



# Rapport du Projet : Gestion de Bibliothèque

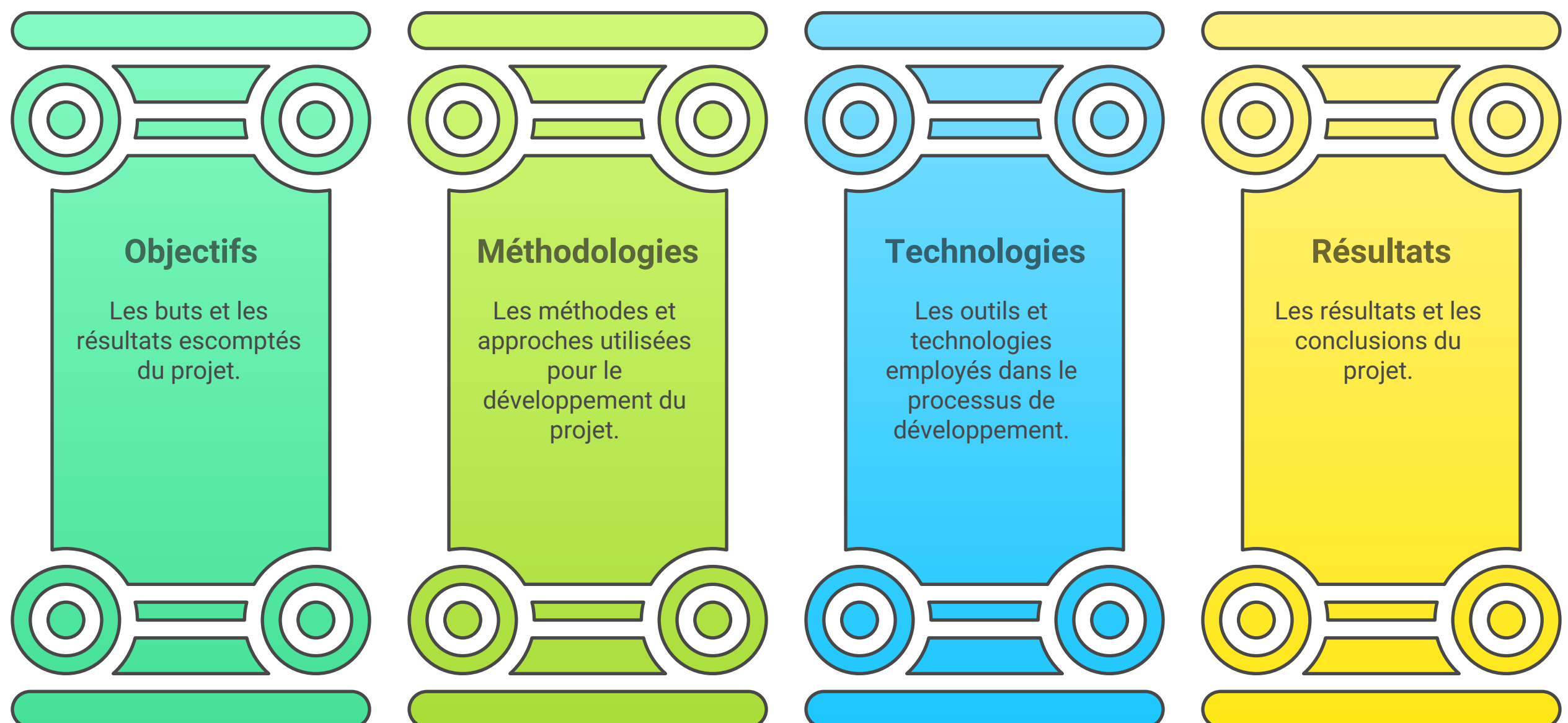
**Réalisé par :** Dyhia Berkhouche & Sarah Ould Fella & Salah Mouhand kaci

**Date :** 20 MARS 2025

## 1. Introduction

Le présent document constitue un rapport détaillé du projet de développement d'une application web de gestion d'une bibliothèque, exposant clairement les objectifs, méthodologies, technologies employées et résultats obtenus.

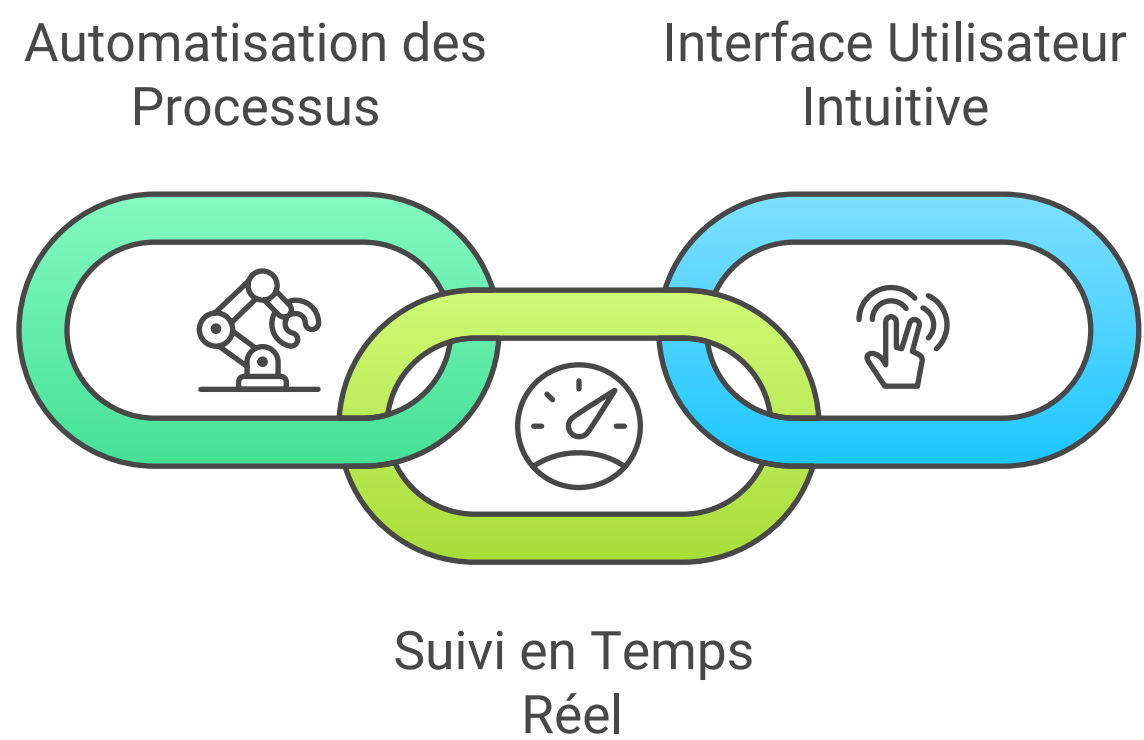
### Rapport Complet sur le Développement d'une Application Bibliothèque



## 2. Objectifs du Projet

- Automatiser et simplifier les processus de gestion d'adhérents, de livres et d'emprunts.
- Assurer un suivi efficace et en temps réel des opérations de prêt.
- Fournir une interface utilisateur intuitive et réactive.

Optimisation de la Gestion Bibliothèque par  
Technologie Avancée



3. Analyse des Besoins

- Gestion centralisée des données des adhérents.
- Suivi en temps réel des livres disponibles et empruntés.
- Outils permettant l'ajout manuel ou en masse des documents.
- Dashboard récapitulatif des statistiques importantes.

Optimisation de la Gestion Bibliothécaire par des Outils Numériques



Gestion des  
Adhérents

Permet une gestion centralisée des données des adhérents pour une organisation efficace.



Suivi des  
Livres

Facilite le suivi en temps réel des livres disponibles et empruntés.



Outils d'Ajout  
de Documents

Fournit des outils pour ajouter des documents manuellement ou en masse.



Tableau de  
Bord  
Statistique

Offre un tableau de bord récapitulatif pour des statistiques importantes.

## 4. Conception Technique

### 4.1 Front-End

- HTML5, CSS3 : structuration et style cohérent sur toutes les pages.
- JavaScript : interactivité et dynamisme côté client avec utilisation d'AJAX pour appels API.

### 4.2 Back-End

- Python avec Flask : Création d'une API REST robuste et sécurisée.
- SQLite : Base de données légère et efficace pour stocker les données.

### Création d'une Infrastructure Web Dynamique et Sécurisée



#### Structuration et style cohérent

Utilisation de HTML5  
et CSS3 pour une  
présentation web  
uniforme et  
attrayante.



#### Interactivité côté client

Mise en œuvre de  
JavaScript et AJAX  
pour une expérience  
utilisateur  
dynamique.



#### API sécurisée

Développement  
d'une API sécurisée  
et fiable avec Python  
et Flask.



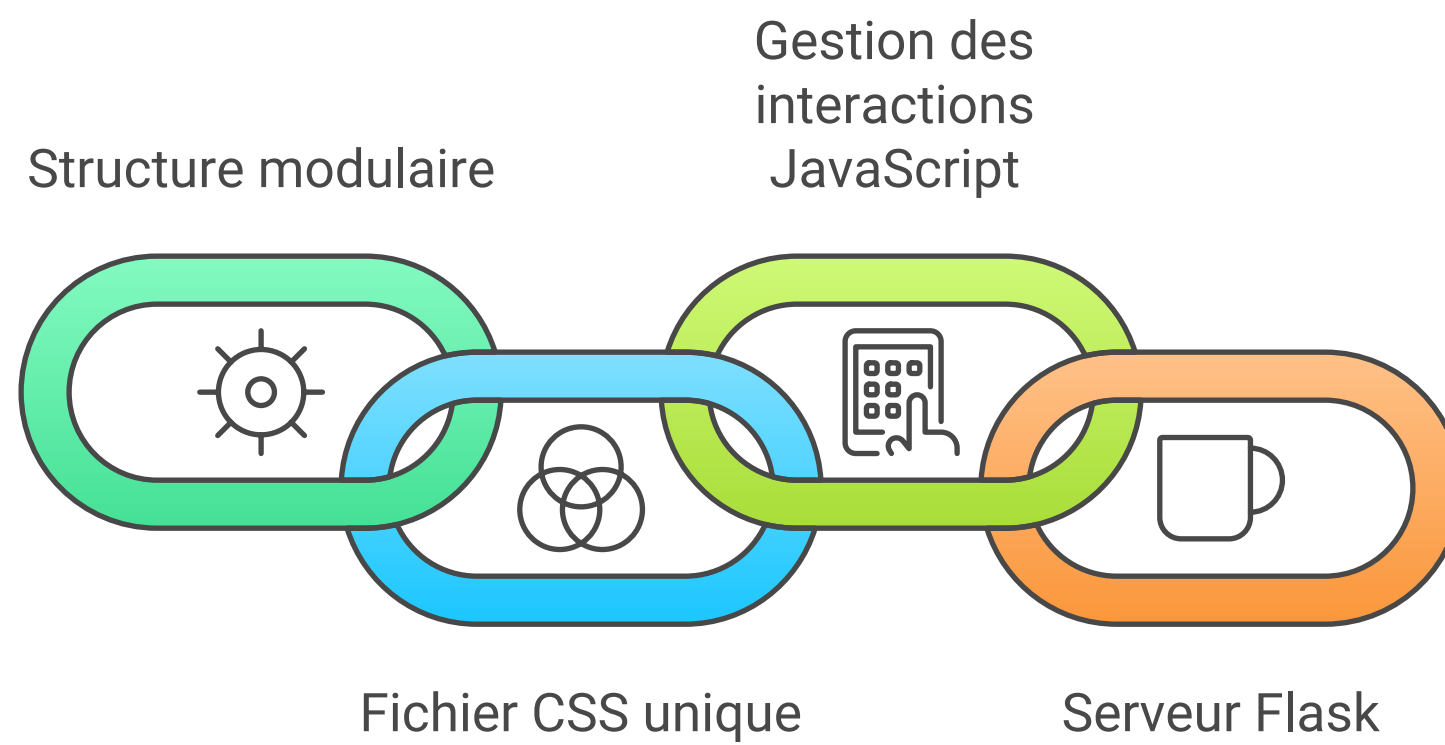
#### Stockage de données efficace

Emploi de SQLite  
pour une gestion de  
données légère et  
efficace.

## 5. Architecture de l'application

- Structure modulaire : Pages indépendantes (adhérents, livres, emprunts).
- Fichier CSS unique pour garantir l'homogénéité visuelle.
- JavaScript pour gérer les interactions utilisateur et communication avec l'API.
- Flask comme serveur pour les services RESTful avec endpoints clairement définis pour chaque type de données.

## Conception d'une Architecture d'Application Efficace et Cohérente



## 6. Description des Fonctionnalités

### 6.1 Module de Gestion des Adhérents

- Formulaire détaillé pour saisir et enregistrer des informations sur les adhérents (nom, prénom, adresse, etc.).
- Visualisation et suppression dynamique des données.

### 6.2 Module de Gestion des Livres

- Ajout manuel et importation des livres via fichiers CSV.
- Affichage clair des livres disponibles avec possibilité d'édition ultérieure.

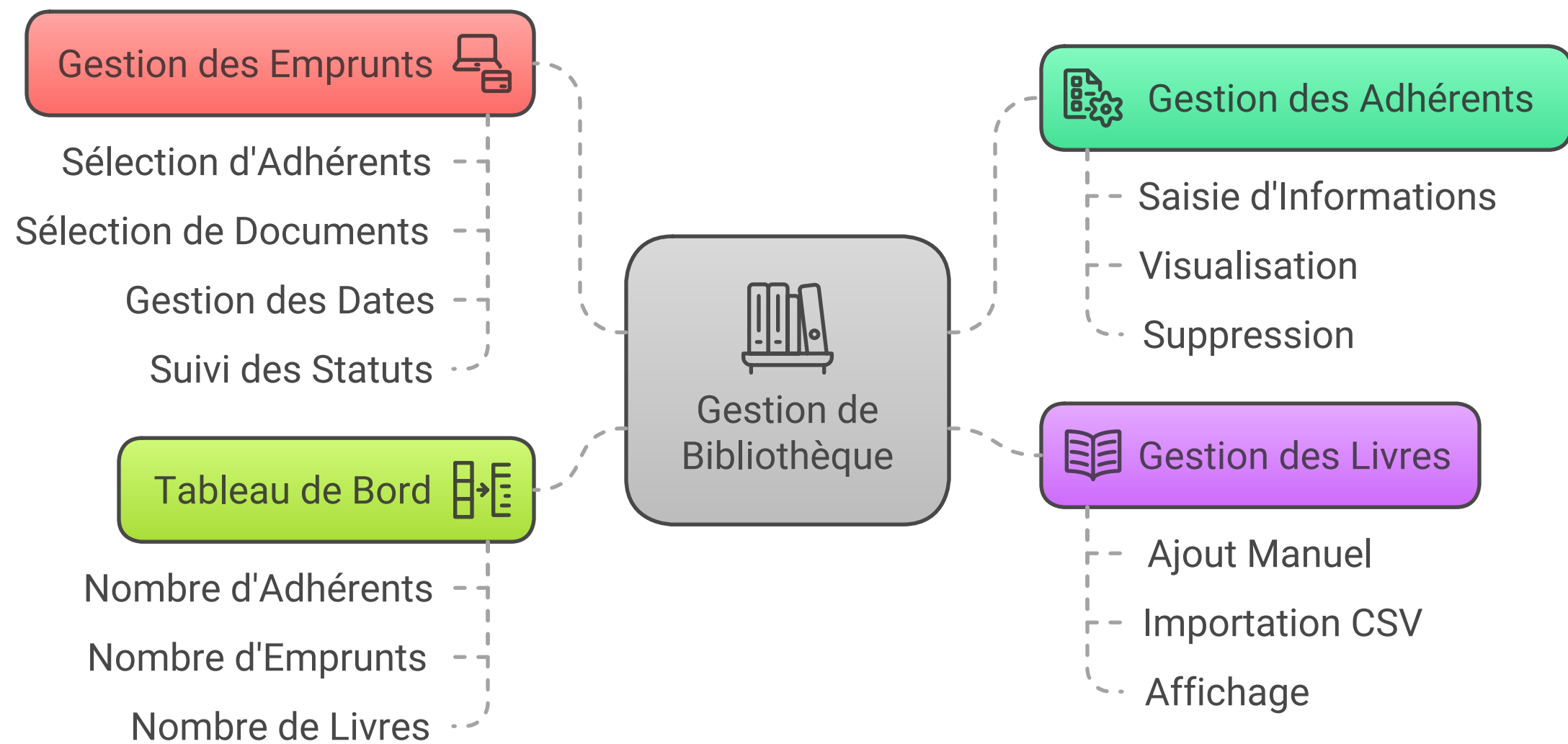
### 6.3 Module de Gestion des Emprunts

- Sélection d'adhérents et de documents à partir de listes déroulantes dynamiques.
- Gestion précise des dates d'emprunt et de retour.
- Suivi des statuts des emprunts en cours ou clôturés.

### 6.4 Tableau de Bord

- Synthèse visuelle et dynamique de l'état actuel de la bibliothèque (nombre total d'adhérents, emprunts et livres).

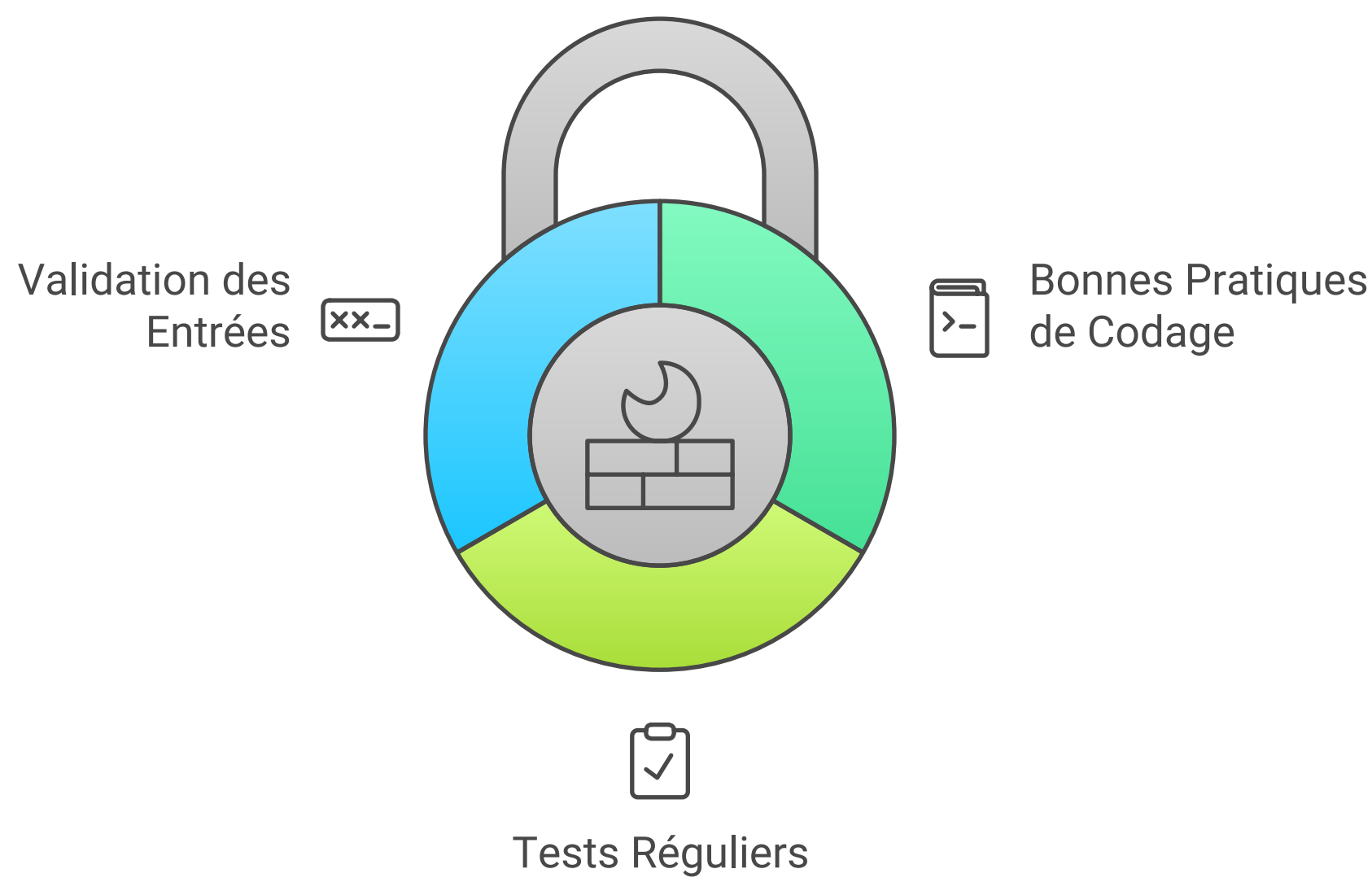
## Gestion de Bibliothèque: Modules et Fonctionnalités



### 7. Développement et Mise en œuvre

- Respect des bonnes pratiques de codage (clarté, commentaires explicatifs dans le code).
- Tests réguliers à chaque étape clé du développement pour assurer stabilité et fonctionnalité.
- Validation systématique des entrées utilisateur pour renforcer la sécurité des données.

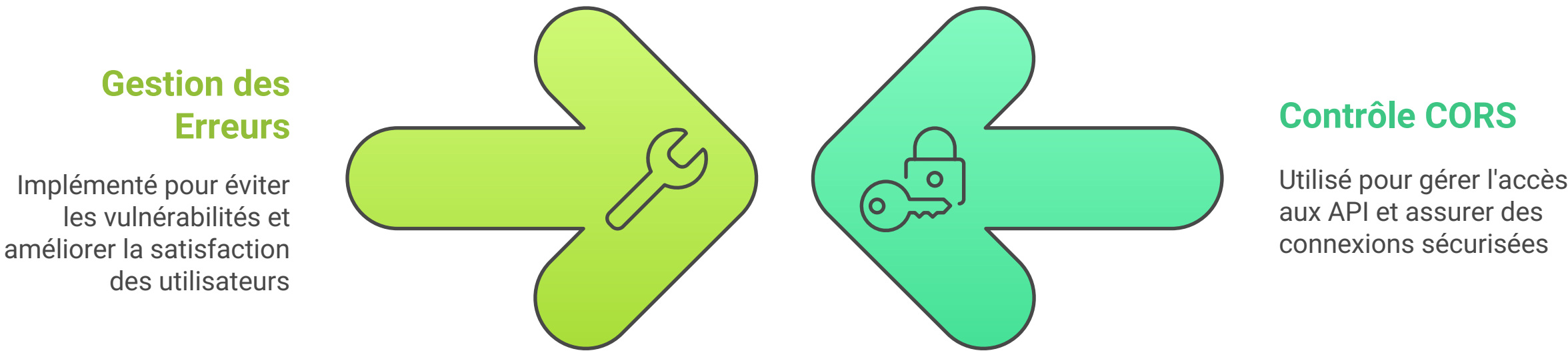
Assurer Stabilité et Sécurité par des Pratiques de Développement Rigoureuses



8. Sécurité

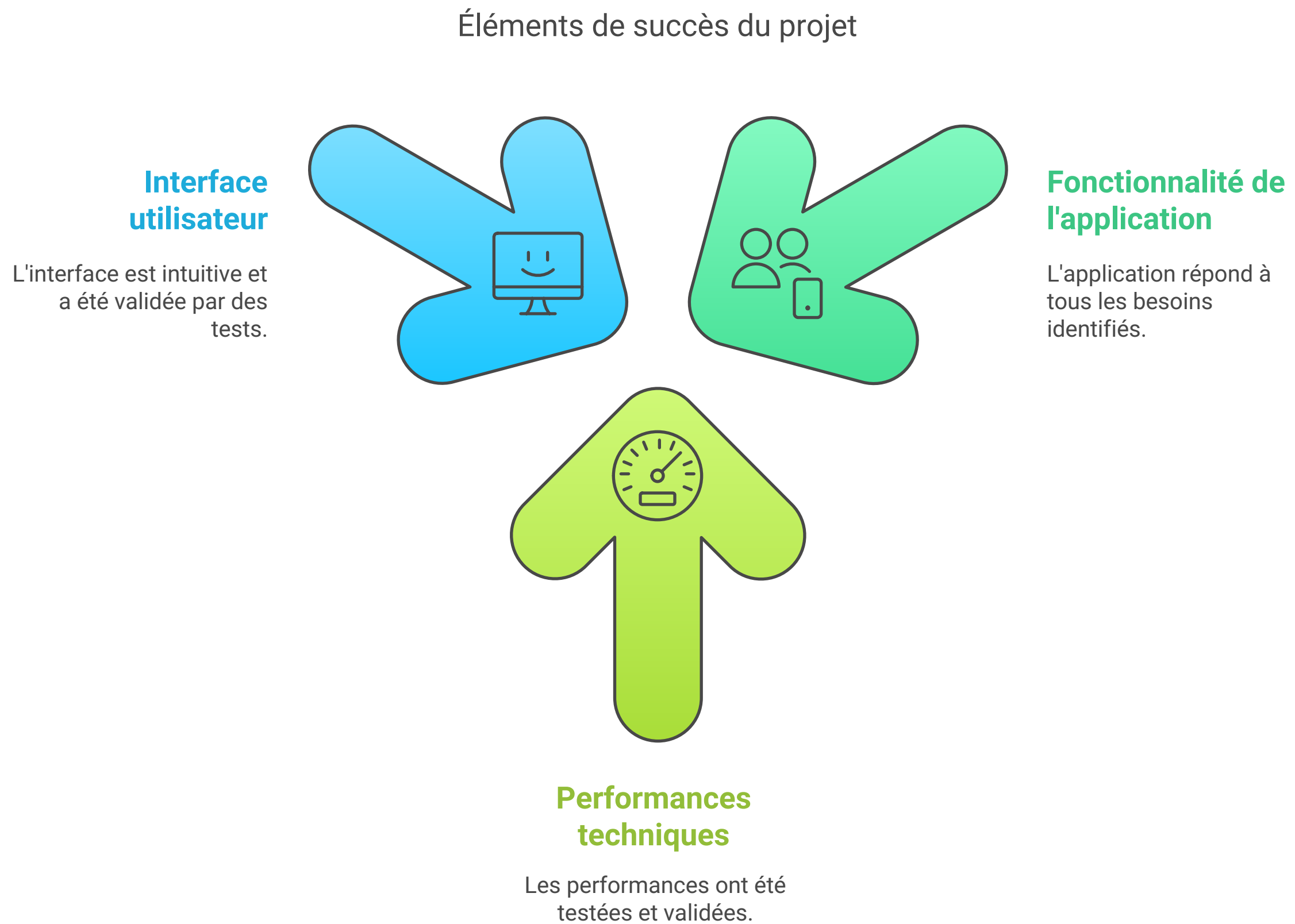
- Utilisation de CORS pour contrôler l'accès API.
- Gestion rigoureuse des erreurs pour prévenir les failles potentielles et assurer une expérience utilisateur optimale.

Amélioration de la Sécurité et de l'Expérience Utilisateur



## 9. Évaluation et Résultats

- Application entièrement fonctionnelle répondant à tous les besoins initialement identifiés.
- Performances techniques testées et validées.
- Interface intuitive validée par des tests utilisateur.



## 10. Conclusion et Perspectives

Ce projet atteint pleinement ses objectifs initiaux et présente un fort potentiel d'évolution.

Les prochaines étapes envisagées comprennent :

- Déploiement en production sur un serveur distant.
- Intégration de fonctionnalités complémentaires telles que la recherche avancée.
- Renforcement continu de la sécurité avec une gestion avancée des droits utilisateurs.

# Plan de développement de projet



## Déploiement en production

Déployer le projet sur un serveur distant



## Intégration de la recherche avancée

Ajouter des fonctionnalités de recherche avancée au projet



## Amélioration de la sécurité

Mettre en œuvre des mesures de sécurité renforcées



### Validé par :

Dyhia Berkhouche | Sarah Ould Fella | Salah Mouhand kaci