Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE CHARGÉ DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▸ Meunier
Nom d'usage ▸ *Entrez votre nom d'usage ici.*
Prénom ▸ Lydie
Adresse ▸ 140 Cours Émile Zola, 69100 Villeurbanne

Titre professionnel visé

Développeur Web et Web Mobile

MODALITE D'ACCES :

☒ Parcours de formation
☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen**.

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. Des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. Du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. Des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. De l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- Pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- Un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- Une déclaration sur l'honneur à compléter et à signer ;
- Des documents illustrant la pratique professionnelle du candidat (facultatif)
- Des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

http://travail-emploi.gouv.fr/titres-professionnels

Sommaire

Exemples de pratique professionnelle

Intitulé de l'activité-type n° 1 Développer la partie front-end d'une application web ou web mobile sécurisée	**p.**	**5**
▸ Intitulé de l'exemple n° 1 Installer et configurer son environnement de travail en fonction du projet web ou web mobile	p.	5
▸ Intitulé de l'exemple n° 2 Maquetter des interfaces utilisateur web ou web mobile	p.	9
▸ Intitulé de l'exemple n° 3 Réaliser des interfaces utilisateur statique web ou web mobile	p.	12
▸ Intitulé de l'exemple n° 4 Développer la partie dynamique des interfaces utilisateur web ou web mobile	p.	19
Intitulé de l'activité-type n° 2 Développer la partie back-end d'une application web ou web mobile sécurisée	**p.**	**23**
▸ Intitulé de l'exemple n° 1 Mettre en place une base de données relationnelle	p.	23
▸ Intitulé de l'exemple n° 2 Développer des composants d'accès aux données SQL et NoSQL	p.	27
▸ Intitulé de l'exemple n° 3 Développer des composants métiers côté serveur	p.	32
▸ Intitulé de l'exemple n° 4 Documenter le déploiement d'une application dynamique web ou web mobile	p.	40
Titres, diplômes, CQP, attestations de formation *(facultatif)*	**p.**	43
Déclaration sur l'honneur	**p.**	44
Documents illustrant la pratique professionnelle *(facultatif)*	**p.**	45
Annexes *(Si le RC le prévoit)*	**p.**	46

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1 — Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°1 ▸ Installer et configurer son environnement de travail en fonction du projet web ou web mobile

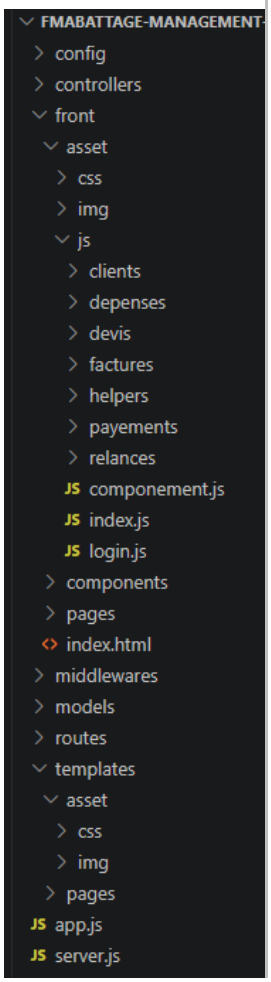
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai installé et configuré un environnement de développement complet afin de réaliser l'application web.
Cette étape a consisté à mettre en place l'ensemble des outils nécessaires au développement, aux tests et au déploiement de l'application.
Le projet a été développé dans un premier temps en environnement local afin de faciliter les phases de test, de correction et d'évolution du code. Une fois l'application stabilisée, elle a été déployée sur un environnement distant.

Mise en place de l'environnement de travail
L'environnement de développement a été structuré autour de plusieurs outils :

- Un IDE de développement pour le codage de l'application (VS Code)
- Java pour la logique serveur
- MySQL pour la gestion de la base de données
- GitHub pour la gestion de version et le suivi du code
- Render pour l'hébergement de l'application
- Aiven pour l'hébergement de la base de données

Cette organisation permet de séparer les environnements de travail, de test et de production.



Architecture du projet
Une architecture structurée a été mise en place afin de séparer les différentes parties de l'application :

- Interface utilisateur
- Logique serveur
- Accès aux données
- Configuration
- Ressources statiques

Cette séparation permet d'améliorer la lisibilité du projet, sa maintenabilité et son évolutivité.

Environnement local et environnement distant

Le développement a été réalisé en deux phases :

- **Phase locale**

Dans un premier temps, l'application a été développée en local afin de tester rapidement les fonctionnalités et corriger les erreurs.

- **Phase de déploiement**

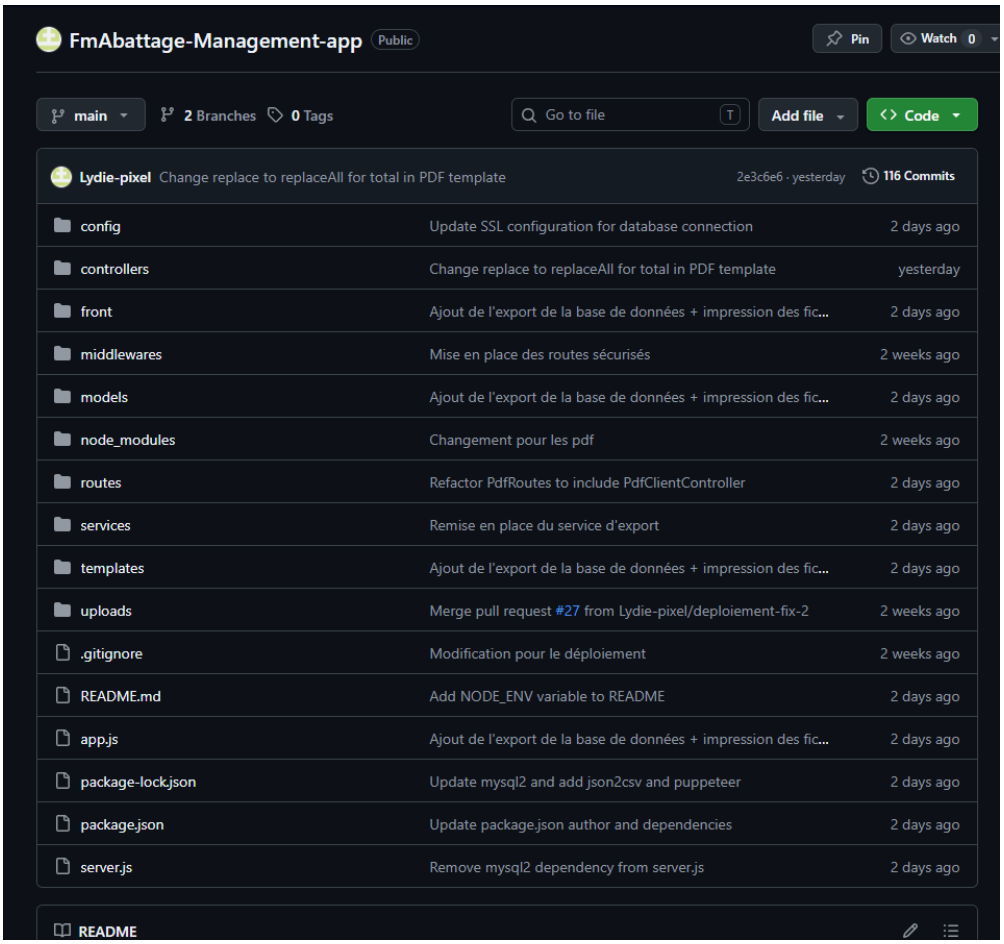
Une fois l'application fonctionnelle, elle a été déployée sur un environnement distant afin de simuler des conditions proches de la production.

Gestion de version

J'ai utilisé GitHub pour :

- Versionner le code source
- Suivre les modifications
- Organiser les différentes évolutions du projet
- Sécuriser l'historique du développement

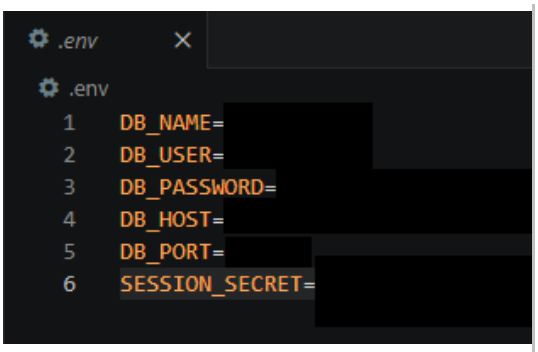
Cette méthode permet de travailler de manière structurée et de revenir à une version antérieure en cas de problème.



Gestion des variables sensibles
Afin de sécuriser l'application, des variables d'environnement ont été utilisées pour stocker les informations sensibles, notamment :

- Les identifiants de base de données
- Les paramètres de connexion

Cela permet d'éviter d'exposer des données sensibles directement dans le code source.



Déploiement de l'application
L'application a été déployée sur Render, tandis que la base de données est hébergée sur Aiven.
Cette architecture permet :

- Un accès distant à l'application
- Une séparation entre données et application
- Une simulation d'environnement de production

2. Précisez les moyens utilisés :

- Java
- MySQL
- IDE de développement
- GitHub
- Render (hébergement applicatif)
- Aiven (base de données)
- Variables d'environnement pour sécurisation des accès

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

La mise en place de cet environnement m'a permis de travailler dans des conditions proches d'un projet professionnel réel. Elle constitue la base technique indispensable au développement de l'application.
Le projet est disponible sur GitHub au lien : https://github.com/Lydie-pixel/FmAbattage-Management-app.git

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°2 ▸ Maquetter des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

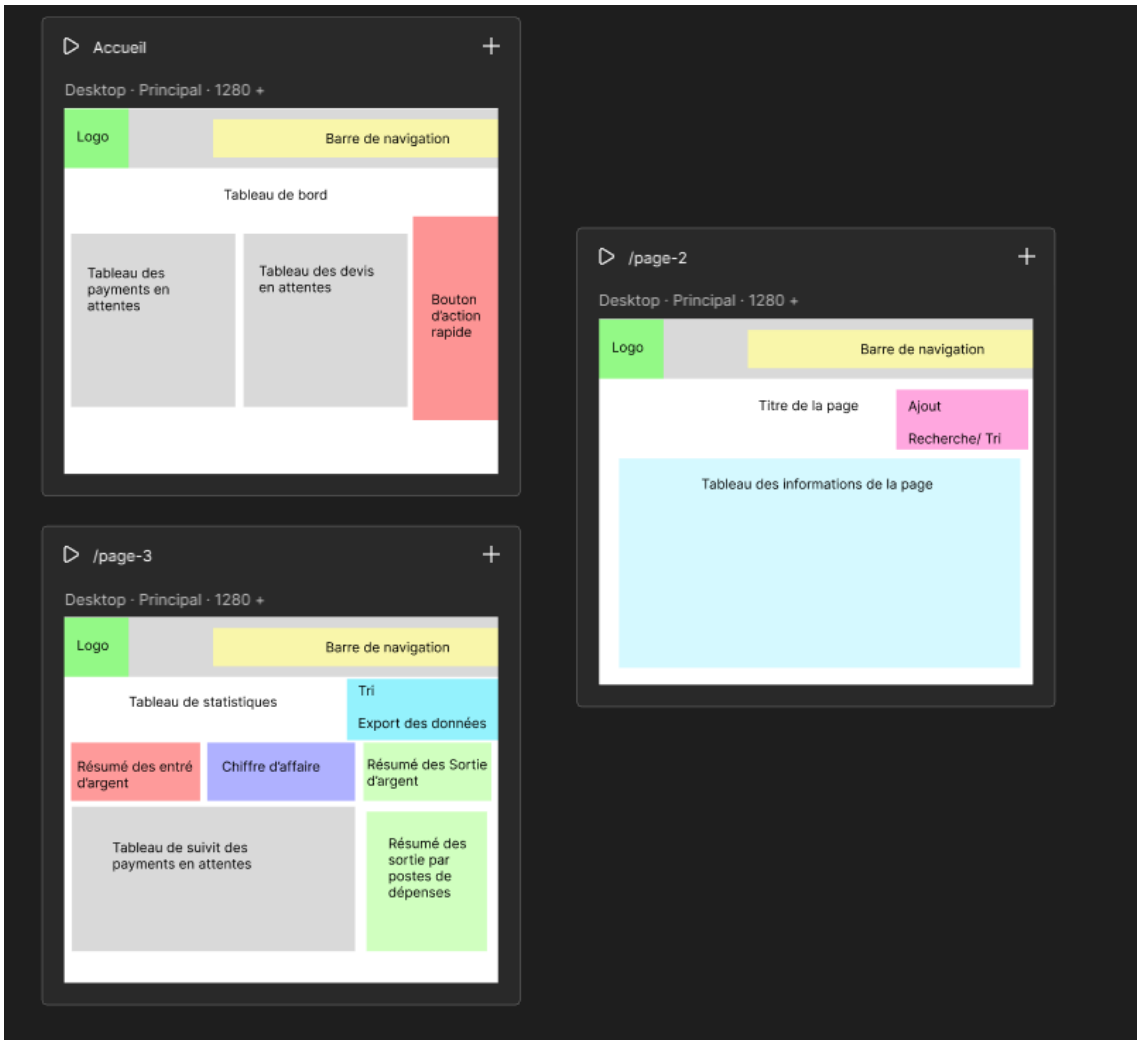
Dans le cadre de la conception de l'application, j'ai réalisé le maquettage de l'interface utilisateur afin de définir la structure générale du projet avant le développement.
Cette étape m'a permis d'anticiper l'organisation des différentes pages et de réfléchir à l'expérience utilisateur, en tenant compte du profil d'un utilisateur peu à l'aise avec les outils informatiques.
L'objectif principal était de concevoir une interface simple, cohérente et intuitive, permettant une prise en main rapide de l'application.

Analyse des besoins et structuration de l'interface
Avant la réalisation des maquettes, j'ai défini une structure commune pour l'ensemble des pages afin d'assurer une cohérence globale de l'application.
Cette structure repose sur trois éléments principaux :
Un header regroupant la navigation
Une zone d'actions permettant les interactions principales (ajout, recherche, tri)
Une zone centrale dédiée à l'affichage des données sous forme de tableau
Cette organisation a été pensée pour limiter la charge cognitive de l'utilisateur et lui permettre de retrouver rapidement les fonctionnalités essentielles.

Pages spécifiques de l'application
Certaines pages ont été conçues avec une organisation différente afin de répondre à des besoins fonctionnels particuliers.
Le dashboard a été pensé comme une page de synthèse regroupant les informations principales ainsi que des actions rapides, notamment le suivi des devis et des paiements en cours.
La page de statistiques, quant à elle, permet une analyse globale de l'activité en centralisant les données financières telles que les factures, les paiements et les dépenses.

Réalisation des maquettes
Les maquettes ont été réalisées à l'aide de l'outil Figma, ce qui m'a permis de visualiser concrètement l'organisation des différentes pages avant leur développement.
Cette étape a été essentielle pour valider les choix d'ergonomie et s'assurer que la structure répondait bien aux besoins définis en amont.

Le maquettage a servi de base au développement front-end et a permis de garantir une cohérence entre la conception et l'implémentation finale.



2. Précisez les moyens utilisés :

Le maquettage a été réalisé avec Figma, en complément d'une réflexion sur l'expérience utilisateur et la structure des pages.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Uniformisation des pages (header, zone d'actions, tableau) et conception de pages spécifiques (dashboard, statistiques).

Activité-type 1 — Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°3 ▸ Réaliser des interfaces utilisateur statique web ou web mobile

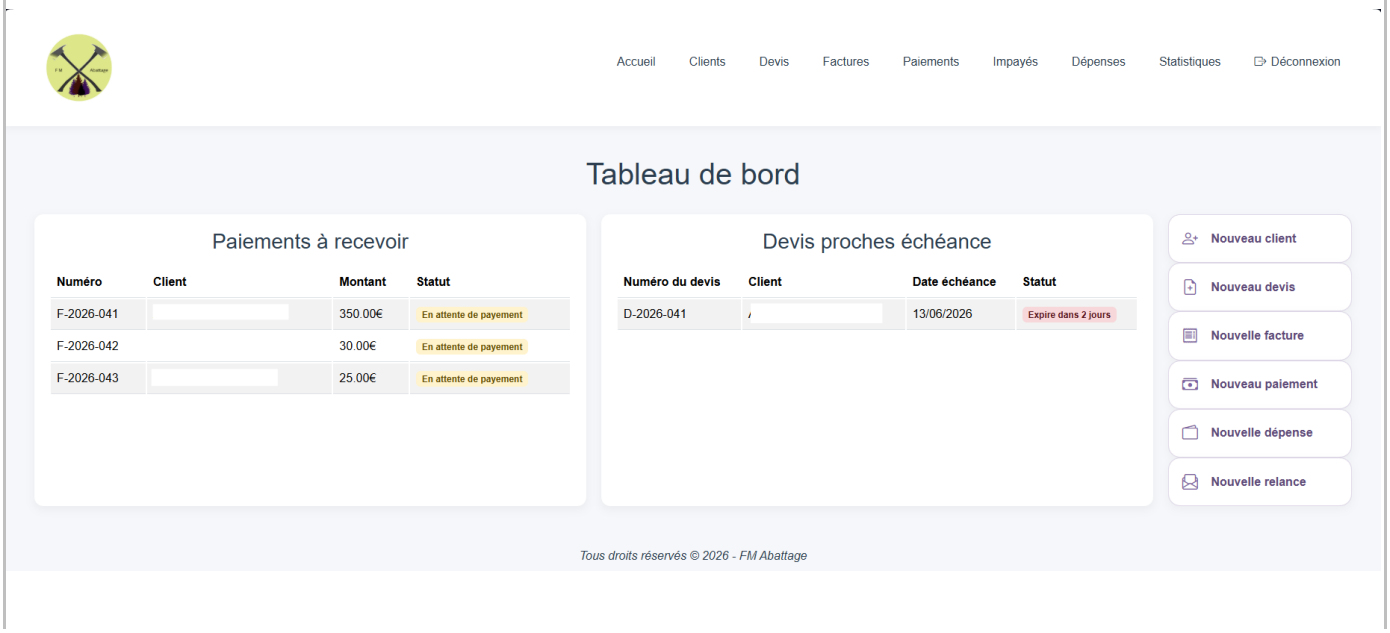
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

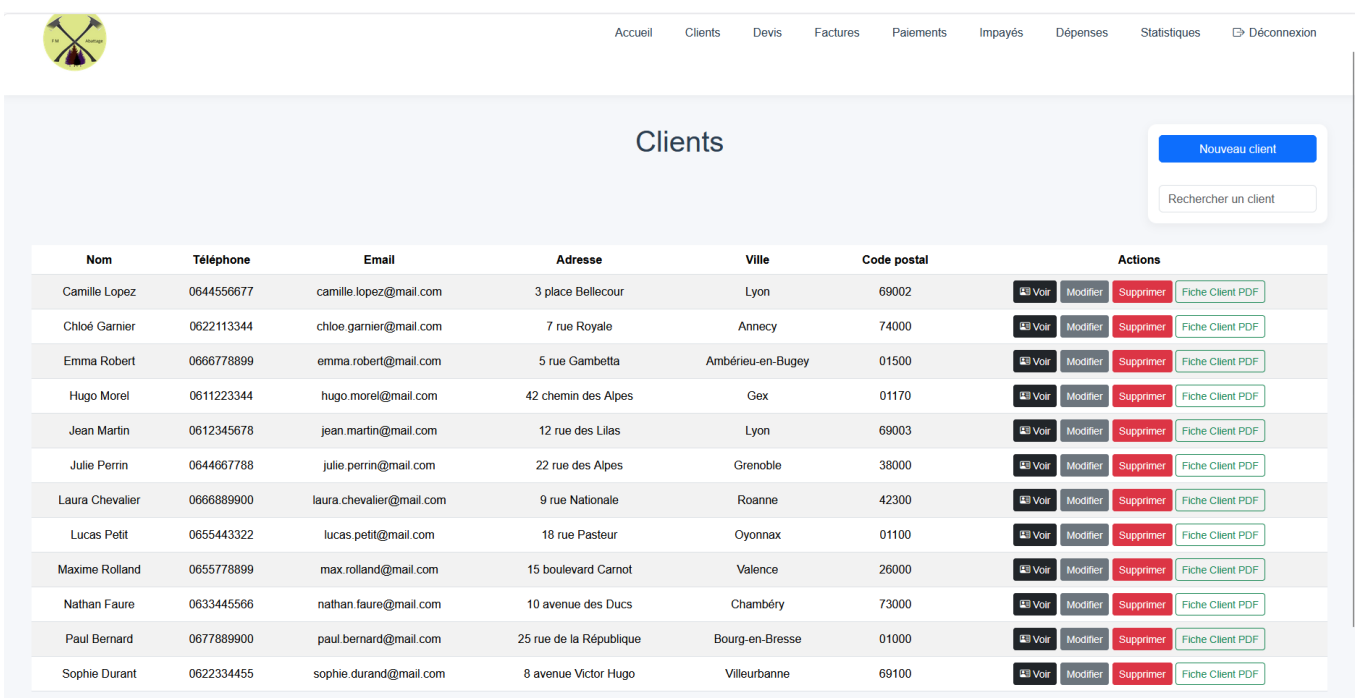
J'ai développé les interfaces utilisateur de l'application en HTML et CSS à partir des maquettes réalisées en amont.
L'objectif était de transformer les maquettes en pages fonctionnelles tout en respectant la structure définie lors de la phase de conception.

Structure des pages
Chaque page de l'application suit une structure commune afin d'assurer une cohérence globale :

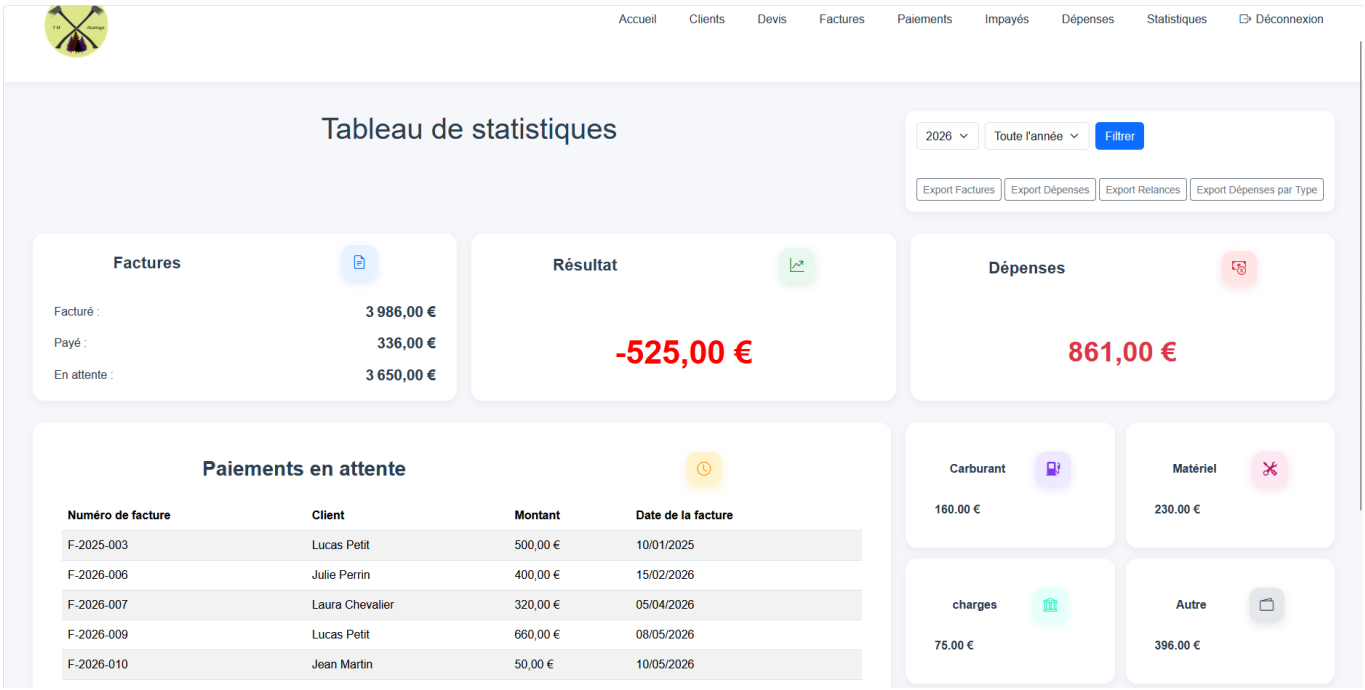
- Un header de navigation
- Une zone d'actions (boutons, recherche, tri)
- Un tableau d'affichage des données

Cette organisation permet une navigation fluide et une prise en main rapide de l'application.





Nom	Téléphone	Email	Adresse	Ville	Code postal	Actions
Camille Lopez	0644556677	camille.lopez@mail.com	3 place Bellecour	Lyon	69002	Voir Modifier Supprimer Fiche Client PDF
Chloé Garnier	0622113344	chloe.garnier@mail.com	7 rue Royale	Annecy	74000	Voir Modifier Supprimer Fiche Client PDF
Emma Robert	0666778899	emma.robert@mail.com	5 rue Gambetta	Ambérieu-en-Bugey	01500	Voir Modifier Supprimer Fiche Client PDF
Hugo Morel	0611223344	hugo.morel@mail.com	42 chemin des Alpes	Gex	01170	Voir Modifier Supprimer Fiche Client PDF
Jean Martin	0612345678	jean.martin@mail.com	12 rue des Lilas	Lyon	69003	Voir Modifier Supprimer Fiche Client PDF
Julie Perrin	0644667788	julie.perrin@mail.com	22 rue des Alpes	Grenoble	38000	Voir Modifier Supprimer Fiche Client PDF
Laura Chevalier	0666889900	laura.chevalier@mail.com	9 rue Nationale	Roanne	42300	Voir Modifier Supprimer Fiche Client PDF
Lucas Petit	0655443322	lucas.petit@mail.com	18 rue Pasteur	Oyonnax	01100	Voir Modifier Supprimer Fiche Client PDF
Maxime Rolland	0655778899	max.rolland@mail.com	15 boulevard Carnot	Valence	26000	Voir Modifier Supprimer Fiche Client PDF
Nathan Faure	0633445566	nathan.faure@mail.com	10 avenue des Ducs	Chambéry	73000	Voir Modifier Supprimer Fiche Client PDF
Paul Bernard	0677889900	paul.bernard@mail.com	25 rue de la République	Bourg-en-Bresse	01000	Voir Modifier Supprimer Fiche Client PDF
Sophie Durant	0622334455	sophie.durand@mail.com	8 avenue Victor Hugo	Villeurbanne	69100	Voir Modifier Supprimer Fiche Client PDF



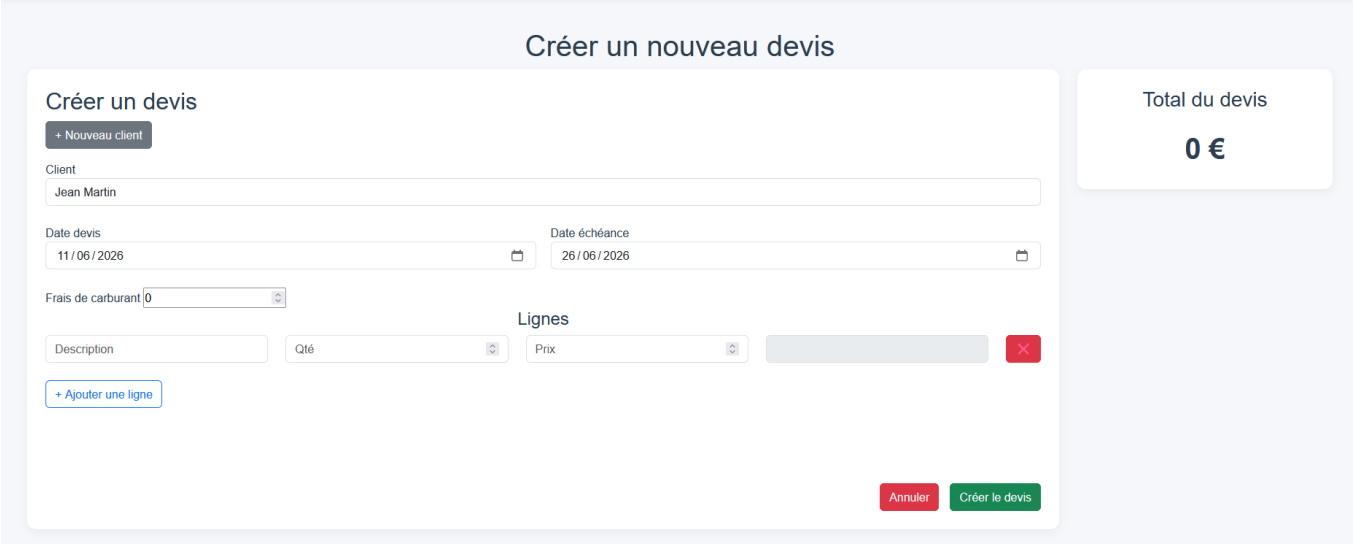
Numéro de facture	Client	Montant	Date de la facture
F-2025-003	Lucas Petit	500,00 €	10/01/2025
F-2026-006	Julie Perrin	400,00 €	15/02/2026
F-2026-007	Laura Chevalier	320,00 €	05/04/2026
F-2026-009	Lucas Petit	660,00 €	08/05/2026
F-2026-010	Jean Martin	50,00 €	10/05/2026

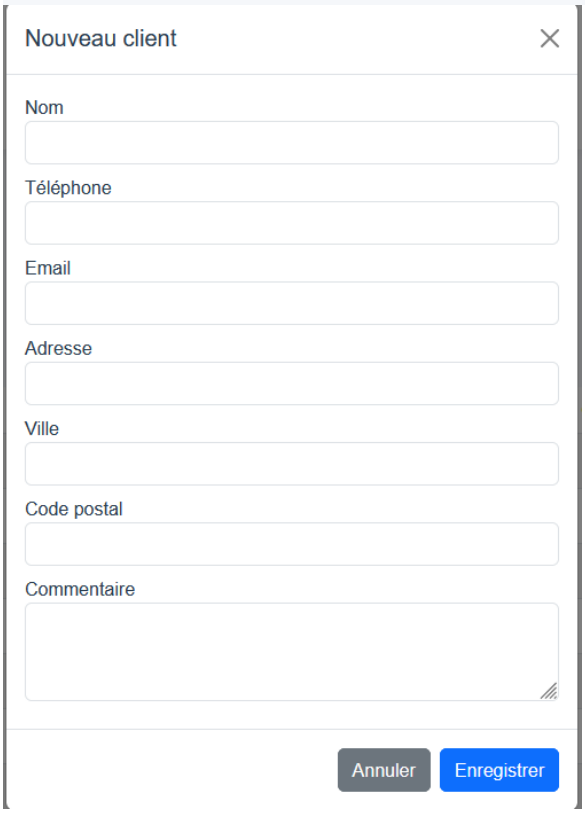
Formulaires et interaction utilisateur

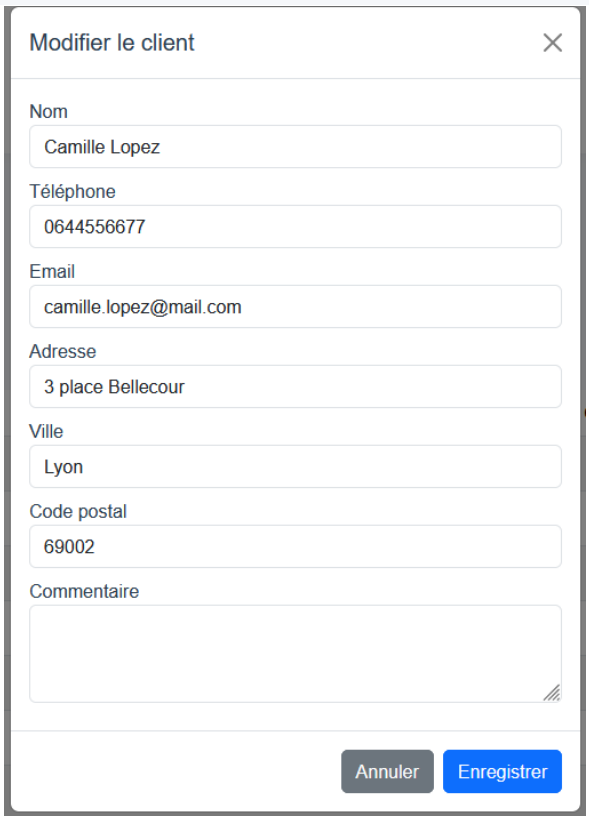
Des formulaires ont été développés afin de permettre la création et la modification des données :

- Clients
- Devis
- Factures
- Paiements
- Dépenses

Les champs ont été organisés de manière à limiter les erreurs de saisie et à améliorer la lisibilité.







Mise en page et ergonomie

Une attention particulière a été portée à la lisibilité et à la hiérarchisation des informations :

- Utilisation de tableaux structurés
- Séparation claire des zones de contenu
- Boutons d'action visibles et cohérents

L'objectif était de rendre l'application simple et intuitive à utiliser.

Tableau des devis

Créer un devis

2026

Numéro du devis	Client	Téléphone	Date échéance	Montant	Statut	Actions
D-2026-013	Emma Robert	0666778899	04/06/2026	120,00 €	En attente	PDF Modifier Supprimer
D-2026-011	Maxime Rolland	0655778899	30/05/2026	400,00 €	En attente	PDF Modifier Supprimer
D-2026-010	Hugo Morel	0611223344	21/05/2026	76,00 €	En attente	PDF Modifier Supprimer
D-2026-018	Jean Martin	0612345678	24/05/2026	100,00 €	Accepté	PDF Facturer
D-2026-008	Jean Martin	0612345678	21/05/2026	170,00 €	Accepté	PDF Facturer
D-2026-017	Jean Martin	0612345678	16/05/2026	190,00 €	Accepté	PDF Facturer
D-2026-004	Laura Chevalier	0666889900	15/04/2026	320,00 €	Archivé	PDF
D-2026-002	Julie Perrin	0644667788	25/02/2026	400,00 €	Archivé	PDF
D-2026-016	Chloé Garnier	0622113344	23/05/2026	862,00 €	Refusé	PDF Supprimer
D-2026-015	Jean Martin	0612345678	29/05/2026	52,00 €	Refusé	PDF Supprimer
D-2026-009	Lucas Petit	0655443322	21/05/2026	650,00 €	Refusé	PDF Supprimer
D-2026-014	Jean Martin	0612345678	18/05/2026	161,00 €	Refusé	PDF Supprimer
D-2026-007	Chloé Garnier	0622113344	10/05/2026	170,00 €	Refusé	PDF Supprimer

Documents imprimables

Les documents générés (devis, factures, relances) ont été conçus avec une mise en page adaptée à l'impression.

Cela permet leur utilisation dans un contexte administratif réel.

Lucas Petit

← Retour

Téléphone : 0655443322
Email : lucas.petit@mail.com
Adresse : 18 rue Pasteur
Ville : Oyonnax
Code postal : 01100

Commentaire

-

Devis

Numéro	Date	Montant	Statut	PDF
D-2026-009	07/05/2026	650.00 €	refuse	PDF
D-2025-001	05/01/2025	500.00 €	accepte	PDF

Factures

Numéro	Date	Montant	Statut	PDF
F-2025-003	10/01/2025	500.00 €	partielle	PDF
F-2026-009	08/05/2026	660.00 €	en_attente	PDF
F-2026-011	10/05/2026	650.00 €	en_attente	PDF
F-2026-014	18/05/2026	610.00 €	en_attente	PDF
F-2026-017	23/05/2026	630.00 €	en_attente	PDF

Paiements

Facture	Date	Montant	Mode de paiement
F-2025-003	20/01/2025	200.00 €	Virement compte Sumup
F-2025-003	01/02/2025	100.00 €	Espèces
F-2025-003	16/05/2026	100.00 €	Carte Bancaire

Relances

Facture	Niveau de relance	Date de relance	PDF
F-2026-011	Première relance	17/05/2026	PDF
F-2026-009	Première relance	17/05/2026	PDF

Lucas Petit

Fiche client

Informations

Téléphone
0655443322

Email
lucas.petit@mail.com

Adresse
18 rue Pasteur 01100 Oyonnax

Commentaires

Aucun commentaire pour le moment.

Devis

Numéro	Date	Montant	Statut
D-2025-001		500,00 €	Acceptée
D-2026-009		650,00 €	Refusée

Factures

Numéro	Date	Montant	Statut
F-2025-003	10/01/2025	500,00 €	partielle
F-2026-009	08/05/2026	660,00 €	En attente
F-2026-011	10/05/2026	650,00 €	En attente
F-2026-014	18/05/2026	610,00 €	En attente
F-2026-017	23/05/2026	630,00 €	En attente

Paiements

Facture	Date	Montant	Mode
F-2025-003	20/01/2025	200,00 €	Virement compte Sumup
F-2025-003	01/02/2025	100,00 €	Espèces
F-2025-003	16/05/2026	100,00 €	Carte Bancaire
F-2025-003	27/05/2026	10,00 €	Virement compte Nickel

Relances

Facture	Niveau	Date	Pénalité
F-2025-003	Première relance	23/05/2026	0,00 €
F-2026-009	Première relance	17/05/2026	0,00 €
F-2026-011	Première relance	17/05/2026	0,00 €
F-2026-014	Première relance	23/05/2026	0,00 €

Accueil Clients Devis Factures Paiements Impayés Dépenses Statistiques Déconnexion

Créer un nouveau devis

Créer un devis

+ Nouveau client

Client
Julie Perrin

Date devis
11 / 06 / 2026

Date échéance
26 / 06 / 2026

Frais de carburant 20

Lignes

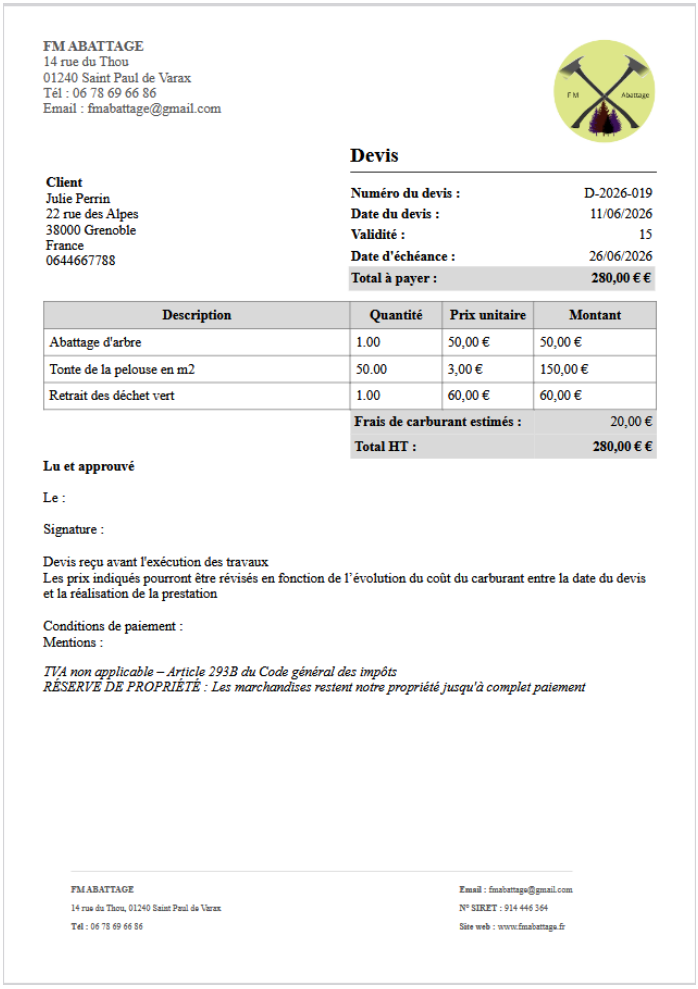
Abattage d'arbre	1	50	50.00 €
Tonte de la pelouse en m2	50	3	150.00 €
Retrait des déchet vert	1	60	60.00 €

+ Ajouter une ligne

Annuler Créer le devis

Total du devis

280.00 €



FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Devis

Client
Julie Perrin
22 rue des Alpes
38000 Grenoble
France
0644667788

Numéro du devis :	D-2026-019
Date du devis :	11/06/2026
Validité :	15
Date d'échéance :	26/06/2026
Total à payer :	**280,00 € €**

Description	Quantité	Prix unitaire	Montant
Abattage d'arbre	1.00	50,00 €	50,00 €
Tonte de la pelouse en m2	50.00	3,00 €	150,00 €
Retrait des déchet vert	1.00	60,00 €	60,00 €
	Frais de carburant estimés :		20,00 €
	Total HT :		**280,00 € €**

Lu et approuvé

Le :

Signature :

Devis reçu avant l'exécution des travaux
Les prix indiqués pourront être révisés en fonction de l'évolution du coût du carburant entre la date du devis et la réalisation de la prestation

Conditions de paiement :
Mentions :

TVA non applicable – Article 293B du Code général des impôts
RÉSERVE DE PROPRIÉTÉ : Les marchandises restent notre propriété jusqu'à complet paiement

FM ABATTAGE
14 rue du Thou, 01240 Saint Paul de Varax
Tél : 06 78 69 66 86

Email : fmabattage@gmail.com
N° SIRET : 914 446 364
Site web : www.fmabattage.fr

2. Précisez les moyens utilisés :

- HTML
- CSS
- Structure de projet web,
- Outils de test navigateur

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Les interfaces ont été conçues pour être cohérentes avec les maquettes réalisées sous Figma, afin de garantir une continuité entre conception et développement.

Activité-type 1 Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°4 ▸ Développer la partie dynamique des interfaces utilisateur web ou web mobile

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai développé la partie dynamique de l'application afin de rendre les interfaces interactives et connectées aux données de la base MySQL.
L'objectif était de permettre une mise à jour automatique des informations affichées, sans rechargement manuel de la page, ainsi qu'une meilleure expérience utilisateur dans la gestion quotidienne des données (clients, devis, factures, paiements et dépenses).

Fonctionnement de la dynamique de l'application
L'application repose sur une communication entre le front-end et le back-end permettant de récupérer et manipuler les données.
Les informations sont envoyées et récupérées via des requêtes vers le serveur, puis affichées dynamiquement dans les tableaux de gestion.
Les données sont mises à jour automatiquement lors des actions suivantes :

- Ajout d'un élément
- Modification d'un élément
- Suppression d'un élément

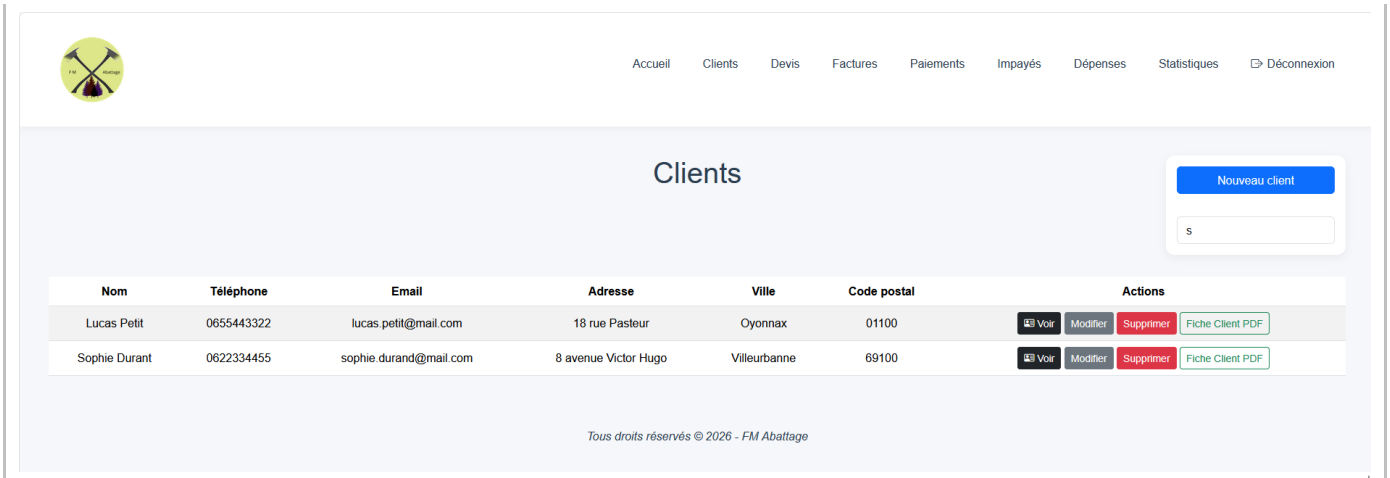
Cela permet de garantir une interface toujours à jour.

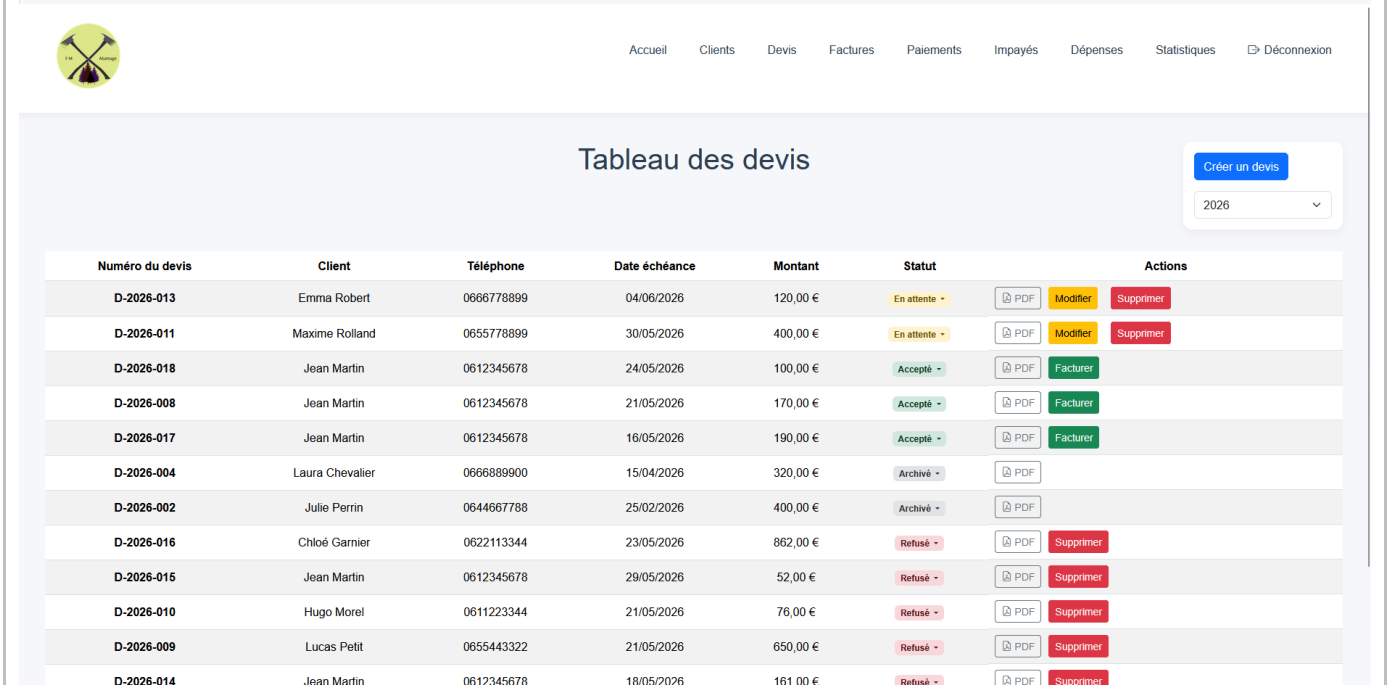
Fonctionnalités développées
J'ai implémenté un système de recherche permettant de filtrer les données affichées dans les tableaux.
Ce système améliore la rapidité d'accès aux informations et facilite la gestion des données volumineuses.
Un système de tri a été mis en place afin d'organiser les informations selon différents critères (date, montant, statut…).
Cela permet une meilleure lisibilité et une analyse plus rapide des données.

Certaines informations sont calculées dynamiquement par l'application, notamment :

- Le montant restant à payer sur une facture
- Le statut d'une facture (payée / partielle / impayée)
- Les données utilisées pour les statistiques financières

Ces calculs évitent les erreurs manuelles et garantissent la cohérence des données affichées.


Accueil
Clients
Devis
Factures
Paiements
Impayés
Dépenses
Statistiques
Déconnexion
Clients
Nouveau client
s
Nom
Téléphone
Email
Adresse
Ville
Code postal
Actions
Lucas Petit
0655443322
lucas.petit@mail.com
18 rue Pasteur
Oyonnax
01100
Voir
Modifier
Supprimer
Fiche Client PDF
Sophie Durant
0622334455
sophie.durand@mail.com
8 avenue Victor Hugo
Villeurbanne
69100
Voir
Modifier
Supprimer
Fiche Client PDF
Tous droits réservés © 2026 - FM Abattage


Accueil
Clients
Devis
Factures
Paiements
Impayés
Dépenses
Statistiques
Déconnexion
Tableau des devis
Créer un devis
2026
Numéro du devis
Client
Téléphone
Date échéance
Montant
Statut
Actions
D-2026-013 Emma Robert 0666778899 04/06/2026 120,00 € En attente PDF Modifier Supprimer
D-2026-011 Maxime Rolland 0655778899 30/05/2026 400,00 € En attente PDF Modifier Supprimer
D-2026-018 Jean Martin 0612345678 24/05/2026 100,00 € Accepté PDF Facturer
D-2026-008 Jean Martin 0612345678 21/05/2026 170,00 € Accepté PDF Facturer
D-2026-017 Jean Martin 0612345678 16/05/2026 190,00 € Accepté PDF Facturer
D-2026-004 Laura Chevalier 0666889900 15/04/2026 320,00 € Archivé PDF
D-2026-002 Julie Perrin 0644667788 25/02/2026 400,00 € Archivé PDF
D-2026-016 Chloé Garnier 0622113344 23/05/2026 862,00 € Refusé PDF Supprimer
D-2026-015 Jean Martin 0612345678 29/05/2026 52,00 € Refusé PDF Supprimer
D-2026-010 Hugo Morel 0611223344 21/05/2026 76,00 € Refusé PDF Supprimer
D-2026-009 Lucas Petit 0655443322 21/05/2026 650,00 € Refusé PDF Supprimer
D-2026-014 Jean Martin 0612345678 18/05/2026 161,00 € Refusé PDF Supprimer

Numéro de la facture	Devis	Client	Date	Montant	Payé	Reste à payer	Statut	Actions
F-2026-006	D-2026-002	Julie Perrin	15/02/2026	400.00€	150.00 €	250.00 €	Paiement partiel	PDF Supprimer
F-2026-007	D-2026-004	Laura Chevalier	05/04/2026	320.00€	0.00 €	320.00 €	En attente de payement	PDF Supprimer
F-2026-008	D-2026-010	Hugo Morel	08/05/2026	76.00€	76.00 €	0.00 €	Facture payée	PDF Supprimer
F-2026-009	D-2026-009	Lucas Petit	08/05/2026	660.00€	0.00 €	660.00 €	En attente de payement	PDF Supprimer

J'ai développé un système de génération automatique de documents PDF (devis, factures et relances). Les données utilisées proviennent directement de la base de données et sont injectées dans les documents afin de garantir leur exactitude.

Exemple de scénario utilisateur
Lorsqu'un utilisateur crée une facture :

1. Les données sont envoyées au serveur
2. La base de données enregistre la nouvelle facture
3. L'interface est automatiquement mise à jour
4. Le statut de la facture est recalculé
5. Le montant restant est ajusté dynamiquement

Ce fonctionnement permet d'assurer une cohérence entre les données stockées et les données affichées.

Séparation des responsabilités
Une séparation a été mise en place entre :

- La logique serveur (traitement des données)
- L'affichage côté client (interface utilisateur)
- La base de données (stockage)

Cette architecture permet une meilleure maintenabilité et une organisation claire du code.

2. Précisez les moyens utilisés :

- JavaScript (logique côté client)

- Communication avec le backend Java
- Base de données MySQL
- Manipulation du DOM pour l'affichage dynamique
- API interne de l'application

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Les interfaces sont conçues pour fonctionner en temps réel, ce qui permet à l'utilisateur de voir immédiatement les modifications effectuées sans action supplémentaire.
Les fonctionnalités dynamiques ont été pensées pour limiter les erreurs humaines et automatiser les tâches répétitives.

Activité-type 2 — Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 1 ▸ Mettre en place une base de données relationnelle

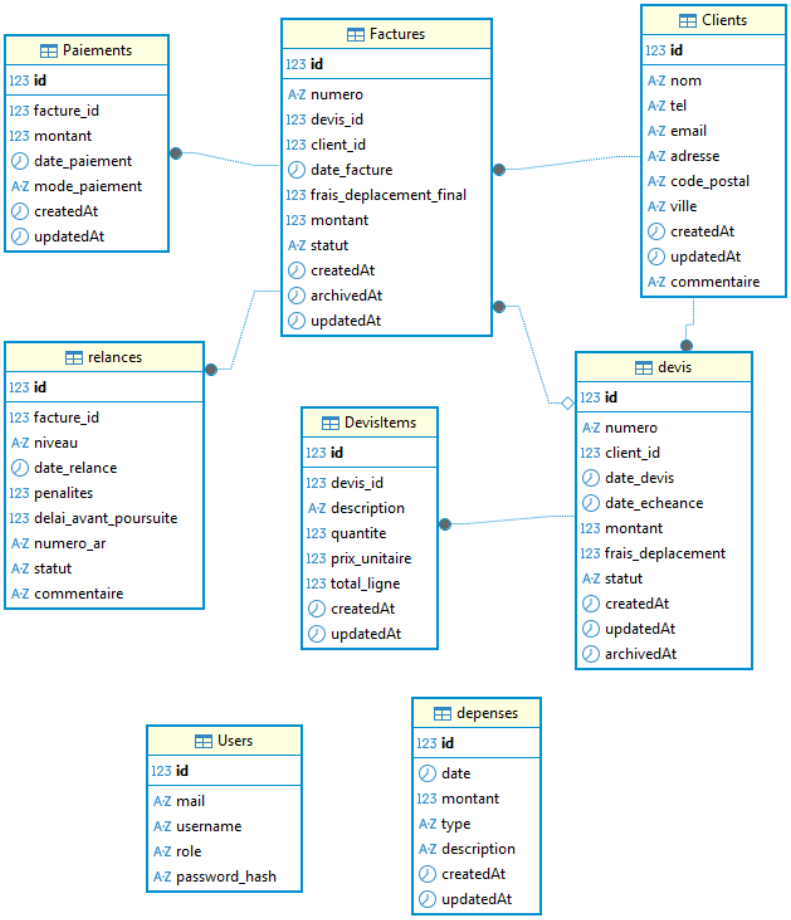
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai conçu et modélisé une base de données relationnelle MySQL permettant de gérer l'ensemble des données de l'application (clients, devis, factures, paiements, relances, dépenses et utilisateurs). L'objectif était de structurer les données afin d'assurer leur cohérence, leur intégrité et leur exploitation dans l'application.

Conception de la base de données

Avant l'implémentation, j'ai réfléchi à la structure globale des données en identifiant les entités principales du système :

- Clients
- Devis
- Factures
- Paiements
- Relances
- Dépenses
- Utilisateurs



Chaque entité a été traduite en table relationnelle.

Les relations ont été définies afin de respecter la logique métier :

- Un client peut avoir plusieurs devis
- Un devis peut générer une facture
- Une facture peut avoir plusieurs paiements
- Une facture peut générer plusieurs relances

Choix techniques

J'ai choisi MySQL car il s'agit d'un système de gestion de base de données relationnelle fiable et adapté aux applications structurées.

Ce choix permet :

- Une bonne gestion des relations entre tables
- L'utilisation de requêtes SQL complexes
- Une compatibilité avec Java côté serveur
- Une facilité d'hébergement sur Aiven

Implémentation

La base de données a été créée avec des clés primaires et étrangères afin de garantir l'intégrité des données.

Chaque table contient des champs adaptés à son rôle :

- Identifiants uniques
- Données métiers (nom, montant, statut, dates…)
- Clés de liaison entre entités

```sql
-- ========================
-- Table Factures
-- ========================

CREATE TABLE factures (
    id INT AUTO_INCREMENT PRIMARY KEY,
    numero VARCHAR(50) UNIQUE NOT NULL,
    devis_id INT NOT NULL,
    client_id INT NOT NULL,
    date_facture DATE NOT NULL,
    frais_deplacement_final DECIMAL(10,2) DEFAULT 0,
    montant DECIMAL(10, 2) NOT NULL,
    statut ENUM(
        'en_attente',
        'payee',
        'partielle'
    ) DEFAULT 'en_attente',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (devis_id) REFERENCES devis(id) ON DELETE CASCADE,
    FOREIGN KEY (client_id) REFERENCES clients(id) ON DELETE CASCADE
);

-- ========================
-- Table Paiements
-- ========================

CREATE TABLE paiements (
    id INT AUTO_INCREMENT PRIMARY KEY,
    facture_id INT NOT NULL,
    montant DECIMAL(10, 2) NOT NULL,
    date_paiement DATE NOT NULL,
    mode_paiement ENUM(
        'especes',
        'cheque',
        'virement_A',
        'virement_B',
        'cb',
        'autre'
    ) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (facture_id) REFERENCES factures(id) ON DELETE CASCADE
);
```

```javascript
const exportFactures = async () => {

  const sql = `
    SELECT
      f.date_facture AS 'Date de facturation',

      f.numero AS 'Numéro de facture',

      d.numero AS 'Numéro de devis',

      CONCAT(f.montant, '€')
        AS 'Montant de la facture',

      CONCAT(f.frais_deplacement_final, '€')
        AS 'Frais de déplacements',

      c.nom AS 'Nom du client',

      p.date_paiement AS 'Date de paiement',

      CONCAT(p.montant, '€')
        AS 'Montant payé',

      p.mode_paiement AS 'Mode de paiement'

    FROM Factures f

    INNER JOIN devis d
      ON d.id = f.devis_id

    INNER JOIN Clients c
      ON c.id = d.client_id

    INNER JOIN Paiements p
      ON p.facture_id = f.id
  `;
```

Exemples de fonctionnement
Lorsqu'un devis est accepté :

- Une facture peut être générée
- Les données du devis sont transférées vers la facture

Lorsqu'un paiement est ajouté :

- Le système vérifie le montant restant dû
- Le statut de la facture est mis à jour automatiquement

Sécurisation et cohérence
Des contraintes ont été mises en place pour éviter les incohérences :

- Impossibilité de créer une facture sans devis associé
- Interdiction d'ajouter un paiement supérieur au montant total
- Contrôle des statuts (devis / facture)

```
exports.createFactureFromDevis = async (req, res) => {

  const { frais_deplacement_final = 0 } = req.body;

  const devis = await Devis.findByPk(req.params.id, {
    include: [{ model: DevisItem, as: "items" }]
  });

  if (!devis) {
    return res.status(404).json({ error: "Devis non trouvé" });
  }

  try {

    if (!devis) {
      return res.status(404).json({ error: "Devis non trouvé" });
    }

    if (devis.statut !== "accepte") {
      return res.status(400).json({ error: "Le devis doit être accepté" });
    }
```

2. Précisez les moyens utilisés :

- MySQL
- Modélisation relationnelle
- Requêtes SQL

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Gestion des relations clients/devis/factures/paiements/relances et intégration d'une table utilisateurs pour la sécurité.

Activité-type 2 — Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 2 ▸ *Développer des composants d'accès aux données SQL et NoSQL*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai développé les composants côté serveur permettant la communication entre l'application et la base de données MySQL.
Ces composants ont pour rôle de gérer les échanges de données entre l'interface utilisateur et la base de données, en assurant les opérations de création, lecture, modification et suppression des informations.
Chaque entité principale de l'application dispose de ses propres traitements serveur afin de garantir une organisation claire et maintenable.

Mise en place des opérations CRUD
J'ai créé des contrôleurs dédiés pour chaque entité de l'application :

- Clients
- Devis
- Factures
- Paiements
- Relances
- Dépenses

Ces contrôleurs permettent la gestion complète des données via les opérations CRUD :

- **Create** : ajout de nouvelles données
- **Read** : récupération des données
- **Update** : modification des données existantes
- **Delete** : suppression des données

```
exports.getAllClients = async (req, res) => {
  try {
    const clients = await Client.findAll();
    res.json(clients);
  }
  catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.getClientById = async (req, res) => {
  try {
    const clients = await Client.findByPk(req.params.id);
    if (clients) {
      res.json(clients);
    } else {
      res.status(404).json({ error: "Client non trouvé" });
    }
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.createClient = async (req, res) => {
  try {
    const client = await Client.create(req.body);
    res.status(201).json(client);
  }
  catch (error) {
    res.status(500).json({
      error: error.message
    });
  }
};
```

```
exports.updateClient = async (req, res) => {
  try {
    const client = await Client.findByPk(req.params.id);
    if (!client) {
      return res.status(404).json({
        error: "Client non trouvé"
      });
    }
    await client.update(req.body);
    res.json(client);
  }
  catch (error) {
    res.status(500).json({
      error: error.message
    });
  }
};

exports.deleteClient = async (req, res) => {
  try {
    const deleted = await Client.destroy({
      where: {
        id: req.params.id
      }
    });
    if (!deleted) {
      return res.status(404).json({
        error: "Client non trouvé"
      });
    }
    res.json({
      message: "Client supprimé"
    });
  }
  catch (error) {
    res.status(500).json({
      error: error.message
    });
  }
};
```

Communication avec la base de données

Les contrôleurs exécutent des requêtes SQL permettant d'interagir directement avec la base de données MySQL.

Ces requêtes sont envoyées depuis le serveur et permettent :

- D'enregistrer les données saisies par l'utilisateur
- De récupérer les informations pour affichage
- De mettre à jour les enregistrements
- De supprimer les données obsolètes

La base de données joue ici le rôle de stockage centralisé des informations de l'application.

Séparation des responsabilités
Une architecture structurée a été mise en place afin de séparer les différentes couches de l'application :
- La couche serveur (traitement des données et logique métier)
- La couche d'accès aux données (requêtes SQL)
- La couche interface utilisateur (affichage des données)

Cette séparation permet d'améliorer :
- La lisibilité du code
- La maintenabilité de l'application
- L'évolution future du projet

Exemple de fonctionnement
Lorsqu'un utilisateur crée un client :
1. Les données sont envoyées au serveur
2. Le contrôleur "clients" traite la demande
3. Une requête SQL INSERT est exécutée
4. La base de données enregistre le client
5. Les données sont renvoyées à l'interface

Lorsqu'un utilisateur modifie une facture :
1. Les nouvelles informations sont envoyées au serveur
2. Le contrôleur met à jour les données
3. Une requête SQL UPDATE est exécutée
4. L'interface est mise à jour automatiquement

Gestion des données et cohérence
Les contrôleurs permettent également de garantir la cohérence des données en vérifiant certaines conditions avant exécution des requêtes.
Par exemple :
- Impossibilité de modifier une facture inexistante
- Vérification de l'existence d'un client avant création d'un devis
- Contrôle des données avant suppression

Ces vérifications permettent d'éviter les erreurs et les incohérences dans la base de données.

```
exports.deleteDevis = async (req, res) => {
  try {
    const devis = await Devis.findByPk(req.params.id);
    if (!devis) {
      return res.status(404).json({ error: "Devis non trouvé" });
    }
    if (devis.statut === "accepte") {
      return res.status(400).json({ error: "Impossible de supprimer un devis accepté" });
    }
    await devis.destroy();
    res.json({ message: "Devis supprimé avec succès" });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.archiveDevis = async (req, res) => {
  try {
    const devis = await Devis.findByPk(req.params.id);
    if (devis.statut === "accepte") {
      return res.status(400).json({
        error: "Impossible de modifier un devis accepté"
      });
    }
    if (!devis) {
      return res.status(404).json({ error: "Devis non trouvé" });
    }
    if (devis.statut === "archive") {
      return res.status(400).json({ error: "Impossible de modifier un devis archivé" });
    }
    if (devis.statut === "accepte") {
      return res.status(400).json({ error: "Impossible d'archiver un devis accepté" });
    }
    devis.statut = "archive";
    devis.archivedAt = new Date();
    await devis.save();
    res.json({ message: "Devis archivé avec succès", devis });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};
```

Organisation du code

Le code a été organisé afin de séparer clairement les responsabilités :

- Un contrôleur par entité
- Des fonctions dédiées aux opérations CRUD
- Une logique serveur indépendante de l'interface utilisateur

Cette organisation facilite la maintenance et l'évolution de l'application.

2. Précisez les moyens utilisés :

- Java (côté serveur)

- SQL (requêtes de gestion des données)
- MySQL (base de données relationnelle)
- Architecture de type MVC (séparation logique des couches)

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Ce fonctionnement permet de centraliser les traitements des données côté serveur et d'assurer une cohérence globale entre les actions de l'utilisateur et les données stockées.

Activité-type 2 — Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 3 ▸ *Développer des composants métiers côté serveur*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai mis en place plusieurs règles métier afin de garantir la cohérence des données et le bon fonctionnement de l'application.
Ces règles permettent de contrôler les actions de l'utilisateur, d'éviter les incohérences dans la base de données et d'assurer un suivi logique des données liées à la gestion (clients, devis, factures et paiements).
J'ai également développé un système d'authentification sécurisé ainsi qu'un mécanisme de génération automatique de documents administratifs au format PDF.

Règles métier mises en place :

Gestion des clients

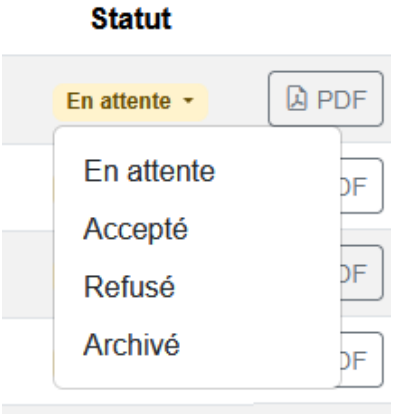
Lors de la création d'un client, certains champs sont obligatoires, notamment le nom et le numéro de téléphone.
Cela permet de garantir que les données enregistrées sont exploitables dans les autres modules de l'application.

Gestion des devis

Un devis ne peut pas être enregistré s'il ne contient pas de lignes complètes.
Chaque devis suit un cycle de vie avec différents statuts :

- En attente
- Accepté
- Refusé
- Archivé

Seuls les devis acceptés peuvent être transformés en facture.



Gestion des factures

Les factures sont générées uniquement à partir de devis validés.
Ce fonctionnement permet de garantir la cohérence entre les documents commerciaux et d'éviter la création de factures incohérentes ou isolées.

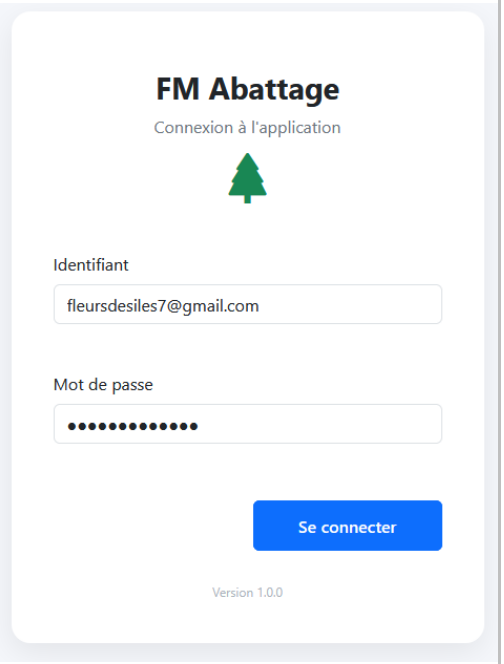
Gestion des paiements

Lors de l'ajout d'un paiement, plusieurs vérifications sont effectuées :

- La facture associée doit exister
- Le montant du paiement ne peut pas dépasser le montant total de la facture
- Le statut de la facture est mis à jour automatiquement selon le montant restant dû

Gestion des relances

La création de relances est limitée aux factures impayées ou partiellement réglées.
Le contenu de la relance s'adapte automatiquement au montant restant à payer, afin de refléter la situation réelle du client.



Sécurité de l'application

Un système d'authentification a été mis en place afin de sécuriser l'accès à l'application.
Lors de la création d'un compte utilisateur :

- Le mot de passe est haché avant stockage en base de données
- Aucune donnée sensible n'est enregistrée en clair

Lors de la connexion :

- Le mot de passe saisi est comparé au hash stocké
- L'accès est accordé uniquement si les informations sont valides

Ce système permet de limiter les accès non autorisés et de protéger les données de gestion.

```
exports.login = async (req, res) => {
    const { mail, password } = req.body;
    try {
        const user = await User.findOne({
            where: { mail }
        });

        if (!user) {
            return res.status(401).json({
                error: "Identifiants invalides"
            });
        }

        const validPassword =
            await bcrypt.compare(
                password,
                user.password_hash
            );

        if (!validPassword) {
            return res.status(401).json({
                error: "Identifiants invalides"
            });
        }
        req.session.userId = user.id;
        req.session.role = user.role;
        console.log("Session créée :", req.session);

        res.json({
            message: "Connexion réussie"
        });

    } catch (error) {
        console.error(error);
        res.status(500).json({
            error: error.message
        });
    }
};
```

Génération automatique de documents PDF

J'ai développé un système de génération automatique de documents au format PDF pour les devis, factures et relances.

Les données utilisées proviennent directement de la base de données :

- Informations client
- Lignes de devis
- Montants
- Statut des paiements

Ces données sont injectées automatiquement dans les documents afin de garantir leur exactitude.

Cette automatisation permet de réduire les erreurs de saisie manuelle et d'accélérer la production des documents administratifs.

```
22  exports.generateDevisPDF = async (req, res) => {
23    try {
24      const devis = await Devis.findByPk(req.params.id, {
25        include: [
26          { model: Client, as: "client" },
27          { model: DevisItem, as: "items" }
28        ]
29      });
30
31      if (!devis) {
32        return res.status(404).json({ error: "Devis non trouvé" });
33      }
34
35      let html = fs.readFileSync(
36        path.join(__dirname, "../templates/pages/devis.html"),
37        "utf8"
38      );
39
40      const css = fs.readFileSync(
41        path.join(__dirname, "../templates/asset/css/Devis.css"),
42        "utf8"
43      );
44
45      html = `<style>${css}</style>${html}`;
46
47      const logoPath = path.join(__dirname, "../templates/asset/img/logo.png");
48      const logoBase64 = fs.readFileSync(logoPath, { encoding: "base64" });
49
50
51      // Items
52      const itemsHTML = (devis.items || []).map(item => `
53        <tr>
54          <td>${item.description}</td>
55          <td>${item.quantite}</td>
56          <td>${formatPrice(item.prix_unitaire)}</td>
57          <td>${formatPrice(item.prix_unitaire * item.quantite)}</td>
58        </tr>
59      `).join("");
60
61      html = html
62        .replace("{{client_nom}}", devis.client?.nom || "")
63        .replace("{{client_tel}}", devis.client?.tel || "")
64        .replace("{{client_adresse}}", devis.client?.adresse || "")
65        .replace("{{client_code_postal}}", devis.client?.code_postal || "")
66        .replace("{{client_ville}}", devis.client?.ville || "")
67        .replace("{{client_email}}", devis.client?.email || "")
68        .replace("{{numero}}", devis.numero || "")
```

```
60
61        html = html
62          .replace("{{client_nom}}", devis.client?.nom || "")
63          .replace("{{client_tel}}", devis.client?.tel || "")
64          .replace("{{client_adresse}}", devis.client?.adresse || "")
65          .replace("{{client_code_postal}}", devis.client?.code_postal || "")
66          .replace("{{client_ville}}", devis.client?.ville || "")
67          .replace("{{client_email}}", devis.client?.email || "")
68          .replace("{{numero}}", devis.numero || "")
69          .replace("{{date_devis}}", formatDate(devis.date_devis))
70          .replace("{{date_echeance}}", formatDate(devis.date_echeance))
71          .replace("{{frais}}", formatPrice(devis.frais_deplacement))
72          .replaceAll("{{total}}", formatPrice(devis.montant))
73          .replace("{{delai}}", getDelai(devis.date_devis, devis.date_echeance))
74          .replace("{{items}}", itemsHTML)
75          .replace("{{logo}}", logoBase64);
76
77    const browser = await launchBrowser();
78    const page = await browser.newPage();
79
80    await page.setContent(html, { waitUntil: "networkidle0" });
81
82    filePath = path.join(__dirname, "../uploads", `devis_${devis.id}.pdf`);
83
84    await page.pdf({
85      path: filePath,
86      format: "A4",
87      printBackground: true,
88      margin: {
89            top: "8mm",
90            left: "10mm",
91            right: "10mm",
92            bottom: "8mm"
93          }
94    });
95
96    await browser.close();
97
98    return res.sendFile(filePath);
99
100     } catch (error) {
101       console.error(error);
102       return res.status(500).json({ error: error.message });
103     }
104   };
```

Exemples de fonctionnement

Lorsqu'un devis est accepté :

- Son statut est mis à jour
- Une facture est automatiquement générée

Lorsqu'un paiement est ajouté :

- Le montant restant dû est recalculé
- Le statut de la facture est modifié

Lorsqu'une relance est créée :

- Seules les factures non soldées sont proposées
- Le contenu de la relance est adapté automatiquement

```
// Calcul total payé par facture
const paiementsParFacture = {};

data.forEach(p => {
  if (!paiementsParFacture[p.facture_id]) {
    paiementsParFacture[p.facture_id] = 0;
  }
  paiementsParFacture[p.facture_id] += Number(p.montant);
});

const div = document.getElementById("paiementTable");

if (data.length === 0) {
  div.innerHTML = "<p>Aucun paiement pour le moment</p>";
  return;
}
```

```
exports.createPaiement = async (req, res) => {
  const { facture_id, montant, mode_paiement, date_paiement } = req.body;

  try {
    const facture = await Facture.findByPk(facture_id, {
      include: [{ model: Paiement, as: "paiements" }]
    });

    if (!facture) {
      return res.status(404).json({ error: "Facture non trouvée" });
    }

    //calcul déjà payé
    let totalPaye = 0;
    facture.paiements.forEach(p => {
      totalPaye += parseFloat(p.montant);
    });

    const reste = facture.montant - totalPaye;
    const montantNumber = parseFloat(montant);

    if (montantNumber > reste) {
      return res.status(400).json({
        error: "Montant supérieur au reste à payer"
      });
    }

    // création paiement
    const paiement = await Paiement.create({
      facture_id,
      montant: montantNumber,
      mode_paiement,
      date_paiement
    });

    // mise à jour statut facture
    const nouveauTotal = totalPaye + montantNumber;

    if (Math.abs(nouveauTotal - facture.montant) < 0.01) {
      facture.statut = "payee";
    } else {
      facture.statut = "partielle";
    }

    await facture.save();

    res.status(201).json({
      message: "Paiement ajouté",
```

Organisation et séparation des responsabilités

L'application est structurée selon une séparation claire des responsabilités :

- La logique métier est géré côté serveur
- Les règles de contrôle sont appliquées avant l'enregistrement en base
- L'interface utilisateur affiche les données traitées

Cette organisation permet de sécuriser les traitements et d'assurer la cohérence globale de l'application.

2. Précisez les moyens utilisés :

- Java (logique serveur)
- JavaScript (interaction côté client)
- MySQL (stockage des données)
- Système de hachage pour les mots de passe via Bcrypt
- Génération de fichiers PDF via Puppeteer
- Requêtes backend pour la communication entre les couches

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

L'ensemble des règles métier a été conçu afin d'automatiser un maximum de tâches administratives. Cela permet de limiter les erreurs humaines et de garantir une gestion fiable des données financières et commerciales.

Activité-type 2 — Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 4 ▸ *Documenter le déploiement d'une application dynamique web ou web mobile*

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai rédigé une documentation d'installation et d'utilisation de l'application afin de faciliter sa prise en main.
J'ai également procédé au déploiement de l'application afin de la rendre accessible en conditions réelles.

Documentation technique
La documentation précise :

- Les prérequis techniques
- La configuration de la base de données
- Les étapes d'installation et de lancement
- Les informations nécessaires au déploiement

Cette documentation permet de reproduire l'environnement de l'application.

README

FM Abattage - Application de ges

Application web de gestion développée pour l'entreprise FM Abattag

Fonctionnalités

Gestion des clients
- Création et modification des fiches clients
- Consultation de la liste des clients

Gestion des devis
- Création de devis
- Modification des devis
- Génération PDF
- Gestion des statuts :
  - En attente
  - Accepté
  - Refusé
  - Archivé

Gestion des factures
- Création de factures à partir des devis
- Génération PDF
- Suivi du paiement des factures

Gestion des paiements
- Enregistrement des paiements
- Gestion des paiements partiels

Filter headings
- FM Abattage - Application de gestion
  - Fonctionnalités
    - Gestion des clients
    - Gestion des devis
    - Gestion des factures
    - Gestion des paiements
    - Gestion des relances
    - Gestion des dépenses
    - Tableau de bord
  - Technologies utilisées
    - Front-end
    - Back-end
    - Base de données
    - Hébergement
  - Installation
    - Cloner le projet
    - Installer les dépendances
    - Configurer les variables d'environnement
    - Lancer le projet
  - Auteur
  - État du projet

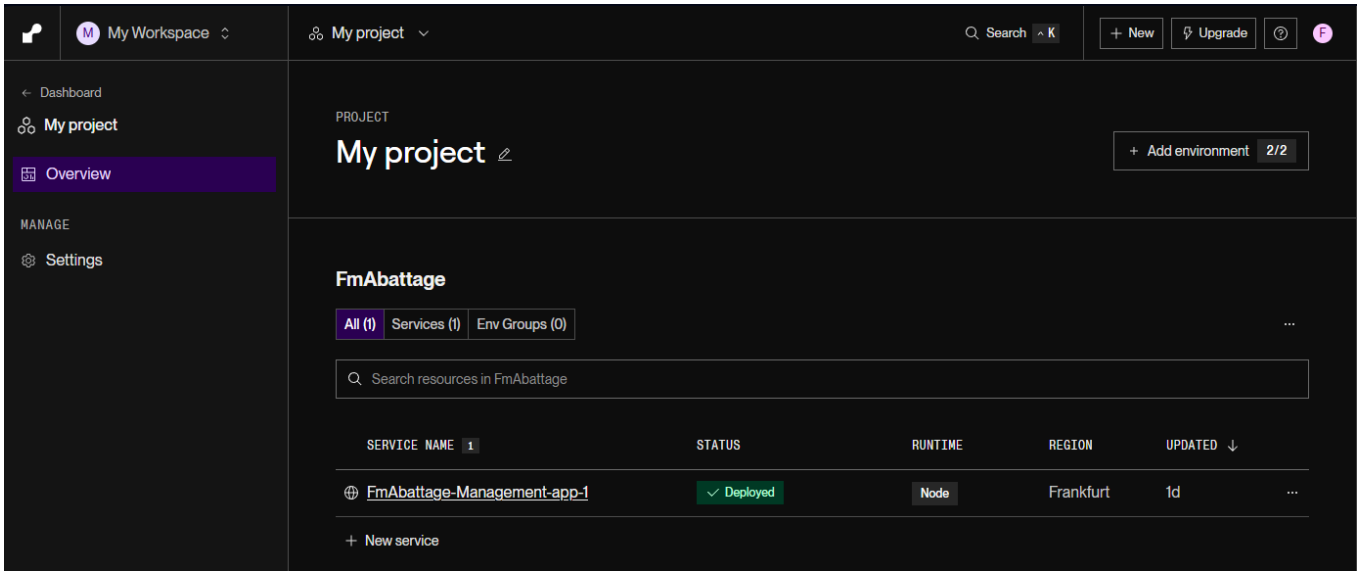
Déploiement de l'application
L'application a été déployée sur la plateforme Render.
La base de données est hébergée sur Aiven.
Cette architecture permet une séparation entre :

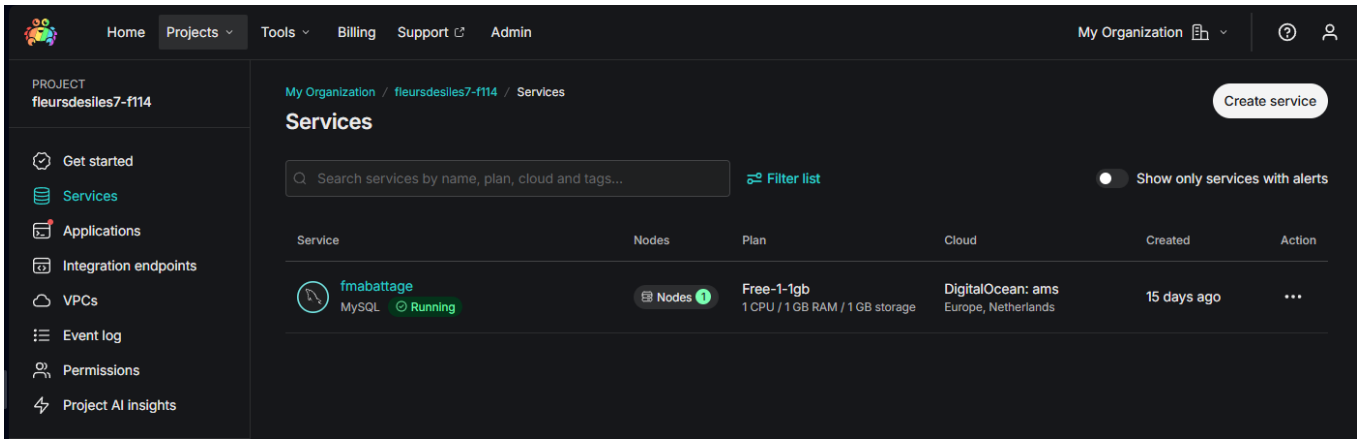
- L'application (front/back)
- La base de données

Environnement de production

Le déploiement permet une utilisation en conditions réelles, proche d'un environnement professionnel. Cela permet de tester :

- La stabilité de l'application
- L'accès distant aux données
- Le bon fonctionnement global du système





2. Précisez les moyens utilisés :

- Render (hébergement applicatif)
- Aiven (base de données)
- Documentation technique
- Configuration serveur

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce projet seule.

4. Contexte

Nom de l'entreprise, organisme ou association ▸ *FMAbattage*

Chantier, atelier, service ▸ Développement application métier/ projet personnel

Période d'exercice ▸ Du : *24/04/2026* au : *11/06/2026*

5. Informations complémentaires *(facultatif)*

Le déploiement constitue la dernière étape du projet et permet de rendre l'application utilisable en dehors de l'environnement de développement local.

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Cliquez ici.	Cliquez ici pour taper du texte.	Cliquez ici pour sélectionner une date.

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Lydie Meunier ,

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à Villeurbanne le 11/06/2026

pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

ANNEXES

README du repositorie GitHub

FM Abattage - Application de gestion
Application web de gestion développée pour l'entreprise FM Abattage afin de simplifier le suivi administratif quotidien.
Fonctionnalités
Gestion des clients

- *Création et modification des fiches clients*
- *Consultation de la liste des clients*

Gestion des devis

- *Création de devis*
- *Modification des devis*
- *Génération PDF*
- *Gestion des statuts :*
    - *En attente*
    - *Accepté*
    - *Refusé*
    - *Archivé*

Gestion des factures

- *Création de factures à partir des devis*
- *Génération PDF*
- *Suivi du paiement des factures*

Gestion des paiements

- *Enregistrement des paiements*
- *Gestion des paiements partiels*
- *Calcul automatique du reste dû*

Gestion des relances

- *Création des relances clients*
- *Historique des relances*
- *Gestion des mises en demeure*
- *Génération PDF*

Gestion des dépenses

- *Enregistrement des dépenses de l'entreprise*
- *Suivi financier*

Tableau de bord

- *Chiffre d'affaires*
- *Montant encaissé*
- *Factures en attente*
- *Dépenses*
- *Résultat réel (paiements reçus - dépenses)*

Technologies utilisées

Front-end

- *HTML5*
- *CSS3*
- *Bootstrap 5*
- *JavaScript*

Back-end

- *Node.js*
- *Express*

Base de données

- *MySQL*
- *Sequelize ORM*

Hébergement

- *Render*
- *Aiven MySQL*

Installation

Cloner le projet

git clone https://github.com/Lydie-pixel/FmAbattage-Management-app.git

Installer les dépendances

npm install

Configurer les variables d'environnement

Créer un fichier .env :

DB_HOST=
DB_PORT=
DB_NAME=
DB_USER=
DB_PASSWORD=
SESSION_SECRET=
NODE_ENV=development

Lancer le projet

npm start

Auteur

Projet développé par Lydie dans le cadre de la gestion administrative de l'entreprise FM Abattage.

État du projet

Projet actuellement utilisé en conditions réelles et en cours d'amélioration continue selon les besoins des utilisateurs.

Dictionnaire de données

Table : users

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant unique de l'utilisateur
mail	VARCHAR(300)	NOT NULL, UNIQUE	Adresse email
username	VARCHAR(100)	NOT NULL, UNIQUE	Nom d'utilisateur
role	ENUM('user','admin')	DEFAULT 'user'	Rôle de l'utilisateur
password_hash	VARCHAR(255)	NOT NULL	Mot de passe sécurisé

Table : clients

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant client
nom	VARCHAR(500)	NOT NULL	Nom du client
tel	VARCHAR(20)	NOT NULL	Numéro de téléphone
email	VARCHAR(255)	NULL	Email
adresse	VARCHAR(255)	NOT NULL	Adresse postale
code_postal	VARCHAR(5)	NOT NULL	Code postal
ville	VARCHAR(255)	NOT NULL	Ville
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date de création

Table : devis

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant devis
numero	VARCHAR(50)	UNIQUE, NOT NULL	Numéro du devis
client_id	INT	FK → clients.id	Client associé
date_devis	DATE	NOT NULL	Date du devis
date_echeance	DATE	NULL	Date limite
frais_deplacement	DECIMAL(10,2)	DEFAULT 0	Frais de déplacement
montant	DECIMAL(10,2)	NOT NULL	Montant total
statut	ENUM('en_attente','accepte','refuse')	DEFAULT 'en_attente'	Statut du devis
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Table : devis_items

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant ligne
devis_id	INT	FK → devis.id	Devis associé
description	TEXT	NOT NULL	Description prestation
quantite	INT	NOT NULL	Quantité
prix_unitaire	DECIMAL(10,2)	NOT NULL	Prix unitaire
total_ligne	DECIMAL(10,2)	NOT NULL	Total ligne
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Table : factures

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant facture
numero	VARCHAR(50)	UNIQUE, NOT NULL	Numéro facture
devis_id	INT	FK → devis.id	Devis lié
client_id	INT	FK → clients.id	Client associé
date_facture	DATE	NOT NULL	Date facture
frais_deplacement_final	DECIMAL(10,2)	DEFAULT 0	Frais final
montant	DECIMAL(10,2)	NOT NULL	Montant total
statut	ENUM('en_attente','payee','partielle')	DEFAULT 'en_attente'	Statut paiement
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Table : paiements

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant paiement
facture_id	INT	FK → factures.id	Facture liée
montant	DECIMAL(10,2)	NOT NULL	Montant payé
date_paiement	DATE	NOT NULL	Date paiement
mode_paiement	ENUM(...)	NOT NULL	Mode de paiement
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Table : depenses

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant dépense
date_depenses	DATE	NOT NULL	Date dépense
montant	DECIMAL(10,2)	NOT NULL	Montant
type	ENUM('frais_carburant','frais_materiel','charges','autre')	NOT NULL	Type de dépense
description	VARCHAR(250)	NULL	Description
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Table : relances

Champ	Type	Contraintes	Description
id	INT	PK, AI	Identifiant relance
facture_id	INT	FK → factures.id	Facture concernée
niveau	ENUM(...)	NOT NULL	Niveau de relance
date_relance	DATE	NOT NULL	Date de relance
penalites	DECIMAL(10,2)	DEFAULT 0	Pénalités
delai_avant_poursuite	INT	DEFAULT 0	Délai
numero_ar	VARCHAR(50)	NULL	Numéro recommandé
statut	ENUM(...)	DEFAULT 'envoyee'	Statut
commentaire	VARCHAR(600)	NULL	Commentaire
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date création

Exemple de PDF

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Devis

Client
Laura Chevalier
9 rue Nationale
42300 Roanne
France
0666889900

Numéro du devis :	D-2026-004
Date du devis :	01/04/2026
Validité :	14
Date d'échéance :	15/04/2026
Total à payer :	**320,00 € €**

Description	Quantité	Prix unitaire	Montant
Abattage	1.00	250,00 €	250,00 €
Nettoyage	1.00	70,00 €	70,00 €
	Frais de carburant estimés :		15,00 €
	Total HT :		**320,00 € €**

Lu et approuvé

Le :

Signature :

Devis reçu avant l'exécution des travaux
Les prix indiqués pourront être révisés en fonction de l'évolution du coût du carburant entre la date du devis et la réalisation de la prestation

Conditions de paiement :
Mentions :

TVA non applicable – Article 293B du Code général des impôts
RÉSERVE DE PROPRIÉTÉ : Les marchandises restent notre propriété jusqu'à complet paiement

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Facture

Client
Laura Chevalier
9 rue Nationale
42300 Roanne
France
0666889900

Numéro de la facture :	F-2026-007
Date de la facture :	05/04/2026
Validité :	30 jours
Date d'échéance :	05/05/2026
Total à payer :	**320,00 €**

Description	Quantité	Prix unitaire	Montant
Abattage	1.00	250,00 €	250,00 €
Nettoyage	1.00	70,00 €	70,00 €
	Frais de carburant final :		15,00 €
	Total HT :		**320,00 €**

Le :

Signature :

Conditions de paiement :
Mentions :

TVA non applicable – Article 293B du Code général des impôts

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Laura Chevalier
9 rue Nationale
42300 Roanne
0666889900

Objet : Lettre de première relance amiable – Facture n° F-2026-007

Saint Paul de Varax, le 16/05/2026

Madame, Monsieur,

Sauf erreur ou omission de notre part, le règlement de la facture n° F-2026-007 d'un montant de 320,00 €, émise le 05/04/2026 et arrivée à échéance le 05/05/2026, ne nous est pas encore parvenu à ce jour.

Nous vous remercions de bien vouloir procéder à son règlement dans les meilleurs délais.

Vous trouverez ci-joint une copie de la facture concernée afin de faciliter vos démarches. Si votre paiement a été effectué entre-temps, nous vous remercions de ne pas tenir compte de la présente relance.

À défaut de régularisation, des pénalités de retard prévues par nos conditions générales de vente pourront être appliquées.

Restant à votre disposition pour toute information complémentaire, nous vous prions d'agréer, Madame, Monsieur, l'expression de nos salutations distinguées.

Franck Meunier
Dirigeant de FM Abattage

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Laura Chevalier
9 rue Nationale
42300 Roanne
0666889900

Objet : Lettre de deuxième relance de paiement – Facture n° F-2026-007

Saint Paul de Varax, le 29/05/2026

Madame, Monsieur,

Malgré notre précédente relance en date du 16/05/2026, nous constatons que la facture n° F-2026-007 d'un montant de 320,00 €, émise le 05/04/2026 et arrivée à échéance le 05/05/2026, demeure impayée à ce jour.

Nous vous remercions de bien vouloir régulariser votre situation dans les meilleurs délais afin d'éviter l'application de pénalités de retard prévues par nos conditions générales de vente.

Vous trouverez ci-joint une copie de la facture concernée ainsi qu'une copie de notre précédent courrier de relance.

Si votre règlement nous a été adressé entre-temps, nous vous remercions de ne pas tenir compte de ce courrier.

Nous restons à votre disposition pour toute information complémentaire et vous prions d'agréer, Madame, Monsieur, l'expression de nos salutations distinguées.

Franck Meunier
Dirigeant de FM Abattage

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Laura Chevalier
9 rue Nationale
42300 Roanne
0666889900

Objet : Dernière relance avant mise en demeure – Facture n° F-2026-007

Saint Paul de Varax, le 18/05/2026

Madame, Monsieur,

À ce jour, malgré nos précédents courriers de relance en date des 16/05/2026 et 29/05/2026, nous n'avons toujours pas reçu le règlement de la facture n° F-2026-007, d'un montant de 320,00 €, arrivée à échéance le 05/05/2026.

Nous vous demandons de procéder au règlement immédiat de cette facture.

À défaut de régularisation rapide de votre situation, nous nous verrons contraints d'engager une procédure de recouvrement et d'appliquer les pénalités de retard prévues par nos conditions générales de vente.

Vous trouverez ci-joint une copie de la facture concernée ainsi que des précédentes relances.

Dans le cas où votre paiement aurait été effectué avant réception du présent courrier, nous vous remercions de ne pas tenir compte de cette relance.

Dans l'attente de votre retour, veuillez agréer, Madame, Monsieur, l'expression de nos salutations distinguées.

Franck Meunier
Dirigeant de FM Abattage

FM ABATTAGE
14 rue du Thou
01240 Saint Paul de Varax
Tél : 06 78 69 66 86
Email : fmabattage@gmail.com

Laura Chevalier
9 rue Nationale
42300 Roanne
0666889900

Objet : Mise en demeure de payer
Lettre recommandée avec accusé de réception
AR n° 1A 0123 4567 7891 234 2

Saint Paul de Varax, le 01/06/2026

Madame, Monsieur,

Malgré nos précédentes relances restées sans réponse en date des 16/05/2026, 29/05/2026 et 18/05/2026, la facture n° F-2026-007, émise le 05/04/2026 et arrivée à échéance le 05/05/2026, demeure impayée à ce jour.

Par la présente, nous vous mettons en demeure de régler la somme totale de 370,00 €, correspondant :

- 320,00 € au montant principal de la facture ;
- 50,00 € de pénalités de retard applicables conformément à nos conditions générales de vente et à l'article L441-10 du Code de commerce.

Nous vous demandons de procéder au règlement sous un délai de 20 jours à compter de la réception du présent courrier.

À défaut de paiement dans ce délai, nous serons contraints d'engager une procédure de recouvrement contentieuse afin d'obtenir le règlement des sommes dues, ainsi que l'indemnisation du préjudice subi.

Vous trouverez ci-joint une copie de la facture concernée ainsi que des précédents courriers de relance.

Si votre règlement a été effectué avant réception de ce courrier, nous vous remercions de ne pas tenir compte de la présente mise en demeure.

Veuillez agréer, Madame, Monsieur, l'expression de nos salutations distinguées.

Franck Meunier
Dirigeant de FM Abattage

Lydie Meunier

Fiche client

Informations

Téléphone
0631152810

Email
fleursdesiles7@gmail.com

Adresse
140 cours emile zola 69100 Villeurbanne

Commentaires

Aucun commentaire pour le moment.

Devis

Numéro	Date	Montant	Statut
Aucun devis			

Factures

Numéro	Date	Montant	Statut
Aucune facture			

Paiements

Facture	Date	Montant	Mode
Aucun paiement			

Relances

Facture	Niveau	Date	Pénalité
Aucune relance			

Laura Chevalier

Fiche client

Informations

Téléphone
0666889900

Email
laura.chevalier@mail.com

Adresse
9 rue Nationale 42300 Roanne

Commentaires

Aucun commentaire pour le moment.

Devis

Numéro	Date	Montant	Statut
D-2026-004		320,00 €	Acceptée

Factures

Numéro	Date	Montant	Statut
F-2026-007	05/04/2026	320,00 €	En attente

Paiements

Facture	Date	Montant	Mode
Aucun paiement			

Relances

Facture	Niveau	Date	Pénalité
F-2026-007	Première relance	16/05/2026	0,00 €
F-2026-007	Deuxième relance	29/05/2026	20,00 €
F-2026-007	Troisième relance	18/05/2026	0,00 €

Documentation Swagger de l'API

API FM Abattage

1.0.0 OAS 3.0

Documentation API du logiciel de gestion FM Abattage

Servers

https://fmabattage-management-app-1.onrender.com

Stats

GET	/api/stats/month/{year}/{month}	Récupérer les statistiques mensuelles
GET	/api/stats/year/{year}	Récupérer les statistiques annuelles

Relance

POST	/api/relance	Créer une relance
GET	/api/relance	Récupérer toutes les relances
GET	/api/relance/{id}	Récupérer une relance
PUT	/api/relance/{id}	Modifier une relance
DELETE	/api/relance/{id}	Supprimer une relance

Pdf

GET	/api/pdf/devis/{id}	Génère dynamiquement le PDF d'un devis à partir de son identifiant
GET	/api/pdf/facture/{id}	Génère dynamiquement le PDF d'une facture à partir de son identifiant

GET	/api/pdf/relance/{id}	Génère dynamiquement le PDF d'une relance à partir de son identifiant
GET	/api/pdf/client/{id}	Génère dynamiquement le PDF d'un client à partir de son identifiant

Paiement

POST	/api/paiement	Ajouter un paiement
GET	/api/paiement	Récupérer tous les paiements
GET	/api/paiement/{id}	Récupérer un paiement
PUT	/api/paiement/{id}	Modifier un paiement
DELETE	/api/paiement/{id}	Supprimer un paiement

Facture

GET	/api/facture	Récupérer toutes les factures
GET	/api/facture/archives	Récupérer les factures archivées
GET	/api/facture/{id}	Récupérer une facture
DELETE	/api/facture/{id}	Supprimer une facture
PUT	/api/facture/{id}/statut	Modifier le statut d'une facture
POST	/api/facture/from-devis/{id}	Créer une facture à partir d'un devis
PUT	/api/facture/{id}/archive	Archiver une facture

Export

GET	/api/export/factures	Exporter les factures
GET	/api/export/depenses	Exporter les dépenses

GET	/api/export/relances	Exporter les relances
GET	/api/export/depenses-par-type	Exporter les dépenses par type

Devis

POST	/api/devis	Ajouter un devis
GET	/api/devis	Récupérer tous les devis
GET	/api/devis/archives	Récupérer les devis archivés
GET	/api/devis/accueil	Récupérer les devis affichés sur l'accueil
GET	/api/devis/{id}	Récupérer un devis
PUT	/api/devis/{id}	Modifier un devis
DELETE	/api/devis/{id}	Supprimer un devis
PUT	/api/devis/{id}/statut	Modifier le statut d'un devis
PUT	/api/devis/{id}/archive	Archiver un devis

Dépense

POST	/api/depense	Ajouter une dépense
GET	/api/depense	Récupérer toutes les dépenses
GET	/api/depense/stats	Récupérer les statistiques des dépenses
GET	/api/depense/by-type	Récupérer les dépenses par type
GET	/api/depense/{id}	Récupérer une dépense
DELETE	/api/depense/{id}	Supprimer une dépense

Clients

POST	/api/client	Ajouter un client
GET	/api/client	Récupère tous les clients
GET	/api/client/{id}	Récupérer un client
DELETE	/api/client/{id}	Supprimer un client
PUT	/api/client/{id}	Modifier un client

Auth

POST	/api/auth/login	Connexion utilisateur
POST	/api/auth/logout	Déconnexion utilisateur
GET	/api/auth/me	Récupérer l'utilisateur connecté