

**Rapport de projet**

**3ème année**

**Ingénierie Informatique et Réseaux**

Sous le thème

to do list-application

**Réalisé par :**

* Hafsa BOUCHENAFA
* Ikram BOUTAHRI
* Yassmine ESSALEH

**Encadré par :**

Tuteur de l’école : Khalid Nafil

Année universitaire :2024-2025

Remerciement

Je tiens à exprimer ma sincère gratitude à Mr. Khalid Nafil pour son soutien constant et ses conseils précieux tout au long de ce projet. Grâce à ses explications claires, son engagement et sa disponibilité, j’ai pu surmonter les défis techniques et réussir à mener ce projet à bien. Ses enseignements ont été une véritable source d’inspiration et ont grandement contribué à l’acquisition de nouvelles compétences en gestion de projet.

**Sommaire**

[Chapitre 1 : 1](#_Toc187782535)

[Contexte Générale 1](#_Toc187782536)

[**1.** **Introduction** 2](#_Toc187782537)

[**2.** **Objectifs du projet** 2](#_Toc187782538)

[**3.** **Technologies et outils utilisées :** 2](#_Toc187782539)

[**4.** **Conclusion** 3](#_Toc187782540)

[Chapitre 2 : 4](#_Toc187782541)

[Analyses des besoins 4](#_Toc187782542)

[**1.** **Introduction :** 5](#_Toc187782543)

[**2.** **Problématique :** 5](#_Toc187782544)

[**3.** **Exigences Fonctionnelles :** 5](#_Toc187782545)

[**4.** **Exigences non Fonctionnelles :** 6](#_Toc187782546)

[**5.** **Conclusion :** 7](#_Toc187782547)

[Chapitre 3 : 8](#_Toc187782548)

[Conception du système 8](#_Toc187782549)

[**1.** **Introduction :** 9](#_Toc187782550)

[**2.** **Identification des acteurs :** 9](#_Toc187782551)

[**3.** **Diagramme de cas d’utilisation :** 10](#_Toc187782552)

[**4.** **Conclusion :** 12](#_Toc187782554)

[Chapitre 4 : 13](#_Toc187782555)

[Réalisation de projet 13](#_Toc187782556)

[1. **Introduction** 14](#_Toc187782557)

[**2.** **Les interfaces graphiques :** 14](#_Toc187782558)

[**a.** connexion 14](#_Toc187782559)

[**b .** Interface utilisateur de la To-Do List 17](#_Toc187782561)

[**c .** Historique des actions sur les tâches dans une application de gestion de tâches](#_Toc187782562) 18

**[f.](#_Toc187782563)** [Tableau de bord de progression des tâches dans une application de gestion](#_Toc187782563)

[18](#_Toc187782563)

[**3.** **Conclusion :** 19](#_Toc187782564)

[Chapitre 5 : 20](#_Toc187782565)

[Conclusion 20](#_Toc187782566)

Introduction

Ce projet consiste à développer une application web de gestion de tâches (To-Do List) à l’aide du Framework Django, qui permet de construire rapidement des sites web sécurisés et maintenables. L’objectif principal de ce travail est de créer une plateforme simple, intuitive et efficace permettant aux utilisateurs de gérer leurs tâches quotidiennes de manière centralisée.

La To-Do List est une application classique mais très utile, qui permet d’organiser les activités personnelles ou professionnelles. Elle permet d’ajouter, modifier, marquer comme terminée ou supprimer des tâches. Ce projet met l’accent sur l’aspect pratique de l’organisation numérique, tout en mettant en œuvre les bonnes pratiques du développement web.

Le choix de Django se justifie par sa robustesse, sa richesse fonctionnelle, ainsi que sa communauté très active. Il intègre par défaut un ORM performant, un système de routage clair, une interface d’administration, et un système de gestion des utilisateurs.

Ce projet s'inscrit également dans une logique pédagogique, permettant d’aborder l’ensemble du cycle de développement logiciel : analyse des besoins, conception, implémentation, tests, et documentation. Il met l’accent sur la modularité, la réutilisabilité du code, et la séparation des responsabilités.

# Chapitre 1 :

# Contexte Générale

## **Introduction**

Dans cette section, nous présentons les objectifs fonctionnels et pédagogiques du projet de To-Do List. Le but principal est de développer une application web simple mais complète, qui permet à un utilisateur de gérer ses tâches de manière fluide. À travers ce projet, nous visons non seulement à mettre en œuvre les fonctionnalités essentielles d’un gestionnaire de tâches, mais aussi à appliquer les bonnes pratiques en matière de développement web, d’architecture logicielle et de gestion de base de données.

L’objectif principal est de concevoir une application web de gestion de tâches qui permet à un utilisateur d’ajouter, modifier, supprimer et marquer des tâches comme terminées. Ce projet vise aussi à renforcer la maîtrise des outils de développement web modernes.

## **Objectifs du projet**

L’objectif principal est de concevoir une application web de gestion de tâches qui permet à un utilisateur d’ajouter, modifier, supprimer et marquer des tâches comme terminées. Ce projet vise aussi à renforcer la maîtrise des outils de développement web modernes.

## **Technologies et outils utilisées :**



Visual Studio Code est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l’édition, à la construction, jusqu’au débogage.

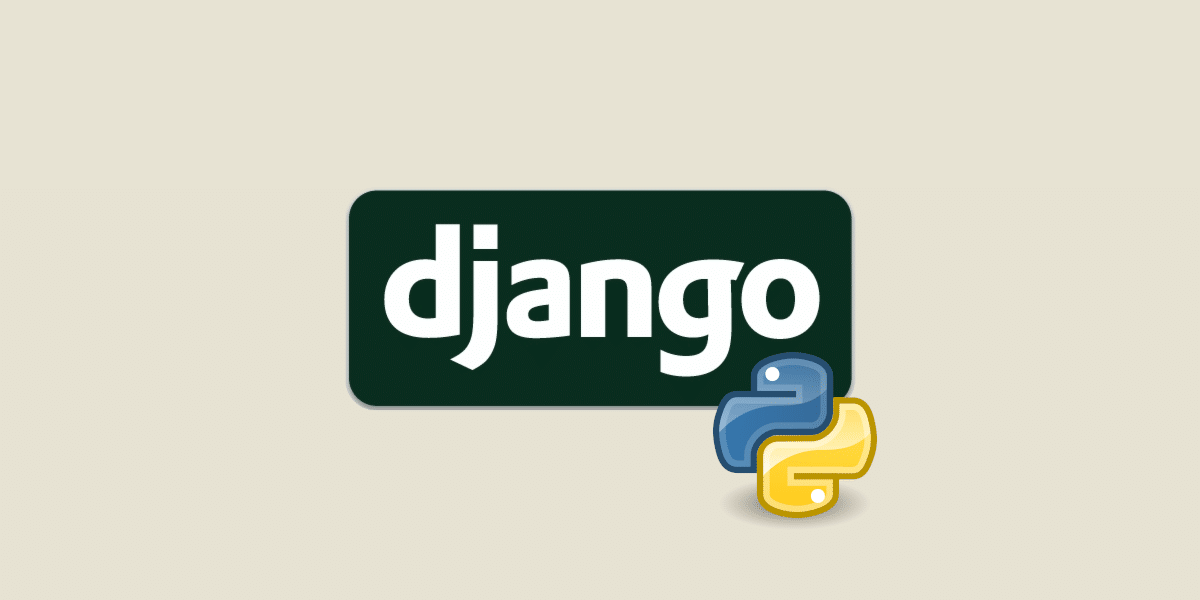
Le HTML est donc un langage de balisage, c’est-à-dire un langage qui va nous permettre de définir les différents contenus d’une page.

Le CSS vient résoudre un problème bien différent du HTML en effet, le HTML sert à définir les différents éléments d’une page, à leur donner du sens. Le CSS, lui, va servir à mettre en forme les différents contenus définis par le HTML en leur appliquant des styles.





Utilisé pour développer l’ensemble de la logique de l’application. C’est un langage clair, puissant, et orienté objet, largement utilisé dans le développement web grâce à sa simplicité et sa richesse.



Django est un Framework de haut niveau basé sur Python. Il permet un développement rapide et propre d’applications web grâce à son ORM intégré, ses fonctionnalités de sécurité, et son système de Template.

 Base de données légère, intégrée à Django et adaptée aux petits projets ou au développement local. Elle permet de stocker toutes les données utilisateurs et les tâches créées dans l’application

## **Conclusion**

le premier chapitre a permis de poser les bases du projet en définissant son contexte, ses objectifs et les outils utilisés. Le choix de réaliser une application de type To-Do List répond à un besoin concret d'organisation et de productivité, tout en offrant un terrain idéal pour appliquer les notions fondamentales du développement web. Grâce à Django et aux technologies associées, ce projet allie efficacité de développement, simplicité de mise en œuvre, et bonnes pratiques logicielles.

Le chapitre suivant portera sur l'analyse détaillée des besoins fonctionnels et non fonctionnels, afin de cadrer précisément les attentes du système avant sa conception.

# Chapitre 2 :

# Analyses des besoins

## **Introduction :**

## Une bonne analyse des besoins permet de cadrer le périmètre du projet. Elle définit les attentes du client ou des utilisateurs finaux. Dans notre cas, l'application To-Do List vise à améliorer l’organisation personnelle ou professionnelle des utilisateurs grâce à un outil numérique simple et accessible. Cette étape permet de comprendre ce que l’application doit faire, pour qui, et dans quelles conditions elle sera utilisée.

## **Problématique :**

Dans la vie quotidienne ou professionnelle, les utilisateurs sont confrontés à une multitude de tâches à gérer, ce qui peut engendrer de la désorganisation ou des oublis. L'absence d’un système structuré rend difficile la planification et le suivi des tâches. Ce projet répond à cette problématique en fournissant une interface conviviale, intuitive et accessible depuis n’importe quel navigateur, permettant aux utilisateurs de créer, modifier et suivre leurs tâches de manière centralisée.

## **Exigences Fonctionnelles :**

Les exigences fonctionnelles décrivent les fonctionnalités que le système doit

Obligatoirement offrir :

• Authentification des utilisateurs (inscription, connexion, déconnexion)  
 • Création de nouvelles tâches avec un titre et éventuellement une description  
 • Modification des tâches existantes  
 • Suppression de tâches terminées ou inutiles  
 • Marquage de tâches comme "terminées"  
 • Affichage organisé des tâches en cours et terminées  
 • Interface utilisateur accessible uniquement après connexion  
 • Consultation de l’historique des tâches réalisées  
 • Affichage de statistiques (par priorité, état, etc.)  
 • Exportation des tâches et statistiques au format PDF

## **Exigences non Fonctionnelles :**

Les exigences non fonctionnelles décrivent les caractéristiques attendues du système sans lien direct avec des fonctionnalités :

* Interface claire, responsive et facile à prendre en main
* Rapidité d'exécution des opérations (ajout, suppression, etc.)
* Sécurité des comptes utilisateurs (mots de passe cryptés, sessions sécurisées)
* Compatibilité multi-navigateurs
* Structure de code propre et bien documentée
* Évolutivité pour accueillir des fonctionnalités futures (ex. rappels, catégories

## **Conclusion :**

Cette analyse permet d’identifier les attentes précises du système, tant au niveau technique que fonctionnel. Elle constitue une base solide pour établir les choix de conception qui guideront l’architecture de l’application et son développement futur.

# 

# 

# Chapitre 3 :

# Conception du système

## **Introduction :**

La phase de conception consiste à modéliser les différents composants du système avant leur implémentation. Elle permet de structurer les idées, de définir les entités principales, leurs interactions, ainsi que les parcours utilisateur. Cette étape est cruciale pour anticiper les difficultés, garantir la cohérence des traitements, et faciliter le développement technique ultérieur. La modélisation se base sur les exigences recueillies lors de l'analyse des besoins et s’articule autour de deux éléments clés : les acteurs et leurs cas d’utilisation, puis les structures de données à manipuler.

## **Identification des acteurs :**

Dans cette application de To-Do List, l’acteur principal est :

* **Utilisateur** : Il s’agit de toute personne disposant d’un compte sur l’application. Il peut s’inscrire, se connecter, consulter la liste de ses tâches, en ajouter de nouvelles, les modifier, les supprimer ou les marquer comme terminées. Ce rôle unique simplifie l’organisation des fonctionnalités et permet une interface homogène.

#### Diagramme de cas d’utilisation :

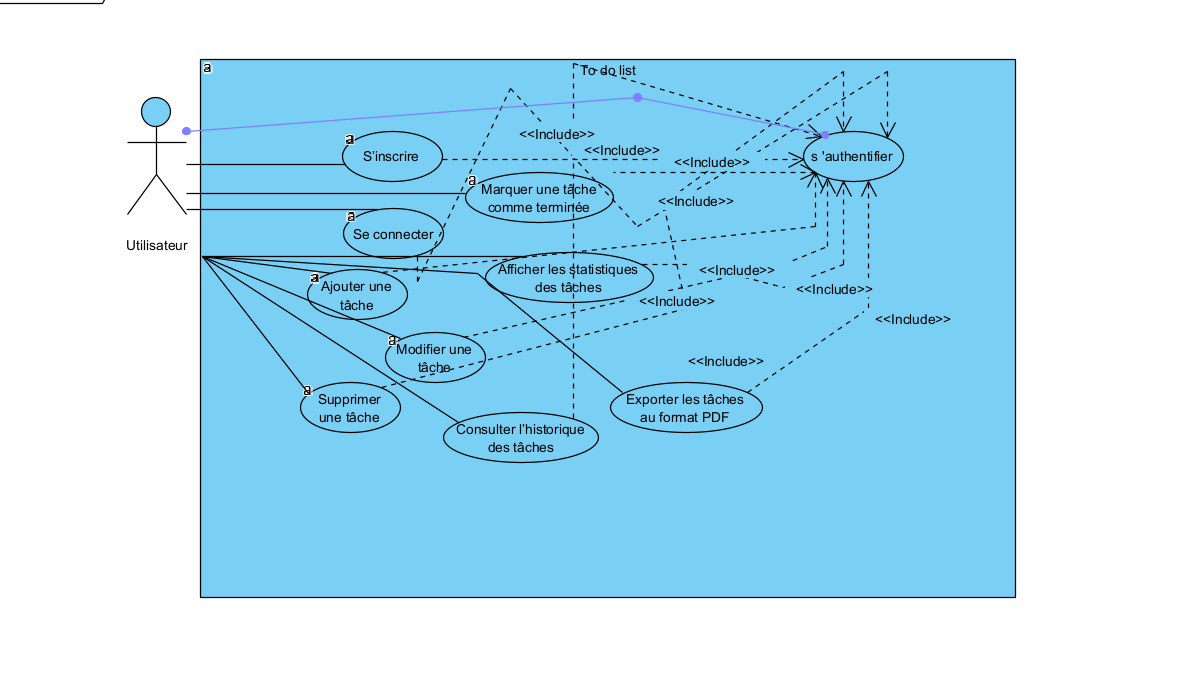
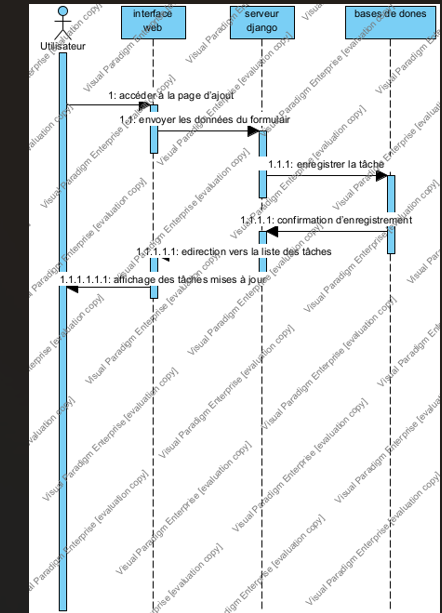


Figure 1:Diagramme de contexte statique

Ce diagramme modélise les interactions entre l’acteur principal (**Utilisateur**) et le système. Voici les cas d'utilisation principaux :

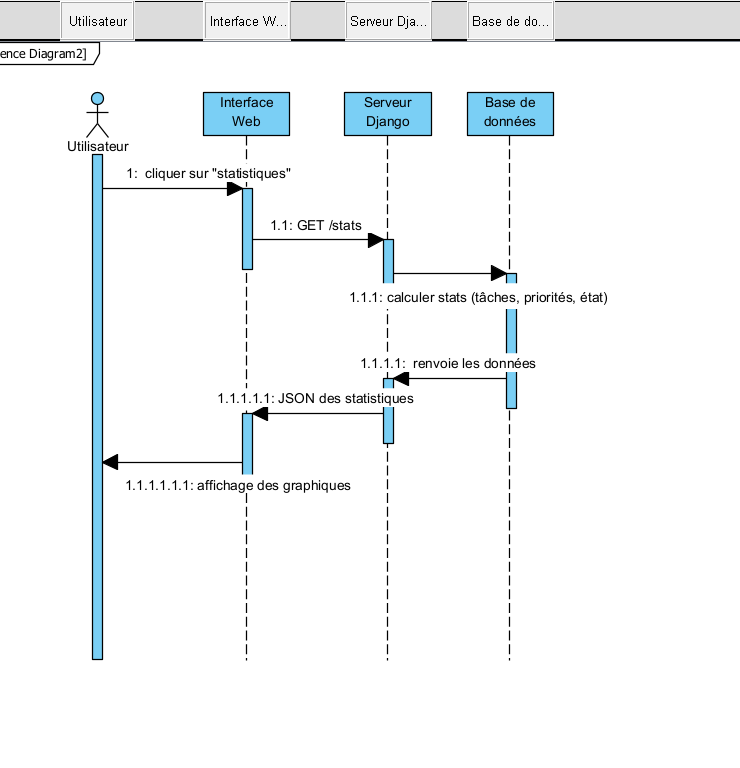
• **S’inscrire** : L’utilisateur fournit les informations requises pour créer un compte.  
• **Se connecter** : L’utilisateur entre ses identifiants pour accéder à son espace.  
• **Ajouter une tâche** : L’utilisateur crée une nouvelle tâche avec un titre et éventuellement une description.  
• **Modifier une tâche** : L’utilisateur peut modifier le contenu ou l’état d’une tâche existante.  
• **Supprimer une tâche** : Permet de supprimer définitivement une tâche.  
• **Marquer une tâche comme terminée** : Permet de signaler qu’une tâche a été accomplie.  
• **Consulter l’historique des tâches** : L’utilisateur peut visualiser les actions passées (créations, modifications, suppressions).  
• **Afficher les statistiques** : Permet à l’utilisateur de voir des statistiques sur ses tâches (nombre total, taux de complétion, répartition par priorité, etc.).  
• **Exporter les données en PDF** : Génère un document contenant la liste des tâches, l’historique et les statistiques.

* Ce diagramme est un outil essentiel pour s’assurer que toutes les fonctionnalités prévues sont bien associées à des actions utilisateurs claires.
* Ce diagramme est un outil essentiel pour s’assurer que toutes les fonctionnalités prévues sont bien associées à des actions utilisateurs claires

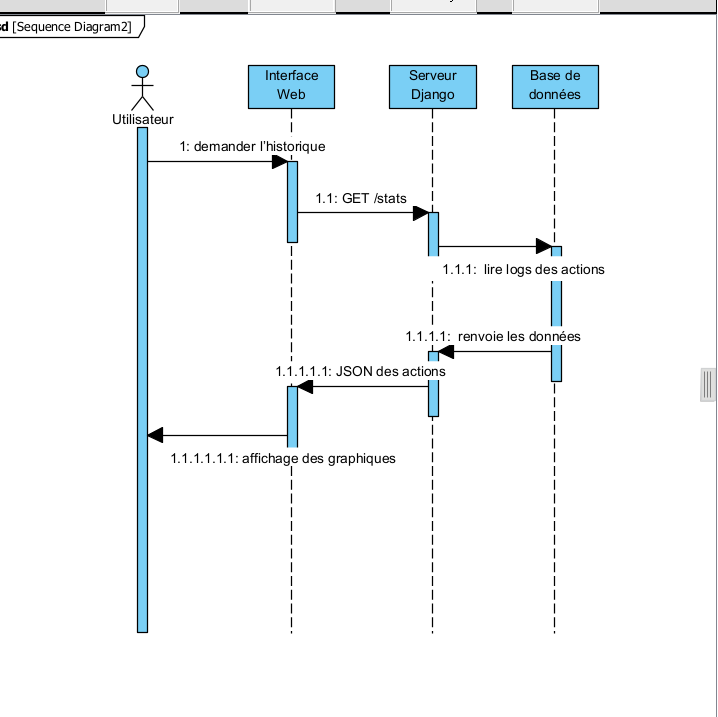
*Diagramme de séquence*

Ce diagramme montre les échanges successifs qui se produisent dans le système de manière dynamique.

### Voir les statistiques :



Consulter l’historique :



## **Conclusion :**

la conception permet de structurer le projet, d’assurer la cohérence entre les modules, et de faciliter l’implémentation en clarifiant les responsabilités de chaque composant. Elle sert de base technique solide pour entamer le développement du système.

# 

# Chapitre 4 :

# Réalisation de projet

## **Introduction**

La phase de réalisation consiste à transformer la conception en un produit fonctionnel. Elle mobilise l’ensemble des outils et technologies présentés précédemment, et comprend le développement des modèles, des vues, des Template, ainsi que la configuration des routes et la gestion des sessions utilisateurs.

## **Les interfaces graphiques :**

### **Page de connexion :**

Cette page représente une interface de connexion en français, conçue pour permettre aux utilisateurs d'accéder à une plateforme sécurisée. Elle contient deux champs de saisie : l’un pour le Nom d'utilisateur et l’autre pour le Mot de passe. Un bouton bleu intitulé Se connecter permet de valider les informations saisies. En dessous, un lien Mot de passe oublié ? est disponible pour aider les utilisateurs à récupérer leur mot de passe en cas d'oubli. Le design est simple, épuré et centré sur un fond clair.

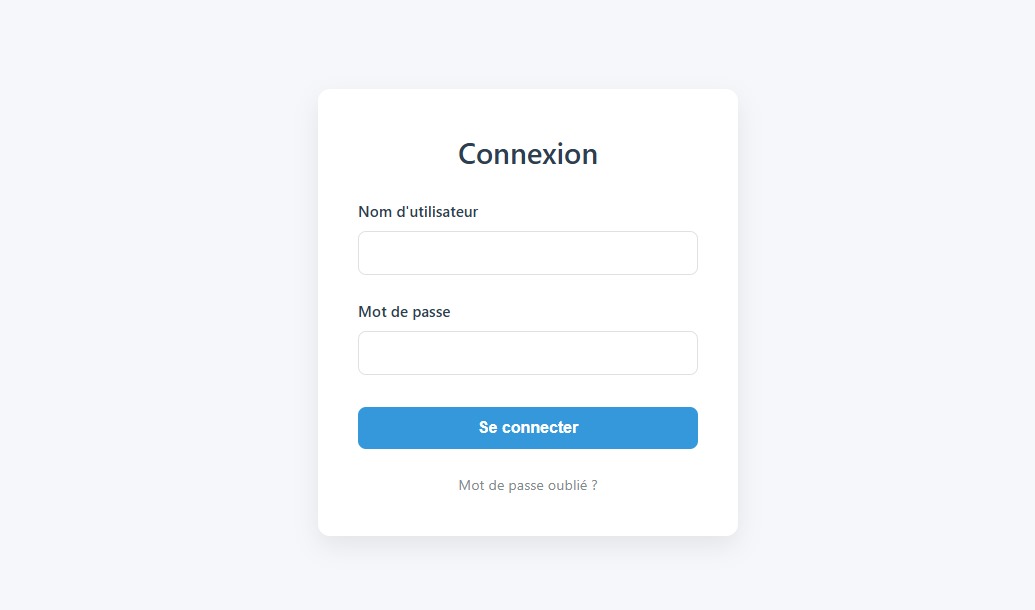


Figure :page de connexion

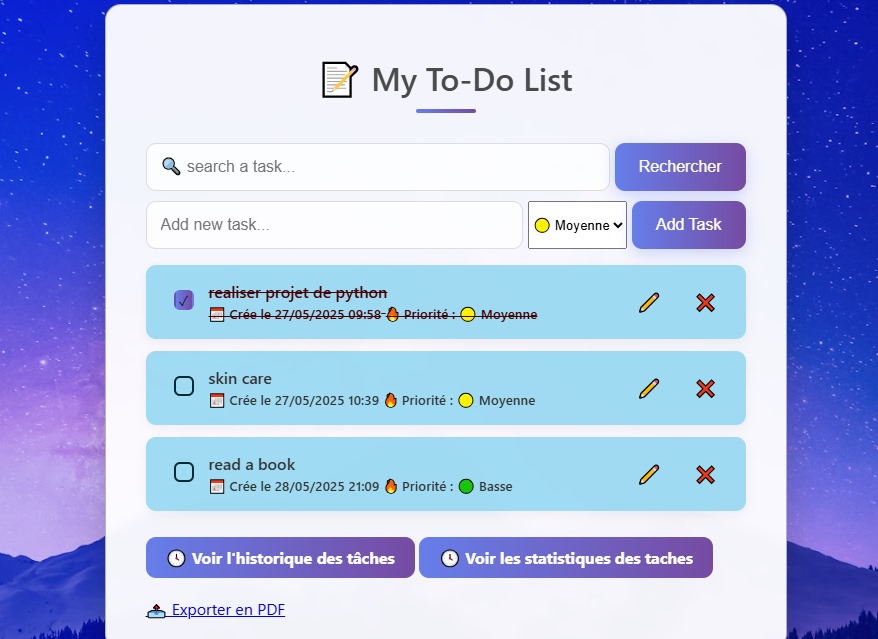


Figure : Interface utilisateur de la To-Do List

Cette interface présente l’application To-Do List avec plusieurs fonctionnalités intégrées. L’utilisateur peut rechercher une tâche, en ajouter une nouvelle avec un niveau de priorité (basse, moyenne ou haute), modifier ou supprimer les tâches existantes.  
Les tâches terminées sont affichées avec un texte barré et une coche cochée.  
En bas de l’interface, trois fonctionnalités avancées sont proposées :

* **Voir l’historique des tâches** : permet de consulter les actions effectuées sur les tâches.
* **Voir les statistiques des tâches** : affiche des indicateurs de productivité.
* **Exporter en PDF** : génère un rapport contenant les tâches et les statistiques.



Figure 6

Historique des actions sur les tâches dans une application de gestion de tâches

Cette figure présente l'interface utilisateur d’une application affichant l’historique des tâches réalisées ou modifiées. Intitulée **« Historique des tâches »**, elle affiche une liste chronologique des actions effectuées sur différentes tâches, avec les éléments suivants pour chaque entrée :

* 📅 La **date et l’heure** de l’action (par exemple : 28/05/2025 21 :09),
* 📄 Le **nom de la tâche** (ex : Read a book, réaliser projet de python, skin care),
* 🔧 Le **type d’action** réalisée (création, modification, complétée, réouverte),
* Un **bouton de retour** intitulé « Retour à la liste » en bas de l’écran.

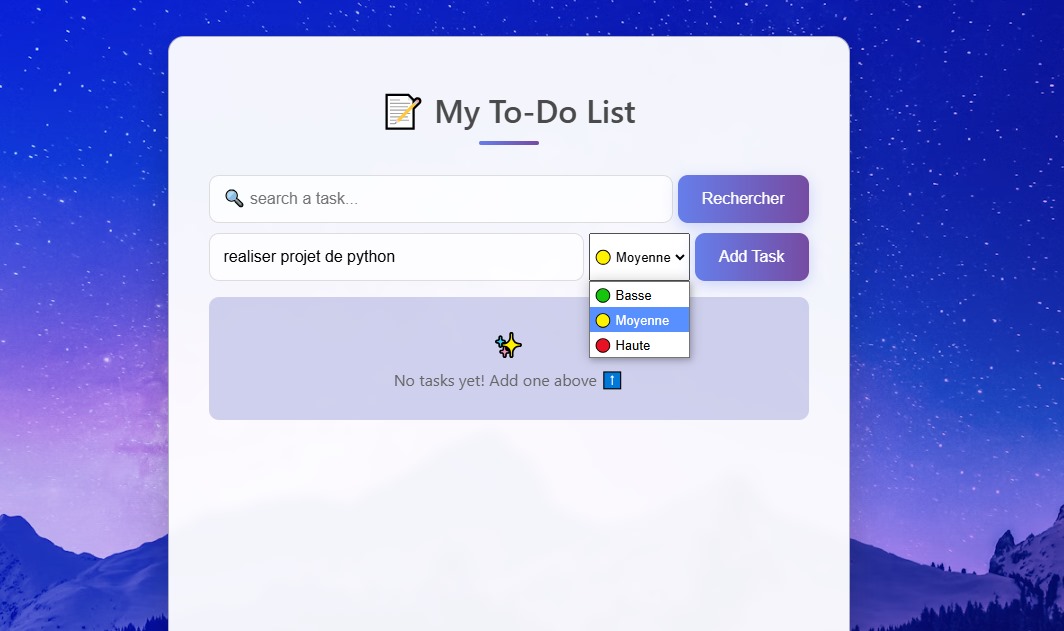


Figure 7

L'utilisateur peut sélectionner un niveau de priorité pour chaque tâche à l'aide d'un menu déroulant :

Basse (pastille verte)

Moyenne (pastille jaune)

Haute (pastille rouge)

Cette priorité permet de classer visuellement les tâches selon leur importance.

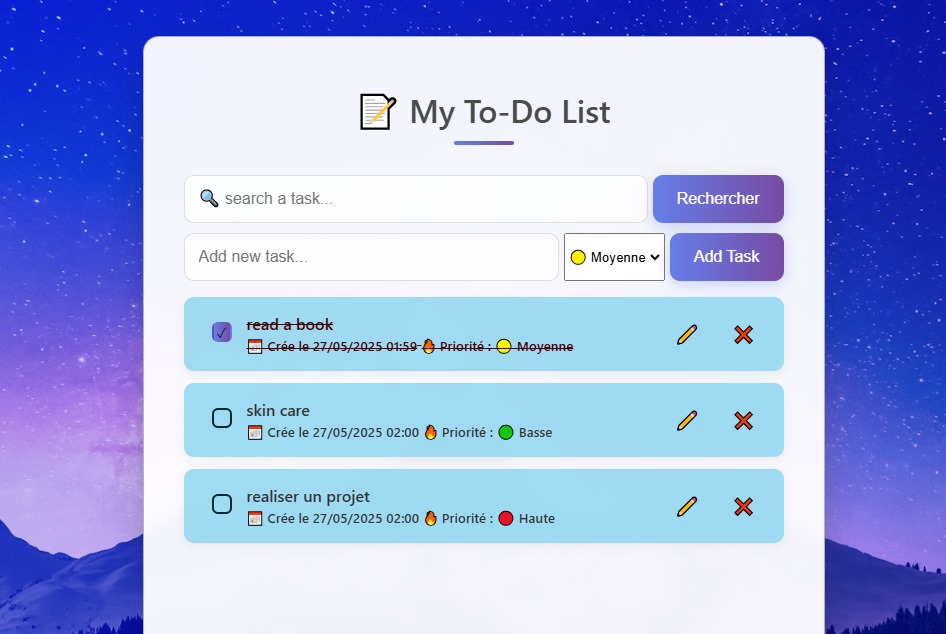


Figure 8

Cette capture d'écran montre une application de liste de tâches ("To-Do List") avec trois tâches visibles : Read a book, skin care, et réaliser un projet. Chaque tâche affiche la date de création, le niveau de priorité (basse, moyenne ou haute), et offre des options pour modifier, supprimer ou cocher la tâche comme terminée. La tâche "Read a book" est barrée, ce qui indique qu'elle est complétée.

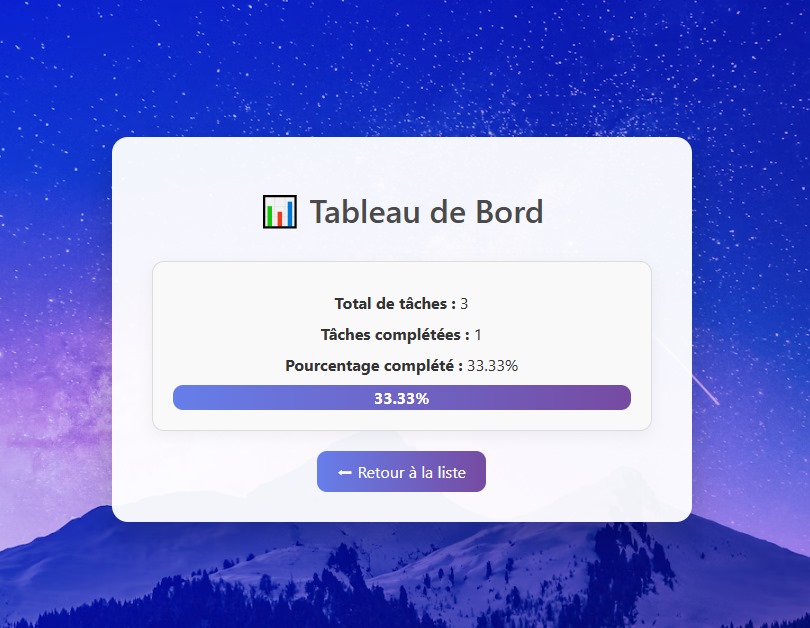


Tableau de bord de progression des tâches dans une application de gestion

Cette figure représente le **tableau de bord** d'une application de gestion de tâches. Elle fournit un aperçu synthétique de l’état d’avancement de l'utilisateur à travers trois indicateurs principaux :

* **Total de tâches :** 3
* **Tâches complétées :** 1
* **Pourcentage complété :** 33.33 %

Une barre **de progression horizontale** colorée illustre visuellement ce pourcentage. La barre affiche également la valeur numérique (33.33 %) en son centre, renforçant la lisibilité. En bas de l’interface, un bouton intitulé **« Retour à la liste »** permet de revenir à l'affichage principal des tâches.

# Chapitre 5 :

# Conclusion

Ce projet de développement d'une application web de type To-Do List avec Django a permis de parcourir l'ensemble du cycle de vie d'une application, depuis la phase d'analyse des besoins jusqu'à la mise en œuvre concrète de ses fonctionnalités.

Tout d'abord, la phase de **définition des objectifs** et de **choix technologiques** a posé les bases solides du projet. Le choix de Django comme Framework s'est avéré pertinent, notamment pour sa structure bien organisée, sa richesse fonctionnelle et son interface d'administration intégrée. Les autres technologies telles que HTML, CSS, SQLite et Git ont également contribué à la réussite du projet.

La phase d’**analyse fonctionnelle et non fonctionnelle** a permis de bien cerner les besoins des utilisateurs, notamment la nécessité d'une interface simple, réactive, et sécurisée pour la gestion de tâches. Ces besoins ont été traduits en exigences claires qui ont orienté la conception technique.

Ensuite, lors de la **conception du système**, une attention particulière a été portée sur la modélisation des cas d’utilisation et la structure de la base de données. Le diagramme de séquence et le modèle entité-relation ont servi de guide tout au long de l’implémentation.

La **réalisation** s’est traduite par le développement concret des fonctionnalités essentielles de l’application : inscription, connexion, création, modification, suppression et consultation des tâches. Les interfaces ont été pensées pour offrir une expérience utilisateur fluide et agréable.

Ce projet a également été l’occasion de renforcer plusieurs compétences techniques :

* La structuration du code en architecture MVC/MVT avec Django.
* L’intégration du système de gestion des utilisateurs et des sessions.
* La gestion de la base de données avec migrations automatiques.
* L'utilisation de Git pour le suivi de version et la collaboration potentielle.

Enfin, l’ajout de fonctionnalités avancées telles que l’historique des actions, l’affichage de statistiques et l’exportation au format PDF constitue une réelle valeur ajoutée pour l’utilisateur. Elles renforcent l’utilité de l’application dans un cadre personnel comme professionnel, tout en illustrant l’adaptabilité du projet à de futures évolutions.