



Soulard-Pasqual
Nolan
BTS SIO option SLAM
Business School
96 rue Henri Depagneux, 69400 Limas

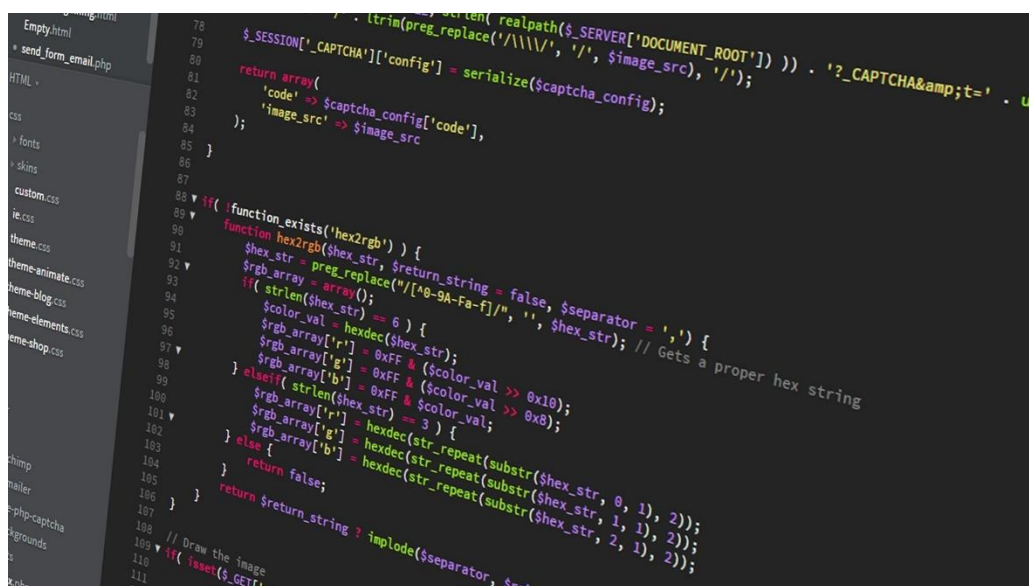
Rapport de stage

La Boite A Marquer

5 Rue de la République, 69580 Sathonay-Camp, France

Développeur applicatif

6 janvier 2025 au 14 février 2025



Remerciement

Je tiens tout d'abord à remercier La Boite A Marquer et particulièrement Jean-Philippe Joulji, mon tuteur, de m'avoir permis de réaliser ce stage de 6 semaines au sein de votre entreprise, dans le cadre du développement applicatif.

Je vous remercie également pour cette expérience extrêmement enrichissante sur le plan professionnel et personnel. J'ai eu l'occasion d'utiliser mes compétences techniques dans un projet concret mais également de les développer et comment les appliquer dans le monde professionnel, tout cela grâce à votre bienveillance.

Je vous remercie également pour votre disponibilité, votre patience, et vos conseils précieux, je me sens désormais mieux préparé à relever de futurs défis professionnels et je suis convaincu que les compétences acquises durant ce stage me seront d'une grande utilité dans ma future carrière professionnelle.

Je vous remercie encore une fois pour cette opportunité que vous m'avez accordée et pour l'accueil chaleureux que vous m'avez réservé.

Je vous prie d'agréer, Monsieur l'expression de mes salutations distinguer.

Cordialement SOULARD PASQUAL Nolan

Sommaire

1. Présentation de l'entreprise

- a. Historique

- b. Activités et mission de l'entreprise

2. Projet

- a. Présentation du projet

- b. Etape pour le finaliser

- c. Visuel de l'application

- d. Difficultés rencontrées et résolutions

3. Conclusion

- a. Apport dans mon parcours académique et professionnel

Présentation de l'entreprise

Historique :

Création et Développement :

La Boite a Marquer a été fondée en 2005, avec une vision axée sur le développement technique et une orientation vers les services multiples. À ses débuts, l'activité principale était centrée sur la gravure, une spécialité que le fondateur a menée en solitaire. Très vite, l'entreprise a évolué pour intégrer une série de services complémentaires, notamment la signalétique, la fabrication de tampons, la duplication de clés, et la photocopie. Ces services ont permis à l'entreprise de se positionner comme un prestataire multiservice de confiance dans son domaine.

Évolution et Transformation

Au fil des années, l'entreprise a su s'adapter aux besoins du marché. Avec l'essor des technologies et l'augmentation des demandes, la gravure et les services informatiques ont commencé à occuper une place de plus en plus importante, nécessitant une restructuration des activités. Par la suite, certaines activités ont été abandonnées pour permettre de se concentrer sur des domaines plus techniques et chronophages, notamment la gravure et l'informatique.

Automatisation et Innovation

Il y a environ huit ans, l'entreprise a franchi une nouvelle étape de son développement en investissant dans l'automatisation. L'objectif était de gagner en efficacité et de réduire le temps consacré aux tâches répétitives. Cette phase de modernisation s'est accompagnée d'une migration progressive des outils technologiques, passant d'Excel à Excel/ASP, puis de ASP/MySQL pour la

gestion commerciale. De plus, la gestion des données a évolué, passant de Visual Basic Access à une architecture plus robuste avec PHP/MySQL, et enfin à Laravel/MySQL.

Défis et Expansion

L'entreprise a démarré dans un garage, un lieu modeste mais suffisant pour les débuts de l'activité. Après dix ans, un local commercial a été acquis, transformé par la suite en atelier à mesure que les besoins augmentaient. Le parcours de l'entreprise a également été marqué par des opportunités professionnelles dans des institutions comme CSNIS 2i et l'URSSAF, qui ont cependant été refusées, probablement en faveur de l'engagement total envers le développement de l'entreprise.

Infrastructure Technologique

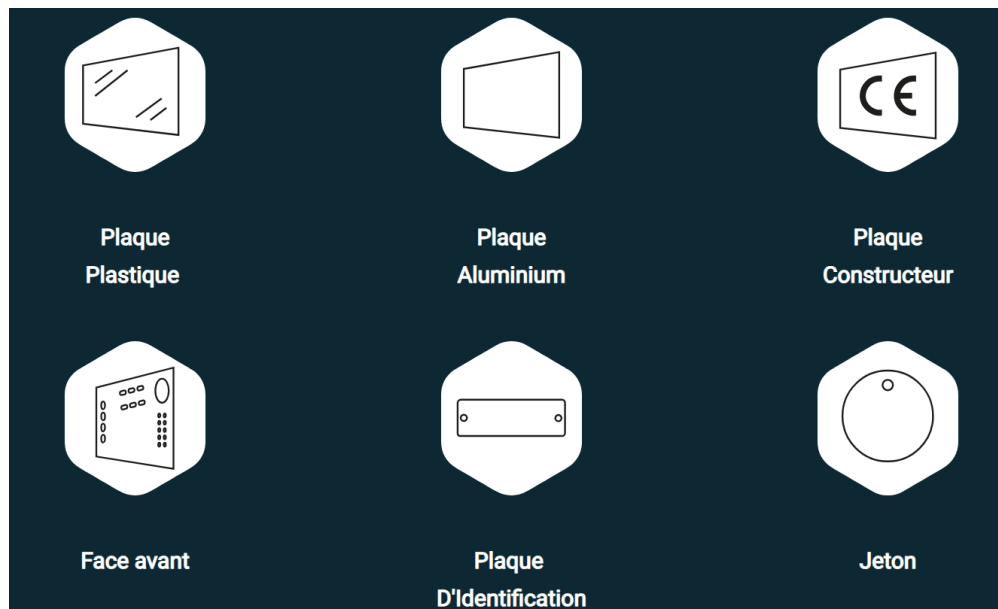
Sur le plan technologique, l'entreprise a progressivement mis en place une infrastructure solide. À l'origine, un serveur web fonctionnant sous Windows NT avec ASP pour l'attaque des bases de données locales en MySQL a été installé. L'entreprise a également mis en place un serveur FTP pour permettre l'accès à distance, et des modules PHP CGI ont été utilisés pour supporter des langages supplémentaires sur le web. Un premier stagiaire en SISR a été recruté pour migrer le serveur en ligne et mettre en place un VPN, permettant ainsi une gestion à distance complète des fichiers.

Objectifs Futurs

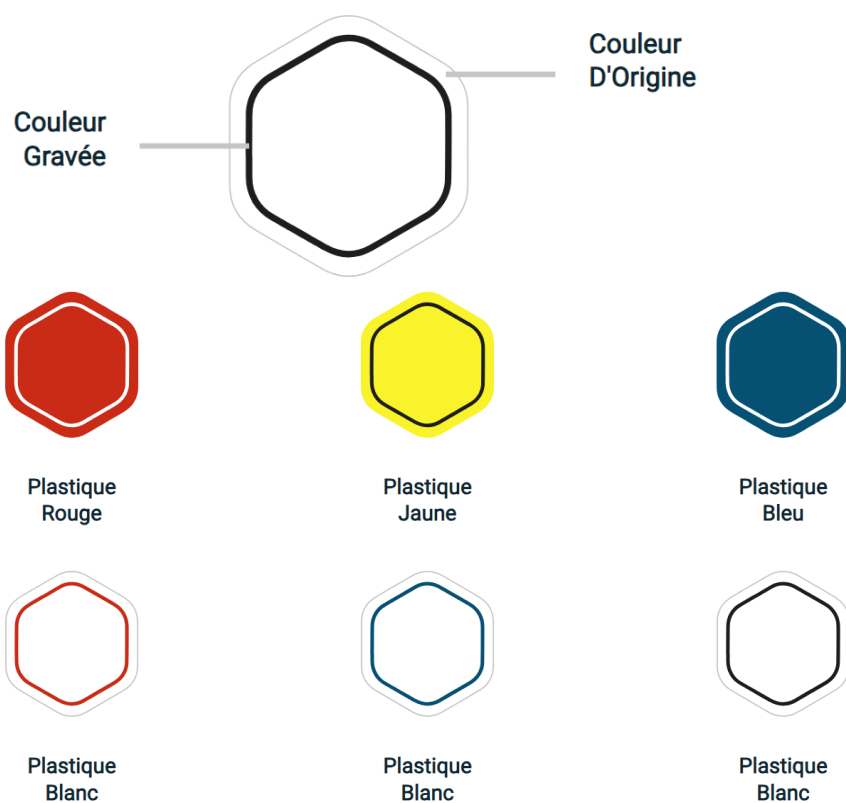
Actuellement, l'un des objectifs clés de l'entreprise est de migrer l'ensemble des systèmes vers Laravel, afin de moderniser et d'optimiser la gestion des plaques, un produit central dans leur offre.

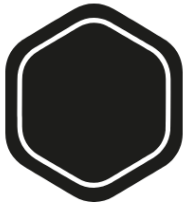
Activités et mission de l'entreprise :

Produit proposé :



Personnalisation proposée :

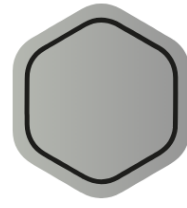




Plastique
Noir



Plastique
Noir



Plastique ou Aluminium
Mat

Matériel :



Gravure
Laser



Impression
UV



Gravure
Mécanique

Projet

Présentation du projet

Dans le cadre d'un projet de développement, j'ai eu à travailler sur l'amélioration et la correction d'un logiciel existant en utilisant le framework Laravel, impliquant des technologies telles que Java, PHP, HTML, CSS et SQL. Mon principal objectif était d'optimiser plusieurs fonctionnalités, notamment la gestion de l'authentification. J'ai corrigé le système de renvoi des erreurs afin d'assurer une meilleure gestion des connexions et des messages d'erreur affichés aux utilisateurs. Une autre amélioration majeure a été l'ajout d'une fonctionnalité permettant l'envoi automatique d'un email à un comptable lorsque son adresse était renseignée, garantissant ainsi une meilleure communication des informations financières. J'ai également corrigé une page dédiée à la gestion des relances, en optimisant son fonctionnement pour qu'elle permette un suivi plus efficace des paiements en attente. Enfin, j'ai développé une nouvelle page permettant de lister tous les règlements des factures, offrant ainsi une meilleure visibilité et un accès simplifié aux informations financières. Ce projet m'a permis de renforcer mes compétences en Laravel et d'approfondir ma maîtrise des différents langages utilisés.

Etape pour le finaliser

Croquis et diagramme

Tout d'abord, j'ai dû analyser le code existant afin de le comprendre et de m'y adapter de la manière la plus optimale possible. Cette étape était essentielle pour identifier les points à améliorer et assurer une intégration fluide des corrections et nouvelles fonctionnalités. Une fois cette analyse effectuée, j'ai entrepris une étude approfondie des besoins afin d'ajouter une table spécifique dans la base de données. Cette table avait pour objectif d'améliorer la gestion de l'historisation des relances, permettant ainsi un suivi plus précis des actions effectuées. Grâce à cette structuration, il était désormais possible de conserver un historique détaillé des relances envoyées, de mieux organiser le traitement des paiements en attente et d'optimiser l'efficacité du processus de recouvrement.

Base de données

	Rgl_Id	Rgl_NumFac	1	Rgl_Date	Rgl_Montant	Rgl_TypReg	Rgl_Ref	Rgl_Banque	Rgl_Commentaire	Rgl_CreatedAt	Rgl_UpdatedAt
--	--------	------------	---	----------	-------------	------------	---------	------------	-----------------	---------------	---------------

Ensuite j'ai fait un jeu de données (que je ne peux pas montrer) afin de faciliter le débogage pour le lien du code vers la base de données.

Contrôleurs

```
public function getListReglementByNumFac($numFacture): array|Collection|string
{
    $reglements = Reglement::where(column: 'Rgl_NumFac', operator: $numFacture)
        ->join(table: 'TypeReglement', first: 'Rgl_TypReg', operator: '=', second: 'TypeReglement.Typ_CodeReference')
        ->get();

    $facture = DB::table(table: 'Facture')
        ->where(column: 'Fac_Num', operator: $numFacture) // Utilisez $numFacture ici
        ->select(columns: 'Fac_TTC')
        ->first();

    if (!$facture) {
        return "Facture introuvable";
    }

    $totalReglements = DB::table(table: 'Reglement')
        ->where(column: 'Rgl_NumFac', operator: $numFacture) // Utilisez $numFacture ici
        ->sum(column: 'Rgl_Montant');

    $montantRestant = $facture->Fac_TTC - $totalReglements;

    // Ajouter le montant restant à chaque règlement
    foreach ($reglements as $reglement) {
        $reglement->montant_restant = $montantRestant;
    }

    return $reglements; // Pas besoin de toJson() ici
}
```

Ici nous avons un exemple de contrôleur qui permet de récupérer la liste des règlements d'une facture donnée et y ajoute le montant restant à payer

HTML/CSS

```
<div class="select">
  <select class="select" id="listMois" onchange="listGestPaiement('doc',this.value,null)">
    <option value='00'>Mois</option>
    <option value='01'>Janvier</option>
    <option value='02'>Février</option>
    <option value='03'>Mars</option>
    <option value='04'>Avril</option>
    <option value='05'>Mai</option>
    <option value='06'>Juin</option>
    <option value='07'>Juillet</option>
    <option value='08'>Août</option>
    <option value='09'>Septembre</option>
    <option value='10'>Octobre</option>
    <option value='11'>Novembre</option>
    <option value='12'>Decembre</option>
  </select>
</div>
<div class="column">
  <div class="select">
    <select class="select" id="listAnneeAvecRelance" onchange="listGestPaiement('doc',null,this.value); "></select>
    <script type="text/javascript" charset="utf-8">listAnneeAvecRelance('R');</script>
  </div>
</div>
<div class="menu-label">
  Client
</div>
<div class="menu-list">
  <li>
    <div class="select">
      <select class="select" id="listClientavecRelance" name="listClientavecRelance" onchange="listGestPaiement('doc',null,this.value)"></select>
      <script type="text/javascript" charset="utf-8">listClientavecRelance();</script>
    </div>
  </li>
</div>
```

Ce code HTML représente un formulaire avec plusieurs listes déroulantes (<select>) permettant de filtrer des données en fonction du mois, de l'année et du client complété en jQuery

API

```
Route::get(uri: 'getListReglement/{numFacture}', action: [App\Http\Controllers\ReglementController::class, "getListReglementByNumFac"]);
```

Ceci nous montre une api en Laravel, établissant une connexion entre une URL et une action de contrôleur. La route est configurée pour répondre aux requêtes GET à l'URI "getListReglement/{numFacture}", où "{numFacture}" est un paramètre dynamique représentant un numéro de facture. L'action associée à cette route est la méthode "getListReglementByNumFac" du contrôleur "ReglementController" situé dans l'espace de noms "App\Http\Controllers". Ainsi, lorsqu'une requête GET est faite à l'URL spécifiée avec un numéro de facture, la méthode "getListReglementByNumFac" du contrôleur "ReglementController" est exécutée, permettant de récupérer et de traiter les données de règlement associées à ce numéro de facture.

JavaScript

jQuery :

```
$('#table_id').on('click', 'tbody tr', function(e) {
    let $row = $(this);
    var file = "../api/getListReglement/" + $row.find('td:first-child').text();

    id = "listeReglement";
    $("#" + id).empty(); //efface la balise identifier par id

    var entete = "<th>id</th><th>Date</th><th>Montant</th><th>Type de Règlement</th><th>Référence</th><th>Banque</th><th>Commentaire</th>";
    message = "<table class='table is-hoverable' id='flexme'><thead><tr>" + entete + "</tr></thead><tbody>";
    $.getJSON(file, function(data) {
        for (let i = 0; i < data.length; i++) {
            let item = data[i];
            if (i === data.length - 1) { // Vérifier si c'est le dernier élément
                let dernier = data.length - 1;
                message = affichageListeReglements(message, item, dernier);
            } else {
                message = affichageListeReglements(message, item, null);
            }
        }
    });

    message += "</tbody><tfoot><tr>" + entete + "</tr></tfoot></table>";
    document.getElementById(id).innerHTML = message;
});
```

Ce code JavaScript utilise jQuery pour détecter un clic sur une ligne (<tr>) d'un tableau (#table_id). Lorsqu'une ligne est cliquée, il extrait la valeur de la première cellule (td:first-child), construit une URL d'API (../api/getListReglement/ + valeur extraite), puis effectue une requête AJAX avec \$.getJSON pour récupérer les données associées. Il vide (empty()) ensuite le contenu d'un élément identifié par listeReglement, puis génère dynamiquement un tableau HTML affichant les règlements récupérés. Chaque élément des données est traité dans une boucle for, et la fonction affichageListeReglements est appelée pour formater les lignes du tableau. Enfin,

le tableau complété est inséré dans l'élément HTML correspondant via `document.getElementById(id).innerHTML = message;`

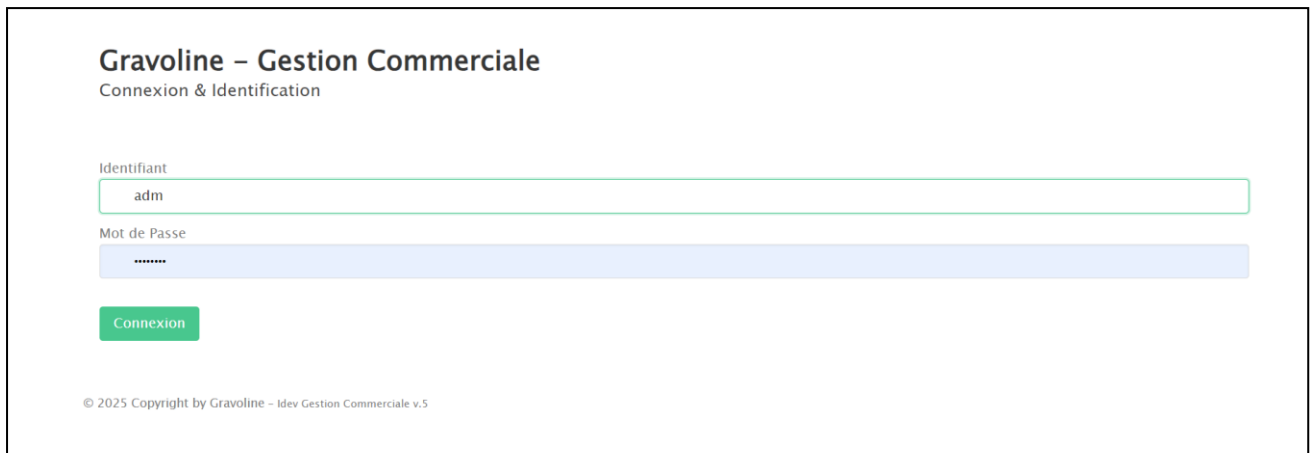
Requête html :

```
function affichageListeReglements(message, item, dernier) {  
    message += "<tr>";  
    message += "<td id='" + item["Rgl_Id"] + "'>" + item["Rgl_Id"] + "</td>";  
    message += "<td id='data1' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + getDateFormatFR(item["Rgl_Date"]) + "</td>";  
    message += "<td id='data2' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["Rgl_Montant"] + "</td>";  
    message += "<td id='data3' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["Rgl_Libelle"] + "</td>";  
    message += "<td id='data4' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["Rgl_Ref"] + "</td>";  
    message += "<td id='data5' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["Rgl_Banque"] + "</td>";  
    message += "<td id='data6' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["Rgl_Commentaire"] + "</td>";  
    if (dernier != null) {  
        message += "<td id='data7' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''>" + item["montant_restant"] + "</td>";  
    } else {  
        message += "<td id='data7' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''></td>";  
    }  
    message += "<td id='data8' + item['Rgl_Id'] + '' name='data3' + item['Rgl_Id'] + ''><div class='field has-addons'>";  
    if (item["Rgl_Montant"] > 0) {  
        message += "<p class='control'><button data-action='lettreRemboursement' id='lr' value='" + item["Rgl_Id"] + "' class='button is-";  
    }  
    message += "<p class='control'><button data-action='supprimer' id='suppr' value='" + item["Rgl_Id"] + "' class='button is-danger is-";  
    message += "<i class='las la-trash-alt la-2x'></i></span></button></p>";  
    message += "<p class='control'><button data-action='modifier' id='modif' value='" + item["Rgl_Id"] + "' class='button is-info is-out-";  
    message += "</div></td>";  
    message += "</tr>";  
    return message;  
}
```

Ceci présente une fonction JavaScript nommée `affichageListeReglements` qui génère du code HTML pour afficher une liste de règlements dans un tableau. Elle prend en paramètres un message HTML existant, un élément de données `item` représentant un règlement et un indicateur `dernier` pour déterminer si l'élément est le dernier de la liste. La fonction construit des lignes de tableau (`<tr>`) avec des cellules (`<td>`) contenant les informations du règlement, telles que l'identifiant, la date, le montant, la référence, la banque et un commentaire. Elle ajoute également des boutons d'action pour chaque règlement, permettant de le modifier, de le supprimer ou de générer une lettre de remboursement. Si l'élément est le dernier de la liste, une cellule supplémentaire est ajoutée pour afficher le montant restant. La fonction renvoie le message HTML mis à jour, prêt à être inséré dans une page web.

Visuel de l'application :

Page authentication



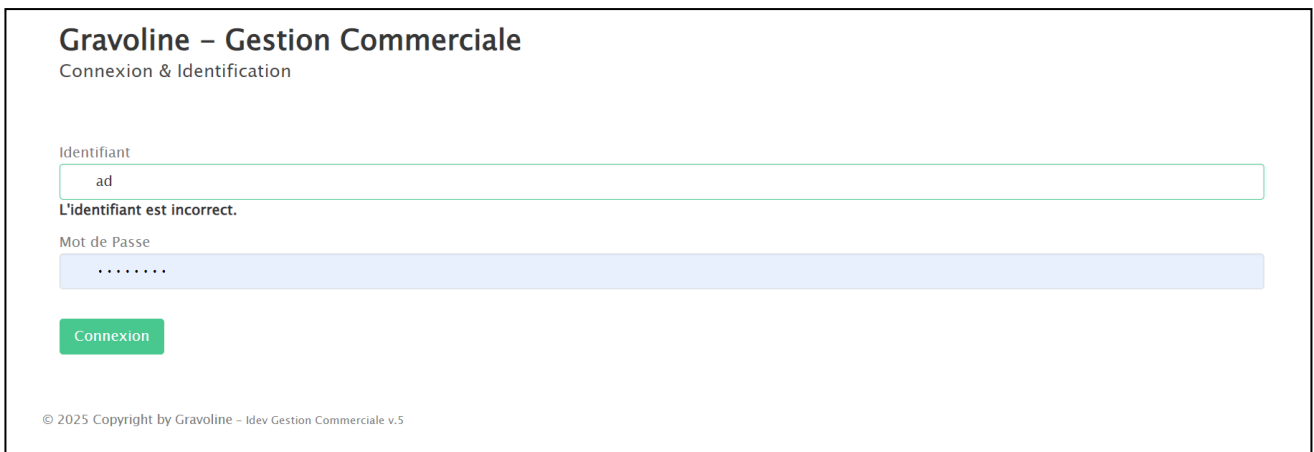
Gravoline – Gestion Commerciale
Connexion & Identification

Identifiant
adm

Mot de Passe

Connexion

© 2025 Copyright by Gravoline – Idev Gestion Commerciale v.5



Gravoline – Gestion Commerciale
Connexion & Identification

Identifiant
ad

L'identifiant est incorrect.

Mot de Passe

Connexion

© 2025 Copyright by Gravoline – Idev Gestion Commerciale v.5

L'image représente une interface de connexion pour un logiciel de gestion commerciale, mettant en lumière le processus d'authentification et la gestion des erreurs. On y trouve un formulaire simple demandant un identifiant et un mot de passe, avec un bouton de soumission "Connexion". La spécificité de cette image réside dans l'affichage d'un message d'erreur clair et concis, "L'identifiant est incorrect", situé directement sous le champ de saisie de l'identifiant. Cette présentation indique que le système effectue une vérification de l'identifiant saisi par rapport à une base de données, et fournit un retour

immédiat à l'utilisateur en cas d'identifiant non reconnu. Cette approche améliore l'expérience utilisateur en fournissant des informations précises sur la nature de l'erreur, facilitant ainsi la correction et l'accès au logiciel.

Mail :

R2501001

Création de Relance en PDF :


 PDF **avec** entete




 PDF **sans** entete





Création des factures associées en PDF :


 Factures en 1 seul PDF **avec** entete







 Factures en 1 seul PDF **sans** entete






Envoyer un mail à *soulardnolan@gmail.com* pour Espace Immobilier Lyonnais


 Message Personnalisé

 Envoyer email

soulardnolan@gmail.com

 Envoyer par Email à

Envoyer au comptable :

 Envoyer email

equinoxnono@gmail.com

L'image décrit une interface de gestion de documents et de communication pour un dossier identifié "R2501001", offrant la possibilité de générer des PDF de relance et de factures (avec ou sans en-tête), d'envoyer des e-mails personnalisés à des clients ou au comptable si l'adresse email est spécifié dans la base de données, et ce, directement depuis l'interface. Les options de

prévisualisation et de téléchargement sont également disponibles, facilitant ainsi la gestion des relances et des communications associées.

Page Relance

← Retour

Relance n°R2501001

Données générales

Client :

Espace Immobilier Lyonnais – "Le Maréchal"▼

Date de relance :

23/01/2025

SDC LE GALIBIER C/O

Nolan

4 Rue Etienne Gagnaire

69100 Villeurbanne

Total

Net TTC en EUR 300.00

Enregistrement de la relance

Détails

Ajouter une ligne au détail

3 ligne(s)

Ordre	N° de fac	Date	Références	Montant	Actions
1	2401001	02/02/2025	DXFGHJKL;MÛ	100	<div><div></div><div></div><div></div></div>

Ceci représente l'interface de gestion d'une relance de paiement pour un client. En haut à gauche, un bouton "Retour" permet de naviguer en arrière, tandis qu'à droite, le numéro de relance "R2501001" est affiché. La section "Données générales" fournit des informations sur le client ("Espace Immobilier Lyonnais"), la date de relance (23/01/2025) et l'adresse du client. À droite, la section "Total" indique le montant total dû (300.00 EUR) et propose un bouton pour enregistrer la relance. En bas, la section "Détails" permet d'ajouter des lignes de détail à la relance. Un tableau affiche les lignes existantes, avec des colonnes pour l'ordre, le numéro de facture, la date, les références et le montant. Des actions sont également disponibles pour chaque ligne (modifier, supprimer, etc.). L'interface est conçue pour faciliter la gestion des relances de paiement, en fournissant une vue d'ensemble des informations pertinentes et en permettant d'effectuer des actions rapides.

Page Relance

Gestion paiement

Documents

Gestion clients

Autre

Admin

Liste des règlements

Tous

toutes les années

Recherche une facture

Rechercher

RAZ

numéro de facture	date de la facture	montant de la facture
2502017	26/02/2025	34
2502014	19/02/2025	25
2502008	11/02/2025	39
2502016	24/02/2025	34
2502015	24/02/2025	43
2502003	07/02/2025	39
2502002	07/02/2025	25
2502013	18/02/2025	19
2502011	14/02/2025	43
2502009	11/02/2025	33
2502010	13/02/2025	36

Recherche dans la liste

id	Date	Montant	Type de Règlement	Référence	Banque	Commentaire	Restant à payer	Actions
107	01/03/2025	34	Espèces	ESPui	CA			<div></div>
108	01/03/2025	34	Annulation			annulation du règlement		<div></div>
111	01/03/2025	34	Espèces	ESPesp	CA		-102	<div></div>
112	02/03/2025	34	Annulation			annulation du règlement		<div></div>

Cette image représente une interface de gestion de règlements, structurée en deux tableaux distincts pour une clarté optimale. Le tableau de gauche liste les factures avec leurs informations essentielles, tandis que le tableau de droite affiche les règlements associés à une facture sélectionnée, offrant ainsi une vue détaillée des paiements. L'interface propose des options de navigation et de filtrage, permettant une recherche rapide et efficace des informations. De plus, des actions sont disponibles pour chaque règlement, telles que la modification et la suppression, facilitant ainsi la gestion des paiements. Cette conception intuitive et fonctionnelle améliore l'expérience utilisateur en simplifiant le suivi des règlements et en fournissant des informations détaillées en un coup d'œil.

Conclusion

Apport dans mon parcours académique et professionnel :

Ce projet m'a permis de consolider mes compétences en Java, PHP, HTML, CSS et SQL à travers la conception et le développement d'une application web de gestion de règlements et d'erreurs. J'ai notamment acquis une expérience significative dans l'usage du framework Laravel, ce stage ayant renforcé mes compétences dans son utilisation, J'ai notamment acquis une expérience significative dans :

- **L'architecture logicielle en Laravel** : En mettant en place une application à plusieurs niveaux, j'ai pu appréhender les interactions entre les différentes couches (présentation, métier, données).
- **Les bases de données relationnelles** : La gestion des données de Règlements à l'aide de SQL m'a permis de maîtriser les requêtes complexes et les optimisations de performances.
- **Les interfaces utilisateur** : La création d'une interface intuitive et ergonomique grâce à HTML, CSS et JavaScript a été un défi que j'ai relevé avec succès.
- **Les méthodes agiles**: L'application a été développée en suivant une méthodologie agile, ce qui m'a permis de m'adapter aux évolutions du projet et de livrer des fonctionnalités de manière itérative.