

Rapport de Mission : Optitop

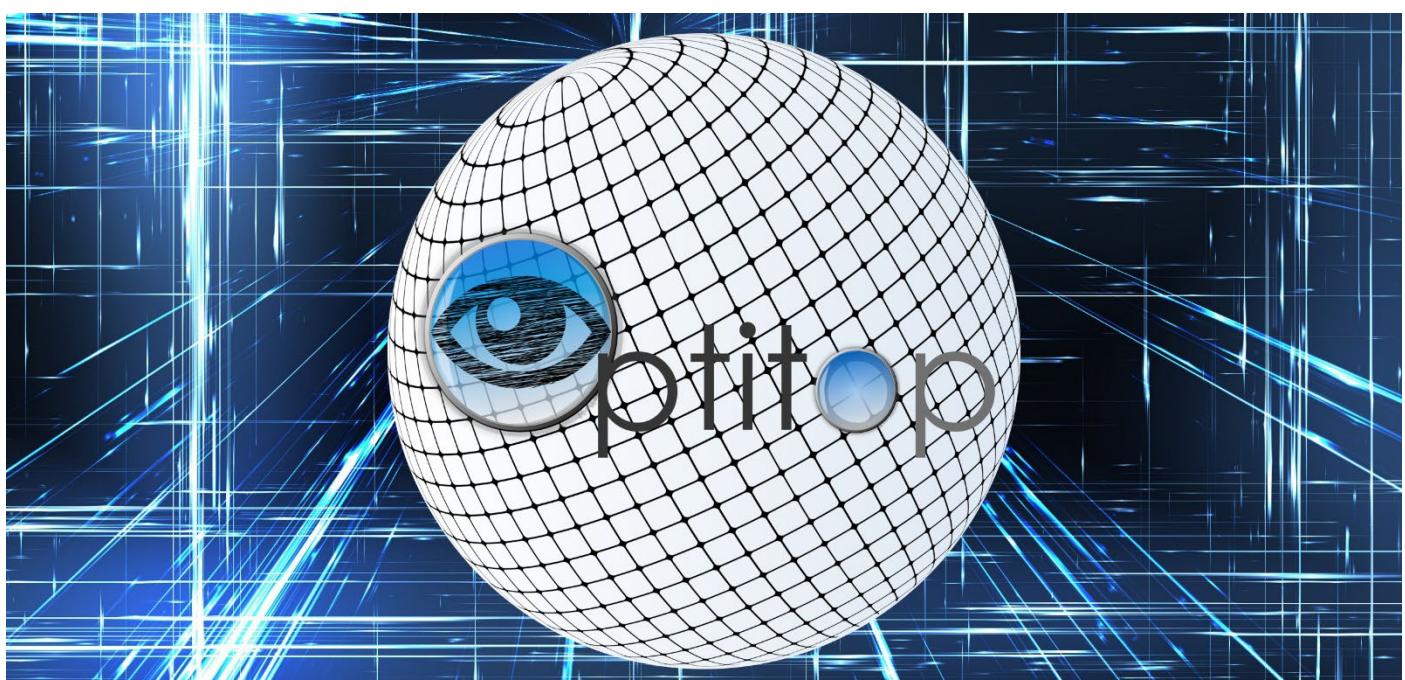


Table des matières

I. PREPARATION ET FRONTEND	5
A. Mockups de l'application web.....	5
1. Mockups.....	5
B. Création de l'interface utilisateur.....	5
1. IHM de l'application web.....	6
a) Page de connexion et Mot de passe oublié	6
b) La page Dashboard et son affichage selon l'utilisateur	6
➤ Admin, supermanager et manager	7
➤ Collaborateur	7
c) La page Mon compte	8
➤ L'onglet Profil.....	8
➤ L'onglet Password	9
d) La page Devis Optiques et son affichage selon l'utilisateur.....	9
➤ Admin, supermanager, manager	9
➤ Collaborateur	10
e) La page Paniers Moyens & Primes.....	11
f) La page Chiffre d'Affaires (CA).....	12
g) La page Synthèse	13
h) La page Exportation	14
i) La page Comptes.....	15
➤ Admin	15
➤ Supermanager et manager	16
C. Réalisation des modélisations de données.....	16
1. Diagrammes de Cas d'utilisation.....	16
a) Application Web	16
b) Application Mobile.....	17
2. Modèle Logique de données.....	18
D. Développement de l'application mobile	18
1. Mise en place de l'environnement Flutter	18
a) Installation de Flutter et configuration de VS Code	18
b) Installation Android Studio et SDK Android	20
c) Création d'un appareil virtuel Android (émulateur).....	22
d) Vérification de l'environnement avec Flutter Doctor.....	23
II. MISE EN PLACE DU BACKEND	25
A. Base de Données	25
1. Script de création	25
2. Insertion de données fictives.....	32
3. Déploiement de MySQL sur une machine virtuelle Ubuntu.....	32
a) Installation de MySQL.....	33
b) Sécurisation de MySQL	33
c) Vérification du service MySQL.....	34
d) Création de la Base de données Optitop	35
e) Création des tables	36
➤ Génération du fichier SQL par phpadmin sur Windows (environnement de développement).....	36
➤ Import du fichier SQL	36
f) Vérification des tables.....	36
g) Création d'un utilisateur MySQL dédié à l'API	37
h) Installation de phpMyAdmin sur Ubuntu	38
➤ Installation	38
➤ Test de fonctionnement	40
B. Définition des différents environnements.....	41
1. Mise en place des profils Spring Boot	41

2.	Développement (profil dev).....	42
3.	Pré-production (profil pprod)	42
a)	Paramétrage de l'environnement	42
b)	Configuration de MySQL sur la VM	43
➤	Modification du fichier de configuration	43
➤	Ouverture du pare-feu pour le port 3306	44
c)	Tests de fonctionnement	45
➤	Lancement de l'application pour pprod	45
➤	Requête à l'API depuis Postman.....	45
➤	Appel via le frontend web : test sur l'import d'un .csv.....	45
4.	Conclusion.....	47
C.	API REST Spring Boot.....	48
1.	L'API dans l'environnement de développement.....	48
a)	Installation.....	48
b)	Test du fonctionnement de l'API.....	49
III.	VERSIONNAGE.....	51
A.	Dépôt GitHub du projet.....	51
B.	Initialisation du dépôt Optitop.....	51
C.	Utilisation de Git et GitHub	52
IV.	IMPLEMENTATION	53
A.	Choix des langages	53
1.	JavaScript	53
2.	PHP	53
a)	Sécurisation de l'accès aux pages selon les rôles.....	53
b)	Gestion efficace des fichiers CSV volumineux.....	54
3.	Une complémentarité idéale entre PHP et Javascript	54
B.	Préparation de fichiers .csv exploitables	54
1.	Problématique.....	54
a)	Confidentialité des données.....	55
b)	Volume des données.....	55
c)	Simplification du jeu de données	55
2.	Script en Python	55
a)	Avantages	55
➤	Automatisation et gain de temps	55
➤	Facilité de manipulation des données avec Pandas	55
➤	Flexibilité.....	55
➤	Respect de la logique métier	56
b)	Installation des outils nécessaires	56
➤	Python et pandas	56
➤	Chardet	56
c)	Script	57
V.	SECURITE	59
A.	Chiffrement des mots de passe avec BCrypt	59
1.	Ajout des dépendances dans pom.xml.....	59
2.	Création du service de hashage	59
3.	Modification des Controllers	60
a)	Controller de changement de mot de passe.....	60
b)	Controller d'authentification.....	61
4.	Script de migration des mots de passe.....	62
B.	Gestion des mots de passe en cas d'ajout d'utilisateur.....	63

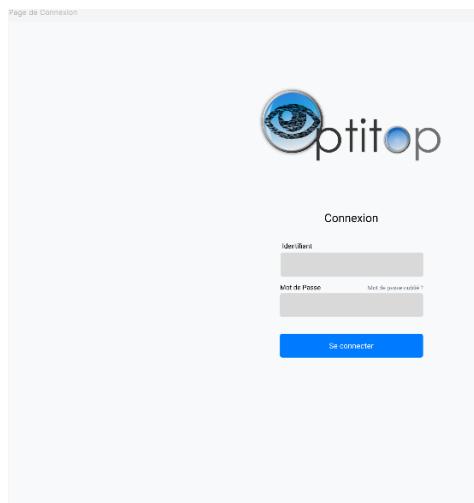
1.	Contexte.....	63
2.	Configuration SMTP pour Gmail	64
3.	Stockage des paramètres nécessaires en base de données.....	65
a)	Création de la table email_config.....	65
b)	Insertion des valeurs dans la table email_config	66
c)	Création de l'entité Springboot EmailConfig.....	66
4.	Modification des fichiers Service.java impliqués.....	67
a)	Création de EmailService.java	67
b)	Modification de PendingAccountService.java.....	68
5.	Test fonctionnel : Envoi d'un email à la création d'un compte utilisateur	68
a)	Objectif.....	68
b)	Mise en œuvre.....	68
c)	Résultat	69
VI. GESTION DE PROJET ET ORGANISATION		70
A.	Trello pour le suivi du projet	70
1.	Méthode Kanban.....	70
2.	Avantages.....	72

I. Préparation et Frontend

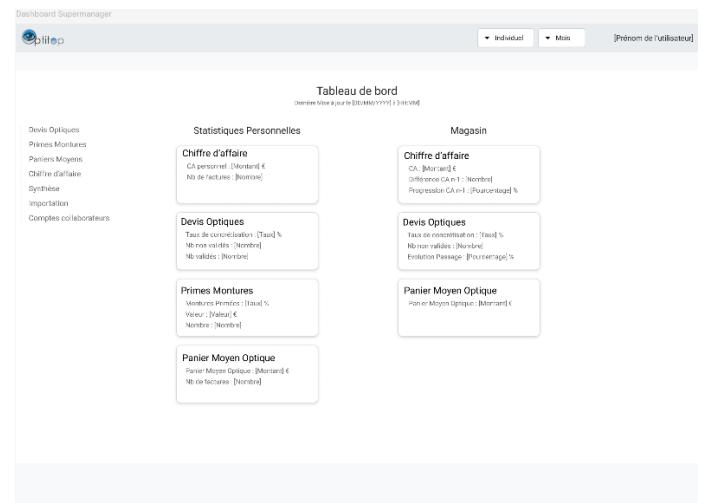
A. Mockups de l'application web

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. Crédit à 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) Page de connexion et Mot de passe oublié



Connexion

[Mot de passe oublié ?](#)

[Se connecter](#)



Mot de passe : Réinitialisation

Votre nouveau mot de passe vous sera transmis par email

[Envoyer](#)

Les pages de connexion et de mot de passe oublié

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

La page Dashboard constitue le point d'entrée de l'application après 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.

The screenshot shows the Optitop Dashboard interface. At the top, there is a header bar with the Optitop logo, a date selector ('Sélectionner une période'), and a user profile ('Joy'). Below the header, the word 'Dashboard' is displayed in blue. On the left, a sidebar menu lists navigation options: 'Dashboard', 'Devis Optiques', 'Paniers Moyens & Primes', 'CA', 'Synthèse', 'Exportation', and 'Comptes'. The main content area is divided into several sections: 'Magasin' (with a phone icon), 'Joy' (with a person icon), and four data cards. The 'Magasin' card shows a euro symbol and a minus sign. The 'Joy' card shows a person icon and the name 'Joy'. The four data cards provide specific statistics: 'Panier Moyen (opt) : -', 'Montures Primées : - Nombre : -', 'Panier Moyen (opt) : - Panier Moyen P2 : -', and 'Montures primées : - Prime Montures : -'. The bottom right corner of the dashboard displays the text 'Mis à jour le 17/04/2025 à 05:55'.

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

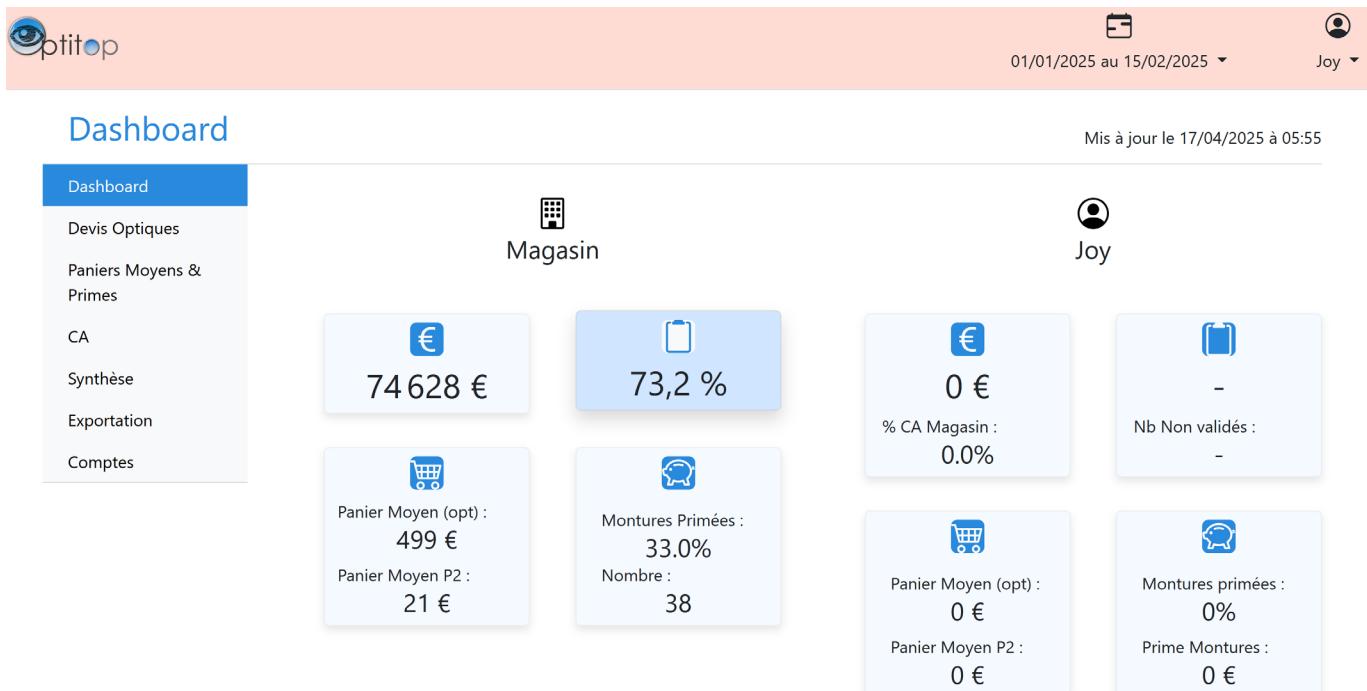
➤ Admin, supermanager et manager

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

 représentant le taux de concrétisation

 Cette carte est aussi un lien

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

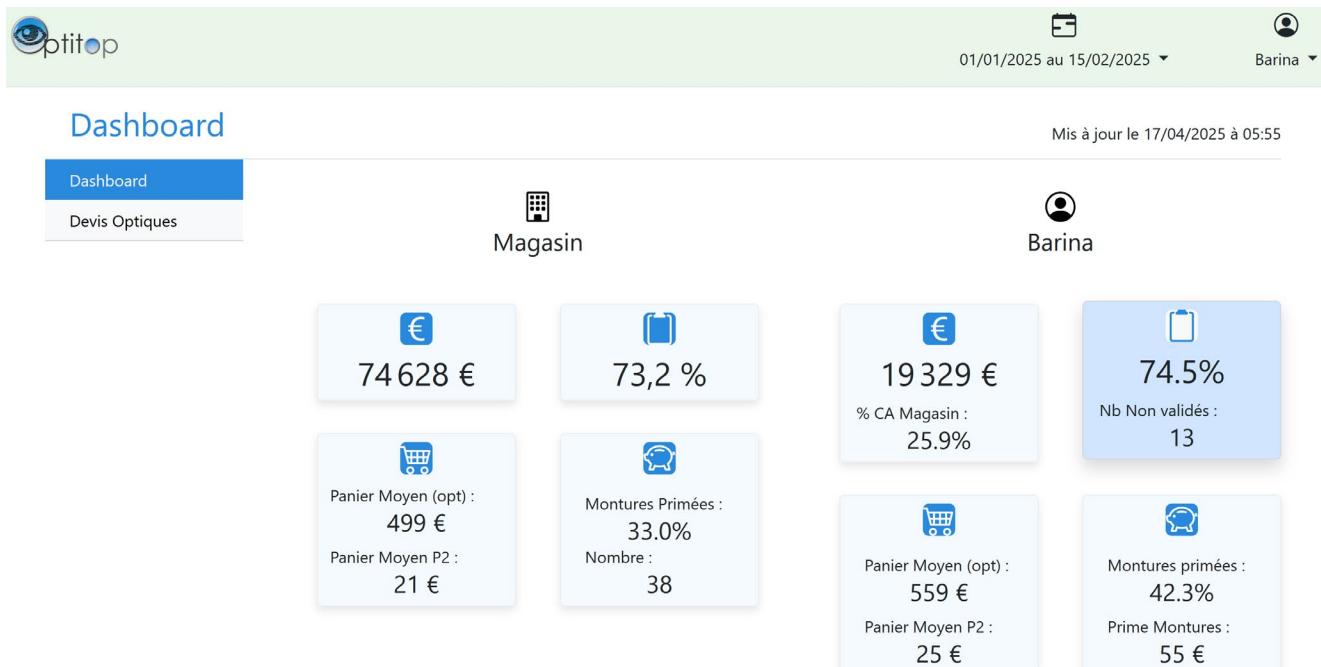


The screenshot shows the Admin Dashboard with the following data:

Section	Value	Details
Magasin	74 628 €	CA
Magasin	73,2 %	Taux de concrétisation
Panier Moyen (opt)	499 €	Panier Moyen (opt) : 499 €
Paniers Moyens & Primes	21 €	Panier Moyen P2 : 21 €
Montures Primées	33.0%	Montures Primées : 33.0%
Montures Primées	38	Nombre : 38
CA Magasin	0 €	% CA Magasin : 0.0%
Devis Optiques	-	Nb Non validés : -
Panier Moyen (opt)	0 €	Panier Moyen (opt) : 0 €
Paniers Moyens & Primes	0 €	Panier Moyen P2 : 0 €
Montures primées	0%	Montures primées : 0%
Montures primées	0 €	Prime Montures : 0 €

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

➤ Collaborateur



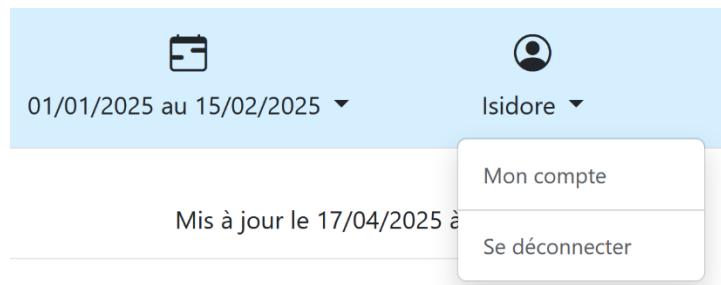
The screenshot shows the Collaborator Dashboard with the following data:

Section	Value	Details
Magasin	74 628 €	CA
Magasin	73,2 %	Taux de concrétisation
Panier Moyen (opt)	499 €	Panier Moyen (opt) : 499 €
Paniers Moyens & Primes	21 €	Panier Moyen P2 : 21 €
Montures Primées	33.0%	Montures Primées : 33.0%
Montures Primées	38	Nombre : 38
CA Magasin	19 329 €	% CA Magasin : 25.9%
Devis Optiques	74.5%	Nb Non validés : 13
Panier Moyen (opt)	559 €	Panier Moyen (opt) : 559 €
Paniers Moyens & Primes	25 €	Panier Moyen P2 : 25 €
Montures primées	42.3%	Montures primées : 42.3%
Montures primées	55 €	Prime Montures : 55 €

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

c) 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.

A screenshot of the "Mon compte" page under the "Profil" tab. On the left, there is a sidebar with links: Dashboard, Devis Optiques, Paniers Moyens & Primes, CA, Synthèse, Exportation, and Comptes. The main content area shows a summary card with a user icon, the name "Isidore Rau", the role "Manager", and the text "Depuis le 17/04/2025". To the right, there is a detailed "Détails du Profil" section with tables for "Profil" and "Password". The "Profil" table contains the following data:

Nom :	Rau
Prénom :	Isidore
Identifiant :	irau
Email :	jhure.dev@gmail.com
Création :	17/04/2025

The "Password" section is currently empty. At the top right of the main content area, it says "Mis à jour le 17/04/2025 à 05:55".

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

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 Optitop application's user profile page. At the top, there is a header with the Optitop logo, a date range from "01/01/2025 au 15/02/2025", and a user dropdown for "Isidore". Below the header, the page title is "Mon compte" and it was last updated "Mis à jour le 17/04/2025 à 05:55". On the left, a sidebar lists navigation options: Dashboard, Devis Optiques, Paniers Moyens & Primes, CA, Synthèse, Exportation, and Comptes. The main content area features a user profile card for "Isidore Rau" (Manager), showing a placeholder profile picture, the name, title, and the date "Depuis le 17/04/2025". To the right of the profile card is a "Password" section with three input fields labeled "Actuel", "Nouveau", and "Confirmation", each with a corresponding text input field. A blue button at the bottom right of this section says "Changer le mot de passe".

Section password qui permet de modifier son mot de passe

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)
- L'ajout d'un commentaire libre

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

Cela permet d'évaluer les performances de chacun.

Date	Nom	Client	Action	Commentaire
04/01/2025	EG	Thieuloy Lucas	Sélectionner	TEST
07/01/2025	HI	Jacquet Maelys	Sélectionner	
07/01/2025	BS	Poulain Louis	Sélectionner	
09/01/2025	IR	Mariano Jayden	Sélectionner	
09/01/2025	BS	Alarçon Elio	Sélectionner	
10/01/2025	EG	Matthias Baptiste	Sélectionner	
10/01/2025	IR	Mazaud Enora	Sélectionner	
10/01/2025	BS	Alliaume Lois	Sélectionner	
11/01/2025	BS	Moniri Léandre	Sélectionner	
11/01/2025	IR	Marquant Mariam	Sélectionner	
13/01/2025	HI	Manin Zoé	Sélectionner	
14/01/2025	EG	Meery Charline	Sélectionner	
15/01/2025	EG	Faure Isaac	Sélectionner	
16/01/2025	BS	Herrera Inès	Sélectionner	
16/01/2025	EG	Tessier Arya	Sélectionner	

La page Devis Optiques d'un admin s'apprêtant à enregistrer 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.

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	Arnaudy 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.

Paniers Moyens & Primes

Mis à jour le 17/04/2025 à 05:55

Dashboard

Devis Optiques

Paniers Moyens & Primes

CA

Synthèse

Exportation

Comptes

Paniers Moyens

Nom	PM	Nb factures	PM P1 Montures	PM P1 Verres	PM P2
BS	559.38€	31	173.11€	173.97€	24.63€
EG	450.06€	40	144.27€	136.69€	17.91€
HI	427.36€	28	134.70€	130.15€	29.06€
IR	556.00€	36	162.78€	174.85€	17.30€
Total	498.71€	135	154.19€	154.00€	21.37€

Primes Montures

Nom	Nb montures	Nb primées	%	Montant
BS	26	11	42.3%	55.00€
EG	36	11	30.6%	55.00€
HI	21	5	23.8%	25.00€
IR	32	11	34.4%	55.00€
Total	115	38	33.0%	190.00€

498.71€
Panier Moyen

 21.37€
Panier Moyen P2

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.

Chiffre d'Affaires

Mis à jour le 17/04/2025 à 05:55

Dashboard Devis Optiques Paniers Moyens & Primes CA Synthèse Exportation Comptes	📅 Année	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th>2025</th><th>Janv.</th><th>Fev.</th><th>Mars</th><th>Avril</th><th>Mai</th><th>Juin</th><th>Juill.</th><th>Août</th><th>Sept.</th><th>Oct.</th><th>Nov.</th><th>Déc.</th><th>Année</th></tr> </thead> <tbody> <tr> <td>CA</td><td>43 898 €</td><td>30 730 €</td><td>0 €</td><td>0 €</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>74 628 €</td></tr> <tr> <td>Delta n-1</td><td>+4 297 €</td><td>+10 548 €</td><td>Inconnu</td><td>Inconnu</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>+14 845 €</td></tr> <tr> <td>Delta %</td><td>+10,9 %</td><td>+52,3 %</td><td>Indéfini</td><td>Indéfini</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>À venir</td><td>-52,6 %</td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th>2024</th><th>Janv.</th><th>Fev.</th><th>Mars</th><th>Avril</th><th>Mai</th><th>Juin</th><th>Juill.</th><th>Août</th><th>Sept.</th><th>Oct.</th><th>Nov.</th><th>Déc.</th><th>Année</th></tr> </thead> <tbody> <tr> <td>CA</td><td>39 601 €</td><td>20 182 €</td><td>43 133 €</td><td>54 496 €</td><td>52 377 €</td><td>55 523 €</td><td>57 430 €</td><td>71 120 €</td><td>59 023 €</td><td>46 107 €</td><td>47 610 €</td><td>76 087 €</td><td>622 689 €</td></tr> <tr> <td>Delta n-1</td><td>+8 335 €</td><td>-21 948 €</td><td>-7 201 €</td><td>-301 €</td><td>+6 419 €</td><td>-5 167 €</td><td>-9 963 €</td><td>+18 935 €</td><td>+4 409 €</td><td>+151 €</td><td>+9 072 €</td><td>+18 010 €</td><td>+20 751 €</td></tr> <tr> <td>Delta %</td><td>+26,7 %</td><td>-52,1 %</td><td>-14,3 %</td><td>-0,5 %</td><td>+14,0 %</td><td>-8,5 %</td><td>-14,8 %</td><td>+36,3 %</td><td>+8,1 %</td><td>+0,3 %</td><td>+23,5 %</td><td>+31,0 %</td><td>+3,4 %</td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th>2023</th><th>Janv.</th><th>Fev.</th><th>Mars</th><th>Avril</th><th>Mai</th><th>Juin</th><th>Juill.</th><th>Août</th><th>Sept.</th><th>Oct.</th><th>Nov.</th><th>Déc.</th><th>Année</th></tr> </thead> <tbody> <tr> <td>CA</td><td>31 266 €</td><td>42 130 €</td><td>50 334 €</td><td>54 797 €</td><td>45 958 €</td><td>60 690 €</td><td>67 394 €</td><td>52 185 €</td><td>54 614 €</td><td>45 955 €</td><td>38 539 €</td><td>58 076 €</td><td>60 1938 €</td></tr> <tr> <td>Delta n-1</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td></tr> <tr> <td>Delta %</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td><td>Inconnu</td></tr> </tbody> </table>		2025	Janv.	Fev.	Mars	Avril	Mai	Juin	Juill.	Août	Sept.	Oct.	Nov.	Déc.	Année	CA	43 898 €	30 730 €	0 €	0 €	À venir	74 628 €	Delta n-1	+4 297 €	+10 548 €	Inconnu	Inconnu	À venir	+14 845 €	Delta %	+10,9 %	+52,3 %	Indéfini	Indéfini	À venir	-52,6 %		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	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 €	60 1938 €	Delta n-1	Inconnu	Delta %	Inconnu																																																
	2025	Janv.	Fev.	Mars	Avril	Mai	Juin	Juill.	Août	Sept.	Oct.	Nov.	Déc.	Année																																																																																																																																																																		
CA	43 898 €	30 730 €	0 €	0 €	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	74 628 €																																																																																																																																																																		
Delta n-1	+4 297 €	+10 548 €	Inconnu	Inconnu	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	+14 845 €																																																																																																																																																																		
Delta %	+10,9 %	+52,3 %	Indéfini	Indéfini	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	À venir	-52,6 %																																																																																																																																																																		
	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	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 €	60 1938 €																																																																																																																																																																			
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																																																																																																																																																																			

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.

Synthèse

Mis à jour le 17/04/2025 à 05:55

- [Dashboard](#)
- [Devis Optiques](#)
- [Paniers Moyens & Primes](#)
- [CA](#)
- [Synthèse](#)
- [Exportation](#)
- [Comptes](#)

Synthèse

Chiffres du Magasin

Du 01/01/2025 au 15/02/2025				Mis à jour le 17/04/2025 à 05:55
Catégorie	CA Facturé	Delta CA	Delta CA %	Taux Concret
N	74 628 €	+15 870 €	+27,0 %	73,2 %
N-1	58 758 €			76,5 %

Statistiques Collaborateurs

Nom	Primes		P2		Devis		CA
	Prime Monture	% Mont. primées	Nombre	Panier Moyen	Nombre	Concrétisation	
BS	55 €	42.3%	61	25 €	51	74,5 %	19 329 €
EG	55 €	30.6%	66	18 €	65	70,8 %	19 728 €
HI	25 €	23.8%	46	29 €	48	66,7 %	13 655 €
IR	55 €	34.4%	80	17 €	49	81,6 %	21 917 €
Total	190 €	33.0%	253	21 €	213	73,2 %	74 628 €

La page Synthèse

h) La page Exportation

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, ces données 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)

Exportation

Mis à jour le 17/04/2025 à 05:55

- [Dashboard](#)
- [Devis Optiques](#)
- [Paniers Moyens & Primes](#)
- [CA](#)
- [Synthèse](#)
- [Exportation](#)
- [Comptes](#)



Exportation des données de vente

Choisir un fichier CSV

 Parcourir... Aucun fichier sélectionné.

La page Exportation

i) 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

Comptes Utilisateurs

Nouvelle demande

Demande ▾ Nom Prénom Email Rôle Identifiant Envoyer

Ajouter Modifier Supprimer cours ▾

Choix disponibles pour une demande concernant un compte utilisateur

➤ Admin

Optitop 01/01/2025 au 15/02/2025 Joy

Mis à jour le 17/04/2025 à 05:55

Comptes

Dashboard Devis Optiques Paniers Moyens & Primes CA Synthèse Exportation Comptes

Comptes Utilisateurs

Nouvelle demande

Demandes en cours ▾

Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
18/04/2025	jhure	Stud	Barina	bstud	collaborator	jhure.dev@gmail.com	suppression	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

Page Comptes de l'admin : il peut valider ou refuser une demande

➤ Supermanager et manager

Optitop

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

Comptes Mis à jour le 17/04/2025 à 05:55

Dashboard
Devis Optiques
Paniers Moyens & Primes
CA
Synthèse
Exportation
Comptes

Comptes Utilisateurs

+ Nouvelle demande

Demande Nom Prénom Email Rôle Identifiant Envoyer

Demandes en cours

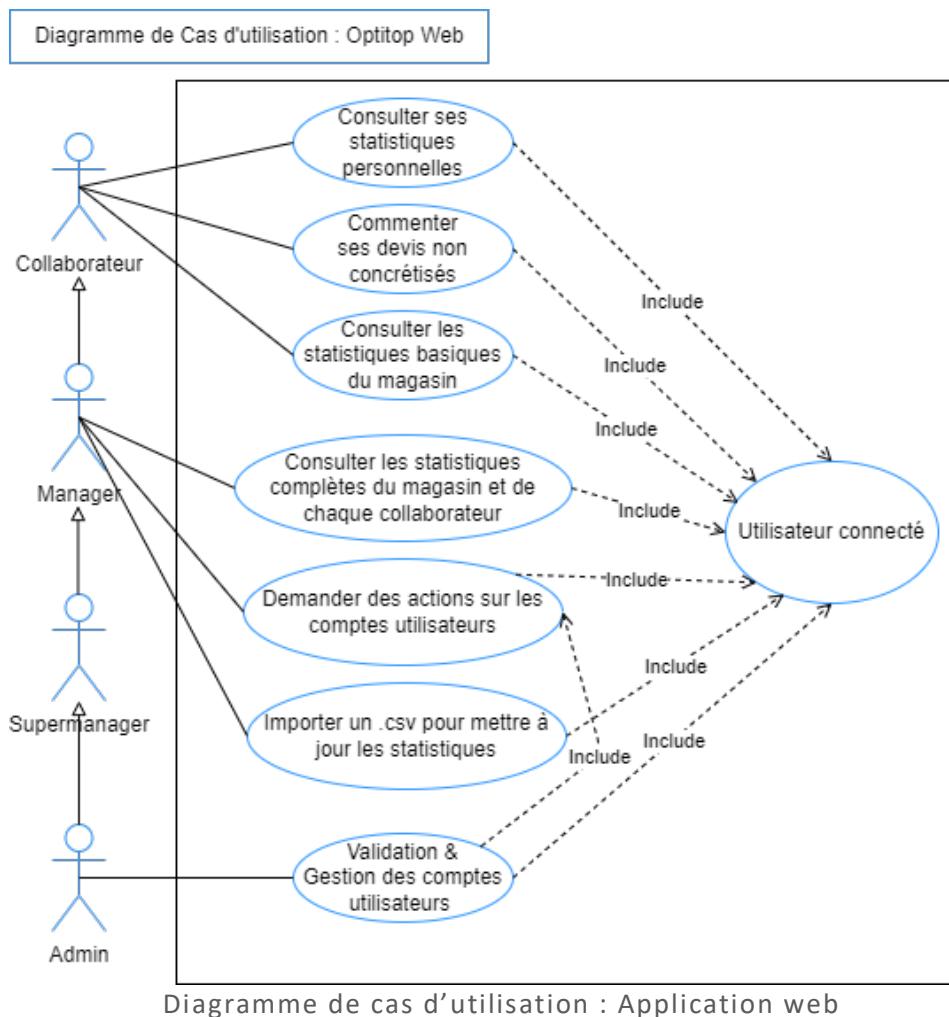
Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
18/04/2025	jhere	Stud	Barina	bstud	collaborator	jhere.dev@gmail.com	suppression	Valider Refuser

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

C. Réalisation des modélisations de données

1. Diagrammes de Cas d'utilisation

a) Application Web



b) Application Mobile

Diagramme de Cas d'utilisation : Optitop Mobile

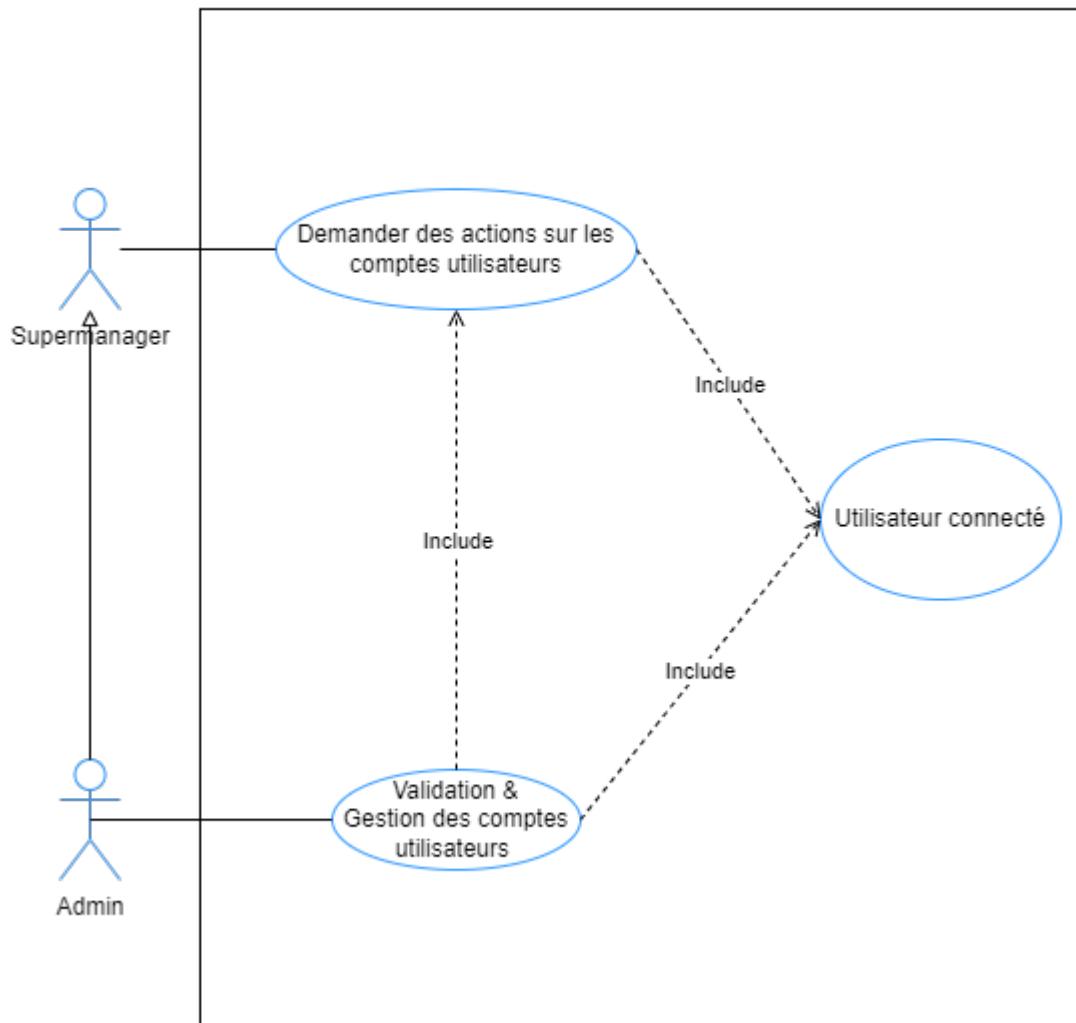
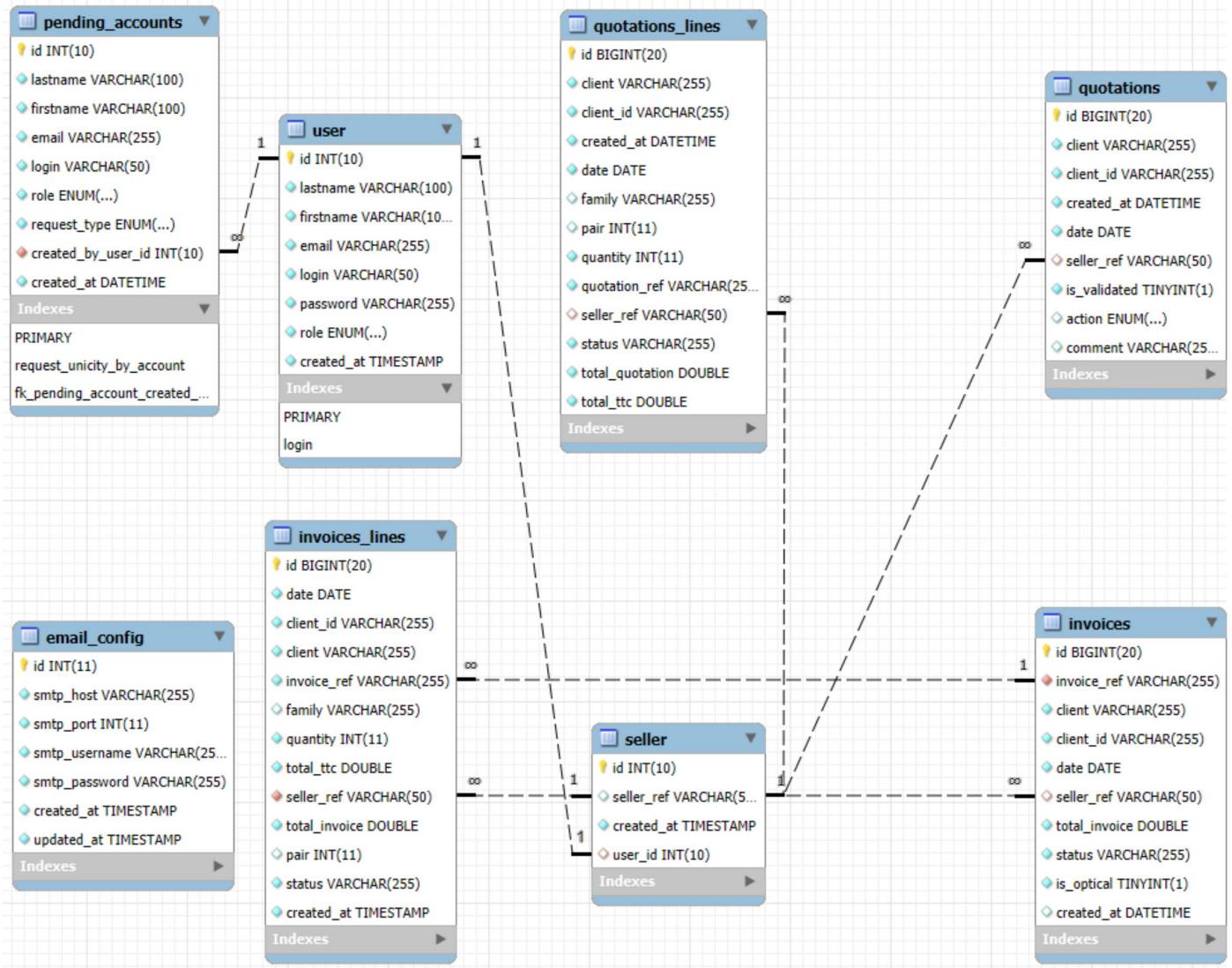


Diagramme de cas d'utilisation : Application web

2. Modèle Logique de données



Modèle logique de données de la base de données optitop

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**.

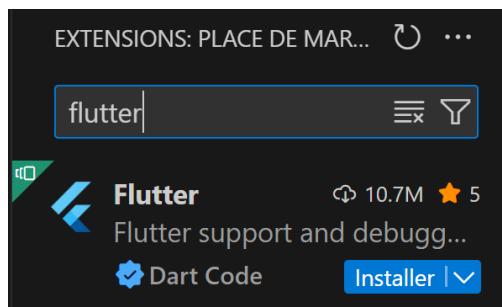
This option sends you the Flutter install page if you have not installed Git for Windows as directed in the [development tools prerequisites](#).

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

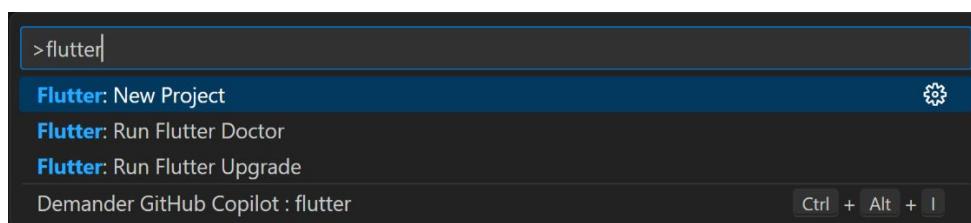
Les étapes de configuration ont été les suivantes :

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



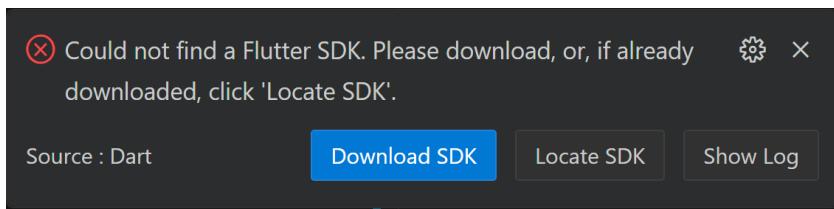
Installation de l'extension VS Code Flutter

- 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](#).



1. Launch **Android Studio**.

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

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

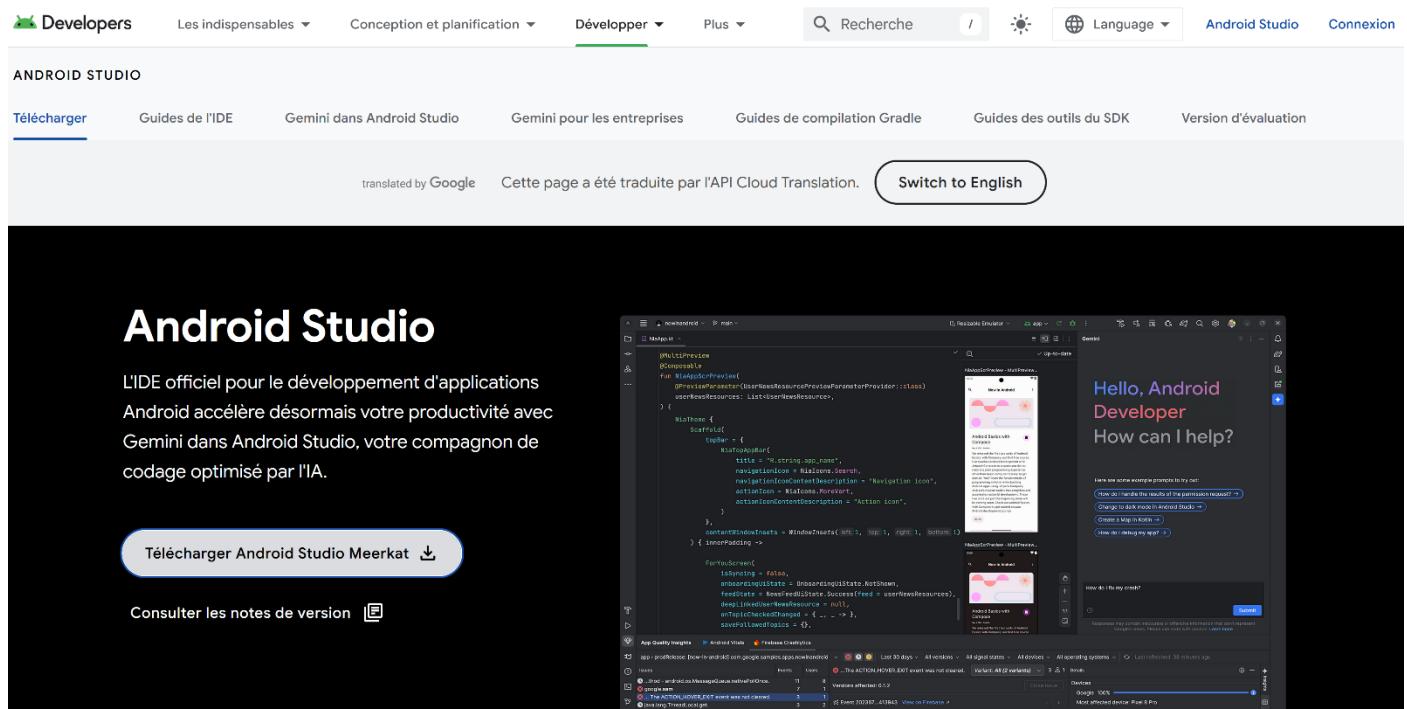
3. Install the following components:

- **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

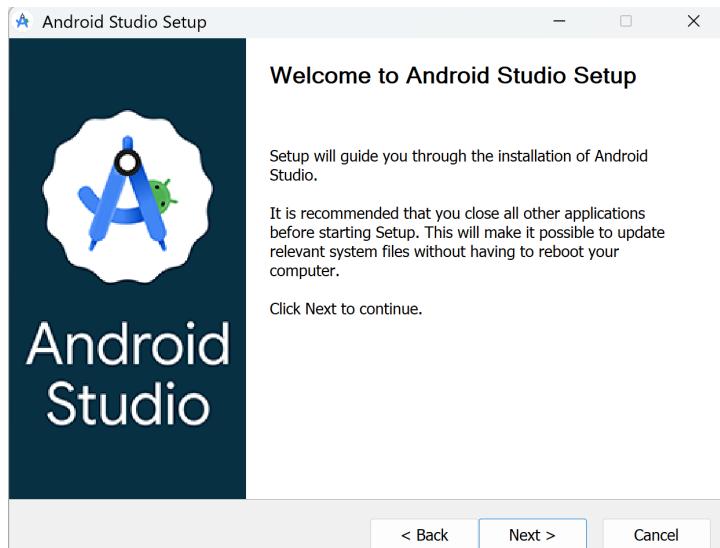
Étapes :

- Téléchargement d'Android Studio depuis le site officiel

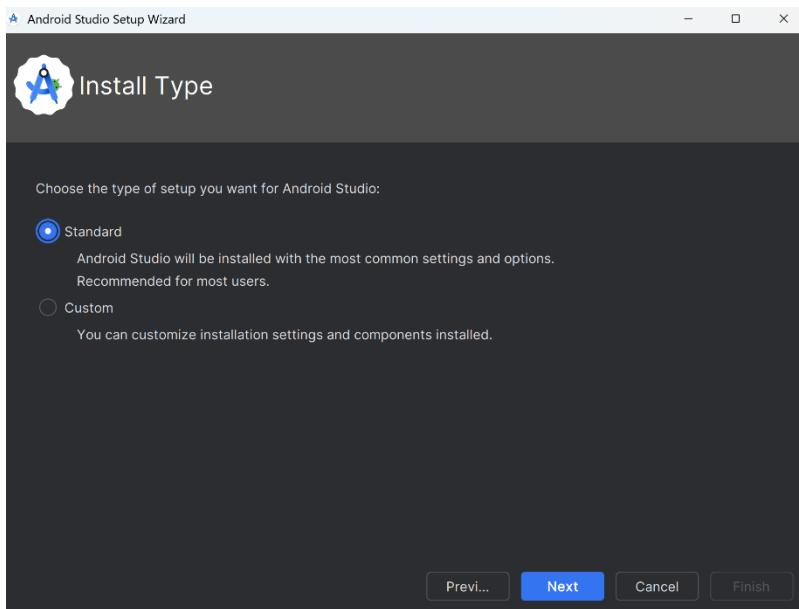


Site officiel d'Android Studio pour le téléchargement

- Installation pas à pas via l'assistant de configuration

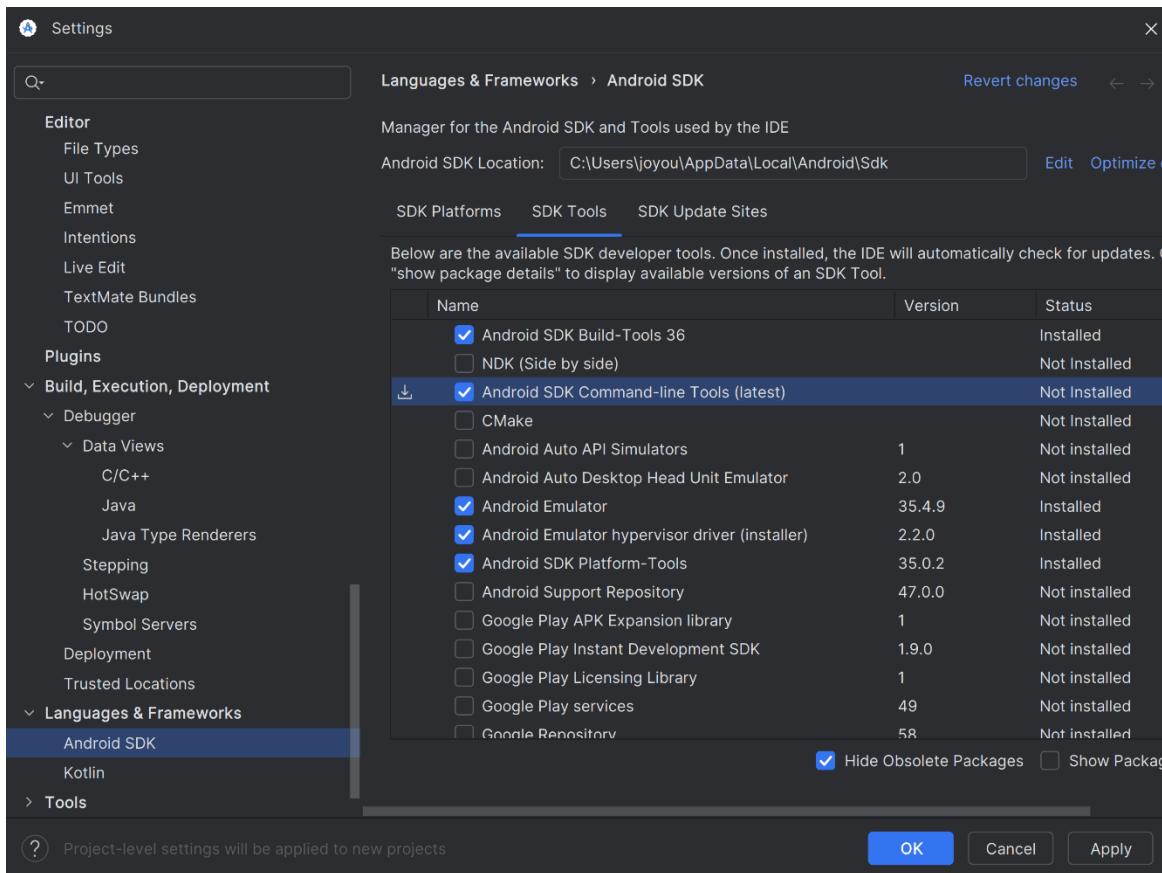


Écran de bienvenue



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

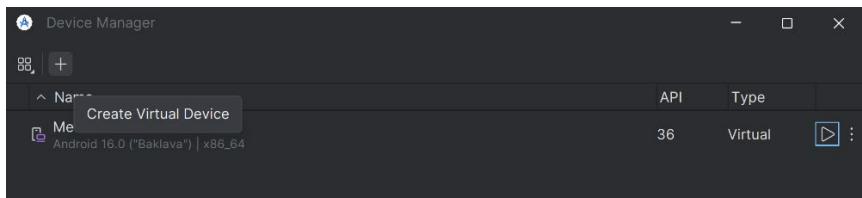


Sélection des SDK Tools à installer

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

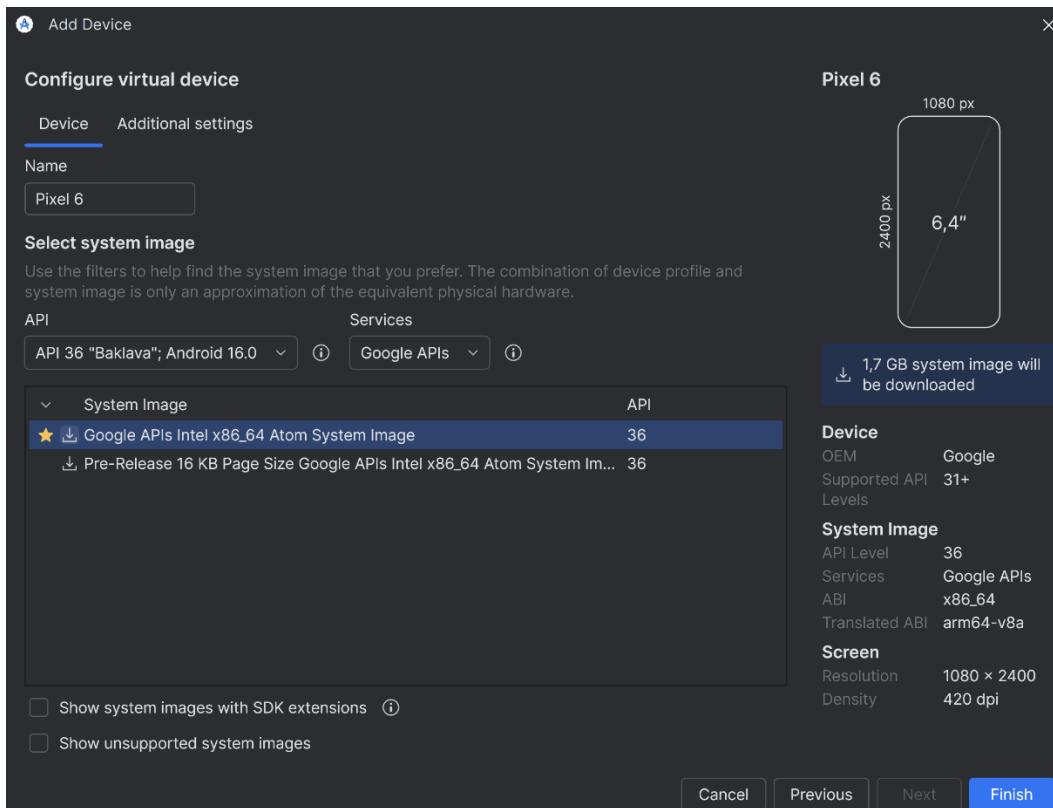
Après avoir configuré le SDK Android, j'ai procédé à la création d'un appareil virtuel Android afin de pouvoir tester mon application mobile directement depuis mon environnement de développement.

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
[✗] Visual Studio - develop Windows apps
    ✗ 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**.



II. 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_RETOUR','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;

```

Script de création des tables pending_account, quotations et quotations_lines

```
-- 
-- 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;
```

```

-- Constraintes pour les tables déchargées
--

-- Constraintes 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;

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

-- Constraintes 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;

-- Constraintes 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;

-- Constraintes 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;

-- Constraintes 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 priviléges.

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.
~
~
~
```

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 phpadmin sur Windows (environnement de développement)

The screenshot shows the phpMyAdmin interface for exporting tables from a database named 'optitop'. The 'Tables' section lists eight tables: email_config, invoices, invoices_lines, pending_accounts, quotations, quotations_lines, seller, and user. Each table has three checkboxes next to it; the first two columns are checked, while the third is not. The 'Format' section is set to 'SQL'. The 'Sortie' section contains two unchecked options: 'Renommer les bases de données/tables/colonnes exportées' and '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.

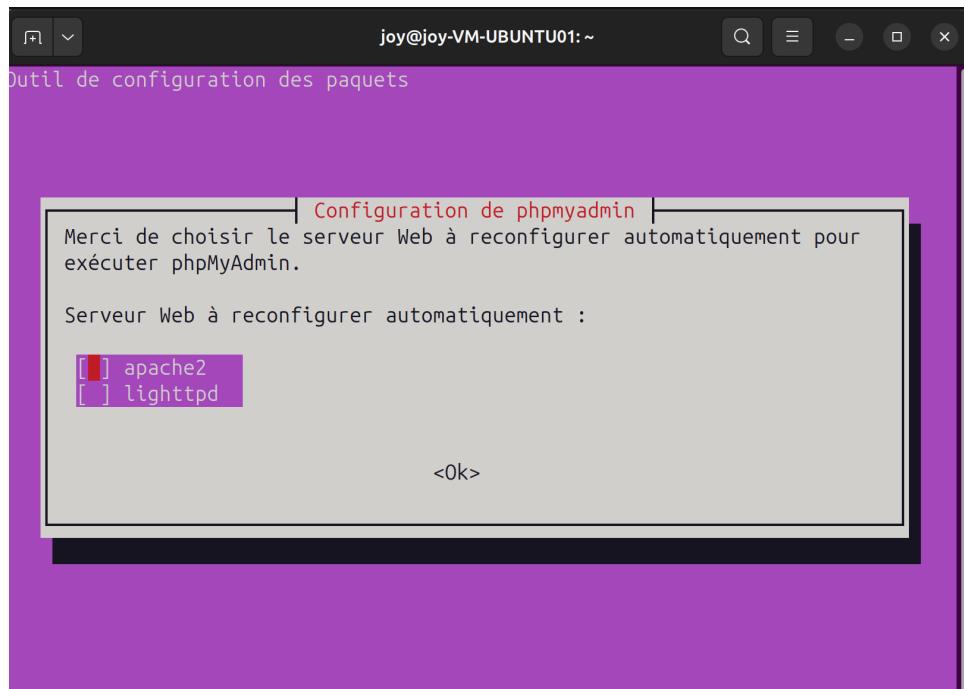
h) Installation de phpMyAdmin sur Ubuntu

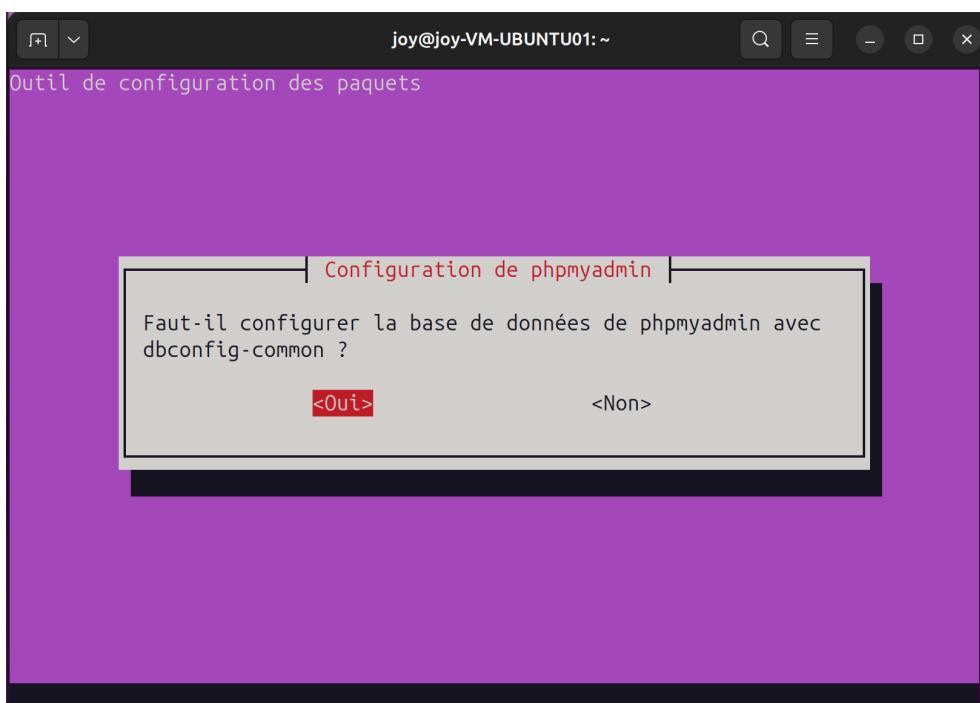
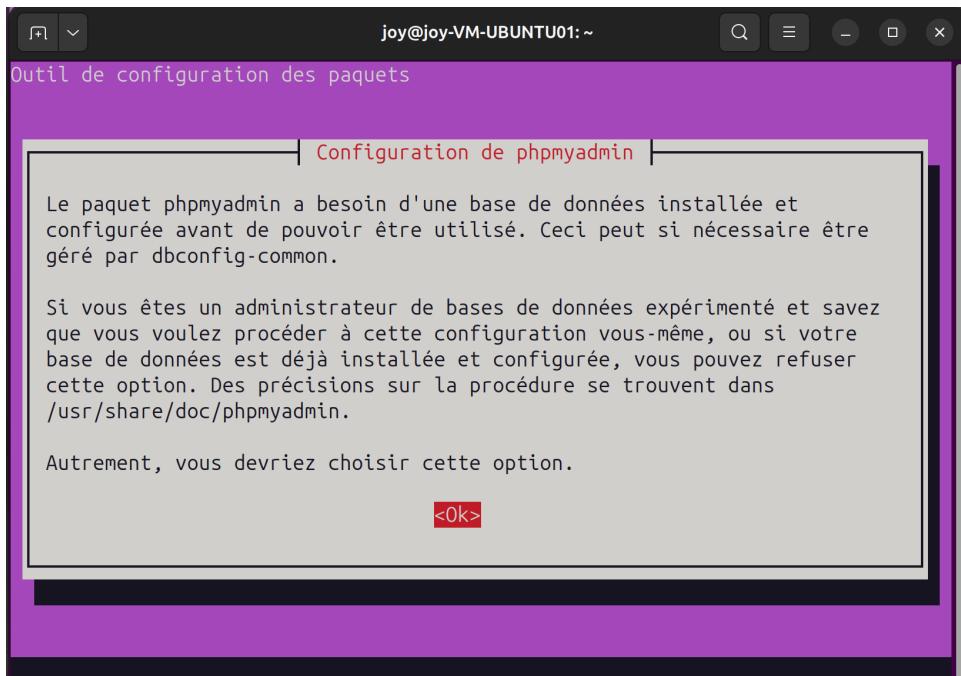
J'ai choisi d'installer **phpMyAdmin** sur ma VM Ubuntu. C'est une interface web légère et intuitive permettant d'administrer MySQL directement depuis un navigateur. Ce choix me permet de manipuler les tables, visualiser les données, exécuter des requêtes SQL et suivre l'évolution des statistiques sans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

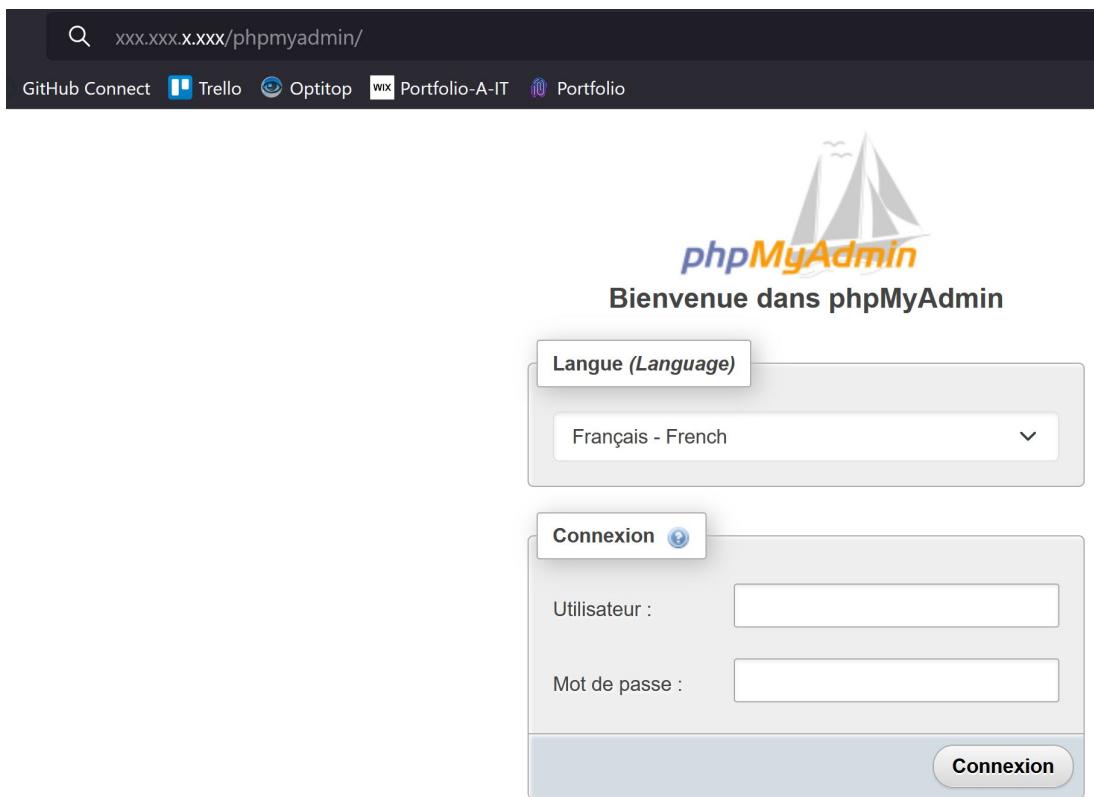






Les étapes de l'installation où différents choix sont proposés

➤ Test de fonctionnement



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.

B. 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 :

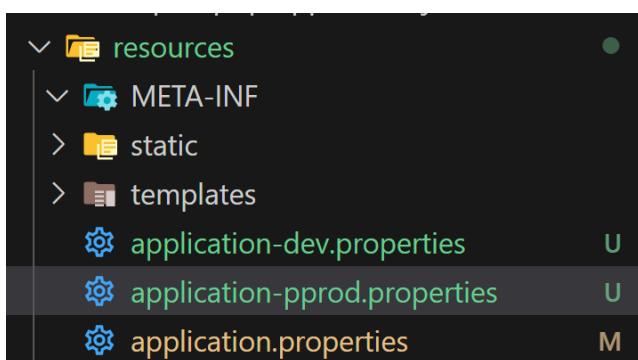
- **application-dev.properties** pour le développement
- **application-pprod.properties** pour la préproduction

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 dossier 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=XXXXXXXX

# 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 rechargeé

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, 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"
```

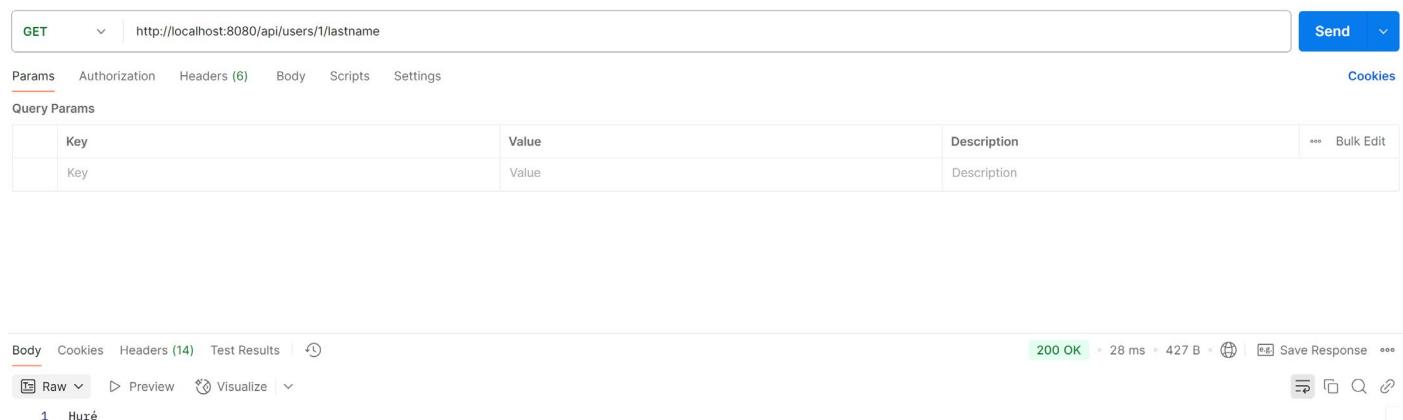
Lancement de l'application en pprod à partir du terminal de VS Code

```
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.



The screenshot shows the Postman interface with a successful GET request to `http://localhost:8080/api/users/1/lastname`. The response is `200 OK` with `28 ms` latency and `427 B` size. The response body contains the name `Hurié`.

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 :

➤ Exportation des données de vente

Choisir un fichier CSV

Parcourir...

test_20250214.csv

test_20250214.csv

Importer

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 !

Table : invoices_lines															
	id	date	client_id	client	invoice_ref	family	quantity	total_ttc	seller_ref	total_invoice	pair	status	created_at		
<input type="checkbox"/>	Éditer	Copier	Supprimer	630	2023-01-02	103007619	Marron Owen	F103056577	CLI	1	29	bstud	29	NULL facture	2025-04-17 05:32:41
<input type="checkbox"/>	Éditer	Copier	Supprimer	631	2023-01-02	103011989	Jacquin Lino	F103056579	DIV	1	40	irau	40	NULL facture	2025-04-17 05:32:41
<input type="checkbox"/>	Éditer	Copier	Supprimer	632	2023-01-02	103017425	Mazoiller Albane	F103056582	SOL	1	29	irau	29	NULL facture	2025-04-17 05:32:41
<input type="checkbox"/>	Éditer	Copier	Supprimer	633	2023-01-03	102018267	Maurelès Capucine	F103056586	MON	1	30	bstud	210	1 facture	2025-04-17 05:32:41
<input type="checkbox"/>	Éditer	Copier	Supprimer	634	2023-01-03	102018267	Maurelès Capucine	F103056586	VER	1	90	bstud	210	1 facture	2025-04-17 05:32:41
<input type="checkbox"/>	Éditer	Copier	Supprimer	635	2023-01-03	102018267	Maurelès Capucine	F103056586	VER	1	90	bstud	210	1 facture	2025-04-17 05:32:41
<input type="checkbox"/>	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.).

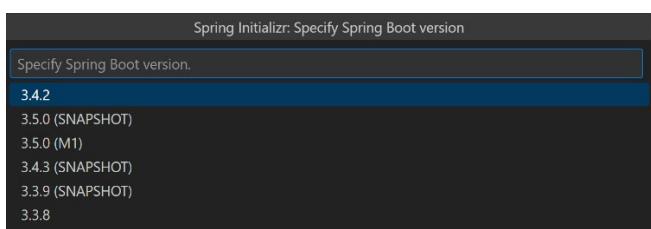
C. API REST Spring Boot

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.

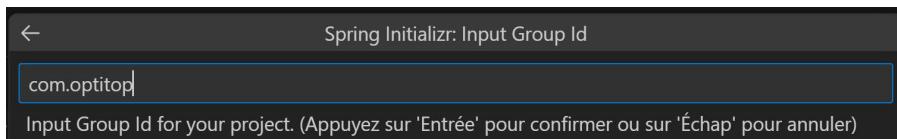
1. L'API dans l'environnement de développement

a) Installation

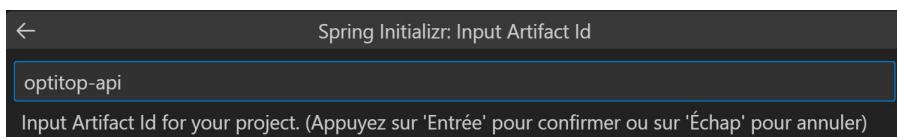
J'ai installé mon API depuis VSCode et l'extension **spring initializr** :



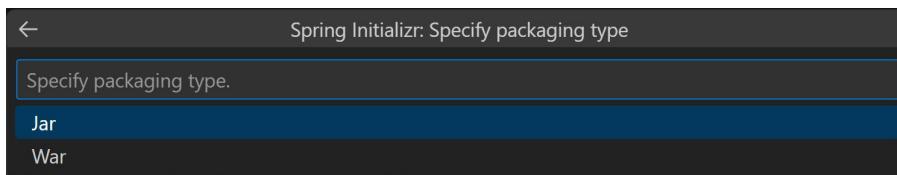
Sélection de la version Spring Boot



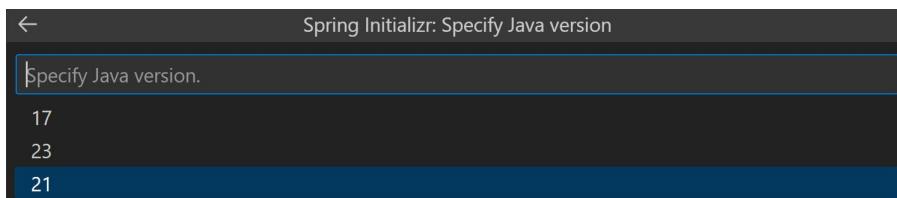
Définition de l'input « Group Id »



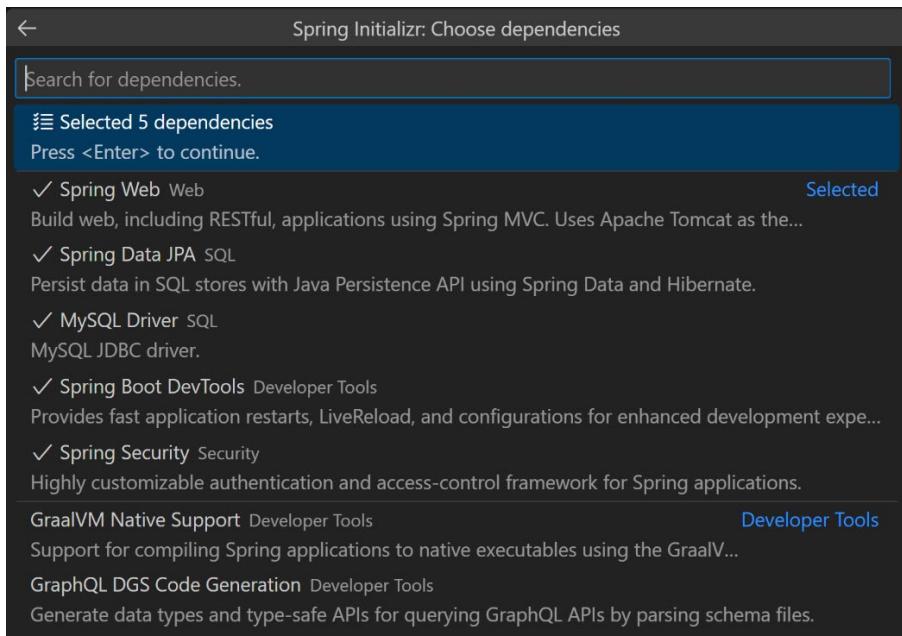
Définition de l'Artifact Id



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 crée 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 ».

The screenshot shows the Postman interface with the following details:

- Request URL:** GET http://localhost:8080/test
- Method:** GET
- Auth Type:** No Auth
- Headers:** (7)
- Body:** (Raw, Preview, Visualize)
- Response Status:** 200 OK
- Response Time:** 75 ms
- Response Size:** 365 B
- Content:** L'API fonctionne correctement !

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

III. Versionnage

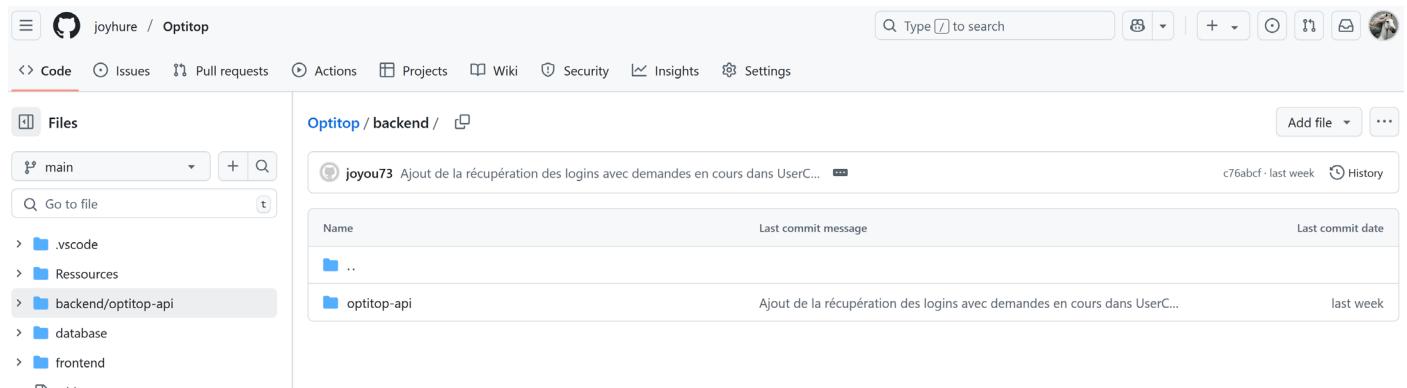
A. Dépôt GitHub du projet

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.



The screenshot shows the GitHub repository interface for 'Optitop'. The left sidebar lists project files: main, .vscode, Ressources, backend/optitop-api, database, frontend, and .gitignore. The main area displays a single commit from user 'joyou73' with the message 'Ajout de la récupération des logins avec demandes en cours dans UserC...'. The commit was made last week and has a history link. The commit table includes columns for Name, Last commit message, and Last commit date.

Name	Last commit message	Last commit date
..		
optitop-api	Ajout de la récupération des logins avec demandes en cours dans UserC...	last week

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

The screenshot shows a GitHub repository interface. On the left, there's a sidebar with various branches and their status (e.g., 'feat: Suppress...', '@pprod'). The main area displays a list of commits from 'feat-csvprepare.py' to 'feat-bdd'. A specific commit by 'joyou73' is highlighted, showing its details: suppression of file 'id&pass' due to security reasons, with 6 suppressed lines. Below this, a terminal window shows the command 'git merge 876c96a..e3f106d' followed by 'Fast-forward' and a list of changes (1 file changed, 6 deletions). The bottom of the terminal shows the path 'PS C:\Users\joyou\Informatique\Projets\Optitop>'.

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

IV. Implémentation

A. Choix des langages

1. JavaScript

J'ai choisi **JavaScript** pour gérer la logique d'optimiser l'expérience utilisateur et de 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 (ou son token) 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, fluidifier l'expérience utilisateur et améliorer la sécurité lors de l'authentification. 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

b) Gestion efficace des fichiers CSV volumineux

Mon application doit permettre l'importation de fichiers CSV contenant plus de 40 000 lignes. JavaScript seul ne peut pas gérer efficacement des fichiers de cette taille, car il fonctionne côté client, ce qui pose plusieurs problèmes.

Avantages de PHP :

- **Capacité à traiter de gros fichiers** : PHP peut lire et traiter un fichier ligne par ligne sans tout charger en mémoire, contrairement à JavaScript qui pourrait causer un crash du navigateur.
- **Sécurité et validation des données** : Avec PHP, chaque ligne peut être vérifiée et nettoyée avant son insertion en base de données.
- **Gestion de la mémoire côté serveur** : Les limites du navigateur en terme de mémoire ne sont plus un frein puisqu'avec PHP la mémoire nécessaire sera gérée côté serveur.

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, le contrôle d'accès et la gestion des fichiers volumineux 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. 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("\nEnterz 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
if "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)}")

```

Script python de création de données fictives

V. Sécurité

A. Chiffrement des mots de passe avec BCrypt

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 hashage

```
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

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 »

	<input type="button" value="← T →"/>		<input type="button" value="id"/>	<input type="button" value="seller_ref"/>	<input type="button" value="created_at"/>	<input type="button" value="user_id"/>
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	146 atest	2025-04-06 16:47:49	NULL
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	147 egaligai	2025-04-06 16:47:49	9
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	148 btest	2025-04-06 16:47:49	NULL
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	149 bstud	2025-04-06 16:47:49	10
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	150 hiskander	2025-04-06 16:48:16	8
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="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

Demandes en cours ▾

Date	Initiateur	Nom	Prénom	Login	Rôle	Email	Demande	Action
12/04/2025	jhure	GALIGAI	Eden	egaligai	collaborator	egaligai@example.fr	suppression	<input type="button" value="Valider"/> <input type="button" value="Refuser"/>
12/04/2025	jhure	RAU	Isidore	irau	collaborator	iroas@example.fr	modification	<input type="button" value="Valider"/> <input type="button" value="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.

	<input type="button" value="← T →"/>	<input type="button" value="id"/>	<input type="button" value="lastname"/>	<input type="button" value="firstname"/>	<input type="button" value="email"/>	<input type="button" value="login"/>	<input type="button" value="role"/>	<input type="button" value="request_type"/>	<input type="button" value="created_by_user_id"/>	<input type="button" value="created_at"/>
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	20 RAU	Isidore	iroas@example.fr	irau	collaborator	modification	5 2025-04-12 11:51:51
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="Supprimer"/>	21 GALIGAI	Eden	egaligai@example.fr	egaligai	collaborator	suppression	5 2025-04-12 11:52:01
<input type="checkbox"/>	<input type="button" value="Éditer"/>	<input type="button" value="Copier"/>	<input type="button" value="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.

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) Crédation 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.findFirstByIdDesc();
        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.findFirstByIdDesc();

        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			Administrateur	jhure.dev@gmail.com	ajout	<button>Valider</button> <button>Refuser</button>
12/04/2025	jhure				Administrateur	egaligai@example.fr	suppression	<button>Valider</button> <button>Refuser</button>
12/04/2025	jhure				Administrateur	iroas@example.fr	modification	<button>Valider</button> <button>Refuser</button>

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.lg8u/.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\$uiRQFF9WPSI6hYCDltlAiumcrGIFnwxDR32xPMnJnml...	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/z6ev42cu9hfnNbwmwO43NbfV...	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 ×



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.

VI. 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'aimeraï 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

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 + ⏪ Suivre

Rejoindre

Membres

Étiquettes

Checklist

Dates

Pièce jointe

Image de couverture

Champs personnalisés

Description

Ajouter une description plus détaillée...

Prérequis

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

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

J Écrivez un commentaire...

J **jhure** a ajouté cette carte à A faire
10 févr. 2025, 15:54

Afficher les détails

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

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 toutes les livrables de l'épreuve E6, c'était rassurant d'avoir un outil aussi performant sur lequel s'appuyer et 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.