

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité Instrumentation

Présenté par

Toumia Moussa

&

Hamhoum Abderrahmane

Plateforme pour TP Vibration et Onde

Proposé par : Mme. Amirouche Kais Nesrine

Année Universitaire 2023-2024

Remerciements

Nous offrons notre grande gratitude à Dieu qui nous a aidés à faire ce travail.

Nous exprimons notre profonde gratitude à nos parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont endurés.

Nous remercions notre promotrice Mme Amirouche Kais Nesrine pour les efforts qu'elle a déployés, pour nous aider, conseiller, encourager et corriger. Nous voudrions remercier les membres du jury d'avoir accepté d'examiner notre travail.

Nous remercions aussi tout le corps enseignant dans le département de Génie Électrique qui a contribué à notre formation universitaire. Sans oublier tous nos amis, Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici notre sincère reconnaissance.

Dédicaces

À mes parents, dont l'amour inconditionnel et le soutien sans faille m'ont donné la force d'aller de l'avant. Merci pour votre confiance et vos encouragements constants.

À mes enseignants et professeurs, pour leur dévouement et leur passion pour l'enseignement. Vous avez allumé en moi la flamme de la curiosité et de la soif de savoir.

À mes amis, pour leur compréhension, leur patience et leurs encouragements tout au long de ce parcours. Votre amitié a été une source inestimable de réconfort et de motivation.

À tous ceux qui ont cru en moi et m'ont soutenu d'une manière ou d'une autre dans la réalisation de ce travail. Votre présence et vos conseils ont été essentiels.

Et enfin, à tous les étudiants et chercheurs qui poursuivent la quête de la connaissance avec passion et détermination. Puissiez-vous trouver autant de satisfaction et d'épanouissement dans votre parcours que j'en ai trouvé dans le mien.

ملخص: تتضمن هذه المشروع تطوير منصة تفاعلية باستخدام لغة البرمجة بايثون ومكتبة تكتنر ، مخصصة لطلاب السنة الثانية (المستوى L2). تقوم المنصة بتركيز الأعمال العملية لمادة "الاهتزازات والموجات"، وتقدم واجهة مستخدم بديهية. تشمل الميزات الرئيسية تمارين تفاعلية، ورسوم متحركة لتوضيح المفاهيم، ودلائل تفصيلية، وتقييمات مع ردود فعل. الهدف الرئيسي هو تعزيز إمكانية الوصول والمشاركة في تعلم الاهتزازات والموجات من خلال أعمال عمل عملية تفاعلية

وديناميكية .

كلمات المفاتيح: الأعمال العملية، الاهتزازات والموجات، بايثون، تكتنر، واجهة المستخدم، تطبيق تفاعلي، التعليم العلمي،

المحاكاة، التقييم، ردود

الفعل

Résumé : Ce projet présente le développement d'une plateforme interactive en Python, utilisant la bibliothèque graphique Tkinter, destinée aux étudiants de deuxième année de licence (L2). Cette application centralise les travaux pratiques (TP) du module "Vibrations et Ondes", offrant une interface utilisateur intuitive. Les fonctionnalités incluent des exercices interactifs et des animations pour illustrer les concepts, accompagnés de guides détaillés et d'évaluations avec feedback. L'objectif principal est d'améliorer l'accessibilité et l'engagement des étudiants dans l'apprentissage des concepts de vibrations et d'ondes par le biais de TP interactifs et dynamiques.

Mots clés : Travaux Pratiques ; Vibrations et Ondes ; Python ; Tkinter ; Interface Utilisateur ; Application Interactive ; Education Scientifique ; Simulation ; Evaluation ; Feedback

Abstract: This project involves the development of an interactive platform in Python using the Tkinter graphical library, aimed at second-year undergraduate students (L2). The platform centralizes practical work (TP) for the "Vibrations and Waves" module, offering an intuitive user interface. Key features include interactive exercises, animations to illustrate concepts, detailed guides, and assessments with feedback. The primary objective is to enhance accessibility and engagement in learning vibrations and waves through interactive and dynamic practical work.

Keywords: Practical Work; Vibrations and Waves; Python; Tkinter; User Interface; Interactive Application; Science Education; Simulation; Evaluation; Feedback

Table des matières

| | |
|------------------------------------|---|
| INTRODUCTION GENERALE | 1 |
|------------------------------------|---|

Chapitre 1 Uniformiser et Moderniser les Travaux Pratiques

| | | |
|-------|---|---|
| 1.1 | Introduction | 2 |
| 1.2 | Système LMD | 2 |
| 1.2.1 | Structure en trois niveaux | 2 |
| 1.2.2 | Flexibilité et modularité | 3 |
| 1.2.3 | Système de crédits | 3 |
| 1.2.4 | Adaptabilité internationale | 3 |
| 1.2.5 | Orientation professionnelle | 4 |
| 1.2.6 | Harmonisation européenne | 4 |
| 1.3 | Le cursus ST | 4 |
| 1.3.1 | Fondements scientifiques | 4 |
| 1.3.2 | Disciplines technologiques | 4 |
| 1.3.3 | Travaux pratiques et projets | 5 |
| 1.3.4 | Options de spécialisation | 5 |
| 1.3.5 | Stage en entreprise ou projet de fin d'études | 5 |
| 1.3.6 | Débouchés professionnels | 5 |
| 1.4 | Travaux Pratiques | 6 |
| 1.4.1 | Encouragement de la créativité | 7 |
| 1.4.2 | Préparation à la vie professionnelle | 7 |
| 1.5 | Conclusion..... | 7 |

Chapitre 2 Outils pour Interfaces Graphiques

| | | |
|-------|-----------------------------------|----|
| 2.1 | Introduction..... | 9 |
| 2.2 | Différents Outils Existants | 10 |
| 2.2.1 | En Python | 10 |

| | | |
|-------|---|----|
| 2.2.2 | En Java | 15 |
| 2.2.3 | En C# | 17 |
| 2.2.4 | En JavaScript/HTML/CSS | 20 |
| 2.3 | Justification du Choix de Tkinter | 22 |
| 2.3.1 | Simplicité et Accessibilité..... | 23 |
| 2.3.2 | Documentation et Communauté | 23 |
| 2.3.3 | Portabilité | 25 |
| 2.3.4 | Adéquation pour les Applications Simples à Modérées | 26 |
| 2.3.5 | Comparaison des Outils | 29 |
| 2.4 | Conclusion | 31 |

Chapitre 3 Développement de l'Interface Graphique

| | | |
|--------|---|----|
| 3.1 | Introduction..... | 33 |
| 3.2 | Développement de l'interface graphique | 35 |
| 3.2.1 | Importation des bibliothèques | 35 |
| 3.2.2 | Classe 'MainApp'..... | 36 |
| 3.2.3 | Méthode 'results1'..... | 41 |
| 3.2.4 | Méthode 'Results2'..... | 45 |
| 3.2.5 | Organisation de l'Interface | 50 |
| 3.2.6 | Interactivité..... | 50 |
| 3.2.7 | Validation des Entrées Numériques..... | 51 |
| 3.2.8 | Les fonctions | 51 |
| 3.2.9 | Quitter l'Application | 53 |
| 3.2.10 | Afficher la Boîte de Dialogue "À Propos"..... | 53 |
| 3.3 | Conclusion..... | 54 |

Chapitre 4 Implémentation

| | | |
|-----|-------------------|----|
| 4.1 | Introduction..... | 55 |
|-----|-------------------|----|

| | | |
|-------|--|-----------|
| 4.2 | Présentation des interfaces de notre application | 55 |
| 4.2.1 | Interface ‘Log In’ | 56 |
| 4.2.2 | Interface des TPs | 59 |
| 4.2.3 | Interface de TP1 | 60 |
| 4.2.4 | Fenêtre des Résultats TP1 | 62 |
| 4.2.5 | Fenêtre des Résultats TP2 | 71 |
| 4.3 | Conclusion | 82 |
| | CONCLUSION GENERALE | 83 |
| | BIBLIOGRAPHY | 85 |

Liste des figures

| | |
|--|----|
| <i>FIGURE 4-1- Interface 'Log In' de l'application</i> | 56 |
| <i>FIGURE 4-2- Fenêtre d'Erreur</i> | 57 |
| <i>FIGURE 4-3- Fenêtre de succès</i> | 58 |
| <i>FIGURE 4-4- Fenêtre de TP1</i> | 59 |
| <i>FIGURE 4-5- Fenêtre de TP2</i> | 59 |
| <i>FIGURE 4-6- Fenêtre d'option TP1</i> | 60 |
| <i>FIGURE 4-7- Question A Statique</i> | 63 |
| <i>FIGURE 4-8- Question B Statique</i> | 63 |
| <i>FIGURE 4-9- Question C Statique</i> | 64 |
| <i>FIGURE 4-10- Question D Statique</i> | 65 |
| <i>FIGURE 4-11- Question A, B Dynamique</i> | 66 |
| <i>FIGURE 4-12- Question C Dynamique</i> | 67 |
| <i>FIGURE 4-13- Question 4 Dynamique</i> | 69 |
| <i>FIGURE 4-14- Question E Dynamique</i> | 70 |
| <i>FIGURE 4-15- Graphe $T^2=f(m)$</i> | 70 |
| <i>FIGURE 4-16- Question F Dynamique</i> | 71 |
| <i>FIGURE 4-17- Question 1 Non Amortie</i> | 73 |
| <i>FIGURE 4-18- Question 2 Non Amortie</i> | 74 |
| <i>FIGURE 4-19- Graphe $T^2=f(L)$</i> | 75 |
| <i>FIGURE 4-20- Question 3 Non Amortie</i> | 76 |
| <i>FIGURE 4-21- Question 1 Amortie</i> | 76 |
| <i>FIGURE 4-22- Question 2 Amortie</i> | 77 |
| <i>FIGURE 4-23- Question 3 Amortie</i> | 78 |
| <i>FIGURE 4-24- Question 4 Amortie</i> | 79 |
| <i>FIGURE 4-25- Question 5 Amortie</i> | 80 |
| <i>FIGURE 4-26- Question 6 Amortie</i> | 81 |

Liste des tableaux

| | |
|---|----|
| Tableau 2-1- Tkinter | 29 |
| Tableau 2-2- PyQt/Kivy/wxPython. | 29 |
| Tableau 2-3- Swing/JavaFX (Java) | 30 |
| Tableau 2-4- Windows Forms/WPF (C#) | 30 |
| Tableau 2-5- Electron/NW.js (JavaScript). | 31 |

Introduction générale

L'enseignement des travaux pratiques (TP) joue un rôle crucial dans la formation des étudiants, leur permettant de mettre en pratique les connaissances théoriques acquises en classe. Cependant, les méthodes traditionnelles d'enseignement des TP montrent souvent leurs limites, notamment en termes d'engagement et d'efficacité pédagogique. Une problématique récurrente dans ce domaine est l'uniformisation des TP, où les étudiants suivent des procédures rigides et peu flexibles, limitant ainsi leur créativité et leur compréhension profonde des concepts.

L'uniformisation des travaux pratiques est essentielle car elle garantit que tous les étudiants suivent les mêmes instructions et atteignent les mêmes objectifs d'apprentissage. Cela assure une équité dans l'évaluation des compétences et favorise le développement de compétences professionnelles clés nécessaires pour réussir dans leur domaine d'études et au-delà.

Pour remédier à cette situation, il est crucial de repenser les méthodes d'enseignement des TP en intégrant des outils technologiques modernes, tels que les interfaces graphiques interactives. Ces outils offrent une opportunité unique de rendre l'apprentissage plus engageant et personnalisé, tout en conservant les avantages de la standardisation. En permettant aux étudiants de visualiser les concepts en temps réel et d'expérimenter directement avec les éléments théoriques, ces interfaces peuvent transformer l'enseignement des TP, le rendant plus dynamique, interactif et efficace.

Ainsi, l'intégration de technologies interactives dans les TP constitue une réponse prometteuse à la problématique de l'uniformisation, ouvrant la voie à une éducation plus riche, plus flexible et plus adaptée aux besoins des étudiants d'aujourd'hui.

Chapitre 1 Uniformiser les Travaux Pratiques

1.1 Introduction

Les travaux pratiques (TP) occupent une place centrale dans l'enseignement scientifique et technique, offrant aux étudiants l'occasion d'appliquer concrètement les concepts théoriques abordés en cours. Cependant, l'importance croissante de ces TP se heurte souvent à des contraintes matérielles et logistiques. Le manque de moyens, qu'ils soient financiers ou humains, et le nombre croissant d'étudiants rendent difficile l'organisation efficace de ces séances. Face à ces défis, notre démarche vise à proposer une solution innovante pour uniformiser l'enseignement de certains TP. Cette solution se veut immédiatement utilisable, facilement diffusable et, surtout, évolutive et adaptable rapidement aux besoins pédagogiques changeants.

1.2 Système LMD :

Le système LMD, qui signifie Licence-Master-Doctorat, est un système d'enseignement supérieur adopté par de nombreux pays à travers le monde. Voici quelques généralités sur ce système :

1.2.1 Structure en trois niveaux :

- **Licence (L)** : Premier cycle universitaire, d'une durée généralement de trois ans, où les étudiants acquièrent des connaissances générales dans leur domaine d'étude.
- **Master (M)** : Deuxième cycle universitaire, d'une durée généralement de deux ans, où les étudiants approfondissent leurs connaissances et se spécialisent dans un domaine spécifique.

- **Doctorat (D) :** Troisième cycle universitaire, d'une durée variable (habituellement de trois à cinq ans), où les étudiants effectuent des recherches originales sous la supervision d'un directeur de thèse pour obtenir le grade de docteur. [1]

1.2.2 Flexibilité et modularité : Le système LMD offre une certaine flexibilité aux étudiants en leur permettant de choisir des modules et des cours en fonction de leurs intérêts et objectifs professionnels.

1.2.3 Système de crédits : Les cours et modules sont généralement évalués en crédits ECTS (European Credit Transfer and Accumulation System) ou un système similaire, permettant la reconnaissance des acquis et la mobilité des étudiants entre les institutions et les pays. Le concept de crédits se résume comme suit :

- Garantir une capitalisation de crédits, au sein de parcours ainsi des reprises d'études.
- Promouvoir des passerelles entre cursus
- Favoriser la validation des acquis de l'expérience ou plus généralement de compétences.
- Permettre une meilleure définition de l'offre globale de formation d'une université en intégrant toutes ses compétences.
 - 1 semestre = 30 crédits - 1 année
 - 1 licence = 3 ans = 6 semestres = 180 crédits
 - 1 master = 5 ans = 10 semestres = 300 crédits
 - Total = 480 crédits (du baccalauréat au Doctorat) [2]

1.2.4 Adaptabilité internationale : Le système LMD vise à faciliter la comparabilité et la reconnaissance des qualifications à l'échelle internationale, favorisant ainsi la mobilité des étudiants et des chercheurs qui veulent acquérir plus de connaissances et expériences.

1.2.5 Orientation professionnelle : Le système LMD met souvent l'accent sur l'employabilité en intégrant des stages, des projets professionnels et des compétences transversales dans les programmes d'études afin de donner aux étudiants une bonne formation qui leurs permettent de s'intégrer facilement dans le monde du travail. [3]

1.2.6 Harmonisation européenne : Initialement développé dans le cadre du processus de Bologne en Europe, le système LMD a été largement adopté dans les pays européens et au-delà, ce qui a contribué à l'harmonisation des systèmes éducatifs et à la création d'un espace européen de l'enseignement supérieur (EEES). [4]

En résumé, le système LMD représente une approche structurée et flexible de l'enseignement supérieur, visant à fournir aux étudiants les compétences et les qualifications nécessaires pour réussir sur le marché du travail mondial tout en favorisant la mobilité et la coopération internationale.

1.3 Le cursus ST :

Le cursus ST, souvent désigné comme "Sciences et Technologies" ou "Science et Technologie", est un domaine d'études qui englobe un large éventail de disciplines scientifiques et technologiques. Voici une vue d'ensemble de ce cursus

1.3.1 Fondements scientifiques : Les étudiants suivent des cours de mathématiques, de physique, de chimie et parfois de biologie pour acquérir une base solide dans les sciences fondamentales. Ces cours fournissent les outils théoriques nécessaires à la compréhension des phénomènes naturels et des principes scientifiques.

1.3.2 Disciplines technologiques : Le cursus ST inclut également des cours axés sur les applications pratiques des sciences, notamment en ingénierie, informatique, électronique, mécanique, génie civil, génie électrique, génie chimique, etc. Ces disciplines permettent aux étudiants de comprendre et faire des analyses afin de développer des solutions technologiques pour des problèmes réels.

1.3.3 Travaux pratiques et projets : Une composante importante du cursus ST est cernée souvent autour des travaux pratiques en laboratoire et les projets de groupe. Cela permet aux étudiants d'appliquer les concepts théoriques appris en classe à des situations pratiques, de développer leurs compétences expérimentales et de travailler en équipe sur des projets concrets. Cette procédure permet aux étudiants de manipuler et utiliser les matériaux technologiques afin d'avoir une idée sur le monde du travail.

1.3.4 Options de spécialisation : À mesure que les étudiants progressent dans leurs cursus, ils ont souvent la possibilité et le choix de se spécialiser dans un domaine particulier en choisissant des options ou des filières spécifiques. Cela leur permet de se concentrer sur des domaines qui les intéressent particulièrement ou qui correspondent à leurs objectifs de carrière.

1.3.5 Stage en entreprise ou projet de fin d'études : De nombreux programmes ST comprennent un stage en entreprise ou un projet de fin d'études, où les étudiants ont l'occasion de mettre en pratique leurs connaissances dans un environnement professionnel réel, souvent sous la supervision d'un parrain désigné par l'entreprise industrielle accueillante.

1.3.6 Débouchés professionnels : Les diplômés en sciences et technologies ont accès à un large éventail de débouchés professionnels dans des secteurs tels que l'ingénierie, la

recherche et développement, l'informatique, la santé, l'énergie, l'environnement, l'aérospatiale, etc. Ils peuvent également poursuivre leurs études en doctorat pour une carrière académique ou de recherche.

En résumé, le cursus ST offre aux étudiants une formation approfondie en sciences et technologies, combinant des connaissances théoriques avec des applications pratiques, ce qui les prépare à des carrières diverses et stimulantes dans des domaines variés de l'industrie, de la recherche et de l'innovation ; de ce fait, ils participent à la croissance et au développement du pays.

1.4 Travaux Pratiques

Les travaux pratiques dans le système LMD revêtent une grande importance pour plusieurs raisons :

- **Application des concepts théoriques** : Les travaux pratiques permettent aux étudiants d'appliquer les concepts théoriques appris en cours à des situations réelles. Cela renforce leur compréhension des phénomènes vibratoires et ondulatoires en leur donnant une expérience pratique.
- **Développement des compétences expérimentales** : Les travaux pratiques aident les étudiants à développer des compétences expérimentales telles que la manipulation d'équipements de mesure, la collecte de données, l'analyse des résultats et la résolution de problèmes techniques. Ces compétences sont précieuses dans de nombreux domaines de l'ingénierie et des sciences.
- **Visualisation des phénomènes** : Les expériences pratiques permettent aux étudiants de visualiser directement les phénomènes vibratoires et ondulatoires. Cela peut rendre ces concepts abstraits plus concrets et plus faciles à comprendre afin d'aboutir aux solutions du problème.
- **Corrélation avec le monde réel** : Les travaux pratiques permettent aux étudiants de voir comment les concepts de vibrations et d'ondes sont utilisés dans des applications du monde réel, telles que l'ingénierie des structures,

l'acoustique, la médecine (par exemple, l'échographie, scanners), les télécommunications, etc. [5]

1.4.1 Encouragement de la créativité : Les travaux pratiques peuvent également encourager la créativité des étudiants en leur permettant de concevoir et de réaliser leurs propres expériences, d'explorer des questions ouvertes et de proposer des solutions innovantes. En collaboration université / entreprise, Ils peuvent même faire des recherches et proposent des solutions aux problèmes des entreprises privés.

1.4.2 Préparation à la vie professionnelle : Les compétences acquises lors des travaux pratiques, telles que l'analyse et la résolution de problèmes, la collaboration en équipe et la communication des résultats, sont essentielles pour réussir dans le monde professionnel, que ce soit dans la recherche, l'industrie ou d'autres domaines.

En résumé, les travaux pratiques dans le système LMD sont essentiels pour compléter l'apprentissage théorique des étudiants, en leur fournissant une expérience concrète et en développant leurs compétences pratiques, ce qui les prépare efficacement à leur future carrière professionnelle dans des domaines variés.

Dans le cadre de notre projet nous nous somme pencher sur le module Vibrations et onde, qui compte parmi les plus important du tronc commun ST, il apparait dans le programme de toutes les filières, vu l'importance des phénomènes vibratoires autant à l'échelles microscopique que macroscopiques, qui apparaissent et intervienne dans quasiment tous les domaines technologiques.

1.5 Conclusion :

En conclusion, l'importance des travaux pratiques dans le cursus éducatif ne peut être sous-estimée, car ils représentent un pont essentiel entre la théorie et la pratique. Pourtant, les contraintes actuelles, notamment le manque de moyens et l'augmentation du nombre d'étudiants, exigent des solutions nouvelles et efficaces. Notre proposition s'inscrit précisément dans cette optique : fournir un outil pratique, diffusible et adaptable pour l'enseignement des TP. Une telle approche non seulement uniformise et améliore la qualité de l'enseignement, mais ouvre également la voie à des évolutions rapides et pertinentes, répondant ainsi aux défis contemporains de l'éducation scientifique et technique.

Chapitre 2 : Choix des Outils de développement

2.1 Introduction :

De nos jours, le développement d'applications avec des interfaces graphiques utilisateur (GUI) conviviales est essentiel pour offrir une expérience agréable aux utilisateurs finaux. Plusieurs langages de programmation populaires disposent d'outils et de bibliothèques puissants pour créer des interfaces graphiques attrayantes et fonctionnelles.

En Python, des bibliothèques comme **Tkinter**, **PyQt**, **Kivy** et **wxPython** permettent aux développeurs de construire des applications de bureau avec des interfaces riches. **Tkinter**, faisant partie de la bibliothèque standard de Python, est souvent le choix privilégié pour les applications simples à modérées en raison de sa simplicité et de sa portabilité multi-plateformes.

Java propose également des bibliothèques **GUI** robustes telles que Swing et **JavaFX**. Swing, bien qu'ancienne, reste une option viable, tandis que **JavaFX** offre des capacités modernes pour créer des interfaces réactives et riches en graphiques.

Pour le développement d'applications Windows, **C#** offre **Windows Forms** et **Windows Presentation Foundation (WPF)**. Windows Forms est adapté aux applications de bureau traditionnelles, tandis que WPF permet de créer des interfaces graphiques plus avancées avec des capacités de conception déclaratives.

Enfin, les technologies web comme **JavaScript**, **HTML** et **CSS** peuvent être utilisées pour développer des applications de bureau multiplateformes grâce à des **frameworks** comme **Electron** et **NW.js**. Ces outils permettent aux développeurs web de tirer parti de leurs compétences existantes pour créer des applications de bureau riches et interactives.

Chaque langage et chaque outil a ses forces et ses faiblesses, et le choix dépendra des besoins spécifiques du projet, des compétences de l'équipe de développement et des contraintes de

déploiement. Une évaluation approfondie des exigences du projet et une comparaison des différentes options disponibles permettront de sélectionner l'outil le plus adapté pour créer une application avec une interface graphique attrayante et fonctionnelle.

2.2 Différents Outils Existants :

Pour le développement d'applications avec des interfaces graphiques utilisateur (GUI) attrayantes et fonctionnelles, divers outils et bibliothèques sont disponibles dans plusieurs langages de programmation. Voici une introduction succincte à certains des outils les plus utilisés

2.2.1 En Python :

Python est un langage de programmation polyvalent et très populaire, apprécié pour sa simplicité et sa lisibilité. Il est largement utilisé dans divers domaines tels que le développement web, l'analyse de données, l'intelligence artificielle, et bien plus encore.

2.2.1.1 Tkinter :

- **Description :** **Tkinter** est la bibliothèque standard de Python pour la création d'interfaces graphiques (GUI). Elle est incluse dans les distributions standard de Python, ce qui la rend facilement accessible sans nécessiter d'installation supplémentaire. **Tkinter** fournit une interface vers le toolkit graphique **Tk**, ce qui permet de créer des applications avec des fenêtres, des boutons, des champs de texte, des menus, et plus encore.
- **Avantages :**
 - Tkinter est souvent recommandé aux débutants en programmation GUI en raison de sa simplicité et de son intégration directe avec Python. Les concepts de base peuvent être appris rapidement, permettant aux nouveaux développeurs de créer des applications fonctionnelles sans une courbe d'apprentissage abrupte.

- Étant inclus avec Python, Tkinter ne nécessite pas d'installation supplémentaire, ce qui simplifie son utilisation sur différentes plateformes. Les programmes Tkinter écrits sur une plateforme fonctionnent généralement de manière identique sur les autres plateformes majeures.
- La bibliothèque Tkinter est bien documentée, avec de nombreuses ressources disponibles en ligne, y compris des tutoriels, des exemples de code, et des forums d'assistance communautaire. Cela facilite l'apprentissage et la résolution des problèmes que les développeurs peuvent rencontrer.
- **Inconvénients** : Bien que **Tkinter** soit pratique pour les applications simples et moyennement complexes, les interfaces graphiques qu'il produit peuvent sembler désuètes ou rudimentaires par rapport à celles créées avec des bibliothèques plus avancées comme PyQt, Kivy, ou wxPython. Les applications nécessitant des interfaces utilisateur modernes et sophistiquées pourraient bénéficier de l'utilisation de bibliothèques alternatives. [6]

2.2.1.2 PyQt :

- **Description** : **PyQt** est un ensemble de liaisons Python pour le framework Qt, l'un des frameworks GUI les plus complets et professionnels disponibles. Il permet aux développeurs de créer des interfaces graphiques élégantes et riches en fonctionnalités. PyQt combine la robustesse du framework Qt avec la simplicité et la puissance de Python, permettant ainsi de développer des applications GUI modernes et sophistiquées.
- **Avantages** :
 - **Puissance et Flexibilité** : **PyQt** est extrêmement puissant et flexible, permettant de créer des interfaces graphiques complexes et professionnelles. Grâce à ses nombreuses fonctionnalités avancées, les développeurs peuvent concevoir des applications sophistiquées répondant à des besoins spécifiques et variés.
 - **Widgets Avancés** : **PyQt** propose une vaste gamme de widgets, allant de la simple étiquette aux tableaux de bord complexes. Il inclut des capacités avancées telles que les graphiques 2D et 3D, les animations, les vues en arbre, et plus encore. Cela permet de créer des interfaces utilisateur très interactives et visuellement attractives.

- **Outils de Conception : Qt Designer**, un outil WYSIWYG (What You See Is What You Get), permet de concevoir visuellement des interfaces utilisateur. Cela facilite grandement le développement en permettant aux développeurs de voir directement à quoi ressemblera l'interface et d'apporter des modifications en temps réel sans écrire de code.
- **Multiplateforme** : Les applications PyQt peuvent fonctionner sur diverses plateformes, y compris Windows, macOS et Linux, sans nécessiter de modifications majeures du code. Cela permet aux développeurs de créer des applications portables et d'atteindre un public plus large avec un seul code base.
- **Inconvénients :**
 - **Courbe d'Apprentissage : PyQt** a une courbe d'apprentissage plus raide, en particulier pour les débutants. La richesse de ses fonctionnalités et la complexité de ses capacités peuvent rendre son apprentissage plus difficile par rapport à des bibliothèques plus simples comme Tkinter.
 - **Licence : PyQt** est sous licence GPL, ce qui signifie que pour les applications commerciales, une licence commerciale est nécessaire. Cela peut entraîner des coûts supplémentaires pour les développeurs et les entreprises souhaitant utiliser PyQt pour des projets commerciaux. [7]

2.2.1.3 Kivy :

- **Description : Kivy** est une bibliothèque open-source pour la création d'applications multitouch. Elle est particulièrement adaptée pour le développement d'applications sur des appareils mobiles ainsi que des ordinateurs de bureau. Kivy permet de créer des interfaces utilisateur interactives et modernes en utilisant des gestes multitouch et des animations fluides.
- **Avantages :**
 - **Multiplateforme : Kivy** supporte le développement multiplateforme, incluant Android, iOS, Linux, macOS et Windows. Cela permet aux développeurs de créer des applications qui fonctionnent sur différents dispositifs avec un seul code base, facilitant ainsi le déploiement sur plusieurs plateformes sans nécessiter de modifications majeures.

- **Support des Gestes** : **Kivy** prend en charge les gestes multitouch et les interfaces utilisateur interactives. Cette fonctionnalité est particulièrement utile pour les applications mobiles modernes où les interactions par toucher sont courantes. Les développeurs peuvent facilement implémenter des gestes complexes comme le pincement, le glissement et la rotation.
- **Performance** : **Kivy** utilise OpenGL pour le rendu graphique, offrant des performances élevées pour les applications nécessitant des animations fluides et des interfaces graphiques avancées. Cela permet de créer des applications réactives et visuellement attractives avec des animations en temps réel et des effets graphiques sophistiqués.
- **Extensibilité** : La bibliothèque est facilement extensible avec des modules supplémentaires, ce qui permet d'ajouter des fonctionnalités spécifiques selon les besoins du projet. Les développeurs peuvent créer leurs propres widgets et modules pour étendre les capacités de Kivy et personnaliser leurs applications en fonction des exigences spécifiques.
- **Inconvénients** :
 - **Complexité pour des Applications Simples** : Pour des applications de bureau simples, **Kivy** peut être excessif en raison de ses capacités orientées vers les interfaces interactives et les animations avancées.
 - **Ressources et Documentation** : Bien que Kivy ait une communauté active, elle dispose de moins de ressources et de documentation comparé à d'autres bibliothèques comme Tkinter ou PyQt. [8]

2.2.1.4 wxPython :

- **Description** : **wxPython** est une bibliothèque Python pour la création d'interfaces graphiques (GUI) qui utilise le framework wxWidgets. Elle permet aux développeurs de créer des applications avec des interfaces utilisateur natives sur plusieurs plateformes, y compris Windows, macOS et Linux. wxPython fournit une interface Pythonique pour les éléments de l'interface utilisateur de wxWidgets, permettant de créer des applications GUI de manière intuitive et efficace.
- **Avantages** :

- **Apparence Native : wxPython** utilise les contrôles natifs de chaque système d'exploitation, ce qui donne aux applications une apparence et une sensation authentiques sur chaque plateforme. Cela garantit que les applications développées avec wxPython s'intègrent bien avec les autres applications du système et respectent les conventions d'interface utilisateur propres à chaque plateforme.
 - **Richesse en Widgets : wxPython** offre une vaste gamme de widgets et de contrôles, y compris des boutons, des cases à cocher, des champs de texte, des listes, des arborescences, des dialogues et bien plus. Cela permet de créer des interfaces utilisateur riches et fonctionnelles sans nécessiter de bibliothèques supplémentaires.
 - **Multiplateforme** : Les applications wxPython peuvent être déployées sur diverses plateformes, y compris Windows, macOS et Linux, sans nécessiter de modifications majeures du code. Cela permet de développer une seule base de code qui fonctionne de manière cohérente sur plusieurs systèmes d'exploitation.
 - **Inconvénients :**
 - **Courbe d'Apprentissage** : Bien que wxPython soit puissant et flexible, il peut avoir une courbe d'apprentissage plus raide, surtout pour les développeurs débutants. La richesse de ses fonctionnalités et la complexité de certaines de ses options peuvent nécessiter du temps pour maîtriser pleinement la bibliothèque.
 - **Performances** : Les performances de wxPython peuvent parfois être inférieures à celles d'autres bibliothèques, en particulier pour les applications nécessitant des animations fluides ou des mises à jour rapides de l'interface utilisateur. Cela peut limiter son utilisation dans des contextes nécessitant une réactivité élevée.
- [9]

2.2.2 En Java :

Java est un langage de programmation de haut niveau, orienté objet et indépendant de la plateforme, développé par Sun Microsystems (maintenant détenu par Oracle Corporation) et publié pour la première fois en 1995. Il est conçu pour être simple, sécurisé et portable, permettant aux développeurs d'écrire du code qui peut être exécuté sur n'importe quelle machine équipée d'une machine virtuelle Java (JVM).

2.2.2.1 Swing :

- **Description :** **Swing** est une bibliothèque graphique pour Java, faisant partie de l'API Java Foundation Classes (JFC). Elle permet de créer des interfaces utilisateur graphiques (GUI) riches et interactives pour des applications Java. Swing offre un ensemble de composants graphiques (widgets) pour construire des interfaces utilisateur, tels que des boutons, des champs de texte, des tableaux, des listes déroulantes, des menus, etc.
- **Avantages :**
 - **Indépendance de la Plateforme :** Les applications Swing sont indépendantes de la plateforme car elles sont écrites en Java. Elles peuvent s'exécuter sur n'importe quel système d'exploitation disposant d'une JVM, comme Windows, macOS et Linux, offrant ainsi une grande portabilité.
 - **Richesse en Composants :** **Swing** propose une large gamme de composants graphiques avancés et personnalisables. Cela inclut non seulement les composants de base comme les boutons et les labels, mais aussi des composants plus complexes comme les tableaux (JTable), les arbres (JTree) et les éditeurs de texte (JTextPane).
 - **Extensibilité et Personnalisation :** Les composants Swing sont hautement extensibles. Les développeurs peuvent facilement créer des composants personnalisés en étendant les composants existants ou en créant de nouveaux composants à partir de zéro. De plus, Swing permet de personnaliser l'apparence des composants grâce à la prise en charge du Look and Feel (L&F).
- **Inconvénients :**

- **Performance** : Bien que Swing soit riche en fonctionnalités, il peut être plus lent que les bibliothèques natives car il est entièrement écrit en Java et repose sur la JVM. Les applications Swing peuvent parfois souffrir de performances réduites, notamment lorsqu'elles gèrent des interfaces très complexes ou lourdes en ressources.
 - **Complexité** : Pour les développeurs débutants, Swing peut paraître complexe à utiliser en raison de la richesse de ses fonctionnalités et de la multitude de classes et d'interfaces qu'il propose. La conception d'interfaces graphiques avancées peut nécessiter une courbe d'apprentissage importante.
- [10]

2.2.2.2 JavaFX :

- **Description** : **JavaFX** est une plateforme de développement d'applications riches (RIA) en Java, introduite par Oracle pour succéder à Swing. Elle permet de créer des interfaces utilisateur modernes, avec des graphismes avancés, des animations, et du multimédia. JavaFX offre une bibliothèque de composants graphiques, des outils de conception, et une infrastructure robuste pour le développement d'applications de bureau et mobiles.
- **Avantages** :
 - **Interface Utilisateur Moderne** : **javaFX** permet de créer des interfaces utilisateur modernes avec des composants graphiques avancés, des animations et des effets visuels. Les développeurs peuvent utiliser des styles CSS pour personnaliser l'apparence des composants, facilitant la création de GUI attractives et réactives.
 - **Support Multimédia** : **JavaFX** prend en charge la lecture de vidéos, d'audios, et d'images, ainsi que l'intégration de contenus Web grâce à son composant WebView. Cela permet de créer des applications riches en contenu multimédia sans nécessiter de bibliothèques supplémentaires.
 - **Scène Graphique 2D et 3D** : **JavaFX** offre une API graphique puissante pour dessiner des formes 2D et 3D, appliquer des transformations, des effets visuels et des animations. Cela permet de créer des visualisations de données complexes et des interfaces utilisateur dynamiques.
 - **Bindings et Observables** : **JavaFX** introduit le concept de propriétés observables et de liaisons de données (bindings), facilitant la synchronisation

automatique de l'état de l'interface utilisateur avec les données sous-jacentes. Cela simplifie le développement et améliore la réactivité des applications.

- **Inconvénients :**

- **Complexité :** Pour les débutants, JavaFX peut sembler complexe en raison de ses fonctionnalités avancées et de la richesse de son API. La courbe d'apprentissage peut être raide pour ceux qui ne sont pas familiers avec la programmation GUI ou Java en général.
- **Adoption Limitée :** Malgré ses avantages, JavaFX n'a pas été aussi largement adopté que d'autres technologies GUI comme Swing ou des frameworks web modernes. Certaines entreprises et projets peuvent donc préférer des alternatives mieux établies. [11]

2.2.3 En C# :

C# (prononcé "C sharp") est un langage de programmation moderne, orienté objet, développé par Microsoft dans le cadre de sa plateforme .NET. Lancé en 2000, C# a été conçu pour être simple, puissant, et adapté à une large gamme de tâches de développement, allant des applications de bureau aux services web, en passant par les applications mobiles et les jeux vidéo. C# combine la puissance de langages comme C++ avec la simplicité et la productivité de langages comme Visual Basic.

2.2.3.1 Windows Forms :

- **Description :** Windows Forms, souvent abrégé en WinForms, est une bibliothèque graphique incluse dans le .NET Framework et .NET Core de Microsoft. Elle permet de créer des applications de bureau avec des interfaces utilisateur graphiques (GUI) sous Windows. WinForms fournit une collection de classes et de contrôles pour concevoir des interfaces utilisateur interactives et gérer les événements associés.
- **Avantages :**
 - **Facilité d'Utilisation :** WinForms est simple à apprendre et à utiliser, ce qui le rend accessible aux débutants en développement d'applications de bureau. L'intégration avec Visual Studio, un environnement de développement intégré (IDE) de Microsoft, permet de concevoir des interfaces utilisateur par glisser-déposer, simplifiant ainsi le développement.

- **Large Éventail de Contrôles** : WinForms propose une vaste collection de contrôles standard comme les boutons, les labels, les boîtes de texte, les listes déroulantes, les grilles de données, etc. Ces contrôles facilitent le développement rapide d'interfaces utilisateur complètes et fonctionnelles.
- **Intégration avec le .NET Framework** : Étant une partie intégrante du .NET Framework, WinForms bénéficie de l'ensemble des bibliothèques et fonctionnalités offertes par le framework, incluant l'accès aux bases de données, les services web, la gestion de fichiers, etc.
- **Inconvénients** :
 - **Limité aux Applications Windows** : WinForms est principalement conçu pour le développement d'applications de bureau sous Windows. Bien que des projets .NET Core puissent théoriquement être exécutés sur d'autres systèmes d'exploitation, les applications WinForms restent généralement limitées à l'environnement Windows.
 - **Apparence et Fonctionnalités Dépassées** : Comparé à des frameworks plus modernes comme WPF (Windows Presentation Foundation) ou UWP (Universal Windows Platform), WinForms peut sembler limité en termes de fonctionnalités et d'apparence. Les applications WinForms peuvent avoir une apparence plus vieillotte et manquer de certaines fonctionnalités avancées disponibles dans les technologies plus récentes.
 - **Performances** : Pour des applications graphiquement intensives ou nécessitant des animations fluides, WinForms peut ne pas offrir les performances optimales comparées à des technologies comme WPF qui utilisent DirectX pour le rendu graphique. [12]

2.2.3.2 WPF (Windows Presentation Foundation) :

- **Description** : **Windows Presentation Foundation (WPF)** est une bibliothèque graphique développée par Microsoft pour la création d'applications de bureau riches en fonctionnalités sous Windows. Introduite avec .NET Framework 3.0, WPF utilise DirectX pour le rendu graphique, offrant des capacités graphiques avancées, des animations, et un modèle de programmation déclaratif via XAML (Extensible Application Markup Language).
- **Avantages** :

- **Graphismes Avancés** : WPF utilise DirectX pour le rendu graphique, permettant de créer des interfaces utilisateur avec des graphismes avancés, des effets visuels, et des animations fluides. Cela le rend idéal pour des applications nécessitant des rendus graphiques sophistiqués.
- **Séparation du Contenu et de la Présentation** : Grâce à XAML, WPF permet une séparation claire entre la logique de l'application et sa présentation. Cela facilite le développement collaboratif, permettant aux développeurs de se concentrer sur la logique de l'application et aux designers de travailler sur l'apparence et la mise en page.
- **Data Binding Puissant** : WPF propose un modèle de liaison de données (data binding) robuste qui simplifie la synchronisation des interfaces utilisateur avec les données sous-jacentes. Il prend en charge les liaisons unidirectionnelles, bidirectionnelles, et les notifications de changement de propriété, rendant les interfaces utilisateur réactives et dynamiques.
- **Inconvénients** :
 - **Courbe d'Apprentissage** : WPF est plus complexe que WinForms en raison de ses nombreuses fonctionnalités avancées et de l'utilisation de XAML pour la conception d'interfaces. Les développeurs doivent investir du temps pour maîtriser les concepts et les outils de WPF.
 - **Performance** : Bien que WPF offre des capacités graphiques avancées, il peut être gourmand en ressources, surtout pour les applications graphiquement intensives. Les performances peuvent varier en fonction du matériel et de la complexité des interfaces utilisateur.
 - **Compatibilité** : WPF est conçu principalement pour les applications de bureau Windows. Bien qu'il puisse être utilisé avec .NET Core pour une compatibilité multiplateforme partielle, son utilisation est généralement limitée à l'environnement Windows. [13]

2.2.4 En JavaScript/HTML/CSS :

JavaScript, **HTML** (HyperText Markup Language), et **CSS** (Cascading Style Sheets) forment le trio de base des technologies web, permettant de créer des pages web interactives et stylisées. Ensemble, ils constituent les fondations du développement web front-end.

- **HTML** : Utilisé pour structurer le contenu d'une page web. Il définit les éléments tels que les titres, les paragraphes, les liens, les images, et autres composants de la page.
- **CSS** : Utilisé pour styliser le contenu défini par HTML. Il contrôle l'apparence visuelle des éléments, incluant les couleurs, les polices, les marges, les alignements, et les animations.
- **JavaScript** : Utilisé pour ajouter de l'interactivité et de la dynamique aux pages web. Il permet de manipuler les éléments HTML et CSS, de gérer les événements utilisateur, de communiquer avec des serveurs, et bien plus encore.

2.2.4.1 Electron :

- **Description** : **Electron** est un framework open-source développé par GitHub qui permet de créer des applications de bureau multiplateforme en utilisant des technologies web telles que JavaScript, HTML et CSS. Electron combine le moteur de rendu de Chromium et l'environnement d'exécution Node.js pour offrir aux développeurs la possibilité de créer des applications de bureau avec les mêmes technologies qu'ils utilisent pour le développement web.
- **Avantages** :
 - **Multiplateforme** : Electron permet de développer des applications qui fonctionnent sur Windows, macOS et Linux à partir d'une seule base de code. Cela réduit le temps et les ressources nécessaires pour maintenir des versions séparées pour chaque plateforme.
 - **Technologies Web** : Les développeurs peuvent utiliser JavaScript, HTML et CSS, des technologies qu'ils connaissent déjà, pour créer des applications de bureau. Cela facilite la transition pour les développeurs web vers le développement d'applications de bureau.
 - **Large Écosystème** : Grâce à l'intégration avec Node.js, Electron permet d'accéder à une vaste collection de modules npm, enrichissant les capacités

des applications avec des fonctionnalités supplémentaires comme l'accès au système de fichiers, la gestion des bases de données, et plus encore.

- **Développement Rapide** : L'utilisation de frameworks et bibliothèques JavaScript populaires comme React, Angular ou Vue.js avec Electron accélère le développement d'applications de bureau complexes et interactives.

- **Inconvénients :**

- **Consommation de Ressources** : Les applications Electron peuvent être gourmandes en ressources (mémoire et CPU) car chaque application inclut une instance complète de Chromium. Cela peut entraîner des performances moins optimales comparées aux applications natives.
- **Taille des Applications** : Les applications Electron ont tendance à être plus volumineuses en termes de taille de fichier en raison de l'inclusion du moteur Chromium et de Node.js. Cela peut être un inconvénient pour les utilisateurs ayant des limitations de stockage ou de bande passante.
- **Sécurité** : Les applications Electron peuvent être exposées à des vulnérabilités de sécurité communes aux applications web. Les développeurs doivent être vigilants et suivre les meilleures pratiques de sécurité pour protéger les utilisateurs et leurs données. [14]

2.2.4.2 NW.js :

- **Description** : **NW.js** est un framework open-source permettant de développer des applications de bureau multiplateforme en utilisant des technologies web telles que HTML, CSS et JavaScript, combinées avec Node.js pour accéder aux fonctionnalités système. Initialement connu sous le nom de node-webkit, NW.js permet aux développeurs de créer des applications de bureau qui s'exécutent à la fois sur Windows, macOS et Linux à partir d'une seule base de code.
- **Avantages :**
 - **Multiplateforme** : **NW.js** permet de créer des applications de bureau qui fonctionnent sur différentes plateformes sans nécessiter de modifications importantes du code source. Cela simplifie le processus de développement et de maintenance pour les développeurs.
 - **Utilisation de Technologies Web** : Les développeurs peuvent utiliser des technologies web standard telles que HTML, CSS et JavaScript pour créer

des interfaces utilisateur et des fonctionnalités d'application. NW.js offre une intégration fluide avec ces technologies bien connues et largement utilisées.

- **Accès aux Fonctionnalités Système** : Grâce à l'intégration avec Node.js, NW.js permet d'accéder aux fonctionnalités du système d'exploitation, telles que les fichiers, les processus, le réseau, et plus encore. Cela permet de développer des applications de bureau riches en fonctionnalités.
- **Performance** : Les applications NW.js bénéficient de bonnes performances grâce à l'utilisation du moteur de rendu WebKit (ou Blink) et de Node.js. Cela permet d'offrir une expérience utilisateur fluide, même pour des applications graphiquement intensives ou demandant une manipulation de données rapide.
- **Inconvénients** :
 - **Consommation de Ressources** : Comme pour Electron, NW.js peut être gourmand en ressources en raison de l'inclusion du moteur de rendu web complet et de Node.js dans chaque application. Cela peut affecter les performances sur des systèmes avec des spécifications matérielles limitées.
 - **Taille des Applications** : Les applications NW.js peuvent avoir une taille de fichier plus importante en raison de l'inclusion du moteur de rendu et de Node.js. Cela peut influencer le temps de téléchargement et de mise à jour des applications, ainsi que la consommation d'espace disque pour les utilisateurs.
 - **Sécurité** : En raison de l'utilisation de moteurs de rendu web complets, les applications NW.js peuvent potentiellement être exposées à des vulnérabilités de sécurité similaires à celles rencontrées dans les navigateurs web. [15]

2.3 Justification du Choix de Tkinter :

Le choix de Tkinter pour le développement d'interfaces graphiques (GUI) dans notre projet peut être motivé par plusieurs facteurs clés. Tkinter, en tant que bibliothèque standard incluse avec Python, offre une simplicité et une facilité d'utilisation idéales pour notre équipe, permettant un démarrage rapide sans complications liées à l'installation de packages supplémentaires. De plus, la vaste documentation et la communauté active autour de Tkinter nous fourniront un soutien précieux tout au long du développement. Sa portabilité sur diverses

plateformes garantit que notre application pourra atteindre un public plus large sans nécessiter de modifications majeures du code. Enfin, la compatibilité de Tkinter avec d'autres bibliothèques Python nous permettra d'intégrer facilement des fonctionnalités supplémentaires, rendant notre application encore plus robuste et polyvalente. Voici une justification plus détaillée qui prend en compte divers aspects de l'utilisation de Tkinter :

2.3.1 Simplicité et Accessibilité :

Tkinter est inclus dans la distribution standard de Python, rendant son installation et utilisation immédiates et simples. Cela réduit le temps et l'effort nécessaires pour démarrer un projet GUI. Pour les débutants ou les projets nécessitant une mise en œuvre rapide, Tkinter offre une courbe d'apprentissage douce par rapport à des outils plus complexes comme PyQt ou JavaFX

2.3.2 Documentation et Communauté :

Dans le cadre de notre projet, l'utilisation de Tkinter pour le développement d'interfaces graphiques bénéficie d'une documentation détaillée et d'une communauté active, ce qui constitue un atout significatif pour notre équipe de développement. [16]

2.3.2.1 Documentation :

- **Documentation Officielle de Python :** La documentation officielle de Python inclut une section complète sur Tkinter, détaillant chaque aspect de la bibliothèque, depuis les concepts de base jusqu'aux fonctionnalités avancées. Cette ressource est notre référence principale pour comprendre et utiliser efficacement Tkinter dans notre projet.
- **Exemples et Tutoriels en Ligne :** Nous avons accès à une multitude d'exemples et de tutoriels disponibles en ligne. Ces ressources pratiques nous aident à implémenter des fonctionnalités spécifiques, comme l'intégration de widgets personnalisés ou la gestion des événements, en suivant des cas d'utilisation réels.
- **Guides de Bonnes Pratiques :** En complément des tutoriels, nous consultons des guides de bonnes pratiques qui nous aident à optimiser notre utilisation de Tkinter. Ces guides abordent des sujets tels que la structuration du code, l'organisation des interfaces utilisateur, et la gestion efficace des ressources.

2.3.2.2 Communauté

- **Forums et Groupes de Discussion** : Nous participons activement aux forums de discussion sur Tkinter, où nous posons des questions, partageons des expériences et trouvons des solutions à des défis spécifiques rencontrés lors du développement. Cela nous permet de bénéficier de l'expertise collective et de résoudre rapidement les problèmes rencontrés.
- **Collaboration avec la Communauté** : Nous contribuons également à la communauté en partageant nos propres solutions et en apportant des réponses aux questions d'autres développeurs. Cette collaboration mutuelle renforce notre compréhension de Tkinter et enrichit l'écosystème de la bibliothèque.

2.3.2.3 Avantages pour Notre Projet

- **Rapidité de Développement** : En utilisant la documentation détaillée et les ressources de la communauté, nous pouvons développer rapidement des interfaces graphiques robustes et fonctionnelles pour notre application. Cela nous permet de respecter les délais et d'itérer rapidement sur les fonctionnalités.
- **Fiabilité et Cohérence** : La documentation officielle garantit que nous utilisons Tkinter de manière conforme aux normes et aux meilleures pratiques recommandées par Python. Cela assure la fiabilité et la cohérence de notre code à travers différents modules de l'application.
- **Support Continu** : Grâce à la communauté active, nous avons accès à un support continu pour résoudre les problèmes éventuels et pour obtenir des conseils sur l'amélioration de nos implémentations Tkinter. Cela minimise les interruptions et maintient notre progression de développement sans heurts.

En conclusion, l'accès à une documentation complète et à une communauté engagée renforce notre capacité à utiliser efficacement Tkinter dans notre projet. Ces ressources nous permettent d'exploiter pleinement les fonctionnalités de Tkinter tout en bénéficiant du soutien d'une communauté dynamique et collaborative.

2.3.3 Portabilité :

Dans le cadre de notre projet, l'utilisation de Tkinter pour le développement d'interfaces graphiques présente plusieurs avantages en termes de portabilité, adaptabilité et facilité de déploiement.

2.3.3.1 Compatibilité Multiplateforme :

- **Support Universel :**

- Tkinter étant intégré à la bibliothèque standard de Python, notre application sera compatible et disponible sur toutes les plateformes principales, y compris Windows, macOS et Linux.
- Cette uniformité garantit que les fonctionnalités et l'expérience utilisateur de notre application seront cohérentes quel que soit le système d'exploitation utilisé par nos utilisateurs.

2.3.3.2 Facilité de Distribution et Absence de Dépendances Externes :

- **Simplicité de Déploiement :**

- En utilisant Tkinter, nous évitons les complexités liées à la gestion des dépendances externes. Il suffit d'inclure Tkinter, qui est déjà inclus dans Python, dans les packages d'installation de notre application.
- Cela simplifie le processus de déploiement et réduit les risques d'erreurs ou d'incompatibilités lors de l'installation sur les systèmes des utilisateurs finaux.

2.3.3.3 Adaptabilité aux Environnements Variés :

- **Flexibilité de Déploiement :**

- Tkinter s'adapte facilement à divers environnements de développement et de déploiement, ce qui est essentiel pour notre projet qui pourrait nécessiter des déploiements sur des infrastructures variées, comme des postes de travail individuels ou des serveurs.

2.3.3.4 Avantages Stratégiques :

- **Focus sur le Développement Fonctionnel :**
 - En utilisant Tkinter, notre équipe peut se concentrer sur le développement des fonctionnalités et des aspects métier de notre application, sans être entravée par des défis liés à la compatibilité des plateformes.
 - Cela augmente notre efficacité et nous permet de livrer des fonctionnalités plus rapidement tout en maintenant une haute qualité de l'expérience utilisateur.

2.3.3.5 Cas d'Utilisation Spécifiques :

- **Applications Internes et Éducatives :**
 - Tkinter est particulièrement adapté pour les applications internes à l'entreprise et éducatives, où la portabilité et la facilité d'utilisation sont essentielles.
 - Nous pouvons envisager des scénarios tels que des outils de gestion interne, des interfaces pour la formation et l'éducation, ainsi que des prototypes rapides pour tester des concepts.

En résumé, l'adoption de Tkinter dans notre projet garantit une portabilité sans faille, une facilité de déploiement et une adaptation flexible aux divers environnements d'utilisation. Ces avantages contribuent à notre objectif de développer une application GUI robuste et performante tout en optimisant notre processus de développement et de déploiement.

2.3.4 Adéquation pour les Applications Simples à Modérées :

Dans le cadre spécifique de notre projet, l'adéquation de Tkinter pour les applications simples à modérées se définit par sa capacité à répondre efficacement aux besoins spécifiques de développement d'interfaces graphiques. Voici ce que cela implique pour notre contexte [17]

2.3.4.1 Applications Simples :

- **Scénarios d'Utilisation :**
 - Nous envisageons d'intégrer Tkinter pour développer des interfaces graphiques pour des outils internes, des utilitaires de gestion de données simples et des applications éducatives.
- **Caractéristiques :**
 - **Interface Utilisateur Basique : Tkinter** nous permettra de créer des interfaces simples avec des éléments standard tels que des boutons, des champs de texte et des listes déroulantes.
 - **Fonctionnalités Limitées :** Nous nous concentrerons sur des fonctionnalités de base sans nécessiter d'intégrations complexes avec d'autres systèmes ou bases de données.
 - **Rapidité de Développement : Tkinter** facilitera le développement rapide de prototypes et d'applications simples en minimisant la complexité du code.

2.3.4.2 Applications Modérées :

- **Exigences Supplémentaires :**
 - Pour des parties de notre projet nécessitant des interfaces plus élaborées, Tkinter nous permettra de créer des mises en page plus sophistiquées et d'intégrer des fonctionnalités plus avancées.
- **Caractéristiques :**
 - **Interfaces Plus Complexes :** Nous utiliserons Tkinter pour concevoir des interfaces utilisateur avec des mises en page plus structurées, des widgets personnalisés et une navigation utilisateur améliorée.
 - **Fonctionnalités Étendues :** L'ajout de fonctionnalités telles que des graphiques interactifs, des formulaires complexes, et la gestion de données plus avancée sera facilité par Tkinter.
 - **Adaptabilité : Tkinter** nous permettra de répondre à des scénarios d'utilisation divers, tout en assurant une expérience utilisateur cohérente et efficace.

2.3.4.3 Avantages pour Notre Projet

- **Simplicité et Accessibilité :** Tkinter simplifie le processus de développement d'interfaces graphiques en offrant une syntaxe intuitive et une intégration transparente avec Python.

- **Flexibilité** : Nous pouvons utiliser Tkinter pour développer à la fois des applications simples et des interfaces plus complexes, ce qui correspond aux différents besoins et exigences de notre projet.
- **Rapidité de Déploiement** : En utilisant Tkinter, nous pourrions itérer rapidement sur les prototypes et livrer des fonctionnalités clés à nos utilisateurs finaux dans des délais raisonnables.

En conclusion, **Tkinter** se révèle être un choix approprié pour notre projet en raison de sa capacité à gérer efficacement une gamme variée d'exigences d'interface utilisateur, que ce soit pour des applications simples nécessitant une mise en œuvre rapide ou des applications modérées nécessitant des fonctionnalités plus avancées et une interface utilisateur plus complexe.

2.3.5 Comparaison des outils :

La création d'interfaces graphiques (GUI) est une composante essentielle du développement d'applications modernes. Les interfaces graphiques permettent aux utilisateurs d'interagir de manière intuitive avec les logiciels, facilitant ainsi leur utilisation et augmentant l'efficacité. Il existe une multitude d'outils et de frameworks disponibles pour le développement de GUI, chacun offrant des fonctionnalités spécifiques, des avantages et des inconvénients. Ce document vise à comparer certains des outils les plus populaires utilisés pour la création d'interfaces graphiques, afin d'aider les développeurs à choisir l'outil le mieux adapté à leurs besoins. [18]

Tableau 2-1- Tkinter

| Critère | Évaluation |
|----------------------------------|--|
| Facilité d'apprentissage | Facile à apprendre, idéal pour débutants |
| Complexité des fonctionnalités | Fonctionnalités de base suffisantes |
| Intégration avec le langage | Intégré avec Python, syntaxe claire |
| Portabilité | Multiplateforme (Windows, macOS, Linux) |
| Ressources nécessaires | Léger en termes de ressources système |
| Utilisation pour projets simples | Très pratique pour des projets simples |
| Avantages spécifiques | Simplicité, intégration facile avec Python |
| Limites spécifiques | Fonctionnalités limitées pour des applications complexes |

Tableau 2-2- PyQt/Kivy/wxPython

| Critère | Évaluation |
|----------------------------------|--|
| Facilité d'apprentissage | Plus complexe, nécessite plus d'apprentissage |
| Complexité des fonctionnalités | Fonctionnalités plus avancées |
| Intégration avec le langage | Intégré avec python mais plus complexe |
| Portabilité | Multiplateforme (Windows, macOS, Linux) |
| Ressources nécessaires | Modéré |
| Utilisation pour projets simples | Moins pratique pour des projets simples |
| Avantages spécifiques | Richesse fonctionnelle, support de widgets avancés |
| Limites spécifiques | Plus complexe à maîtriser |

Tableau 2-3- Swing/JavaFX (Java)

| Critère | Évaluation |
|----------------------------------|--|
| Facilité d'apprentissage | Courbe d'apprentissage plus rapide |
| Complexité des fonctionnalités | Fonctionnalités avancées |
| Intégration avec le langage | Intégré avec java, plus de configuration |
| Portabilité | Moins portable (dépend de java) |
| Ressources nécessaires | Modéré |
| Utilisation pour projets simples | Moins pratique pour des projets simples |
| Avantages spécifiques | Grande communauté java, outils robustes |
| Limites spécifiques | Courbe d'apprentissage rapide, nécessite plus de configuration |

Tableau 2-4- Windows Forms/WPF (C#)

| Critère | Évaluation |
|----------------------------------|--|
| Facilité d'apprentissage | Courbe d'apprentissage modérée |
| Complexité des fonctionnalités | Fonctionnalités avancées |
| Intégration avec le langage | Intégré avec c#, spécifique à windows |
| Portabilité | Limité à l'environnement windows |
| Ressources nécessaires | Modéré |
| Utilisation pour projets simples | Moins pratique pour des projets simples |
| Avantages spécifiques | Forte intégration avec windows, outils puissants |
| Limites spécifiques | Limitée à windows |

Tableau 2-5- Electron/NW.js (JavaScript)

| Critère | Évaluation |
|----------------------------------|---|
| Facilité d'apprentissage | Courbe d'apprentissage plus rapide |
| Complexité des fonctionnalités | Fonctionnalités avancées |
| Intégration avec le langage | Utilise JavaScript, nécessite une bonne maîtrise des technologies web |
| Portabilité | Multiplateforme (Windows, macOS, Linux) |
| Ressources nécessaires | Plus lourd en termes de ressources |
| Utilisation pour projets simples | Moins pratique pour des projets simples |
| Avantages spécifiques | Support de technologies web modernes |
| Limites spécifiques | Nécessite une bonne maîtrise de JavaScript, applications plus lourdes |

2.4 Conclusion :

En conclusion, Tkinter a été notre choix privilégié pour le développement de notre plateforme de travaux pratiques (TP) en Python, et ce pour plusieurs raisons essentielles.

Tout d'abord, la simplicité d'utilisation de Tkinter est un atout majeur. Sa syntaxe intuitive et sa courbe d'apprentissage douce permettent aux développeurs, qu'ils soient débutants ou expérimentés, de créer des interfaces graphiques avec une relative facilité. Cette simplicité se traduit par une productivité accrue et une réduction des erreurs de programmation, ce qui est particulièrement bénéfique dans un contexte éducatif où les étudiants doivent se concentrer sur l'apprentissage des concepts fondamentaux sans être submergés par la complexité technique.

Ensuite, la disponibilité immédiate de Tkinter est un autre facteur déterminant. Intégré dans la distribution standard de Python, Tkinter ne nécessite aucune installation supplémentaire. Cela simplifie grandement la configuration des environnements de développement et assure une compatibilité immédiate avec tous les systèmes disposant de Python, réduisant ainsi les obstacles techniques et logistiques.

La documentation riche de Tkinter constitue un avantage significatif. Avec une abondance de ressources pédagogiques, de tutoriels, de livres, et d'exemples de code disponibles, les utilisateurs peuvent facilement trouver les informations nécessaires pour résoudre les problèmes qu'ils rencontrent et améliorer leurs compétences. De plus, une communauté active de développeurs utilise et soutient Tkinter, offrant un réseau de support et d'entraide précieux.

Enfin, la portabilité de Tkinter est un atout indéniable. Les applications développées avec cette bibliothèque fonctionnent de manière cohérente sur différents systèmes d'exploitation, y compris Windows, macOS et Linux, sans nécessiter de modifications substantielles du code. Cette compatibilité multiplateforme assure que les applications peuvent être déployées et utilisées de manière uniforme, quel que soit l'environnement utilisateur.

Pour des applications simples à modérées, Tkinter offre un ensemble de fonctionnalités suffisantes tout en étant facile à apprendre et à utiliser. Sa simplicité, sa disponibilité immédiate, sa documentation riche et sa portabilité font de Tkinter une solution idéale pour le développement de notre plateforme de TP, répondant efficacement à nos besoins tout en facilitant l'apprentissage et le développement rapide.

Chapitre 3 : Conception et Développement de l'Interface Graphique

3.1 Introduction :

Dans le domaine du développement logiciel, nous sommes souvent confrontés à la nécessité de créer des interfaces utilisateur conviviales et interactives pour nos applications. L'une des bibliothèques les plus populaires pour accomplir cette tâche en Python est tkinter. Avec tkinter, nous avons accès à un ensemble robuste d'outils et de widgets graphiques qui nous permettent de concevoir et de mettre en œuvre des interfaces utilisateur riches en fonctionnalités.

En utilisant tkinter, nous pouvons créer des fenêtres, des boutons, des zones de texte, des listes déroulantes et bien plus encore, le tout intégré de manière harmonieuse dans nos applications Python. Ce module nous offre également la flexibilité nécessaire pour personnaliser l'apparence et le comportement des éléments d'interface afin de répondre aux besoins spécifiques de nos utilisateurs.

Dans cette introduction, nous explorerons les principes fondamentaux du développement d'interface avec tkinter, en mettant l'accent sur la création de fenêtres, l'ajout de widgets, la gestion des événements utilisateur et la structuration générale de notre application graphique. En fin de compte, nous découvrirons comment tkinter peut non seulement améliorer l'expérience utilisateur de nos programmes, mais aussi simplifier le processus de développement en nous fournissant des outils puissants et intuitifs pour créer des interfaces graphiques interactives.

- **Limites des Méthodes Traditionnelles :**

Malgré leur importance, les méthodes traditionnelles d'enseignement des TP présentent plusieurs limitations :

- Infrastructure et Coûts : Les laboratoires bien équipés sont coûteux à maintenir et peuvent ne pas être disponibles pour tous les étudiants.
- Temps et Accessibilité : Les horaires de laboratoire peuvent être contraignants et ne pas convenir à tous les étudiants.
- Engagement et Interaction : Les TP traditionnels peuvent manquer d'interactivité et d'engagement, rendant l'apprentissage passif.

- **Bénéfices d'une Interface Graphique Interactive :**

L'utilisation d'une interface graphique interactive pour les TP offre plusieurs avantages :

- Accessibilité Accrue : Les étudiants peuvent accéder aux simulations et aux exercices depuis n'importe où, à tout moment.
- Visualisation Avancée : Les concepts abstraits peuvent être visualisés de manière dynamique, ce qui facilite la compréhension.
- Interactivité et Engagement : Les étudiants peuvent interagir avec les simulations, ce qui rend l'apprentissage plus engageant.
- Personnalisation : Les exercices peuvent être adaptés aux besoins individuels des étudiants, permettant un apprentissage plus personnalisé.

- **Objectifs de l'Interface Graphique :**

L'objectif principal de l'interface graphique est de fournir une plateforme interactive qui complète les TP traditionnels et permet aux étudiants de mieux comprendre les concepts de vibrations et d'ondes. Les objectifs spécifiques incluent :

- Simulation de Phénomènes Physiques : Permettre aux étudiants de visualiser et de manipuler des systèmes de vibration et d'ondes.

- Mesure et Analyse : Fournir des outils pour mesurer et analyser les résultats des simulations.
- Feedback Immédiat : Offrir un retour d'information immédiat pour aider les étudiants à corriger leurs erreurs et à comprendre les concepts en profondeur.
- Accessibilité : Assurer que l'interface est accessible sur diverses plateformes, incluant ordinateurs, tablettes, et smartphones.

3.2 Développement de l'interface graphique :

3.2.1 Importation des bibliothèques :

Notre code commence par l'importation de diverses bibliothèques nécessaires à la création et à la gestion de notre interface graphique ainsi qu'à certaines fonctionnalités supplémentaires. Voici une explication détaillée de chaque importation :

- **Tkinter :**

tkinter est la bibliothèque standard pour créer des interfaces graphiques en Python. Elle fournit des outils pour créer des fenêtres, des boutons, des étiquettes, des champs de texte, etc. Les modules `ttk` (Themed Tk) offrent des widgets thématiques, `messagebox` permet de montrer des boîtes de dialogue et `PhotoImage` gère les images.

- **Pandas :**

Pandas est une bibliothèque utilisée pour la manipulation et l'analyse de données, particulièrement utile pour travailler avec des structures de données comme les DataFrame.

- **customtkinter :**

Customtkinter est une bibliothèque basée sur `tkinter` qui offre des widgets personnalisés et améliorés pour une apparence plus moderne et flexible.

- **matplotlib :**

matplotlib est une bibliothèque pour créer des graphiques en 2D. Le module ``pyplot`` est couramment utilisé pour créer des graphiques de manière simple.

- **reportlab :**

reportlab est une bibliothèque pour générer des documents PDF. Les modules importés ici permettent de créer et de styler des documents, de gérer la mise en page, et d'incorporer des images et des tableaux.

- **os :**

os est une bibliothèque standard pour interagir avec le système d'exploitation. Elle permet de gérer des fichiers, des répertoires, et d'exécuter des commandes système.

- **webbrowser :**

webbrowser est une bibliothèque pour ouvrir des URL dans un navigateur web par défaut.

- **subprocess :**

subprocess permet de lancer des sous-processus et de gérer leur entrée/sortie/erreur. Utile pour exécuter des commandes système.

- **logging :**

logging` est une bibliothèque pour gérer les messages de log, permettant de suivre et de déboguer le code.

- **PIL (Python Imaging Library) :**

PIL (ou son fork ``Pillow``) est utilisée pour ouvrir, manipuler et enregistrer différents formats d'image. ``ImageTk`` est un sous-module pour interfacer les images avec ``tkinter``.

Chacune de ces bibliothèques a un rôle spécifique et contribue aux différentes fonctionnalités de notre application.

3.2.2 Classe MainApp :

Dans cette section, nous décrivons les différentes méthodes que nous avons créées pour notre application. Ces méthodes sont essentielles pour réaliser les calculs nécessaires et présenter une interface utilisateur intuitive et fonctionnelle.

La classe `MainApp` est une sous-classe de `CTk` (CustomTkinter) qui définit la structure principale de l'interface graphique. Elle gère l'interface utilisateur, le processus d'authentification, et d'autres fonctionnalités.

- **Initialisation de la fenêtre :**

- `super().__init__()` : Appelle le constructeur de la classe parente `CTk`.
- `self.student_logged_in = False` : Initialise une variable pour vérifier si un étudiant est connecté.
- `self.title("Login")` : Définit le titre de la fenêtre.
- `self.geometry("400x400")` : Définit la taille de la fenêtre.
- `self.resizable(False, False)` : Empêche le redimensionnement de la fenêtre.
 - `self.bind('<FocusOut>', self.on_focus_out)` : Associe un événement pour retirer l'attribut `-topmost` lorsque la fenêtre perd le focus.

- **Image de fond :**

Charge une image de fond (`background2.jpg`) et l'affiche en utilisant un `CTkLabel`.

- **Frame pour le formulaire de connexion :**

Crée une `CTkFrame` blanche au centre de l'image de fond pour contenir les champs de connexion.

- **Barre de menu :**

Appelle une méthode (à définir) pour créer une barre de menu.

- **Champs et boutons de connexion :**

Crée des labels, des champs d'entrée (`CTkEntry`), et un bouton de connexion dans la `CTkFrame`.

- **Chargement des données étudiantes :**

Charge les données des étudiants depuis un fichier Excel.

- **Validation de l'entrée :**

Configure la validation de l'entrée pour le champ du matricule.

- **Bind des touches :**

Associe la touche ``Return`` (Entrée) à la méthode d'authentification.

- **Méthode `load_student_data` :**

Charge les données des étudiants depuis un fichier Excel ``students.xlsx``.

Nettoie les données en supprimant les espaces et en convertissant les chaînes en minuscules.

En cas d'erreur, affiche un message d'erreur et retourne un DataFrame vide. (Dans cette méthode nouvellement créée, les noms et les matricules sont saisis dans un fichier Excel qui est relié avec l'application). L'étudiant ne pourra se connecter à l'application qu'après avoir saisi son nom et son matricule.

- **Méthode `student_authenticate` :**

Récupère les entrées de l'utilisateur (le nom et le matricule), ensuite ces données sont comparées par rapport aux données qui sont saisies dans le fichier Excel

Si une correspondance est trouvée, affiche un message de succès, sinon un message d'erreur.

- **Méthode `on_focus_out` :**

Désactive l'attribut ``-topmost`` lorsque la fenêtre perd le focus, permettant à d'autres fenêtres de passer au premier plan.

- **Méthode `show_tp_buttons` :**

Cette méthode affiche une nouvelle fenêtre contenant des boutons pour accéder aux différents TP.

- **Création d'une nouvelle fenêtre :**

Une nouvelle fenêtre `CTkToplevel` est créée avec un titre, une taille fixe, une couleur de fond, et configurée pour rester au-dessus des autres fenêtres.

- **Gestion de la perte de focus :**

Lorsque la fenêtre perd le focus, une méthode de rappel est liée pour retirer l'attribut `-topmost`.

- **Image de fond :**

Charge une image de fond et l'affiche dans la nouvelle fenêtre.

- **Création de vues à onglets :**

Un `CTkTabview` est créé pour contenir les onglets TP1, TP2, et TP3. Chaque onglet est configuré pour avoir une colonne avec un poids de 0.

- **Contenu de l'onglet "TP 1" :**

- **Cadre pour TP1 :** Un cadre (`CTkFrame`) est créé dans l'onglet TP1, contenant des labels et des boutons.
- **Titres et descriptions :** Labels décrivant le TP1 avec des informations sur l'expérience et la loi de Hooke.
- **Bouton TP1 :** Un bouton est créé pour accéder au TP1.
- **Images :** Deux images sont chargées et affichées dans l'onglet TP1.

- **Méthode `open_tp_window` :**

- **Vérification de connexion :** Vérifie si l'étudiant est connecté avant d'ouvrir la fenêtre TP.
- **Commande TP :** Exécute la commande associée au TP si l'utilisateur est connecté.

- **Méthode `create_menu_bar` :**

- **Création de la barre de menu :** Une barre de menu est créée et configurée pour l'application principale.
- **Menu Fichier :** Contient une option pour quitter l'application.

- **Menu Aide** : Contient une option pour afficher un dialogue "À propos".

- **Méthode tp1** :
 - **Création d'une nouvelle fenêtre pour TP 1** :
 - Une fenêtre ``CTkToplevel`` est créée, avec un titre "Tp 1", des dimensions de 400x300, et une couleur de fond
 - **Label et boutons** :
 - Un label est ajouté avec le texte "Choisissez une option pour TP 1 :".
 - Trois boutons sont ajoutés :
 - Manuel : Exécute ``self.open_tp1``.
 - Simulation : Exécute ``self.sim``.
 - Résultats : Exécute ``self.results1``.
 - **Fenêtre au premier plan** :
 - La fenêtre est configurée pour rester au premier plan avec `app1.attributes(-topmost, True)`.
 - Un événement est lié pour détecter la perte de focus et appeler `self.on_tp_focus_out(app1)`.

- **Méthode tp2** :
 - La structure est identique à ``tp1``, mais avec des commandes spécifiques à TP2 (``open_tp2``, ``results2``).

- **Méthode ``open_pdf``** :
 - Détermination du chemin du fichier PDF :
 - Le chemin absolu du fichier PDF est déterminé en utilisant ``os.path.abspath``.
 - Vérification de l'existence du fichier :

- Si le fichier existe, il est ouvert dans le navigateur web par défaut à l'aide de ``webbrowser.open``.
- Si le fichier n'existe pas, un message d'erreur est enregistré et affiché.
- **Méthode ``sim``**
 - Lancement de l'application de simulation :
 - Le chemin absolu de l'exécutable de simulation est déterminé.
 - L'application est lancée avec ``subprocess.Popen``.
 - En cas d'erreur, un message d'erreur est enregistré et affiché.

3.2.3 Méthode ``results1``

Création d'une nouvelle fenêtre pour afficher les résultats de TP1 :

Une fenêtre ``CTkToplevel`` est créée avec un titre "**Résultats**", des dimensions de 1920x1080, et une couleur de fond.

Cette section de notre code met en place l'interface utilisateur pour afficher les résultats du TP1. Elle inclut la création d'une nouvelle fenêtre avec des composants d'entrée, des boutons et une table pour afficher les résultats des calculs.

- **Création de la fenêtre `app_results1` :**
 - **Fenêtre principale :** Une nouvelle fenêtre de niveau supérieur (``CTkToplevel``) est créée pour afficher les résultats du TP1. Cette fenêtre est configurée pour être toujours au premier plan avec ``attributes('-topmost', True)``.
 - **Détection de perte de focus :** Un événement est lié pour détecter lorsque la fenêtre perd le focus.
- **Cadre principal `main_frame1` :**
 - Un cadre défilant (``CTkScrollableFrame``) est utilisé pour contenir les éléments de l'interface.

3.2.3.1 Section "Partie Statique" :

Cadre `static_frame` : Contient les éléments de l'interface pour la partie statique du TP1.

- **Question 1 (QSTA_frame) :**

- **Cadre `QSTA_frame` :** Un cadre pour la question spécifique de la partie statique.

Étiquettes et champs de saisie : Inclut les étiquettes et les champs de saisie pour entrer la masse (M) et la différence d'allongement (X).

- **Tableau des résultats :**

- Tableau `treeAS` : Un widget `Treeview` est utilisé pour afficher les résultats sous forme de tableau.
- Barre de défilement : Une barre de défilement verticale est ajoutée pour le tableau.

- **Boutons d'action :**

- Bouton "**Reset**" : Réinitialise les champs de saisie.
- Bouton "**Add to Table**" : Ajoute les données saisies dans le tableau.
- Bouton "**Clear Table**" : Efface les entrées du tableau.
- Boutons de calcul : Calculer le poids (P) et la division poids/distance (P/X).

- **Section "Conclusion" :**

Étiquette et zone de texte : Une étiquette et une zone de texte pour entrer une conclusion.

- **Question 2 (QSTB_frame) :**

- Cadre `QSTB_frame` pour la question 2.
- Étiquette `QSTB_label` pour la question "Que représente P/X ?".
- Zone de texte `text_areaBS` pour entrer la réponse.
- Étiquette `label_instructionBS` pour la dimension.
- Menu déroulant `combo_dimension` pour sélectionner la dimension parmi les options

- **Question 3 (QSTC_frame):**

- Cadre `QSTC_frame` pour la question 3.

- Étiquette ``QSTC_label`` pour la question "Calculer les valeurs des masses inconnues".
 - Étiquette ``label_instructionCS`` pour l'instruction.
 - Vue par onglets ``tabview`` pour saisir les différences d'allongement et calculer les masses pour les trois couleurs de masses.
 - Dans chaque onglet (rouge, verte, marron) :
 - Étiquettes ``label_instruction_tabX`` et ``label_x_tabX`` pour les instructions et la saisie de X.
 - Champ de saisie ``entry_x_tabX`` pour entrer la différence d'allongement.
 - Étiquette ``mass_label_tabX`` pour afficher la masse calculée.
 - Bouton ``calculate_mass_button_tabX`` pour calculer la masse en appelant la fonction ``calculate_mass_from_x_tabX``.
- **Question 4 (QSTD_frame) :**
 - Cadre ``QSTD_frame`` pour la question 4.
 - Étiquette ``QSTD_label`` pour la question "Calculer la constante de raideur".
 - Étiquette ``label_instructionDS`` pour les instructions.
 - Champ de saisie ``entry_mass3`` pour entrer la masse.
 - Champ de saisie ``entry_distance3`` pour entrer la différence d'allongement.
 - Étiquette ``k_label`` pour afficher la constante de raideur.
 - Bouton ``calculate_k_button`` pour calculer la constante de raideur en appelant la fonction ``calculate_k``.
 - Étiquette ``conclusion_labelDS`` pour la conclusion.
 - Zone de texte ``text_areaDS`` pour entrer la conclusion.

3.2.3.2Section "Partie Dynamique" :

Cette section concerne la partie dynamique de notre application. Elle crée deux nouvelles questions dans la section dynamique, permettant aux utilisateurs d'entrer des données, de faire des calculs et d'afficher les résultats.

Le ``dynamic_frame`` contient toutes les questions liées à la partie dynamique de l'application.

`dynamic_label` est l'étiquette principale pour cette section.

- **Question 1 (QSTAD_frame) :**

- `QSTAD_frame` pour la première question dynamique.
- Labels et champs de saisie pour ` X (m)` et ` t (s)`.
- Bouton `calculate_T_button` pour calculer la période ` T `.
- Table pour afficher les résultats avec une `Treeview`.
- Boutons pour ajouter et effacer les données de la table.
- Texte pour la conclusion.

- **Question 4 (QSTDD_frame) :**

- `QSTDD_frame` pour la deuxième question dynamique.
- Labels et champs de saisie pour ` M (kg)` et ` t (s)`.
- Bouton `calculate_T_T_squared_button` pour calculer la période ` T ` et ` T^2 `.
- Table pour afficher les résultats avec une `Treeview`.
- Boutons pour ajouter et effacer les données de la table.
- Texte pour la conclusion.

- **Question 5 (QSTED_frame) :**

- `QSTED_frame` pour tracer le graphique de ` T^2 ` en fonction de ` M `.
- Instructions pour entrer les valeurs de ` M ` et ` T^2 `.
- Boucle pour créer les champs de saisie pour les valeurs de ` M ` et ` T^2 `.
- Bouton `Draw Graph` pour tracer le graphique en utilisant la méthode `draw_graph`.
- Bouton `Reset` pour réinitialiser les champs de saisie en utilisant la méthode `reset_values`.

- **Question 6 (QSTFD_frame) :**
 - `QSTFD_frame` pour le calcul de la pente.
 - Instructions pour entrer les valeurs de `T` et `M`.
 - Champs de saisie pour `T1`, `M1`, `T2` et `M2`.
 - Bouton `Calculer la pente` pour calculer la pente en utilisant la méthode `calculate_slope`.

- **Validation des Champs d'Entrée :**
 - La méthode `configure` est utilisée sur chaque champ d'entrée pour ajouter une validation qui n'accepte que des valeurs numériques.
 - La méthode `validate_only_Num` est enregistrée pour chaque champ d'entrée via `validatecommand`.
 - Cette méthode vérifie si l'entrée est un chiffre ou vide, ce qui permet uniquement les valeurs numériques.

- **Liaison des Événements :**
 - La méthode `bind` est utilisée pour associer la touche `<Return>` (la touche Entrée) à des fonctions spécifiques de calcul.
 - Lorsque l'utilisateur appuie sur la touche Entrée après avoir saisi une valeur, la fonction de calcul correspondante est exécutée.

3.2.4 Méthode `results2`

Cette section crée une nouvelle fenêtre nommée `results2` avec les propriétés et widgets nécessaires pour afficher les résultats d'un deuxième TP (Travail Pratique).

- **Création de la Fenêtre :**
 - `CTkToplevel` est utilisé pour créer une nouvelle fenêtre.

- La fenêtre est configurée avec un titre, une taille, et un fond.
- **Configuration de la Fenêtre :**
 - ``focus_force``, ``lift``, et ``grab_set`` sont utilisés pour s'assurer que la fenêtre est mise en avant et que tous les événements lui sont envoyés.
- **Ajout d'un Cadre Principal Déroulable**
 - ``CTkScrollableFrame`` est utilisé pour permettre le défilement du contenu.

3.2.4.1 Cadre Principal pour les Oscillations Libres Non Amorties

- `CTkFrame(main_frame2)` : Crée un cadre pour contenir les éléments relatifs aux oscillations libres non amorties, ajouté à ``main_frame2``.
- `CTkLabel(non_amortie_frame, text='Oscillations libres non amorties')` : Ajoute un label avec le texte spécifié pour identifier cette section.

3.2.4.1.1 Premier Cas (m=0.5kg et L=2m)

- `CTkFrame(non_amortie_frame, fg_color="white")` : Crée un sous-cadre pour le premier cas.
- `CTkLabel(premier_cas_frame, text='a)Première cas : Pour m=0.5kg et L=2m:')` : Ajoute un label décrivant le premier cas.
- **Instructions et Entrées pour le Premier Cas**
 - `CTkLabel(premier_cas_frame2, text="Pour différentes valeurs de θ° ...")` : Instructions pour l'utilisateur.
 - `CTkLabel(premier_cas_frame2, text="θ°:")` : Label pour l'entrée de l'angle θ .
 - `CTkEntry(premier_cas_frame2, placeholder_text="Theta")` : Champ de saisie pour l'angle θ .
 - `CTkButton(premier_cas_frame2, text="Calculer la Période")` : Bouton pour calculer la période des oscillations en fonction de l'angle θ .
- **Table pour Afficher les Résultats du Premier Cas**
 - `CTkFrame(premier_cas_frame2)` : Cadre pour la table des résultats.
 - `ttk.Treeview(table_frameNA1, columns=("Theta", "Periode"), show="headings", height=6)` : Table pour afficher les valeurs de θ et les périodes correspondantes.
 - `ttk.Scrollbar(table_frameNA1, orient="vertical", command=treeNA1.yview)` : Barre de défilement verticale pour la table.

- **Boutons pour Ajouter/Supprimer des Données dans la Table**
 - **add_to_tableNA1(self, app_results2)** : Fonction pour ajouter une ligne de résultats dans la table.
 - **CTkButton(premier_cas_frame2, text="Add to Table")** : Bouton pour ajouter des données dans la table.
 - **clear_tableNA1()** : Fonction pour effacer toutes les lignes de la table.
 - **CTkButton(premier_cas_frame2, text="Clear Table")** : Bouton pour effacer la table.
- **Conclusion et Zone de Texte pour le Premier Cas**
 - **CTkLabel(premier_cas_frame2, text="Conclusion :")** : Label pour la section de conclusion.
 - **CTkTextbox(premier_cas_frame2, height=50, width=550)** : Zone de texte pour que l'utilisateur puisse écrire une conclusion.

3.2.4.1.2 Deuxième Cas ($\theta=10^\circ$ et $L=2\text{m}$)

Le deuxième cas est similaire au premier, mais

Utilise des masses comme entrée principale au lieu de θ . Voici un extrait de cette section :

- **CTkFrame(non_amortie_frame, fg_color="white")** : Crée un sous-cadre pour le deuxième cas.
- **CTkLabel(deuxieme_cas_frame, text='b)Deuxième cas : Pour $\theta=10^\circ$ et $L=2\text{m}$:')** : Ajoute un label décrivant le deuxième cas.
- **Instructions et Entrées pour le Deuxième Cas**
 - **CTkLabel(deuxieme_cas_frame2, text="Pour différentes valeurs de M...")** : Instructions pour l'utilisateur.
 - **CTkLabel(deuxieme_cas_frame2, text="m(kg):")** : Label pour l'entrée de la masse.
 - **CTkEntry(deuxieme_cas_frame2, placeholder_text="La Masse")** : Champ de saisie pour la masse.
 - **CTkButton(deuxieme_cas_frame2, text="Calculer la Période")** : Bouton pour calculer la période des oscillations en fonction de la masse.

- **Table pour Afficher les Résultats du Deuxième Cas**
 - **CTkFrame(deuxieme_cas_frame2)** : Cadre pour la table des résultats.
 - **-ttk.Treeview(table_frameNA2, columns=("Masse", "Periode"), show="headings", height=4)** : Table pour afficher les valeurs de masse et les périodes correspondantes.
 - **ttk.Scrollbar(table_frameNA2, orient="vertical", command=treeNA2.yview)** : Barre de défilement verticale pour la table.
- **Boutons pour Ajouter/Supprimer des Données dans la Table**
 - **Add_to_tableNA2(self, app_results2)** : Fonction pour ajouter une ligne de résultats dans la table.
 - **CTkButton(deuxieme_cas_frame2, text="Add to Table")** : Bouton pour ajouter des données dans la table.
 - **Clear_tableNA2()** : Fonction pour effacer toutes les lignes de la table.
 - **CTkButton(deuxieme_cas_frame2, text="Clear Table")** : Bouton pour effacer la table.
- **Conclusion et Zone de Texte pour le Deuxième Cas**
 - **CTkLabel(deuxieme_cas_frame2, text="Conclusion :")**** : Label pour la section de conclusion.
 - **CTkTextbox(deuxieme_cas_frame2, height=50, width=550)**** : Zone de texte pour que l'utilisateur puisse écrire une conclusion.

3.2.4.1.3 Troisième cas ($\theta = 10^\circ$; $M=1\text{Kg}$) :

Le troisième cas est défini dans un cadre séparé avec les labels et les entrées nécessaires pour l'expérience.

- **Instructions et Entrées pour le Troisième Cas**

Ces lignes définissent les instructions et les entrées nécessaires pour collecter les données pour différentes longueurs (L).

- **Tableau pour Afficher les Résultats du Troisième Cas**

Ici, nous créons un tableau pour afficher les résultats des périodes et leurs carrés.

- **Boutons pour Ajouter/Supprimer des Données dans la Table**

Ces boutons permettent d'ajouter des données au tableau et de le nettoyer.

- **Entrée des Valeurs pour le Graphique**

Nous créons ici des entrées pour les valeurs de longueur (L) et leurs carrés (T^2) pour le graphique.

- **Calcul de la Pente**

Cela définit un bouton pour calculer la pente à partir des entrées des valeurs de T^2 et de L.

3.2.4.2 Cadre Principal pour les Oscillations Libres Amorties

- On a créé un cadre principal ``amortie_frame`` pour contenir toutes les fonctionnalités liées aux oscillations libres amorties du pendule.
- Ce cadre est positionné dans la fenêtre principale de notre application.

3.2.4.2.1 Éléments d'Interface Utilisateur

- On a utilisé des étiquettes (``amortie_label`` et ``friction_label``) pour afficher des informations sur les oscillations libres amorties, comme le titre et une note supplémentaire.
- Des sous-cadres (``QSTAA_frame``, ``QSTBA_frame``, ``QSTCA_frame``, ``QSTDA_frame``, ``QSTEA_frame``, ``QSTFA_frame``) ont été créés pour organiser différents éléments de l'interface utilisateur de manière logique.

3.2.4.2.2 Collecte de Données

- On a utilisé des champs de texte pour collecter différentes données :
- Pour la période des oscillations (``text_areaA`` dans ``QSTAA_frame2``).

- Pour le nombre de périodes effectuées jusqu'à l'arrêt (`text_areaBA` dans `QSTBA_frame2`).
- Pour les valeurs de θ dans différents scénarios (`entry_thetaA` dans `QSTCA_frame2` et `theta_entries` dans `QSTDA_frame2`).
- Pour les valeurs de $\theta(0)$ et $\theta(T)$ pour calculer le décrétement logarithmique (`entry_theta0` et `entry_thetaT` dans `QSTEA_frame2`).
- Pour le décrétement logarithmique lui-même (`decrement_label` dans `QSTEA_frame2`).
- Pour le facteur d'amortissement (`entry_dec` et `entry_T` dans `QSTFA_frame2`).

3.2.4.2.3 Fonctionnalités de Calcul

- On a inclus des boutons pour déclencher divers calculs :
- Ajouter des données au tableau.
- Effacer le tableau.
- Dessiner un graphique basé sur les valeurs de θ et de T .
- Calculer le décrétement logarithmique et le facteur d'amortissement en fonction des données saisies.

3.2.5 Organisation de l'Interface

- On a organisé les éléments de manière à ce qu'ils soient faciles à comprendre et à utiliser pour l'utilisateur.
- Les différentes parties de l'interface sont clairement délimitées dans des cadres distincts, ce qui facilite la navigation et l'utilisation de l'application.

3.2.6 Interactivité

- Les éléments de l'interface, tels que les boutons et les champs de texte, sont configurés pour interagir avec l'utilisateur et déclencher des actions spécifiques lorsque des données sont saisies ou des boutons sont cliqués.

3.2.7 Validation des Entrées Numériques

- But : Vérifier que l'entrée utilisateur est un nombre positif ou vide.
- Utilisation : Assure que les champs de saisie n'acceptent que des valeurs numériques positives ou sont laissés vides.

3.2.8 Les fonctions :

3.2.8.1 Calcul du Poids

- But : Calculer le poids d'un objet à partir de sa masse.
- Utilisation : Multiplie la masse par la constante gravitationnelle `GRAVITY` pour obtenir le poids en Newtons (N).

3.2.8.2 Calcul de la Distance de Poids

- But : Calculer la force par unité de distance (P/x).
- Utilisation : Divise le poids par la distance pour obtenir la force par unité de distance.

3.2.8.3 Calcul de la Masse à partir de `x`

- But : Calculer la masse en utilisant `x` et une constante.
- Utilisation : Utilise la formule $m = \frac{(p/x) \times x}{g}$ pour calculer la masse.

3.2.8.4 Calcul de la Constante de Raideur `k`

- But : Calculer la constante de raideur d'un ressort.
- Utilisation : Utilise la formule $k = \frac{mg}{x}$ pour calculer `k`.

3.2.8.5 Calcul de la Période `T`

- But : Calculer la période des oscillations.
- Utilisation : Utilise la formule $T = \frac{2\pi}{\sqrt{\frac{k}{x}}}$

3.2.8.6 Calcul de `T` et `T²`

- But : Calculer la période et la période au carré.
- Utilisation : Utilise les mêmes formules pour `T` et calcule ensuite `T²`.

3.2.8.7 Réinitialisation des Valeurs

- But : Effacer le contenu des champs de saisie.

- Utilisation : Parcourt les entrées et supprime le texte pour réinitialiser les valeurs.

3.2.8.8 Calcul de la Pente

- But : Calculer la pente de la courbe de T^2 en fonction de la masse.
- Utilisation : Utilise la formule $= \frac{T_2^2 - T_1^2}{m_2 - m_1}$

3.2.8.9 Calcul de la Période pour TP2

- But : Calculer la période des oscillations pour TP2.
- Utilisation : Calcul similaire à la fonction précédente mais spécifiquement pour TP2.

3.2.8.10 Calcul de la Période pour TP22

- But : Calculer la période des oscillations pour TP22.
- Utilisation : Identique à la fonction précédente mais pour un autre contexte.

3.2.8.11 Calcul de la Période et de T^2 pour TP23

- But : Calculer la période et T^2 pour TP23.
- Utilisation : Utilise la longueur du pendule pour les calculs.

3.2.8.12 Calcul de la Pente pour TP2

- But : Calculer la pente pour TP2.
- Utilisation : Similaire aux autres calculs de pente mais pour TP2.

3.2.8.13 Calcul du Décrément Logarithmique

- But : Calculer le décréement logarithmique.
- Utilisation : Utilise la formule $\delta = \ln\left(\frac{\theta_0}{\theta_T}\right)$

3.2.8.14 Calcul du Facteur d'Amortissement

- But : Calculer le facteur d'amortissement.
- Utilisation : Divise le décréement logarithmique par la période pour obtenir le facteur d'amortissement.

3.2.8.15 Fonction draw_graph

- Cette fonction est responsable de la création d'un graphique à partir des valeurs de masse (m) et de temps au carré (t_squared).
- Elle récupère les valeurs des entrées de masse et de temps au carré, vérifie leur validité et leur cohérence.

- Si les données sont valides, la fonction trace le graphique à l'aide de matplotlib.pyplot et l'enregistre sous forme d'image PNG dans le répertoire actuel avec le nom "graph.png".
- La fonction renvoie le chemin du fichier image du graphique ou None en cas d'erreur.

3.2.8.16 Fonction draw_graph_TP2

- Cette fonction est similaire à draw_graph, mais elle est utilisée pour le graphique du TP2.
- Elle prend en entrée les valeurs de longueur (L) et de temps au carré (T_squared).
- Après avoir vérifié la validité et la cohérence des données, elle trace le graphique $T^2=f(L)$ à l'aide de matplotlib.pyplot.
- À la différence de draw_graph, cette fonction n'enregistre pas le graphique dans un fichier.

Pour le bon fonctionnement de l'application nous avons jugé nécessaire de créer des méthodes selon le concept de fonctionnement que nous avons conçu pour notre application. Ces méthodes sont énumérées comme suit :

- **Classe MainApp**
- **Méthode load_student_data**
- **Méthode student_authenticate**
- **Méthode on_focus_out :**
- **Méthode show_tp_buttons :**
- **Méthode `results1**
- **Méthode `results2**
- **Les fonctions de calculs**

Note : Les autres méthodes font partie de la bibliothèque du tkinter

3.2.9 Quitter l'Application

- But : Fermer l'application.
- Utilisation : Appelle la méthode `quit()` pour arrêter l'application.

3.2.10 Afficher la Boîte de Dialogue "À Propos"

- But : Afficher une boîte de dialogue avec des informations sur l'application.

- Utilisation : Utilise `messagebox.showinfo` pour afficher un message avec des détails sur l'application.

3.3 Conclusion :

En conclusion, tkinter se révèle être une ressource indispensable pour les développeurs Python désireux de créer des interfaces graphiques efficaces et attrayantes. Avec sa panoplie complète de widgets et d'outils personnalisables, tkinter permet non seulement de développer des applications avec une interface utilisateur riche en fonctionnalités, mais aussi de répondre précisément aux besoins variés des utilisateurs. En explorant les principes fondamentaux de la création d'interfaces avec tkinter, nous avons vu comment ce module simplifie le processus de développement tout en améliorant significativement l'expérience utilisateur. En somme, tkinter demeure un choix optimal pour ceux qui cherchent à intégrer des interfaces graphiques interactives et professionnelles dans leurs projets Python.

Chapitre 4 : Implémentation et fonctionnement

4.1 Introduction

Pour mener à bien un projet informatique, il est essentiel de sélectionner des technologies adaptées qui facilitent sa réalisation. Après avoir présenté dans le chapitre précédent les outils utilisés pour notre travail ainsi que la mise en œuvre de ce dernier, nous abordons désormais la phase d'implémentation.

Ce chapitre se consacre à la présentation du fonctionnement de notre application. Nous y détaillons les différentes interfaces utilisateur, illustrant ainsi les interactions possibles entre l'utilisateur et le système. Nous décrivons chaque interface en termes de fonctionnalités, d'accessibilité, et d'ergonomie, montrant comment elles contribuent à une expérience utilisateur optimale.

L'objectif est de fournir une vision claire et exhaustive des composants interactifs de l'application, mettant en lumière les choix technologiques effectués et leur impact sur l'efficacité et la fluidité de l'utilisation. Cette section servira de guide pour comprendre comment les utilisateurs naviguent dans l'application et interagissent avec ses diverses fonctionnalités.

4.2 Présentation des interfaces de notre application :

Les interfaces graphiques de l'application jouent un rôle crucial, car elles facilitent l'interaction entre l'utilisateur et la machine tout en optimisant les performances de l'application. Dans cette section, nous décrivons les principales fonctionnalités de notre application à travers la présentation de quelques interfaces clés.

4.2.1 Interface 'Log In'

La figure suivante présente le premier lancement de l'application, la première fenêtre qui s'affiche est la fenêtre « Log In » suivante :

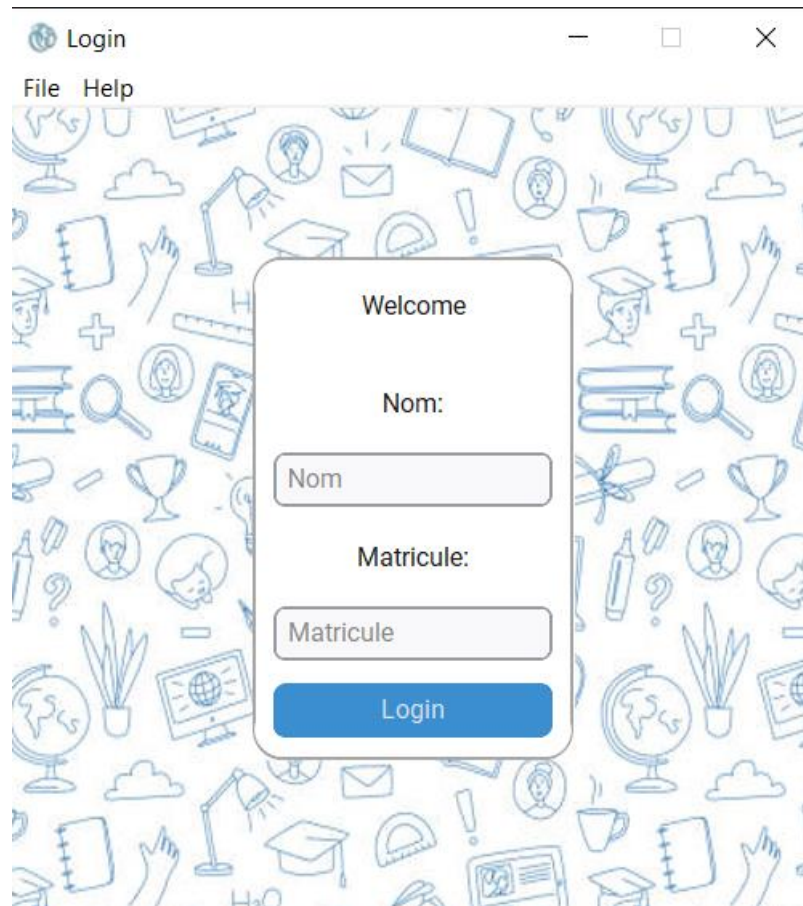


FIGURE 4-1- Interface 'Log In' de l'application

Cette interface montre une interface de connexion (login) de l'application, Voici une description générale du contenu :

4.2.1.1 Fenêtre de Connexion :

- La fenêtre a un titre "Log in".
- Il y a un menu avec les options "File" et "Help" en haut.

4.2.1.2 Formulaire de Connexion :

- Le formulaire de connexion est centré dans la fenêtre.
- Il y a un message de bienvenue "Welcome".
- Deux champs de saisie :
 - Un champ intitulé "Nom" pour entrer le nom de l'utilisateur.
 - Un champ intitulé "Matricule" pour entrer le numéro de matricule de l'utilisateur.
- Un bouton "**Login**" en bleu pour soumettre les informations de connexion.

4.2.1.3 Arrière-plan :

- L'arrière-plan est orné de diverses illustrations liées à l'éducation et la technologie, telles que des globes terrestres, des livres, des diplômes, des trophées, des plantes, etc.

4.2.1.4 Fenêtre d'Erreur :

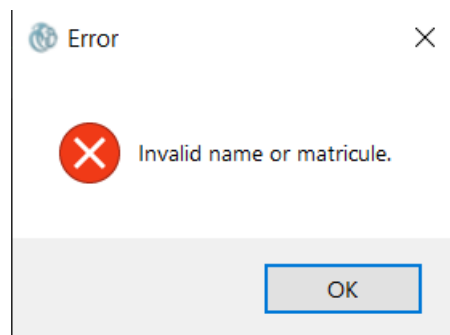


FIGURE 4-2- Fenêtre d'Erreur

Cette fenêtre de message d'erreur liée à la tentative de connexion dans l'application, Voici une description détaillée :

- **Fenêtre d'Erreur :**
 - La fenêtre a un titre "Error".
 - Un symbole d'erreur (croix rouge) est affiché à gauche du message.
- **Message d'Erreur :**
 - Le message d'erreur indique : "Invalid name or matricule." (Nom ou matricule invalide.)
- **Bouton de Confirmation :**
 - Un bouton "OK" permet de fermer la fenêtre d'erreur.

Cette fenêtre apparaît probablement lorsque les informations de connexion saisies (le nom ou le numéro de matricule) ne sont pas reconnues par le système. L'utilisateur est invité à vérifier ses informations et à réessayer.

4.2.1.5 Fenêtre de Succès :



FIGURE 4-3- Fenêtre de succès

Cette fenêtre de confirmation de succès après une tentative de connexion réussie dans l'application, Voici une description détaillée :

- **Fenêtre de Succès :**
 - La fenêtre a un titre "Success".
 - Un symbole d'information (cercle bleu avec un "i" blanc) est affiché à gauche du message.
- **Message de Confirmation :**
 - Le message indique : "Login successful!" (Connexion réussie !)
- **Bouton de Confirmation :**
 - Un bouton "OK" permet de fermer la fenêtre de confirmation.

Cette fenêtre apparaît lorsque les informations de connexion saisies (le nom et le numéro de matricule) sont reconnues et acceptées par le système, indiquant à l'utilisateur que la connexion

a été effectuée avec succès. De plus, les noms des étudiants et leurs numéros de matricule sont stockés dans une base de données, sous la forme d'un fichier Excel.

4.2.2 Interface des TPs :

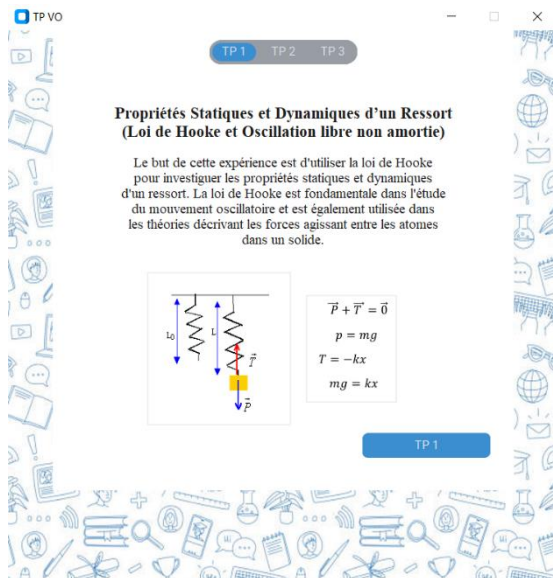


FIGURE 4-4- Fenêtre de TP1

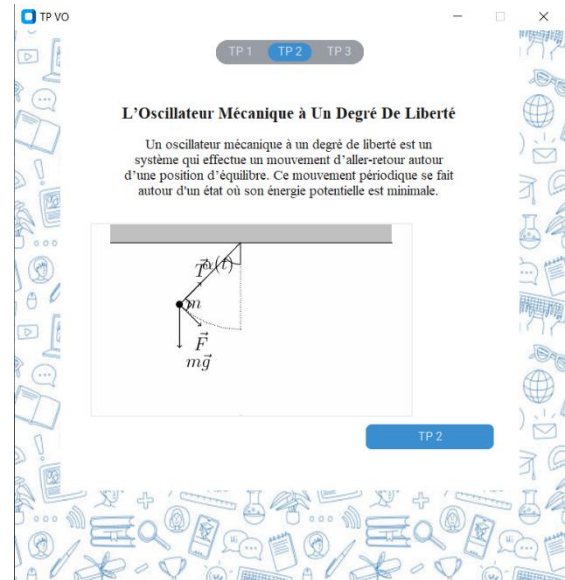


FIGURE 4-5- Fenêtre de TP2

En général, les figures montrent l'interface utilisateur de l'application après une connexion réussie. Cette interface propose plusieurs travaux pratiques (TP), avec des descriptions et des illustrations pour chaque TP. Voici un résumé des principaux éléments visibles :

- **Interface de Navigation :**

- Un menu en haut permet de naviguer entre différents travaux pratiques : TP1, TP2, TP3.
- Le menu montre clairement quel TP est actuellement sélectionné.

- **Travaux Pratiques :**

- Chaque TP a un titre, une description et une illustration explicative.
- Un bouton spécifique à chaque TP permet de valider ou d'accéder à plus d'informations.

4.2.2.1 TP1 : Propriétés Statique et Dynamiques d'un Ressort

- **Titre** : "Propriétés Statique et Dynamiques d'un Ressort (Loi de Hooke et Oscillation libre non amortie)"
- **Description** : Le but est d'utiliser la loi de Hooke pour étudier les propriétés des ressorts.
- **Illustration** : Montre un ressort avec des forces et des formules associées.

4.2.2.2 TP2 : L'Oscillateur Mécanique à Un Degré De Liberté

- **Titre** : "L'Oscillateur Mécanique à Un Degré De Liberté"
- **Description** : Étudie un système oscillant autour d'une position d'équilibre avec un degré de liberté.
- **Illustration** : Montre un oscillateur avec des vecteurs de forces.

Ces fenêtres permettent aux utilisateurs de choisir et d'explorer différents travaux pratiques après s'être connectés avec succès à l'application.

4.2.3 Interface de TP1 :

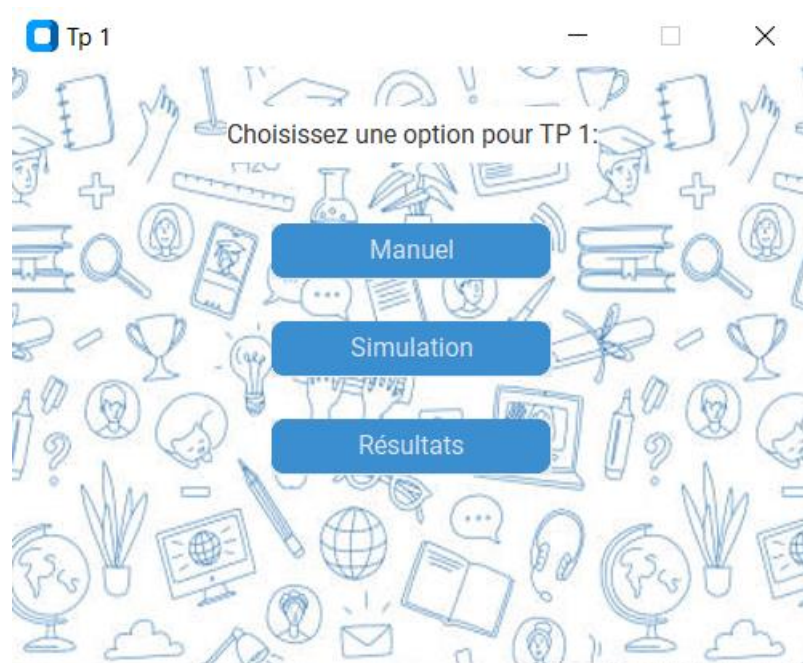


FIGURE 4-6- Fenêtre d'option TP1

Lorsque l'utilisateur clique sur le bouton "TP 1" dans l'interface des TPs, une nouvelle fenêtre s'affiche, permettant de choisir parmi différentes options pour le TP 1. Voici une description détaillée de cette fenêtre :

- **Titre et Indication** :

- La fenêtre a un titre "Tp 1".
 - Un message demande à l'utilisateur de "Choisissez une option pour TP 1".
- **Options Disponibles :**
 - **Manuel :** Un bouton pour accéder au manuel du TP, qui pourrait contenir des instructions détaillées, des explications théoriques, et les étapes de l'expérience.
 - **Simulation :** Un bouton pour accéder à une simulation interactive du TP, permettant à l'utilisateur de visualiser et d'interagir avec les concepts en temps réel.
 - **Résultats :** Un bouton pour consulter les résultats obtenus lors de l'expérience ou de la simulation, incluant potentiellement des graphiques, des analyses de données, et des conclusions.

Cette interface permet à l'utilisateur de choisir la manière dont il souhaite aborder le TP 1, offrant des options pour étudier le manuel, participer à une simulation, ou consulter les résultats. Cela permet une approche flexible et interactive de l'apprentissage des concepts scientifiques abordés dans le TP.

NB : L'interface de sélection pour le TP 2 offre les mêmes fonctionnalités et la même organisation que celle du TP 1, permettant aux utilisateurs de choisir facilement comment aborder chaque TP, que ce soit à travers le manuel, la simulation ou les résultats. Cette uniformité dans la présentation et les options disponibles assure une expérience utilisateur fluide et intuitive.

Lorsque l'utilisateur clique sur "Manuel" pour le TP1, le manuel du TP1 s'ouvre au format PDF. De même, lorsque l'utilisateur clique sur "Manuel" pour le TP2, le manuel du TP2 s'ouvre au format PDF.

Lorsque l'utilisateur clique sur le bouton "Simulation", l'application de simulation spécifique à cette expérience s'ouvre, permettant de simuler les conditions et les résultats de l'expérience de manière interactive et détaillée.

4.2.4 Fenêtre des Résultats TP1 :

Lors de la sélection du bouton "Résultats", l'utilisateur accède à une fenêtre dédiée à la saisie, au calcul et à l'analyse des données expérimentales pour les parties statique et dynamique de l'expérience. Cette interface interactive est conçue pour faciliter le processus de calcul et pour aider à tirer des conclusions basées sur les données obtenues à partir de la simulation.

4.2.4.1 Partie Statique

- **Compléter le Tableau :**
 - **Objectif :** Déterminer la constante de raideur k du ressort en utilisant les mesures d'allongement sous différentes masses.
 - **Formule Utilisée :** $P = mg$ où P est le poids, m la masse, et g l'accélération due à la gravité.
 - **Étapes :**
 1. Saisir la **Masse (m)** en kilogrammes (kg). (Ces valeurs proviennent de l'application de simulation.)
 2. Saisir la **Différence d'allongement (x)** en mètres (m). (Ces valeurs proviennent de l'application de simulation.)
 3. Cliquer sur **Calculer P** pour obtenir le Poids (P) en Newtons (N).
 4. La valeur de P/x est calculée et affichée automatiquement.
 5. Ajouter les valeurs au tableau en cliquant sur **Add to Table**.
 6. Réinitialiser les champs si nécessaire en cliquant sur **Clear Table**.
 - **Tableau Récapitulatif :** Affiche les colonnes pour les masses, les différences d'allongement, les poids, et les rapports P/x .
 - **Conclusion :** Espace pour noter les observations et conclusions basées sur les données saisies.

a) Compléter le tableau ci-dessus, conclure:

$$p = mg$$

Veuillez saisir les valeurs de masse (m) et de différence d'allongement (x) obtenues à partir de la simulation.

m(kg): Calculer le Poids P P : Reset

x(m): Calculer le P/x P/x :

Add to Table

| Masse m(kg) | Différence d'allongement x(m) | Poids P(N) | P/x(N/m) |
|-------------|-------------------------------|------------|----------|
| | | | |

Clear Table

Conclusion :

FIGURE 4-7- Question A Statique

- **Interprétation de P/x :**

- **Objectif :** Comprendre la signification dimensionnelle de P/x.
- **Dimension de P/x :** Le rapport P/x représente la constante de raideur k du ressort, exprimée en N/m.

b) Que représente P/x?

P/x :

Sa dimension : ▼

FIGURE 4-8- Question B Statique

- **Calcul des Masses Inconnues :**

- **Objectif** : Utiliser les valeurs d'allongement pour déterminer les masses inconnues.
- **Formule Utilisée** : $m = \frac{kx}{g}$
- **Étapes** :
 1. Sélectionner la masse (rouge, verte, marron).
 2. Saisir la valeur de l'allongement (x) en mètres. (Ces valeurs proviennent de l'application de simulation.)
 3. Cliquer sur **Calculer la Masse** pour obtenir la masse inconnue.
 4. La valeur de la masse est affichée automatiquement.

c) Calculer les valeurs des masses inconnues:

$$m = \frac{kx}{g}$$

Veuillez saisir la valeur de différence d'allongement (x) obtenue à partir de la simulation.

Veuillez saisir la valeur de différence d'allongement (x) de la masse rouge

x(rouge) (m):

La Masse m(rouge) :

FIGURE 4-9- Question C Statique

• **Calcul de la Constante de Raideur (k) :**

- **Objectif** : Déterminer la constante de raideur k du ressort.
- **Formule Utilisée** : $k = \frac{mg}{x}$
- **Étapes** :
 1. Saisir la **Masse (m)** en kilogrammes (kg). (Ces valeurs proviennent de l'application de simulation.)

2. Saisir la **Différence d'allongement (x)** en mètres (m). (Ces valeurs proviennent de l'application de simulation.)
 3. Cliquer sur **Calculer k** pour obtenir la constante de raideur.
 4. La valeur de k est affichée automatiquement.
- **Conclusion :** Espace pour noter les observations et conclusions basées sur les calculs.

d) Calculer la constante de raideur, conclure:

$$k = \frac{mg}{x}$$

Veillez saisir les valeurs de masse (m) et de différence d'allongement (x) obtenues à partir de la simulation.

m(kg) :

x(m) :

La constante de raideur :

Conclusion :

FIGURE 4-10- Question D Statique

4.2.4.2 Partie Dynamique

- **Compléter le Tableau :**

- **Objectif :** Analyser les périodes d'oscillation pour différentes valeurs de masse et d'allongement.

- **Formules Utilisées :**

1. $\omega = \sqrt{\frac{k}{m}}$

2. $T = \frac{2\pi}{\omega}$

- **Étapes :**

1. Saisir la **Différence d'allongement (x)** en mètres (m). (Ces valeurs proviennent de l'application de simulation.)
 2. Saisir le **Temps (t)** en secondes (s). (Ces valeurs proviennent de l'application de simulation.)
 3. Cliquer sur **Calculer T** pour obtenir la période TTT en secondes.
 4. Ajouter les valeurs au tableau en cliquant sur **Add to Table**.
 5. Réinitialiser les champs si nécessaire en cliquant sur **Clear Table**.
- **Tableau Récapitulatif** : Affiche les colonnes pour les différences d'allongement, les temps, et les périodes.
 - **Conclusion** : Espace pour noter les observations et conclusions basées sur les données saisies.

a) Compléter le tableau ci-dessus, conclure:

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{g}{x}}$$

$$T = \frac{2\pi}{\omega}$$

Veuillez saisir la valeur différence d'allongement (x) et le Temps (t) obtenues à partir de la simulation.

x(m) : **Calculer T** La période T :

t(s) :

Add to Table

| Différence d'allongement x(m) | Temps t(s) | Période T(s) |
|-------------------------------|------------|--------------|
| | | |

Clear Table

Conclusion :

FIGURE 4-11- Question A, B Dynamique

- **Calcul des Masses Inconnues :**

- **Objectif** : Utiliser les périodes d'oscillation pour déterminer les masses inconnues.
- **Formules Utilisées** :

$$1. \omega = \sqrt{\frac{k}{m}}$$

$$2. m = \frac{k}{\omega^2}$$

○ **Étapes :**

1. Sélectionner la masse (rouge, verte, marron).
2. Saisir la valeur de l'allongement (x) en mètres. (Ces valeurs proviennent de l'application de simulation.)
3. Cliquer sur **Calculer la Masse** pour obtenir la masse inconnue.
4. La valeur de la masse est affichée automatiquement.

c) Calculer les valeurs des masses inconnues:

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{g}{x}}$$

$$m = \frac{k}{\omega^2}$$

Masse rouge Masse verte Masse marron

Veillez saisir la valeur de différence d'allongement (x) de la masse rouge

x(rouge)(m) :

La Masse m(rouge) :

FIGURE 4-12- Question C Dynamique

• **Compléter le Tableau :**

- **Objectif :** L'objectif de cet exercice est d'analyser les périodes d'oscillation d'un système en fonction de différentes masses et des temps observés.
- **Formules Utilisées :** Les formules suivantes sont utilisées pour les calculs :

1. Pulsation (ω) : $\omega = \sqrt{\frac{k}{m}}$

Où k est la constante de raideur du ressort et m est la masse.

2. Période (T) : $T = \frac{2\pi}{\omega}$

○ **Étapes :**

1. **Saisir la Masse (m) en kilogrammes (kg) :** Entrez la valeur de la masse dans le champ "m(kg)". (Ces valeurs proviennent de l'application de simulation.)
2. **Saisir le Temps (t) en secondes (s) :** Entrez la valeur du temps dans le champ "t(s)". (Ces valeurs proviennent de l'application de simulation.)
3. **Calculer T et T² :** Cliquez sur le bouton "**Calculer T et T²**" pour obtenir la période T et la période au carré T²
4. **Ajouter les Valeurs au Tableau :** Cliquez sur "**Add to Table**" pour ajouter les valeurs de masse, temps, période, et T² au tableau récapitulatif.
5. **Réinitialiser les Champs si Nécessaire :** Utilisez le bouton "**Clear Table**" pour réinitialiser le tableau.

○ **Tableau Récapitulatif :** Le tableau en bas de l'interface affiche les colonnes suivantes :

- **Masse (m) en kilogrammes (kg)**
- **Temps (t) en secondes (s)**
- **Période (T) en secondes (s)**
- **T² en secondes carrées (s²)**

○ **Conclusion :** Un espace est prévu pour noter les observations et les conclusions basées sur les données saisies et les résultats obtenus.

d) Compléter le tableau ci-dessus

$$T = \frac{2\pi}{\omega}$$

$$\omega = \sqrt{\frac{k}{m}}$$

Veuillez saisir les valeurs de Masses (m) et le Temps (t) obtenues à partir de la simulation.

m(kg) : La période T :

t(s) : La valeur T² :

| Masse m(kg) | Temps t(s) | Période T(s) | T²(s²) |
|-------------|------------|--------------|--------|
| | | | |

FIGURE 4-13- Question 4 Dynamique

- **Tracer le Graphe $T^2 = f(m)$:**
 - **Objectif :** Visualiser la relation entre la période au carré (T^2) et la masse (m).
 - **Étapes :**
 1. Saisir les différentes **Masses (m)** en kilogrammes (kg). (Ces valeurs proviennent de tableau.)
 2. Saisir les valeurs correspondantes de T^2 en secondes carrées (s^2). (Ces valeurs proviennent de tableau.)
 3. Cliquer sur **Draw Graph** pour tracer le graphe.
 4. Réinitialiser les valeurs si nécessaire en cliquant sur **Reset**.

e) Tracer le graphe $T^2=f(m)$

Veuillez Saisir les valeurs de m et T^2 obtenues à partir du tableau.

| | | | | |
|-----|----------------------|---------|----------------------|---|
| m : | <input type="text"/> | T^2 : | <input type="text"/> | <input type="button" value="Draw Graph"/> <input type="button" value="Reset"/> |
| m : | <input type="text"/> | T^2 : | <input type="text"/> | |
| m : | <input type="text"/> | T^2 : | <input type="text"/> | |
| m : | <input type="text"/> | T^2 : | <input type="text"/> | |

FIGURE 4-14- Question E Dynamique

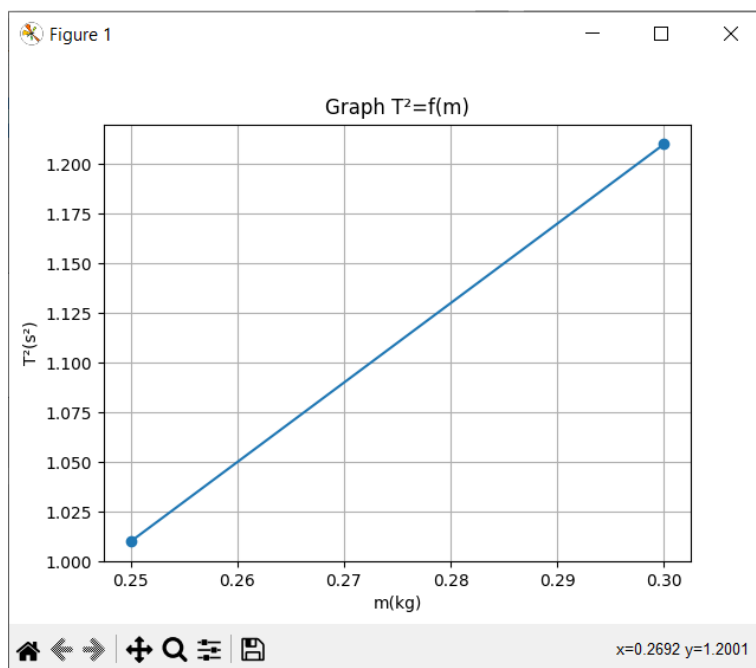


FIGURE 4-15- Graphe $T^2=f(m)$

• **Calculer la Pente :**

- **Objectif :** Déterminer la pente de la courbe $T^2 = f(m)$.

- **Formule Utilisée :** $\alpha = \frac{T_1^2 - T_2^2}{m_1 - m_2}$

- **Étapes :**

1. Saisir la valeur de T_1^2 en secondes carrées (s²).
2. Saisir la valeur de m_1 en kilogrammes (kg).
3. Saisir la valeur de T_2^2 en secondes carrées (s²).

4. Saisir la valeur de m_2 en kilogrammes (kg).
5. Cliquer sur **Calculer la pente** pour obtenir la pente.
6. La valeur de la pente est affichée automatiquement.

f) Calculer la pente.

$$\alpha = \frac{T_1^2 - T_2^2}{m_1 - m_2}$$

Veuillez saisir les valeurs de T^2 et m obtenues à partir de le graphe.

Entrez la valeur de T_1^2 (s²):

Entrez la valeur de m_1 (kg):

Entrez la valeur de T_2^2 (s²):

Entrez la valeur de m_2 (kg):

La pente :

FIGURE 4-16- Question F Dynamique

- **Enregistrer en PDF :**

- **Objectif :** Documenter et sauvegarder les résultats et les graphiques pour analyse ultérieure.
- **Étapes :**

Cliquer sur **Save to PDF** pour sauvegarder toutes les données et les graphiques en format PDF.

4.2.5 Fenêtre des Résultats TP2 :

Cette interface permet aux utilisateurs de mener des expériences et d'analyser les résultats afin de mieux comprendre les oscillations libres non amorties. Les utilisateurs peuvent saisir les

données, calculer les périodes, et observer les tendances pour tirer des conclusions basées sur leurs observations.

4.2.5.1 Oscillations libres non amorties :

- **Premier Cas :** Pour $m = 0.5 \text{ kg}$ et $L = 2 \text{ m}$
 - **Objectif :** Étudier l'effet de l'angle initial (θ) sur la période des oscillations (T).
 - **Paramètres Fixes :**
 - **Masse (m) :** 0.5 kg
 - **Longueur (L) :** 2 m
 - **Étapes :**
 1. **Theta (θ°) :** Saisir la valeur de l'angle initial en degrés.
 2. **Calculer la Période :** Cliquer sur **Calculer la Période** pour obtenir la période (T) des oscillations en secondes.
 3. **Ajouter au Tableau :** Cliquer sur **Add to Table** pour ajouter les valeurs saisies et calculées au tableau.
 4. **Réinitialiser les Champs :** Cliquer sur **Clear Table** pour vider le tableau et réinitialiser les champs.
 - **Tableau Récapitulatif :** Affiche les colonnes pour les valeurs de θ et les périodes T.
 - **Conclusion :** Espace pour noter les observations et conclusions basées sur les données saisies.

a)Premièr cas : Pour $m=0.5\text{kg}$ et $L=2\text{m}$:

Pour différentes valeurs de θ° , relever la période des oscillations $T(\text{s})$, conclure:

θ° : La période :

| Theta θ° | Période $T(\text{s})$ |
|----------------------|-----------------------|
| | |

Conclusion :

FIGURE 4-17- Question 1 Non Amortie

- **Deuxième Cas :** Pour $\theta = 10^\circ$ et $L = 2 \text{ m}$
 - **Objectif :** Étudier l'effet de la masse (m) sur la période des oscillations (T).
 - **Paramètres Fixes :**
 - **Angle Initial (θ) :** 10°
 - **Longueur (L) :** 2 m
 - **Étapes :**
 1. **Masse (m) :** Saisir la valeur de la masse en kilogrammes.
 2. **Calculer la Période :** Cliquer sur **Calculer la Période** pour obtenir la période (T) des oscillations en secondes.
 3. **Ajouter au Tableau :** Cliquer sur **Add to Table** pour ajouter les valeurs saisies et calculées au tableau.
 4. **Réinitialiser les Champs :** Cliquer sur **Clear Table** pour vider le tableau et réinitialiser les champs.

- **Tableau Récapitulatif** : Affiche les colonnes pour les valeurs de masse et les périodes T.
- **Conclusion** : Espace pour noter les observations et conclusions basées sur les données saisies.

b)Deuxième cas : Pour $\theta=10^\circ$ et $L=2m$:

Pour différentes valeurs de M , relever la période des oscillations T(s), conclure:

m(kg): La période :

| Masse m(kg) | Période T(s) |
|-------------|--------------|
| | |

Conclusion :

FIGURE 4-18- Question 2 Non Amortie

- **Troisième cas** : Pour $\theta=10^\circ$ et $m=1kg$
 - **Objectif** : Étudier la relation entre la période des oscillations T(s), $T^2(s)$ et la longueur L(m) de l'oscillateur.
 - **Paramètres fixes** :
 - **Angle initial (θ)** : 10°
 - **Masse (m)** : 1kg
 - **Étapes** :
 1. **Longueur L(m)** : Saisir la valeur de la longueur en mètres.
 2. **Calculer la Période** : Cliquer pour calculer la période T(s) et $T^2(s)$ correspondantes.
 3. **Ajouter au Tableau** : Cliquer pour ajouter les valeurs L, T et T^2 au tableau.
 4. **Réinitialiser le Tableau** : Cliquer sur "Clear Table" pour vider le tableau.
 - **Tracer le graphe T^2 -f(L)** :

- Saisir les valeurs de L et T² obtenues à partir du tableau dans les champs de saisie.
- Cliquer sur "Draw Graph" pour tracer le graphe de T² en fonction de L.
- Cliquer sur "Reset" pour réinitialiser le graphe.

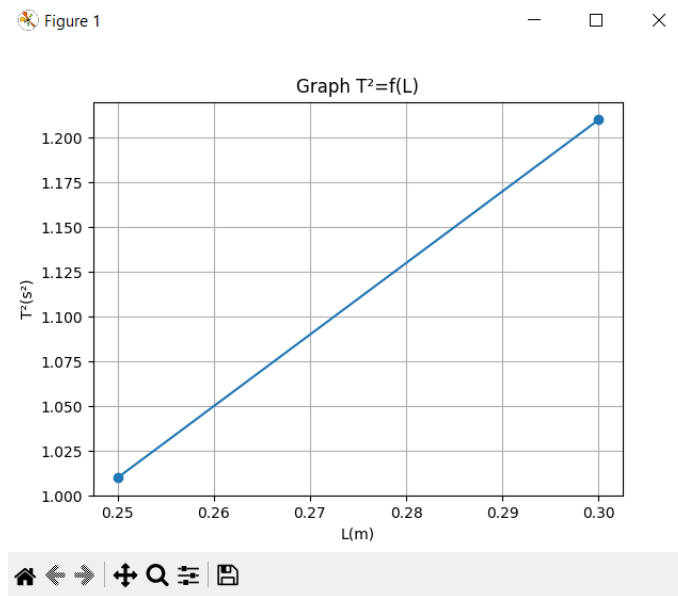


FIGURE 4-19- Graphe T²=f(L)

○ **Calculer la pente :**

- Saisir deux ensembles de coordonnées (T², L) à partir du graphe.
- Cliquer sur "Calculer la pente" pour calculer la pente de la droite passant par ces deux points.

c) Troisième cas : Pour $\theta=10^\circ$ et $m=1\text{kg}$:

Pour différentes valeurs de L , relever la période des oscillations T(s) et T²(s²)

L(m): La période:

| Longeur L(m) | Période T(s) | T ² (s ²) |
|--------------|--------------|----------------------------------|
| | | |
| | | |
| | | |
| | | |

Tracer le graphe T²=f(L):

Saisir les valeurs de L et T* obtenues à partir de tableau:

L: T*:

L: T*:

L: T*:

L: T*:

Calculer la pente

Veillez saisir les valeurs de T* et L obtenues à partir de graphe.

Entrez la valeur de T²₁(s²):

Entrez la valeur de L₁(m):

Entrez la valeur de T²₂(s²):

Entrez la valeur de L₂(m):

La pente:

FIGURE 4-20- Question 3 Non Amortie

4.2.5.2 Oscillations libres amorties :

- **Pour $\theta=15^\circ$, mesurer la période :**
 - **Entrée des valeurs :**
 - T : La période des oscillations obtenue à partir de la simulation

a) Pour $\theta=15^\circ$, mesurer la période

Veillez Saisir la valeur de T obtenue à partir de simulation

T=

Figure 4-21- Question 1 Amortie

- **Mesurer le nombre de période :**
 - **Entrée des valeurs :**
 - Nombre de périodes effectuées par le pendule jusqu'à l'arrêt

b)

Veillez Saisir le nombre de périodes effectuées par le pendule jusqu'à l'arrêt

le nombre de périodes:

FIGURE 4-22- Question 2 Amortie

- **En mode ralenti, relever les amplitudes maximales :**
 - **Entrée des valeurs :**
 - θ : L'angle θ obtenu à partir de la simulation
 - **Tableau de résultats :**
 - Colonne 1 : Période T (s)
 - Colonne 2 : θ (°)

c) En mode ralenti, relever les amplitudes maximales et remplir le tableau suivant :

Veuillez Saisir la valeur de θ obtenue à partir de simulation

Theta θ° :

| Période T(s) | Theta θ° |
|--------------|----------------------|
| 0 | |
| T/2 | |
| T | |
| 3T/2 | |

FIGURE 4-23- Question 3 Amortie

- Dessiner la courbe $\theta=f(T)$:
 - Graphique :
 - Entrer les valeurs de θ obtenues à partir de la simulation
 - Tracer la courbe $\theta=f(T)$

d) Dessiner la courbe : $\theta^\circ=f(T)$.

Veillez Saisir la valeur de θ obtenue à partir de simulation

θ° :

θ° :

θ° :

θ° :

θ° :

θ° :

θ° :

FIGURE 4-24- Question 4 Amortie

- **Calculer le Décrément Logarithmique :**

Le décrément logarithmique δ est une mesure de l'amortissement des oscillations. Il est défini

par la formule :
$$\delta = \frac{\theta(0)}{\theta(T)}$$

- **Entrée des valeurs :**

- $\theta(0)$: L'amplitude initiale des oscillations
- $\theta(T)$: L'amplitude des oscillations après une période T

- **Calcul :**

- Cliquez sur "Calculer le décrément" pour obtenir le décrément logarithmique δ

e) Calculer le décrément logarithmique.

$$\delta = \text{Ln}\left(\frac{\theta^\circ(0)}{\theta^\circ(T)}\right)$$

Veillez Saisir les valeurs de θ obtenue à partir de tableau

$\theta^\circ(0)$:

$\theta^\circ(T)$:

Calculer le décrément

le décrément logarithmique δ :

FIGURE 4-25- Question 5 Amortie

- **Calculer la Valeur du Facteur d'Amortissement :**

Le facteur d'amortissement (λ) est une autre mesure de l'amortissement des oscillations, reliant le décrément logarithmique à la période des oscillations.

- **Formule :** $\lambda = \frac{\delta}{T}$
- **Entrée des valeurs :**
 - δ : Le décrément logarithmique obtenu précédemment
 - T : La période des oscillations
- **Calcul :**
 - Cliquez sur "**Calculer le facteur**" pour obtenir le facteur d'amortissement λ .

f) Calculer la valeur du facteur d'amortissement.

Veillez Saisir la valeur de δ et T obtenue à partir des questions précédentes

δ :

T :

Calculer le facteur

le facteur d'amortissement λ :

FIGURE 4-26- Question 6 Amortie

- **Enregistrer en PDF :**

- **Objectif :** Documenter et sauvegarder les résultats et les graphiques pour analyse ultérieure.
- **Étapes :**

Cliquer sur **Save to PDF** pour sauvegarder toutes les données et les graphiques en format PDF.

4.3 Conclusion :

En conclusion, ce chapitre a détaillé la phase d'implémentation de notre projet, en mettant l'accent sur la présentation et le fonctionnement de l'application. Nous avons exploré les différentes interfaces utilisateur, en illustrant les interactions possibles entre l'utilisateur et le système. Chaque interface a été décrite en termes de fonctionnalités, d'accessibilité et d'ergonomie, soulignant ainsi comment elles contribuent à offrir une expérience utilisateur optimale.

Les choix technologiques, soigneusement sélectionnés, ont été mis en avant pour montrer leur impact sur l'efficacité et la fluidité de l'utilisation de l'application. En fournissant une vision claire et exhaustive des composants interactifs, nous avons démontré comment les utilisateurs peuvent naviguer et interagir de manière intuitive et efficace avec les diverses fonctionnalités proposées.

Ce guide de présentation des interfaces servira de référence essentielle pour comprendre l'architecture de notre application, facilitant ainsi une prise en main rapide et une utilisation satisfaisante. L'approche centrée sur l'utilisateur adoptée dans ce chapitre garantit que les besoins et attentes des utilisateurs sont pleinement pris en compte, assurant ainsi le succès et l'adoption de l'application.

Conclusion générale

Pour améliorer l'enseignement des travaux pratiques, il est crucial de repenser les méthodes traditionnelles en intégrant des outils technologiques modernes. Le développement d'une interface graphique interactive peut jouer un rôle significatif dans cette transformation. En fournissant aux étudiants une plateforme visuelle et pratique, ils peuvent mieux comprendre et appliquer les concepts théoriques appris en classe. Cette approche rend l'apprentissage plus engageant et intuitif, permettant aux étudiants de visualiser les concepts en temps réel et d'expérimenter directement avec les éléments théoriques.

L'utilisation d'une interface graphique interactive offre de nombreux avantages. Elle permet aux étudiants de manipuler des variables et de voir instantanément les résultats de leurs actions, ce qui renforce leur compréhension des relations de cause à effet. De plus, une telle interface peut être conçue pour inclure des simulations et des modèles dynamiques, offrant ainsi un environnement d'apprentissage immersif et stimulant. Les étudiants peuvent explorer différents scénarios et hypothèses, ce qui favorise le développement de compétences en résolution de problèmes et en pensée critique.

Enfin, l'utilisation d'une interface graphique interactive dans l'enseignement des travaux pratiques prépare les étudiants aux exigences du monde professionnel. De nombreux secteurs, tels que l'ingénierie, les sciences de la vie et l'informatique, utilisent des outils logiciels avancés pour modéliser et simuler des processus complexes. En se familiarisant tôt avec ces technologies, les étudiants acquièrent des compétences précieuses qui leur seront utiles tout au long de leur carrière.

Cette plate-forme, permet aux enseignants de mieux évaluer les connaissances et les acquis des étudiants, grâce à la génération, in situ, de comptes rendu, enregistré par l'éconsultable uniquement par l'enseignant.

En somme, l'intégration de cette plateforme interactive dans l'enseignement des travaux pratiques enrichit l'expérience d'apprentissage, facilite une meilleure assimilation des connaissances et prépare les étudiants aux défis professionnels futurs, ainsi qu'une meilleure évaluation des connaissances. Cette approche moderne et interactive représente une évolution essentielle de l'enseignement, alliant théorie et pratique de manière efficace.

Dans ce travail, nous avons réussi à réaliser une plateforme fonctionnelle, à destination des étudiants et enseignants des L2 ST, à accès réglementé, car elle ne donne accès qu'aux étudiants figurant dans une base de données définie, et sécurisée par le matricule de chaque étudiant, Elle offre aussi un meilleur contrôle à l'enseignant en charge du TP.

Cette plate-forme pour les TP Vibrations et Ondes, peut facilement être adaptée à d'autres TP, par l'intégration des énoncés et outils nécessaires et spécifiques à chaque module.

Bibliographie

- [1] H. Loechner, «letudiant,» 22 12 2021. [En ligne]. Available: <https://www.letudiant.fr/etudes/fac/qu-est-ce-que-le-systeme-lmd-licence-master-doctorat.html>. [Accès le 02 06 2024].
- [2] «diplomeo,» 26 08 2022. [En ligne]. Available: https://diplomeo.com/actualite-credits_ects_european_credits_transfer_system. [Accès le 02 06 2024].
- [3] D. m. kalanda, «unikin,» 12 06 2023. [En ligne]. Available: <https://www.unikin.ac.cd/fac/fpsysed/?p=150>. [Accès le 02 06 2024].
- [4] M.-F. Fave-Bonnet, «Du Processus de Bologne au LMD: analyse de la "traduction" française de "quality assurance",» Université Paris X Nanterre , Paris, 2008.
- [5] «Guide du LMD Guide du LMD,» 2011.
- [6] «geeksforgeeks,» 11 12 2020. [En ligne]. Available: <https://www.geeksforgeeks.org/python-gui-pyqt-vs-tkinter/>. [Accès le 03 06 2024].
- [7] amigos-maker, «dev,» 31 10 2019. [En ligne]. Available: <https://dev.to/amigosmaker/python-gui-pyqt-vs-tkinter-5hdd>. [Accès le 03 06 2024].
- [8] «educba,» 03 04 2023. [En ligne]. Available: <https://www.educba.com/what-is-kivy/>. [Accès le 03 06 2024].
- [9] amigos-maker, «dev,» 27 10 2019. [En ligne]. Available: <https://dev.to/amigosmaker/pyqt-vs-wxpython-51d9#:~:text=wxPython%20is%20easy%20and%20intuitive%2C%20while%20it%20can,the%20impressive%20amount%20of%20resources%20and%20user%20communities..> [Accès le 03 06 2024].
- [10] Prepbytes, «prepbytes,» 22 03 2023. [En ligne]. Available: <https://www.prepbytes.com/blog/java/introduction-to-java-swing/>. [Accès le 03 06 2024].
- [11] P. LARKIN, «careerkarma,» 20 07 2022. [En ligne]. Available: <https://careerkarma.com/blog/javafx-vs-java-swing/>. [Accès le 03 06 2024].

- [12] A. JOSHI, «rdglobalinc,» 04 06 2019. [En ligne]. Available: <https://www.rdglobalinc.com/wpf-vs-winforms-what-to-choose/#:~:text=Pros%20You%20can%20easily%20have%20your%20own%20look,it%20is%20possible%20to%20have%20advanced%20data%20binding>. [Accès le 03 06 2024].
- [13] «sofvare,» 22 04 2020. [En ligne]. Available: <https://sofvare.com/blog/advantages-of-windows-presentation-foundation-wpf/#:~:text=Advantages%20of%20WPF%20%E2%80%A2%20Limitless%20possibilities%20and%20flexibility,possibilities%20in%20controls.%20%E2%80%A2%20Microsoft%20supports%20it%20constantly..> [Accès le 04 06 2024].
- [14] E. Williams, «medium,» 21 11 2023. [En ligne]. Available: <https://medium.com/@emmaw4430/pros-and-cons-of-electron-js-development-should-you-use-it-407f67dae310>. [Accès le 04 06 2024].
- [15] EDITORIAL STAFF, «turtleverse,» 01 12 2022. [En ligne]. Available: <https://turtleverse.com/nw-js-vs-electron-what-are-the-differences-between-the-two/>. [Accès le 04 06 2024].
- [16] «geeksforgeeks,» 01 06 2024. [En ligne]. Available: <https://www.geeksforgeeks.org/python-gui-tkinter/>. [Accès le 05 06 2024].
- [17] «pythontutorial,» 2023. [En ligne]. Available: <https://www.pythontutorial.net/tkinter/>. [Accès le 05 06 2024].
- [18] «openfuture,» 2023. [En ligne]. Available: <https://openfuture.ai/blog/comparaison-des-meilleurs-frameworks-d-interface-utilisateur-python-graphique>. [Accès le 06 06 2024].