

Université Cadi Ayyad  
École Supérieure De Technologie-Safi  
Département : Informatique  
Filière : DUT ou GI

Rapport De projet De Fin D'étude  
DUT ou GI

---

Développement d'une application graphique-Python-Web de traduction  
de formules infixes en code Postscript

---

---

*Réalisé par :*

JADIR Mohammed

ROUDANI Zainab

---

---

*Encadré par :*

BAYAR Abdelouahad

---

ANNÉE UNIVERSITAIRE : 2022/2023

## Dédicaces

*Je dédie ce projet :*

A nos chers parents pour leurs sacrifices  
pour leur patience, leur amour et leur confiance en nous.

Ils ont tout fait pour notre bonheur et notre réussite.

nous vous rends hommage par ce modeste ouvrage en  
témoignage de notre gratitude éternelle et de notre infinie  
amour envers eux, témoignage de notre profonde affection et  
de notre attachement indéfectible,

A nos chers amis en gage de notre sincère appréciation Et  
gratitude pour leurs efforts, leur soutien continu leur aide  
précieuse et leur amour durant nos études.

Que Dieu nous gardons toujours unis,

A notre cher superviseur BAYAR Abdelouahad qui nous lui  
accompagnés durant ce projet. Qu'il reçoive nos sincères  
nous lui nous sincèrement reconnaissant pour ses efforts, son  
soutien et son aide dans la réalisation de ce projet.

# Remerciements

Nous tenons à exprimer nos sincères remerciements et notre profonde gratitude envers M. BAYAR Abdelouahad, de nous avoir donné l'opportunité de réaliser cette mission ce projet. Nous sommes reconnaissants envers lui pour sa disponibilité et ses conseils qui nous ont permis d'accomplir toutes nos tâches avec succès.

Enfin, nous voudrions exprimer nos remerciements pour cette opportunité de travailler ensemble sur ce projet passionnant.

Notre collaboration avec vous a été remarquable, et nous avons apprécié votre professionnalisme et votre enthousiasme tout au long du processus. Merci de nous avoir permis de contribuer à cette réussite..

# Table des matières

<b>Résumé</b>	<b>7</b>
<b>Abstract</b>	<b>8</b>
<b>Introduction Générale</b>	<b>9</b>
<b>Chapitre I    Présentation du projet</b>	<b>10</b>
1    Introduction . . . . .	10
2    Cahier de charge . . . . .	10
ii    La problématique . . . . .	10
ii    Notations d’expressions . . . . .	10
ii    Solution . . . . .	11
3    Les fonctionnalités d’application . . . . .	11
iii    Analyseur syntaxique pour la conversion d’infixe en postfixe : . . . . .	11
iii    Convertisseur postfixe en PostScript : . . . . .	11
iii    Ajouter une fonction d’infixe et sa valeur en PostScript dans un fichier : . . . .	12
iii    Modification de fichier contenant les fonctions d’infixe et leurs valeurs en Post-Script : . . . . .	12
4    Outils et environnements techniques : . . . . .	12
<b>Chapitre II    gestion du projet</b>	<b>17</b>
1    Introduction . . . . .	17
2    Méthodologie Agile . . . . .	17
ii    Nécessité de la méthodologie agile : . . . . .	17
ii    La méthode Kanban : . . . . .	18
3    Diagramme de Gantt . . . . .	20
4    Benchmarking . . . . .	20
5    Désigne d’application . . . . .	21
v    Zoning . . . . .	21
v    wireframing . . . . .	22
<b>Chapitre III    Réalisation du projet</b>	<b>24</b>
1    Introduction . . . . .	24
2    Analyse et conception : . . . . .	24
3    Développement et Mise en place : . . . . .	27
<b>Conclusion Générale</b>	<b>28</b>
. . . . .	28
. . . . .	28
<b>Bibographie</b>	<b>29</b>



# Table des figures

I.1	Logo HTML5 . . . . .	12
I.2	Logo css . . . . .	12
I.3	Logo JS . . . . .	13
I.4	Logo Python . . . . .	13
I.5	Logo BOOTSTRAP . . . . .	13
I.6	Logo vs-code . . . . .	14
I.7	Logo git . . . . .	14
I.8	Logo github . . . . .	14
I.9	Logo ganttproject . . . . .	14
I.10	Logo figma . . . . .	15
I.11	Logo Trello . . . . .	15
I.12	Logo EEL . . . . .	15
II.1	Liste de tâches de PFE sur la plateforme TRELLO . . . . .	19
II.2	diagramme de gantt . . . . .	20
II.3	site1 . . . . .	21
II.4	site2 . . . . .	21
II.5	Zoning . . . . .	22
II.6	landing page with figma . . . . .	23
II.7	page de transformation de postfixe en postscript . . . . .	23
III.1	Schéma explicatif . . . . .	24
III.2	Structure de projet . . . . .	25
III.3	Schéma de conception . . . . .	26

# Résumé

Le présent rapport détaille la réalisation d'un projet de développement d'une application web de traduction de formules infixes en postfixe et en code Postscript.

Le but principal du projet est de créer un programme qui puisse convertir une expression fonctionnelle, écrite en notation infixée, en une notation postfixée. La conversion s'effectuera en utilisant l'algorithme de notation polonaise inversée. Cette conversion est importante pour faciliter les calculs mathématiques et les opérations sur les expressions arithmétiques.

Le programme devra permettre à l'utilisateur d'entrer l'expression en notation infixée et d'obtenir en sortie la même expression en notation postfixée. Pour cela, le programme devra appliquer les différentes étapes de l'algorithme de notation polonaise inversée, notamment la gestion des opérateurs et des parenthèses.

Le projet nécessitera donc la conception d'un algorithme efficace pour la conversion et la mise en place d'une interface utilisateur conviviale. Le programme devra également être testé pour s'assurer de son bon fonctionnement dans différents scénarios et situations.

# Abstract

This report describes the implementation of a project to develop a web application for translating infix formulas into postfix code.

The main goal of the project is to create a program that can convert an arithmetic expression written in infix notation into postfix notation. The conversion will be done using the Reverse Polish Notation (RPN) algorithm. This conversion is important to facilitate mathematical calculations and operations on arithmetic expressions.

The program should allow the user to input the expression in infix notation and obtain the same expression in postfix and notation as output. To achieve this, the program will apply the various steps of the RPN algorithm, including handling operators and parentheses.

The project will require the design of an efficient algorithm for conversion and the implementation of a user-friendly interface. The program will also need to be tested to ensure its proper functioning in different scenarios and situations.



# Introduction Générale

L'écriture des expressions mathématiques en notation infixe est une méthode courante et largement utilisée dans les manuels scolaires, les livres, les documents scientifiques, Familiarité, Compatibilité et les sources de programmation. Cependant, cette notation peut être difficile à manipuler et à évaluer automatiquement pour les ordinateurs. Pour résoudre ce problème, les notations postfixes telles que postscript ont été développées.

Le but de ce projet de fin d'études est de créer une application web qui peut convertir des expressions écrites en notation infixe en notation postfixe en particulier Postscript. Cette application permettra aux utilisateurs de saisir des expressions mathématiques complexes, qui seront ensuite converties en notation postfixe ou postscript. Les avantages de cette conversion incluent l'évaluation plus rapide et facilité d'évaluation pour la machine.

Ce projet couvrira plusieurs aspects, notamment l'analyse syntaxique des expressions mathématiques, la conception d'algorithmes de conversion efficaces et l'implémentation de l'application en utilisant python en générale.

L'application résultante sera utile pour les enseignants de mathématiques, les étudiants, les ingénieurs et Les scientifiques impliqués dans des projets utilisant des applications manipulant des codes complexes ont souvent recours à l'exportation en notation postfixée, en particulier en PostScript.

# Chapitre I

## Présentation du projet

### 1 Introduction

Dans ce chapitre, nous exposons la solution que nous avons proposée et la conception du système à réaliser. L'objectif principal est d'apporter de la flexibilité à la gestion, en offrant une approche qui répond aux besoins spécifiques.

### 2 Cahier de charge

#### ii La problématique

La conversion d'expressions de la notation infixée à la notation postfixée ou postscript est courante en programmation. Cela permet une manipulation plus rapide de ces expressions dans les programmes informatiques.

Cependant, le temps d'analyse de l'expression mathématique nécessaire pour effectuer cette conversion peut varier en fonction de la complexité de l'expression, et il est important de prendre en compte ce temps d'analyse pour garantir l'efficacité et la performance de l'application de conversion.

Dans ce projet de fin d'études, l'objectif est d'analyser les différentes méthodes de conversion existantes et de proposer des améliorations pour réduire le temps d'analyse des expressions. De plus, il sera étudié comment implémenter efficacement un algorithme de conversion d'expression en notation infixée vers la notation postfixée et postScript, en prenant en compte les différentes règles de priorité et d'associativité des opérateurs.

#### ii Notations d'expressions

Dans la plupart des langages de programmation, les formules mathématiques sont écrites en notation infixée

- **Notation infixe** C'est la notation couramment utilisée en programmation pour les opérations arithmétiques, où l'opérateur est positionné entre les deux opérandes. Par exemple :

$(5 + 3) * 2$

- **Notation postfixe** : aussi appelée notation polonaise inverse, l'opérateur est placé après les deux opérandes. Par exemple :

$5\ 3 + 2 *$

- **Notation postscript** : c'est une variante de la notation postfixe utilisée dans le langage de programmation PostScript. Les opérandes sont empilés sur une pile, et les opérateurs sont appliqués aux éléments du sommet de la pile. Par exemple :

$5\ 3\ \text{add}\ 2\ \text{mul}$

## ii Solution

Dans le cadre de ce projet de fin d'études, nous nous concentrons sur le développement d'une application graphique Python-Web visant à résoudre la problématique de la conversion d'expressions de la notation infixée à la notation postfixée ou postscript. Cette conversion est couramment utilisée en programmation pour permettre une manipulation plus efficace des expressions mathématiques dans les programmes informatiques.

## 3 Les fonctionnalités d'application

### iii Analyseur syntaxique pour la conversion d'infixe en postfixe :

Cette fonctionnalité devrait prendre en entrée une expression en notation infixée et la convertir en notation postfixée en utilisant un algorithme de pile (Stack).

L'analyseur syntaxique doit être en mesure de gérer les opérateurs de calcul de base tels que +, -, \*, /, les parenthèses et les fonctions avec ses paramètres comme cos sin et d'autres fonctions avec personnalisation de chaque fonction.

### iii Convertisseur postfixe en PostScript :

Cette fonctionnalité devrait prendre en entrée une expression en notation postfixée et la convertir en notation PostScript en utilisant une syntaxe appropriée.

Il doit également gérer les opérateurs de calcul de base et les fonctions mathématiques telles que sin, cos et avec des fonctions personnalisées comme 'somme' en infixe peut être 'sum' en postscript.

### iii Ajouter une fonction d'infixe et sa valeur en PostScript dans un fichier :

Cette fonctionnalité doit permettre à l'utilisateur d'ajouter une nouvelle fonction d'infixe dans l'application avec sa valeur correspondante en PostScript.

Il est possible de stocker les fonctions et leurs correspondants dans un fichier texte ou une base de données, afin de conserver une liste des fonctions disponibles et de leurs paramètres associés pour une utilisation ultérieure dans l'analyseur syntaxique.

### iii Modification de fichier contenant les fonctions d'infixe et leurs valeurs en PostScript :

Cette fonctionnalité doit permettre à l'utilisateur de modifier le fichier contenant les fonctions d'infixe et leurs correspondants en PostScript.

Il est également possible d'ajouter une fonctionnalité de recherche pour faciliter la modification de fichiers volumineux, permettant ainsi de trouver rapidement les fonctions ou correspondances souhaitées dans le fichier.

## 4 Outils et environnements techniques :

### 1. HTML5 :



FIGURE I.1 – Logo HTML5

Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web. tem **CSS3** :

### 2. CSS3 :



FIGURE I.2 – Logo css

Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.

### 3. **Java Script :**



FIGURE I.3 – Logo JS

Il s'agit d'un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser. Bon, peut-être pas tout, mais vous pouvez faire bien des choses avec quelques lignes de JavaScript.

### 4. **Python :**



FIGURE I.4 – Logo Python

Python est un langage de programmation interprété, orienté objet et de haut niveau. Il est conçu pour être facile à lire, à écrire et à maintenir, et est souvent utilisé pour des applications telles que le développement web, la science des données, l'analyse de données, l'automatisation de tâches et la création de scripts.

### 5. **BOOTSTRAP :**



FIGURE I.5 – Logo BOOTSTRAP

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

## 6. Visual-studio-code :



FIGURE I.6 – Logo vs-code

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git.

## 7. GIT :



FIGURE I.7 – Logo git

Git est un système de contrôle de version distribué gratuit et open source conçu pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité.

## 8. GITHUB :



FIGURE I.8 – Logo github

GitHub est une plateforme de développement qui fournit un service cloud aux développeurs pour stocker et gérer leur code. De l'open source à l'entreprise, vous pouvez héberger et revoir du code, apporter des modifications, gérer des projets et construire des logiciels aux côtés de millions d'autres développeurs .

## 9. GANTTPROJECT :



FIGURE I.9 – Logo ganttproject

GanttProject est un outil puissant pour la gestion de projet, offrant des fonctionnalités essentielles pour la planification, l'organisation et le suivi des tâches. Il peut être utilisé par les professionnels de tous les secteurs pour gérer leurs projets de manière efficace et productive.

## 10. Figma



FIGURE I.10 – Logo figma

Figma est un outil de conception et de prototypage basé sur le cloud utilisé par les designers et les équipes pour créer des interfaces utilisateur, des prototypes interactifs et des éléments de design. Il fournit une plateforme collaborative où plusieurs designers peuvent travailler ensemble en temps réel, ce qui en fait un choix idéal pour les équipes distantes et réparties.

## 11. Trello



FIGURE I.11 – Logo Trello

Trello est un outil de gestion de projet basé sur le concept de tableau Kanban. Il permet aux équipes de collaborer et d'organiser leurs tâches de manière visuelle et intuitive. Avec Trello, vous pouvez créer des tableaux pour représenter vos projets, des listes pour regrouper les différentes étapes ou catégories de tâches, et des cartes pour représenter les tâches individuelles.

## 12. EEL

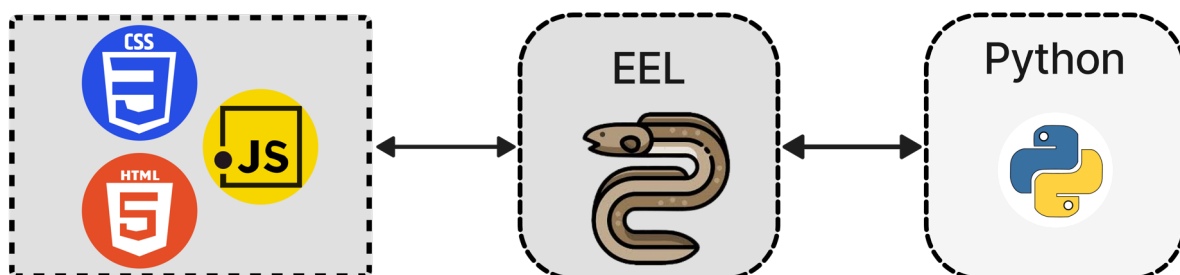


FIGURE I.12 – Logo EEL

Eel est une bibliothèque Python légère qui permet de créer des applications de bureau avec une interface graphique utilisateur (GUI) similaire à Electron. Elle offre un moyen simple de

développer des applications hors ligne en utilisant HTML, CSS et JavaScript, tout en ayant un accès complet aux fonctionnalités et aux bibliothèques de Python.

Avec Eel, vous pouvez concevoir des applications de bureau qui s'exécutent localement sur l'ordinateur de l'utilisateur, offrant ainsi une expérience utilisateur fluide et réactive. Eel facilite la création d'applications en combinant les langages de programmation web avec les capacités puissantes de Python.

En utilisant Eel, vous pouvez créer des interfaces utilisateur interactives en utilisant des technologies web familières telles que HTML pour la structure, CSS pour la présentation et JavaScript pour l'interactivité. Vous pouvez également tirer parti de toutes les bibliothèques et fonctionnalités de Python pour effectuer des tâches de traitement des données, des interactions système, des calculs complexes, etc.

Eel simplifie le processus de développement en offrant une intégration transparente entre Python et les technologies web, vous permettant ainsi de tirer parti de votre expérience en Python pour créer des applications de bureau performantes et personnalisées. Que vous souhaitiez créer des outils utilitaires, des interfaces graphiques pour vos scripts Python ou des applications autonomes, Eel vous offre une solution flexible et efficace..



# Chapitre II

## gestion du projet

### 1 Introduction

Dans cette partie, nous aborderons la gestion de projet, les tâches que nous avons effectuées, la méthode de travail que nous avons utilisée, ainsi que les tâches extraites du cahier des charges et les spécifications fonctionnelles telles que la réalisation de zoning et de wireframes. Toutes ces étapes ont été réalisées avant l'implémentation et le développement de l'application.

### 2 Méthodologie Agile

La méthodologie Agile est une approche de gestion de projet de développement logiciel qui met l'accent sur l'adaptabilité, la collaboration et l'itération.

Elle se concentre sur la satisfaction du client en fournissant des résultats de manière itérative et en répondant rapidement aux changements et aux retours d'information.

L'Agilité vise à fournir des logiciels fonctionnels rapidement et à maintenir une communication continue entre l'équipe de développement et le client tout au long du processus de développement.

#### **ii Nécessité de la méthodologie agile :**

Les projets informatiques étaient confrontés à un taux d'échec élevé en raison du décalage entre les attentes des clients et les solutions livrées. Face à cette problématique, 17 experts du développement logiciel se sont réunis aux États-Unis en 2001 pour unifier leurs méthodes respectives et définir une nouvelle approche du développement de logiciels. Cette initiative était motivée par la nécessité d'adapter les techniques informatiques en constante évolution à la réalité des projets et de répondre de manière plus efficace aux besoins des clients.

Les méthodes agiles ont été conçues pour résoudre les problèmes associés aux projets informatiques de grande envergure. Les projets traditionnels peuvent être longs, coûteux, difficiles à piloter et ne garantissent pas nécessairement un produit final qui répond aux attentes du client. Les méthodologies

agiles, quant à elles, fixent des objectifs à court terme et utilisent une gestion de projet itérative et adaptative. Au lieu de planifier l'intégralité du projet dès le départ, le projet est découpé en sous-parties, ou sprints, que les équipes de développement doivent atteindre progressivement en ajustant si nécessaire les objectifs pour répondre au mieux aux attentes du client.

Les méthodes agiles permettent également de renforcer les relations entre les membres de l'équipe ainsi qu'entre l'équipe et le client. La flexibilité et la souplesse sont des éléments clés de la méthodologie agile, qui permettent de s'adapter aux réalités du contexte et de l'environnement projet. En travaillant de manière itérative et en s'adaptant aux besoins changeants du client, les équipes peuvent créer des produits finaux de meilleure qualité tout en gagnant en efficacité et en productivité.

Les méthodes agiles sont des modes de gestion de projets informatiques qui se caractérisent par une forte collaboration entre toutes les parties prenantes, une grande capacité à s'adapter aux changements, et une livraison rapide des produits. Elles visent à remplacer les méthodes traditionnelles, plus rigides et axées sur des spécifications contractuelles. Les méthodes agiles permettent ainsi de mieux répondre aux attentes des clients et utilisateurs finaux en proposant des solutions plus flexibles et mieux adaptées aux réalités du contexte projet.

L'approche agile encourage donc à valoriser :

- **l'équipe**, c'est-à-dire des individus et des interactions plutôt que des processus et des outils .
- **l'application** , soit des fonctionnalités opérationnelles plutôt que de la documentation exhaustive .
- **la collaboration** avec le client plutôt que la négociation contractuelle.
- **l'adaptation au changement** plutôt que le suivi d'un plan.

Afin d'obtenir des résultats optimaux avec la méthode AGILE, il est primordial de privilégier les relations humaines et la communication en mettant en place certains rituels tels que les réunions quotidiennes, les rétrospectives et les réunions de planification des cycles de développement. Cela favorisera la création d'une équipe engagée et autonome.

## ii La méthode Kanban :

**Kanban** est une technique de gestion de projet qui repose sur les principes de l'approche Lean, visant à améliorer constamment les processus de production. Elle a été développée par l'ingénieur industriel japonais Taiichi Ōno dans les années 1950 pour aider Toyota à optimiser sa fabrication de voitures.

L'objectif principal de la méthode Kanban est de répondre en permanence aux besoins du client tout en minimisant les risques de surproduction et de gaspillage, ainsi que de réduire les délais et les coûts de production.

La méthode Kanban est devenue une approche populaire pour la gestion des tâches au sein des entreprises, car elle offre une organisation plus souple et une responsabilisation accrue des membres de l'équipe. Cette méthode prône la visualisation des flux de travail grâce à un tableau Kanban qui permet de prioriser les tâches à accomplir et de suivre leur état d'avancement en temps réel. Cela

permet d'optimiser la gestion des flux de production tout en limitant les risques de surproduction et de gaspillage, ainsi que de réduire les délais et les coûts.

Kanban signifie « étiquette » en japonais. Chaque étiquette correspond à une tâche à réaliser. Pour visualiser l'état d'avancement d'une commande ou d'un projet, elles seront réparties dans un tableau Kanban divisé en quatre parties : "À faire", "En cours", "En revue" et "Réalisé". Chacun sait ainsi ce qu'il doit faire, quand et comment.

*Les étapes :*

- **A faire :**  
méthode Kanban consiste à ajouter les tâches identifiées sur des cartes Kanban et les placer dans la colonne "à faire" de votre tableau Kanban.
- **En cours :** méthode Kanban consiste à déplacer les cartes Kanban de la colonne "à faire" à la colonne "en cours" pour représenter l'état actuel de chaque tâche en cours de réalisation.
- **En revue :** la méthode Kanban consiste à déplacer les cartes Kanban de la colonne "en cours" à la colonne "en revue" pour indiquer que la tâche est terminée mais nécessite une vérification ou une validation avant d'être considérée comme terminée.
- **Terminé :** la méthode Kanban consiste à déplacer les cartes Kanban de la colonne "en revue" à la colonne "réalisé" pour indiquer que la tâche a été terminée avec succès et qu'elle est considérée comme terminée. C'est la dernière étape de la méthode Kanban qui permet de clôturer une tâche.

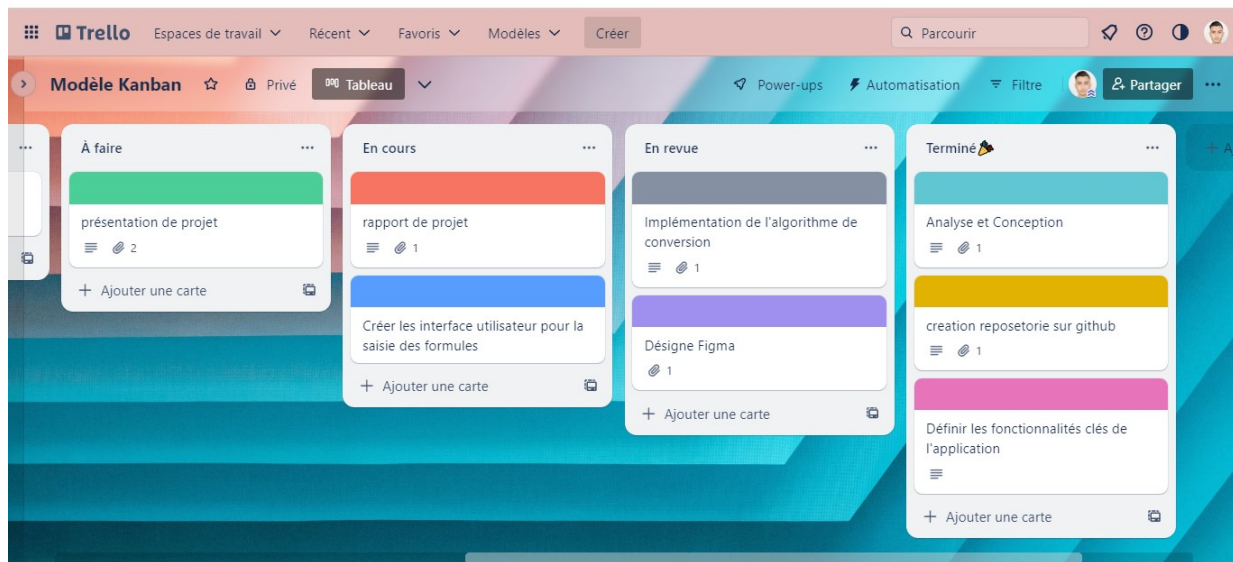


FIGURE II.1 – Liste de tâches de PFE sur la plateforme TRELLO

Au moment du lancement de projet, les tâches sont réparties, et donc les étiquettes, par membre de l'équipe ; toutes les étiquettes se trouvent dans la première colonne «To Do». Au fur et à mesure de l'avancement du projet, le tableau Kanban reflète visuellement la réalisation des tâches : l'étiquette passe de colonne en colonne jusqu'à la colonne «Done».

### 3 Diagramme de Gantt

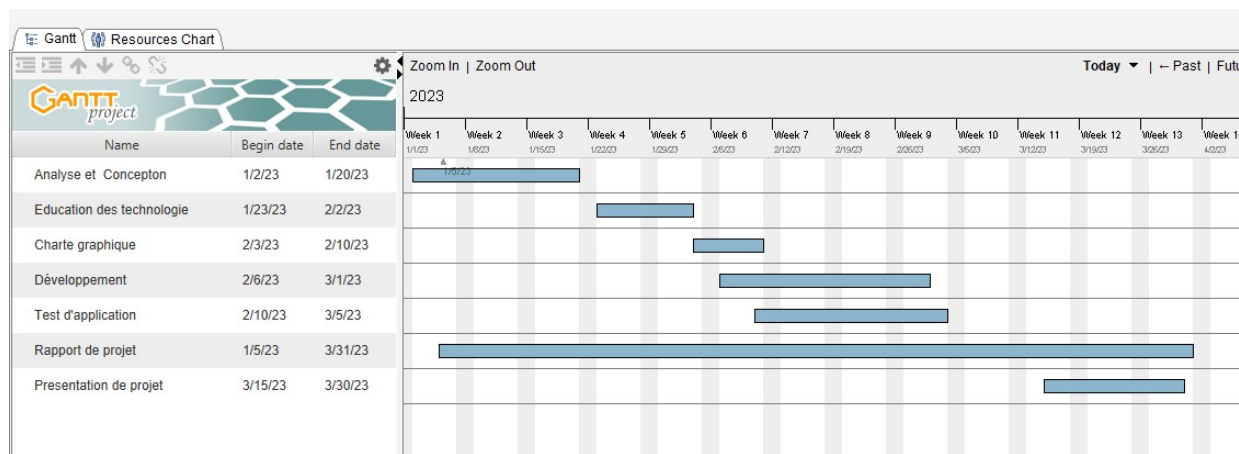


FIGURE II.2 – diagramme de gantt

Notre projet est constitué de plusieurs étapes, distribuées comme le montre le diagramme ci-dessus. Le graphique indique quand chaque tâche doit être effectuée. Cela nous permet d'exécuter chaque sous-tâche dans les délais et de terminer l'ensemble du projet à la date prévue. Certaines des tâches peuvent être effectuées à tout moment, tandis que d'autres doivent être exécutées avant ou après le début ou la fin d'une autre tâche.

### 4 Benchmarking

Le benchmarking est une pratique courante dans le domaine de la gestion et de l'amélioration des performances. Il consiste à comparer les performances d'une entreprise, d'un produit ou d'un processus avec celles des meilleurs acteurs du secteur, afin d'identifier les écarts et de déterminer les mesures correctives nécessaires.

Nous avons effectué le benchmarking sur plusieurs sites web et nous avons extrait les points forts et les points faibles pour chacun d'entre eux. Cependant, nous allons implémenter uniquement deux sites web.

la première site web : <https://www.calcont.in/Conversion/>

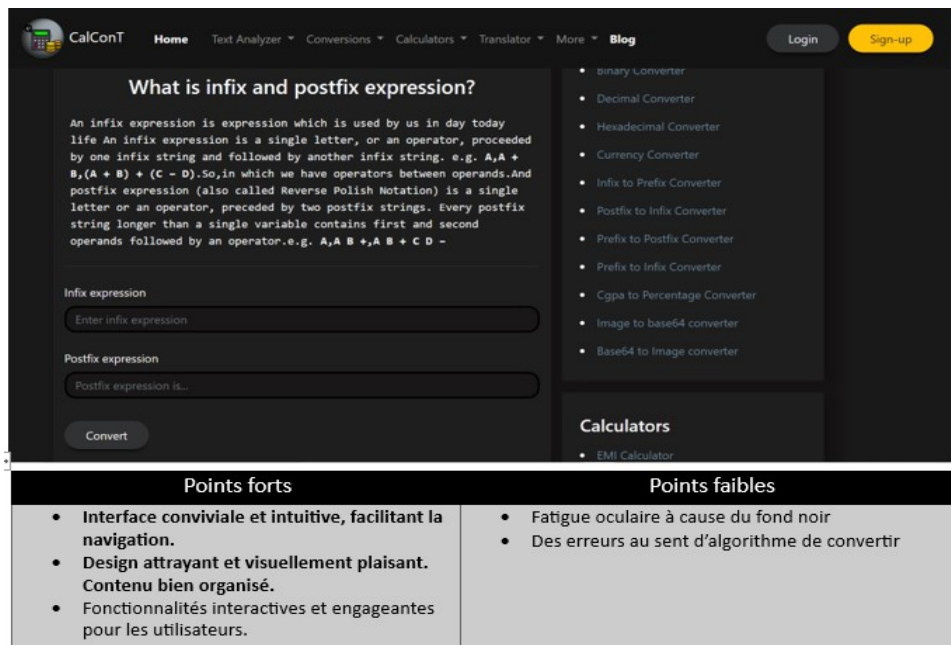


FIGURE II.3 – site1

la deuxième : <https://raj457036.github.io/Simple-Tools/prefixAndPostfixConvertor.html>

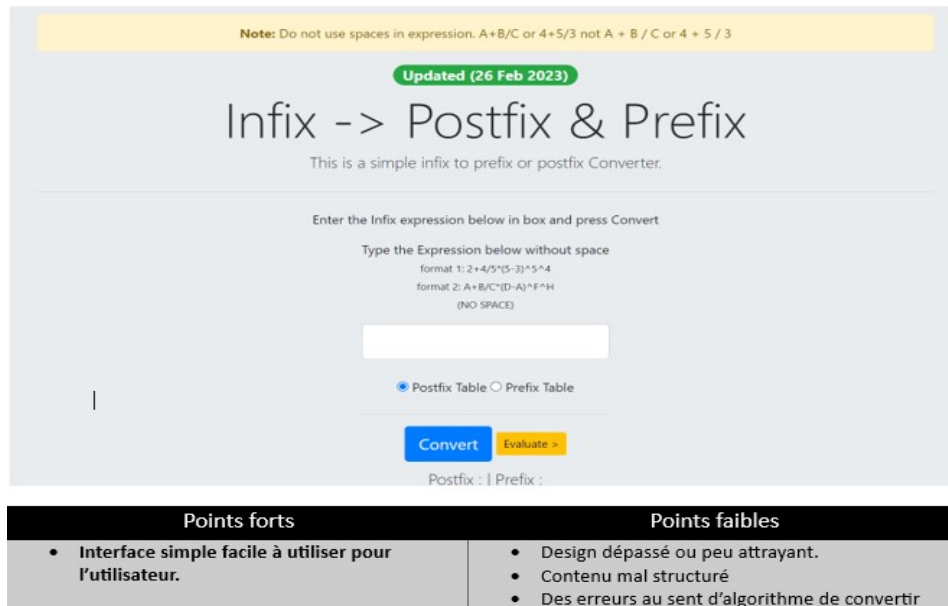


FIGURE II.4 – site2

## 5 Désigne d'application

### v Zoning

Le zoning est une méthode de maquettage à faible fidélité qui consiste à structurer le contenu d'une page web en la divisant en différentes zones distinctes.

À ce stade, l'objectif principal du zoning est de définir le découpage, l'organisation et la structure des pages, en mettant l'accent sur la répartition des différentes zones. Bien que ces zones puissent éventuellement contenir plusieurs éléments graphiques, lors de la phase de zoning, ces détails spécifiques sont délibérément omis afin de se concentrer sur la mise en place de la structure globale.

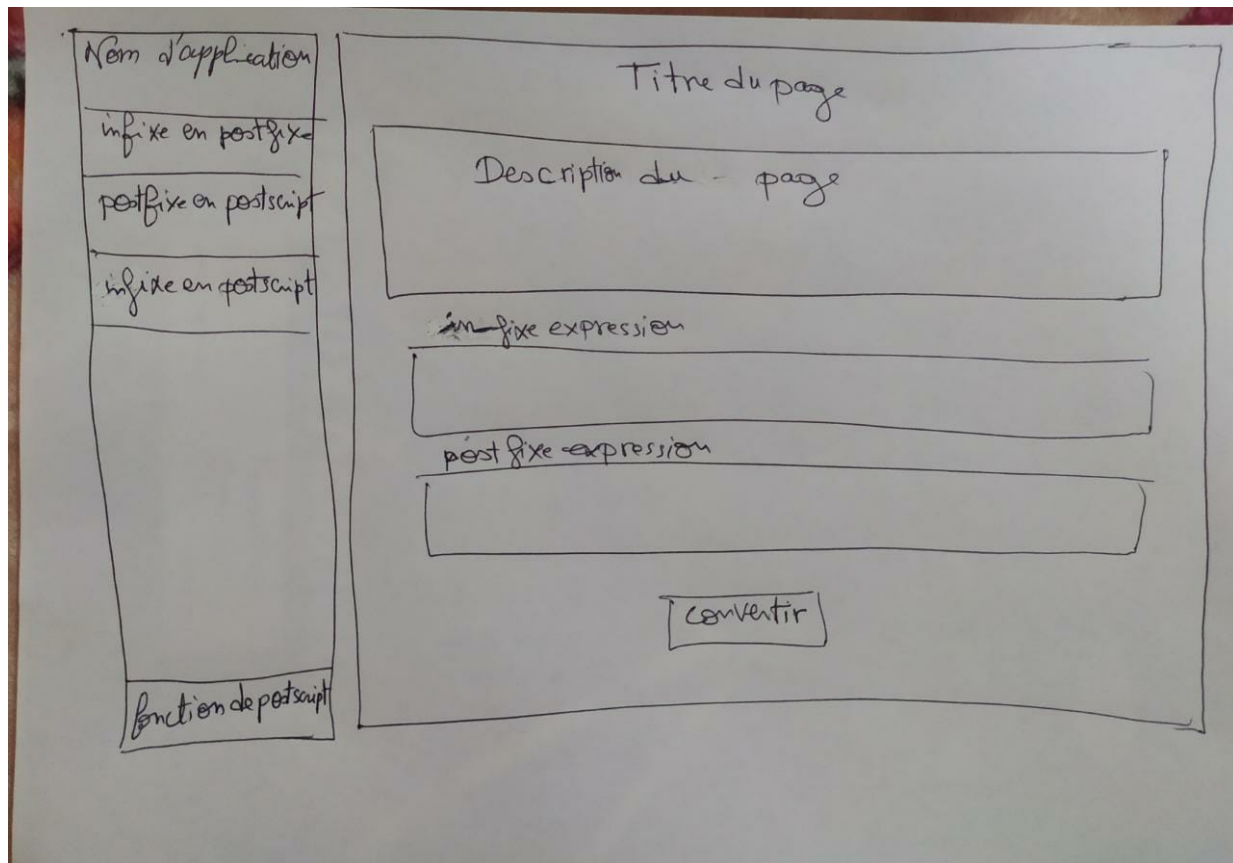


FIGURE II.5 – Zoning

## v wireframing

Le wireframing, également connu sous le nom de maquette fil de fer, est une méthode utilisée pour concevoir des versions à faible ou moyenne fidélité d'un produit final.

En d'autres termes, le wireframing ajoute plus de précision aux maquettes en permettant de remplacer les zones par les éléments spécifiques qui les composent. L'idée est de partir d'une structure de base bien définie, puis d'ajouter progressivement les détails nécessaires pour donner vie au produit.

Dans le cadre de la conception de notre projet web, nous avons utilisé l'outil de design d'interface utilisateur Figma. Figma nous a permis de créer des maquettes et des prototypes interactifs pour visualiser et communiquer efficacement nos idées de conception. Nous avons également utilisé Figma pour passer de la phase de zoning à la création de wireframes.

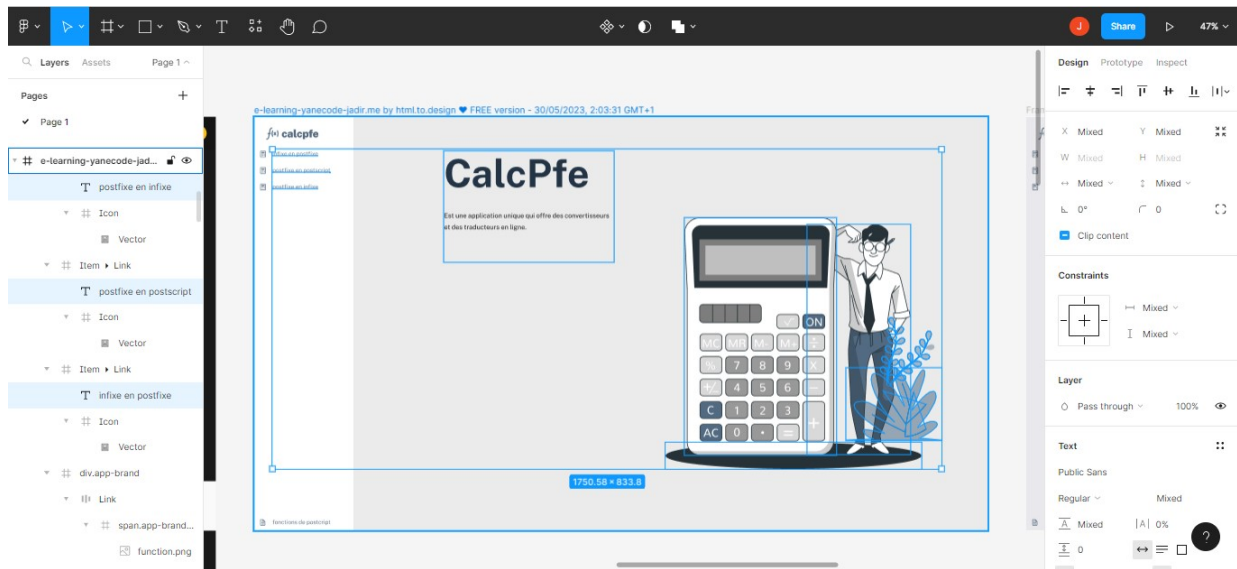


FIGURE II.6 – landing page with figma

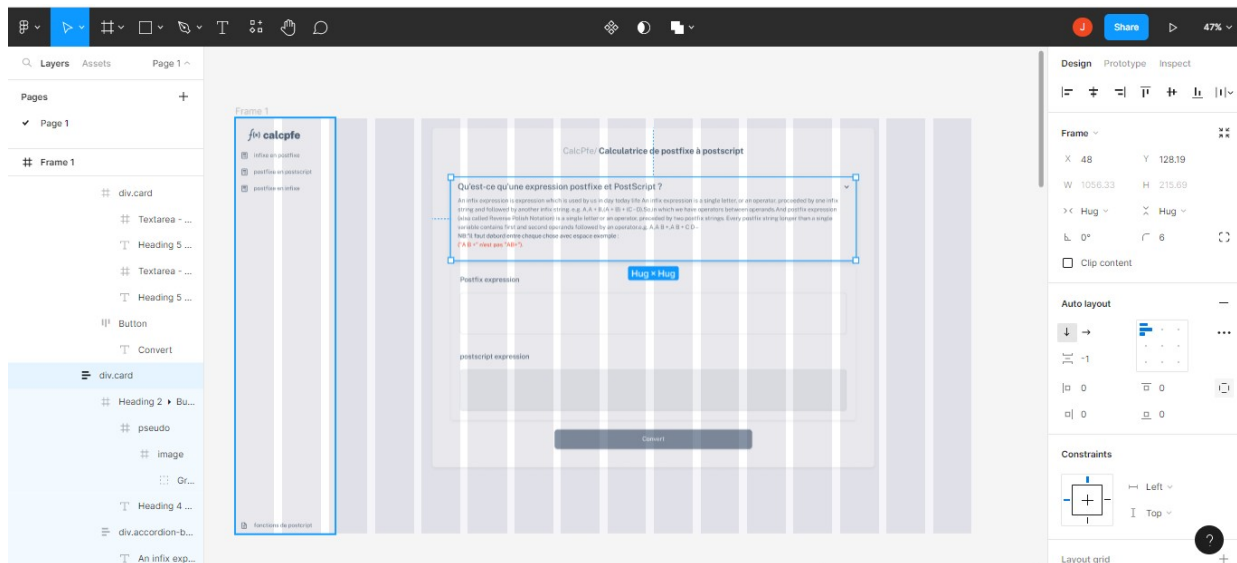


FIGURE II.7 – page de transformation de postfixe en postscript

# Chapitre III

## Réalisation du projet

### 1 Introduction

Dans ce chapitre, nous nous intéresserons aux fonctionnalités de l'application et à la réalisation du projet de fin d'études. Nous aborderons ma conception préalable à la mise en œuvre, ainsi que l'installation soutenue de figures commentées.

### 2 Analyse et conception :

Après avoir détaillé le cahier des charges et bien assimilé les objectifs de notre projet, une conception avant l'initialisation de la réalisation de notre projet est illustrée comme suit :



FIGURE III.1 – Schéma explicatif

Le code de notre projet de fin d'études (PFE) est organisé de manière structurée dans le dossier dédié. Le dossier principal contient plusieurs éléments essentiels. Tout d'abord, le dossier "web" rassemble les fichiers HTML, CSS et JavaScript qui constituent l'interface utilisateur de notre application. Ces fichiers définissent la mise en page, les styles visuels et le comportement interactif de notre application web.

En parallèle, nous avons également un fichier Python spécifique qui joue un rôle central dans notre



projet. Ce fichier contient l'ensemble des traitements et des algorithmes nécessaires à la fonctionnalité de notre application. Il agit comme un moteur de calcul et assure le bon fonctionnement de toutes les fonctionnalités que nous avons développées.

Pour faciliter la maintenance et la compréhension du code, nous avons également inclus un fichier texte qui répertorie toutes les fonctions de notation infixée utilisées dans notre application, ainsi que leurs correspondants en notation postscript. Ce fichier constitue une référence pratique pour nous guider dans la conversion des expressions et garantir la cohérence de notre application.

Grâce à cette organisation méthodique du code, nous avons pu gérer efficacement les différentes parties de notre projet, séparer les responsabilités entre les fichiers HTML, CSS, JavaScript et Python, et créer une application web fonctionnelle et cohérente.

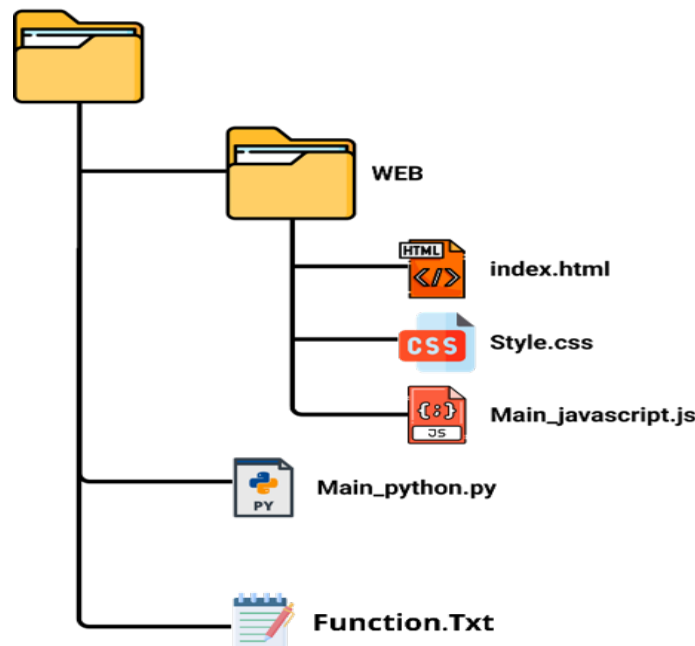


FIGURE III.2 – Structure de projet

Notre Input, il s'agit d'une expression. A l'aide d'un code python on va traiter la formule pour l'adaptation de besoin de client tels que :

- **Input :**

Un utilisateur crée une formule soit infixée, postfixée ...

- **Output :**

La forme du résultat dépend des besoins de l'utilisateur (infixe, postfixe, PostScript).

- **Fonction besoin à ajouter(infixe=>key) :**

La fonctionnalité à ajouter consiste à traduire des formules en notation infixée vers la notation PostScript. Cette fonctionnalité sera activée par une clé spécifique.

- Fonction à saisir par l'utilisateur comme =>'somme()'

- **Fonction besoin à ajouter (postscript=>value) :**

La fonction de conversion doit également prendre en compte l'affichage de la formule si l'utilisateur souhaite spécifier une fonction, telle que "sum()". Ainsi, si l'utilisateur entre "Somme()", la fonction doit être convertie en "sum()". La valeur de clé en PostScript est également un aspect important à considérer lors de la conversion de la formule en PostScript.

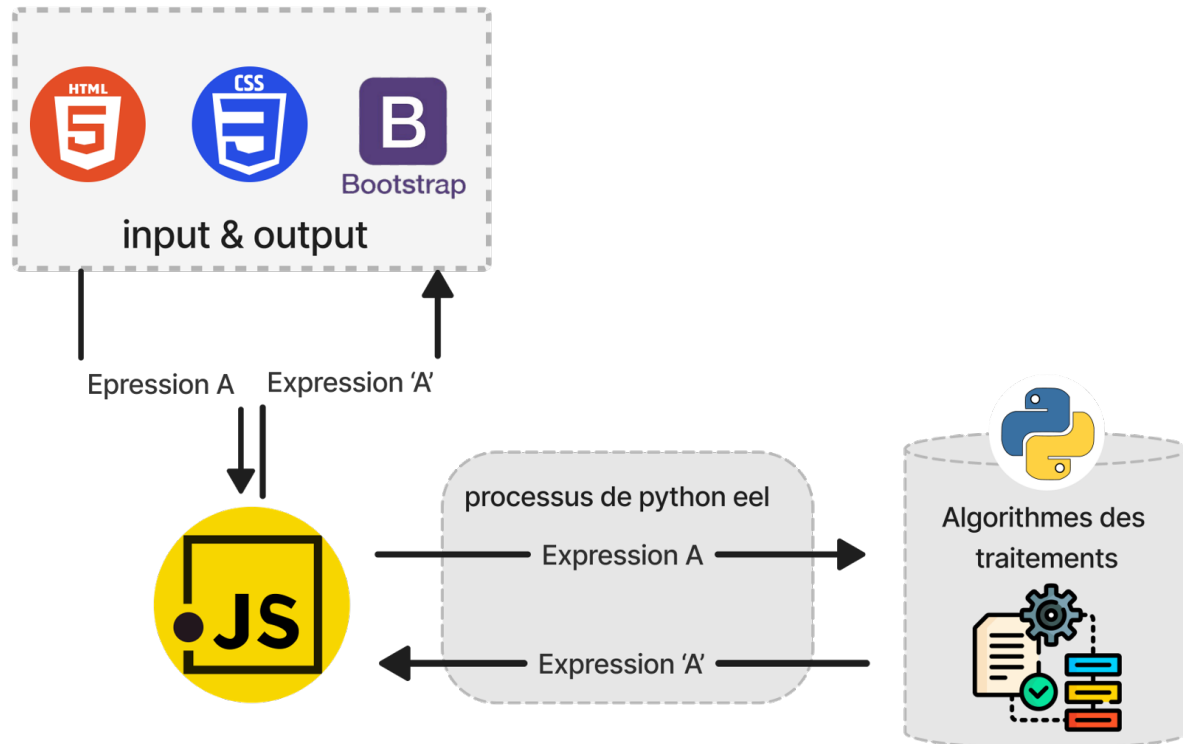


FIGURE III.3 – Schéma de conception

### 3 Développement et Mise en place :

Screeeeeenshots after talking with mr Bayar about the web app

# Conclusion Générale

kihvugjyhfgvujyhg

# Bibographie

- [https ://www.marsamaroc.co.ma/](https://www.marsamaroc.co.ma/) (consulté le ...)
- [https ://www.openclassrooms.com](https://www.openclassrooms.com)

# Annexe 1