

Rapport de Mission : Optitop

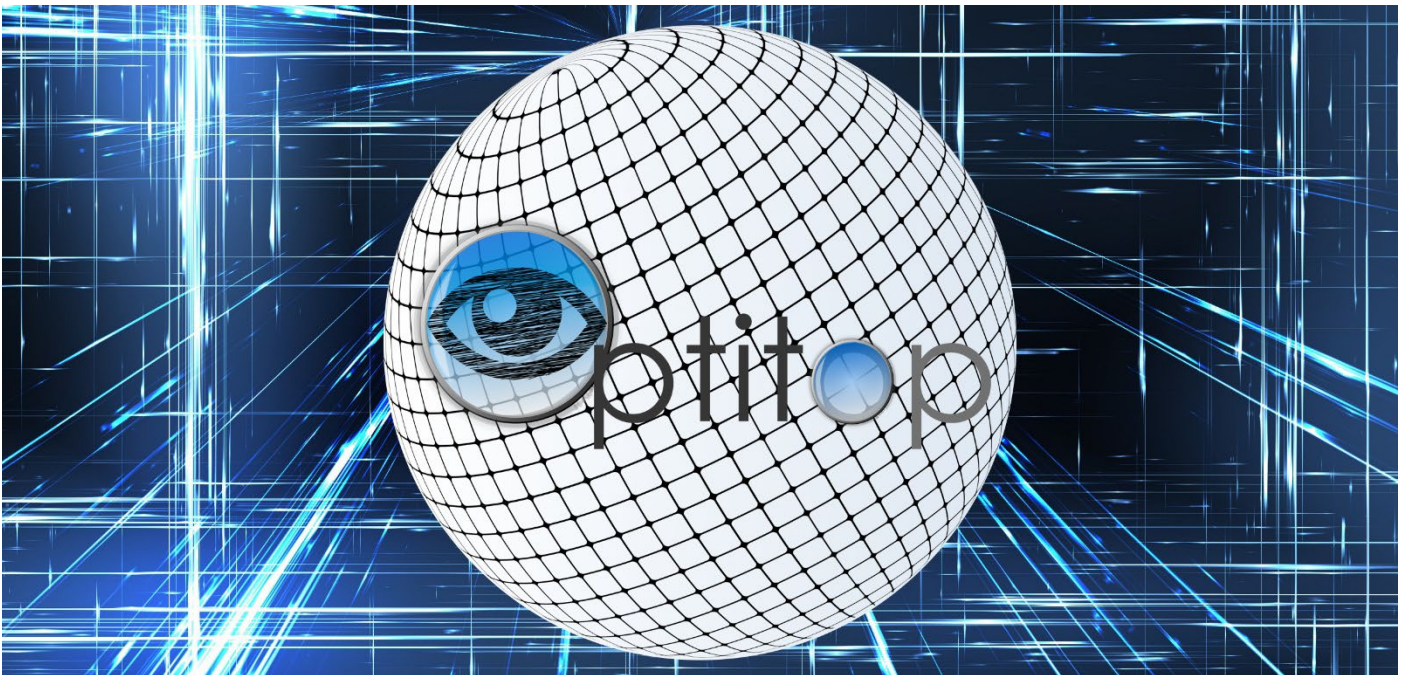


Table des matières

I.	INTRODUCTION	5
II.	PREPARATION ET FRONTEND	6
A.	Mockups de l'application web.....	6
1.	Mockups.....	6
B.	Création de l'interface utilisateur.....	7
1.	IHM de l'application web.....	7
a)	Connexion et Compte	7
➤	Connexion	7
➤	La page Mon compte.....	7
❖	L'onglet Profil	8
❖	L'onglet Password.....	9
b)	La page Importation.....	9
c)	La page Dashboard et son affichage selon l'utilisateur	10
➤	Admin, supermanager et manager	11
➤	Collaborateur	11
d)	La page Devis Optiques et son affichage selon l'utilisateur.....	12
➤	Admin, supermanager, manager	12
➤	Collaborateur	13
e)	La page Paniers Moyens & Primes.....	14
f)	La page Chiffre d'Affaires (CA).....	15
g)	La page Synthèse	16
h)	La page Comptes.....	17
➤	Admin	18
➤	Supermanager et manager	19
2.	IHM de l'application Mobile	20
a)	Connexion.....	20
b)	La page Comptes.....	21
➤	Vue d'ensemble	21
❖	Admin.....	21
❖	Supermanager	22
➤	Nouvelle demande	22
❖	Ajout.....	22
❖	Modification	24
❖	Suppression	25
C.	Réalisation des modélisations de données.....	27
1.	Diagrammes de Cas d'utilisation.....	27
a)	Application Web	27
b)	Application Mobile.....	28
2.	Base de données Optitop	29
a)	Modèle logique de données.....	29
b)	Contenu des tables	30
➤	user.....	30
➤	seller	30
➤	quotations_lines et invoices_lines	31
➤	quotations.....	33
➤	invoices	34
➤	pending_accounts	35
3.	Fichiers .csv	36
D.	Développement de l'application mobile	37
1.	Mise en place de l'environnement Flutter	37
a)	Installation de Flutter et configuration de VS Code	37
b)	Installation Android Studio et SDK Android	38
c)	Création d'un appareil virtuel Android (émulateur).....	42

d)	Vérification de l'environnement avec Flutter Doctor.....	43
III.	MISE EN PLACE DU BACKEND.....	44
A.	Base de Données.....	44
1.	Script de création.....	44
2.	Insertion de données fictives.....	50
3.	Déploiement de MySQL sur une machine virtuelle Ubuntu.....	50
a)	Installation de MySQL.....	51
b)	Sécurisation de MySQL.....	51
c)	Vérification du service MySQL.....	52
d)	Création de la Base de données Optitop.....	53
e)	Création des tables.....	54
➤	Génération du fichier SQL par phpmyadmin sur Windows (environnement de développement).....	54
➤	Import du fichier SQL.....	54
f)	Vérification des tables.....	54
g)	Création d'un utilisateur MySQL dédié à l'API.....	55
h)	Installation de phpMyAdmin sur Ubuntu.....	56
➤	Installation.....	56
➤	Test de fonctionnement.....	58
i)	Sauvegarde automatique de la base de données.....	59
➤	Création d'un utilisateur MySQL dédié aux sauvegardes.....	59
➤	Montage d'un disque réseau pour stocker les sauvegardes.....	60
➤	Création d'un script de sauvegarde automatique.....	62
➤	Planification de l'exécution automatique.....	63
B.	API REST Spring Boot.....	64
1.	L'API dans l'environnement de développement.....	64
a)	Installation.....	64
b)	Test du fonctionnement de l'API.....	65
2.	Documentation de l'API avec Swagger.....	68
a)	Installation.....	68
b)	Documentation des contrôleurs.....	68
c)	Routes de l'API.....	70
C.	Définition des différents environnements.....	72
1.	Mise en place des profils Spring Boot.....	72
2.	Développement (profil dev).....	73
3.	Pré-production (profil pprod).....	73
a)	Paramétrage de l'environnement.....	73
b)	Configuration de MySQL sur la VM.....	74
➤	Modification du fichier de configuration.....	74
➤	Ouverture du pare-feu pour le port 3306.....	75
c)	Tests de fonctionnement.....	76
➤	Lancement de l'application pour pprod.....	76
➤	Requête à l'API depuis Postman.....	76
➤	Appel via le frontend web : test sur l'import d'un .csv.....	77
4.	Conclusion.....	79
D.	Schéma définitif de l'infrastructure.....	80
IV.	VERSIONNAGE.....	81
A.	Dépôt GitHub du projet.....	81
B.	Initialisation du dépôt Optitop.....	81
C.	Utilisation de Git et GitHub.....	82
V.	IMPLEMENTATION.....	83

A. Choix des langages	83
1. JavaScript	83
2. PHP	83
a) Sécurisation de l'accès aux pages selon les rôles.....	83
3. Une complémentarité idéale entre PHP et Javascript	84
B. Préparation de fichiers .csv exploitables	85
1. Problématique.....	85
a) Confidentialité des données.....	85
b) Volume des données.....	85
c) Simplification du jeu de données	85
2. Script en Python	85
a) Avantages	85
➤ Automatisation et gain de temps	85
➤ Facilité de manipulation des données avec Pandas	85
➤ Flexibilité.....	86
➤ Respect de la logique métier	86
b) Installation des outils nécessaires	86
➤ Python et pandas	86
➤ Chardet	86
c) Script	87
VI. SECURITE	89
A. Chiffrement des mots de passe avec BCrypt	89
1. Ajout des dépendances dans pom.xml.....	89
2. Création du service de hachage	89
3. Modification des Controllers	90
a) Contrôleur de changement de mot de passe.....	90
b) Contrôleur d'authentification.....	91
4. Script de migration des mots de passe.....	92
B. Gestion des mots de passe en cas d'ajout d'utilisateur.....	93
1. Contexte.....	93
2. Configuration SMTP pour Gmail	94
3. Stockage des paramètres nécessaires en base de données.....	96
a) Création de la table email_config.....	96
b) Insertion des valeurs dans la table email_config	96
c) Création de l'entité Springboot EmailConfig.....	97
4. Modification des fichiers Service.java impliqués.....	98
a) Création de EmailService.java	98
b) Modification de PendingAccountService.java.....	99
5. Test fonctionnel : Envoi d'un email à la création d'un compte utilisateur	99
a) Objectif.....	99
b) Mise en œuvre.....	100
c) Résultat	100
VII. GESTION DE PROJET ET ORGANISATION.....	102
A. Trello pour le suivi du projet	102
1. Méthode Kanban.....	102
2. Avantages.....	105

I. Introduction

L'entreprise support de ce projet est un magasin d'optique franchisé dans lequel je travaille. Elle propose des prestations classiques du secteur : vente de lunettes, lentilles, réalisation d'exams de vue, et accompagnement des clients. L'équipe est structurée autour de différents niveaux hiérarchiques : opticiens collaborateurs, manager, et un directeur (appelé ici supermanager) supervisant l'ensemble de l'activité.

En tant qu'opticienne manager au sein de cette structure, j'ai réalisé plusieurs fichiers Excel dédiés au suivi des objectifs commerciaux. Ces fichiers semi-automatisés sont fastidieux à remplir car ils sont alimentés manuellement à partir du logiciel métier utilisé (Cosium). Ce fonctionnement, bien que fonctionnel, est chronophage et source d'erreurs.

Le besoin de l'entreprise est donc clair : centraliser, automatiser et sécuriser la gestion des statistiques de vente à travers une application moderne, accessible selon le rôle de chaque utilisateur. L'objectif principal du projet Optitop est d'y répondre en proposant un outil applicatif capable de traiter les données de vente, d'en extraire des indicateurs pertinents (CA, taux de concrétisation, paniers moyens...) afin de les restituer de manière hiérarchisée via une interface claire et intuitive.

- L'application web permet aux différents profils utilisateurs d'accéder à des indicateurs personnalisés selon leurs responsabilités.
- Une version mobile complémentaire, développée avec Flutter, est dédiée à la gestion des comptes utilisateurs et réservée aux profils administratifs (admin et supermanager). Cette déclinaison mobile permet de répondre à des besoins urgents, tels que la création ou la suppression de comptes, même en situation de mobilité.

Sur le plan technique, Optitop repose sur une architecture robuste : une API REST sécurisée développée avec Spring Boot, connectée à une base de données MySQL hébergée sur une machine virtuelle Ubuntu à cet instant (environnement de pré-production).

L'authentification repose sur un système de rôles, avec mots de passe chiffrés, et une gestion fine des habilitations. Les données présentées sont entièrement fictives et anonymisées afin de respecter les contraintes de confidentialité.

Ce projet a été mené de manière autonome dans le cadre de ma formation BTS SIO option SLAM. Il illustre ma capacité à analyser un besoin métier réel, à structurer une réponse technique complète, et à organiser le développement d'une solution logicielle de bout en bout.

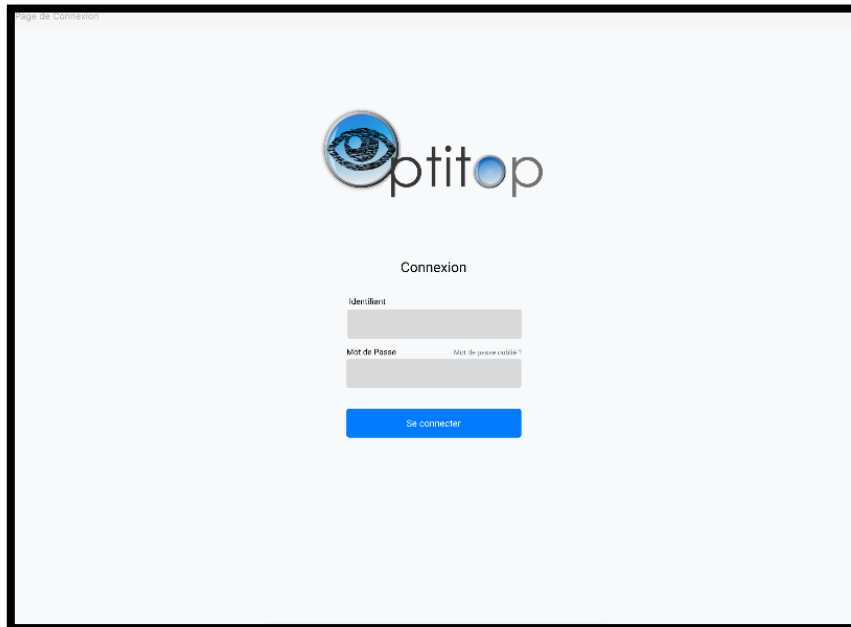


II. Préparation et Frontend

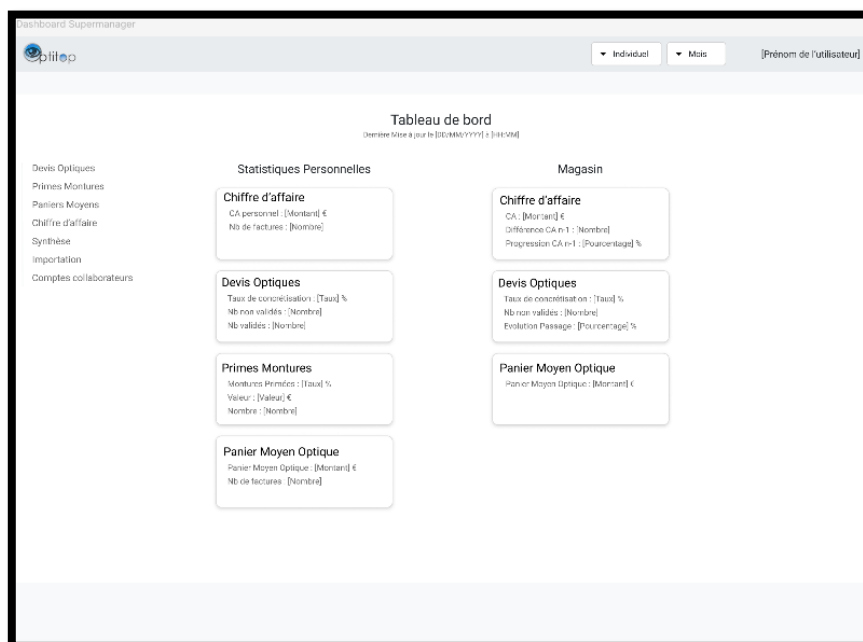
A. Mockups de l'application web

1. Mockups

Les mockups ont été réalisées avec le logiciel figma car c'est un outil complet que j'avais déjà rencontré lors d'un de mes stages. J'avais réalisé le logo avec Photoshop au préalable.



Page de connexion



Page d'accueil Supermanager

B. Création de l'interface utilisateur

J'ai choisi d'utiliser la bibliothèque de composants Bootstrap pour sa simplicité et son efficacité à créer rapidement des interfaces modernes et responsives. En l'installant via Node.js/NPM, je profite d'une gestion centralisée des dépendances, facilitant les mises à jour et l'intégration avec d'autres outils front-end. Cette approche garantit une structure professionnelle et évolutive pour mes interfaces utilisateur.

```
2 npm init -y
3 npm install bootstrap
4 npm install bootstrap-icons
```

Installation de Bootstrap avec npm

1. IHM de l'application web

a) Connexion et Compte

➤ Connexion



Connexion

Identifiant

Mot de Passe

[Mot de passe oublié ?](#)



Mot de passe : Réinitialisation

Identifiant

Email

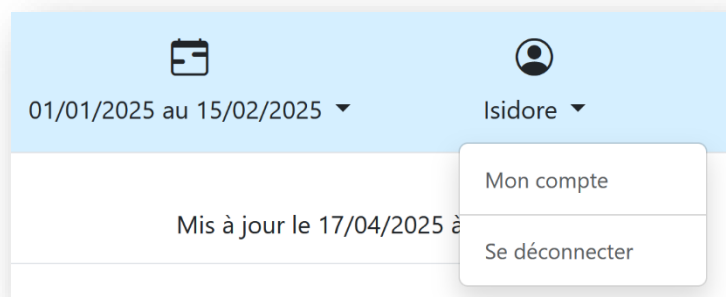
Votre nouveau mot de passe vous sera transmis par email

[Retour à la connexion](#)

Les pages de connexion et de mot de passe oublié

➤ La page Mon compte

Accessible depuis le menu déroulant en haut à droite, cette interface permet à chaque utilisateur connecté de gérer ses informations personnelles en toute autonomie.



Accès à la page Mon compte

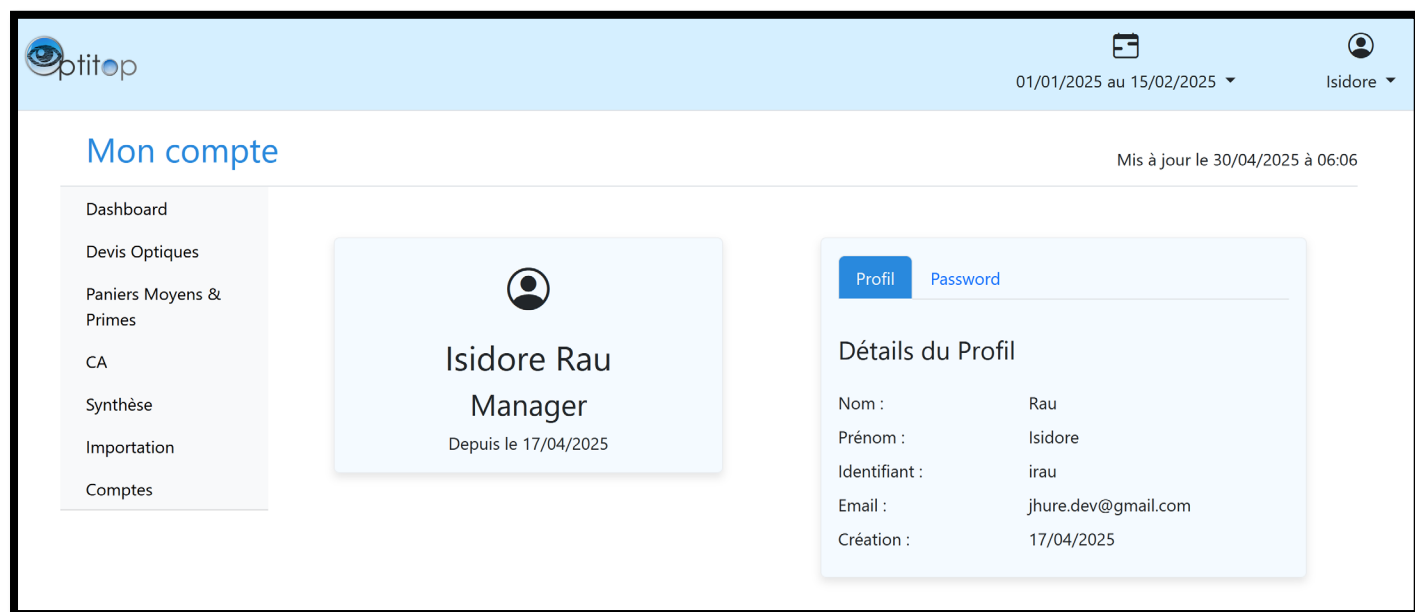
Elle est divisée en deux onglets principaux.

❖ L'onglet Profil

Cette section affiche les informations liées au compte :

- Nom, prénom
- Identifiant (login)
- Adresse e-mail
- Date de création du compte

Ces informations sont en lecture seule pour l'utilisateur, assurant l'intégrité des données de connexion.



Section Profil

❖ L'onglet Password

Cette section permet à l'utilisateur de modifier son mot de passe de manière sécurisée :

- L'ancien mot de passe est requis
- Le nouveau mot de passe doit être confirmé
- Complexité de 12 caractères, majuscule(s), minuscule(s), chiffre(s) et caractère(s) spéciaux
- Les deux saisies correspondent
- Hachage du nouveau mot de passe avant stockage en BDD
- Retours d'erreurs spécifiques pour guider l'utilisateur

L'objectif de cette page est de renforcer la sécurité et la confidentialité, tout en laissant une autonomie de gestion à chaque utilisateur, quel que soit son rôle.

The screenshot shows the 'Mon compte' (My account) page of the Optitop application. The header includes the Optitop logo, a date range from 01/01/2025 to 15/02/2025, and the user's name 'Isidore'. The main content area is titled 'Mon compte' and shows the user's profile as 'Isidore Rau, Manager' with a last update date of 30/04/2025 at 06:06. A sidebar on the left lists navigation options: Dashboard, Devis Optiques, Paniers Moyens & Primes, CA, Synthèse, Importation, and Comptes. The 'Password' tab is active, displaying fields for 'Actuel' (Current), 'Nouveau' (New), and 'Confirmation' (Confirmation), followed by a 'Changer le mot de passe' (Change password) button.

Section password qui permet de modifier son mot de passe

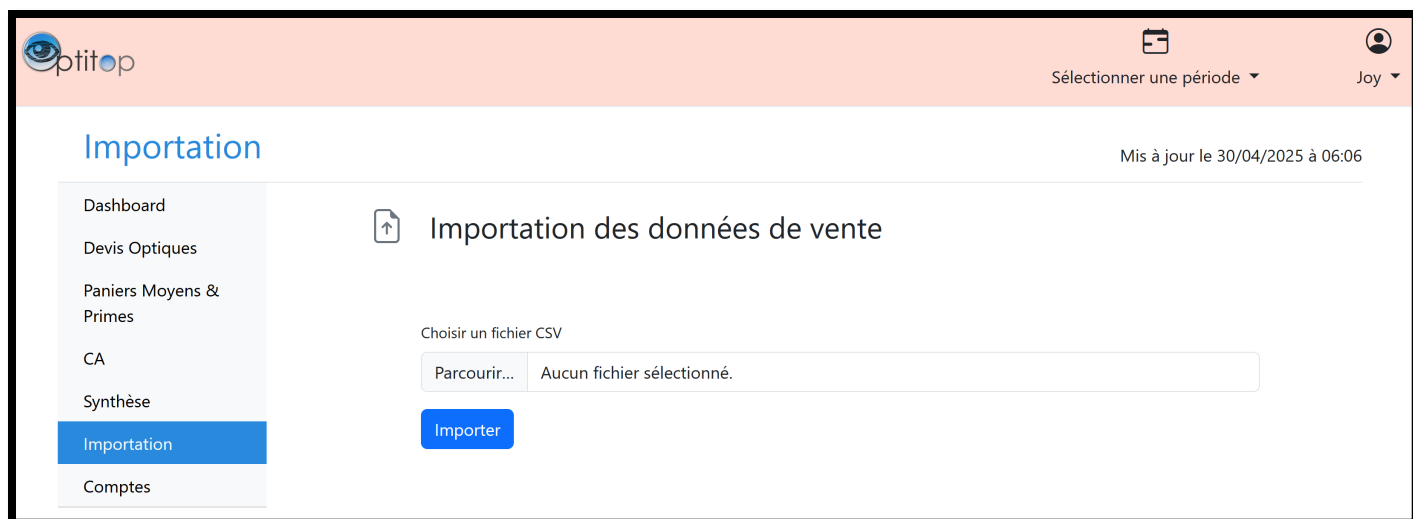
b) La page Importation

Cette interface permet aux utilisateurs autorisés (manager, supermanager, admin) de téléverser un fichier CSV contenant les données de vente issues du logiciel métier (Cosium).

Une fois importées en base de données, elles sont utilisées pour :

- Alimenter les statistiques de l'ensemble de l'application
- Mettre à jour les tableaux des devis, primes, paniers moyens et chiffres d'affaires

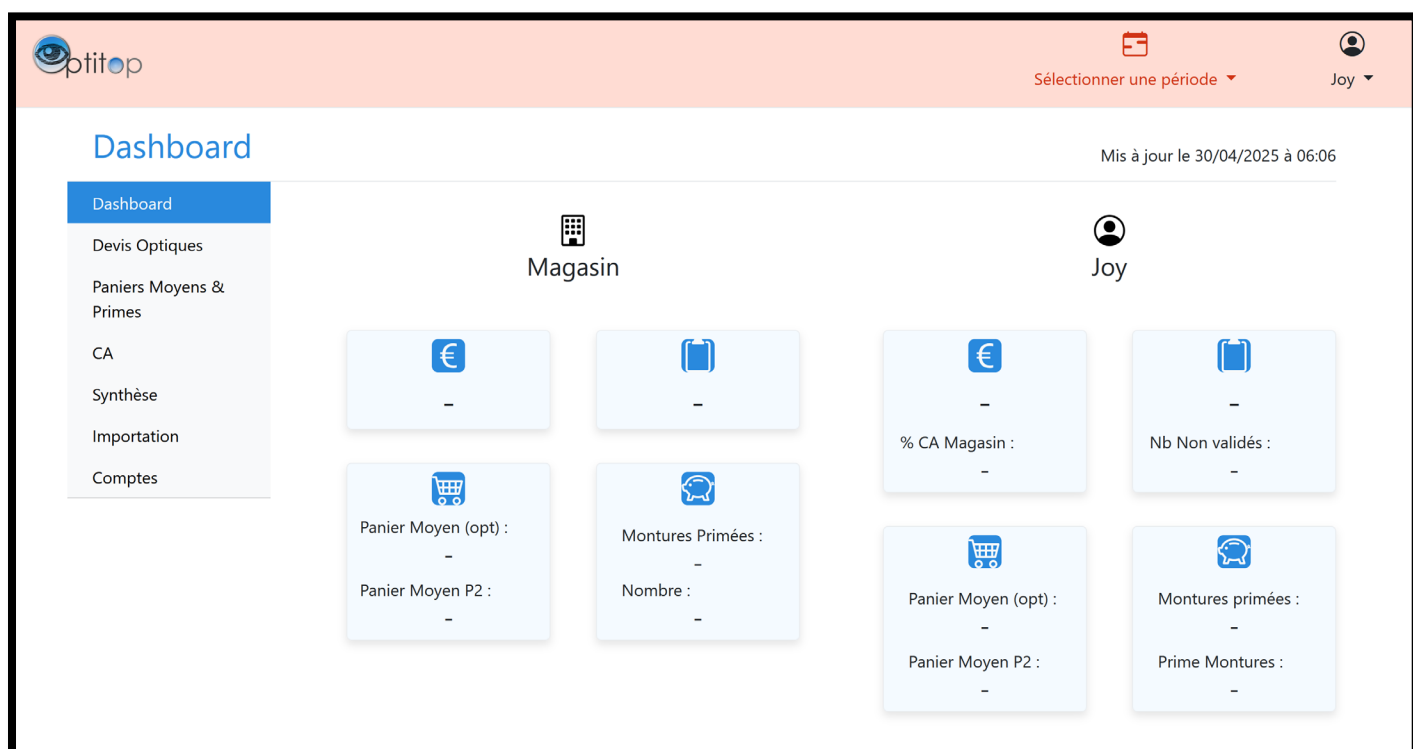
Le bouton "Importer" déclenche un traitement côté serveur via l'API Spring Boot, qui lit, valide, puis insère les données dans la base MySQL (insertion dans les tables **quotations_lines**, **quotations**, **invoices_lines**, **invoices**, **seller**)



La page Importation

c) La page Dashboard et son affichage selon l'utilisateur

La page Dashboard constitue le point d'entrée de l'application suite à la connexion de l'utilisateur. Cette interface a été pensée pour fournir une vue synthétique des statistiques commerciales du magasin ainsi que des performances individuelles de l'utilisateur connecté sur une période donnée. La première étape sur cette page consiste donc à sélectionner une période afin de récupérer en BDD, traiter puis afficher les données de la période définie.



Le Dashboard pour un profil admin avant de sélectionner la période de calcul des statistiques

➤ Admin, supermanager et manager

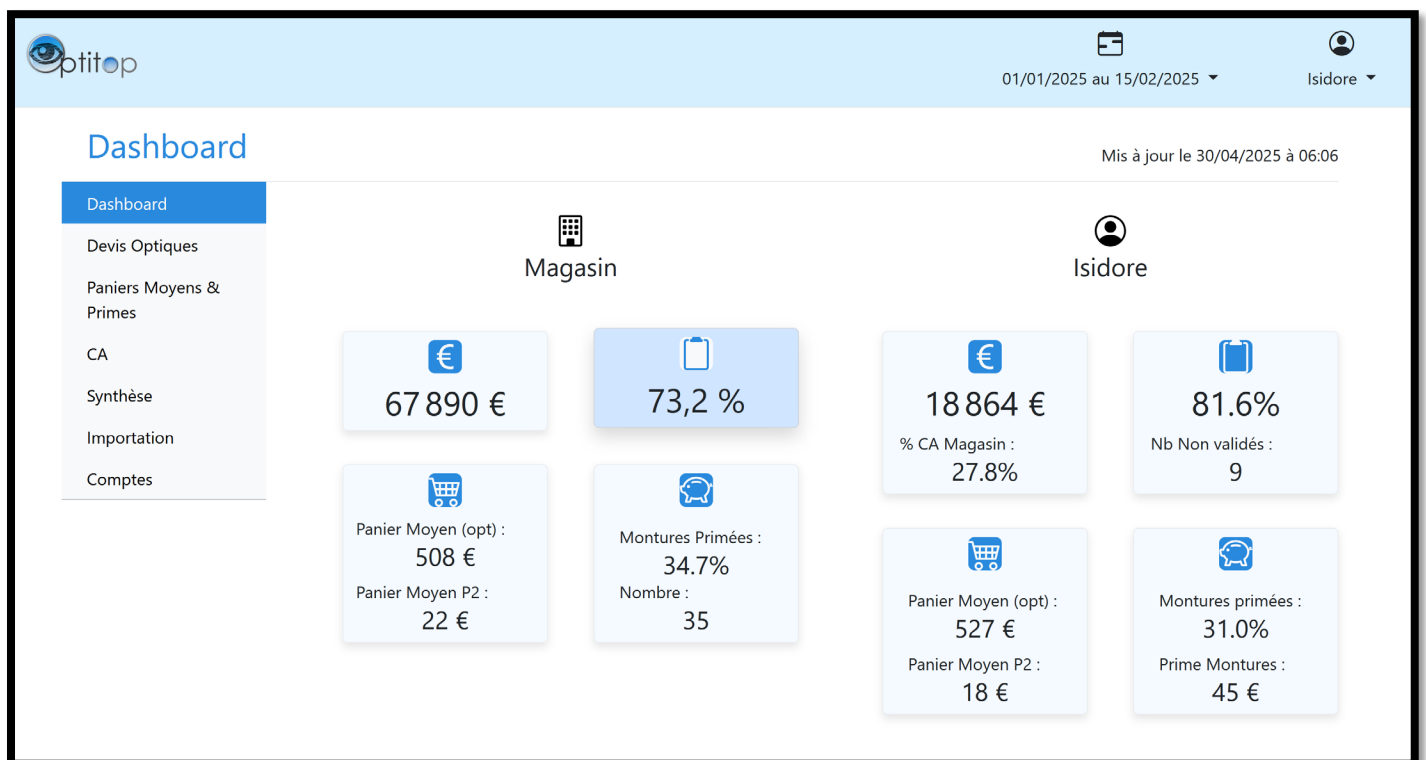
Les cartes situées en-dessous de « Magasin » sont des liens dont le style change au survol telle que :

- € représentant le CA
- 📄 représentant le taux de concrétisation
- 🛒 représentant le panier moyen optique
- 💰 représentant les montures optiques primées (200€ et plus)

Les cartes situées en-dessous du prénom de l'utilisateur ne sont pas des liens hormis celle concernant le taux de concrétisation des devis.

Les liens redirigent respectivement vers les pages « CA », « Devis Optiques », « Paniers Moyens & Primes » qui sont également accessibles par le menu de gauche.

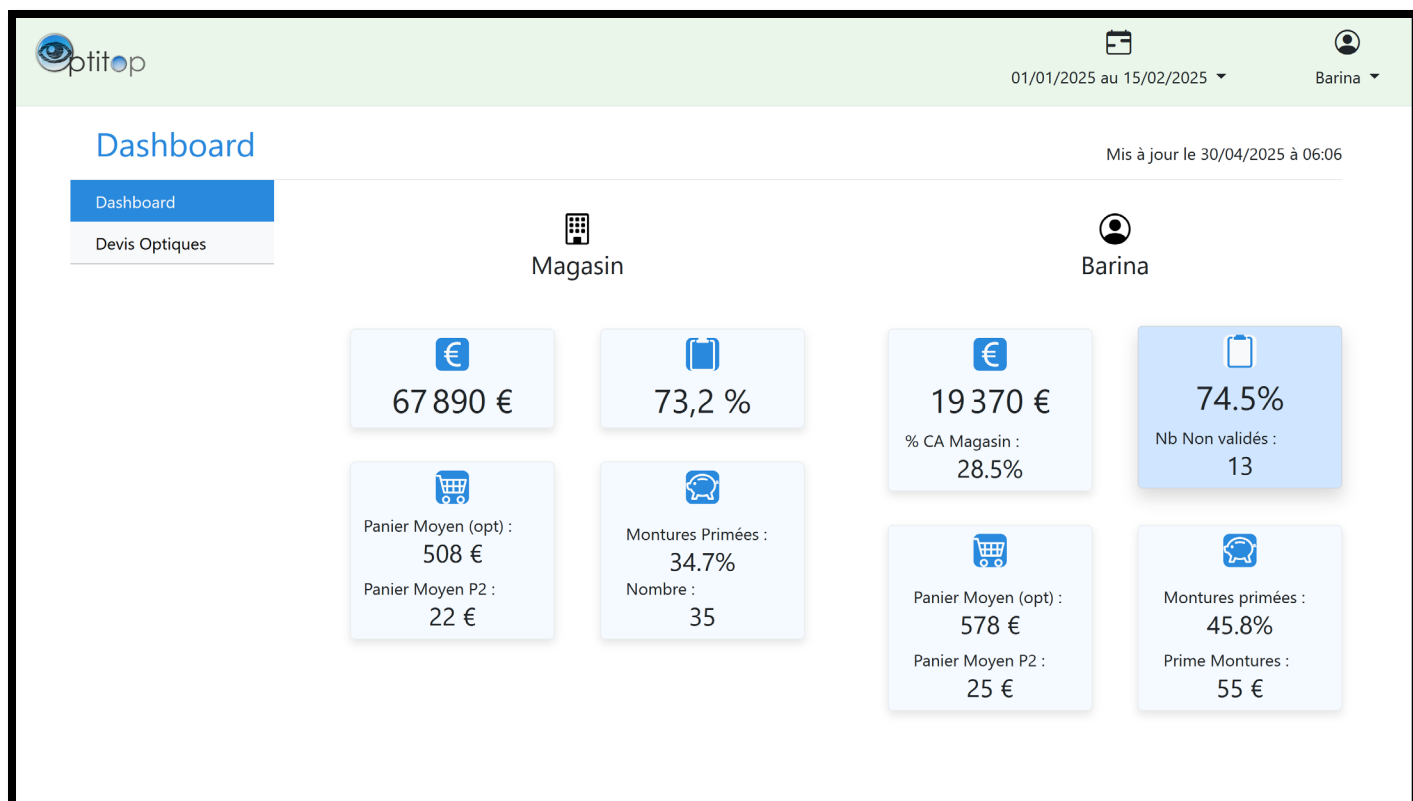
Toutes les valeurs affichées proviennent de la base de données, puis d'un traitement (calculs).



Page Dashboard d'un manager s'apprêtant à cliquer sur le lien vers la page « Devis Optiques »

➤ Collaborateur

Pour les collaborateurs, le seul lien disponible est celui concernant le taux de concrétisation personnel (en-dessous du prénom de l'utilisateur) et redirigeant vers la page « Devis Optiques » puisque les autres pages ne sont pas accessibles à ce rôle.



Page Dashboard du rôle collaborator s'apprêtant à cliquer sur le lien vers la page « Devis Optiques »

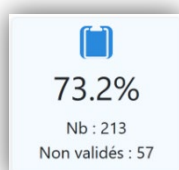
d) La page Devis Optiques et son affichage selon l'utilisateur

➤ Admin, supermanager, manager

Cette interface permet de visualiser et gérer la liste des devis optiques non validés sur une période donnée. Chaque ligne représente un devis, avec des informations essentielles telles que :

- La date de création
- Le collaborateur (par ses initiales) ayant saisi le devis
- Le client concerné
- La possibilité de sélectionner une action prédéfinie (**attente mutuelle, attente retour, à relancer, non validé, voir opticien**) → champ **action** de la table **quotations**
- L'ajout d'un commentaire libre → champ **comment** de la table **quotations**
- Il est possible de trier les lignes par Date, Nom ou Client

La page affiche également les taux de concrétisation calculés automatiquement :



pour l'ensemble du magasin

Nom	Nb	Non V.	Taux
BS	51	13	74.5%
EG	65	19	70.8%
HI	48	16	66.7%
IR	49	9	81.6%

par collaborateur

01/01/2025 au 15/02/2025
Ludovic

Devis Optiques

Mis à jour le 30/04/2025 à 06:06

Dashboard
Devis Optiques
Paniers Moyens & Primes
CA
Synthèse
Importation
Comptes

73.2%

Nb : 213
Non validés : 57

Nom	Nb	Non V.	Taux
BS	51	13	74.5%
EG	65	19	70.8%
HI	48	16	66.7%
IR	49	9	81.6%

Devis Non Validés

Date	Nom	Client	Action	Commentaire
12/02/2025	BS	Moron Nino	Sélectionner	TEST
12/02/2025	EG	Thiollet Louis	Sélectionner	
11/02/2025	BS	Toutain Malone	Voir opticien	
10/02/2025	EG	Thiollet Louis	Attente mutuelle	
08/02/2025	HI	Jacques Noa	Non validé	
06/02/2025	HI	Tribouiller Eva	A relancer	
05/02/2025	BS	Audrin Lenny	Attente de retour	
05/02/2025	HI	Martz Ilyes	Sélectionner	
04/02/2025	HI	Jacques Noa	Sélectionner	
04/02/2025	BS	Arnaudy Charline	Sélectionner	
04/02/2025	HI	Tascher Marceau	Sélectionner	
04/02/2025	BS	Allessandro Esteban	Sélectionner	
03/02/2025	EG	Mesnoux Sofia	Sélectionner	
03/02/2025	IR	Anglade Louise	Sélectionner	
03/02/2025	HI	Morot Adrien	Sélectionner	

Enregistrer

La page Devis Optiques d'un supermanager s'apprêtant à enregistrer une action et un commentaire sur un devis.

➤ Collaborateur

Cette page permet aux collaborateurs de consulter, commenter et mettre à jour le statut de leurs devis non validés sur une période définie.

Cette interface assiste chaque collaborateur dans le suivi de ses ventes, permet de réagir en temps réel à la validation des devis et de communiquer efficacement avec ses supérieurs hiérarchiques, notamment le manager et le supermanager.

Les collaborateurs n'ont pas accès aux taux de concrétisation par collaborateurs.

01/01/2025 au 15/02/2025
Barina

Devis Optiques

Mis à jour le 17/04/2025 à 05:55

Dashboard
Devis Optiques

Devis Non Validés

Date	Nom	Client	Action	Commentaire
12/02/2025	BS	Moron Nino	Sélectionner	TEST
11/02/2025	BS	Toutain Malone	Sélectionner	
05/02/2025	BS	Audrin Lenny	Sélectionner	
04/02/2025	BS	Arnaud Charline	Sélectionner	
04/02/2025	BS	Allessandro Esteban	Sélectionner	
03/02/2025	BS	Montazel Mahé	Sélectionner	
29/01/2025	BS	Martin Margaux	Sélectionner	
27/01/2025	BS	Moron Nino	Sélectionner	
16/01/2025	BS	Herrera Inès	Sélectionner	
11/01/2025	BS	Moniri Léandre	Sélectionner	
10/01/2025	BS	Alliaume Lois	Sélectionner	
09/01/2025	BS	Alarçon Elio	Sélectionner	
07/01/2025	BS	Poulain Louis	Sélectionner	

Enregistrer

73.2%
Nb : 213
Non validés : 57

Page Devis Optiques pour le rôle « collaborator »

e) La page Paniers Moyens & Primes

Les utilisateurs de rôle “**collaborator**” n’ont pas accès à cette page, conformément aux règles de permissions définies dans l’application.

Cette interface permet aux utilisateurs de suivre les performances commerciales de leur équipe, à travers deux indicateurs principaux :

Les paniers Moyens (des ventes optiques)

Un tableau regroupe pour chaque collaborateur :

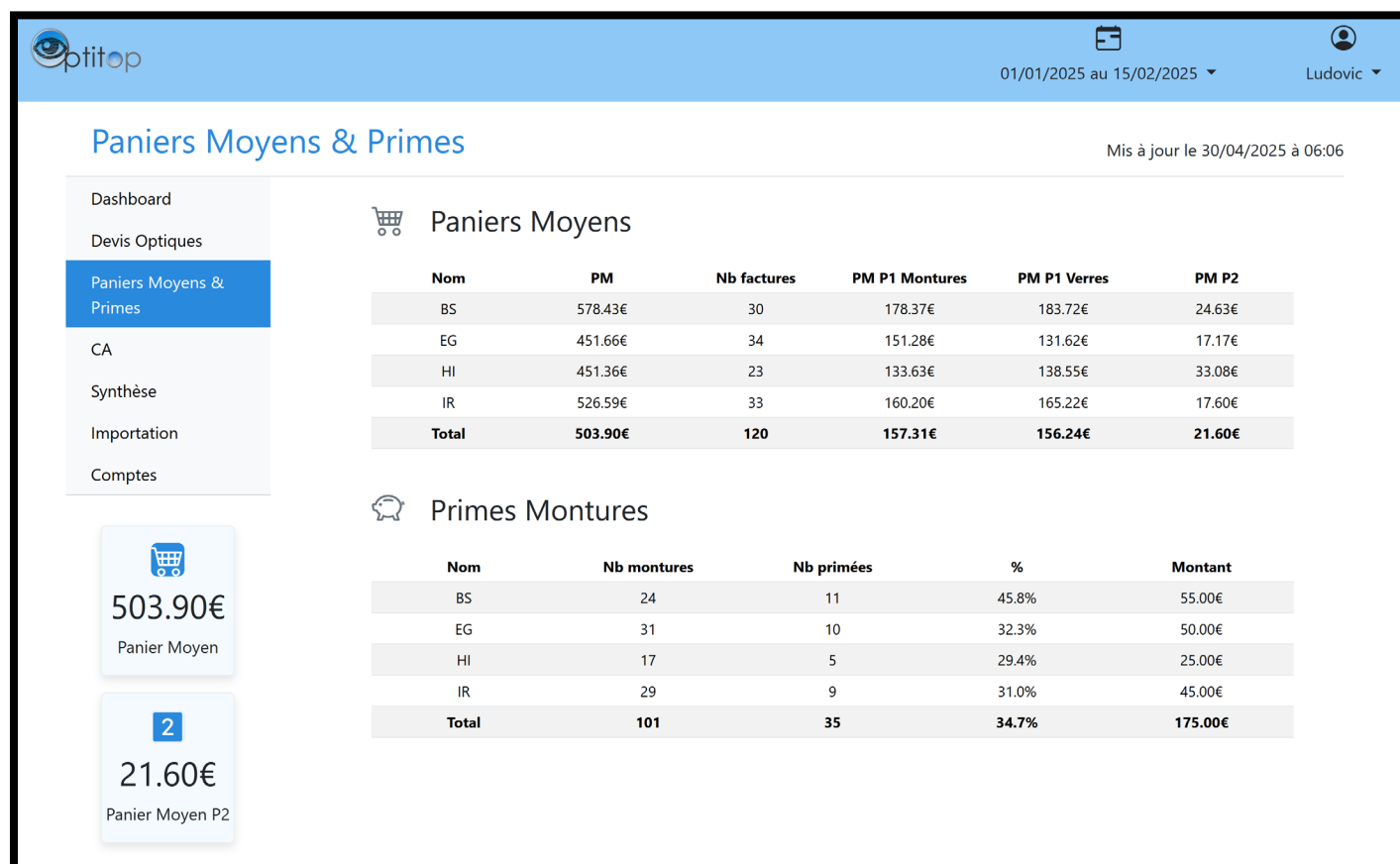
- Le panier moyen global d’une vente optique (contenant au minimum une ligne de family = « VER » en BDD
- Le nombre de factures émises
- Le panier moyen des montures vendues en 1ere paire
- Le panier moyen des verres vendus en 1ere paire
- Le panier moyen concernant les 2emes paires de lunettes

Les primes sur les montures optiques vendues

Un second tableau synthétise :

- Le nombre de montures vendues
- Le nombre de montures “primées” (éligibles à une prime)
- Le pourcentage de primes obtenues sur le nombre de montures vendues en 1ere paire
- Le montant total de la prime obtenue

Ces données permettent un suivi précis de l'efficacité commerciale et de la valorisation du travail de chaque collaborateur.



Page des Paniers moyens & Primes

f) La page Chiffre d'Affaires (CA)

Cette page permet de suivre l'évolution du chiffre d'affaires (CA) sur plusieurs années, mois par mois, les rôles « **collaborator** » n'y ont pas accès. Elle propose une visualisation claire du CA par période, accompagnée d'indicateurs de performance :

- Delta n-1 : différence entre l'année en cours et l'année précédente
- Delta % : variation en pourcentage

Les données futures sont affichées comme "À venir" pour signaler leur absence.

Le composant latéral affiche :

- Le CA cumulé de la période définie par l'utilisateur
- Le taux de progression par rapport à la même période n-1
- La répartition du CA par collaborateur sous forme de tableau

Cette interface est un véritable outil d'aide à la décision pour les managers et supermanagers, leur permettant d'identifier rapidement les périodes fortes ou faibles, et de motiver les équipes avec des objectifs réalistes.

Dashboard

Devis Optiques

Paniers Moyens &
Primes

CA

Synthèse

Importation

Comptes



Année

2025 ▼

2025	Janv.	Fev.	Mars	Avril	Mai	Juin	Juill.	Août	Sept.	Oct.	Nov.	Déc.	Année
CA	43 898 €	23 487 €	0 €	0 €	0 €	À venir	À venir	À venir	À venir	À venir	À venir	À venir	67 386 €
Delta n-1	+4 297 €	+3 305 €	Inconnu	Inconnu	Inconnu	À venir	À venir	À venir	À venir	À venir	À venir	À venir	+7 603 €
Delta %	+10,9 %	+16,4 %	Indéfini	Indéfini	Indéfini	À venir	À venir	À venir	À venir	À venir	À venir	À venir	-67,9 %

2024 ▼

2024	Janv.	Fev.	Mars	Avril	Mai	Juin	Juill.	Août	Sept.	Oct.	Nov.	Déc.	Année
CA	39 601 €	20 182 €	43 133 €	54 496 €	52 377 €	55 523 €	57 430 €	71 120 €	59 023 €	46 107 €	47 610 €	76 087 €	622 689 €
Delta n-1	+8 335 €	-21 948 €	-7 201 €	-301 €	+6 419 €	-5 167 €	-9 963 €	+18 935 €	+4 409 €	+151 €	+9 072 €	+18 010 €	+20 751 €
Delta %	+26,7 %	-52,1 %	-14,3 %	-0,5 %	+14,0 %	-8,5 %	-14,8 %	+36,3 %	+8,1 %	+0,3 %	+23,5 %	+31,0 %	+3,4 %

2023 ▼

2023	Janv.	Fev.	Mars	Avril	Mai	Juin	Juill.	Août	Sept.	Oct.	Nov.	Déc.	Année
CA	31 266 €	42 130 €	50 334 €	54 797 €	45 958 €	60 690 €	67 394 €	52 185 €	54 614 €	45 955 €	38 539 €	58 076 €	601 938 €
Delta n-1	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu
Delta %	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu	Inconnu



67 386 €

+14,7 %

Nom	CA	% CA
BS	19 370 €	28,7 %
EG	17 082 €	25,3 %
HI	12 070 €	17,9 %
IR	18 864 €	28,0 %

Page CA (n'incluant pas le header qui est bien présent)

g) La page Synthèse

Cette interface regroupe en un seul écran les principaux indicateurs du magasin et des collaborateurs pour une période sélectionnée, les rôles « collaborator » n'y accèdent pas.



01/01/2025 au 15/02/2025
Ludovic

Synthèse

Mis à jour le 30/04/2025 à 06:06

Dashboard
Devis Optiques
Paniers Moyens & Primes
CA
Synthèse
Importation
Comptes

Chiffres du Magasin

Du 01/01/2025 au 15/02/2025

Mis à jour le 30/04/2025 à 06:06

Catégorie	CA Facturé	Delta CA	Delta CA %	Taux Concret
N	67386 €	+8 628 €	+14,7 %	73,2 %
N-1	58758 €			76,5 %

Statistiques Collaborateurs

Nom	Primes		P2		Devis		CA
	Prime Monture	% Mont. primées	Nombre	Panier Moyen	Nombre	Concrétisation	
BS	55 €	45.8%	61	25 €	51	74,5 %	19370 €
EG	50 €	32.3%	60	17 €	65	70,8 %	17082 €
HI	25 €	29.4%	37	33 €	48	66,7 %	12070 €
IR	45 €	31.0%	70	18 €	49	81,6 %	18864 €
Total	175 €	34.7%	228	22 €	213	73,2 %	67386 €

La page Synthèse

h) La page Comptes

Cette page est dédiée à la gestion des utilisateurs dans l'application. Elle offre plusieurs fonctionnalités :

- Demandes de création, modification ou suppression de comptes : les managers, supermanagers et admins peuvent soumettre une demande via le bouton "Nouvelle demande"
- Visualiser la liste des demandes en attente, avec les informations associées : nom, rôle demandé, initiateur, type de demande (une unique demande est autorisée concernant un utilisateur donné)
- Validation ou refus par le rôle admin exclusivement
- Liste des comptes existants visibles exclusivement par le rôle admin

Les demandes en cours sont enregistrées dans la table **pending_accounts** puis supprimées de celle-ci lors du refus ou de la validation de la demande.

17

Comptes Utilisateurs

 Nouvelle demande

Demande ▾

Ajouter
Modifier
Supprimer

Nom

Prénom

Email

Rôle ▾

Identifiant

Envoyer

COURS ▾

Choix disponibles pour une demande concernant un compte utilisateur

➤ Admin



01/01/2025 au 15/02/2025 ▾
Joy ▾

Comptes

Dashboard
Devis Optiques
Paniers Moyens & Primes
CA
Synthèse
Importation
Comptes

Comptes Utilisateurs

 Nouvelle demande

Demandes en cours ▾

Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
01/05/2025	jhure	Stud	Barina	bstud	collaborator	jhure.dev@gmail.com	suppression	Valider Refuser
01/05/2025	jhure	TEST	Bruno	btest	collaborator	jhure.dev@gmail.com	ajout	Valider Refuser
01/05/2025	jhure	TEST	Aurore	atest	manager	jhure.dev@gmail.com	modification	Valider Refuser

Comptes Utilisateurs ▾

Date Création	Login	Rôle	Nom	Prénom	Email
16/04/2025	jhure	admin	Huré	Joy	jhure.dev@gmail.com
17/04/2025	irau	manager	Rau	Isidore	jhure.dev@gmail.com
17/04/2025	egaligai	collaborator	Galigai	Eden	jhure.dev@gmail.com
17/04/2025	bstud	collaborator	Stud	Barina	jhure.dev@gmail.com
17/04/2025	hiskander	collaborator	Iskander	Henry	jhure.dev@gmail.com
17/04/2025	ldupanloup	supermanager	Dupanloup	Ludovic	jhure.dev@gmail.com
29/04/2025	atest	collaborator	TEST	Aurore	jhure.dev@gmail.com

Page Comptes de l'admin : il peut valider ou refuser une demande et voir les comptes existants

➤ Supermanager et manager

Optitop 01/01/2025 au 15/02/2025 Isidore

Comptes

Mis à jour le 30/04/2025 à 06:06

Comptes Utilisateurs

[+ Nouvelle demande](#)

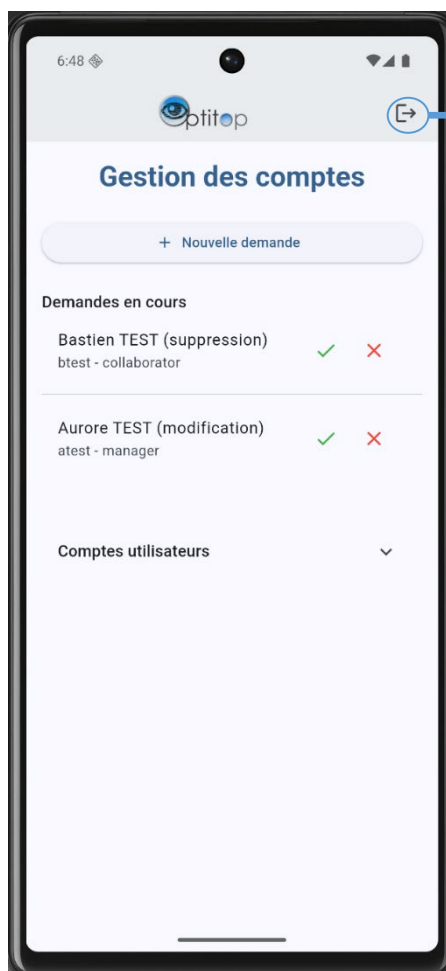
Demands en cours

Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
01/05/2025	jhure	Stud	Barina	bstud	collaborator	jhure.dev@gmail.com	suppression	Valider Refuser
01/05/2025	jhure	TEST	Bruno	btest	collaborator	jhure.dev@gmail.com	ajout	Valider Refuser
01/05/2025	jhure	TEST	Aurore	atest	manager	jhure.dev@gmail.com	modification	Valider Refuser

La page Comptes d'un manager, les boutons « Valider » et « Refuser » sont inactifs

2. IHM de l'application Mobile

a) Connexion



Déconnexion

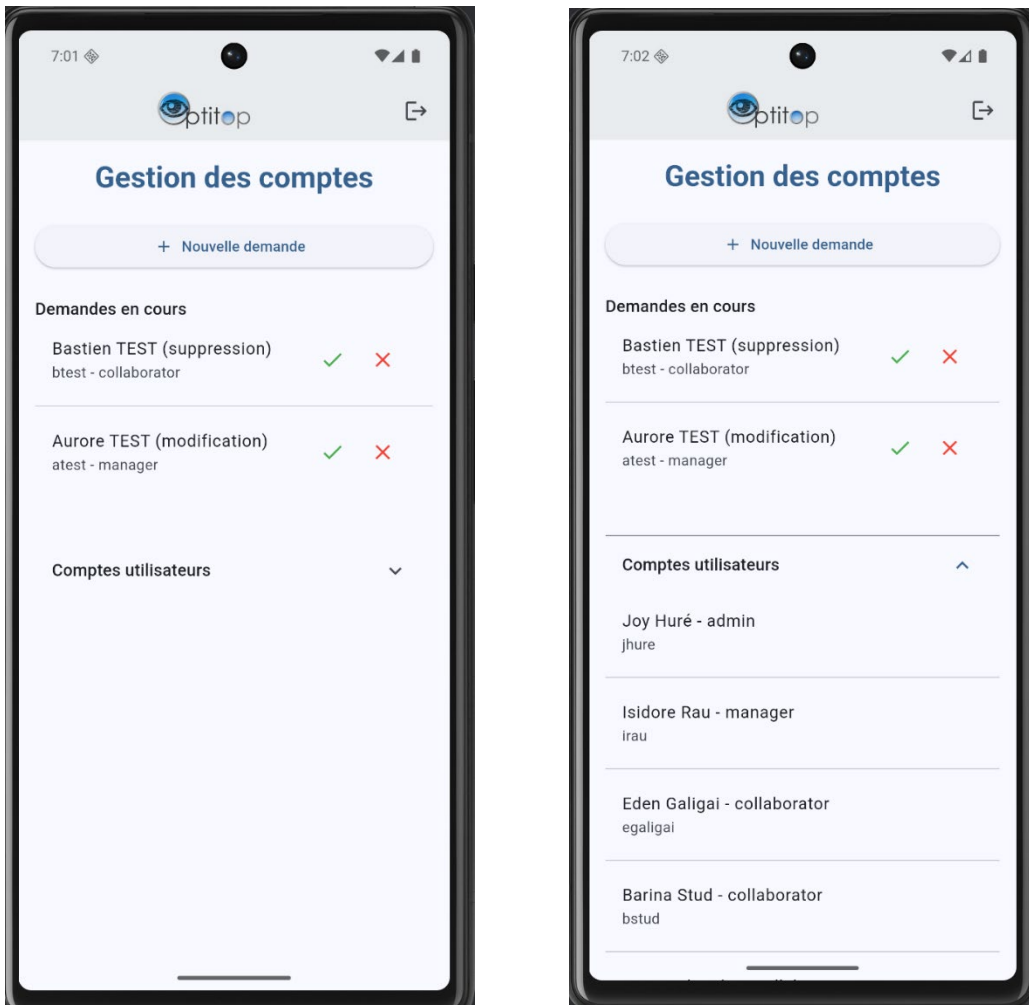
Page de connexion et déconnexion

b) La page Comptes

➤ Vue d'ensemble

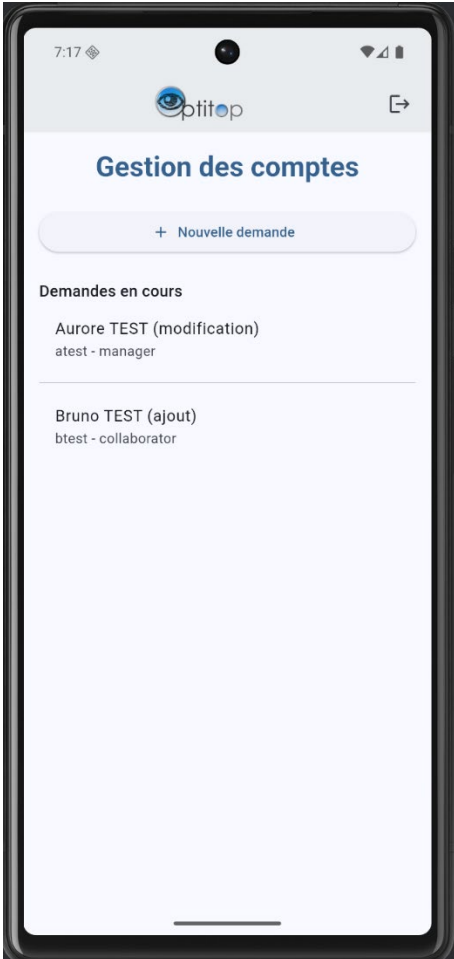
❖ Admin

Les demandes en cours affichent le contenu de la table **pending_accounts** qui regroupe les demandes sur les comptes et « Comptes utilisateurs » affichent les données à partir de la table **user**.



L'admin a accès aux boutons de validation / Refus des demandes et à l'ensemble des comptes utilisateurs

❖ Supermanager



Le supermanager a accès à la consultation des demandes en cours et peut créer de nouvelles demandes. Cependant il ne peut pas valider ou refuser les demandes déjà créées et consulter les comptes utilisateurs.

Vue d'ensemble de la page pour les supermanagers

➤ Nouvelle demande

Lorsqu'une nouvelle demande est créée, elle s'enregistre dans la table **pending_accounts** et apparait dans la liste des demandes en cours. Une pop up informe l'admin de l'arrivée d'une nouvelle demande.

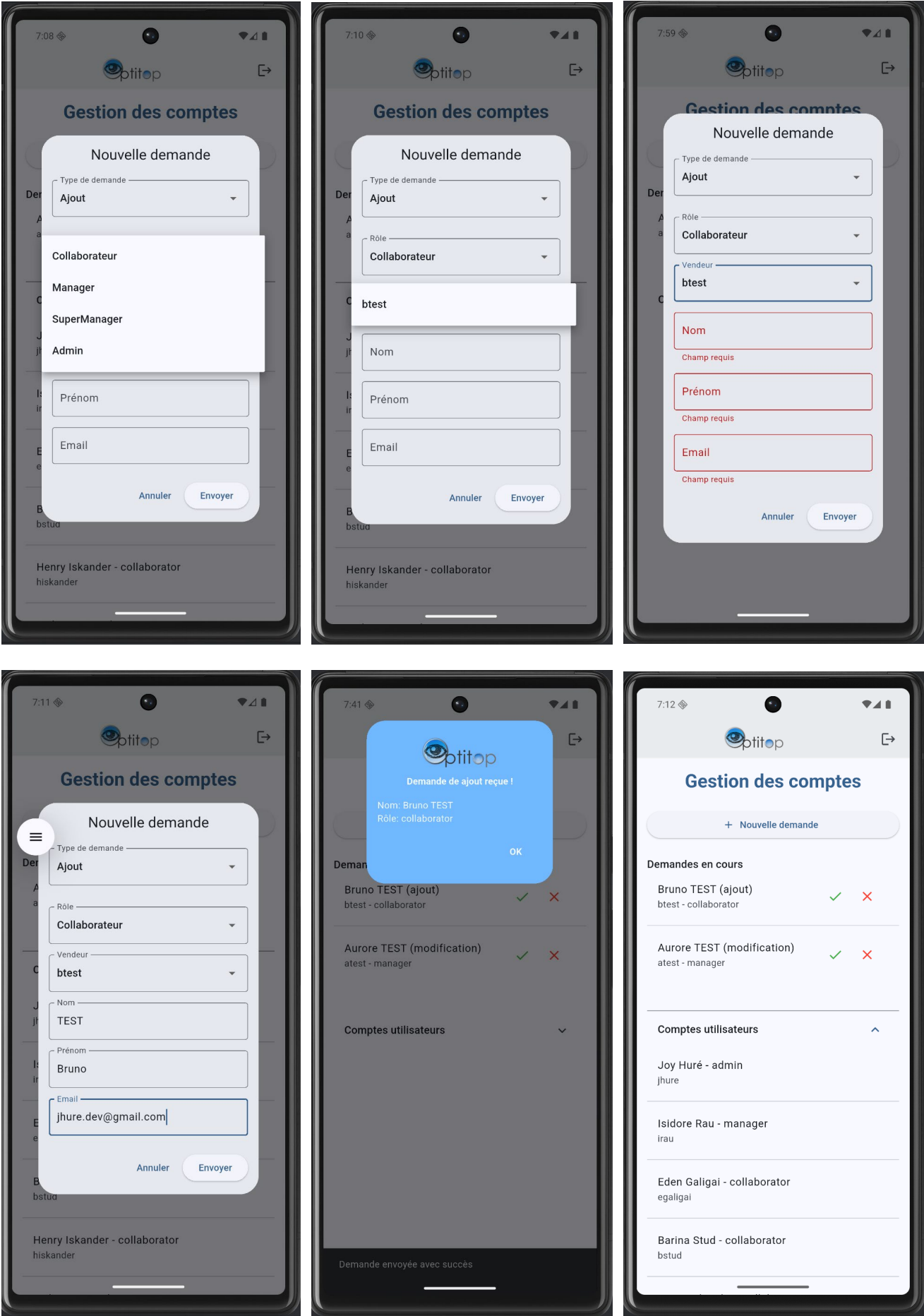
Aucune demande ne peut être créée si une demande concernant le même utilisateur est déjà en cours. La vérification se fait sur le champ **login** de la table **pending_accounts**.

❖ Ajout

Quand la demande est un ajout, la sélection du rôle conditionne les possibilités...

- Si collaborateur ou manager est sélectionné alors le champ vendeur est un choix déroulant qui propose uniquement les identifiants des vendeurs (table **seller**) n'ayant pas de compte utilisateur (table **user**).
- Si supermanager ou admin est sélectionné pour le rôle, alors le champ vendeur est libre de saisie puisque ces rôles ne sont pas nécessairement des vendeurs (donc présents dans la table seller suite à l'import des fichiers .csv, source des données de vente).

Tous les champs doivent être renseignés pour pouvoir envoyer la demande.



L'ensemble d'actions pour l'ajout d'un utilisateur

❖ Modification

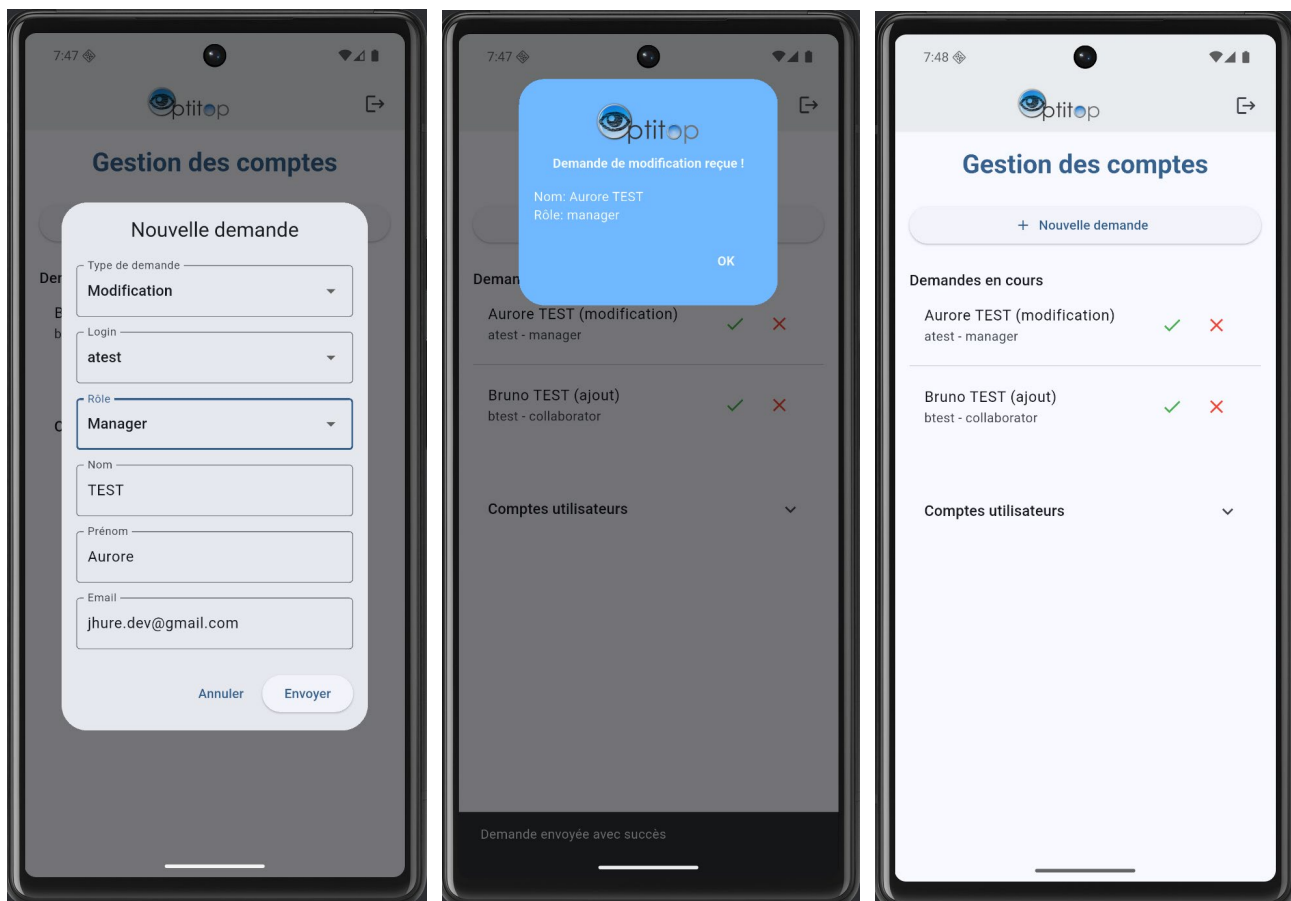
Quand la demande est la modification d'un compte existant il faut d'abord sélectionner le login concerné parmi ceux des utilisateurs existants (table **user**). Ensuite le reste du formulaire se dévoile prérempli avec les données actuelles de l'utilisateur. Le bouton « Envoyer » est débloqué seulement si une modification est réalisée sur l'un de ces champs.

The image displays three sequential screenshots of a mobile application interface for account management, titled "Gestion des comptes".

Screenshot 1: Shows the "Nouvelle demande" (New request) modal. The "Type de demande" (Request type) is set to "Modification". The "Login" field is visible. The modal has "Annuler" (Cancel) and "Envoyer" (Send) buttons.

Screenshot 2: Shows the "Nouvelle demande" modal with the "Type de demande" set to "Modification". A list of logins is displayed below the modal, including "jhure", "Irau", "egaligai", "bstud", "hiskander", "Idupanoloup", and "atest".

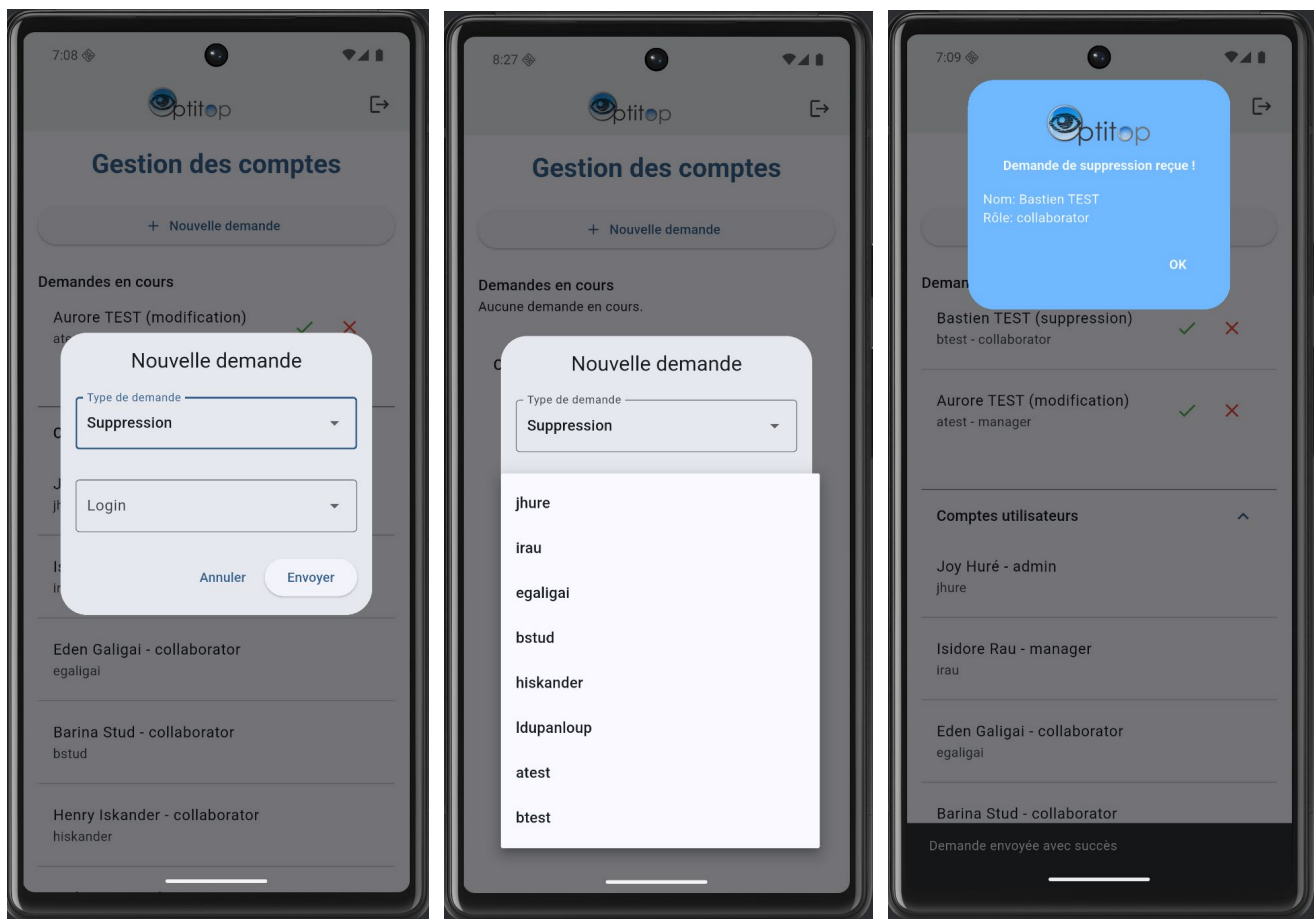
Screenshot 3: Shows the "Nouvelle demande" modal with the "Type de demande" set to "Modification". The "Login" field is filled with "atest". The "Rôle" (Role) is set to "Collaborateur". The "Nom" (Name) is "TEST", the "Prénom" (First name) is "Aurore", and the "Email" is "jhure.dev@gmail.com". The modal has "Annuler" (Cancel) and "Envoyer" (Send) buttons.



L'ensemble d'actions pour la modification d'un utilisateur existant

❖ Suppression

Quand la demande est la suppression d'un compte existant, les logins des tous les comptes des utilisateurs sont proposés dans un choix déroulant (issus de la table **user** et sans les logins ayant déjà une demande en cours).



L'ensemble d'actions pour la suppression d'un utilisateur existant

C. Réalisation des modélisations de données

1. Diagrammes de Cas d'utilisation

a) Application Web

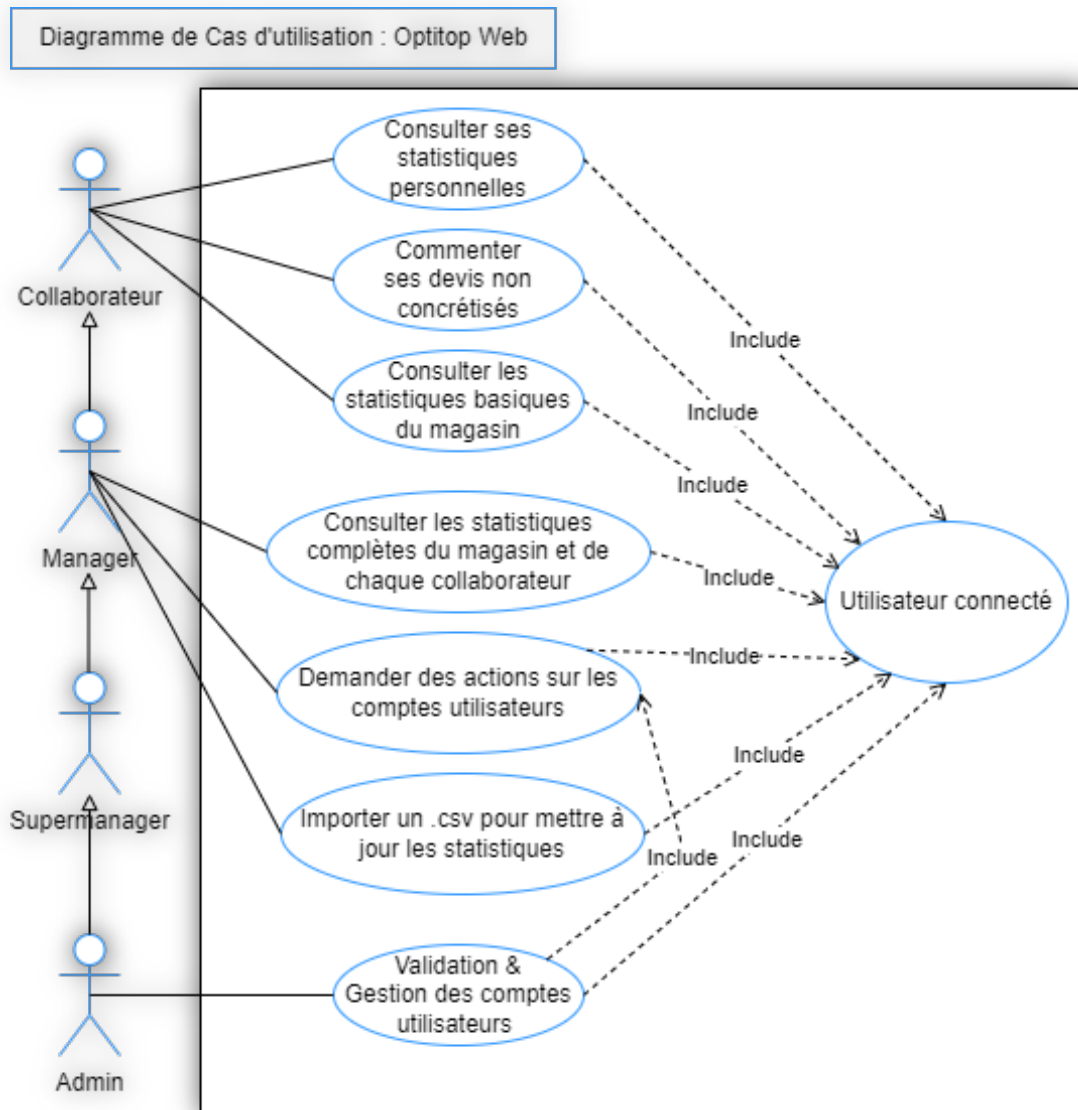


Diagramme de cas d'utilisation : Application web

b) Application Mobile

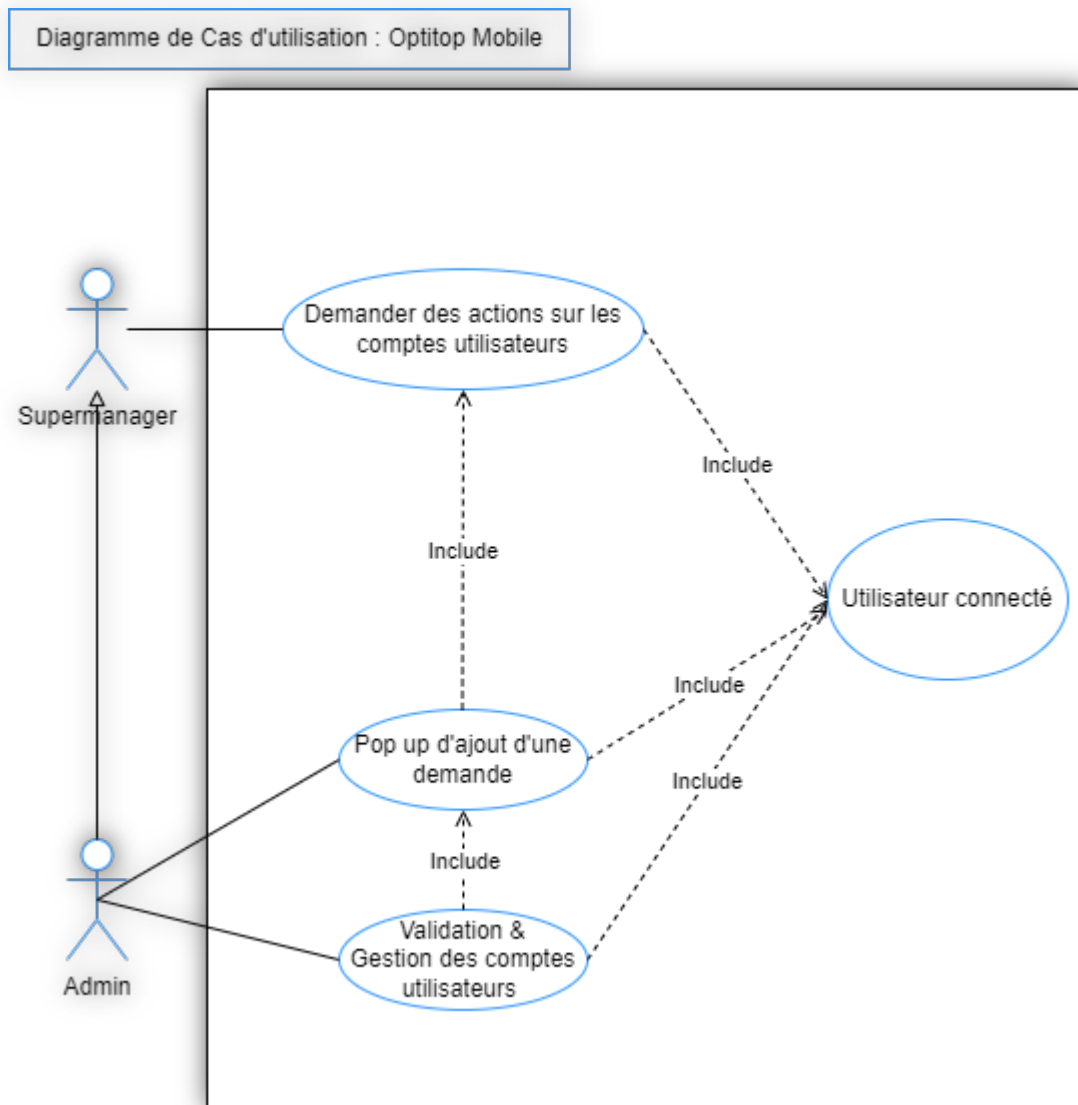
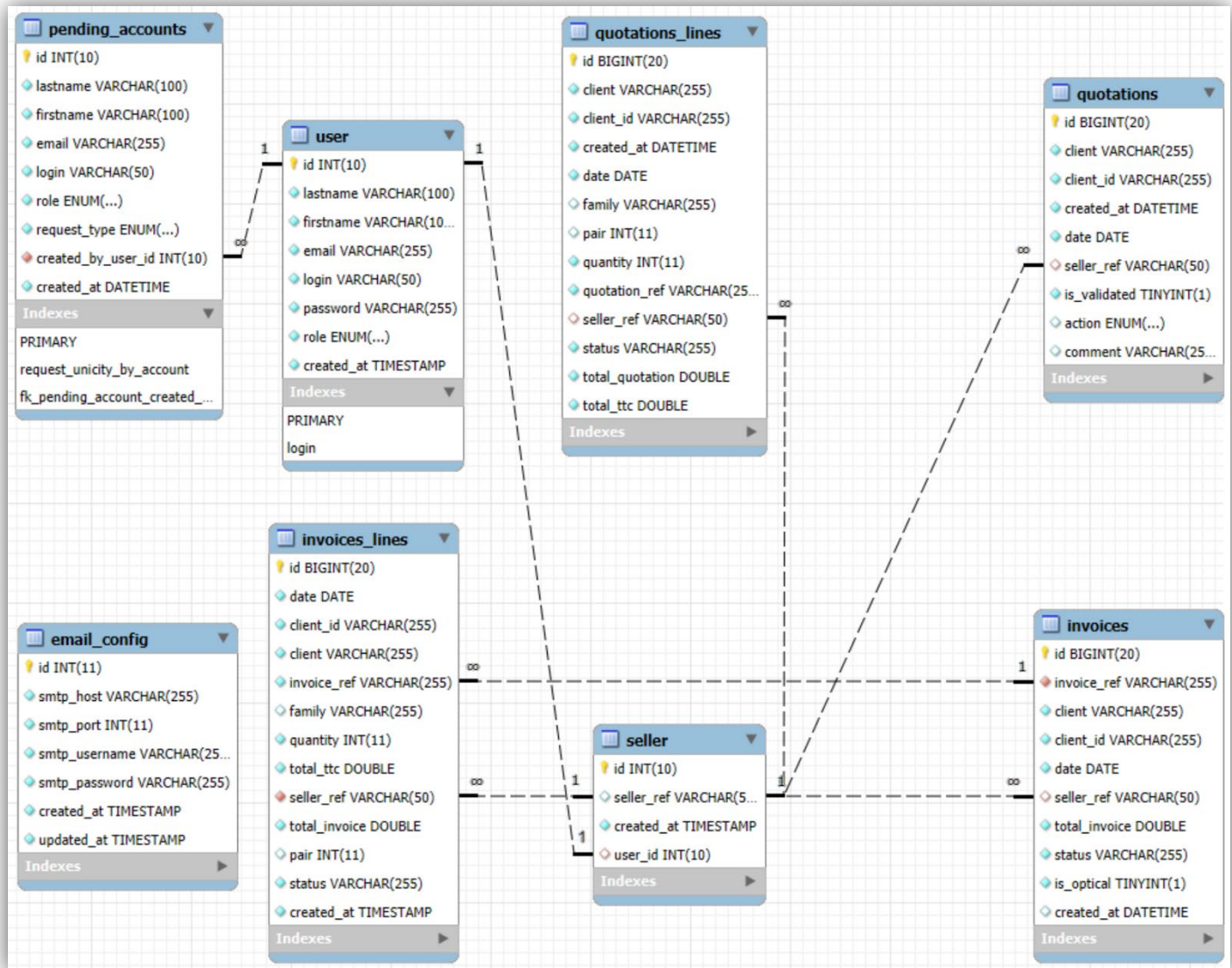


Diagramme de cas d'utilisation : Application mobile

2. Base de données Optitop

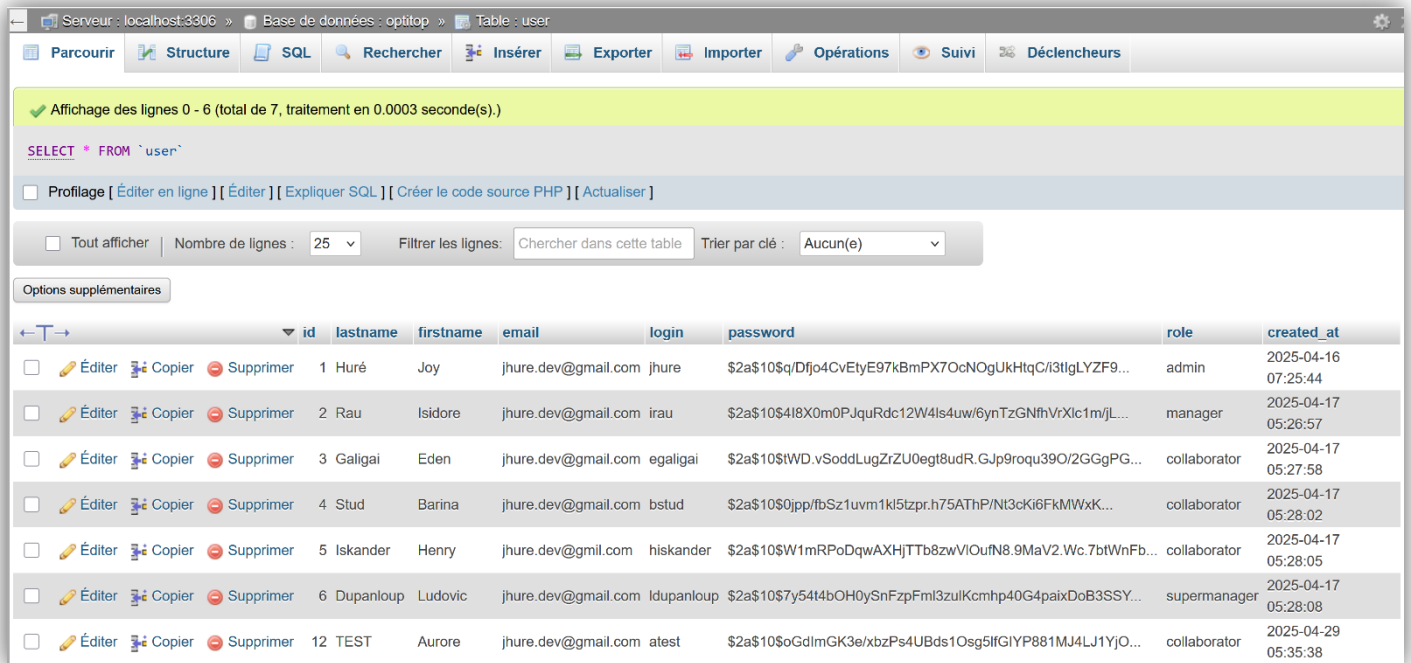
a) Modèle logique de données



Les tables de l'application Optitop

b) Contenu des tables

➤ user



The screenshot shows a database management interface with a toolbar at the top containing buttons like 'Parcourir', 'Structure', 'SQL', 'Rechercher', 'Insérer', 'Exporter', 'Importer', 'Opérations', 'Suivi', and 'Déclencheurs'. Below the toolbar, a green status bar indicates 'Affichage des lignes 0 - 6 (total de 7, traitement en 0.0003 seconde(s))'. The SQL query 'SELECT * FROM `user`' is entered in the query editor. Below the query editor, there are options for 'Profilage', 'Éditer en ligne', 'Éditer', 'Expliquer SQL', 'Créer le code source PHP', and 'Actualiser'. A filter section shows 'Tout afficher', 'Nombre de lignes : 25', 'Filtrer les lignes: Chercher dans cette table', and 'Trier par clé : Aucun(e)'. Below the filter section, there is a button for 'Options supplémentaires'. The table data is displayed in a grid with columns: id, lastname, firstname, email, login, password, role, and created_at. The table contains 7 rows of data, with the last row having an id of 12.

	id	lastname	firstname	email	login	password	role	created_at
☐	1	Huré	Joy	jhure.dev@gmail.com	jhure	\$2a\$10\$q/Dfjo4CvEtyE97kBmPX7OcNOgUkHtqC/f3tIgLYZF9...	admin	2025-04-16 07:25:44
☐	2	Rau	Isidore	jhure.dev@gmail.com	irau	\$2a\$10\$4i8X0m0PJquRdc12W4ls4uw/6ynTzGNfhVrXlc1m/JL...	manager	2025-04-17 05:26:57
☐	3	Galigai	Eden	jhure.dev@gmail.com	egaligai	\$2a\$10\$tWD.vSoddLugZrZU0egt8udR.GJp9roqu39O/ZGGgPG...	collaborator	2025-04-17 05:27:58
☐	4	Stud	Barina	jhure.dev@gmail.com	bstud	\$2a\$10\$0jpp/fbSz1uvm1kl5tzpr.h75AThP/Ni3cKi6FkMWxK...	collaborator	2025-04-17 05:28:02
☐	5	Iskander	Henry	jhure.dev@gmail.com	hiskander	\$2a\$10\$W1mRPoDqwAXHjTTb8zwVlOuFn8.9MaV2.Wc.7btWnFb...	collaborator	2025-04-17 05:28:05
☐	6	Dupanloup	Ludovic	jhure.dev@gmail.com	ldupanloup	\$2a\$10\$7y54t4bOH0ySnFzpFml3zulKcmhp40G4paixDoB3SSY...	supermanager	2025-04-17 05:28:08
☐	12	TEST	Aurore	jhure.dev@gmail.com	atest	\$2a\$10\$0GdlmGK3e/xbzPs4UBds1Osg5IfGIYP881MJ4LJ1YJO...	collaborator	2025-04-29 05:35:38

Table user

➤ seller

Lors de l'import des fichiers .csv, chaque vendeur présent dans le fichier .csv est inséré dans la table seller s'il n'est pas présent.

Serveur : localhost:3306 » Base de données : optitop » Table : seller

Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

Affichage des lignes 0 - 5 (total de 6, traitement en 0.0004 seconde(s).)

SELECT * FROM `seller`

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

				id	seller_ref	created_at	user_id
☐	Éditer	Copier	Supprimer	1	hiskander	2025-04-16 07:28:56	5
☐	Éditer	Copier	Supprimer	2	irau	2025-04-16 07:28:56	2
☐	Éditer	Copier	Supprimer	3	bstud	2025-04-16 07:28:56	4
☐	Éditer	Copier	Supprimer	4	egaligai	2025-04-16 07:28:57	3
☐	Éditer	Copier	Supprimer	5	atest	2025-04-26 11:32:11	12
☐	Éditer	Copier	Supprimer	6	btest	2025-04-26 11:32:11	NULL

Table seller

➤ quotations_lines et invoices_lines

Lors de l'import des fichiers .csv, chaque ligne de facture ou ligne de devis présent est insérée dans les tables **quotations_lines** ou **invoices_lines** selon leur type. Les types « Devis » ou « Devis Validé » s'insèrent dans **quotations_lines** alors que les types « facture » ou « Avoir » s'insèrent dans **invoices_lines**.

Server : localhost:3306 » Base de données : optitop » Table : quotations_lines

Parcourir

Structure

SQL

Rechercher

Insérer

Exporter

Importer

Opérations

Suivi

Déclencheurs

Affichage des lignes 0 - 24 (total de 25747, traitement en 0.0005 seconde(s).)

SELECT * FROM `quotations\_lines`

Profilage

Éditer en ligne

Éditer

Expliquer SQL

Créer le code source PHP

Actualiser

1

>

>>

Nombre de lignes : 25

Filtrer les lignes: Chercher dans cette table

Trier par clé : Aucun(e)

Options supplémentaires

				id	client	client_id	created_at	date	family	pair	quantity	quotation_ref	seller_ref	status	total_quotation	total_ttc
☐	Éditer	Copier	Supprimer	948	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	1	1	D103047701	irau	devis validé	420	90
☐	Éditer	Copier	Supprimer	949	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	MON	1	1	D103047701	irau	devis validé	420	119
☐	Éditer	Copier	Supprimer	950	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	3	1	D103047701	irau	devis validé	420	30
☐	Éditer	Copier	Supprimer	951	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	3	1	D103047701	irau	devis validé	420	30
☐	Éditer	Copier	Supprimer	952	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	2	1	D103047701	irau	devis validé	420	30
☐	Éditer	Copier	Supprimer	953	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	MON	2	1	D103047701	irau	devis validé	420	1
☐	Éditer	Copier	Supprimer	954	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	1	1	D103047701	irau	devis validé	420	90
☐	Éditer	Copier	Supprimer	955	Alarçon Liam	103012307	2025-04-17 05:32:41	2023-01-02	VER	2	1	D103047701	irau	devis validé	420	30
☐	Éditer	Copier	Supprimer	956	Masclaux Ismaël	103002164	2025-04-17 05:32:41	2023-01-02	LENT	NULL	1	D103047702	irau	devis validé	90	45
☐	Éditer	Copier	Supprimer	957	Masclaux Ismaël	103002164	2025-04-17 05:32:41	2023-01-02	LENT	NULL	1	D103047702	irau	devis validé	90	45
☐	Éditer	Copier	Supprimer	958	Alesi Liam	103012230	2025-04-17 05:32:41	2023-01-02	VER	1	1	D103047703	egaligai	devis validé	786.2	238.2
☐	Éditer	Copier	Supprimer	959	Alesi Liam	103012230	2025-04-17 05:32:41	2023-01-02	MON	1	1	D103047703	egaligai	devis validé	786.2	208.8

Extrait de la table quotations_lines

Server: localhost:3306 » Base de données : optitop » Table : invoices_lines

Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

✓ Affichage des lignes 0 - 24 (total de 19830, traitement en 0.0009 seconde(s).)

SELECT * FROM `invoices\_lines`

Profilage [Éditer en ligne] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Options supplémentaires

			id	date	client_id	client	invoice_ref	family	quantity	total_ttc	seller_ref	total_invoice	pair	status	created_at
☐	Éditer	Copier	Supprimer	630	2023-01-02	103007619	Marron Owen	F103056577	CLI	1	29 bstud	29	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	631	2023-01-02	103011989	Jacquín Lino	F103056579	DIV	1	40 irau	40	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	632	2023-01-02	103017425	Mazoiller Albane	F103056582	SOL	1	29 irau	29	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	633	2023-01-03	102018267	Maurelés Capucine	F103056586	MON	1	30 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	634	2023-01-03	102018267	Maurelés Capucine	F103056586	VER	1	90 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	635	2023-01-03	102018267	Maurelés Capucine	F103056586	VER	1	90 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	636	2023-01-03	103016833	Leger Anae	A103001088	ENT	-1	-6 egaligai	-6	NULL	avoir	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	637	2023-01-03	103000004	PASSAGE passage	F103056589	DIV	1	7 egaligai	7	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	638	2023-01-03	103000004	PASSAGE passage	F103056590	ENT	1	6 hispander	30	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	639	2023-01-03	103000004	PASSAGE passage	F103056590	LENT	1	24 bstud	30	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	640	2023-01-03	103017426	Martines Younes	F103056591	DIV	1	7 egaligai	7	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	641	2023-01-04	103017405	Maretti Kylian	F103056593	VER	1	0 irau	0	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	642	2023-01-04	103017431	Charles Maxime	F103056600	SOL	1	29 hispander	29	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	643	2023-01-04	103011266	Texier Aicha	F103056604	SOL	1	131 irau	131	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	644	2023-01-05	103000004	PASSAGE passage	F103056607	DIV	1	5 bstud	5	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	645	2023-01-05	103000004	PASSAGE passage	F103056608	ENT	1	32 irau	32	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	646	2023-01-05	103017435	Mazuret Lino	F103056609	MON	1	30 egaligai	105	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	647	2023-01-05	103017435	Mazuret Lino	F103056609	VER	1	37.5 egaligai	105	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	648	2023-01-05	103017435	Mazuret Lino	F103056609	VER	1	37.5 egaligai	105	1	facture	2025-04-17 05:32:41

Console de requêtes SQL

Extrait de la table invoices_lines

➤ quotations

Après l'insertion des lignes de devis dans **quotations_lines**, une entrée unique est créée dans **quotations** :

- par paire client/date
- Seulement si au moins une de ses lignes est de **family** "VER" (code des verres optiques). Cela m'a permis de filtrer les devis dit « optiques », concernant les lunettes de vue.

Un devis est considéré comme validé si une de ses lignes avec la même paire client / date a le **status** "devis validé".

Si un devis existe déjà pour ce client à cette date, le **status** du devis est mis à jour si nécessaire.

Sinon, la nouvelle entrée est créée dans **quotations**, les champs **action** et **comments** sont créés et initialisés à **NULL**.

Serveur : localhost:3306 » Base de données : optitop » Table : quotations

Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencher

✓ Affichage des lignes 0 - 24 (total de 3660, traitement en 0.0035 seconde(s).) [date: 2023-01-02... - 2023-01-06...]

SELECT * FROM `quotations` ORDER BY `quotations`.`date` ASC

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

				id	client	client_id	created_at	date	1	seller_ref	is_validated	action	comment
☐	Éditer	Copier	Supprimer	370	Alesi Liam	103012230	2025-04-17 05:34:55	2023-01-02		egaligai	1	NULL	NULL
☐	Éditer	Copier	Supprimer	473	Alarçon Liam	103012307	2025-04-17 05:34:57	2023-01-02		irau	1	NULL	NULL
☐	Éditer	Copier	Supprimer	592	Méhaye Soan	103014900	2025-04-17 05:34:59	2023-01-02		hiskander	1	NULL	NULL
☐	Éditer	Copier	Supprimer	835	Mayniel Albane	103017423	2025-04-17 05:35:02	2023-01-02		bstud	0	NULL	NULL
☐	Éditer	Copier	Supprimer	562	Alquin Sara	102030887	2025-04-17 05:34:58	2023-01-03		irau	1	NULL	NULL
☐	Éditer	Copier	Supprimer	821	Martines Younes	103017426	2025-04-17 05:35:02	2023-01-03		egaligai	1	NULL	NULL
☐	Éditer	Copier	Supprimer	462	Chevalier Maryam	102001314	2025-04-17 05:34:57	2023-01-03		hiskander	1	NULL	NULL
☐	Éditer	Copier	Supprimer	284	Maurelès Capucine	102018267	2025-04-17 05:34:54	2023-01-03		bstud	1	NULL	NULL
☐	Éditer	Copier	Supprimer	508	Mondiner Inaya	103013971	2025-04-17 05:34:57	2023-01-03		hiskander	1	NULL	NULL
☐	Éditer	Copier	Supprimer	606	Royer Elya	103017281	2025-04-17 05:34:59	2023-01-04		bstud	1	NULL	NULL
☐	Éditer	Copier	Supprimer	807	Mauri Marceau	103017434	2025-04-17 05:35:02	2023-01-04		irau	1	NULL	NULL
☐	Éditer	Copier	Supprimer	796	Gauthier Lyana	103017432	2025-04-17 05:35:02	2023-01-04		egaligai	1	NULL	NULL
☐	Éditer	Copier	Supprimer	428	Amiard Haroun	100011437	2025-04-17 05:34:56	2023-01-04		irau	1	NULL	NULL
☐	Éditer	Copier	Supprimer	675	Morrères Lino	103009997	2025-04-17 05:35:00	2023-01-04		bstud	1	NULL	NULL
☐	Éditer	Copier	Supprimer	803	Martinozzi Salome	103017433	2025-04-17 05:35:02	2023-01-04		bstud	0	NULL	NULL
☐	Éditer	Copier	Supprimer	891	Maretti Kylian	103017405	2025-04-17 05:35:03	2023-01-04		irau	1	NULL	NULL
☐	Éditer	Copier	Supprimer	197	Junker Clément	103015567	2025-04-17 05:34:53	2023-01-04		egaligai	0	NULL	NULL
☐	Éditer	Copier	Supprimer	785	Mazuret Lino	103017435	2025-04-17 05:35:02	2023-01-05		egaligai	1	NULL	NULL

Extrait de la table quotations

➤ invoices

Après l'insertion des lignes de factures dans **invoices_lines**, une entrée unique est créée par **invoice_ref(invoices_lines)** dans **invoices**.

Le champ **is_optical** est défini sur **1** si au moins une de ses lignes est de **family "VER"** (code des verres optiques).

Severeur : localhost:3306 » Base de données : optitop » Table : invoices

Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

✓ Affichage des lignes 0 - 24 (total de 6567, traitement en 0.0065 seconde(s).) [date: 2023-01-02... - 2023-01-07...]

SELECT * FROM `invoices` ORDER BY `invoices`.`date` ASC

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Options supplémentaires

				id	invoice_ref	client	client_id	date	1	seller_ref	total_invoice	status	is_optical	created_at
☐	Éditer	Copier	Supprimer	1154	F103056582	Mazoiller Albane	103017425	2023-01-02		irau	29	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1168	F103056577	Marron Owen	103007619	2023-01-02		bstud	29	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1173	F103056579	Jacquín Lino	103011989	2023-01-02		irau	40	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1238	F103056591	Martines Younes	103017426	2023-01-03		egaligai	7	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1235	F103056590	PASSAGE passage	103000004	2023-01-03		hiskander	30	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1625	A103001088	Leger Anae	103016833	2023-01-03		egaligai	-6	avoir	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1129	F103056586	Maurelès Capucine	102018267	2023-01-03		bstud	210	facture	1	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1133	F103056589	PASSAGE passage	103000004	2023-01-03		egaligai	7	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	946	F103056604	Texier Aicha	103011266	2023-01-04		irau	131	facture	0	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	1245	F103056593	Maretti Kylian	103017405	2023-01-04		irau	0	facture	1	2025-04-17 05:34:50
☐	Éditer	Copier	Supprimer	941	F103056600	Charles Maxime	103017431	2023-01-04		hiskander	29	facture	0	2025-04-17 05:34:50

Extrait de la table invoices

➤ pending_accounts

La table **pending_accounts** regroupe les demandes en cours sur les comptes utilisateurs existants (modification, suppression) ou sur les comptes utilisateurs à créer (ajout).

Severeur : localhost:3306 » Base de données : optitop » Table : pending_accounts

Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0.0015 seconde(s).)

SELECT * FROM `pending\_accounts`

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Options supplémentaires

				id	lastname	firstname	email	login	role	request_type	created_by_user_id	created_at
☐	Éditer	Copier	Supprimer	57	TEST	Bruno	jhure.dev@gmail.com	btest	collaborator	suppression	1	2025-05-01 11:19:54
☐	Éditer	Copier	Supprimer	58	TEST	Aurore	jhure.dev@gmail.com	atest	manager	modification	1	2025-05-01 11:20:05

Table pending_accounts, 2 demandes sont en attente de traitement

3. Fichiers .csv

Lors d'un import, les données du fichier .csv sont importées dans la base de données pour alimenter plus tard les statistiques affichées par l'application. Ils ont tous la même structure.

Pour les besoins de l'examen et du développement, afin de garantir la confidentialité et protéger les données sensibles du magasin dans lequel je travaille, les données utilisées sont fictives et générées par un script en python qui modifie les données originales.

Les fichiers d'origine sont issus directement de notre logiciel magasin utilisé pour les ventes, la facturation, la gestion administrative, la caisse etc...

```
test_2023.csv x
Ressources > Fichiers .csv > Tests > test_2023.csv > data
1 Date;C.;Num client;client;N° fact.;F.;Q;TotHT;TotTTC;Taux TVA;RefVendeur;Rm / Mj;Total facture;Paire;Type
2 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;75,00;90,00;20,00;irau;0,00;420,00;1;Devis Validé
3 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;MON;1;99,17;119,00;20,00;irau;0,00;420,00;1;Devis Validé
4 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;25,00;30,00;20,00;irau;-19,00;420,00;3;Devis Validé
5 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;25,00;30,00;20,00;irau;-19,00;420,00;3;Devis Validé
6 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;25,00;30,00;20,00;irau;-19,00;420,00;2;Devis Validé
7 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;MON;1;0,83;1,00;20,00;irau;-118,00;420,00;2;Devis Validé
8 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;75,00;90,00;20,00;irau;0,00;420,00;1;Devis Validé
9 02/01/2023;MIFG;103012307;Alarçon Liam;D103047701;VER;1;25,00;30,00;20,00;irau;-19,00;420,00;2;Devis Validé
10 02/01/2023;MIFG;103002164;Masclaux Ismaël;D103047702;LENT;1;37,50;45,00;20,00;irau;-1,00;90,00;;Devis Validé
11 02/01/2023;MIFG;103002164;Masclaux Ismaël;D103047702;LENT;1;37,50;45,00;20,00;irau;-1,00;90,00;;Devis Validé
12 02/01/2023;MIFG;103007619;Marron Owen;F103056577;CLI;1;24,17;29,00;20,00;bstud;0,00;29,00;;facture
13 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;198,50;238,20;20,00;egaligai;-4,00;786,20;1;Devis Validé
14 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;MON;1;174,00;208,80;20,00;egaligai;-52,20;786,20;1;Devis Validé
15 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;0,00;0,00;20,00;egaligai;-49,00;786,20;3;Devis Validé
16 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;0,00;0,00;20,00;egaligai;-49,00;786,20;3;Devis Validé
17 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;MON;1;0,00;0,00;20,00;egaligai;-119,00;786,20;3;Devis Validé
18 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;41,67;50,00;20,00;egaligai;-90,00;786,20;2;Devis Validé
19 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;MON;1;0,83;1,00;20,00;egaligai;-118,00;786,20;2;Devis Validé
20 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;41,67;50,00;20,00;egaligai;-90,00;786,20;2;Devis Validé
21 02/01/2023;MIFG;103012230;Alesi Liam;D103047703;VER;1;198,50;238,20;20,00;egaligai;-4,00;786,20;1;Devis Validé
22 02/01/2023;MIFG;103011989;Jacquin Lino;F103056579;DIV;1;33,33;40,00;20,00;irau;0,00;40,00;;facture
23 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;VER;1;97,50;117,00;20,00;bstud;0,00;354,00;1;Devis
24 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;MON;1;99,17;119,00;20,00;bstud;0,00;354,00;1;Devis
25 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;VER;1;0,00;0,00;20,00;bstud;-49,00;354,00;2;Devis
26 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;VER;1;0,00;0,00;20,00;bstud;-49,00;354,00;2;Devis
27 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;MON;1;0,83;1,00;20,00;bstud;-118,00;354,00;2;Devis
28 02/01/2023;MIFG;103017423;Mayniel Albane;D103047704;VER;1;97,50;117,00;20,00;bstud;0,00;354,00;1;Devis
29 02/01/2023;MIFG;103017424;Maas Kenza;D103047707;DIV;1;0,00;0,00;20,00;hiskander;0,00;0,00;;Devis
30 02/01/2023;MIFG;103017425;Mazoiller Albane;F103056582;SOL;1;24,17;29,00;20,00;irau;0,00;29,00;;facture
31 02/01/2023;MIFG;103014900;Méhayé Soan;D103047708;VER;1;120,83;145,00;20,00;hiskander;0,00;592,00;1;Devis
32 02/01/2023;MIFG;103014900;Méhayé Soan;D103047708;VER;1;120,83;145,00;20,00;hiskander;0,00;592,00;1;Devis
33 02/01/2023;MIFG;103014900;Méhayé Soan;D103047708;SOL;1;251,67;302,00;20,00;hiskander;0,00;592,00;1;Devis
34 02/01/2023;MIFG;103014900;Méhayé Soan;D103047709;VER;1;105,83;127,00;20,00;hiskander;0,00;412,00;1;Devis
```

Extrait d'un fichier .csv importable dans Optitop avec les données de vente (fictives) d'une période

D. Développement de l'application mobile

1. Mise en place de l'environnement Flutter

a) Installation de Flutter et configuration de VS Code

Pour le développement mobile, j'ai choisi d'utiliser le framework **Flutter** avec le langage **dart** car il permet une compatibilité avec mon backend existant en Spring Boot (via API REST) et un fonctionnement multiplateforme (Android/iOS). Le développement a été réalisé sous **Visual Studio Code**.

Pour installer les outils, je me suis conformée à la documentation de flutter.

Install the Flutter SDK

To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.

Use VS Code to install

Download and install

Use VS Code to install Flutter

To install Flutter using these instructions, verify that you have installed [Visual Studio Code](#) 1.86 or later and the [Flutter extension for VS Code](#).

Prompt VS Code to install Flutter

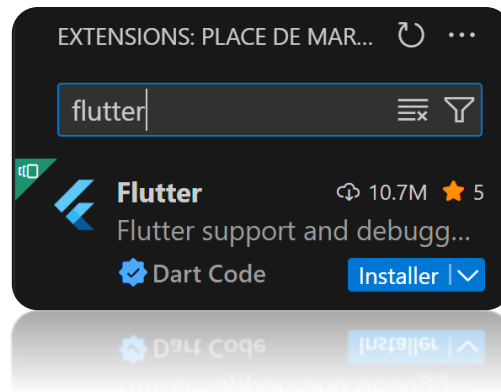
1. Launch VS Code.
2. To open the **Command Palette**, press **Control** + **Shift** + **P**.
3. In the **Command Palette**, type `flutter`.
4. Select **Flutter: New Project**.
5. VS Code prompts you to locate the Flutter SDK on your computer.
 1. If you have the Flutter SDK installed, click **Locate SDK**.
 2. If you do not have the Flutter SDK installed, click **Download SDK**.

This option sends you the Flutter install page if you have not installed Git for Windows as directed in the [development tools prerequisites](#).
6. When prompted **Which Flutter template?**, ignore it. Press **Esc**. You can create a test project after checking your development setup.

Documentation de flutter utilisée, dans mon cas j'ai utilisé VS Code

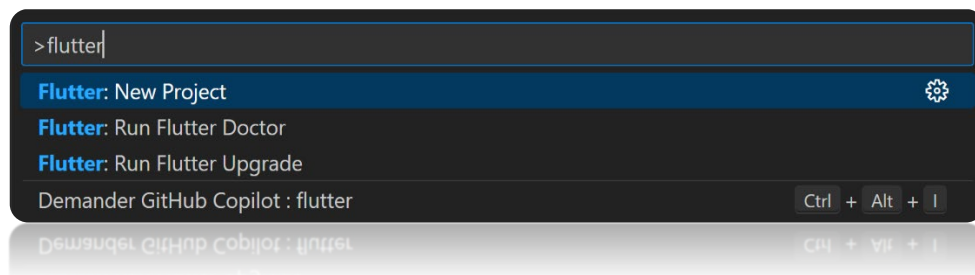
Les étapes de configuration ont été les suivantes :

- Installation de Flutter SDK via l'extension VS Code Flutter/Dart



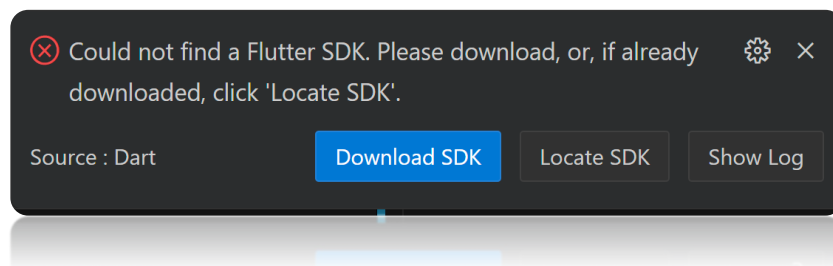
Installation de l'extension VS Code Flutter

- Utilisation de la palette de commandes **Flutter: New Project**



Flutter : New Project

- Téléchargement automatique de Flutter SDK proposé par l'extension



L'extension propose de télécharger Flutter SDK

- Choix du dossier d'installation

b) Installation Android Studio et SDK Android

L'installation du SDK Android est une étape obligatoire pour développer une application mobile avec Flutter sur Android. Même si le code est rédigé sous Visual Studio Code, **Android Studio** est nécessaire pour fournir tous les outils.

Configure Android development

Configure the Android toolchain in Android Studio

[? Help](#)

To create Android apps with Flutter, verify that the following Android components have been installed.

- **Android SDK Platform, API 35.0.2**
- **Android SDK Command-line Tools**
- **Android SDK Build-Tools**
- **Android SDK Platform-Tools**
- **Android Emulator**

If you haven't installed these, or you don't know, continue with the following procedure.

Otherwise, you can skip to the [next section](#).

First time using Android Studio

Current Android Studio User

1. Launch **Android Studio**.

The **Welcome to Android Studio** dialog displays.

2. Follow the **Android Studio Setup Wizard**.

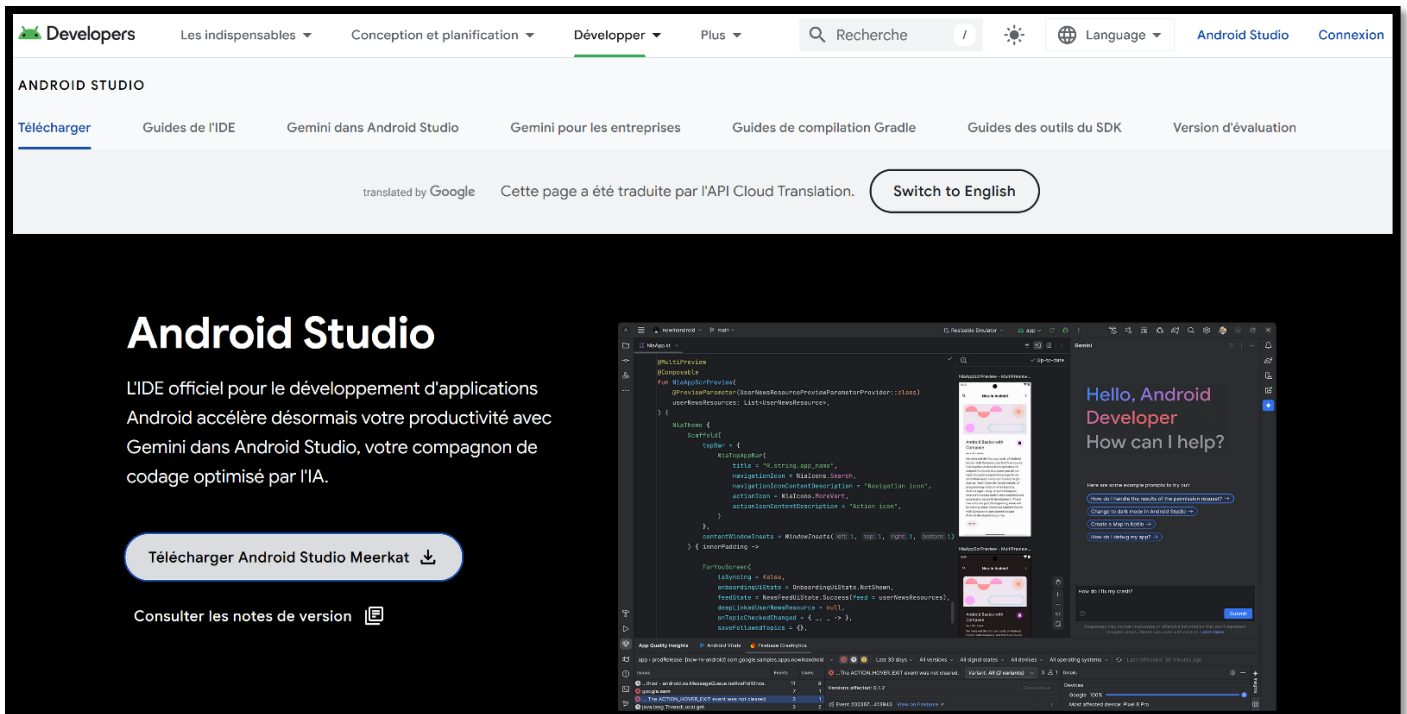
3. Install the following components:

- **Android SDK Platform, API 35.0.2**
- **Android SDK Command-line Tools**
- **Android SDK Build-Tools**
- **Android SDK Platform-Tools**
- **Android Emulator**

Documentation de flutter utilisée pour la configuration d'Android Studio

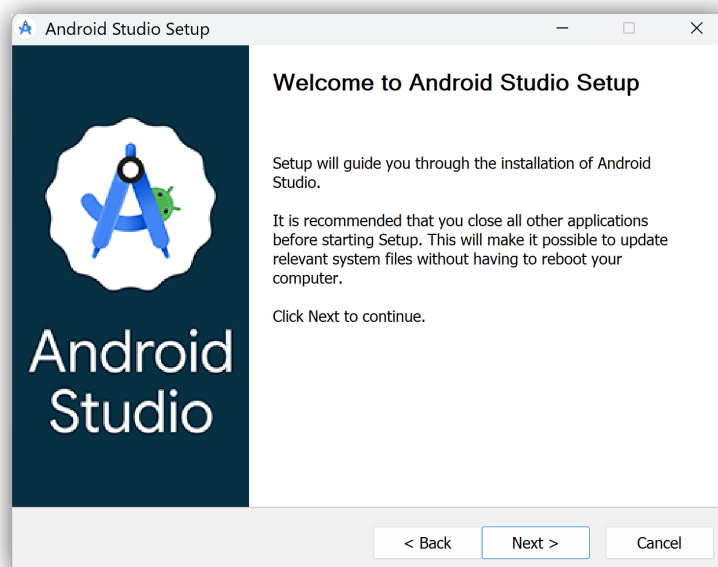
Étapes :

- Téléchargement d'Android Studio depuis le site officiel

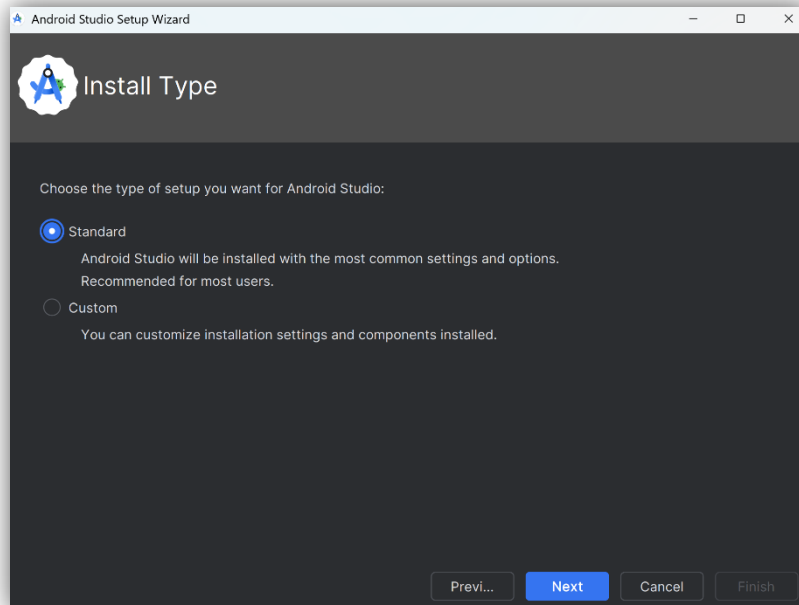


Site officiel d'Android Studio pour le téléchargement

- Installation pas à pas via l'assistant de configuration

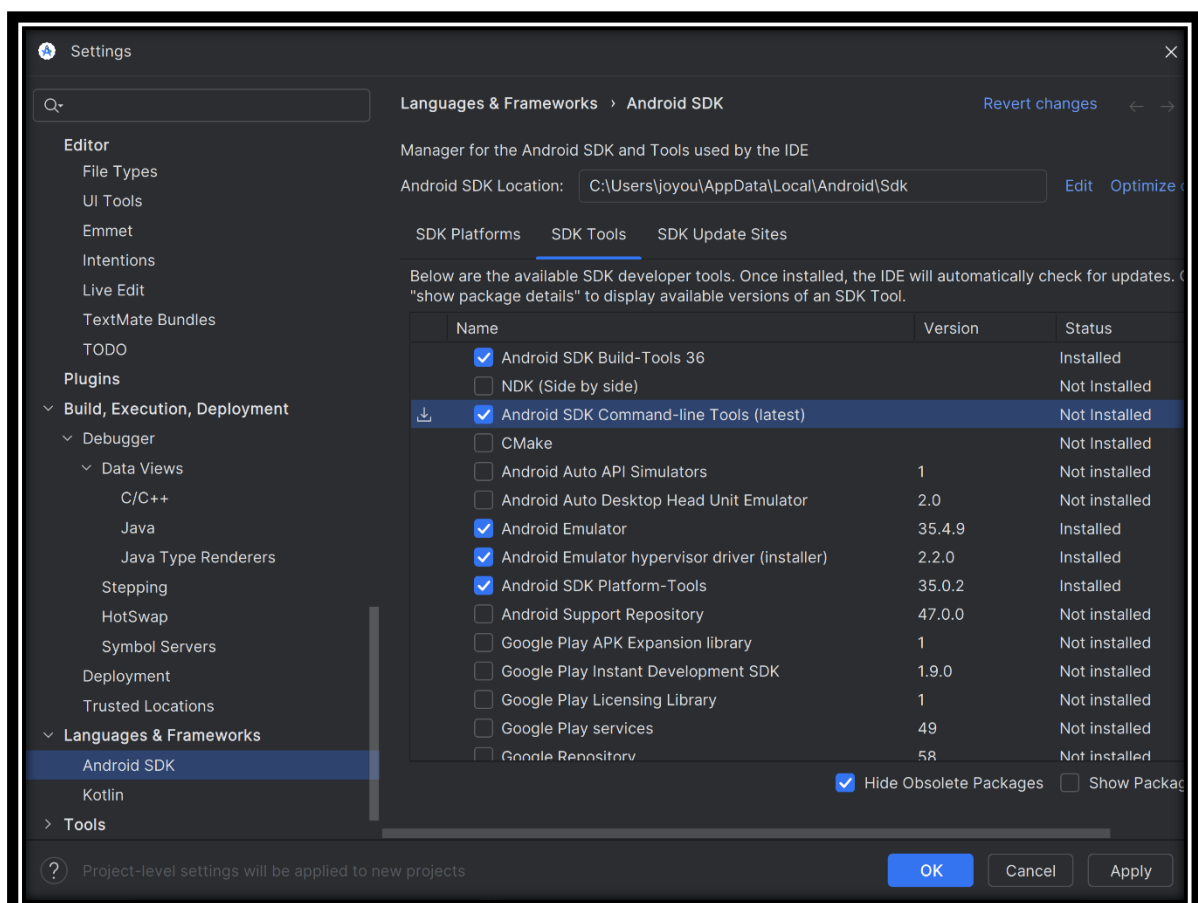


Écran de bienvenue



Sélection du mode "Standard" recommandé

- Configuration du SDK Android en choisissant les composants SDK Platform, Build Tools, Command Line Tools, Platform Tools et Android Emulator

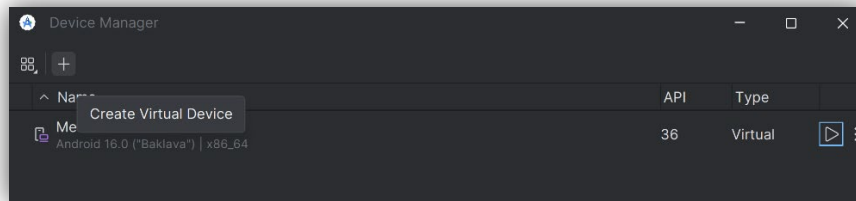


Sélection des SDK Tools à installer

c) Création d'un appareil virtuel Android (émulateur)

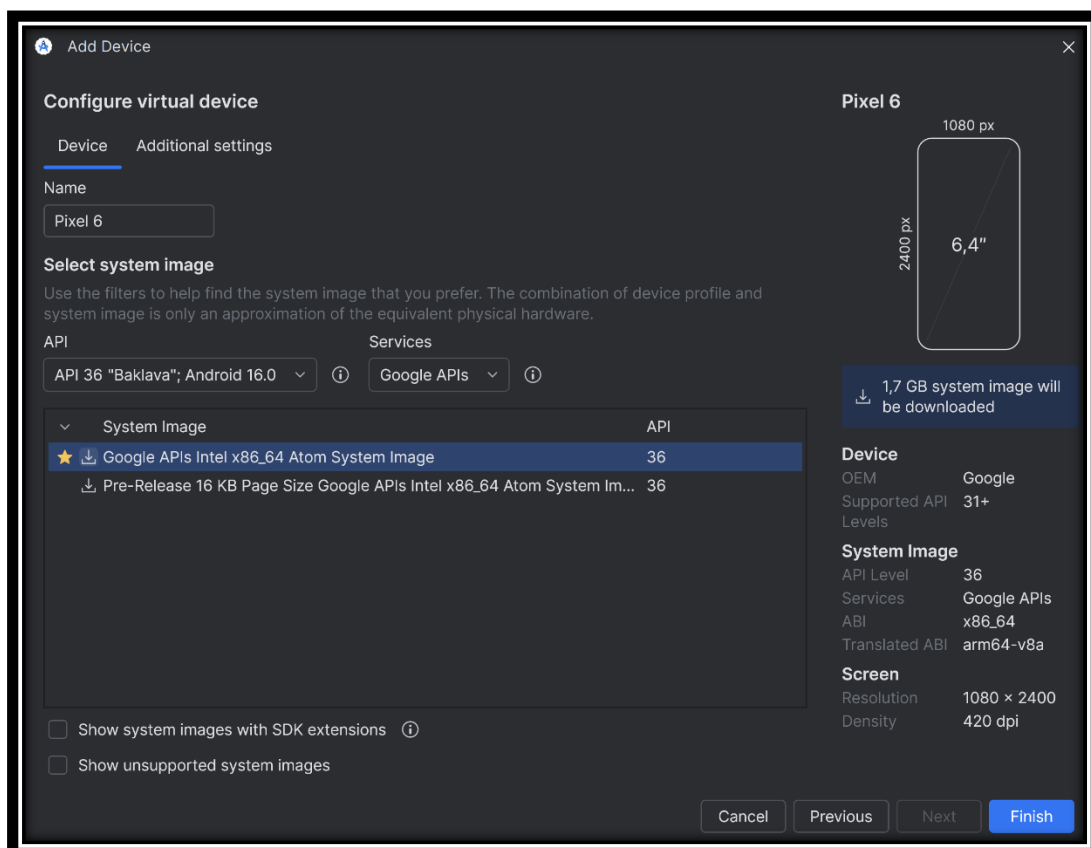
Après avoir configuré le SDK Android, j'ai procédé à la création d'un appareil virtuel Android afin de pouvoir tester mon application mobile directement depuis mon environnement de développement.

Cette opération a été réalisée depuis le Device Manager intégré à Android Studio.



Cliquer sur le « + » permet de créer un nouvel appareil

J'ai sélectionné le modèle **Pixel 6**, un appareil moderne et bien adapté aux tests Flutter, puis j'ai choisi une **image système Android 16.0 (API 36)** avec les **Google APIs** et l'architecture **x86_64**, optimisée pour l'émulation.



Choix de configuration

Une fois configuré, l'appareil a été ajouté à la liste des émulateurs disponibles et peut être lancé à tout moment depuis Android Studio pour simuler un smartphone.

Cet environnement virtuel me permet de visualiser l'application Flutter en temps réel tout en conservant mon flux de développement dans VS Code.

d) Vérification de l'environnement avec Flutter Doctor

Une fois l'ensemble des composants installés (Flutter, Android Studio, SDK Android, émulateur), j'ai exécuté la commande **flutter doctor** depuis le terminal intégré à Visual Studio Code afin de vérifier que tous les outils nécessaires au développement Flutter étaient correctement détectés. Cette commande permet d'analyser l'environnement de développement et de détecter d'éventuels problèmes ou manques.

```
PS C:\Users\joyou\Informatique\Projets\Optitop> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.29.3, on Microsoft Windows [version 10.0.26100.3624], locale fr-FR)
[✓] Windows Version (11 Professionnel 64-bit, 24H2, 2009)
[!] Android toolchain - develop for Android devices (Android SDK version 35.0.1)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2024.3)
[✓] VS Code, 64-bit edition (version 1.99.3)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.
```

Rapport généré par la commande flutter doctor

Le rapport confirme que :

- Flutter est installé et fonctionnel ;
- L'émulateur Android est reconnu ;
- Le SDK Android est présent et opérationnel ;
- Android Studio et VS Code sont bien pris en charge.

Un avertissement mineur concernant Visual Studio (lié au développement d'applications Windows natives) est apparu, mais n'impacte en rien le développement mobile Flutter. L'ensemble est donc prêt à recevoir le projet mobile **optitop_mobile**.



III. Mise en place du backend

A. Base de Données

1. Script de création

```
--
-- Base de données : `optitop`
--
-- -----
-- Structure de la table `email_config`
--
CREATE TABLE `email_config` (
  `id` int(11) NOT NULL,
  `smtp_host` varchar(255) NOT NULL,
  `smtp_port` int(11) NOT NULL,
  `smtp_username` varchar(255) NOT NULL,
  `smtp_password` varchar(255) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Structure de la table `invoices`
--
CREATE TABLE `invoices` (
  `id` bigint(20) NOT NULL,
  `invoice_ref` varchar(255) NOT NULL,
  `client` varchar(255) NOT NULL,
  `client_id` varchar(255) NOT NULL,
  `date` date NOT NULL,
  `seller_ref` varchar(50) DEFAULT NULL,
  `total_invoice` double NOT NULL,
  `status` varchar(255) NOT NULL,
  `is_optical` tinyint(1) NOT NULL DEFAULT 0,
  `created_at` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Structure de la table `invoices_lines`
--
CREATE TABLE `invoices_lines` (
  `id` bigint(20) NOT NULL,
  `date` date NOT NULL,
  `client_id` varchar(255) NOT NULL,
  `client` varchar(255) NOT NULL,
  `invoice_ref` varchar(255) NOT NULL,
  `family` varchar(255) DEFAULT NULL,
  `quantity` int(11) NOT NULL,
  `total_ttc` double NOT NULL,
  `seller_ref` varchar(50) NOT NULL,
  `total_invoice` double NOT NULL,
  `pair` int(11) DEFAULT NULL,
  `status` varchar(255) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
```

Script de création des tables email_config, invoices et invoices_lines

```

-- -----
--
-- Structure de la table `pending_accounts`
--

CREATE TABLE `pending_accounts` (
  `id` int(10) UNSIGNED NOT NULL,
  `lastname` varchar(100) NOT NULL,
  `firstname` varchar(100) NOT NULL,
  `email` varchar(255) NOT NULL,
  `login` varchar(50) NOT NULL,
  `role` enum('admin','collaborator','manager','supermanager') NOT NULL,
  `request_type` enum('ajout','modification','suppression') NOT NULL,
  `created_by_user_id` int(10) UNSIGNED NOT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Structure de la table `quotations`
--

CREATE TABLE `quotations` (
  `id` bigint(20) NOT NULL,
  `client` varchar(255) NOT NULL,
  `client_id` varchar(255) NOT NULL,
  `created_at` datetime NOT NULL,
  `date` date NOT NULL,
  `seller_ref` varchar(50) DEFAULT NULL,
  `is_validated` tinyint(1) NOT NULL DEFAULT 0,
  `action` enum('ATTENTE_MUTUELLE','ATTENTE_RETOUT','A_RELANCER','NON_VALIDE','VOIR_OPTICIEN') DEFAULT
NULL,
  `comment` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Structure de la table `quotations_lines`
--

CREATE TABLE `quotations_lines` (
  `id` bigint(20) NOT NULL,
  `client` varchar(255) NOT NULL,
  `client_id` varchar(255) NOT NULL,
  `created_at` datetime NOT NULL DEFAULT current_timestamp(),
  `date` date NOT NULL,
  `family` varchar(255) DEFAULT NULL,
  `pair` int(11) DEFAULT NULL,
  `quantity` int(11) NOT NULL,
  `quotation_ref` varchar(255) NOT NULL,
  `seller_ref` varchar(50) DEFAULT NULL,
  `status` varchar(255) NOT NULL,
  `total_quotation` double NOT NULL,
  `total_ttc` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Structure de la table `seller`
--

CREATE TABLE `seller` (
  `id` int(10) UNSIGNED NOT NULL,
  `seller_ref` varchar(50) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `user_id` int(10) UNSIGNED DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----

```

```
-- -----  
  
--  
-- Structure de la table `user`  
--  
  
CREATE TABLE `user` (  
  `id` int(10) UNSIGNED NOT NULL,  
  `lastname` varchar(100) NOT NULL,  
  `firstname` varchar(100) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `login` varchar(50) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `role` enum('admin','collaborator','manager','supermanager') NOT NULL,  
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Script de création de la table user

```

--
-- Index pour les tables déchargées
--

--
-- Index pour la table `email_config`
--
ALTER TABLE `email_config`
  ADD PRIMARY KEY (`id`);

--
-- Index pour la table `invoices`
--
ALTER TABLE `invoices`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `uq_invoice_ref` (`invoice_ref`),
  ADD KEY `fk_invoices_seller_ref` (`seller_ref`);

--
-- Index pour la table `invoices_lines`
--
ALTER TABLE `invoices_lines`
  ADD PRIMARY KEY (`id`),
  ADD KEY `fk_invoices_lines_seller` (`seller_ref`),
  ADD KEY `invoices_lines_invoice_ref` (`invoice_ref`);

--
-- Index pour la table `pending_accounts`
--
ALTER TABLE `pending_accounts`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `request_unicity_by_account` (`login`),
  ADD KEY `fk_pending_account_created_by` (`created_by_user_id`);

--
-- Index pour la table `quotations`
--
ALTER TABLE `quotations`
  ADD PRIMARY KEY (`id`),
  ADD KEY `fk_quotations_seller_ref` (`seller_ref`);

--
-- Index pour la table `quotations_lines`
--
ALTER TABLE `quotations_lines`
  ADD PRIMARY KEY (`id`),
  ADD KEY `fk_quotations_lines_seller` (`seller_ref`);

--
-- Index pour la table `seller`
--
ALTER TABLE `seller`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `seller_ref` (`seller_ref`),
  ADD UNIQUE KEY `fk_seller_user` (`user_id`) USING BTREE;

--
-- Index pour la table `user`
--
ALTER TABLE `user`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `login` (`login`);

```



```

--
-- AUTO_INCREMENT pour les tables déchargées
--

--
-- AUTO_INCREMENT pour la table `email_config`
--
ALTER TABLE `email_config`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `invoices`
--
ALTER TABLE `invoices`
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `invoices_lines`
--
ALTER TABLE `invoices_lines`
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `pending_accounts`
--
ALTER TABLE `pending_accounts`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `quotations`
--
ALTER TABLE `quotations`
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `quotations_lines`
--
ALTER TABLE `quotations_lines`
  MODIFY `id` bigint(20) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `seller`
--
ALTER TABLE `seller`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT pour la table `user`
--
ALTER TABLE `user`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT;

```

```

--
-- Contraintes pour les tables déchargées
--

--
-- Contraintes pour la table 'invoices'
--
ALTER TABLE `invoices`
  ADD CONSTRAINT `fk_invoices_invoice_ref` FOREIGN KEY (`invoice_ref`) REFERENCES `invoices_lines`
  (`invoice_ref`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_invoices_seller_ref` FOREIGN KEY (`seller_ref`) REFERENCES `seller` (`seller_ref`) ON
  DELETE CASCADE ON UPDATE CASCADE;

--
-- Contraintes pour la table 'invoices_lines'
--
ALTER TABLE `invoices_lines`
  ADD CONSTRAINT `fk_invoices_lines_seller` FOREIGN KEY (`seller_ref`) REFERENCES `seller` (`seller_ref`);

--
-- Contraintes pour la table 'pending_accounts'
--
ALTER TABLE `pending_accounts`
  ADD CONSTRAINT `fk_pending_account_created_by` FOREIGN KEY (`created_by_user_id`) REFERENCES `user`
  (`id`) ON UPDATE CASCADE;

--
-- Contraintes pour la table 'quotations'
--
ALTER TABLE `quotations`
  ADD CONSTRAINT `fk_quotations_seller_ref` FOREIGN KEY (`seller_ref`) REFERENCES `seller` (`seller_ref`)
  ON DELETE SET NULL ON UPDATE CASCADE;

--
-- Contraintes pour la table 'quotations_lines'
--
ALTER TABLE `quotations_lines`
  ADD CONSTRAINT `fk_quotations_lines_seller` FOREIGN KEY (`seller_ref`) REFERENCES `seller` (`seller_ref`)
  ON DELETE SET NULL ON UPDATE CASCADE;

--
-- Contraintes pour la table 'seller'
--
ALTER TABLE `seller`
  ADD CONSTRAINT `fk_seller_user` FOREIGN KEY (`user_id`) REFERENCES `user` (`id`) ON DELETE SET NULL ON
  UPDATE CASCADE;
COMMIT;

```

Mise en place des contraintes de clés étrangères

2. Insertion de données fictives

```
-- Insérer des données de test dans la table user
INSERT INTO user (lastname, firstname, email, login, password, role)
VALUES
('admin', 'test', 'admin.test@example.com', 'tadmin', 'pass', 'admin'),
('collab', 'test', 'collab.test@example.com', 'tcollab', 'pass', 'collaborator'),
('manager', 'test', 'manager.test@example.com', 'tmanager', 'pass', 'manager'),
('supermanager', 'test', 'supermanager.test@example.com', 'tsupermanager', 'pass', 'supermanager');

-- Insérer des données de test dans la table seller
INSERT INTO seller (seller_ref)
VALUES
('tcollab'),
('tmanager'),
('tsupermanager');

-- Insérer des données de test dans la table pending_accounts
INSERT INTO pending_accounts (lastname, firstname, email, login, role, created_by_user_id)
VALUES
('Taylor', 'Chris', 'chris.taylor@example.com', 'ctaylor', 'collaborator', 5),
('Anderson', 'Pat', 'pat.anderson@example.com', 'panderson', 'manager', 6);

-- Insérer des données de test dans la table invoices
INSERT INTO invoices_lines (date, client_id, client, invoice_ref, family, quantity, total_ttc, seller_ref,
total_invoice, pair, status)
VALUES
('2023-01-01', 103014511, 'SABEEB LUCIE', 'F103063788', 'LENT', 1, 100.00, 'bstud', 200.00, NULL,
'facture'),
('2023-01-02', 103014522, 'ZARZ Kelvin', 'F103063790', 'MON', 1, 235.00, 'irau', 335.00, 1, 'facture'),
('2023-01-02', 103014522, 'ZARZ Kelvin', 'F103063790', 'VER', 1, 50.00, 'irau', 335.00, 1, 'facture'),
('2023-01-02', 103014522, 'ZARZ Kelvin', 'F103063790', 'VER', 1, 50.00, 'irau', 335.00, 1, 'facture'),
('2023-01-02', 103014522, 'ZARZ Kelvin', 'F103063790', 'SOL', 1, 0.00, 'irau', 335.00, 2, 'facture'),
('2023-01-02', 103014522, 'ZARZ Kelvin', 'A103058184', 'MON', -1, -235.00, 'irau', 335.00, 1, 'avoir'),
('2023-01-03', 103014527, 'MASSOT Diane', 'F103063793', 'DIV', 1, 30.00, 'hiskander', 30.00, NULL,
'facture'),
('2023-01-04', 103014527, 'BINET Alexandre', 'F103063794', 'CLI', 1, 29.00, 'egaligai', 29.00, NULL,
'facture');

-- Insérer des données de test dans la table quotations_lines
INSERT INTO quotations_lines (date, client_id, client, quotation_ref, family, quantity, total_ttc,
seller_ref, total_quotation, pair, status)
VALUES
('2023-01-01', 103019518, 'ADAM David', 'D103053015', 'MON', 1, 120.00, 'bstud', 220, 1, 'validé'),
('2023-01-01', 103019518, 'ADAM David', 'D103053015', 'VER', 1, 50.00, 'bstud', 220.00, 1, 'validé'),
('2023-01-01', 103019518, 'ADAM David', 'D103053015', 'VER', 1, 50.00, 'bstud', 220.00, 1, 'validé'),
('2023-01-03', 103019523, 'MARTINS Chris', 'D103053020', 'SOL', 1, 300.00, 'hiskander', 300.00, NULL, 'non
validé'),
('2023-01-03', 103019523, 'MARTINS Chris', 'D103053021', 'SOL', 1, 29.00, 'hiskander', 29.00, NULL, 'non
validé'),
('2023-01-03', 103019523, 'MARTINS Chris', 'D103053022', 'SOL', 1, 250.00, 'hiskander', 250.00, NULL,
'validé');
```

Données de test insérées dans les tables user, seller, pending_accounts, invoices_lines et quotations_lines

3. Déploiement de MySQL sur une machine virtuelle Ubuntu

J'ai déployé le service de base de données sur une machine virtuelle Ubuntu afin de reproduire un environnement proche de la réalité d'un serveur de production. Ce choix me permet de mieux contrôler la

configuration, de renforcer la sécurité des données et de tester les connexions dans des conditions plus représentatives qu'en développement local.

a) Installation de MySQL

- Avant d'installer MySQL, j'ai mis à jour la distribution ce qui permet de s'assurer que tous les paquets sont à jour, y compris les dépendances nécessaires à MySQL.
- Ensuite, j'ai vérifié si MySQL était déjà installé ce qui n'était pas le cas.
- Puis j'ai installé le serveur MySQL

```
joy@joy-VM-UBUNTU01:~$ history
1  sudo apt update && sudo apt upgrade -y && sudo apt dist-upgrade
2  mysql --version
3  sudo apt update && sudo apt install mysql-server -y
```

Commandes Linux utilisées pour l'installation de MySQL

b) Sécurisation de MySQL

Après l'installation de MySQL, j'ai exécuté l'outil **mysql_secure_installation** afin de renforcer la sécurité de l'installation et de finaliser la configuration du service.. Cet assistant interactif permet de réaliser plusieurs actions essentielles :

- Supprimer les composants inutiles comme la base de test
- Appliquer les bonnes pratiques de sécurité pour un usage en environnement de production local

Cette étape m'a permis d'avoir une configuration claire, propre, et adaptée à l'utilisation future de la base avec mon application Spring Boot. Elle garantit que seules les connexions nécessaires sont autorisées et que le serveur MySQL est prêt pour une utilisation stable et sécurisée dans le cadre du projet.


```

joy@joy-VM-UBUNTU01:~$ sudo mysql_secure_installation
[sudo] Mot de passe de joy :

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!

```

Exécution de l'assistant de sécurisation pour supprimer les utilisateurs anonymes, interdire les connexions root distantes, supprimer la base de test et recharger les privilèges.

c) Vérification du service MySQL

```
362 sudo systemctl status mysql
```

Commande de vérification du service

```

● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-14 15:38:40 CEST; 53min ago
     Process: 9026 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 9034 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 19044)
       Memory: 364.7M (peak: 378.7M)
          CPU: 28.333s
       CGroup: /system.slice/mysql.service
               └─9034 /usr/sbin/mysqld

avril 14 15:38:39 joy-VM-UBUNTU01 systemd[1]: Starting mysql.service - MySQL Community Server...
avril 14 15:38:40 joy-VM-UBUNTU01 systemd[1]: Started mysql.service - MySQL Community Server.
~
~
~
~
~
~
~
~
~
~
lines 1-14/14 (END)

```

Réponse obtenue : le service est actif !

d) Création de la Base de données Optitop

```

joy@joy-VM-UBUNTU01:~$ sudo mysql
[sudo] Mot de passe de joy :
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE optitop
-> ;
Query OK, 1 row affected (0,01 sec)

mysql> exit

```

Connexion à MySQL et création de la BDD optitop

e) Création des tables

➤ Génération du fichier SQL par phpmyadmin sur Windows (environnement de développement)

Exportation des tables depuis la base de données « optitop »

Méthode d'exportation :

☐ Rapide, n'afficher qu'un minimum d'options

☒ Personnalisée, afficher toutes les options possibles

Format :

SQL

Tables :

☒	email_config	☒	☒
☐	invoices	☒	☐
☐	invoices_lines	☒	☐
☐	pending_accounts	☒	☐
☐	quotations	☒	☐
☐	quotations_lines	☒	☐
☐	seller	☒	☐
☐	user	☒	☐

Sortie :

☐ Renommer les bases de données/tables/colonnes exportées

☐ Utiliser l'instruction LOCK TABLES

Préparation du fichier `optitop.sql` pour exporter les tables fonctionnelles de l'environnement de développement (seule la table `email_config` sera exportée avec ses données)

J'ai ensuite copié le fichier `optitop.sql` ainsi obtenu sur la VM Ubuntu à cet emplacement :
`/home/joy/Projets/Optitop/`

➤ Import du fichier SQL

```
joy@joy-VM-UBUNTU01:~$ sudo mysql optitop < /home/joy/Projets/Optitop/optitop.sql
```

Import du fichier SQL

f) Vérification des tables

Pour conclure, j'ai vérifié la présence des tables avec les commandes adéquates. Le résultat confirme que l'ensemble des structures de données nécessaires à l'application a été créé dans la base de données.

```

joy@joy-VM-UBUNTU01:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE optitop;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_optitop |
+-----+
| email_config      |
| invoices          |
| invoices_lines    |
| pending_accounts  |
| quotations         |
| quotations_lines   |
| seller            |
| user              |
+-----+
8 rows in set (0,00 sec)

```

Vérification de la bonne importation des tables de la BDD optitop.

g) Création d'un utilisateur MySQL dédié à l'API

Afin de connecter mon API Spring Boot à la base de données de manière sécurisée, j'ai créé un utilisateur MySQL spécifique nommé **optitop_user**, avec des droits limités à la base **optitop**. Ce choix permet de séparer les accès de développement de ceux en production, en évitant l'utilisation du compte administrateur **root**.

```
CREATE USER 'optitop_user'@'%' IDENTIFIED BY 'OptiTop2025!';  
GRANT ALL PRIVILEGES ON optitop.* TO 'optitop_user'@'%;  
FLUSH PRIVILEGES;
```

Création de l'utilisateur optitop_user avec tous les droits sur la BDD optitop.

h) Installation de phpMyAdmin sur Ubuntu

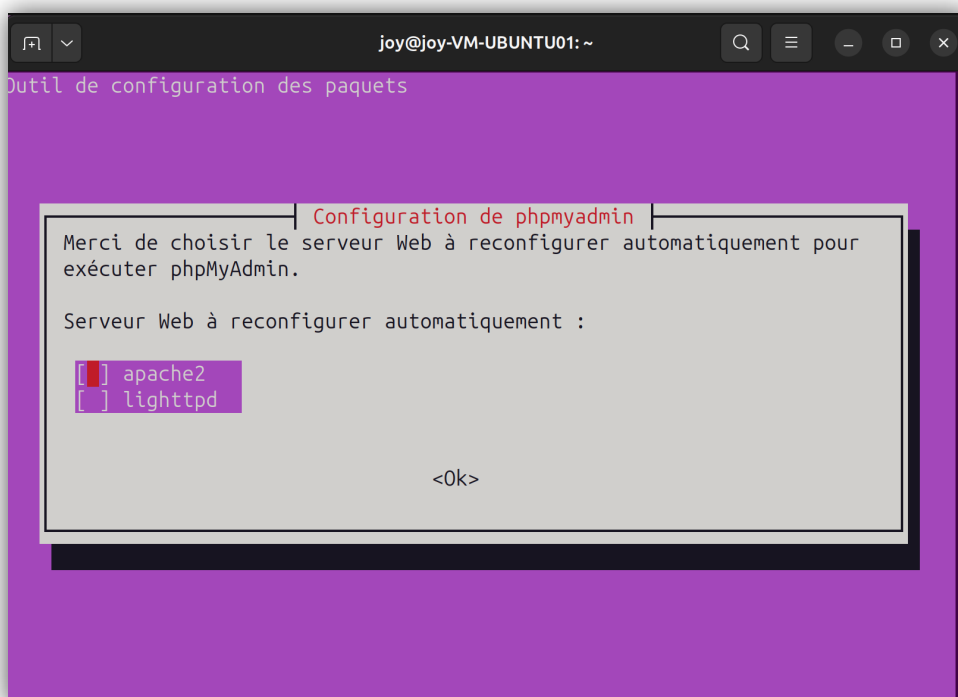
J'ai choisi d'installer **phpMyAdmin** sur ma VM Ubuntu. C'est une interface web légère et intuitive permettant d'administrer MySQL directement depuis un navigateur. Ce choix me permet de manipuler les tables, visualiser les données, exécuter des requêtes SQL et suivre l'évolution des statistiques sans devoir passer systématiquement par la ligne de commande.

Cette solution offre un bon équilibre entre accessibilité, sécurité (accès local uniquement), et rapidité d'exécution.

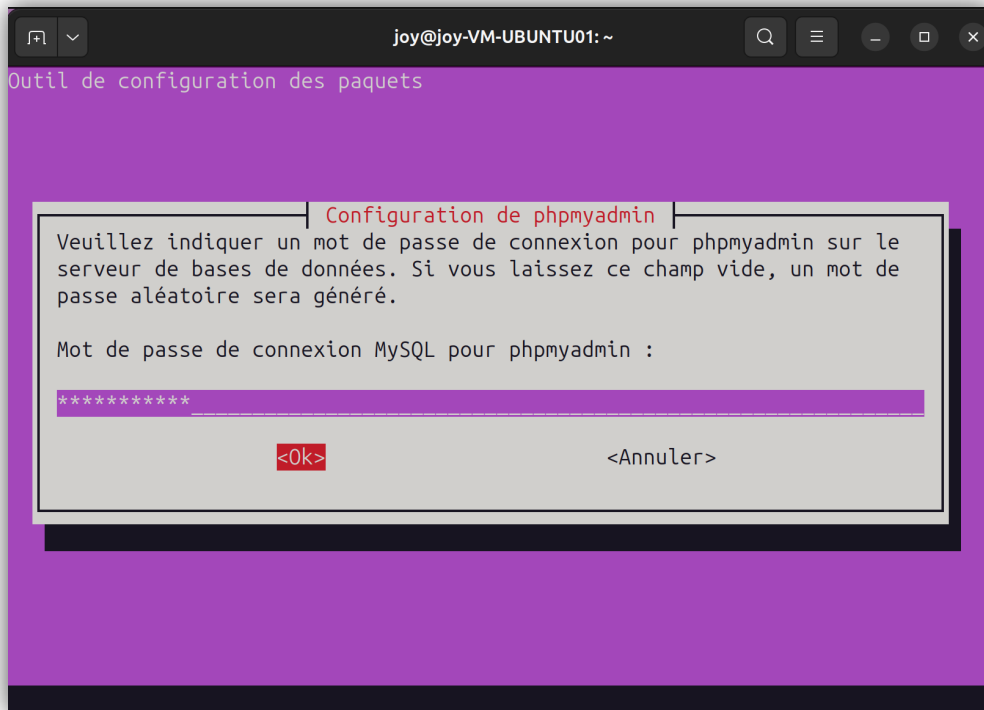
➤ Installation

```
2 sudo apt update && sudo apt upgrade  
3 sudo apt install apache2 php libapache2-mod-php php-mbstring php-zip php-gd php-json php-curl phpmyadmin -y
```

Mise à jour des paquets puis installation d'apache et phpmyadmin en ligne de commandes

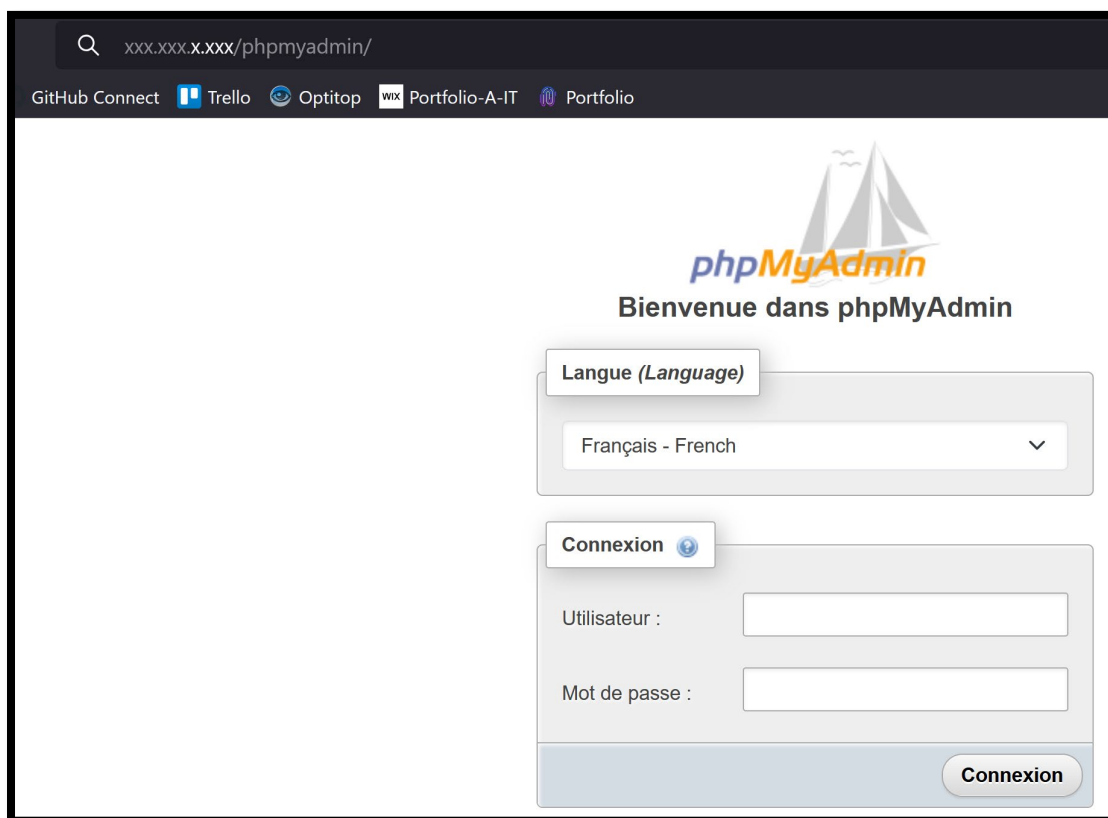






Les étapes de l'installation où différents choix sont proposés

➤ Test de fonctionnement



Connexion à phpmyadmin depuis le navigateur de ma machine physique

Après l'installation de phpMyAdmin, je me suis d'abord connectée avec le compte **phpmyadmin**, ce qui ne permettait pas de visualiser la base **optitop**. En me reconnectant avec l'utilisateur **optitop_user**, dédié à mon application, j'ai pu accéder à toutes les tables de la base. Cela confirme que la gestion des droits fonctionne comme prévu et que mon environnement est correctement configuré.

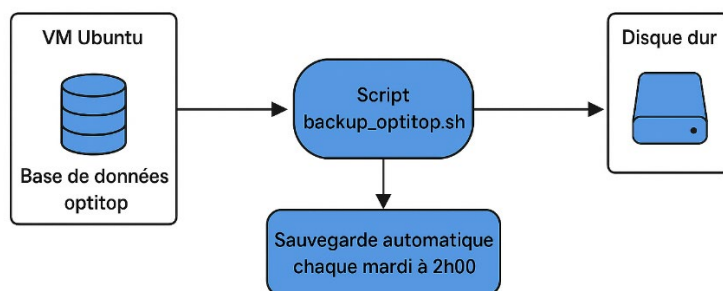
i) Sauvegarde automatique de la base de données

Afin de garantir la pérennité des données du projet Optitop, une solution de sauvegarde automatique de la base de données a été mise en place.

Le besoin initial était de disposer d'une sauvegarde régulière, sans intervention manuelle, pour restaurer rapidement les données en cas de problème système ou de corruption de la base.

Le choix s'est porté sur une sauvegarde complète hebdomadaire de la base **optitop**, réalisée tous les mardis à 2h00 du matin.

Les sauvegardes sont stockées sur un disque dur externe de 5 To connecté à la Freebox du domicile, garantissant un stockage sécurisé et externe au serveur principal.



➤ Création d'un utilisateur MySQL dédié aux sauvegardes

Un utilisateur spécifique, nommé **backup_user**, a été créé dans MySQL. Il dispose uniquement des droits nécessaires pour réaliser un **mysqldump**, à savoir :

- SELECT
- LOCK TABLES
- SHOW VIEW
- EVENT
- TRIGGER
- PROCESS

Cela permet de sécuriser la sauvegarde sans utiliser le compte **root**, en limitant les risques liés à l'utilisation d'un utilisateur ayant trop de privilèges.

```
joy@joy-VM-UBUNTU01:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.41-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'backup_user'@'%' IDENTIFIED BY ' ';
Query OK, 0 rows affected (0,07 sec)

mysql> GRANT SELECT, LOCK TABLES, SHOW VIEW, EVENT, TRIGGER ON optitop.* TO 'backup_user'@'%';
Query OK, 0 rows affected (0,03 sec)

mysql> GRANT PROCESS ON *.* TO 'backup_user'@'%';
Query OK, 0 rows affected (0,02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,01 sec)

mysql> exit
Bye
```

Création de backup_user

➤ Montage d'un disque réseau pour stocker les sauvegardes

Un disque dur externe est connecté à la Freebox du réseau local et monté dans la VM Ubuntu sous le dossier **/mnt/backup_optitop** grâce au protocole CIFS/Samba ce qui permet de stocker les sauvegardes hors de la machine virtuelle, garantissant leur accessibilité même en cas de défaillance du serveur Ubuntu.

J'ai commencé par récupérer l'adresse IP locale de ma freebox.

```

PS C:\Users\joyou> ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet 2 :

    Suffixe DNS propre à la connexion. . . . : 
    Adresse IPv6 de liaison locale. . . . . : 
    Adresse IPv4. . . . . : 
    Masque de sous-réseau. . . . . : 
    Passerelle par défaut. . . . . : 

Carte inconnue OpenVPN Data Channel Offload for NordVPN :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . . : 

Carte inconnue Connexion au réseau local :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . . : 

Carte réseau sans fil Connexion au réseau local* 1 :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . . : 

Carte Ethernet VMware Network Adapter VMnet1 :

    Suffixe DNS propre à la connexion. . . . : 
    Adresse IPv6 de liaison locale. . . . . : 
    Adresse IPv4. . . . . : 
    Masque de sous-réseau. . . . . : 
    Passerelle par défaut. . . . . : 

Carte Ethernet VMware Network Adapter VMnet8 :

    Suffixe DNS propre à la connexion. . . . : 
    Adresse IPv6 de liaison locale. . . . . : 
    Adresse IPv4. . . . . : 
    Masque de sous-réseau. . . . . : 
    Passerelle par défaut. . . . . : 

Carte réseau sans fil Wi-Fi :

    Suffixe DNS propre à la connexion. . . . : 
    Adresse IPv4. . . . . : 
    Masque de sous-réseau. . . . . : 
    Passerelle par défaut. . . . . : 

```

Passerelle par défaut est l'IP de ma freebox

Depuis l'explorateur de fichiers Windows j'ai accédé à l'adresse réseau en saisissant **\\IP FREEBOX** dans la barre d'adresse. Cette action m'a permis de vérifier que le partage SMB/CIFS était bien actif sur le réseau local.

Le disque dur externe connecté à la Freebox, nommé Disque 1, était visible, confirmant ainsi que le périphérique de stockage était correctement partagé et accessible depuis le réseau.

- Sur ma VM Ubuntu, j'ai installé le paquet **cifs-utils** qui est indispensable pour permettre le montage d'un dossier réseau SMB/CIFS sous Linux.(2)

- J'ai ensuite créé un dossier vide pour accueillir le disque réseau, c'est dans ce dossier que le contenu du disque réseau allait apparaître une fois monté.(3)
- Puis j'ai monté le disque dur réseau en échappant l'espace dans le nom **Disque 1** et en utilisant les identifiants de ma Freebox.(4)
- Pour finir j'ai vérifié l'accès au disque (5)

```
1 sudo apt update
2 sudo apt install cifs-utils
3 sudo mkdir /mnt/backup_optitop
4 sudo mount -t cifs //[redacted]/Disque\ 1 /mnt/backup_optitop -o username=Joy0,password=[redacted],vers=1.0
5 ls /mnt/backup_optitop
6 jz \wuf\p9cknb-obrrfob
```

Commandes utilisées pour le montage du disque réseau

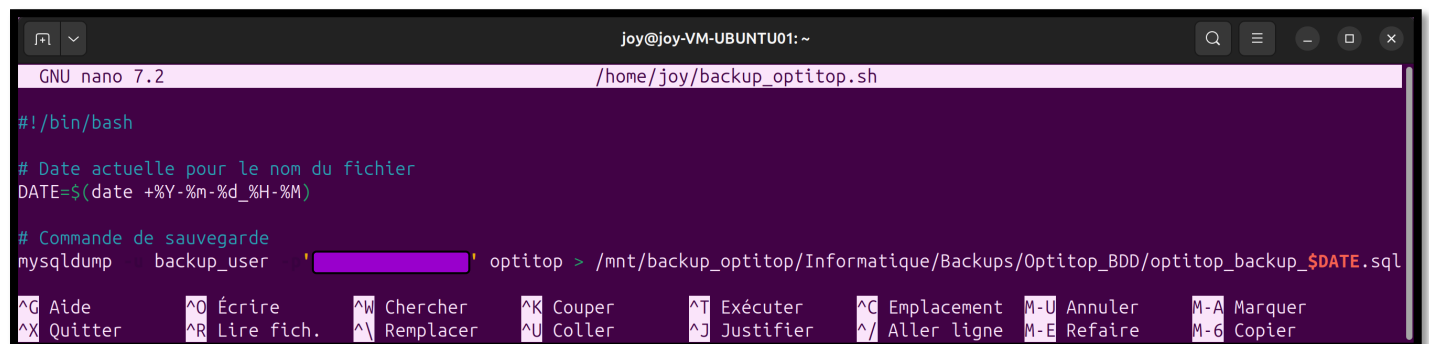
```
joy@joy-VM-UBUNTU01:~$ ls /mnt/backup_optitop
Dépôt  Enregistrements  Informatique  Musiques  Photos  Vidéos
Ebooks  Hypnose          lost+found    Partitions  Téléchargements
```

Résultat concluant : le disque est accessible

➤ Création d'un script de sauvegarde automatique

Un script **backup_optitop.sh** a été développé. Il contient la commande **mysqldump** permettant d'exporter toute la base de données au format SQL, une génération automatique du nom de fichier en fonction de la date et de l'heure (**optitop_backup_YYYY-MM-DD_HH-MM.sql**).

Cela permet d'automatiser complètement le processus sans intervention manuelle, tout en conservant un historique daté des sauvegardes.



```
GNU nano 7.2 /home/joy/backup_optitop.sh

#!/bin/bash

# Date actuelle pour le nom du fichier
DATE=$(date +%Y-%m-%d_%H-%M)

# Commande de sauvegarde
mysqldump backup_user '[redacted]' optitop > /mnt/backup_optitop/Informatique/Backups/Optitop_BDD/optitop_backup_${DATE}.sql
```

Script backup_optitop.sh

```
joy@joy-VM-UBUNTU01:~$ sudo bash /home/joy/backup_optitop.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
joy@joy-VM-UBUNTU01:~$ ls /mnt/backup_optitop/Informatique/Backups/Optitop_BDD
optitop_backup_2025-04-28_16-43.sql
obrrfob-p9cknb-2025-04-28-16-43-2d
```

Lancement du script et résultat : un fichier **optitop_backup_2025-04-28_16-43.sql** vient de s'enregistrer !

➤ Planification de l'exécution automatique

Pour finir, j'ai programmé le script dans la **crontab** système pour qu'il s'exécute automatiquement.

```
GNU nano 7.2 /tmp/crontab.QtGCNV/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 2 * * 2 bash /home/joy/backup_optitop.sh
```

[Lecture de 24 lignes]

^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement
^X Quitter	^R Lire fich.	^_ Remplacer	^U Coller	^J Justifier	^/ Aller ligne

Automatisation de l'exécution du script à 2h du matin tous les mardis

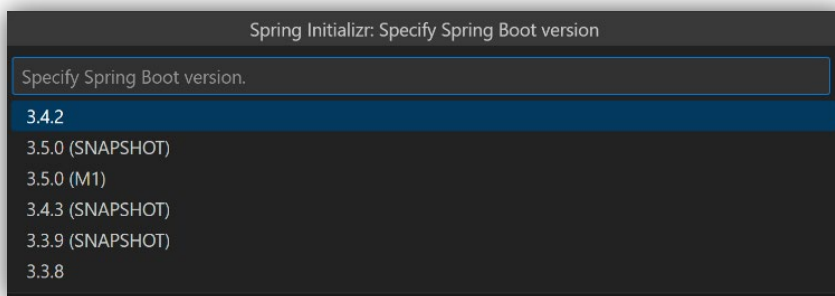
B. API REST Spring Boot

J'ai choisi Spring Boot et une API REST en Java pour mon projet car c'est une solution robuste, performante et adaptée aux applications modernes. Spring Boot permet une configuration simplifiée, une gestion efficace des dépendances et une intégration fluide avec MySQL via JPA/Hibernate. Une API REST en Java garantit une communication rapide et sécurisée entre le backend et les applications (web, mobile, lourde). De plus, Spring Security facilite l'authentification et la gestion des rôles, assurant ainsi la protection des données sensibles.

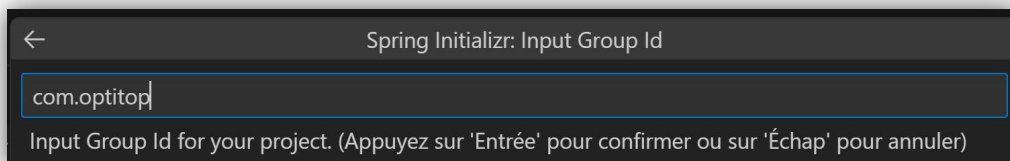
1. L'API dans l'environnement de développement

a) Installation

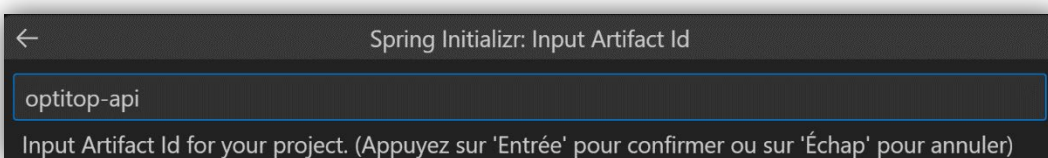
J'ai installé mon API depuis VSCode et l'extension **spring initializr** :



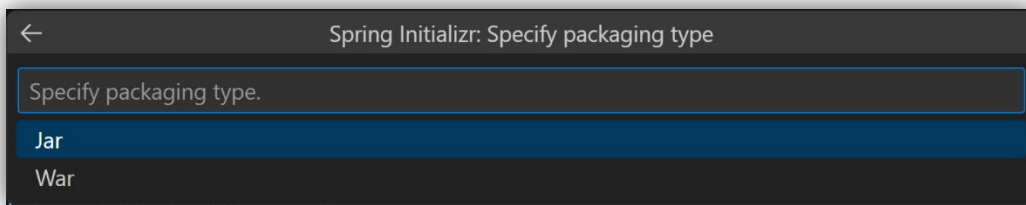
Sélection de la version Spring Boot



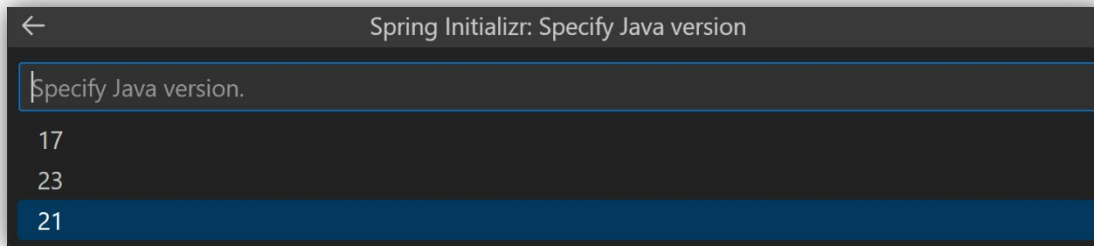
Définition de l'input « Group Id »



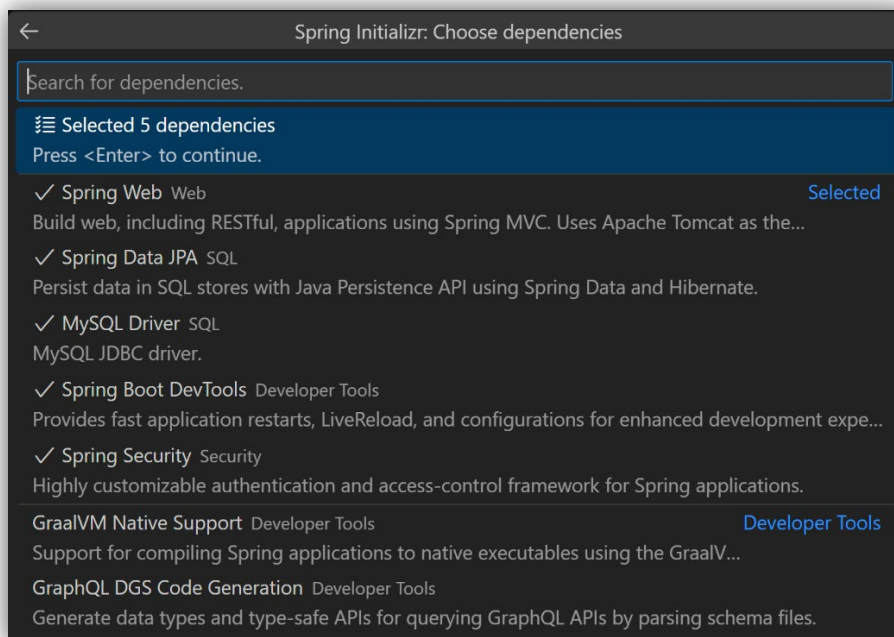
Définition de l'Artifact Id



Format final du fichier généré par Maven



Renseigner la version de Java utilisée : ici 21



Choix des dépendances nécessaires

b) Test du fonctionnement de l'API

Pour contrôler le bon fonctionnement de mon API, j'ai créé un fichier de test affichant « L'API fonctionne correctement » si la requête GET dans Postman aboutit.


```

package com.optitop.optitop_api.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class TestController {

    @GetMapping("/test")
    public String test() {
        return "L'API fonctionne correctement !";
    }
}

```

Fichier TestController.java qui teste le bon fonctionnement de l'API

```

package com.optitop.optitop_api.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .csrf(csrf -> csrf.disable())
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/**").permitAll());
        return http.build();
    }
}

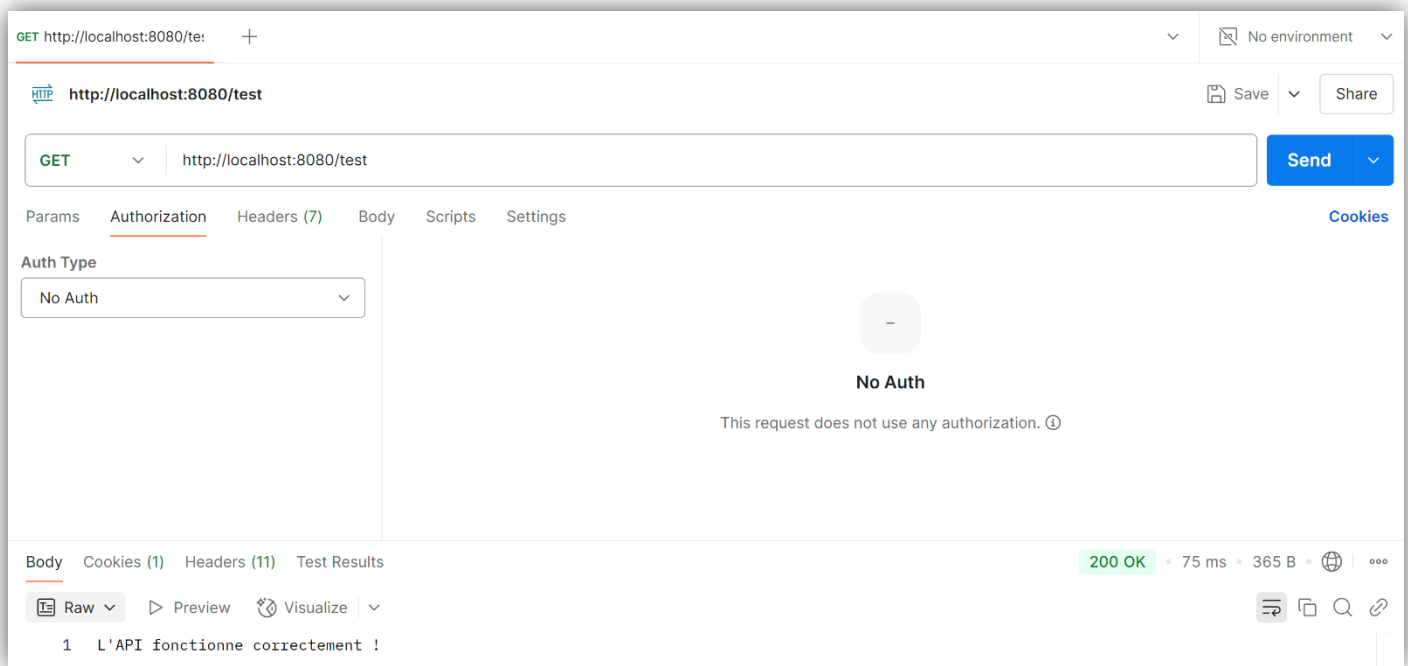
```

Fichier SecurityConfig.java qui désactive l'authentification pour le développement

J'ai ensuite démarré l'API avec cette commande :

mvn spring-boot:run

Ensuite, dans Postman, j'ai lancé ma requête GET après avoir paramétré l'authentification sur « No Auth ».



Requête GET « test » pour confirmer le fonctionnement de l’API

2. Documentation de l'API avec Swagger

Afin de faciliter la visualisation, la compréhension et l'évaluation des routes exposées par mon API REST, j'ai intégré **Swagger UI** via la bibliothèque **Springdoc OpenAPI**.

a) Installation

L'ajout s'est fait en intégrant la dépendance nécessaire dans le fichier **pom.xml**.



```
<!-- SpringDoc OpenAPI UI -->
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-starter-webmvc-ui</artifactId>
  <version>2.3.0</version>
</dependency>
```

Intégration de la dépendance dans le fichier pom.xml

Aucune autre configuration n'a été nécessaire grâce aux conventions automatiques de Springdoc. Une fois l'application lancée, l'interface Swagger est disponible à l'adresse suivante :

<http://localhost:8080/swagger-ui.html>

b) Documentation des contrôleurs

Après l'intégration de Swagger, j'ai procédé à la documentation explicite de chacun des contrôleurs de l'API à l'aide d'annotations spécifiques telles que **@Operation**, **@ApiResponse** ou encore **@Parameter**. Cela permet de fournir une description claire des fonctionnalités, des types de données attendus en entrée, ainsi que des réponses retournées.

Cette étape renforce l'utilité de Swagger pour tout utilisateur de l'API.

```

package com.optitop.optitop_api.controller;

import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import com.optitop.optitop_api.dto.SellerDTO;
import com.optitop.optitop_api.service.SellerService;
import java.util.List;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.tags.Tag;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.responses.ApiResponses;
import io.swagger.v3.oas.annotations.media.Content;
import io.swagger.v3.oas.annotations.media.Schema;

@RestController
@RequestMapping("/api/sellers")
@CrossOrigin(origins = "http://localhost")
@Tag(name = "Vendeurs", description = "Gestion des vendeurs disponibles pour création de compte")
public class SellerController {

    private final SellerService sellerService;

    public SellerController(SellerService sellerService) {
        this.sellerService = sellerService;
    }

    @Operation(summary = "Récupérer les vendeurs disponibles pour la création de compte", description = "Retourne la liste des vendeurs qui n'ont pas encore de compte utilisateur "
        + "et qui n'ont pas de demande en cours")
    @ApiResponses(value = {
        @ApiResponse(responseCode = "200", description = "Vendeurs trouvés avec succès", content =
            @Content(mediaType = "application/json", schema = @Schema(implementation = SellerDTO.class))),
        @ApiResponse(responseCode = "204", description = "Aucun vendeur disponible pour création de compte", content = @Content),
        @ApiResponse(responseCode = "500", description = "Erreur interne du serveur", content =
            @Content())
    })
    @GetMapping("/available-sellers")
    public ResponseEntity<List<SellerDTO>> getAvailableSellers() {
        List<SellerDTO> sellers = sellerService.findAvailableSellers();
        return sellers.isEmpty()
            ? ResponseEntity.noContent().build()
            : ResponseEntity.ok(sellers);
    }
}

```

Contrôleur SellerController.java documenté pour le swagger

c) Routes de l'API

 Swagger
Supported by SMARTBEAR

/api-docs

Explore

API Optitop

1.0.0 OAS 3.0
[/api-docs](#)
API pour la gestion des données d'Optitop

Servers

http://localhost:8080 - Generated server url

Factures (InvoiceController)

Gestion des statistiques de vente liées aux factures

GET

/api/invoices/years

Années d'activités

▼

GET

/api/invoices/total-stats

Obtenir les statistiques globales du magasin

▼

GET

/api/invoices/seller-stats

CA par vendeur et pourcentage du CA total

▼

GET

/api/invoices/period-revenue

Chiffre d'affaires sur période et sa comparaison n-1

▼

GET

/api/invoices/monthly-revenue/{year}

Chiffre d'affaires mensuel par année

▼

GET

/api/invoices/frame-stats

Statistiques des montures primées

▼

GET

/api/invoices/average-baskets

Obtenir les paniers moyens par vendeur

▼

Date de dernière mise à jour (UpdateController)

Gestion de la date de dernière MàJ des données

GET

/api/updates/last

Dernière mise à jour

▼

Utilisateurs (UserController)

Gestion des utilisateurs et de leurs informations

GET

/api/users/{id}/login

Récupérer le login

▼

GET

/api/users/{id}/lastname

Récupérer le nom de famille

▼

GET

/api/users/{id}/email

Récupérer l'email

▼

GET

/api/users/{id}/created-at

Récupérer la date de création du compte utilisateur

▼

GET

/api/users/logins

Liste des logins disponibles à création de demande sur le compte

▼

GET

/api/users/login/{login}

Récupérer un utilisateur par login

▼

GET

/api/users/all

Liste des utilisateurs

▼

POST

/api/users/{id}/change-password

Changer le mot de passe

▼

Demandes sur les comptes (PendingAccountController)

Gestion des demandes de création, modification et suppression de comptes

GET

/api/pending-accounts

Lister toutes les demandes

▼

POST

/api/pending-accounts

Créer une demande de compte

🔒 ▼

POST

/api/pending-accounts/validate/{id}

Valider une demande

🔒 ▼

POST

/api/pending-accounts/reject/{id}

Rejeter une demande

🔒 ▼

Vendeurs (SellerController)

Gestion des vendeurs disponibles pour création de compte

GET

/api/sellers/available-sellers

Récupérer les vendeurs disponibles pour la création de compte

▼

70

Devis (QuotationController) Gestion des devis et de leurs statistiques



GET /api/quotations/unvalidated Récupérer les devis non validés



GET /api/quotations/stats Statistiques des devis



GET /api/quotations/previous-concretization Taux de concrétisation N-1



GET /api/quotations/actions Récupérer les actions possibles



PUT /api/quotations/batch-update Mettre à jour un lot de devis



Import des données de ventes (SalesController) Import des données de ventes (lignes de factures et de devis) depuis un fichier CSV



POST /api/sales/import Importer un fichier de ventes



Authentification (AuthController) Gestion de l'authentification des utilisateurs (connexion/déconnexion)



POST /api/auth/logout Déconnexion utilisateur



POST /api/auth/login Connexion utilisateur



L'ensemble des routes de l'API Optitop

C. Définition des différents environnements

Afin de professionnaliser le cycle de développement de l'application Optitop, j'ai mis en place deux environnements bien distincts :

- Un environnement de développement local (**dev**), me permettant de travailler sur l'interface et l'API en local, avec une base de données locale.
- Un environnement de préproduction (**pprod**), simulant un contexte plus réaliste dans lequel le frontend et l'API sont exécutés localement mais interagissent avec une base de données distante, hébergée sur une machine virtuelle Ubuntu.

Cette séparation permet de tester l'application avec des jeux de données différents, sans impacter mon environnement de travail, et surtout sans devoir recompiler l'application à chaque changement d'environnement.

1. Mise en place des profils Spring Boot

Pour gérer ces deux environnements, j'ai utilisé la fonctionnalité de profils de Spring Boot. Chaque environnement possède son propre fichier de configuration :

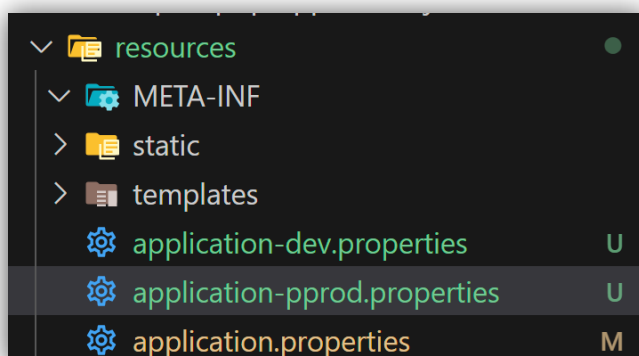
- **application-dev.properties** pour le développement
- **application-pprod.properties** pour la préproduction

Ces fichiers contiennent les paramètres spécifiques à leur environnement : identifiants de base de données, adresse du serveur, niveau de journalisation, etc.

Par ailleurs, j'ai conservé le fichier principal **application.properties**, qui contient les paramètres communs à tous les environnements :

- configuration de JPA
- gestion des logs
- formatage des requêtes SQL
- fuseau horaire, etc.

Cela permet de centraliser les règles générales tout en personnalisant les comportements selon le profil actif.



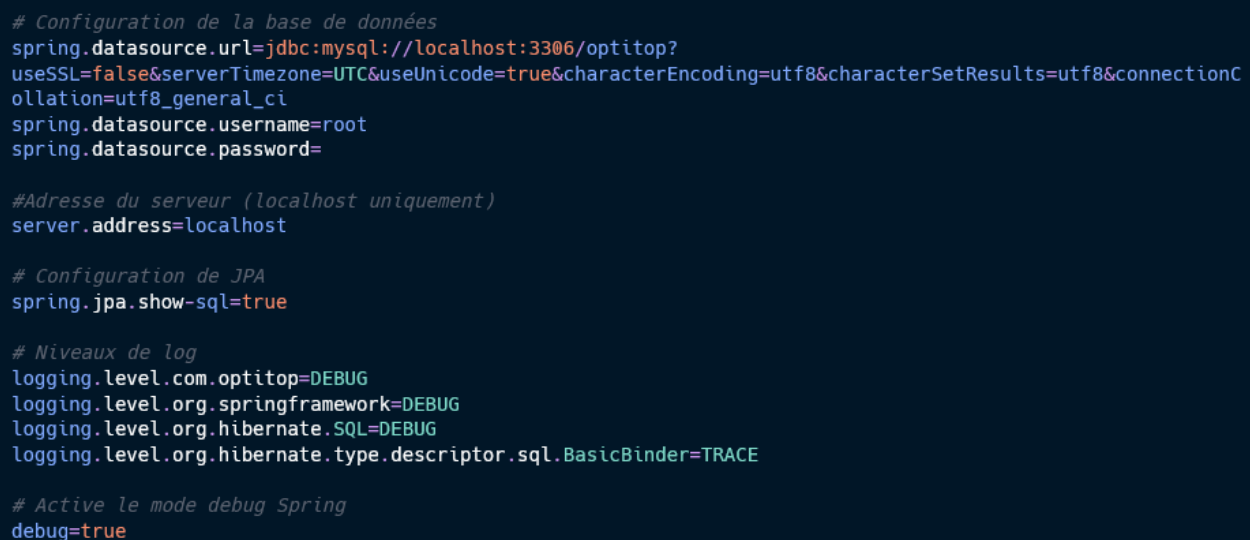
Arborescence du fichier resources contenant les fichiers .properties

2. Développement (profil dev)

L'environnement de développement est paramétré avec le fichier **application-dev.properties**

Ce fichier est utilisé lorsque je travaille uniquement en local. Il contient :

- l'URL de la base de données locale (**localhost**)
- les identifiants par défaut
- un niveau de logs très détaillé (**DEBUG, TRACE**) pour le débogage



```
# Configuration de la base de données
spring.datasource.url=jdbc:mysql://localhost:3306/optitop?
useSSL=false&serverTimezone=UTC&useUnicode=true&characterEncoding=utf8&characterSetResults=utf8&connectionC
ollation=utf8_general_ci
spring.datasource.username=root
spring.datasource.password=

#Adresse du serveur (localhost uniquement)
server.address=localhost

# Configuration de JPA
spring.jpa.show-sql=true

# Niveaux de log
logging.level.com.optitop=DEBUG
logging.level.org.springframework=DEBUG
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

# Active le mode debug Spring
debug=true
```

Fichier application-dev.properties

3. Pré-production (profil pprod)

a) Paramétrage de l'environnement

L'environnement de développement est paramétré avec le fichier **application-pprod.properties**

Ce fichier permet de se connecter à la base de données hébergée sur ma machine virtuelle Ubuntu. Il contient :

- l'adresse IP locale de la VM
- un utilisateur sécurisé **optitop_user**
- un mot de passe dédié
- un niveau de logs plus modéré

```
# Connexion à la base distante
spring.datasource.url=jdbc:mysql://XXX.XXX.X.XXX/optitop
spring.datasource.username=optitop_user
spring.datasource.password=XXXXXXXXX

# Écoute sur toutes les interfaces réseau
server.address=0.0.0.0

# Niveaux de log réduits pour la prod
logging.level.com.optitop=INFO
logging.level.org.springframework=INFO
logging.level.org.hibernate.SQL=WARN
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=OFF

# Pas de mode debug en prod
debug=false
```

Fichier application-pprod.properties

b) Configuration de MySQL sur la VM

Plusieurs ajustements ont été nécessaires pour autoriser l'accès distant.

➤ Modification du fichier de configuration

```
joy@joy-VM-UBUNTU01:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

Commande pour accéder au fichier de configuration mysql

```
GNU nano 7.2 /etc/mysql/mysql.conf.d/mysqld.cnf
# socket      = /var/run/mysqld/mysqld.sock
# port        = 3306
# datadir     = /var/lib/mysql

# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.>
# tmpdir      = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address      = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size    = 16M
# max_allowed_packet = 64M
# thread_stack     = 256K

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```

Définition de bind-adress sur 0.0.0.0

Cela permet à MySQL d'écouter sur toutes les interfaces réseau (et non plus seulement en local).

➤ Ouverture du pare-feu pour le port 3306

Autoriser les connexions entrantes à MySQL depuis le réseau local via le port 3306 est indispensable pour que l'API en local accède à la base distante.

```
396 sudo ufw allow 3306
```

Autorisation des connexions pour le port 3306

Il faut également autoriser les ports web (80 pour HTTP et 443 pour HTTPS) pour permettre l'accès au serveur Apache et donc à PhpMyAdmin.

```
joy@joy-VM-UBUNTU01:~$ sudo ufw status
[sudo] Mot de passe de joy :
État : actif

Vers      Action      De
----      -
3306      ALLOW      Anywhere
3306 (v6) ALLOW      Anywhere (v6)

joy@joy-VM-UBUNTU01:~$ sudo ufw allow 80/tcp
La règle a été ajoutée
La règle a été ajoutée (v6)
joy@joy-VM-UBUNTU01:~$ sudo ufw allow 443/tcp
La règle a été ajoutée
La règle a été ajoutée (v6)
```

Autorisation des connexions pour les ports 80 et 443

```
joy@joy-VM-UBUNTU01:~$ sudo ufw reload
Pare-feu rechargé
```

Rechargement des règles du pare-feu après modifications

```
joy@joy-VM-UBUNTU01:~$ sudo ufw status
[sudo] Mot de passe de joy :
État : actif
```

Vers	Action	De
----	-----	--
3306	ALLOW	Anywhere
80/tcp	ALLOW	Anywhere
443/tcp	ALLOW	Anywhere
3306 (v6)	ALLOW	Anywhere (v6)
80/tcp (v6)	ALLOW	Anywhere (v6)
443/tcp (v6)	ALLOW	Anywhere (v6)

Nouveau statut fonctionnel pour la connexion

Après avoir ajouté ces règles, l'accès aux services a été établie.

- **Point sécurité**

J'ai choisi de conserver l'autorisation "**Anywhere**" pour l'accès à mes services (ports 80, 443 et 3306) afin de garantir la compatibilité avec tout environnement réseau, notamment celui du centre d'examen. Sur un autre réseau que le mien, mon IP locale changerait, ce qui aurait empêché l'accès si j'avais restreint l'autorisation à une plage IP spécifique.

Ce choix est volontaire et adapté au contexte de l'épreuve.

En environnement de production réel, j'aurais restreint les accès uniquement au réseau local ou aux IP autorisées pour renforcer la sécurité.

c) Tests de fonctionnement

➤ Lancement de l'application pour pprod

Une fois la configuration en place, j'ai pu lancer mon API localement avec :

```
PS C:\Users\joyou\Informatique\Projets\Optitop\backend\optitop-api> mvn spring-boot:run "-Dspring-boot.run.profiles=pprod"
```

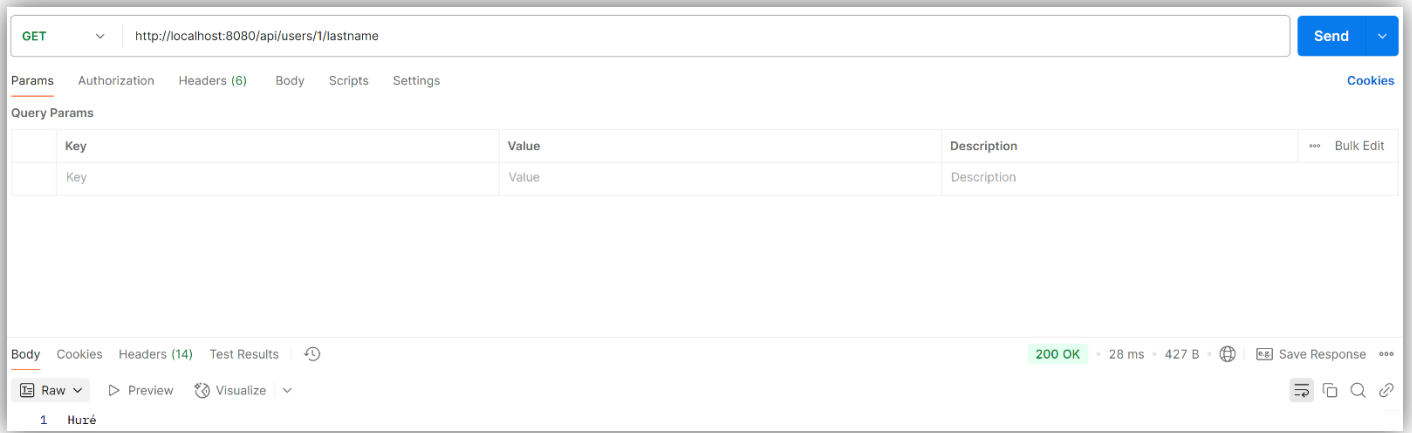
Lancement de l'application en pprod à partir du terminal de VS Code

```
2025-04-17 06:42:52 INFO c.o.o.OptitopApiApplication - The following 1 profile is active: "pprod"
```

Le terminal confirme que le profil actif est bien pprod

➤ Requête à l'API depuis Postman

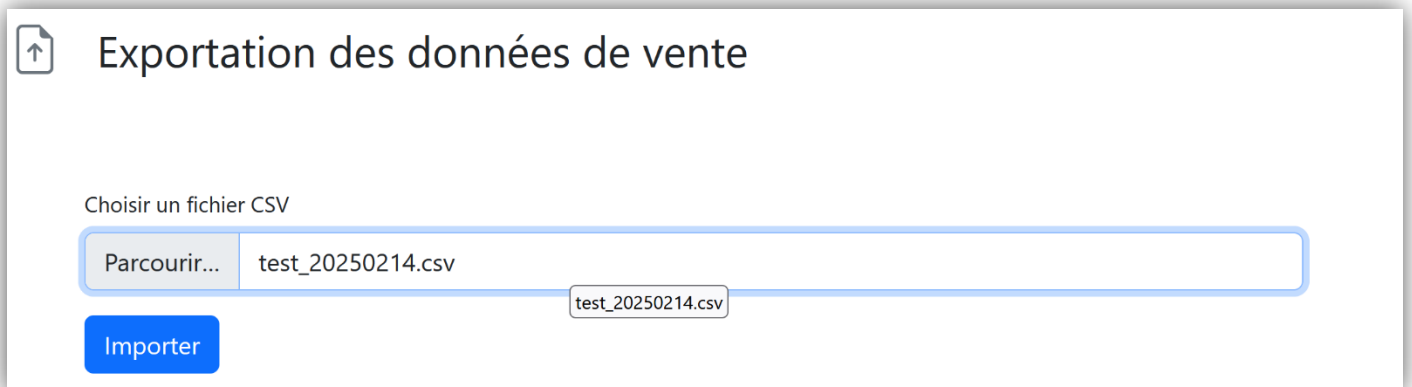
Une requête GET envoyée vers une route (ex. **/api/users/{id}/lastname**) permet de vérifier que les données sont bien récupérées depuis la base distante.



Requête GET dans Postman avec l'API Spring boot lancée avec le profil pprod retourne la donnée souhaitée

➤ Appel via le frontend web : test sur l'import d'un .csv

Depuis le frontend de l'application, toujours avec le profil pprod, j'ai testé la fonction d'import des données sur la page dédiée :



Sélection du fichier .csv à importer



Exportation des données de vente

Choisir un fichier CSV

Parcourir...

test_20250214.csv

Import en cours...

Importer

Import en cours...



Exportation des données de vente

Choisir un fichier CSV

Parcourir...

Aucun fichier sélectionné.

100%

Import réussi !

Importer

L'import a réussi !

Serveur : localhost:3306 » Base de données : optitop » Table : invoices_lines
 Parcourir Structure SQL Rechercher Insérer Exporter Importer Opérations Suivi Déclencheurs

Affichage des lignes 0 - 24 (total de 19936, traitement en 0.0008 seconde(s))
 SELECT * FROM `invoices\_lines`
 Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier par clé : Aucun(e)

Options supplémentaires

			id	date	client_id	client	invoice_ref	family	quantity	total_ttc	seller_ref	total_invoice	pair	status	created_at
☐	Éditer	Copier	Supprimer	630	2023-01-02	103007619	Marron Owen	F103056577	CLI	1	29 bstud	29	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	631	2023-01-02	103011989	Jacquín Lino	F103056579	DIV	1	40 irau	40	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	632	2023-01-02	103017425	Mazoillér Albane	F103056582	SOL	1	29 irau	29	NULL	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	633	2023-01-03	102018267	Maurelès Capucine	F103056586	MON	1	30 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	634	2023-01-03	102018267	Maurelès Capucine	F103056586	VER	1	90 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	635	2023-01-03	102018267	Maurelès Capucine	F103056586	VER	1	90 bstud	210	1	facture	2025-04-17 05:32:41
☐	Éditer	Copier	Supprimer	636	2023-01-03	103016833	Leger Anae	A103001088	ENT	-1	-6 egaligai	-6	NULL	avoir	2025-04-17 05:32:41

2eme preuve de réussite en consultant la BDD distante dont la table invoices_lines s’est remplie

4. Conclusion

La mise en place des profils **dev** et **pprod** m’a permis de structurer proprement mon application selon les bonnes pratiques de développement. Grâce à cette architecture :

- Je peux développer et tester en local sans interférer avec les données de production
- Je peux simuler un fonctionnement en condition réelle, avec une base distante sur une VM
- Je gagne en flexibilité : un seul fichier **.jar** permet d’adresser plusieurs contextes sans recompilation

Ce système prépare le terrain pour un futur déploiement en production complet (hébergement de l’API, sécurisation des accès, etc.).

D. Schéma définitif de l'infrastructure

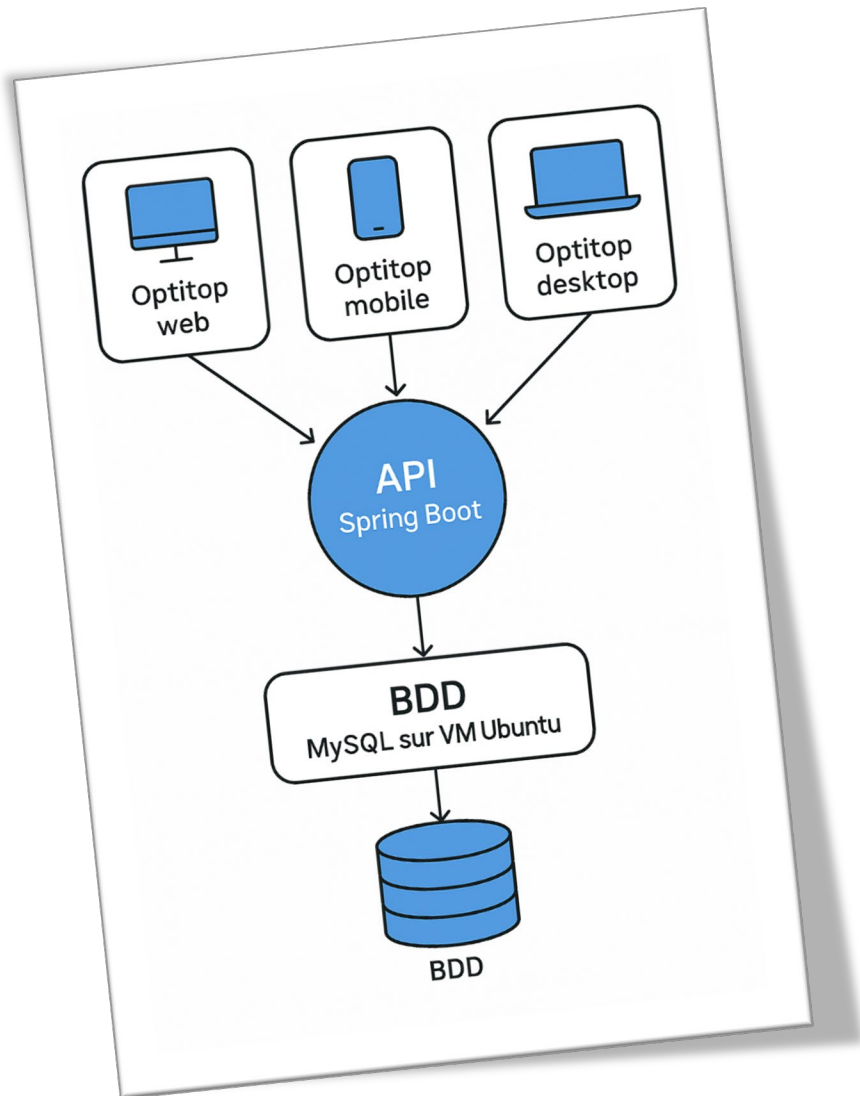


Schéma de l'infrastructure Optitop

IV. Versionnage

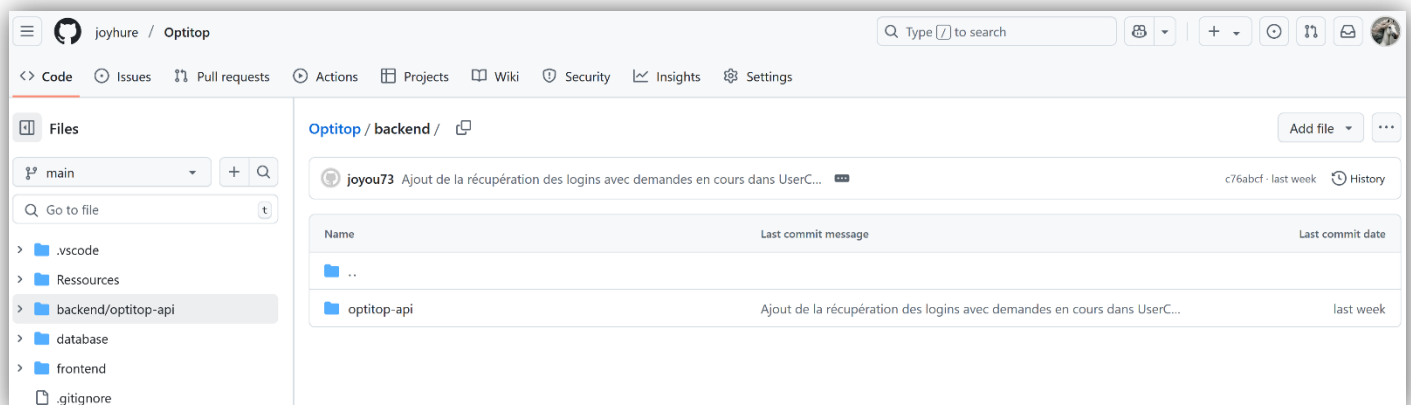
A. Dépôt GitHub du projet

Le projet Optitop est versionné à l'aide de **Git**, et hébergé sur un dépôt public GitHub à l'adresse <https://github.com/joyhure/Optitop>

Ce dépôt regroupe :

- Le code source de l'API Spring Boot
- Les fichiers du frontend PHP/JavaScript
- Les scripts SQL (structure de la base + données de test)
- Les FRP, les annexes et la documentation nécessaire de l'épreuve E6

Le versionnage Git permet un suivi précis de l'évolution du projet, avec un historique de commits clair. Le dépôt est passé en public afin de faciliter l'évaluation par le jury concernant l'épreuve E6 du BTS SIO SLAM.

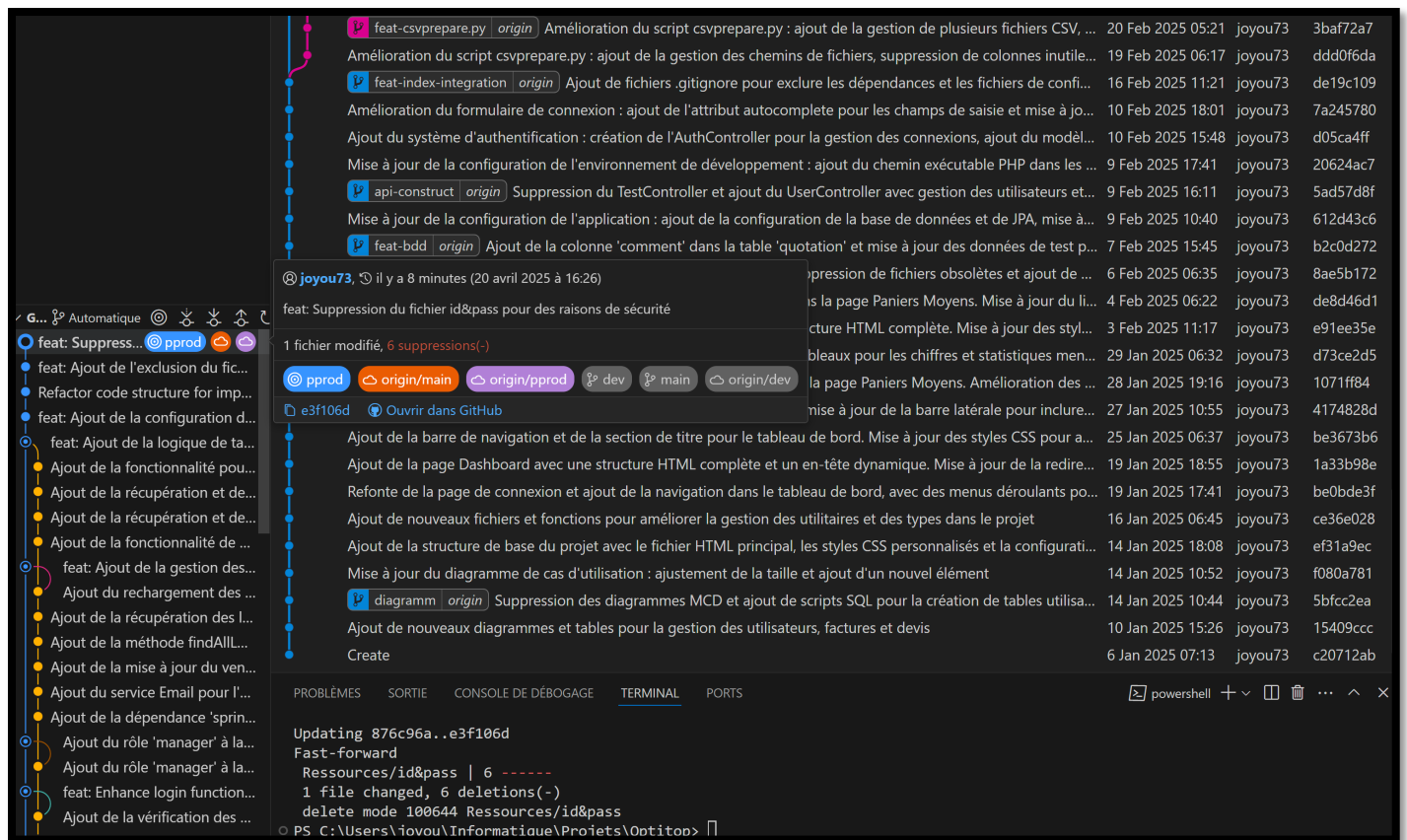


Dépôt Optitop sur github

B. Initialisation du dépôt Optitop

Le dépôt GitHub a été initialement créé via le terminal intégré de VS Code à l'aide des commandes :

- **git init** : initialise un nouveau dépôt Git dans le dossier courant.
- **git add .** : Ajoute tous les fichiers au staging, une zone temporaire, où nous pouvons choisir quels fichiers valider lors du prochain commit.
- **git commit -m "Create"** : Valide les fichiers dans l'historique Git
- **git branch -M main** : Renomme la branche par défaut en **main**
- **https://github.com/joyhure/Optitop.git**: Lie le dépôt local à celui de GitHub
- **git push -u origin main** : Envoyer les modifications vers le dépôt distant



Arborescence des branches avec leurs commits etc...

C. Utilisation de Git et GitHub

Afin de garantir une traçabilité claire et une stabilité du code, j'ai structuré mon dépôt Git en suivant une organisation simple et commune.

La branche **main** est réservée aux versions stables tandis que la branche **dev** sert de zone de développement intermédiaire.

Toutes les nouvelles fonctionnalités ou corrections sont réalisées dans des branches dérivées de **dev**, nommées selon la convention **feat-nom-fonctionnalité** pour les ajouts, ou **fix-nom-correctif** pour les corrections de bugs.

Je réalise des commits très réguliers au sein de ces branches, ce qui me permet de conserver un historique de développement lisible et précis.

Une fois les développements terminés et testés, la branche de travail est fusionnée dans **dev**. Après validation finale, **dev** est ensuite fusionnée dans **main** pour publication.

```
14 git checkout dev
15 git merge feat-pprod
16 git checkout main
17 git merge dev
18 git checkout feat-pprod
```

Merges progressifs de la branche feat-pprod jusqu'à la branche main puis retour sur cette branche

```
18 git checkout feat-pprod
```

V. Implémentation

A. Choix des langages

1. JavaScript

J'ai choisi JavaScript pour optimiser l'expérience utilisateur et garantir une interaction fluide avec l'API.

Voici ses principaux avantages par exemple pour la page index.html qui est aussi la page de connexion :

- **Validation en temps réel** : JavaScript permet de vérifier les champs avant l'envoi au serveur, évitant ainsi les requêtes inutiles et améliorant la réactivité de l'application. L'utilisateur est informé immédiatement en cas d'erreur.
- **Expérience utilisateur optimisée** : Grâce à JavaScript l'authentification se fait sans rechargement de page. Cela réduit les temps d'attente et offre une navigation plus fluide.
- **Gestion avancée des erreurs** : JavaScript permet d'afficher des messages d'erreur contextualisés directement sous les champs, rendant les corrections plus intuitives pour l'utilisateur.
- **Stockage temporaire des sessions** : Après connexion, les données de l'utilisateur peuvent être stockées en **sessionStorage**, évitant de redemander l'authentification à chaque rafraîchissement de page et facilitant la gestion des sessions utilisateur.
- **Séparation du front et du back** : En gérant la logique directement côté client, JavaScript allège la charge du serveur et permet une architecture modulaire et évolutive.

En conclusion, JavaScript m'a semblé être un choix pertinent pour optimiser la performance et fluidifier l'expérience utilisateur. Son intégration avec une API REST permet une interaction rapide et efficace entre le client et le backend en Java/Spring Boot.

2. PHP

a) Sécurisation de l'accès aux pages selon les rôles

Bien que JavaScript permette de gérer dynamiquement l'interface utilisateur et d'empêcher l'affichage de certaines pages en fonction du rôle, il ne peut pas empêcher un utilisateur malveillant d'accéder directement à une URL en la tapant dans la barre d'adresse.

PHP permet en revanche :

- **Un contrôle d'accès sécurisé** : PHP fonctionne côté serveur, ce qui permet de vérifier le rôle de l'utilisateur avant de charger la page.
- **Une Protection contre les accès directs** : Si une page est protégée avec PHP, un utilisateur non autorisé ne pourra pas y accéder, même s'il tape son URL.
- **Une session sécurisée** : Avec **\$_SESSION**, PHP permet de stocker le rôle de l'utilisateur et de bloquer l'accès aux pages non autorisées.

```
<?php
session_start();

// Vérification PHP principale
if (!isset($_SESSION['user']) ||
    !isset($_SESSION['user']['role']) ||
    !in_array($_SESSION['user']['role'], ['admin', 'supermanager'])) {
    header('Location: dashboard.php');
    exit();
}
?>
```

Ici, un utilisateur qui n'est pas admin ni supermanager sera automatiquement redirigé, même s'il connaît l'URL de la page protégée

3. Une complémentarité idéale entre PHP et Javascript

PHP et JavaScript sont complémentaires dans mon application web :

JavaScript permet d'améliorer l'interface utilisateur en rendant l'application plus fluide et interactive.

PHP assure la sécurité et le contrôle d'accès côté serveur.

B. Préparation de fichiers .csv exploitables

1. Problématique

Dans le cadre de mon projet, je dois exploiter des fichiers CSV issus de mon logiciel métier contenant des données sensibles pour mon entreprise et volumineuses. Plusieurs contraintes m'amènent donc à modifier ces fichiers avant leur utilisation dans le cadre du BTS.

a) Confidentialité des données

Les fichiers contiennent des informations sensibles telles que :

- Les noms des vendeurs et des clients qui doivent être anonymisés.
- Les références des produits vendus et les prix d'achat qui sont des données internes à l'entreprise et non pertinentes pour l'analyse demandée.

b) Volume des données

Les fichiers contiennent un grand nombre de lignes dont le volume n'est pas nécessaire au projet dans le contexte du BTS.

La réduction du volume permettra de se concentrer sur des données représentatives et éviter de manipuler un jeu de données trop lourd.

c) Simplification du jeu de données

Certains types de données ne sont pas utiles dans le cadre de mon analyse et contiennent des données sensibles. Ces types de données seront supprimés (ex: références précises des produits, numéros de commande internes, correction optique des clients).

Les informations restantes doivent être cohérentes avec le fichier CSV, même si elles sont fictives car le but final va au-delà de l'examen du BTS : je voudrais déployer cette solution en production par la suite afin de pouvoir réellement l'utiliser en magasin.

2. Script en Python

Pour répondre à ces besoins, j'ai choisi de développer un script en Python, qui offre plusieurs avantages.

a) Avantages

➤ Automatisation et gain de temps

Un script Python permet de traiter automatiquement plusieurs fichiers CSV sans avoir à les modifier manuellement.

Il garantit une homogénéité des fichiers en appliquant toujours les mêmes transformations.

➤ Facilité de manipulation des données avec Pandas

Python et la bibliothèque Pandas permettent de charger, modifier et sauvegarder des fichiers CSV de manière rapide et efficace.

Il est facile de remplacer des valeurs sensibles, supprimer des colonnes ou réduire le nombre de lignes.

➤ Flexibilité

Le script sera réutilisé et adapté à d'autres fichiers notamment lors de l'implémentation de la page « export ».

Il pourra intégrer des règles d'anonymisation évolutives (ex: remplacement des noms par des identifiants aléatoires, modification des dates pour garder une cohérence) et me permettra ainsi de pouvoir obtenir des données toujours adaptées à mes besoins pour tester mon application.

➤ Respect de la logique métier

Python permet de générer des données fictives mais cohérentes (ex: des noms de vendeurs aléatoires mais définis dans une liste, des montants de vente modifiés mais plausibles).

b) Installation des outils nécessaires

➤ Python et pandas

Pour ce faire j'ai dû installer certains outils. Python était déjà installé, ce qui n'était pas le cas de la bibliothèque pandas. J'ai utilisé pip pour mener à bien cette opération :

```
PS C:\Users\joyou\Informatique\Projets\Optitop_mag> python --version
Python 3.12.7
PS C:\Users\joyou\Informatique\Projets\Optitop_mag> python -c "import pandas; print(pandas.__version__)"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'pandas'
PS C:\Users\joyou\Informatique\Projets\Optitop_mag> pip3 install pandas
```

Vérification de la présence de python, de pandas puis installation de pandas avec pip

➤ Chardet

J'ai rencontré des erreurs d'encodage en lançant mes premiers scripts python. Pour y remédier, j'ai utilisé **chardet** afin de détecter automatiquement l'encodage des fichiers avant leur ouverture, garantissant ainsi une lecture correcte des données.

```
pip install chardet
```

Installation de chardet

c) Script

```
import pandas as pd
import chardet
import os
import numpy as np
import json

def detect_file_encoding(file_path):
    # Detect file encoding
    with open(file_path, 'rb') as file:
        raw_data = file.read()
        result = chardet.detect(raw_data)
        print(f"Detected encoding: {result}")
        return result['encoding']

try:
    # Define base directory
    base_dir = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
    input_dir = os.path.join(base_dir, "Fichiers .csv", "Originaux")

    # List available CSV files
    csv_files = [f for f in os.listdir(input_dir) if f.endswith('.csv')]
    print("\nFichiers CSV disponibles :")
    for i, file in enumerate(csv_files, 1):
        print(f"{i}. {file}")

    # Get user input for multiple files
    print("\nEntrez les numéros des fichiers (séparés par des espaces) : ")
    file_numbers = [int(x) for x in input().split()]

    # Process each selected file
    for file_number in file_numbers:
        if 1 <= file_number <= len(csv_files):
            print(f"\nTraitement du fichier {csv_files[file_number - 1]}...")
            input_filename = csv_files[file_number - 1]
            file_path = os.path.join(input_dir, input_filename)

            # Create output filename
            output_filename = f"test_{input_filename}"
            output_dir = os.path.join(base_dir, "Fichiers .csv", "Tests")
            output_path = os.path.join(output_dir, output_filename)

            # Ensure output directory exists
            os.makedirs(output_dir, exist_ok=True)

            encoding = detect_file_encoding(file_path)

            # Read the file using line 3 as header (index 2)
            data = pd.read_csv(file_path,
                               delimiter=';',
                               encoding=encoding,
                               header=2,
                               dtype={'Num client': str, 'Paire': str},
                               on_bad_lines='warn')

            print(f"\nNumber of rows read: {len(data)}")
            print(f"Found columns: {list(data.columns)}")
```

```

# Delete specified columns
columns_to_delete = [
    "Code article", "Ref. Mag.", "Société", "Libellé",
    "Taille", "Couleur", "Matière", "Type de Verre",
    "Genre", "Marque", "Fournisseur", "Code Barre",
    "Gamme", "Prix Achat", "Marge", "MAJ Stock"
]
data = data.drop(columns=columns_to_delete, errors='ignore')

# Identify clients to consider, excluding "PASSAGE passage"
clients_to_consider = data['Client'].unique()
clients_to_consider = [c for c in clients_to_consider if c != "PASSAGE passage"]

# Randomly select about 50% of clients
num_clients_to_remove = int(len(clients_to_consider) * 0.5)
clients_to_remove = set(pd.Series(clients_to_consider).sample(num_clients_to_remove,
random_state=42))

# Filter rows
data_filtered = data[~data['Client'].isin(clients_to_remove)]

# List of possible vendors
vendors = ['irau', 'bstud', 'hiskander', 'egaligai']

# Create a dictionary to map each unique client to a random vendor
unique_clients = data_filtered['Client'].unique()
client_vendor_mapping = {
    client: np.random.choice(vendors)
    for client in unique_clients if client != "PASSAGE passage"
}

# Apply the mapping to replace RefVendeur values for non-PASSAGE clients
data_filtered['RefVendeur'] = data_filtered.apply(
    lambda row: np.random.choice(vendors) if row['Client'] == "PASSAGE passage"
    else client_vendor_mapping[row['Client']],
    axis=1
)

print("\nVendor distribution after replacement:")
print(data_filtered['RefVendeur'].value_counts())

# Check for PASSAGE passage in data_filtered['Client'].unique():

# Load JSON data
json_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), "nomprenom.json")
with open(json_path, 'r', encoding='utf-8') as f:
    name_data = json.loads(f.read())

# Get unique clients (excluding "PASSAGE passage")
unique_clients = data_filtered['Client'].unique()
unique_clients = [c for c in unique_clients if c != "PASSAGE passage"]

# Create random full names
full_names = [
    f"{np.random.choice(name_data['last_names'])}
{np.random.choice(name_data['first_names'])}"
    for _ in range(len(unique_clients))
]

# Create mapping dictionary
client_name_mapping = dict(zip(unique_clients, full_names))

# Add "PASSAGE passage" back if it exists
if "PASSAGE passage" in data_filtered['Client'].unique():
    client_name_mapping["PASSAGE passage"] = "PASSAGE passage"

# Apply the mapping to replace Client values
data_filtered['Client'] = data_filtered['Client'].map(client_name_mapping)

# Save modified file
data_filtered.to_csv(output_path, index=False, sep=';', encoding='utf-8-sig')
print(f"\nFile saved as: {output_filename}")

else:
    print(f"Numéro de fichier invalide ignoré : {file_number}")

except ValueError as ve:
    print(f"Erreur de saisie : {str(ve)}")
except Exception as e:
    print(f"Erreur : {str(e)}")

```

VI. Sécurité

A. Chiffrement des mots de passe avec BCrypt

Afin d'assurer la protection des comptes utilisateurs, les mots de passe sont chiffrés grâce à l'algorithme BCrypt via un service dédié dans l'API. Ce chiffrement est automatiquement appliqué lors de la création d'un compte ou de la modification d'un mot de passe.

1. Ajout des dépendances dans pom.xml

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Ajout des dépendances dans pom.xml

2. Création du service de hachage

```
package com.optitop.optitop_api.service;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Service;

@Service
public class PasswordService {

    private final BCryptPasswordEncoder passwordEncoder;

    public PasswordService() {
        this.passwordEncoder = new BCryptPasswordEncoder();
    }

    public String hashPassword(String plainTextPassword) {
        return passwordEncoder.encode(plainTextPassword);
    }

    public boolean verifyPassword(String plainTextPassword, String hashedPassword) {
        return passwordEncoder.matches(plainTextPassword, hashedPassword);
    }
}
```

Création du fichier PasswordService.java (service de hachage)

3. Modification des Controllers

a) Controller de changement de mot de passe

```
@RestController
@RequestMapping("/api/users")
public class UserController {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private PasswordService passwordService;

    @PostMapping("/{id}/change-password")
    public ResponseEntity<String> changePassword(
        @PathVariable Integer id,
        @RequestBody PasswordChangeRequest request) {

        // Validation des données
        if (request.getCurrentPassword() == null || request.getNewPassword() == null) {
            return ResponseEntity.badRequest().body("Les mots de passe ne peuvent pas être vides");
        }

        User user = userRepository.findById(id).orElse(null);
        if (user == null) {
            return ResponseEntity.notFound().build();
        }

        // Validation du nouveau mot de passe
        if (!isValidPassword(request.getNewPassword())) {
            return ResponseEntity.badRequest()
                .body("Le nouveau mot de passe ne respecte pas les critères de sécurité");
        }

        // Hashage et mise à jour du mot de passe
        String hashedPassword = passwordService.hashPassword(request.getNewPassword());
        user.setPassword(hashedPassword);
        userRepository.save(user);

        return ResponseEntity.ok("Mot de passe modifié avec succès");
    }

    // ... reste du code ...
}
```

Modification de UserController (changement de mot de passe)

b) Controller d'authentification

```
@RestController
@RequestMapping("/api/auth")
@CrossOrigin(origins = "http://localhost")
public class AuthController {

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private PasswordService passwordService;

    @PostMapping("/login")
    public ResponseEntity<Map<String, Object>> login(@RequestBody LoginRequest loginRequest) {
        User user = userRepository.findByLogin(loginRequest.getLogin());

        if (user != null && passwordService.verifyPassword(loginRequest.getPassword(), user.getPassword()))
        {
            Map<String, Object> response = new HashMap<>();
            response.put("id", user.getId());
            response.put("firstname", user.getFirstname());
            response.put("role", user.getRole());
            response.put("seller_ref", user.getLogin()); // Renommé en seller_ref pour plus de clarté

            return ResponseEntity.ok(response);
        }

        return ResponseEntity.status(HttpStatus.UNAUTHORIZED)
            .body(Map.of("error", "Identifiants incorrects"));
    }

    // ...reste du code ...
}
```

Modifications de AuthController (authentification)

4. Script de migration des mots de passe

```
package com.optitop.optitop_api.utils;

import com.optitop.optitop_api.model.User;
import com.optitop.optitop_api.repository.UserRepository;
import com.optitop.optitop_api.service.PasswordService;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.transaction.annotation.Transactional;

import jakarta.annotation.PostConstruct;
import java.util.List;

@Component
public class PasswordMigrationUtil {

    private static final Logger logger = LoggerFactory.getLogger(PasswordMigrationUtil.class);

    @Autowired
    private UserRepository userRepository;

    @Autowired
    private PasswordService passwordService;

    @PostConstruct
    @Transactional
    public void migratePasswords() {
        logger.info("Début de la migration des mots de passe...");
        int count = 0;

        List<User> users = userRepository.findAll();

        for (User user : users) {
            if (!isPasswordHashed(user.getPassword())) {
                String hashedPassword = passwordService.hashPassword(user.getPassword());
                user.setPassword(hashedPassword);
                userRepository.save(user);
                count++;
                logger.debug("Mot de passe migré pour l'utilisateur: {}", user.getLogin());
            }
        }

        logger.info("Migration terminée. {} mots de passe ont été hashés", count);
    }

    private boolean isPasswordHashed(String password) {
        return password != null && (password.startsWith("$2a$") ||
            password.startsWith("$2b$") ||
            password.startsWith("$2y$"));
    }
}
```

Script de migration des mots de passe, hashage des mots de passe existants

B. Gestion des mots de passe en cas d'ajout d'utilisateur

1. Contexte

Un nouvel utilisateur est ajouté selon cette procédure :

- Un nouveau vendeur est ajouté à la table seller par l'import du fichier .csv, s'il n'existe pas encore. Son user_id est alors « NULL »

←T→				▼ id	seller_ref	created_at	user_id	
☐		Éditer	 Copier	 Supprimer	146	atest	2025-04-06 16:47:49	NULL
☐		Éditer	 Copier	 Supprimer	147	egaligai	2025-04-06 16:47:49	9
☐		Éditer	 Copier	 Supprimer	148	btest	2025-04-06 16:47:49	NULL
☐		Éditer	 Copier	 Supprimer	149	bstud	2025-04-06 16:47:49	10
☐		Éditer	 Copier	 Supprimer	150	hiskander	2025-04-06 16:48:16	8
☐		Éditer	 Copier	 Supprimer	151	irau	2025-04-06 16:48:16	6

Table seller où les seller_ref « atest » et « btest » sont des vendeurs sans compte user : user_id = NULL

- Un admin, supermanager ou manager pourra alors faire une demande de création de compte pour chaque vendeur d'user_id **NULL**. L'application propose uniquement les vendeurs référencés dans seller sans compte utilisateur existant pour les demandes d'ajout.

 Comptes Utilisateurs

Nouvelle demande

Ajouter

Nom

Prénom

Email

Collaborateur

atest

Envoyer

Demands en cours

Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
12/04/2025	jhure	GALIGAI	Eden	egaligai	collaborator	egaligai@example.fr	suppression	Valider Refuser
12/04/2025	jhure	RAU	Isidore	irau	collaborator	iroas@example.fr	modification	Valider Refuser

Création d'une demande d'ajout de compte utilisateur possible pour les vendeurs atest et btest uniquement

- La demande crée s'enregistre dans la table pending_accounts qui référence toutes les demandes en cours sur les comptes.

			id	lastname	firstname	email	login	role	request_type	created_by_user_id	created_at
☐	Éditer	Copier	Supprimer	20	RAU	Isidore	iroas@example.fr	irau	collaborator	modification	5 2025-04-12 11:51:51
☐	Éditer	Copier	Supprimer	21	GALIGAI	Eden	egaligai@example.fr	egaligai	collaborator	suppression	5 2025-04-12 11:52:01
☐	Éditer	Copier	Supprimer	22	TEST	Aurore	jhure.dev@gmail.com	atest	collaborator	ajout	5 2025-04-12 11:55:48

Table pending_accounts : demandes en cours sur les comptes utilisateurs

- Les utilisateurs de rôle « admin » accèdent à la possibilité de valider ou de refuser la demande soumise.

Demandes en cours ▼								
Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
12/04/2025	jhure	TEST	Aurore	atest	collaborator	jhure.dev@gmail.com	ajout	<button>Valider</button> <button>Refuser</button>
12/04/2025	jhure	GALIGAI	Eden	egaligai	collaborator	egaligai@example.fr	suppression	<button>Valider</button> <button>Refuser</button>
12/04/2025	jhure	RAU	Isidore	irau	collaborator	iroas@example.fr	modification	<button>Valider</button> <button>Refuser</button>

Demandes en cours, à traiter par un administrateur.

- Si un administrateur valide un ajout, le vendeur est ajouté à la table « user », son mot de passe est défini aléatoirement et haché avec bcrypt

2. Configuration SMTP pour Gmail

Pour permettre à l'application Optitop d'envoyer les mots de passe générés par email aux utilisateurs via les serveurs SMTP de Gmail, il est nécessaire de générer un mot de passe d'application. Cette méthode garantit une connexion sécurisée en contournant l'authentification classique par mot de passe.

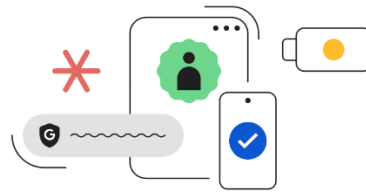
Étapes :

- Se connecter à son compte Google via l'adresse suivante : <https://myaccount.google.com/security>
- Activer l'authentification à deux facteurs (2FA).

Activer la validation en deux étapes

Protégez votre compte des pirates informatiques avec un niveau de sécurité supplémentaire.

Si vous ne vous connectez pas à l'aide d'une clé d'accès, vous aurez à effectuer la plus sécurisée des secondes étapes disponibles dans votre compte. Vous pouvez à tout moment modifier vos secondes étapes et vos options de connexion dans les paramètres. [Accéder aux paramètres de sécurité](#)



Activer la validation en deux étapes

Activer l'authentification à deux facteurs

- Dans la section "**Connexion à Google**", cliquer sur "**Mots de passe des applications**".
- Donner un nom explicite, ici "**Optitop API**", puis cliquer sur **Créer**.

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google sur des applis et des services plus anciens, non compatibles avec les normes de sécurité les plus récentes.

Les mots de passe d'application sont moins sécurisés que les applis et services à jour qui utilisent les normes de sécurité les plus récentes. Avant de créer un mot de passe d'application, vous devez vérifier si votre appli en a besoin pour établir la connexion.

[En savoir plus](#)

Vous n'avez aucun mot de passe d'application.

Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous.

Nom de l'appli
Optitop API

Créer

Renseigner un nom explicite puis « Créer »

Google fournit un mot de passe sécurisé de 16 caractères, à conserver précieusement. Ce mot de passe sera utilisé dans la configuration SMTP de l'application à la place du mot de passe classique.

3. Stockage des paramètres nécessaires en base de données

a) Création de la table email_config

```
CREATE TABLE email_config (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  smtp_host VARCHAR(255) NOT NULL,  
  smtp_port INT NOT NULL,  
  smtp_username VARCHAR(255) NOT NULL,  
  smtp_password VARCHAR(255) NOT NULL,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP  
);
```

Script de création de la table email_config

b) Insertion des valeurs dans la table email_config

```
INSERT INTO email_config (smtp_host, smtp_port, smtp_username, smtp_password)  
VALUES ('smtp.gmail.com', 587, 'jhure.dev@gmail.com', 'xxxx xxxx xxxx xxxx');
```

Script d'insertion des paramètres pour l'envoi automatique des emails

c) Création de l'entité Springboot EmailConfig

```
package com.optitop.optitop_api.model;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name = "email_config")
public class EmailConfig {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @Column(name = "smtp_host", nullable = false)
    private String smtpHost;

    @Column(name = "smtp_port", nullable = false)
    private Integer smtpPort;

    @Column(name = "smtp_username", nullable = false)
    private String smtpUsername;

    @Column(name = "smtp_password", nullable = false)
    private String smtpPassword;
}
```

Entité EmailConfig correspondant à la table email_config de la BDD

4. Modification des fichiers Service.java impliqués

a) Création de EmailService.java

```
@Service
public class EmailService {

    @Autowired
    private EmailConfigRepository emailConfigRepository;

    private JavaMailSender createMailSender() {
        EmailConfig config = emailConfigRepository.findFirstByOrderByIdDesc();
        if (config == null) {
            throw new RuntimeException("Aucune configuration email trouvée en base de données");
        }

        JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
        mailSender.setHost(config.getSmtpHost());
        mailSender.setPort(config.getSmtpPort());
        mailSender.setUsername(config.getSmtpUsername());
        mailSender.setPassword(config.getSmtpPassword());

        Properties props = mailSender.getJavaMailProperties();
        props.put("mail.transport.protocol", "smtp");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.ssl.trust", "smtp.gmail.com");
        props.put("mail.smtp.ssl.protocols", "TLSv1.2");
        props.put("mail.debug", "true");

        return mailSender;
    }

    public void sendPasswordEmail(String to, String login, String password) {
        EmailConfig config = emailConfigRepository.findFirstByOrderByIdDesc();

        SimpleMailMessage message = new SimpleMailMessage();
        message.setFrom(config.getSmtpUsername());
        message.setTo(to);
        message.setSubject("Vos identifiants Optitop");
        message.setText(String.format("""
            Bonjour,

            Votre compte Optitop a été créé avec succès.

            Vos identifiants de connexion :
            Login : %s
            Mot de passe : %s

            Pour des raisons de sécurité, veuillez changer votre mot de passe lors de votre première
            connexion.

            Cordialement,
            L'équipe Optitop
            """, login, password));

        JavaMailSender sender = createMailSender();
        sender.send(message);
    }
}
```

Fichier EmailService.java qui gère l'envoi de l'email lors de la création de compte

b) Modification de PendingAccountService.java

```
switch (pending.getRequestType()) {
    case ajout:
        User newUser = new User();
        newUser.setLastname(pending.getLastname());
        newUser.setFirstname(pending.getFirstname());
        newUser.setEmail(pending.getEmail());
        newUser.setLogin(pending.getLogin());
        newUser.setRole(pending.getRole());

        // Génération et hachage du mot de passe
        String rawPassword = generateSecurePassword();
        newUser.setPassword(passwordEncoder.encode(rawPassword));

        try {
            emailService.sendPasswordEmail(newUser.getEmail(), newUser.getLogin(), rawPassword);
            userRepository.save(newUser);
        } catch (Exception e) {
            throw new RuntimeException("Erreur lors de l'envoi de l'email : " + e.getMessage());
        }
        break;
```

Ajout de la logique d'envoi d'email en cas d'ajout d'utilisateur dans PendingAccountService.java

5. Test fonctionnel : Envoi d'un email à la création d'un compte utilisateur

a) Objectif

Après avoir finalisé la configuration SMTP avec Gmail et intégré le hachage des mots de passe dans mon application, j'ai procédé à un test fonctionnel consistant à créer un nouvel utilisateur via l'interface prévue à cet effet.

L'objectif était de vérifier que :

- le mot de passe est bien haché en base,
- un email de confirmation est automatiquement envoyé à l'adresse indiquée lors de l'inscription.
- L'email contient le mot de passe en clair

b) Mise en œuvre

Demandes en cours ▾								
Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
12/04/2025	jhure	localhost				jhure.dev@gmail.com	ajout	Valider Refuser
12/04/2025	jhure	Demande validée avec succès				egaligai@example.fr	suppression	Valider Refuser
12/04/2025	jhure					iroas@example.fr	modification	Valider Refuser

Validation d'ajout pour le login atest est un succès en frontend

id	lastname	firstname	email	login	password	role	created_at
5	HURE	Joy	jhure.dev@gmail.com	jhure	\$2a\$10\$Ed6/5mAgmQSW62aj8cUNM.W9dFbrfkndC.08Ua9LCc9...	admin	2025-02-15 05:49:19
6	RAU	Isidore	iroas@example.fr	irau	\$2a\$10\$YKF8l9GHZAB.WTBntim4ju64g5PCqQFZ.Ig8u/.DN7p...	manager	2025-02-15 05:50:46
7	DUPANLOUP	Ludovic	ldupanloup@example.fr	ldupanloup	\$2a\$10\$00S1CjE6cMnr7dCxToeFY.bb6RdoU8VLcgnpqfMIOSB...	supermanager	2025-02-15 05:54:09
8	ISKANDER	Henry	hiskander@example.fr	hiskander	\$2a\$10\$uiRQFF9WPSl6hYCDltIAiumcrGIFnwxDR32xPMnJnml...	collaborator	2025-02-15 06:02:15
9	GALIGAI	Eden	egaligai@example.fr	egaligai	\$2a\$10\$Tg02pabF4kfTej1OIMBnDOgT.DflBdjNSA6lo8343/x...	collaborator	2025-02-15 06:02:15
10	STUD	Barina	bstud@example.fr	bstud	\$2a\$10\$7Hrhyh.PcVAr51CRtE/P8OIFp29HPgdO4SJ9hvWBxZ....	collaborator	2025-02-15 06:03:40
19	TEST	Aurore	jhure.dev@gmail.com	atest	\$2a\$10\$/esjDjvr4DfPm2fOud/z6ev42cu9hfnNbfmwO43NbfV...	collaborator	2025-04-12 16:44:04

Le compte de login atest est ajoutée à la table user, son password est haché

c) Résultat

Vos identifiants Optitop

Boîte de réception x

**jhure.dev@gmail.com**
À moi ▾
Bonjour,

Votre compte Optitop a été créé avec succès.

Vos identifiants de connexion :
Login : atest
Mot de passe : 4]mzL[6[AU]-

Pour des raisons de sécurité, veuillez changer votre mot de passe lors de votre première connexion.

Cordialement,
L'équipe Optitop

Email reçu avec les identifiants

L'email a bien été reçu sur l'adresse de test utilisée, confirmant le bon fonctionnement de la configuration SMTP et du workflow de création de compte.

VII. Gestion de projet et organisation

A. Trello pour le suivi du projet

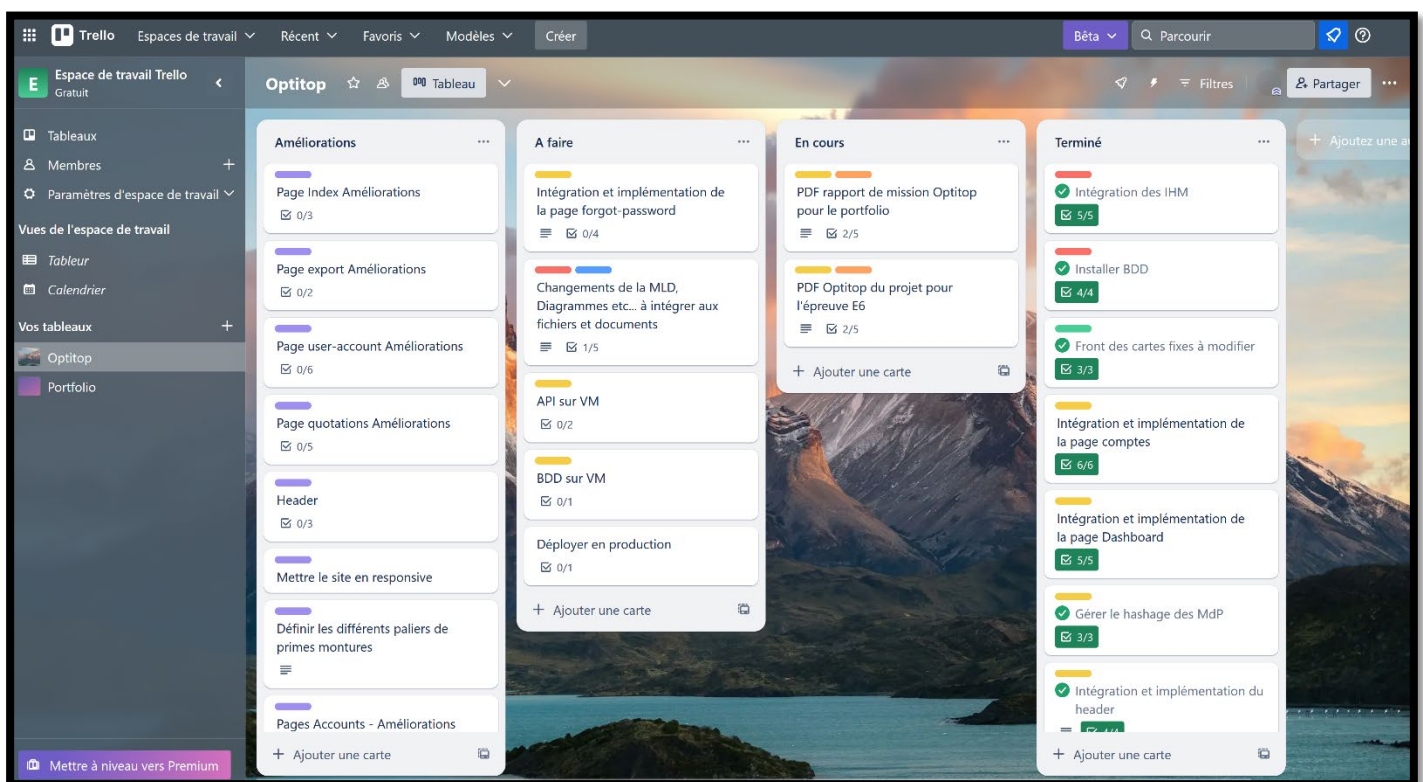
Afin de structurer les différentes tâches de développement du projet Optitop, j'ai utilisé **Trello**, un outil de gestion de projet en ligne basé sur la méthode Kanban.

1. Méthode Kanban

La méthode Kanban est une technique de gestion de projet visuelle. Elle consiste à représenter les tâches à l'aide de cartes réparties dans des colonnes qui symbolisent l'état d'avancement du travail.

Par exemple, mon tableau Kanban se compose de 4 colonnes :

- **Améliorations** : Tout ce que j'aimerais améliorer mais que je n'ai pas pu faire par manque de temps, hélas !
- **À faire** : tâches prévues mais pas encore commencées,
- **En cours** : tâches en cours de développement,
- **Terminé** : tâches terminées.



Mon espace Optitop sur Trello, les 4 colonnes du tableau Kanban pour l'application web

Chaque carte représente une tâche (ex. : "Créer l'import CSV", "Faire le diagramme de classe", etc.), et elle est changée de colonne au fur et à mesure de l'avancement.

Cet outil m'a permis de :

- Lister les fonctionnalités prévues,
- Organiser les priorités (import .csv, backoffice, création de comptes...),
- Suivre ma progression avec des colonnes "À faire", "En cours", "Terminé",
- Ajouter des checklists internes à certaines cartes,

Intégration et implémentation de la page Dashboard

dans la liste **TERMINÉ** ▾

Étiquettes

Dependant d'autre(s) tâches

+

Notifications

👁 Suivre

☰

Description

Ajouter une description plus détaillée...

☑ Prérequis

Masquer les tâches cochées

Supprimer

100%

☑

Le header et la sélection des périodes fonctionnent au préalable

☑

L'importation de données fonctionnent et des données sur plusieurs années sont présentes

Ajouter un élément

☑ Checklist

Masquer les tâches cochées

Supprimer

100%

☑

Les données s'importent et s'affichent correctement

☑

Les liens vers les autres pages pour les détails sont actifs et fonctionnels

☑

Les liens sont disponibles selon les profils

Ajouter un élément

☰

Activité

Afficher les détails

J

Écrivez un commentaire...

J

jhure

a ajouté cette carte à A faire

10 févr. 2025, 15:54

Rejoindre

Membres

Étiquettes

Checklist

Dates

Pièce jointe

Image de couverture

Champs personnalisés

Power-Ups

+ Ajouter des Power-ups

Automatisation

+

Ajouter un bouton

Actions

→ Déplacer

📄 Copier

📄 Miroir

📄 Créer un modèle

📄 Archiver

🔗 Partager

La carte « intégration et implémentation de la page Dashboard » avec ses checklists

104

2. Avantages

L'utilisation de Trello m'a été d'une aide précieuse tout au long du développement de l'application Optitop. Dès que j'ai commencé à l'utiliser, je me suis sentie plus sereine.

Visualiser toutes les tâches à faire m'a permis de prendre du recul et de mieux structurer mon travail.

J'ai pu :

- Lister précisément toutes les étapes importantes, y compris les petits bugs ou améliorations à faire plus tard,
- Me concentrer sur une tâche à la fois, sans avoir peur d'en oublier une autre,
- Revenir sur des idées ou problèmes que j'aurais pu oublier sans cette trace,
- Me libérer l'esprit : ne plus tout garder en tête, c'est aussi gagner en tranquillité mentale,

Et surtout, ressentir un vrai sentiment de progression chaque déplacement d'une carte dans la colonne "Terminé".

Avec tous les livrables de l'épreuve E6, c'était rassurant d'avoir un outil aussi performant sur lequel s'appuyer et avec la possibilité de pouvoir me réorganiser rapidement si besoin.

Ce tableau est devenu un outil central pour m'aider à garder le cap jusqu'à la fin du projet.