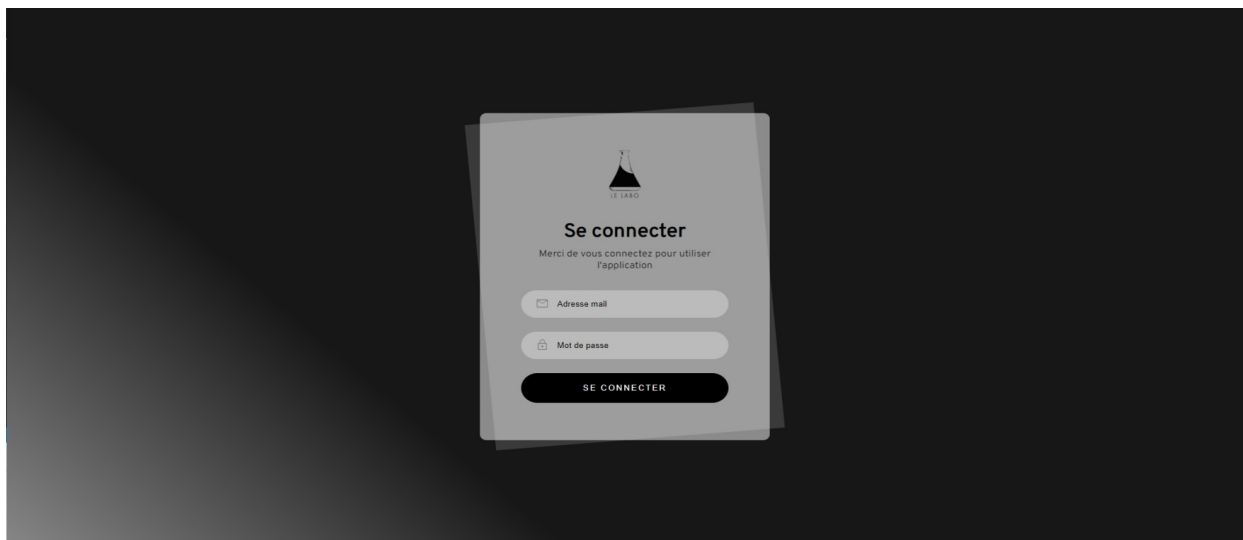
public function isAdmin($user){
    $select = $this->connect()->prepare("SELECT * FROM users WHERE user_email = :email");
    $select->bindParam(':email', $user, PDO::PARAM_STR);
    $select->execute();
    $fetchUser = $select->fetch();

    return $fetchUser;
}

```

Login.php

Le fichier login.php est la vue, ici je crée un formulaire de connexion, dont les données seront ensuite envoyées pour vérifier son existence dans la base de données.



Le style de la page web, parlant de CSS (**Cascading Style Sheets en anglais, ou Feuilles de style en cascade**) , a été réalisé par un autre stagiaire de l'entreprise, je me suis occupé de diviser les parties et de créer un template à partir de son code, j'ai donc créé le fichier «template_connection.php» puis j'ai utilisé ce template pour ajouter mon formulaire.

```
Views > template_connection.php
You, 4 weeks ago | 1 author (You)

1  <?php $css = './assets/css/style.css'; ?>
2
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta http-equiv="X-UA-Compatible" content="IE=edge">
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
9      <title><?php
10         if(!empty($title)){
11             echo $title;
12         }?></title>
13      <link rel="stylesheet" href="<?=$css; ?> ">
14      <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@48,400,0,0" />
15
16      <link rel="icon" type="image/png" href="assets/image/favicon.png" />
17  </head>
18  <body>
19
20      <?php
21          if(!empty($login)){
22              echo $login;
23          }
24
25      ?>
26  </body>
27  </html>
You, 2 months ago • back and front
```

puis nous avons le fichier login.php que nous avons commencé à utiliser `ob_start()` cette fonction initialise le tampon de sortie, c'est-à-dire que tout ce que vous affichez après son appel (avec `echo` ou `print`) est mis dans le tampon au lieu d'être retourné au navigateur et à la fin nous avons `ob_get_clean()` qui renvoie le contenu du tampon de sortie puis supprime le contenu du tampon.

```
<?php
ob_start();

?>
```

```
<?php

$title = "Login";
$login = ob_get_clean();
require_once('template_connection.php');
?>
```

Page client

La page client contient tous les clients de l'entreprise, et dans les dossiers «Models» et «Views» elle a ses propres fichiers comme «Client.php». en

même temps dans les dossiers «Views» j'ai créé un template appelé (template.php) qui sera utilisé pour mes autres pages autres que la page de connexion. Cela fonctionnera de la même manière que la page de connexion, j'utilise ob_start() et ob_get_clean().

Cette page partage certaines fonctions de la page de connexion, comme la fonction «register()» mais dans ce cas, elle s'appelle «addClient()», cette page est nécessaire pour pouvoir par la suite créer un devis, une facture ou un avoir.

- Fonction modal

Dans cette fonction, je crée une fonction sql qui envoie les données saisies dans un formulaire vers la base de données, lors de l'appui sur le bouton "ajouter devis/facture/avoir" un modal apparaîtra, l'interaction du modal s'est faite avec JavaScript.

```
function modal(buttonOpen, buttonClose, container){  
  
    if(document.body.contains(container)){  
        buttonOpen.addEventListener('click', ()=>{  
            container.classList.remove('none')  
        })  
    }  
  
    if(document.body.contains(container)){  
        buttonClose.addEventListener('click', ()=>{  
            container.classList.add('none')  
        })  
    }  
  
}
```

Dans cette fonction, j'ajoute les paramètres « buttonOpen » qui, lorsqu'il est pressé, ouvrira le modal, «buttonClose» qui le fermera et "container" qui est le conteneur qui apparaîtra ou disparaîtra, à l'intérieur de la fonction, nous avons un écouteur (addEventListener) avec le paramètre 'click' que l'utilisateur lors de l'interaction avec le bouton, effectuera l'action, L'action serait « classList.remove » qui supprimera la classe 'none' de mon élément ou « classList.add » qui fera l'inverse.

Dans notre conteneur, nous ajoutons la classe « .none » et en CSS, nous appelons cette classe et lui donnons le style « display: none; » ce qui le fera ne pas apparaître dans l'interface.

Page services.

Sur cette page j'ai créé la fonction pour ajouter des services, avec leur nom, description, prix et un mot-clé qui sera utilisé plus tard dans la génération du PDF.

Toutes les prestations			+ Ajouter un service
Service	Description	Prix	
Pack Gold	rjgdfikofgh;flghlfghfghfthfhf	6500€ Delete Modifier	
Création site E-Commerce	fdbgjntygj,gyjnhfgnghyj,tgy,fqn	4999€ Delete Modifier	
Pack Silver	qfdjhghkjsrgljdrhbkdfhbdnfgj	4000€ Delete Modifier	
Création Site Vitrine	Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500	1999€ Delete Modifier	
Maintenance préventive et curative		1000€ Delete Modifier	
Cession des droits	dsgfdgfhfghfghfghf	600€ Delete Modifier	
Hébergement de votre			

Lorsque vous appuyez sur le bouton pour ajouter un service, un modal apparaîtra, et là nous pouvons sélectionner le mot-clé pour chaque service.

AJOUTER UN SERVICE

Nom du service

Description du service

Prix du service

Mot clé

Mot Clé

Site Vitrine

Hébergement

Maintenance

Assistance

E-Commerce

Pack Gold

Pack Silver

Yeast Premium

Fiche Google My Business

Cession des droits

Frais de déplacement

Prise des photos

Pages supplémentaires

License de exclusivite(6 mois)

Prise de vue drone

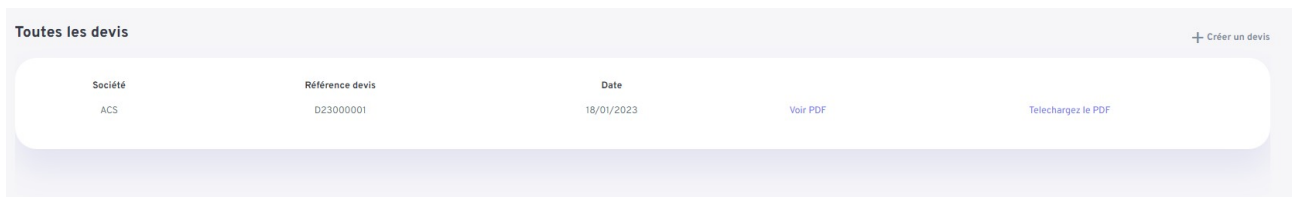
Mot Clé

Envoyer

Pages devis, facture et avoir.

Ces 3 pages ont leurs propres fichiers dans «Models» et «Views», comme Devis.php, Facture.php, Avoir.php.

Ces fichiers ont de nombreuses fonctions en commun, qu'il s'agisse d'afficher toutes les données de la base de données dans un tableau, avec les options pour créer de nouvelles données, afficher le PDF ou télécharger le PDF.



Société	Référence devis	Date	Voir PDF	Téléchargez le PDF
ACS	D23000001	18/01/2023		

Les fonctions qui se ressemblent sont des fonctions comme la création d'un devis, facture... à la base de données, la différence c'est que lors de la création d'un devis on commence par choisir un client, puis un ou plusieurs services

Créer un devis

Choisis le client
Client

- Services
- ☐ Création Site Vitrine
 - ☐ Hébergement de votre site internet
 - ☐ Maintenance préventive et curative
 - ☐ Assistance technique à la demande
 - ☐ Création site E-Commerce
 - ☐ Pack Gold
 - ☐ Pack Silver
 - ☐ Installation Yoast Premium
 - ☐ Création de fiche google my business
 - ☐ Cession des droits
 - ☐ Frais déplacement
 - ☐ Prise des photos
 - ☐ Pages supplémentaires

Envoyer

Au lieu de créer une facture, nous devons choisir le devis, nous aurons également des options pour ajouter une réduction de prix ou un acompte, , pour la réduction nous utilisons la formule:
« Montant total - (Pourcentage/100 * Montant total) »

Créer une facture

Choisis le client
Client

Réduction %

0

Acompte

0

Envoyer

Maintenant, pour créer un avoir, nous devons faire établir une facture.

Créer une avoir

Choisis la facture
Client

Envoyer

Une autre fonction courante serait la création d'un des 3, chacun doit avoir une référence, la référence aurait une lettre plus l'année plus l'id de l'élément créé, par exemple pour un devis, sa référence serait

id_devis	ref_devis
1	D23000001

Pour définir la lettre et l'année, j'ai créé une variable qui contient la lettre spécifique plus la fonction php «date()», la fonction «date()» est d'obtenir la date actuelle, mais nous ne voulons que l'année et les 2 derniers chiffres, donc nous avons simplement nous utilisons «date('y')».

```
$ref_letter = 'D'.date('y');
```

puis pour obtenir l'id, j'ai créé une fonction pour rechercher dans ma base de données le prochain id de la table à choisir.

```
function getNextIncrement($table) {  
    $sql = $this->connect()->prepare("SELECT AUTO_INCREMENT FROM information_schema.TABLES WHERE TABLE_SCHEMA =  
    $sql->execute();  
    $result = $sql->fetch();  
  
    return $result[0];  
}
```

ensuite ce que l'on fait c'est de tout mettre ensemble dans la variable (\$ref), la fonction « sprintf() » sert à retourner des chaînes de texte, c'est à compléter la référence avec les nombres 0.

```
$increment = $this->getNextIncrement('devis');  
  
$ref = $ref_letter.sprintf("%06d", $increment);
```

D'autres fonctions en commun seraient show() qui servira à montrer à l'utilisateur chaque devis/facture/avoir ou la fonction delete() qui servira à les supprimer.

L'unique caractéristique exclusive est de devis, c'est la création de colonnes dans la table de la base de données, je crée une boucle de chaque case cochée pour créer la colonne « service_devis » plus l'index de ma boucle, par exemple, si je coche 3 cases, cette fonction créera « service_devis0 », « service_devis1 », etc, qui servira plus tard à ajouter les services.

```

for ($i=0; $i < count($checkbox); $i++) {
    $col = $this->connect()->prepare("ALTER TABLE devis ADD service_devis".$i." VARCHAR(255) NULL");
    $col->execute();
}

$insert = $this->connect()->prepare("INSERT INTO devis (ref_devis, id_client, date_devis) VALUES (:ref, :id, :dat)");

$insert->bindParam(':ref', $ref, PDO::PARAM_STR);
$insert->bindParam(':id', $fetchId['id'], PDO::PARAM_STR);
$insert->bindParam(':dat', $date, PDO::PARAM_STR);

$insert->execute();

```

puis j'ai créé une boucle de chaque case cochée, pour mettre à jour mes colonnes nouvellement créées, ici je vais chercher le dernier id créé pour pouvoir localiser mon devis et pouvoir le mettre à jour, ensuite chaque donnée saisie dans la base sera le contenu de mes cases à cocher.

```

$getRef = $this->connect()->prepare("SELECT MAX(id_devis) FROM devis");
$getRef->execute();
$fetchRef = $getRef->fetch();

for ($i=0; $i < count($checkbox); $i++) {
    $upd = $this->connect()->prepare("UPDATE devis SET service_devis".$i." = '". $checkbox[$i]."' WHERE id_devis = ".$fetchRef[0]."");
    $upd->execute();
}

```

FPDF

Pendant que je créais les fonctions de base de ma page il me manquait un outil pour pouvoir générer des pdf, dans ma recherche j'ai trouvé FPDF.

Qu'est-ce que FPDF?

FPDF est une librairie PHP qui permet de générer des documents PDF directement depuis PHP. Le F de FPDF signifie Free (gratuit): vous pouvez l'utiliser à n'importe quelle fin et le modifier selon vos besoins.

Comment utiliser FPDF?

Le format général pour écrire une page en PDF serait le suivant :

```
<?php
require('fpdf.php');

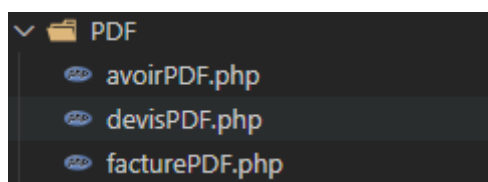
$pdf=new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',16);
$pdf->Cell(40,10,'Hello World !');
$pdf->Output();
?>
```

Source: https://techno.saint-medard-en-jalles.fr/techno/kiosque/administration/web_v7/fpdf/tutoriel/tuto1.htm

- La première chose que nous faisons est d'inclure la bibliothèque fpdf.php
- Dans la ligne « \$pdf=new FPDF(); » ce que nous faisons est de créer l'objet FPDF.
-
- Dans la ligne \$pdf->Cell(); Nous commençons à écrire le contenu de la page.
- Avec SetFont(); nous formatons le texte en disant le type de police, s'il est en gras ou non, et la taille de la police.
- Dans la ligne \$pdf->AddPage(); nous ajoutons une page
- La dernière ligne \$pdf->Output(); ce qu'il fait est de fermer le fichier et de l'envoyer au navigateur.

Quelle méthode ai-je utilisé pour créer un PDF

Après avoir téléchargé le dossier, je l'ai intégré dans la racine de mon dossier principal et en même temps j'ai commencé à apprendre à l'utiliser, en l'occurrence j'ai créé 3 modèles « DevisPDF.php », « FacturePDF.php » et « AvoirPDF.php » dans le dossier Models, ensuite j'ai créé un dossier appelé PDF, où il servirait à appeler mes modèles, contient 3 fichiers portant le même nom que mes modèles



Dossier PDF

Contrairement au format général, pour créer une page PDF, j'ai appelé mes modèles PDF avec leurs fonctions dans les fichiers de ce dossier.

```
<?php
require('../Models/DevisPDF.php');
$pdf=new PDF;
$devis =new Devis;
$getD = $devis->getDevis($_GET['ref']);

$pdf->addPage();
$pdf->pageSiteInternet();
$pdf->pageHebergement();
$pdf->pageOption();
$pdf->pageOption2();
$pdf->pageMentions();
$pdf->clientPage();
$pdf->legalesPage();

if(isset($_GET['download']) && $_GET['download'] == 'D'){
    $pdf->Output('D', 'devis-'. $getD['ref_devis'].'.pdf', TRUE);
}else{
    $pdf->Output('I', 'devis-'. $getD['ref_devis'].'.pdf', TRUE);
}


?> You, last month • creation pdf ...
```

Modèles PDF

Dans cette dossier mes modèles contiennent la structure de mes PDF, la structure de la facture et l'avoir sont pratiquement les mêmes, mais celle de mon devis est totalement différente. bien que dans le cas différent de la structure, certaines fonctions soient similaires, par exemple il y a "getDevis()" qui sert à récupérer les informations de mon devis dans la base de données à partir du lien dans mon navigateur.

```
public function getDevis($devis){
    $findData = $this->connect()->prepare("SELECT * FROM devis WHERE ref_devis = :ref");
    $findData->bindParam(':ref', $devis, PDO::PARAM_STR);
    $findData->execute();
    $fetch = $findData->fetch();
    return $fetch;
}
```

Cette fonction servira à remplir automatiquement mon PDF, soit sur la page principale où j'aurai besoin des données du client, soit sur les pages de service, où j'aurai besoin de savoir de quels services mon "Devis" dispose.


<div data-bbox="287 470 383 582">  </div> <div data-bbox="287 593 383 638"> <p>Devis D23000001 du 18/01/2023 (devis valable 1 mois)</p> </div> <div data-bbox="526 481 718 560"> <p>122 Rue Georges Trouillot - 39000 Lons-le-Saunier Tel: +33 (0)9 82 40 86 00 www.agencelabo.fr</p> </div> <div data-bbox="526 582 758 627"> <p>Siret: 85007317200031 - APE: 5913B - RCS: B 850 073 172 - N° TVA intracom: FR94850073172 - SASU au Capital de: 840€</p> </div> <div data-bbox="526 716 574 739"> <p>Alain</p> </div> <div data-bbox="287 929 734 974"> <p>À l'attention de l'entreprise ACS, Dans le cadre de votre réflexion sur l'ergonomie et le design de votre site web, vous nous avez sollicités afin de vous accompagner dans la refonte de votre site internet.</p> </div> <div data-bbox="287 1164 742 1176"> <p>Siret : 85007317200031 - APE : 5913B- RCS : B 850 073 172 - N° TVA intracom : FR94850073172 - SASU au Capital de : 840€</p> </div>	<table> <tr> <th>SITE INTERNET</th><th>MONTANT</th></tr> <tr> <td> <p>OFFRE DE BASE:</p> <ul style="list-style-type: none"> • Mise en place d'un site internet vitrine responsive (adaptable tablette, mobile) basé sur un template existant. • Réflexion sur l'ergonomie et la navigation. • Mise en place des éléments graphiques du site d'après votre charte graphique et en fonction des contraintes liées au web responsive. • Réalisation des ajustements sur la ligne graphique (5 allers/retours maximum). • Validation de la ligne graphique. • Mise en place de 10 pages à partir du contenu texte, photo, vidéo fourni par vos soins (dont les pages mentions légales obligatoires et informations concernant le responsable des données pour le RGPD) d'après l'arborescence présentée ci-dessous. • Réalisation de 5 allers/retours pour les ajustements sur le contenu (le graphisme reste inchangé). • Validation des contenus. • Installation et configuration CMS Wordpress (nécessaire pour la gestion du contenu du site par vos soins : textes, photos, vidéos). • Intégration HTML5/CSS3/Responsive. • Mise en place des fonctionnalités suivantes : <ul style="list-style-type: none"> - Slider photo, sur la page d'accueil. - Barre de recherche, pour insérer des mots-clés. - Formulaire de prise de contact, pour votre recrutement. - Google Analytics, pour gérer vos statistiques. • Construction du site en fonction de l'optimisation du référencement (sitemap.xml automatique, gestion des balises SEO, rich snippets). Soumission du site à Google. • Mise en ligne. <p>Cette prestation correspond à la réalisation et à la mise en ligne d'un site internet dont le fonctionnement est total au moment de votre validation après mise en ligne. Nous compensons les bugs que vous pourriez constater dans un délai de 2 mois après la mise en ligne(hors piratage du site). Tout dysfonctionnement constaté après ce délai de 2 mois pourra être corrigé mais fera l'objet d'une prestation en sus (sur devis). Aucune garantie de positionnement pour le référencement.</p> <p>Délai de réalisation: à convenir</p> </td><td> <p>1999€ TTC</p> </td></tr> </table> <div data-bbox="813 1164 1276 1176"> <p>Siret : 85007317200031 - APE : 5913B- RCS : B 850 073 172 - N° TVA intracom : FR94850073172 - SASU au Capital de : 840€</p> </div>	SITE INTERNET	MONTANT	<p>OFFRE DE BASE:</p> <ul style="list-style-type: none"> • Mise en place d'un site internet vitrine responsive (adaptable tablette, mobile) basé sur un template existant. • Réflexion sur l'ergonomie et la navigation. • Mise en place des éléments graphiques du site d'après votre charte graphique et en fonction des contraintes liées au web responsive. • Réalisation des ajustements sur la ligne graphique (5 allers/retours maximum). • Validation de la ligne graphique. • Mise en place de 10 pages à partir du contenu texte, photo, vidéo fourni par vos soins (dont les pages mentions légales obligatoires et informations concernant le responsable des données pour le RGPD) d'après l'arborescence présentée ci-dessous. • Réalisation de 5 allers/retours pour les ajustements sur le contenu (le graphisme reste inchangé). • Validation des contenus. • Installation et configuration CMS Wordpress (nécessaire pour la gestion du contenu du site par vos soins : textes, photos, vidéos). • Intégration HTML5/CSS3/Responsive. • Mise en place des fonctionnalités suivantes : <ul style="list-style-type: none"> - Slider photo, sur la page d'accueil. - Barre de recherche, pour insérer des mots-clés. - Formulaire de prise de contact, pour votre recrutement. - Google Analytics, pour gérer vos statistiques. • Construction du site en fonction de l'optimisation du référencement (sitemap.xml automatique, gestion des balises SEO, rich snippets). Soumission du site à Google. • Mise en ligne. <p>Cette prestation correspond à la réalisation et à la mise en ligne d'un site internet dont le fonctionnement est total au moment de votre validation après mise en ligne. Nous compensons les bugs que vous pourriez constater dans un délai de 2 mois après la mise en ligne(hors piratage du site). Tout dysfonctionnement constaté après ce délai de 2 mois pourra être corrigé mais fera l'objet d'une prestation en sus (sur devis). Aucune garantie de positionnement pour le référencement.</p> <p>Délai de réalisation: à convenir</p>	<p>1999€ TTC</p>
SITE INTERNET	MONTANT				
<p>OFFRE DE BASE:</p> <ul style="list-style-type: none"> • Mise en place d'un site internet vitrine responsive (adaptable tablette, mobile) basé sur un template existant. • Réflexion sur l'ergonomie et la navigation. • Mise en place des éléments graphiques du site d'après votre charte graphique et en fonction des contraintes liées au web responsive. • Réalisation des ajustements sur la ligne graphique (5 allers/retours maximum). • Validation de la ligne graphique. • Mise en place de 10 pages à partir du contenu texte, photo, vidéo fourni par vos soins (dont les pages mentions légales obligatoires et informations concernant le responsable des données pour le RGPD) d'après l'arborescence présentée ci-dessous. • Réalisation de 5 allers/retours pour les ajustements sur le contenu (le graphisme reste inchangé). • Validation des contenus. • Installation et configuration CMS Wordpress (nécessaire pour la gestion du contenu du site par vos soins : textes, photos, vidéos). • Intégration HTML5/CSS3/Responsive. • Mise en place des fonctionnalités suivantes : <ul style="list-style-type: none"> - Slider photo, sur la page d'accueil. - Barre de recherche, pour insérer des mots-clés. - Formulaire de prise de contact, pour votre recrutement. - Google Analytics, pour gérer vos statistiques. • Construction du site en fonction de l'optimisation du référencement (sitemap.xml automatique, gestion des balises SEO, rich snippets). Soumission du site à Google. • Mise en ligne. <p>Cette prestation correspond à la réalisation et à la mise en ligne d'un site internet dont le fonctionnement est total au moment de votre validation après mise en ligne. Nous compensons les bugs que vous pourriez constater dans un délai de 2 mois après la mise en ligne(hors piratage du site). Tout dysfonctionnement constaté après ce délai de 2 mois pourra être corrigé mais fera l'objet d'une prestation en sus (sur devis). Aucune garantie de positionnement pour le référencement.</p> <p>Délai de réalisation: à convenir</p>	<p>1999€ TTC</p>				

Ensuite, il y a la structure de la facture et l'avoir que c'est la même, y compris la fonction « getDevis() », qui dans ce cas changerait le nom dudit fichier «getFacture()»...

Maintenant que j'ai expliqué les différentes structures, je vais maintenant vous expliquer la composition des pages PDF.

Facture F23000001

19/01/2023



Emetteur
LE LABO
122 Rue Georges Trouillot
09 82 40 86 00
contact@agencelabo.fr
France
SIRET: ██████████

Destinataire
Société: ACS
Adresse: 2 rue de monaigu, Lons le Saunier,
39000
Pays: France
Numéro de SIRET: 21215103645

LE LABO
IBAN: ██████████
BIC: ████████

Acompte de: 0€

Service/Description	Prix
Création Site Vitrine Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500	1999€
Hébergement de votre site internet Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page	300€
Maintenance préventive et curative Contrairement à une opinion répandue, le Lorem Ipsum n'est pas simplement du texte aléatoire. Il trouve ses racines dans une oeuvre de la littérature latine	1000€

TVA non applicable, art. 293 B du CGI

TOTAL TTC: 3299€

Siret: 85007317200031 - APE: 5813B - RCS: B 850 073 172 - N° TVA intracomm: FR94850073172 - SASU au Capital de: 840€

- Header():

Dans la fonction "header()" je vais faire l'entête de mon PDF, il faut préciser que dans cette fonction j'ai créé une condition pour que mon entête n'apparaisse que sur la première page de mon PDF, normalement il contiendra les informations de le client, l'entreprise ou les deux.

```
function Header() {
    if ( $this->PageNo() === 1 ) {
```

- Toutes les pages:

Pour chaque page créer une fonction en php, normalement le seul PDF avec de nombreuses pages est « DevisPDF.php », mais dans le cas de « FacturePDF.php » ou « AvoirPDF.php » ils peuvent changer, par exemple si dans notre Devis il y a beaucoup de services, lors de

l'affichage des services sur la facture, cela créera automatiquement une autre page à suivre.

DevisPDF est composé avec une fonction de chaque service sélectionné ci-dessus et créera une page si ce service existe. Pour cela j'ai créé une fonction dans mon modèle « Devis.php » appelée « getBool() » qui donnera une liste avec toutes les informations du Devis sélectionné avec sa référence

```
public function getBool($ref){
    $selectDevis = $this->connect()->prepare("SELECT * FROM devis WHERE ref_devis = :ref");
    $selectDevis->bindParam(':ref', $ref, PDO::PARAM_STR);
    $selectDevis->execute();
    $fetch = $selectDevis->fetch(PDO::FETCH_ASSOC);

    $select = $this->connect()->prepare("SELECT mot_cle FROM services WHERE nom_service = :d");
    $array = array();
    foreach ($fetch as $value) {

        $select->bindParam(':d', $value, PDO::PARAM_STR);
        $select->execute();

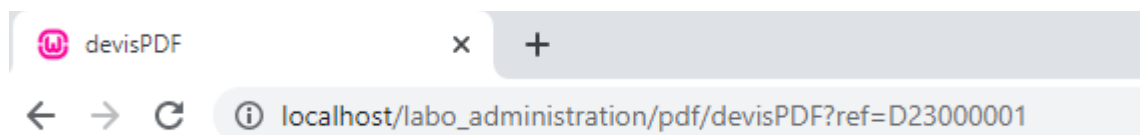
        $fetchAll = $select->fetch(PDO::FETCH_ASSOC);

        if($fetchAll != false){
            $array[] = $fetchAll['mot_cle'];
        }
    }

    return $array;
}
```

Ensuite

j'appelle cette fonction dans DevisPDF.php et lui donne le numéro de référence du lien de ma page



Ensuite, j'utilise la fonction PHP `array_search()` pour rechercher les services avec les mots-clés de ma page PDF actuelle.

```
$getBool = $devis->getBool($getD['ref_devis']);

$array = $getBool;

$boolean = array_search('Site Vitrine', $array);
$booleanCommerce = array_search('E-Commerce', $array);
$booleanCommuneGold = array_search('Gold', $array);
$booleanCommuneSilver = array_search('Silver', $array);
```

- Footer()

Cette fonction permet de créer le pied de page, le pied de page apparaîtra sur chaque page générée.

```
function Footer() { ...  
$this->SetFont('Helvetica','I',9);  
$this->SetY(-12);  
$this->cell(0,5, utf8_decode('Siret : - APE : - N° TVA intracom : - SASU au Capital de : 0').EURO, 0, '',  
'C');  
}
```

Siret : ████████ - APE : ████████ - N° TVA intracom : ████████ au Capital de : ████████€

Description de la veille

J'ai fait de la veille technologique en relation avec la génération des PDF, j'ai trouvé plusieurs méthodes comme « WKHTMLtoPDF » qui est un outil en ligne de commande qui permet de générer un document PDF à partir d'une page HTML ou « HTML2PDF » c'est une librairie faite pour aider à la création de fichiers PDF, pas pour convertir directement une page HTML.

Source «WKHTMLtoPDF »: <https://wkhtmltopdf.org/>

Source « HTML2PDF »: <https://www.html2pdf.fr/>

Au final, j'ai opté pour « FPDF » car il n'a pas besoin d'extension et est compatible avec au moins PHP 5.1 et est compatible avec PHP 7 et PHP 8.

Source « FPDF »: <http://www.fpdf.org/>

Étant donné que FPDF est gratuit et que d'autres programmeurs peuvent ajouter de nouvelles fonctions, j'ai commencé une recherche de nouvelles fonctions qui pourraient m'être utiles.

```
function WriteHTML($html)
{
    //HTML parser
    $html=str_replace("\n", ' ', $html);
    $a=preg_split('/<(.*?)>/U', $html, -1, PREG_SPLIT_DELIM_CAPTURE);
    foreach($a as $i=>$e)
    {
        if($i%2==0)
        {
            //Text
            if($this->HREF)
            {
                $this->PutLink($this->HREF, $e);
            }
            elseif($this->ALIGN=='center')
            {
                $this->Cell(0, 5, $e, 0, 1, 'C');
            }
            else
            {
                $this->Write(5, $e);
            }
        }
        else
        {
            //Tag
            if($e[0]=='/')
            {
                $this->CloseTag(strtoupper(substr($e, 1)));
            }
            else
            {
                //Extract properties
                $a2=explode(' ', $e);
                $tag=strtoupper(array_shift($a2));
                $prop=array();
                foreach($a2 as $v)
                {
                    if(preg_match('/([^\s]*)=("[^\"]*"|' . "\'" . "\'")/' , $v, $a3))
                    {
                        $prop[strtoupper($a3[1])]=$a3[2];
                    }
                }
                $this->OpenTag($tag, $prop);
            }
        }
    }
}
```

J'ai fini par trouver la fonction writeHTML, qui est utilisée pour générer des chaînes de texte écrites comme s'il s'agissait de html, me permettant d'écrire :

```
$this->writeHTML("<b>La recherche et l'achat de photographies et images sur banques d'images </b>")
```

ce qui permettra à mon texte de ressembler à ceci: « **La recherche et l'achat de photographies et images sur banques d'images** »

Description d'une situation ayant nécessité une recherche anglophone

Durant mon stage j'ai dû faire beaucoup de recherches sur comment faire les choses, en règle générale je fais toujours ce que je recherche en anglais, même si la documentation FPDF avait une traduction en français ou en espagnol, je la liras en anglais. Cela me facilite la tâche lors de la recherche dans une partie où j'ai eu des difficultés.

Voici un exemple en anglais pour comprendre le modèle MVC.

Textes en anglais

The Model

The model defines what data the app should contain. If the state of this data changes, then the model will usually notify the view (so the display can change as needed) and sometimes the controller (if different logic is needed to control the updated view).

Going back to our shopping list app, the model would specify what data the list items should contain item, price, etc. and what list items are already present.

The View

The view defines how the app's data should be displayed.

In our shopping list app, the view would define how the list is presented to the user, and receive the data to display from the model.

The Controller

The controller contains logic that updates the model and/or view in response to input from the users of the app.

So for example, our shopping list could have input forms and buttons that allow us to add or delete items. These actions require the model to be updated, so the input is sent to the controller, which then manipulates the model as appropriate, which then sends updated data to the view.

You might however also want to just update the view to display the data in a different format, e.g., change the item order to alphabetical, or lowest to highest price. In this case the controller could handle this directly without needing to update the model.

Textes en français

Le modèle

Le modèle définit les données que l'application doit contenir. Si l'état de ces données change, le modèle notifiera généralement la vue (afin que l'affichage puisse changer selon les besoins) et parfois le contrôleur (si une logique différente est nécessaire pour contrôler la vue mise à jour).

Pour en revenir à notre application de liste de courses, le modèle spécifierait les données que les éléments de la liste devraient contenir - article, prix, etc. et quels éléments de liste sont déjà présents.

La vue

La vue définit comment les données de l'application doivent être affichées. Dans notre application de liste de courses, la vue définirait la manière dont la liste est présentée à l'utilisateur et recevrait les données à afficher à partir du modèle.

Le contrôleur

Le contrôleur contient une logique qui met à jour le modèle et/ou la vue en réponse à l'entrée des utilisateurs de l'application.

Ainsi, par exemple, notre liste de courses pourrait avoir des formulaires de saisie et des boutons qui nous permettent d'ajouter ou de supprimer des éléments. Ces actions nécessitent que le modèle soit mis à jour, de sorte que

l'entrée est envoyée au contrôleur, qui manipule ensuite le modèle selon les besoins, qui envoie ensuite les données mises à jour à la vue.

Cependant, vous pouvez également simplement mettre à jour la vue pour afficher les données dans un format différent, par exemple, modifier l'ordre des articles en ordre alphabétique ou le prix le plus bas au plus élevé. Dans ce cas, le contrôleur pourrait gérer cela directement sans avoir besoin de mettre à jour le modèle.

Conclusion

Le projet à ma disposition remplit plusieurs aspects du développement d'une application web, malgré le fait qu'il y a encore des choses à améliorer, comme l'intégration responsive pour les tablettes et les téléphones mobiles, entre autres, avoir travaillé avec le modèle MVC et faire de l'objet la programmation était totalement enrichissante et amusante.

En tout cas, mon histoire avec Le Labo n'est pas encore terminée, il y a beaucoup de choses à améliorer dans ce projet et aussi la réalisation de nouveaux projets à venir !

Embauché pour le mois de mars 2023 !

Annexes

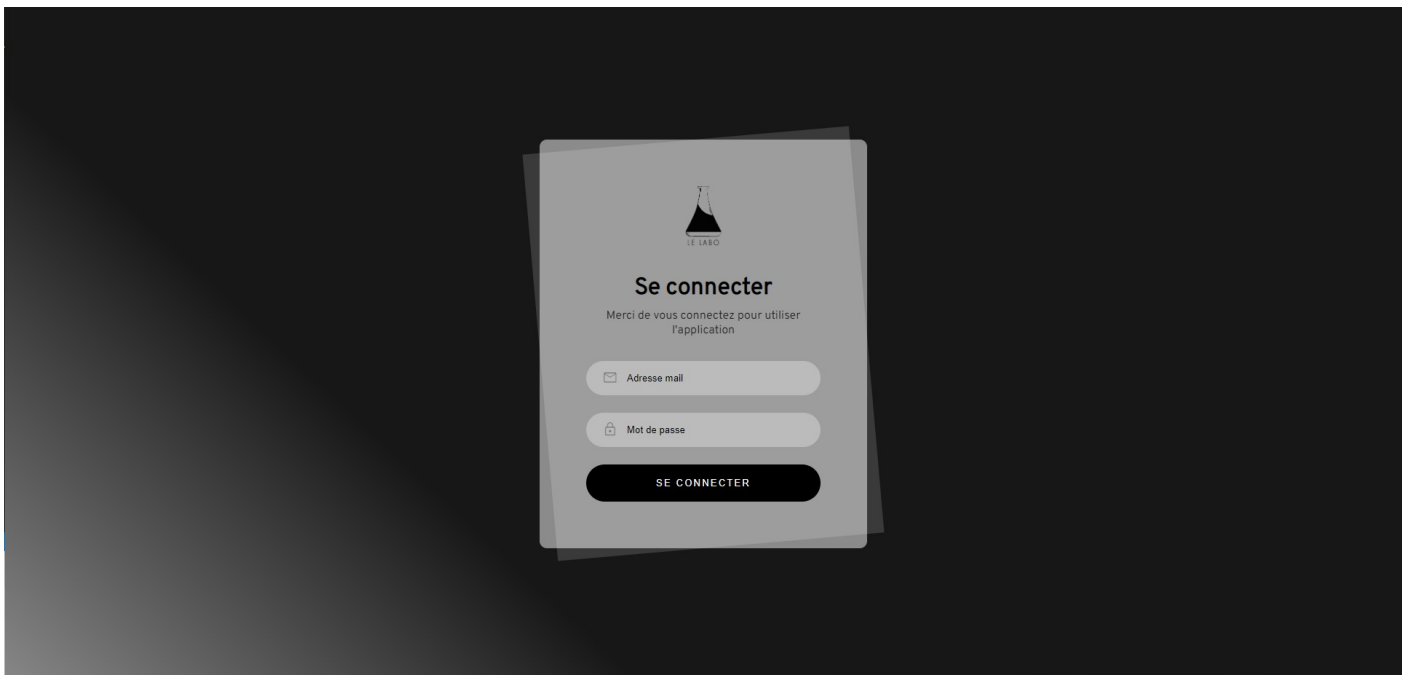
Base de données

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> avoir	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	2,1 kio	72 o
<input type="checkbox"/> client	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	2,1 kio	-
<input type="checkbox"/> devis	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2,2 kio	-
<input type="checkbox"/> facture	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	2,0 kio	-
<input type="checkbox"/> services	★ Parcourir Structure Rechercher Insérer Vider Supprimer	13	MyISAM	latin1_swedish_ci	3,5 kio	64 o
<input type="checkbox"/> users	★ Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	2,3 kio	184 o
6 tables	Somme	19	MyISAM	latin1_swedish_ci	14,2 kio	320 o

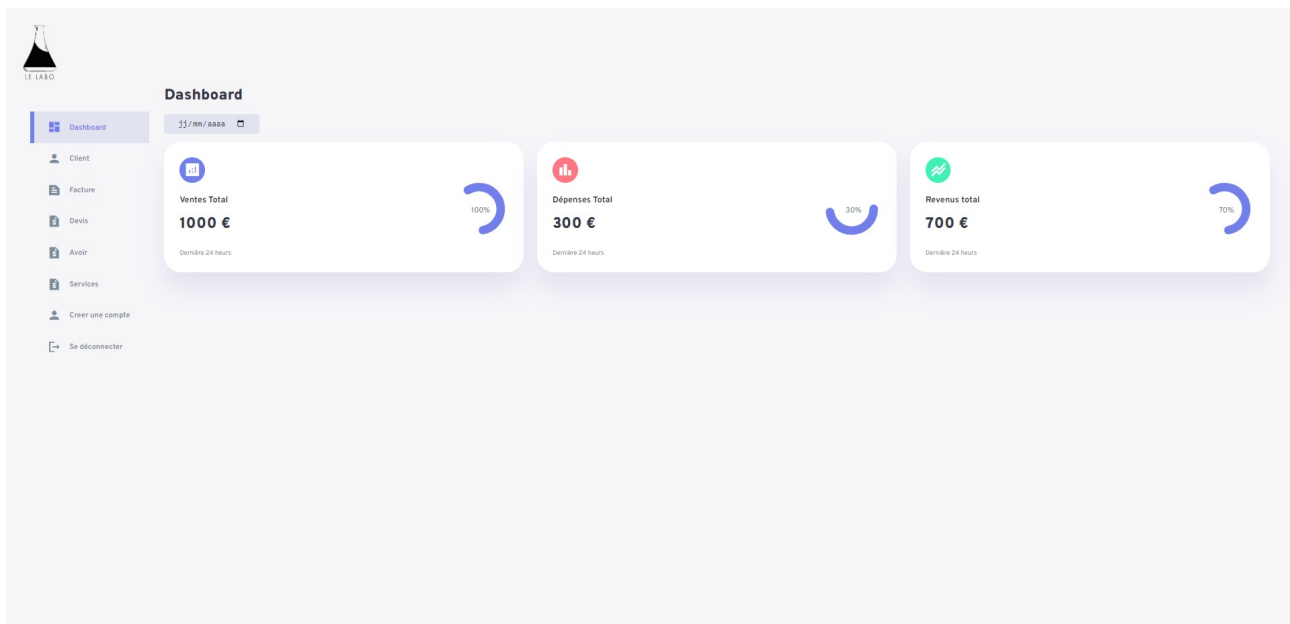
↑ ☐ Tout cocher / Vérifier les tables non optimisées Avec la sélection : ▼

Site web

- Page de connexion



- Page d'accueil



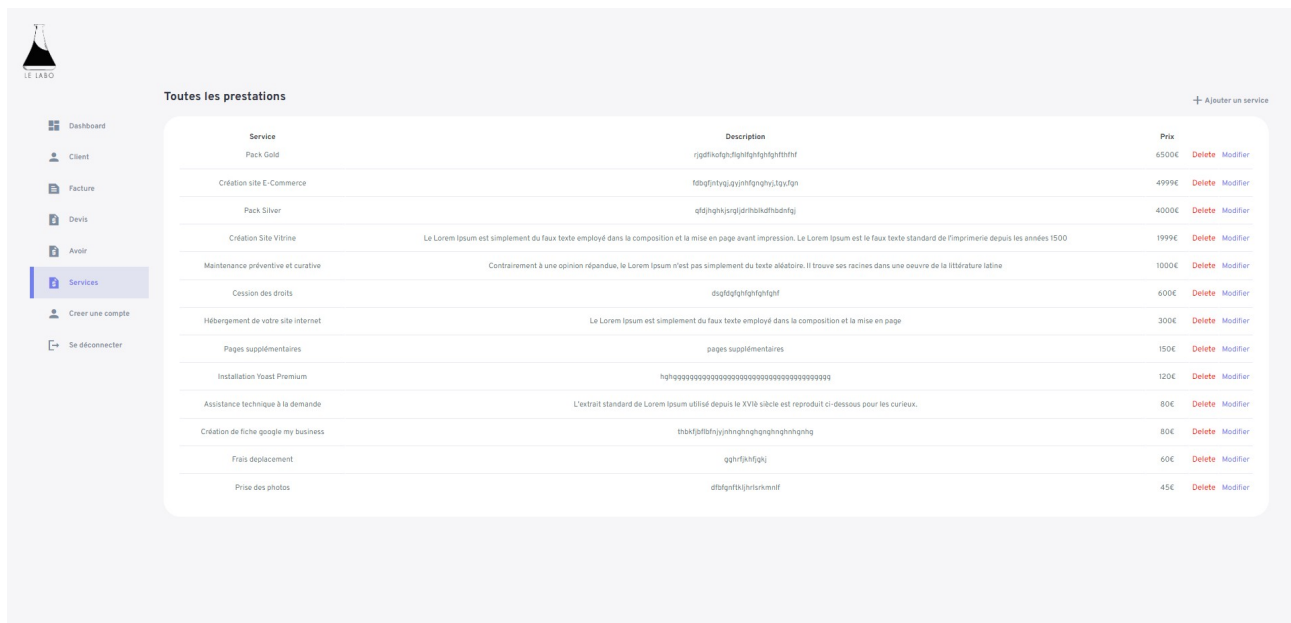
- Pages Client, Devis, Facture, Avoir, Services, Créer un compte

Ces pages sont exactement les mêmes, voici deux exemples

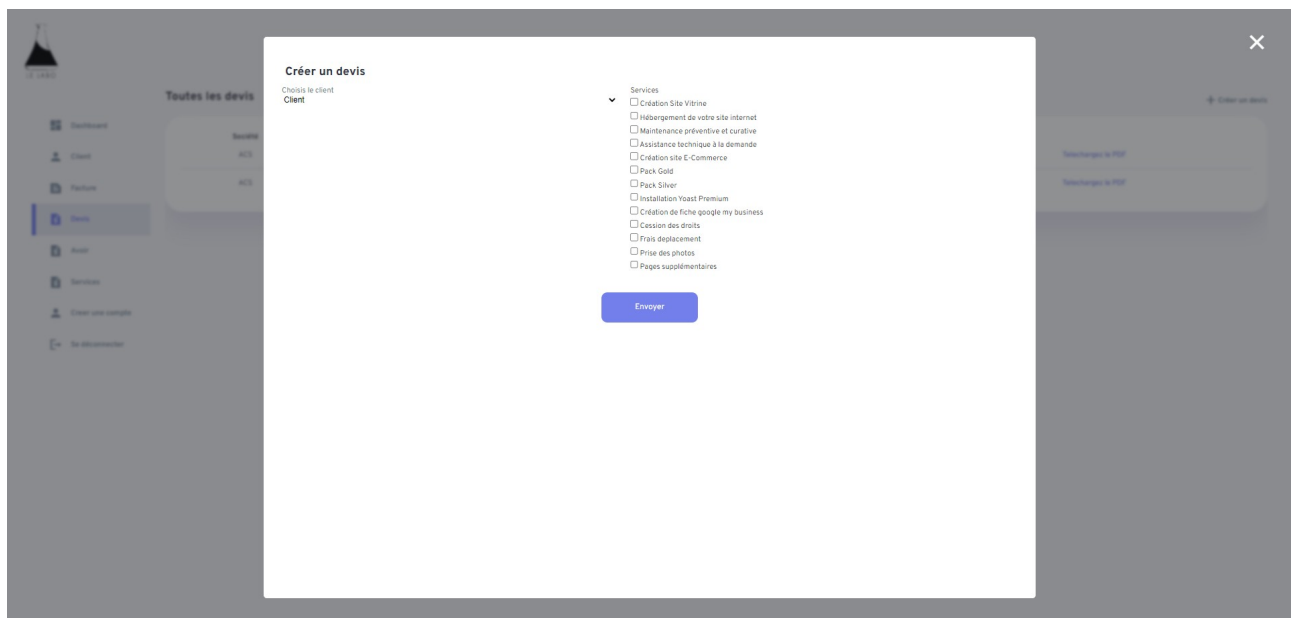
Toutes les devis


+ Créer un devis

Société	Référence devis	Date	Voir PDF	Téléchargez la PDF
ACS	D23000002	20/01/2023	Voir PDF	Téléchargez la PDF
ACS	D23000001	18/01/2023	Voir PDF	Téléchargez la PDF



- Modales





Tous les clients

Client

Facture

Devis

Accueil

Services

Client sans compte

Se déconnecter

+

Ajouter un client

Supprimer

Modifier

SOCIETE

Nom de la société

Nom du dirigeant de la société

Numéro de TVA

SIREN, SIRET

Code NAF/NACE, NOGA

Coordones

Adresse

Code postal

Email

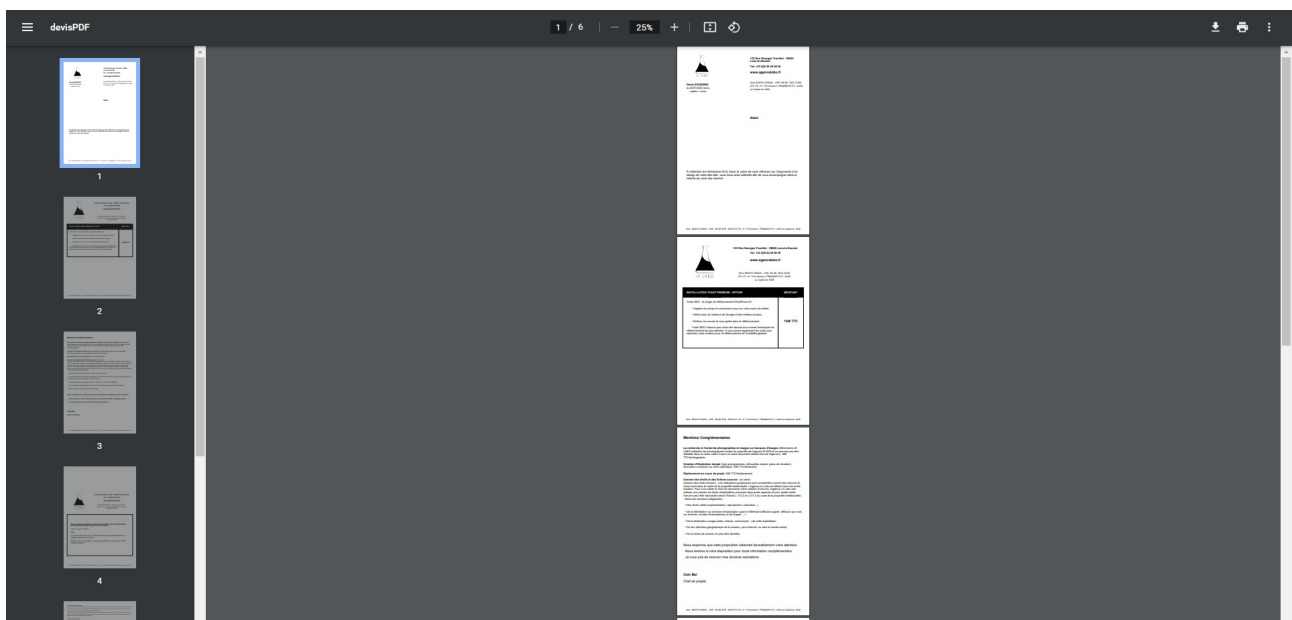
Numéro de téléphone

Ville

Pays

AJOUTER UN CLIENT

- DevisPDF



- FacturePDF

