Une image contenant texte, Police, Graphique, graphisme

Description générée automatiquement

PROJET DE FIN D’ANNEE

*5éme Année en Ingénierie Informatique et Réseaux*

Mise en Œuvre d'une Application de Synthèse Vocale pour les Articles de Presse du Groupe Le Matin

***Réalisé par :*** *HAMZA EL MADANI*

***Tuteur (s) :***

*Encadrants Professionnels : M. YASSINE FATAHLLAH*

*Et M. WALID AZHARI*

*Encadrant Pédagogique : M. MOHAMMED BOUSMAH*

***Membre de jury:***

*M.MOHAMED BOUSMAH*

*M.MOHAMED EL YOUSSEFI*

*M.ABDELMAJID BOUSSELHAM*

*Au sein de (Groupe Le Matin)*

**Année universitaire : 2023/2024**

Une image contenant texte, Police, Graphique, graphisme

Description générée automatiquement

# **Dédicaces**

Louange à « Allah » qui m’a guidé sur le droit chemin tout au long du travail et m’a inspiré les bons pas et les justes réflexes. Sans sa miséricorde, ce travail n’aura pas abouti.

Je dédie ce travail :

**À mes très chers parents,**

« Aucun mot ne saurait être assez éloquent pour exprimer mon amour ni ma profonde gratitude pour tous les efforts que vous n’avez jamais cessé de consentir pour mon instruction et mon bien-être. »

**À mes enseignants,**

« Je vous souhaite une vie pleine de succès et de bonheur. Et tous ceux qui m’ont poussé et incité à persévérer pour avoir un certain niveau culturel. »

**À mes amis et mes camarades de classe,**

« Pour tout leur amour, leur serviabilité, et leur soutien en leur souhaitant du fond du cœur une bonne démarche dans leur vie professionnelle. »

# **Remerciements**

**En cette période marquante de ma carrière professionnelle, je ressens le devoir d’exprimer ma gratitude envers les personnes qui ont joué un rôle significatif dans cette expérience enrichissante. Leur soutien indéfectible, leur expertise et leur dévouement ont été des piliers essentiels de cette réussite professionnelle.**

**Je tiens tout particulièrement à exprimer ma profonde reconnaissance à M. Yassine Fatahllah et M. Walid Azhari, mes encadrants professionnels chez Si Digital. Leur soutien constant, leurs conseils avisés et leur accompagnement précieux ont été d'une valeur inestimable. Je vous remercie sincèrement, M. Fatahllah et M. Azhari, pour m'avoir offert cette opportunité et pour avoir été des mentors exceptionnels tout au long de mon stage.**

**Je souhaite également adresser mes remerciements chaleureux à toute l'équipe de Si Digital. Travailler avec vous tous a été une expérience enrichissante. Votre collaboration, votre créativité et votre engagement continu dans la réalisation de nos objectifs ont été des éléments déterminants dans les résultats que nous avons obtenus. L'équipe de Si Digital, par son dynamisme et son expertise collective, a joué un rôle crucial dans le succès de ce projet. Chacun d'entre vous a apporté une contribution unique, et je suis honoré d'avoir eu la chance de travailler à vos côtés.**

**Je tiens également à exprimer ma gratitude envers M. Bousmah, mon encadrant académique. Son soutien académique, ses conseils pédagogiques et sa supervision rigoureuse ont été essentiels à la réalisation de ce projet. M. Bousmah, votre encadrement a grandement contribué à la qualité et à la réussite de cette expérience.**

**Enfin, je remercie ma famille et mes amis pour leur soutien inébranlable tout au long de cette expérience. Leur encouragement et leur compréhension ont été une source constante de réconfort et de motivation.**

**À M. Fatahllah, M. Azhari, toute l'équipe de Si Digital, ainsi qu'à M. Bousmah, ma famille et mes proches, je vous suis profondément reconnaissant pour votre confiance, votre collaboration et votre amitié. Cette expérience professionnelle restera un chapitre précieux de ma vie grâce à vous tous.**

**Merci… Une image contenant noir, obscurité

Description générée automatiquement**

# **Résumé**

Ce rapport met en lumière le travail réalisé lors de mon stage chez Le Matin, où j'ai développé une application basée sur le modèle Coqui TTS de Hugging Face. Contrairement à ce qui a été initialement décrit, mon projet ne concerne pas le développement d'une application mobile, mais plutôt d'une application utilisant ce modèle pour la synthèse vocale.

Le projet consiste en une application capable de convertir du texte en discours audio. Pour cela, j'ai choisi d'utiliser le modèle Coqui TTS, une technologie de pointe dans le domaine de la synthèse vocale. Cette application sera mise en conteneur à l'aide de Docker pour faciliter son déploiement et son utilisation.

L'application fonctionnera comme suit : l'utilisateur fournira du texte en entrée à l'application via une API, puis le m modèle Coqui TTS convertira ce texte en discours audio. L'API renverra ensuite un lien vers le fichier audio généré, que l'utilisateur pourra écouter en ligne ou télécharger selon ses besoins.

Le rapport détaillera les différentes étapes du processus de développement, y compris la sélection du modèle Coqui TTS, la mise en place de l'infrastructure Docker, et la création de l'API pour gérer les requêtes de conversion de texte en discours audio. En outre, il mettra en évidence les bénéfices et les limitations de l'application ainsi que les perspectives d'amélioration.

**Mots Clés :** Text To Speech TTS, Traitement du langage naturel (NLP), Intelligence artificielle (IA), Clonage de voix.

# **Abstract**

This report highlights the work carried out during my internship at Le Matin, where I developed an application based on Hugging Face's Coqui TTS model. Contrary to what was initially described, my project did not concern the development of a mobile application, but rather an application using this model for speech synthesis.

The project consists of an application capable of converting text into audio speech. For this purpose, I have chosen to use the Coqui TTS model, a state-of-the-art technology in the field of speech synthesis. This application will be containerized using Docker to facilitate its deployment and use.

The application will work as follows: the user will provide text input to the application via an API, then the Coqui TTS m model will convert this text into audio speech. The API will then return a link to the generated audio file, which the user can listen to online or download as required.

The report will detail the various stages of the development process, including the selection of the Coqui TTS model, the setting up of the Docker infrastructure, and the creation of the API to manage requests to convert text into audio speech. In addition, it will highlight the benefits and limitations of the application, as well as prospects for improvement**.**

**Keywords:** Text To Speech TTS, Natural Language Processing (NLP), Artificial Intelligence (AI), Voice Cloning.

# **Liste des abréviations**

|  |  |  |
| --- | --- | --- |
| **AI** | : | Artificial Intelligence |
| **API** | : | Application Programming Interface |
| **CNN** | : | Convolutional Neural Network |
| **GIT** | : | Global Information Tracker |
| **HF** | : | Hugging Face |
| **JSON** | : | JavaScript Object Notation |
| **LSTM** | : | Long Short-Term Memory |
| **NLP** | : | Natural Language Processing |
| **RNN** | : | Recurrent Neural Network |
| **TTS** | : | Text To Speech |
| **UML** | : | Unified Modeling Language |
| **URL** | : | Uniform Resource Locator |
| **VAE** | : | Variational Autoencoder |

# **Liste des figures**

[Figure 1:Logo de groupe le matin 3](#_Toc171514480)

[Figure 2:Histoire de l'entreprise 5](#_Toc171514481)

[Figure 3:Direction du groupe 6](#_Toc171514482)

[Figure 4:Les trois piliers de SCRUM 15](#_Toc171514483)

[Figure 5:Processus de SCRUM 16](#_Toc171514484)

[Figure 6:Diagramme de Gantt 17](#_Toc171514485)

[Figure 7:Backlog de Jira 19](#_Toc171514486)

[Figure 8:Architecture fonctionnelle du projet 34](#_Toc171514487)

[Figure 9:Diagramme de cas d’utilisation 34](#_Toc171514488)

[Figure 10:Diagramme de séquence 35](#_Toc171514489)

[Figure 11:Diagramme d'activité 36](#_Toc171514490)

[Figure 13:Logo de python 39](#_Toc171514491)

[Figure 14:Logo de flask 40](#_Toc171514492)

[Figure 15:Logo de CoquiTTS 41](#_Toc171514493)

[Figure 16:Logo de Visual Studio Code 43](#_Toc171514494)

[Figure 17:Logo de hugging-face 44](#_Toc171514495)

[Figure 18:Logo de starUML 45](#_Toc171514496)

[Figure 19:Logo de draw.io 45](#_Toc171514497)

[Figure 20:Logo de docker 46](#_Toc171514498)

[Figure 21:Logo de Git 47](#_Toc171514499)

[Figure 22:Logo de Termius 47](#_Toc171514500)

[Figure 23:Logo de Slack 48](#_Toc171514501)

[Figure 24:Logo de Jira 49](#_Toc171514502)

[Figure 25:Logo de Streamlit 49](#_Toc171514503)

[Figure 26:Interface de test pour l’entrée du texte 54](#_Toc171514504)

[Figure 27:Interface de test montrant le choix de l’audio pour le clonage 55](#_Toc171514505)

[Figure 28:Audio généré sur l’interface de test 55](#_Toc171514506)

[Figure 29:Démarrage de l’API en local 56](#_Toc171514507)

[Figure 30:Démarrage de l’API sur le serveur 56](#_Toc171514508)

[Figure 31:L’envoie du texte en local 57](#_Toc171514509)

[Figure 32:L’envoie du texte cote serveur 57](#_Toc171514510)

[Figure 33:reponse sous forme de lien de l’audio généré(local) 58](#_Toc171514511)

[Figure 34:reponse sous forme de lien de l’audio généré (Serveur) 58](#_Toc171514512)

[Figure 35:Soumission du texte à l’API 62](#_Toc171514513)

[Figure 36:Chargeement le dictionnaire d'abréviations et de corrections 62](#_Toc171514514)

[Figure 37:Remplacement des mots incorrectes et les abréviations 62](#_Toc171514515)

[Figure 38:La génération et le renvoie de l'URL de l'audio 63](#_Toc171514516)

[Figure 39:Code de l’interface de Streamlit 63](#_Toc171514517)

[Figure 40:Liaison de Streamlit avec l'API 64](#_Toc171514518)

[Figure 41:Initialisation du modèle de synthèse vocale (Coqui) 64](#_Toc171514519)

[Figure 42:La fonction responsable a transformé le texte en audio 64](#_Toc171514520)

# **Liste des tableaux**

[Tableau 1:Les différents modèles de TTS nationale et internationale 28](#_Toc171171724)

[Tableau 2:Benchmarking des différents modèles TTS disponible 29](#_Toc171171725)

[Tableau 3:Environnement matériel (Serveur) 44](#_Toc171171726)

# **Table de Matière**

[Dédicaces II](#_Toc171514521)

[Remerciements III](#_Toc171514522)

[Résumé IV](#_Toc171514523)

[Abstract V](#_Toc171514524)

[Liste des abréviations VI](#_Toc171514525)

[Liste des figures VII](#_Toc171514526)

[Liste des tableaux VIII](#_Toc171514527)

[Table de Matière IX](#_Toc171514528)

[Introduction Générale 1](#_Toc171514529)

[Chapitre I : Contexte général du projet 2](#_Toc171514530)

[Introduction 3](#_Toc171514531)

[1. Présentation de l’organisme d’accueil : 3](#_Toc171514532)

[1.1 Historique de l’entreprise 3](#_Toc171514533)

[1.2 La direction 6](#_Toc171514534)

[1.3 Le département SI Digital 7](#_Toc171514535)

[1.4 Le secteur d’activité de l’entreprise 8](#_Toc171514536)

[1.5 Chiffres-clés et clients de Groupe Le Matin 9](#_Toc171514537)

[2. Présentation du projet 10](#_Toc171514538)

[2.1 Problématique 10](#_Toc171514539)

[2.2 Etude de l’existant 11](#_Toc171514540)

[2.3 Critique de l’existant 11](#_Toc171514541)

[2.4 Les objectifs du projet 13](#_Toc171514542)

[2.5 Cahiers de charge 13](#_Toc171514543)

[3. Méthodologie et planification 14](#_Toc171514544)

[3.1 Méthodologie du projet 14](#_Toc171514545)

[3.2 Planification du projet 16](#_Toc171514546)

[3.3 Suivi du Projet avec Jira 19](#_Toc171514547)

[Conclusion 20](#_Toc171514548)

[Chapitre II : État de l'art sur les techniques Text-to-Speech 21](#_Toc171514549)

[Introduction 22](#_Toc171514550)

[1. Méthodes Historiques 22](#_Toc171514551)

[1.1 Synthèse par concaténation 22](#_Toc171514552)

[1.2 Synthèse par formants 22](#_Toc171514553)

[2. Méthodes Modernes 23](#_Toc171514554)

[2.1 Modèles statistiques (HMM) 23](#_Toc171514555)

[2.2 Réseaux neuronaux 23](#_Toc171514556)

[2.3 Modèles basés sur l'apprentissage profond 24](#_Toc171514557)

[3. Clonage de Voix 24](#_Toc171514558)

[3.1 Techniques de Clonage de Voix 25](#_Toc171514559)

[3.2 Applications et Défis 25](#_Toc171514560)

[4. Coqui TTS 25](#_Toc171514561)

[4.1 Présentation de Coqui TTS 25](#_Toc171514562)

[4.2 Comparaison avec d'autres systèmes TTS 26](#_Toc171514563)

[5. Perspectives et Défis 26](#_Toc171514564)

[5.1 Perspectives 26](#_Toc171514565)

[5.2 Défis 27](#_Toc171514566)

[6. Comparaison des Systèmes Text-to-Speech 27](#_Toc171514567)

[6.1 Comparaison des systèmes TTS au niveau national et international 28](#_Toc171514568)

[6.2 Benchmarking des systèmes TTS 28](#_Toc171514569)

[Conclusion 29](#_Toc171514570)

[Chapitre III : Etude fonctionnelle et conceptuelle 30](#_Toc171514571)

[Introduction 31](#_Toc171514572)

[1. Spécification des besoins 31](#_Toc171514573)

[1.1 Besoins fonctionnels 31](#_Toc171514574)

[1.2 Besoins non fonctionnels 32](#_Toc171514575)

[1.3 Architecture fonctionnelle 33](#_Toc171514576)

[2. Spécification conceptuelle 34](#_Toc171514577)

[2.1 Diagramme de cas d’utilisation 34](#_Toc171514578)

[2.2 Diagramme de séquence 35](#_Toc171514579)

[2.3 Diagramme d'activité 36](#_Toc171514580)

[Conclusion 37](#_Toc171514581)

[Chapitre IV : Mise en Œuvre 38](#_Toc171514582)

[Introduction 39](#_Toc171514583)

[1. Environnement de travail 39](#_Toc171514584)

[1.1 Choix technologique 39](#_Toc171514585)

[1.2 Environnement matériel 42](#_Toc171514586)

[1.3 Outil de développement 43](#_Toc171514587)

[1.4 Gestion de Version 50](#_Toc171514588)

[2. Implémentation de la solution 51](#_Toc171514589)

[2.1 Contraintes 51](#_Toc171514590)

[2.1.1 Contraintes techniques 51](#_Toc171514591)

[2.1.2 Contraintes ergonomiques 52](#_Toc171514592)

[2.2 Les bonnes pratiques 53](#_Toc171514593)

[2.3 Présentation de l'application 54](#_Toc171514594)

[2.3.1 Description générale 54](#_Toc171514595)

[2.3.2 Fonctionnalités principales 54](#_Toc171514596)

[2.4 Utilisation de l'API 56](#_Toc171514597)

[Fonctionnement détaillé 56](#_Toc171514598)

[Conclusion 58](#_Toc171514599)

[Conclusion Générale et Perspectives 59](#_Toc171514600)

[Bibliographie 60](#_Toc171514601)

[Webographie 61](#_Toc171514602)

[Annexes 62](#_Toc171514603)

[Annexe 1 : Code de l'API 62](#_Toc171514604)

[Annexe 2 : Code de l'Interface Streamlit 63](#_Toc171514605)

[Annexes 3 : Code du Modèle de Transformation de Texte en Audio 64](#_Toc171514606)

# **Introduction Générale**

L'expérience professionnelle joue un rôle crucial dans le développement académique et professionnel de chaque individu, offrant une opportunité unique d'appliquer les connaissances théoriques acquises dans un environnement réel. Dans ce rapport, nous partagerons le récit de notre expérience professionnelle, qui s'inscrit dans le cadre d'un stage effectué au sein du groupe Le Matin.

Ce stage, supervisé par M. Yassine Fatahllah et M. Bousmah, a été une opportunité passionnante de mettre en pratique nos compétences dans le domaine du développement d'applications. Notre objectif principal était de créer une application de synthèse vocale (Text to Speech) et de l'intégrer dans une API. Cette API permettrait aux utilisateurs de convertir du texte en discours audio, en fournissant un lien pour télécharger l'audio générer.

Tout au long de ce rapport, nous détaillerons le processus complet de conception, de développement et d'implémentation de cette application et de son API. Nous mettrons en lumière les choix technologiques stratégiques, les défis rencontrés et les solutions apportées tout au long du projet.

Ce rapport vise à être une rétrospective exhaustive de notre expérience professionnelle, mettant en évidence l'importance du soutien de nos encadrants et de notre équipe. Nous espérons que ce récit pourra également inspirer d'autres chercheurs en herbe et servir de ressource précieuse pour ceux qui s'intéressent au développement d'applications dans le domaine de la synthèse vocale.

# **Chapitre I : Contexte général du projet**

# Introduction

Ce chapitre inaugure notre exploration en débutant par une présentation de l'entreprise au sein de laquelle nous avons effectué notre stage de fin d'année. Nous poursuivrons ensuite en abordant l'étude et l'analyse du contexte existant, en exposant les défis rencontrés, et en décrivant le déroulement de notre stage ainsi que ses objectifs au sein de groupe le matin.

## 1. Présentation de l’organisme d’accueil :

Le Groupe Le Matin, un acteur majeur de l'information au Maroc à travers ses versions électronique et papier, se distingue par son engagement à fournir un traitement de qualité sur une variété de sujets et de plateformes. Pour répondre aux besoins diversifiés de son public et s'adapter aux évolutions de la consommation de l'information, il s'appuie sur deux stratégies clés : le renforcement de ses capacités éditoriales et l'investissement dans les nouvelles technologies de production et de diffusion. Cette approche lui permet de consolider sa position dans les domaines économique, social et politique, et de maintenir son leadership en proposant une information pertinente et de qualité à ses lecteurs.



Figure 1:Logo de groupe le matin

Mon choix de rejoindre le Groupe Le Matin, en particulier le département SI Digital, repose sur ma volonté de participer à l'intersection entre l'information et la technologie. La SI Digital, au cœur de la stratégie d'innovation du groupe, offre un environnement propice à l'apprentissage et à l'expérimentation dans le domaine des nouvelles technologies de l'information. Cette expérience représente une opportunité unique pour moi de mettre en pratique mes compétences techniques tout en contribuant à la transformation numérique d'un acteur clé de l'information au Maroc.

### 1.1 Historique de l’entreprise

Groupe Le MATIN est profondément associé à l'histoire du Maroc. Tourné résolument vers l'avenir, il s'adapte en permanence aux nouveaux défis des marques de presse.

* 1908-Création du groupe par Yves MAS : Le groupe Le Matin, initialement connu sous le nom de "Maroc-Soir", a été créé en 1908 par le journaliste influent Yves Mas. Le Maroc, alors sous le protectorat français depuis 1912, était en pleine mutation politique et sociale. En tant que fondateur, Yves Mas a joué un rôle essentiel dans l'évolution du journal, qui s'est rapidement imposé comme une figure influente dans le milieu médiatique marocain. Sous sa direction, le journal s'est imposé comme un acteur clé de la diffusion de l'information, contribuant ainsi à former l'opinion publique et à influencer les débats politiques de l'époque. Le journal a été connu à ses débuts pour son engagement en faveur de la liberté.
* 1971-Marocanisation du groupe par Moulay Ahmed Alaoui : Moulay Ahmed Alaoui est une personnalité politique marocaine. Il était membre de la famille royale Alaouite, cousin du roi Hassan II. Moulay Ahmed Alaoui a occupé divers postes dans l'administration marocaine et a joué un rôle important dans la modernisation et le développement du pays, notamment dans les domaines économique et social. En tant que directeur du Groupe Le Matin lors de sa marocanisation en 1971, il a contribué à la consolidation de ce média dans le paysage médiatique national.
* 1989-Lancement du quotidien as-Sahara al Maghribia : En 1989, le Groupe Le Matin a lancé le quotidien "As-Sahara Al Maghribia". Ce lancement s'inscrivait dans une stratégie visant à diversifier les publications du groupe et à étendre son influence médiatique. "As-Sahara Al Maghribia" a été conçu pour aborder les questions politiques, sociales, et économiques du Maroc, en particulier celles touchant les régions sahariennes.
* Repositionnement stratégique de l'imprimerie (1995- 1998) : Entre 1995 et 1998, le Groupe Le Matin a entrepris de repositionner sa filiale imprimerie comme une activité stratégique. Ce repositionnement a inclus l'introduction au Maroc de la première presse rotative quadri, une avancée technologique majeure pour l'industrie de l'impression dans le pays. Cette presse rotative quadri a permis d'améliorer considérablement la qualité et l'efficacité des impressions, ouvrant la voie à une production plus rapide et de meilleure qualité pour les publications du groupe.
* **2001 : Création du portail lematin.ma**
* **2011-Lancement de la version mobile lematin.ma**
* **2016-Lancement de la chaine LE MATIN TV et du site d'annonces en ligne annonces.lematin.ma**
* **2018-Lancement du site sports.lematin.ma en langue arabe**
* **2021-Lancement de l'impression de livres scolaires**
* **2022-Lancement du BookClub**
* **2023-Lancement de l'agence interne de brand content & events GLM Factory.**

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

Figure 2:Histoire de l'entreprise

### 1.2 La direction

Depuis 2013, **Mohammed Haitami** dirige Groupe LE MATIN, épaulé par un directeur général adjoint et quatre directeurs. Ils forment ensemble le Comex, qui pilote les orientations stratégiques de l’entreprise.

Une image contenant Visage humain, capture d’écran, personne, homme

Description générée automatiquement

Figure 3:Direction du groupe

L'organigramme du Groupe Le Matin présente la hiérarchie et les postes clés de l'entreprise. Au sommet se trouve **Mohammed Haitami**, le président et directeur général, qui supervise l'ensemble du groupe. Juste en dessous de lui se trouve Kamal El Alami, le directeur général adjoint, chargé d'assister la direction générale.

Sous **Kamal El Alami**, il y a quatre directeurs principaux, chacun s'occupant de domaines spécifiques des opérations de la société :

**· Nabil El Khalidy**- Directeur exécutif.

· **Hicham Mekouar**-Directeur du développement et de la stratégie.

· **Souhail Nhaili**- Directeur des rédactions.

· **Ali Alaoui Sossey**- Directeur administratif et financier.

Cette structure témoigne d'une répartition claire des responsabilités au sein de la direction générale, garant

### 1.3 Le département SI Digital

Le département SI Digital (Systèmes d'Information Digitale) joue un rôle crucial au sein du Groupe Le Matin. Il est responsable de la gestion et de l'optimisation des infrastructures numériques, des logiciels et des applications qui soutiennent les opérations quotidiennes de l'entreprise. Ce département se concentre sur l'innovation technologique pour améliorer l'efficacité et la productivité, tout en garantissant la sécurité et la stabilité des systèmes d'information.

Les principales missions du département SI Digital comprennent :

* **Développement et Maintenance des Applications** :

Création et maintenance des applications web et mobiles utilisées par les journalistes, éditeurs et autres employés du Groupe Le Matin.

Développement de solutions sur mesure pour répondre aux besoins spécifiques des différents départements de l'entreprise.

* **Gestion des Infrastructures Numériques** :

Administration des serveurs, bases de données et réseaux pour assurer un fonctionnement optimal et sécurisé.

Mise en place de solutions cloud pour améliorer la flexibilité et l'évolutivité des ressources informatiques.

* **Sécurité Informatique** :

Mise en œuvre de mesures de sécurité pour protéger les données sensibles et prévenir les cyberattaques.

Surveillance continue des systèmes pour détecter et répondre rapidement aux incidents de sécurité.

* **Innovation Technologique** :

Veille technologique pour identifier et intégrer les nouvelles technologies pouvant bénéficier à l'entreprise.

Développement de projets innovants tels que l'utilisation de l'intelligence artificielle et de l'apprentissage automatique pour automatiser et améliorer les processus métier.

* **Support Technique et Formation** :

Fourniture de support technique aux employés pour résoudre les problèmes informatiques quotidiens.

Organisation de sessions de formation pour améliorer les compétences techniques des employés et promouvoir l'utilisation efficace des outils numériques.

### 1.4 Le secteur d’activité de l’entreprise

Le Groupe Le Matin opère dans le secteur des médias et de la communication. En tant que l'un des principaux groupes de presse au Maroc, ses activités principales comprennent la production et la distribution de contenu journalistique sur différentes plateformes, telles que les journaux imprimés et les médias numériques. Voici une description détaillée de ses activités :

* Presse écrite

Le groupe Le Matin publie divers journaux et magazines traitant de divers sujets d'intérêt général. Le journal Le Matin propose des articles sur l'actualité nationale et internationale, l'économie, la politique, la culture et le sport.

* Médias numériques

Grâce aux progrès technologiques et à la digitalisation, le Groupe Le Matin a fait des investissements dans des plateformes numériques afin d'élargir sa visibilité. Il est responsable de la gestion de sites d'information qui diffusent des articles, des vidéos et des contenus multimédias, offrant ainsi aux lecteurs une portée en temps réel sur l'actualité.

* Imprimerie et édition

Le rôle essentiel de la filiale d'impression du groupe réside dans la fourniture de services d'impression de qualité. Cette activité stratégique satisfait non seulement les besoins de publication internes, mais offre également des services d'impression à d'autres entreprises et organisations.

* Publicité et communication

Les services de publicité et de communication proposés par le Groupe Le Matin offrent également aux entreprises la possibilité de promouvoir leurs produits et services sur les divers supports du groupe. Ce domaine englobe la commercialisation d'espaces publicitaires dans les journaux, sur les plateformes en ligne et dans les publications spécialisées.

### 1.5 Chiffres-clés et clients de Groupe Le Matin

Le Groupe Le Matin est l'un des principaux acteurs de la presse écrite et numérique au Maroc. Voici quelques chiffres-clés et informations sur les clients du groupe :

* **Chiffres-clés** :

**Année de création** : 1971

**Nombre d'employés** : Plus de 700 employés

**Tirage quotidien** : Environ 100,000 exemplaires

**Audience numérique** : Plus de 2 millions de visiteurs uniques par mois sur les plateformes numériques

**Publications** : Plusieurs titres renommés, y compris "Le Matin", "Assahraa Al Maghribiya" et divers suppléments spécialisés

* **Clients** :

**Lecteurs et Abonnés** : Le Groupe Le Matin sert une large base de lecteurs fidèles à travers le pays, comprenant des abonnés à ses éditions imprimées et numériques.

**Annonceurs** : Diverses entreprises et organisations utilisent les plateformes du Groupe Le Matin pour leurs campagnes publicitaires, bénéficiant de la portée et de l'influence de ses publications.

**Institutions** : Le groupe collabore avec des institutions gouvernementales et des ONG pour diffuser des informations et des annonces importantes.

**Partenariats** : Le Groupe Le Matin entretient des partenariats stratégiques avec d'autres médias et organisations pour promouvoir des événements et des initiatives communes.

Ces éléments soulignent l'importance du Groupe Le Matin dans le paysage médiatique marocain et sa capacité à évoluer avec les avancées technologiques pour mieux servir ses clients et son audience.

## 2. Présentation du projet

Cette section vise à présenter de manière détaillée le projet de développement d'une application de synthèse vocale pour Le Matin. Il s'agira d'explorer la problématique à laquelle répond ce projet, l'étude des solutions existantes, la critique de ces solutions, les objectifs du projet, et enfin, le cahier des charges. Ce chapitre servira de fondation pour comprendre les raisons et les attentes du projet, en fournissant un contexte complet et pertinent.

### 2.1 Problématique

Avec l'évolution rapide des technologies de l'information et des médias, les modes de consommation des contenus par les utilisateurs ont changé. De plus en plus de lecteurs se tournent vers les podcasts et les livres audio, ce qui entraîne une diminution de l'engagement avec les articles de presse écrits traditionnels. Cette tendance pose un défi majeur pour les médias écrits comme Le Matin, qui doivent trouver des moyens innovants pour attirer et retenir leur audience. Comment Le Matin peut-il adapter ses contenus pour répondre aux nouvelles préférences des utilisateurs tout en conservant la qualité et la richesse de ses articles ? L'intégration de la synthèse vocale, en particulier avec des capacités de clonage de voix, peut-elle offrir une solution viable pour rendre les articles accessibles sous forme audio, tout en offrant une expérience utilisateur naturelle et immersive ?

Le principal défi consiste à créer une solution qui automatise l'ensemble du processus de conversion de texte en audio tout en maintenant une qualité et une naturalité élevées. Notre objectif est de développer une application modulaire qui exploite le modèle linguistique Coqui TTS v2 pour générer automatiquement des fichiers audios à partir de texte. Le modèle Coqui TTS v2 inclut des voix intégrées de haute qualité et offre la possibilité de clonage de voix, permettant de reproduire fidèlement la voix de n'importe quel locuteur à partir d'un échantillon vocal. L'application doit également intégrer des technologies avancées telles que la personnalisation de la voix, le traitement multilingue et la génération d'audio haute fidélité pour répondre aux divers besoins des utilisateurs.

Cette solution doit fournir une interface conviviale permettant aux utilisateurs d'interagir avec l'IA de manière transparente. De plus, les audios générés doivent maintenir une cohérence et une qualité élevées en entraînant le modèle sur un corpus de textes adaptés à divers contextes d'utilisation. L'intégration harmonieuse des différents modules est cruciale pour éviter toute confusion et optimiser le flux de production des fichiers audio.

En résumé, le problème consiste à concevoir et à développer une application robuste et efficace qui remédie aux inefficacités actuelles dans la production de contenu vocal, répond aux besoins spécifiques des utilisateurs et surmonte les limites des solutions existantes. En utilisant le modèle Coqui TTS v2, nous visons à offrir une solution novatrice, cohérente et efficace pour la conversion automatique de texte en parole, tout en développant une expérience professionnelle importante dans l'utilisation des technologies avancées de synthèse vocale, incluant les voix intégrées et le clonage de voix, et en contribuant activement à l'évolution des pratiques dans ce domaine.

### 2.2 Etude de l’existant

Avant de concevoir et de développer notre application de synthèse vocale, il est essentiel de réaliser une étude de l'existant. Cette étude permet de comprendre les solutions actuelles, d'identifier les technologies utilisées et de définir les lacunes et opportunités présentes sur le marché.

* **Google Text-to-Speech** : Utilisé pour convertir le texte en parole à l'aide des réseaux de neurones. Il propose plusieurs voix et prend en charge de nombreuses langues.
* **Amazon Polly** : Un service de synthèse vocale de haute qualité qui convertit le texte en un discours réaliste. Il utilise des techniques avancées d'apprentissage profond pour synthétiser la parole.
* **Microsoft Azure Cognitive Services** : Offre des services de synthèse vocale avec des voix naturelles et des fonctionnalités avancées de personnalisation de la voix.
* **Coqui TTS** : Un projet open source offrant des modèles de synthèse vocale de haute qualité. Il est basé sur des techniques avancées d'apprentissage profond et permet un entraînement personnalisé pour obtenir des voix spécifiques.
* **Mozilla TTS** : Fournit des outils et des modèles pour la synthèse vocale, soutenant plusieurs langues et voix.

Les principales lacunes identifiées dans les solutions existantes incluent la qualité et le naturel des voix, la personnalisation et le clonage de voix spécifiques, la facilité d'intégration, et le support multilingue.

### 2.3 Critique de l’existant

Le Matin utilise actuellement AWS Polly, un service TTS commercial d'Amazon, pour convertir ses articles écrits en audio. Cette étude vise à évaluer l'existant, en se concentrant sur les avantages et les inconvénients d'AWS Polly, ainsi que sur les points d'amélioration potentiels.

**Description du Service :**

AWS Polly est un service de synthèse vocale basé sur le cloud qui convertit le texte en parole réaliste. Il offre une variété de voix et de langues, et propose des fonctionnalités telles que la modulation de la vitesse et de l'intonation.

**Avantages d'AWS Polly :**

- Large Choix de Voix et de Langues : Polly propose de nombreuses voix et langues, permettant à Le Matin de s'adresser à une audience internationale.

- Facilité d’Intégration : Étant une solution cloud, AWS Polly s'intègre facilement avec les autres services AWS et les systèmes existants de Le Matin.

- Disponibilité et Scalabilité : AWS Polly est hautement disponible et peut gérer des volumes importants de demandes, ce qui est essentiel pour un média à grande échelle comme Le Matin.

* **Points Positifs :**

- Fiabilité : En tant que service cloud, AWS Polly garantit une disponibilité continue et une scalabilité pour répondre aux besoins de Le Matin.

- Facilité d’Utilisation : Polly est facile à utiliser et à intégrer, avec une documentation complète et une assistance technique disponible.

* **Points Négatifs :**

- Qualité de la Voix : Bien que Polly offre une variété de voix, la qualité de certaines voix peut sembler mécanique et peu naturelle. Cela peut réduire l'engagement des utilisateurs et la satisfaction générale.

- Coût Élevé : AWS Polly est un service payant. Les coûts peuvent rapidement augmenter avec le volume élevé de contenu audio généré par un média comme Le Matin.

- Personnalisation Limitée : Les capacités de personnalisation de la voix et du ton sont limitées comparées à d'autres solutions, ce qui peut ne pas répondre aux besoins spécifiques de Le Matin pour des voix locales ou régionales.

L'utilisation d'AWS Polly par Le Matin présente des avantages significatifs en termes de fiabilité, de facilité d'intégration et de disponibilité. Cependant, les limitations en termes de qualité de voix, de coût et de personnalisation sont des défis notables. Ces aspects doivent être pris en compte lors de l'évaluation de solutions alternatives pour améliorer l'expérience utilisateur et optimiser les coûts. Une solution comme Coqui TTS pourrait être explorée pour surmonter ces limitations, offrant une meilleure qualité de voix et une flexibilité accrue tout en réduisant les coûts.

### 2.4 Les objectifs du projet

Les buts de ce stage sont variés et ambitieux. Dans un premier temps, l'objectif est de créer et de mettre au point une application modulaire permettant de convertir automatiquement du texte en audio en utilisant un model text-to-speech Coqui, répondant ainsi aux exigences particulières des utilisateurs, des développeurs et des administrateurs.

Par la suite, notre objectif est d'apporter une solution novatrice et efficace pour pallier les lacunes repérées dans les systèmes de synthèse vocale existants, en utilisant les modèles multilingues et multi-dataset de Coqui. En outre, notre objectif est de garantir une intégration fluide entre les divers modules supplémentaires, tels que l'analyse et le prétraitement de texte, la personnalisation de la voix et la conversion audio de haute qualité, afin d'améliorer la création de contenu vocal. Nous visons également à améliorer la qualité et la naturalité des voix générées en formant le modèle sur un corpus spécifique de textes adaptés à divers contextes d'utilisation.

De plus, l'un des principaux objectifs consiste à créer une interface utilisateur conviviale qui simplifie l'interaction entre les utilisateurs et le système de text-to-speech. Finalement, ce stage a pour objectif de mettre en œuvre les compétences que j'ai acquises lors de mes études en master systèmes informatiques distribués et Big Data et aussi en cycle de licence, tout en contribuant à améliorer la qualité et l'efficacité dans le domaine de la synthèse vocale.

En somme, les buts sont axés sur l'élaboration d'une solution novatrice, cohérente et efficace pour la conversion automatique de texte en audio, tout en développant une expérience professionnelle importante dans l'utilisation des technologies avancées de traitement du langage naturel et en contribuant activement à l'évolution des pratiques dans le domaine de la synthèse vocale.

### 2.5 Cahiers de charge

**Besoins Fonctionnels :**

* **Interface Utilisateur Conviviale** : Créez une interface intuitive où les utilisateurs peuvent facilement saisir leur texte, sélectionner une voix parmi plusieurs options intégrées, et télécharger les fichiers audios générés en quelques étapes simples.
* **API Documentée** : Développez une API robuste permettant aux développeurs d'intégrer la synthèse vocale dans d'autres applications. Assurez-vous que l'API est bien documentée, avec des méthodes claires pour convertir du texte en audio en utilisant le modèle Coqui TTS, et incluez des fonctionnalités avancées telles que la personnalisation de la voix et le support multilingue.
* **Conversion de Texte en Audio** : Intégrez le modèle Coqui TTS pour convertir le texte saisi en fichiers audio de haute qualité, en prenant en charge les fonctionnalités avancées comme le clonage de voix et la personnalisation des voix disponibles.
* **Support Multilingue** : Assurez-vous que l'application peut traiter plusieurs langues pour répondre aux besoins d'une audience internationale, en offrant une expérience utilisateur fluide dans divers contextes linguistiques.

**Besoins Non Fonctionnels :**

* **Performance** : La conversion de texte en audio doit être réalisée rapidement, sans temps de latence significatif.
* **Sécurité** : Les données des utilisateurs doivent être protégées et la confidentialité des contenus soumis doit être garantie.
* **Scalabilité** : L'application doit être capable de gérer un grand nombre de requêtes simultanées sans dégradation des performances.
* **Portabilité** : En utilisant Docker, l'application doit être facilement déployable sur différentes infrastructures.
* **Compatibilité** : Assurer la compatibilité de l'application avec une variété de systèmes d'exploitation et de dispositifs.

**Contexte et Cibles :**

* **Contexte** : L'application sera utile dans une multitude de contextes, tels que l'éducation, l'accessibilité, la productivité professionnelle, les loisirs, etc.
* **Cibles** : Les utilisateurs potentiels incluent les étudiants, les professionnels, les personnes avec des besoins spécifiques en matière d'accessibilité, les amateurs des audio book, les utilisateurs cherchant à améliorer leur prononciation dans une langue étrangère, etc.

Ce chapitre a pour but de fournir une vue d'ensemble claire et structurée du projet, en détaillant les problèmes identifiés, les solutions existantes, les critiques de ces solutions, et les objectifs ainsi que les exigences du projet.

## 3. Méthodologie et planification

Cette section décrit les méthodologies et les pratiques adoptées pour la gestion et le développement du projet. Elle détaille les processus de planification, de suivi et de coordination des différentes phases de réalisation, ainsi que les outils et techniques utilisés pour assurer l'efficacité et la qualité des livrables.

### 3.1 Méthodologie du projet

Scrum est l'une des méthodes de gestion de projet Agile. En tant que telle, elle vise à améliorer la productivité des équipes agiles même à distance, tout en permettant l'optimisation du produit grâce à un retour d'information régulier avec les utilisateurs finaux. Dans la méthode agile Scrum, l'objectif est de définir un cadre clair et précis à travers des itérations courtes, afin de faciliter la mise en œuvre du projet complexes. Ce cadre repose sur trois principes fondamentaux :

* Transparence : tous les membres de l'équipe doivent avoir connaissance des informations relatives au produit à développer.
* Inspiration : des évaluations régulières sont indispensables pour réajuster le projet si nécessaire.
* Adaptation : la mise en œuvre de nouvelles mesures est nécessaire lorsqu'une inspection montre des divergences dans les résultats mesurés.

Une image contenant texte, Police, logo, diagramme

Description générée automatiquement

Figure 4:Les trois piliers de SCRUM

Scrum se distingue par ses cycles de développement courts appelés « sprints » et ses rôles clairement définis tels que le Scrum Master, le Product Owner et l'équipe de développement. Tout au long du processus, des réunions et des révisions régulières permettent de suivre les progrès, de résoudre les problèmes et d'ajuster les priorités.

* **Sprints** : Cycles de développement courts avec des rôles clairement définis (Scrum Master, Product Owner, équipe de développement) et des réunions régulières pour contrôler et ajuster les priorités.
* **Backlog du produit** : Liste prioritaire des fonctionnalités à développer, établie en collaboration avec les parties prenantes.
* **Planification du sprint** : Sélection des éléments du backlog à développer au cours du prochain sprint avec des objectifs spécifiques.
* **DailyScrum**: Réunion quotidienne pour synchroniser les activités, partager les progrès et planifier les actions futures
* **Sprint review** : Démonstration des fonctionnalités développées à la fin de chaque sprint avec les commentaires des parties prenantes afin d'ajuster le carnet de commandes et les priorités.
* **Rétrospective du sprint** : Réflexion sur le sprint écoulé afin d'identifier les points forts et les points faibles et proposer des améliorations.
* **Incrément de produit** : Livraison d'un incrément de produit, une version fonctionnelle.

Une image contenant texte, capture d’écran, Police, Graphique

Description générée automatiquement

Figure 5:Processus de SCRUM

Après avoir défini les principes fondamentaux de la méthodologie Scrum, je vais maintenant l'appliquer à mon projet. Le projet sera divisé en sept sprints, chacun ayant un objectif spécifique et des livrables clairement définis. Cette approche me permettra de gérer efficacement le projet tout en assurant une progression constante vers mon objectif final.

### 3.2 Planification du projet

Pour visualiser et suivre l'avancement du projet, nous avons utilisé un diagramme de Gantt. Ce diagramme nous permet de planifier les différentes phases du projet, d'identifier les dépendances entre les tâches et de gérer les délais.

Le diagramme de Gantt est divisé en plusieurs phases principales :

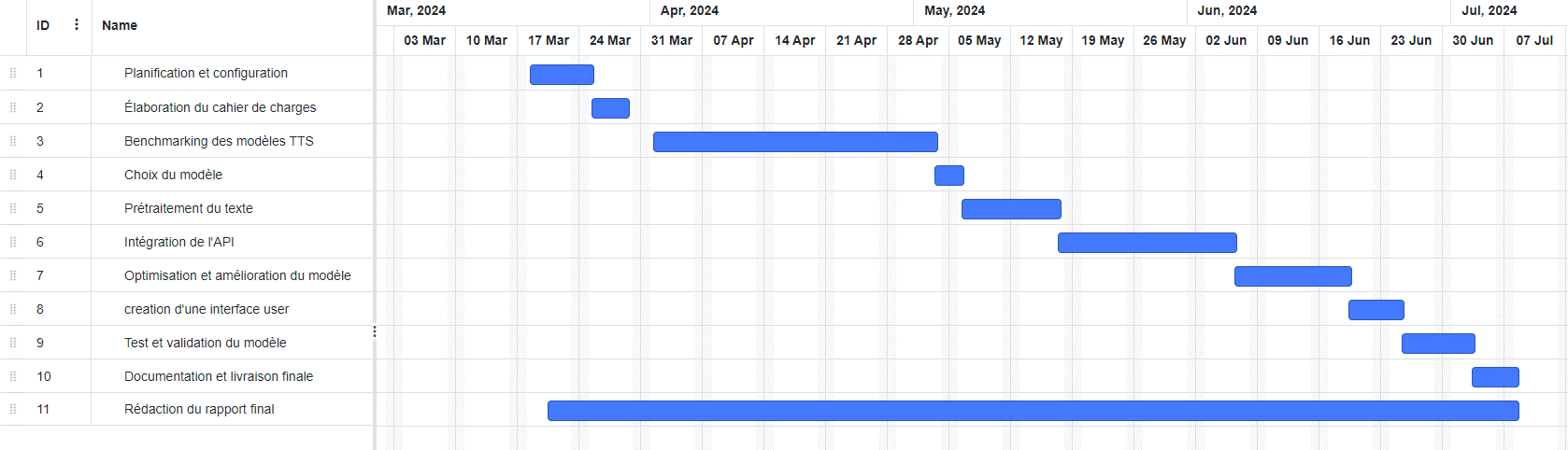


Figure 6:Diagramme de Gantt

**Élaboration du cahier de charges**

* Définir les exigences et les spécifications du projet.
* Identifier les fonctionnalités principales et les contraintes techniques.

**Planification et configuration**

* Planifier les étapes du projet et les ressources nécessaires.
* Configurer l'environnement de développement et les outils nécessaires.

**Benchmarking des modèles TTS**

* Évaluer et comparer différents modèles de synthèse vocale.
* Sélectionner les modèles les plus performants pour le projet.

**Choix du modèle**

* Sélectionner le modèle TTS optimal en fonction des résultats du benchmarking.
* Justifier le choix basé sur des critères de performance et de compatibilité.

**Prétraitement du texte (création d’un dictionnaire pour faire cette tâche)**

* Développer un dictionnaire pour la correction des abréviations et des erreurs.
* Implémenter des techniques de prétraitement pour améliorer la qualité du texte.

**Intégration de l'API**

* Développer et intégrer l'API pour la conversion du texte en audio.
* Assurer la communication entre le frontend et le backend.

**Optimisation et amélioration du modèle**

* Affiner le modèle TTS pour améliorer la qualité et la naturalité de la synthèse vocale.
* Réduire les coûts computationnels et améliorer l'efficacité.

**Création de l'interface utilisateur**

* Développer une interface utilisateur intuitive pour l'application.
* Assurer une expérience utilisateur fluide et accessible.

**Test et validation du modèle**

* Effectuer des tests unitaires et des tests système pour vérifier le fonctionnement du modèle.
* Valider la performance du modèle avec des utilisateurs finaux.

**Documentation et livraison finale**

* Rédiger la documentation technique et utilisateur du projet.
* Livrer le produit final avec toutes les fonctionnalités et améliorations.

**Rédaction du rapport final**

* Documenter les étapes du projet, les résultats obtenus et les conclusions.
* Préparer le rapport final pour la présentation aux parties prenantes.

### 3.3 Suivi du Projet avec Jira

Pour assurer un suivi efficace du projet et une gestion transparente des tâches, nous avons utilisé Jira, un outil de gestion de projet Agile. Jira nous permet de suivre l'avancement des tâches, de gérer les sprints et de faciliter la collaboration au sein de l'équipe.

Fonctionnalités Utilisées

1. **Backlog** : Liste des tâches à réaliser, priorisée par le Product Owner. Chaque tâche (ou User Story) est décrite en termes de critères d'acceptation et de points d'estimation.
2. **Sprints** : Planification et gestion des sprints, avec des tableaux Kanban pour visualiser l'état d'avancement des tâches (à faire, en cours, terminé).
3. **Rapports** : Génération de rapports pour suivre la progression des sprints, la vélocité de l'équipe, et identifier les obstacles.
4. **Tableaux de bord** : Vue d'ensemble personnalisable pour chaque membre de l'équipe, affichant les tâches assignées, les délais et les notifications importantes.

Mise en œuvre dans notre projet

* **Création des User Stories** : Définition des fonctionnalités à développer, avec des critères d'acceptation clairs.
* **Planification des Sprints** : Organisation des User Stories en sprints, en fonction des priorités et des estimations de l'équipe.
* **Suivi et Collaboration** : Utilisation des tableaux Kanban pour suivre l'avancement des tâches, avec des mises à jour régulières par chaque membre de l'équipe.
* **Rapports de Sprint** : Analyse des performances de chaque sprint pour ajuster la planification et améliorer les processus.



Figure 7:Backlog de Jira

# Conclusion

En conclusion, ce premier chapitre a posé les bases fondamentales de notre projet. Nous avons présenté l'organisme d'accueil, exposé les objectifs du projet et détaillé notre approche méthodologique, combinant le diagramme de Gantt pour la planification globale et la méthode agile Scrum pour une gestion réactive. En outre, l'intégration de l'outil de suivi Jira garantira une gestion efficace des tâches et une collaboration optimale au sein de l'équipe. Cette méthodologie hybride offre à la fois une structure solide et une flexibilité nécessaire pour s'adapter aux évolutions du projet. Ces éléments posent ainsi les jalons d'une gestion efficace et efficiente pour la réalisation réussie de notre projet.

# **Chapitre II : État de l'art sur les techniques Text-to-Speech**

# Introduction

Le domaine de la synthèse vocale, ou Text-to-Speech (TTS), a connu des progrès significatifs depuis ses débuts. Ces avancées ont permis de créer des systèmes capables de convertir du texte en parole avec une qualité et une naturalité de plus en plus proche de la voix humaine. Ce chapitre présente un état de l'art des principales techniques utilisées dans le domaine du TTS, depuis les méthodes historiques jusqu'aux approches modernes basées sur l'apprentissage profond.

## 1. Méthodes Historiques

Les premières techniques de synthèse vocale se sont concentrées sur des approches relativement simples et directes pour convertir le texte en parole. Ces méthodes incluent principalement la synthèse par concaténation et la synthèse par formants.

### 1.1 Synthèse par concaténation

La synthèse par concaténation est l'une des premières techniques développées pour la TTS. Elle consiste à assembler des segments de parole préenregistrés, appelés unités, pour former des mots et des phrases. Les unités peuvent être des diphones, des triphones, des syllabes ou des mots entiers. Cette méthode repose sur une base de données vocale étendue et nécessite des algorithmes sophistiqués pour choisir et combiner les unités de manière fluide.

* **Avantages :** Haute qualité sonore pour des phrases simples et des contextes spécifiques.
* **Inconvénients :** Limité par la taille de la base de données et manque de flexibilité.

L'un des exemples les plus notables de synthèse par concaténation est le système de synthèse vocale Festival, développé à l'Université d'Édimbourg. Ce système utilise des diphones comme unités de base et a été largement utilisé dans les premières applications de TTS.

### 1.2 Synthèse par formants

La synthèse par formants utilise des modèles paramétriques pour reproduire les caractéristiques acoustiques de la voix humaine. Les formants, qui sont les pics dans le spectre de la voix, sont modélisés pour générer le signal de parole. Cette méthode permet de synthétiser la parole à partir de paramètres linguistiques tels que les phonèmes, la prosodie, etc.

* **Avantages :** Flexibilité et faible besoin de données.
* **Inconvénients :** Qualité sonore souvent moins naturelle comparée à la synthèse par concaténation.

Un exemple historique de synthèse par formants est le système de synthèse vocale Votrax, qui a été utilisé dans les années 1970 et 1980. Ce système permettait de générer des discours intelligibles mais avec une qualité vocale très mécanique.

À ce stade, les méthodes historiques, bien qu'efficaces dans certains contextes, montrent des limitations importantes en termes de naturalité et de flexibilité. Ces contraintes ont motivé le développement de nouvelles approches plus sophistiquées et adaptatives.

## 2. Méthodes Modernes

Avec l'évolution de la technologie et des techniques de traitement du signal, les méthodes modernes de TTS ont vu le jour. Ces approches intègrent des techniques statistiques et des réseaux neuronaux pour améliorer la qualité et la naturalité de la parole synthétisée.

### 2.1 Modèles statistiques (HMM)

Les modèles de Markov cachés (HMM) ont été une avancée majeure dans la TTS. Ils modélisent la séquence de phonèmes et leur durée ainsi que les paramètres acoustiques de la parole. Cette approche statistique permet de générer de la parole synthétique en utilisant des techniques de modélisation probabiliste.

* **Avantages :** Bonne gestion des variations dans les données et capacité à générer une parole fluide.
* **Inconvénients :** Qualité vocale limitée par les hypothèses simplificatrices des HMM.

Les systèmes basés sur HMM, comme le système HTS (HMM-based Speech Synthesis System), ont montré une grande efficacité dans les applications pratiques, permettant une génération de parole plus fluide et une gestion améliorée des variations prosodiques.

### 2.2 Réseaux neuronaux

Les réseaux neuronaux ont révolutionné le domaine de la TTS grâce à leur capacité à apprendre des représentations complexes et à modéliser des relations non linéaires. Les architectures telles que les réseaux de neurones récurrents (RNN) et les réseaux de neurones convolutifs (CNN) ont été utilisées pour améliorer la qualité de la synthèse vocale.

* **Réseaux de neurones récurrents (RNN) et LSTM :** Les RNN et leurs variantes comme les LSTM (Long Short-Term Memory) sont capables de modéliser les dépendances temporelles dans les données vocales, ce qui améliore la fluidité et la naturalité de la parole générée.
* **Réseaux de neurones convolutifs (CNN) :** Les CNN sont efficaces pour extraire les caractéristiques locales du signal vocal, ce qui est utile pour modéliser les aspects acoustiques fins de la parole.

### 2.3 Modèles basés sur l'apprentissage profond

Les modèles basés sur l'apprentissage profond, tels que WaveNet de Google et Tacotron de Google Brain, ont marqué un tournant décisif. WaveNet génère la parole à partir d'une distribution conditionnelle sur les échantillons audio précédents, tandis que Tacotron utilise une architecture de séquence à séquence pour convertir des représentations textuelles en spectrogrammes, qui sont ensuite transformés en signal audio.

* **WaveNet :** Utilise des réseaux neuronaux convolutionnels dilatés pour générer de l'audio à haute qualité.
* **Tacotron et Tacotron 2 :** Convertissent directement le texte en spectrogrammes mélodiques, qui sont ensuite synthétisés en signal audio.
* **Avantages :** Haute qualité et naturalité de la parole synthétique, capable de capturer des nuances subtiles de la voix humaine.
* **Inconvénients :** Nécessitent des ressources computationnelles importantes pour l'entraînement et la génération.

WaveNet a été particulièrement révolutionnaire en démontrant que des réseaux neuronaux profonds pouvaient produire de la parole synthétique avec une qualité proche de celle de la voix humaine naturelle, surmontant de nombreuses limitations des méthodes précédentes.

Les méthodes modernes ont largement surmonté les limitations des approches historiques, offrant une qualité de parole beaucoup plus naturelle et une flexibilité accrue. Cependant, elles introduisent également de nouveaux défis, notamment en termes de coûts computationnels et de complexité.

## 3. Clonage de Voix

Le clonage de voix est une technique avancée dans le domaine de la TTS qui vise à reproduire la voix d'un individu spécifique à partir d'un ensemble limité d'échantillons vocaux. Cette technologie utilise des modèles d'apprentissage profond pour capturer les caractéristiques uniques de la voix d'un locuteur et les appliquer à la synthèse vocale.

### 3.1 Techniques de Clonage de Voix

* **Voice Conversion :** Utilise des modèles tels que les auto-encodeurs variés (VAE) et les réseaux antagonistes génératifs (GAN) pour transformer la voix d'un locuteur source en celle d'un locuteur cible.
* **Few-Shot Learning :** Permet de créer des modèles de voix avec un nombre limité d'enregistrements du locuteur cible, en s'appuyant sur des architectures comme les réseaux de neurones à mémoire externe (MemNN).

Des projets comme VoCo d'Adobe ont démontré comment il est possible de manipuler et cloner des voix avec une grande précision, ouvrant de nouvelles possibilités tout en soulevant des questions éthiques.

### 3.2 Applications et Défis

* **Applications :** Personnalisation des assistants virtuels, doublage de films, jeux vidéo, et restauration de voix pour les personnes ayant perdu l'usage de la parole.
* **Défis :** Problèmes de confidentialité et de sécurité, risques d'usurpation d'identité vocale, et défis éthiques liés à l'utilisation non consensuelle de la voix d'un individu.

Le clonage de voix représente une avancée significative mais soulève également des questions éthiques et de sécurité importante, nécessitant une réglementation et des pratiques éthiques rigoureuses.

## 4. Coqui TTS

Le modèle Coqui TTS est un exemple avancé de l'application des techniques modernes de TTS basées sur l'apprentissage profond. Coqui TTS est un projet open-source qui utilise des architectures de pointe pour fournir une synthèse vocale de haute qualité.

### 4.1 Présentation de Coqui TTS

* **Architecture :** Coqui TTS utilise une architecture de type Tacotron 2 combinée à un vocodeur WaveRNN, permettant une conversion efficace du texte en spectrogrammes mélodiques et la génération de signal audio de haute qualité.
* **Avantages :** Qualité vocale élevée, flexibilité pour différents accents et langues, et possibilité de personnalisation pour des voix spécifiques.
* **Inconvénients :** Besoin de ressources computationnelles importantes pour l'entraînement, et complexité de la configuration initiale.

### 4.2 Comparaison avec d'autres systèmes TTS

Coqui TTS se distingue par son approche open-source, permettant une large adoption et une personnalisation facile par la communauté. Comparé à des systèmes commerciaux comme WaveNet et Tacotron de Google, Coqui TTS offre une alternative flexible et accessible, tout en maintenant une qualité vocale compétitive.

Coqui TTS se positionne comme un leader dans l'innovation et la démocratisation de la technologie TTS, offrant des possibilités de recherche et de développement continues.

## 5. Perspectives et Défis

Les recherches actuelles se concentrent sur l'amélioration de la naturalité de la synthèse vocale, la réduction des coûts computationnels et l'adaptation de la voix synthétisée à des contextes et des locuteurs variés. Plusieurs défis importants subsistent dans le domaine du Text-to-Speech (TTS).

### 5.1 Perspectives

Les avancées récentes dans les technologies de la synthèse vocale ouvrent de nouvelles perspectives prometteuses :

**Amélioration de la naturalité de la synthèse vocale :** Les recherches se concentrent sur l'amélioration de la fluidité et de la naturalité de la parole synthétisée. Des techniques avancées, telles que les réseaux de neurones récurrents (RNN), les réseaux de neurones convolutifs (CNN) et les modèles de séquence à séquence (Seq2Seq), jouent un rôle crucial dans cette amélioration.

**Réduction des coûts computationnels :** L'un des objectifs majeurs est de rendre les systèmes TTS plus efficaces en termes de coût computationnel. L'utilisation de modèles plus légers et l'optimisation des algorithmes d'apprentissage permettent de réduire la consommation de ressources tout en maintenant une qualité élevée.

**Adaptation contextuelle :** Le développement de modèles capables de s'adapter à différents contextes et locuteurs est essentiel. Cela comprend la capacité de modifier la prosodie, le ton et l'intonation en fonction du contenu et du contexte, offrant ainsi une expérience utilisateur plus personnalisée.

### 5.2 Défis

**Adaptation à des voix spécifiques :** Un défi majeur consiste à créer des systèmes capables de générer la voix de locuteurs spécifiques avec peu de données d'entraînement. Les approches de clonage de voix et de transfert d'apprentissage sont explorées pour relever ce défi. Cela permettrait une personnalisation accrue et l'utilisation de voix synthétisées dans des applications variées, telles que les assistants vocaux et les livres audio.

**Synthèse vocale multilingue :** Le développement de modèles capables de synthétiser la parole dans plusieurs langues avec des transitions fluides est un autre défi important. Cela nécessite des systèmes capables de gérer des caractéristiques linguistiques et phonétiques diversifiées, assurant une qualité et une naturalité constantes indépendamment de la langue.

**Éthique et biais :** Les questions éthiques et les biais potentiels dans les systèmes de TTS doivent être abordés. Les chercheurs doivent s'assurer que les systèmes TTS ne perpétuent pas les stéréotypes ou les préjugés en matière de représentation des voix et des accents. La transparence dans la conception et l'entraînement des modèles, ainsi que l'inclusion de divers ensembles de données, sont cruciales pour réduire ces biais.

Ces perspectives et défis montrent que, bien que des progrès significatifs aient été réalisés dans le domaine du TTS, des efforts continus sont nécessaires pour améliorer la qualité, l'efficacité et l'équité des systèmes de synthèse vocale.

## 6. Comparaison des Systèmes Text-to-Speech

La technologie Text-to-Speech (TTS) est de plus en plus utilisée par les médias pour rendre leurs contenus accessibles sous forme audio. Cela permet non seulement d'améliorer l'accessibilité pour les personnes malvoyantes ou souffrant de troubles de la lecture, mais aussi de répondre aux besoins des utilisateurs préférant les formats audios. Le tableau ci-dessous présente une comparaison des services TTS utilisés par divers journaux et médias au niveau national et international.

### 6.1 Comparaison des systèmes TTS au niveau national et international

|  |  |  |  |
| --- | --- | --- | --- |
| Nom Du journal/Media | Description du service TTS | Niveau | Technologie utilisée |
| Le Matin | Propose des versions audios de ses articles pour une accessibilité accrue. | National | Amazon Web Service (Polly) |
| L'Économiste | Utilise des services TTS pour convertir des articles économiques en audio. | National | IBM Watson Text-to-Speech |
| The Washington Post | Propose des versions audios de ses articles via une application dédiée. | International | Google Cloud Text-to-Speech |
| The New York Times | Convertit les articles en audio pour une accessibilité accrue. | International | Amazon Web Service (Polly) |
| Le Monde | Fournit des articles en audio pour les lecteurs francophones. | International | Azure Cognitive Services |
| The Guardian | Utilise des services TTS pour convertir des articles en podcasts. | International | Amazon Web Service (Polly) |
| BBC News | Offre des podcasts et des articles en format audio. | International | Custom TTS Engine |

Tableau 1:Les différents modèles de TTS nationale et internationale

### 6.2 Benchmarking des systèmes TTS

Pour évaluer les performances des différents modèles Text-to-Speech (TTS), il est essentiel de comparer plusieurs aspects clés qui influencent leur utilisation et leur efficacité. Le tableau ci-dessous offre un aperçu des caractéristiques de divers modèles TTS, incluant Amazon Polly, Melo TTS, Facebook TTS, Microsoft TTS, Bark TTS, et Coqui TTS. Ces critères incluent la disponibilité gratuite, la vitesse, les capacités de formation, le réglage fin, la possibilité de gérer des entrées illimitées, et la personnalisation des paramètres vocaux.

**Critères d'évaluation :**

* **Gratuité (Free)** : Indique si le modèle est disponible gratuitement ou non.
* **Vitesse (Speed)** : Évalue la rapidité avec laquelle le modèle peut générer du texte en parole.
* **Capacités de formation (Training)** : Vérifie si le modèle peut être formé ou s'il utilise des modèles pré-entrainés.
* **Réglage fin (Fine Tuning)** : Vérifie si le modèle permet un ajustement précis des paramètres pour améliorer la qualité de la synthèse vocale.
* **Entrée illimitée (Unlimited Input)** : Indique si le modèle peut gérer des textes de longueur illimitée.
* **Paramètres vocaux (Voice Params)** : Vérifie si le modèle permet de personnaliser divers paramètres de la voix synthétisée, comme le ton, la vitesse et l'intonation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model Name | Free | Speed | Training | Fine Tuning | Unlimited Input | Voice Params |
| Polly AWS |  | ✓ |  |  | ✓ |  |
| Melo TTS | ✓ | ✓ | ✓ | ✓ | ✓ |  |
| Facebook TTS |  | ✓ |  |  | ✓ | ✓ |
| Microsoft TTS | ✓ | ✓ | ✓ |  | ✓ |  |
| Bark TTS | ✓ | ✓ | ✓ |  | ✓ | ✓ |
| Coqui TTS | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Tableau 2:Benchmarking des différents modèles TTS disponible

# **Conclusion**

La synthèse vocale a évolué grâce aux progrès en modélisation statistique et réseaux neuronaux, notamment via les techniques d'apprentissage profond qui ont amélioré la qualité et la naturalité de la parole synthétisée. Cependant, des défis persistent, tels que la personnalisation, le multilinguisme et la gestion des biais. Ce chapitre résume les techniques TTS et leurs avancées, soulignant le choix du modèle Coqui TTS pour sa flexibilité, son caractère open-source et sa capacité à s'adapter aux besoins variés tout en maintenant une haute qualité.

# **Chapitre III : Etude fonctionnelle et conceptuelle**

# Introduction

Ce chapitre est consacré à l'étude fonctionnelle et conceptuelle du projet. Il s'agit d'une étape cruciale qui vise à définir précisément les besoins du système ainsi que son architecture fonctionnelle et conceptuelle. Cette analyse permet de s'assurer que le système répondra aux attentes des utilisateurs et respectera les contraintes techniques. Nous aborderons d'abord la spécification des besoins, en distinguant les besoins fonctionnels et non fonctionnels, puis nous décrirons l'architecture fonctionnelle. Ensuite, nous nous concentrerons sur la spécification conceptuelle, incluant l'identification des acteurs du système et la modélisation des interactions via des diagrammes de cas d'utilisation et de séquence.

## 1. Spécification des besoins

La spécification des besoins est une étape cruciale pour la réussite du projet. Elle permet de définir de manière précise et exhaustive les attentes et les exigences des utilisateurs finaux ainsi que les contraintes techniques à respecter. Cette section est divisée en trois parties : les besoins fonctionnels, les besoins non fonctionnels, et l'architecture fonctionnelle.

### 1.1 Besoins fonctionnels

**Saisie du Texte :**

* Permettre aux utilisateurs de saisir ou de copier-coller le texte de l'article dans l'interface.
* Option pour les utilisateurs de sélectionner la langue et la voix souhaitée pour la synthèse vocale.

**Prétraitement du Texte :**

* Corriger automatiquement les fautes de frappe et les abréviations courantes.
* Filtrer le texte pour les caractères spéciaux ou non pris en charge.

**Synthèse Vocale :**

* Convertir le texte en audio à l'aide du modèle Coqui TTS.
* Offrir la possibilité de sélectionner différentes voix (standard ou clonées) pour la synthèse vocale.

**Gestion des Fichiers Audio :**

* Générer et stocker les fichiers audio résultants sur un serveur sécurisé.
* Fournir des liens URL aux utilisateurs pour télécharger ou écouter les fichiers audios.

**Interface Utilisateur :**

* Fournir une interface utilisateur intuitive pour la saisie du texte et la sélection des options de synthèse vocale.
* Afficher les liens de téléchargement ou de lecture des fichiers audio générés.

**API :**

* Développer une API REST pour la communication entre l'interface utilisateur et le modèle de synthèse vocale.
* Assurer que l'API puisse recevoir des requêtes de texte et renvoyer des réponses avec des liens URL des fichiers audio.

### 1.2 Besoins non fonctionnels

**Performance** :

* **Temps de réponse** : Le temps de conversion de texte en parole doit être inférieur à 2 secondes pour des textes courts (moins de 200 mots).
* **Efficacité** : Le système doit être capable de gérer des conversions de texte en parole de manière fluide sans ralentissement notable, même pour des textes longs.

**Scalabilité** :

* **Gestion des utilisateurs simultanés** : Le système doit pouvoir supporter au moins 1000 utilisateurs simultanés sans dégradation significative de la performance.
* **Évolutivité** : La conception du système doit permettre l'ajout de nouvelles fonctionnalités ou l'augmentation de la capacité sans nécessiter de révisions majeures de l'architecture.

**Sécurité** :

* **Protection des données** : Les données de l'utilisateur, y compris les textes saisis et les fichiers audio générés, doivent être protégées par des mécanismes de chiffrement.
* **Authentification et autorisation** : Mettre en place des systèmes robustes pour authentifier les utilisateurs et autoriser l'accès aux fonctionnalités en fonction des rôles et des permissions.

**Fiabilité** :

* **Disponibilité** : Le système doit être disponible au moins 99,9 % du temps, avec des interruptions minimales pour maintenance.
* **Tolérance aux pannes** : Mettre en place des mécanismes de reprise après sinistre et de tolérance aux pannes pour assurer la continuité du service en cas de problème technique.

**Utilisabilité** :

* **Facilité d'utilisation** : L'interface utilisateur doit être intuitive et facile à utiliser pour tous les types d'utilisateurs, y compris ceux qui ne sont pas techniquement compétents.
* **Documentation** : Fournir une documentation complète et claire pour aider les utilisateurs à comprendre et utiliser toutes les fonctionnalités du système.

**Maintenance** :

* **Facilité de maintenance** : Le code doit être bien structuré et documenté pour faciliter la maintenance et les mises à jour futures.
* **Mises à jour régulières** : Prévoir des cycles de mise à jour réguliers pour améliorer les fonctionnalités existantes, corriger les bogues et ajouter de nouvelles fonctionnalités.

### 1.3 Architecture fonctionnelle

Le système de synthèse vocale est basé sur une architecture client-serveur, où les requêtes sont envoyées depuis une page web vers une API sur un serveur, qui convertit le texte en parole à l'aide de Coqui TTS v2 et renvoie le lien vers le fichier audio généré.

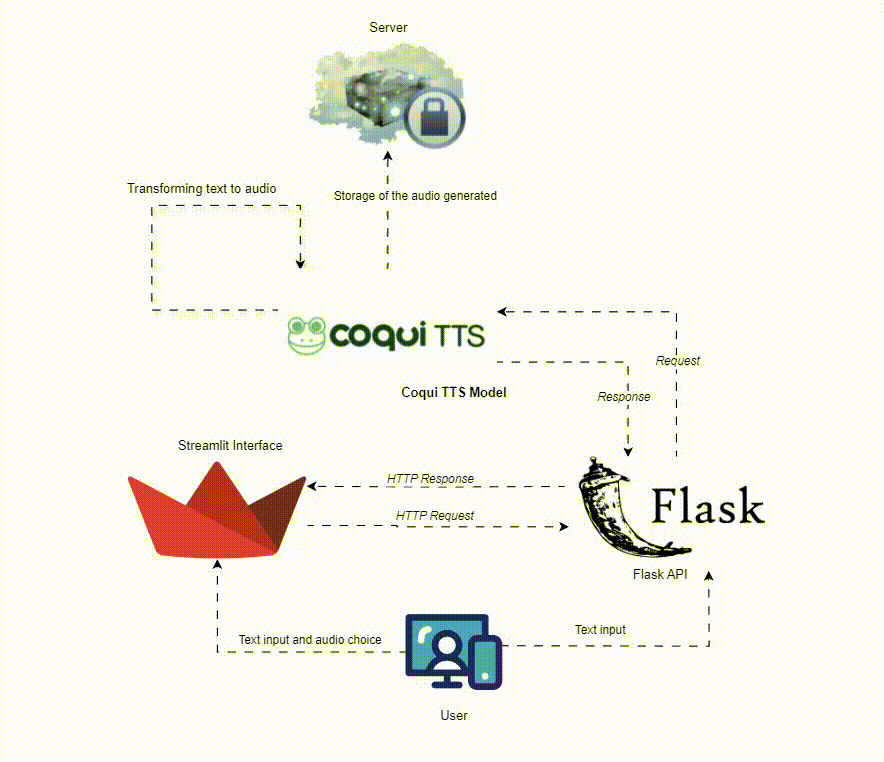


Figure 8:Architecture fonctionnelle du projet

## 2. Spécification conceptuelle

Dans cette section, nous définissons les différents acteurs du système, puis nous illustrons les interactions entre ces acteurs à l'aide de diagrammes de cas d'utilisation, diagrammes de séquence et de diagramme d’activité.

### 2.1 Diagramme de cas d’utilisation

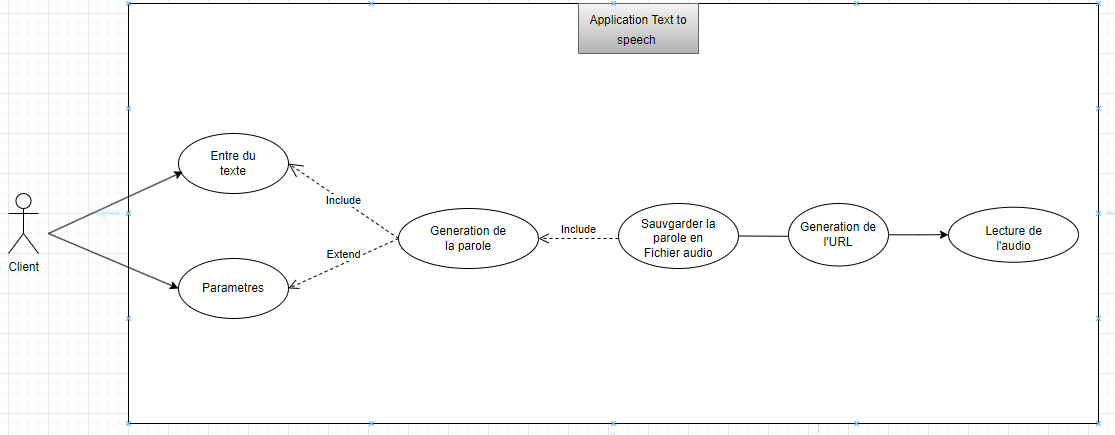


Figure 9:Diagramme de cas d’utilisation

Ce diagramme de cas d'utilisation représente les différentes fonctionnalités que l’application (Text to speech) offre.

L'utilisateur ou le client initie le processus en fournissant le texte à transformer en parole. Ensuite, il a la possibilité de personnaliser les paramètres de l'application, tels que la sélection de la voix et sa vitesse. Une fois ces étapes accomplies, l'application génère la parole correspondante. Ensuite, l'utilisateur a la possibilité de sauvegarder l'audio générer. Enfin, il peut écouter l'audio sauvegarder à sa convenance.

### 2.2 Diagramme de séquence

Une image contenant texte, capture d’écran, nombre, ligne

Description générée automatiquement

Figure 10:Diagramme de séquence

Ce diagramme illustre la séquence d’actions entre l’utilisateur et l’application lors de l’utilisation de la fonction de texte a la parole.

L’application envoie un message d’accueil à l’utilisateur.

L’utilisateur saisi le texte puis modifie les paramètres en suite il lance la transformation du texte en parole.

L’application transforme le texte en audio.

L’application génère l’audio puis le sauvegarde.

Après l’application génère l’URL de l’audio et le renvoie à l’utilisateur

### 2.3 Diagramme d'activité

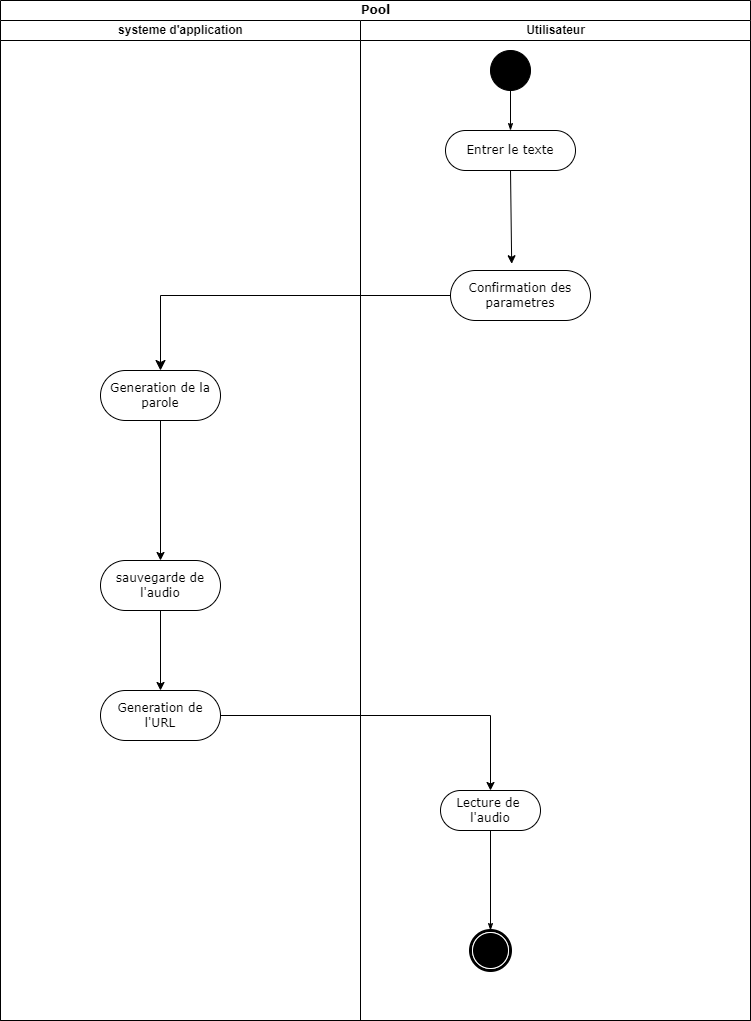


Figure 11:Diagramme d'activité

Ce diagramme présente de manière séquentielle les différentes activités impliquées dans l'utilisation de l'application de texte à la parole.

* Fourniture du texte à convertir.
* Modification des paramètres.
* Transformation du texte en parole.
* Génération de la parole synthétisée.
* Génération de l’audio et son URL.
* Lecture de la sortie audio.

Le processus se termine une fois que toutes les activités sont accomplies et que la sortie audio est lue à l'utilisateur.

# Conclusion

La section de conception détaillée offre une vue complète des éléments structurants et des processus de l'application de synthèse vocale utilisant le modèle Coqui TTS v2. Grâce à l'élaboration de diagrammes de cas d'utilisation, de séquence et d'activité, nous avons clarifié les interactions entre les différents composants et acteurs du système.

En somme, cette conception détaillée pose les bases solides nécessaires pour la mise en œuvre effective de l'application, assurant une intégration fluide des fonctionnalités et une expérience utilisateur optimale. Elle guide le développement en fournissant des repères précis sur les interactions, les processus et la structure globale de l'application.

# **Chapitre IV : Mise en Œuvre**

# Introduction

Dans ce chapitre, nous décrivons le processus de mise en œuvre de notre solution de synthèse vocale, en détaillant l'architecture technique, l'environnement de travail, et les étapes d'implémentation. Ce chapitre vise à fournir une compréhension complète des aspects techniques et pratiques qui ont guidé la réalisation de ce projet.

## 1. Environnement de travail

Dans cette section, nous décrivons en détail les aspects technologiques, matériels, et les outils de développement utilisés pour le projet. Nous abordons également la gestion de versions et les pratiques de collaboration mises en place pour garantir un développement efficace et coordonné.

### 1.1 Choix technologique

Une image contenant Police, logo, Graphique, conception

Description générée automatiquement

Figure 13:Logo de python

**Python** est un langage de programmation interprété, polyvalent et de haut niveau, connu pour sa simplicité et sa lisibilité.

* **Syntaxe Claire et Lisible** : Conçu pour être facile à lire et à écrire, ce qui améliore la productivité des développeurs et réduit le coût de maintenance du code.
* **Bibliothèque Standard Étendue** : Offre une vaste collection de modules pour effectuer des tâches variées, allant du traitement de texte à la manipulation de fichiers, en passant par la communication réseau.
* **Multiparadigme** : Supporte plusieurs styles de programmation, y compris la programmation orientée objet, procédurale et fonctionnelle, offrant une grande flexibilité.
* **Interprété** : Le code Python est exécuté ligne par ligne, ce qui facilite le développement rapide et le débogage.
* **Communauté Active** : Bénéficie d'une large communauté mondiale qui contribue à un écosystème riche de bibliothèques et de Frameworks, tels que Django pour le développement web, NumPy pour l'analyse de données et TensorFlow pour l'apprentissage automatique.
* **Portabilité** : Fonctionne sur divers systèmes d'exploitation, y compris Windows, MacOs et Linux, assurant une grande portabilité des applications développées.

Une image contenant texte, bouteille, boisson gazeuse, noir et blanc

Description générée automatiquement

Figure 14:Logo de flask

Flask est un micro-Framework web léger et flexible pour Python, conçu pour le développement rapide d'applications web.

* **Léger et Extensible** : Fournit uniquement les composants essentiels pour le développement web, avec la possibilité d'ajouter des extensions selon les besoins spécifiques du projet.
* **Facilité d'Utilisation** : Simple à apprendre et à utiliser, ce qui permet un développement rapide et efficace des applications.
* **Flexibilité** : N'impose pas de structure particulière, permettant aux développeurs de concevoir l'architecture de leur application selon leurs préférences.
* **Support RESTful** : Idéal pour créer des API RESTful, grâce à sa gestion intuitive des routes et des requêtes HTTP.
* **Jinja2** : Utilise le moteur de Template Jinja2, permettant la génération dynamique de pages HTML.
* **WSGI** : Basé sur le standard WSGI (Web Server Gateway Interface), assurant une compatibilité avec différents serveurs web et middlewares.
* **Communauté Active** : Dispose d'une large communauté et d'une riche collection d'extensions pour ajouter des fonctionnalités telles que l'authentification, la gestion de bases de données, et bien plus.
* **Débogage Intégré** : Offre des outils de débogage et un rechargement automatique du serveur, facilitant le développement et le test des applications.

Une image contenant Graphique, Police, graphisme, conception

Description générée automatiquement

Figure 15:Logo de CoquiTTS

Coqui TTS est une bibliothèque open-source de synthèse vocale basée sur des modèles d'apprentissage profond, permettant de convertir du texte en parole avec une qualité élevée.

* **Open-Source** : Gratuit et accessible, permettant aux développeurs de contribuer et de modifier le code source selon leurs besoins.
* **Qualité de Synthèse** : Utilise des modèles avancés d'apprentissage profond pour produire une voix synthétique naturelle et fluide.
* **Modèles Pré-entraîner** : Fournit des modèles pré-entraîner pour une variété de langues et de voix, facilitant l'intégration rapide dans les projets.
* **Personnalisation** : Permet l'entraînement de nouveaux modèles avec des données spécifiques, offrant la possibilité de créer des voix personnalisées.
* **Support Multilingue** : Prend en charge plusieurs langues, rendant la bibliothèque utile pour des applications globales.
* **Documentation Complète** : Bien documenté avec des guides et des exemples, facilitant l'utilisation et l'intégration.
* **Communauté Active** : Dispose d'une communauté dynamique qui offre du support et développe de nouvelles fonctionnalités et améliorations.
* **Compatibilité** : Facilement intégrable avec d'autres Frameworks et outils de traitement de la parole et du langage naturel.

### 1.2 Environnement matériel

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Catégorie** | **Serveur de Production** | **Serveur de Préproduction** | **Poste de Travail** | **Périphériques** | **Réseaux et Connectivité** |
| **Type de machine** | Serveur dédié | Serveur dédié | Alienware  M15 R7 | Moniteurs 27" Full HD, résolution 1920x1080 | Réseau local avec routeurs Cisco et commutateurs Netgear |
| **Processeur** | Intel Xeon E5-2670 (2,6 GHz, 8 cœurs, 16 threads) | Intel Xeon E3-1240 (3,4 GHz, 4 cœurs, 8 threads) | Intel Core i7 12th Gen | Claviers mécaniques, souris ergonomiques, imprimante multifonction HP LaserJet Pro | Pare-feu matériel Cisco ASA, VPN pour les connexions à distance |
| **Mémoire RAM** | 64 Go | 32 Go | 32 Go |  |  |
| **Stockage** | 1 To SSD | 100 Go SSD | 1 To SSD |  |  |
| **Réseau** | Connexion gigabit avec une bande passante de 200 Mbps | Connexion gigabit avec une bande passante de 100 Mbps |  |  |  |
| **Hébergement** | Hébergement des applications et services |  |  |  |  |
| **Conteneurisation** | Utilisation de Docker pour la création et la gestion des conteneurs |  |  |  |  |
| **Orchestration** | Kubernetes pour l'orchestration des conteneurs Docker |  |  |  |  |
| **Système d'exploitation** | Ubuntu 20.04 LTS | Ubuntu 20.04 LTS | Windows 11 pro |  |  |

Tableau 3:Environnement matériel (Serveur)

### 1.3 Outil de développement

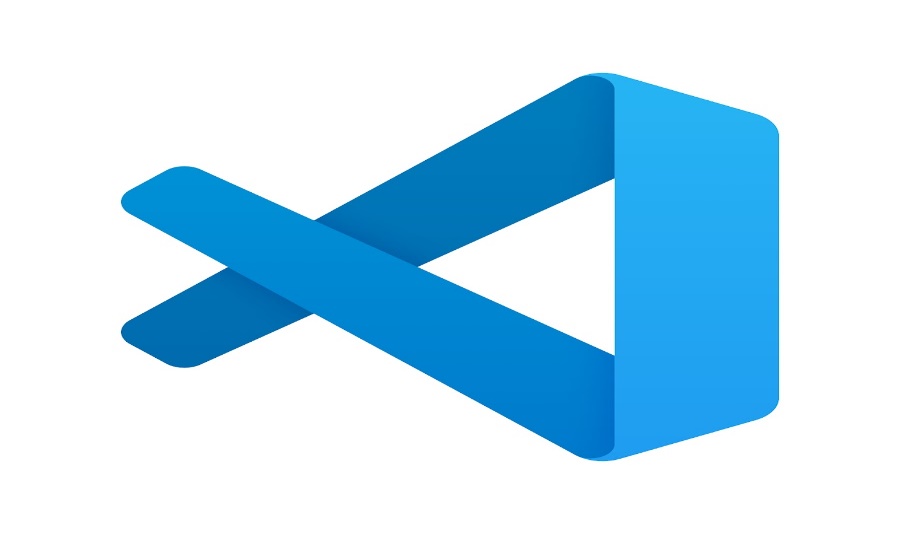


Figure 16:Logo de Visual Studio Code

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

* **Interface Utilisateur Intuitive** : Offre une interface utilisateur moderne et personnalisable, facilitant la navigation et l'édition de code.
* **Support Multilingue** : Prend en charge une large gamme de langages de programmation, y compris Python, JavaScript, TypeScript, C++, Java, et bien d'autres.
* **Extensions Riches** : Dispose d'un vaste écosystème d'extensions qui ajoutent des fonctionnalités supplémentaires, comme le débogage, le contrôle de version, les outils de test, et plus encore.
* **Intégration Git** : Intégration native avec Git et d'autres systèmes de contrôle de version, permettant une gestion facile des versions et des collaborations.
* **Débogage Puissant** : Fonctionnalités de débogage intégrées pour plusieurs langages, avec la possibilité de définir des points d'arrêt, d'inspecter des variables et de suivre l'exécution du code en temps réel.
* **Support des Conteneurs** : Intégration avec Docker et Kubernetes pour le développement et le déploiement d'applications conteneurisées.
* **Editeur de Code Intelligent** : Fonctionnalités comme IntelliSense pour la complétion de code, les refactorisations intelligentes, et la coloration syntaxique, améliorant l'efficacité du codage.
* **Portabilité** : Disponible sur Windows, MacOs et Linux, permettant aux développeurs de travailler sur différents systèmes d'exploitation sans perdre leurs configurations et leurs extensions.



Figure 17:Logo de hugging-face

**Hugging Face** est une entreprise et une plateforme dédiée au traitement du langage naturel (NLP) et à l'intelligence artificielle.

* **Transformers Library** : Bibliothèque de modèles pré-entraînés pour des tâches NLP telles que la classification de texte, la traduction, et la génération de texte.
* **Modèles Pré-entraînés** : Accès à une large collection de modèles d'apprentissage profond, réduisant le besoin de grandes ressources de calcul pour l'entraînement.
* **Communauté Active** : Forte communauté de chercheurs et développeurs contribuant aux projets open source et partageant des ressources.
* **Documentation Complète** : Guides détaillés, tutoriels et exemples pour aider les utilisateurs à démarrer rapidement avec les modèles et bibliothèques.
* **Hugging Face Hub** : Plateforme en ligne pour partager et collaborer sur des modèles NLP, datasets et applications.

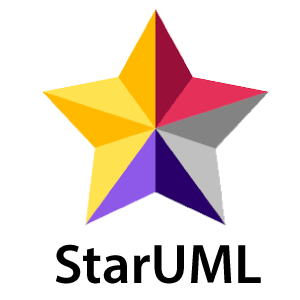


Figure 18:Logo de starUML

**StarUML** est un outil de modélisation UML permettant de créer des diagrammes pour la conception et la documentation de logiciels.

* **Support UML Complet** : Prise en charge de tous les types de diagrammes UML (class, séquence, use case, etc.) pour une modélisation complète.
* **Interface Moderne** : Interface utilisateur moderne et intuitive facilitant la création et la gestion des diagrammes.
* **Extensions et Plugins** : Support pour les extensions et plugins permettant d'ajouter des fonctionnalités supplémentaires et d'intégrer d'autres outils.
* **Exportation et Importation** : Options pour exporter les diagrammes en divers formats et importer des modèles UML depuis d'autres outils.
* **Collaboration** : Fonctionnalités permettant la collaboration en équipe, avec des options de partage et de révision des diagrammes.



Figure 19:Logo de draw.io

**Draw.io** est une application de diagrammes en ligne, utilisée pour créer des organigrammes, des diagrammes UML, des schémas de réseau, et plus encore.

* **Interface Intuitive** : Outils de dessin faciles à utiliser, avec une interface utilisateur intuitive permettant la création rapide de diagrammes.
* **Collaboration en Temps Réel** : Fonctionnalités de collaboration permettant à plusieurs utilisateurs de travailler sur un même diagramme simultanément.
* **Modèles et Formes** : Large bibliothèque de formes et de modèles prédéfinis pour divers types de diagrammes, facilitant le démarrage rapide.
* **Intégrations** : Intégration avec des outils comme Google Drive, OneDrive, et GitHub pour le stockage et le partage de diagrammes.
* **Exportation et Importation** : Options pour exporter les diagrammes dans divers formats (PDF, PNG, SVG) et importer des fichiers depuis d'autres outils de dessin.

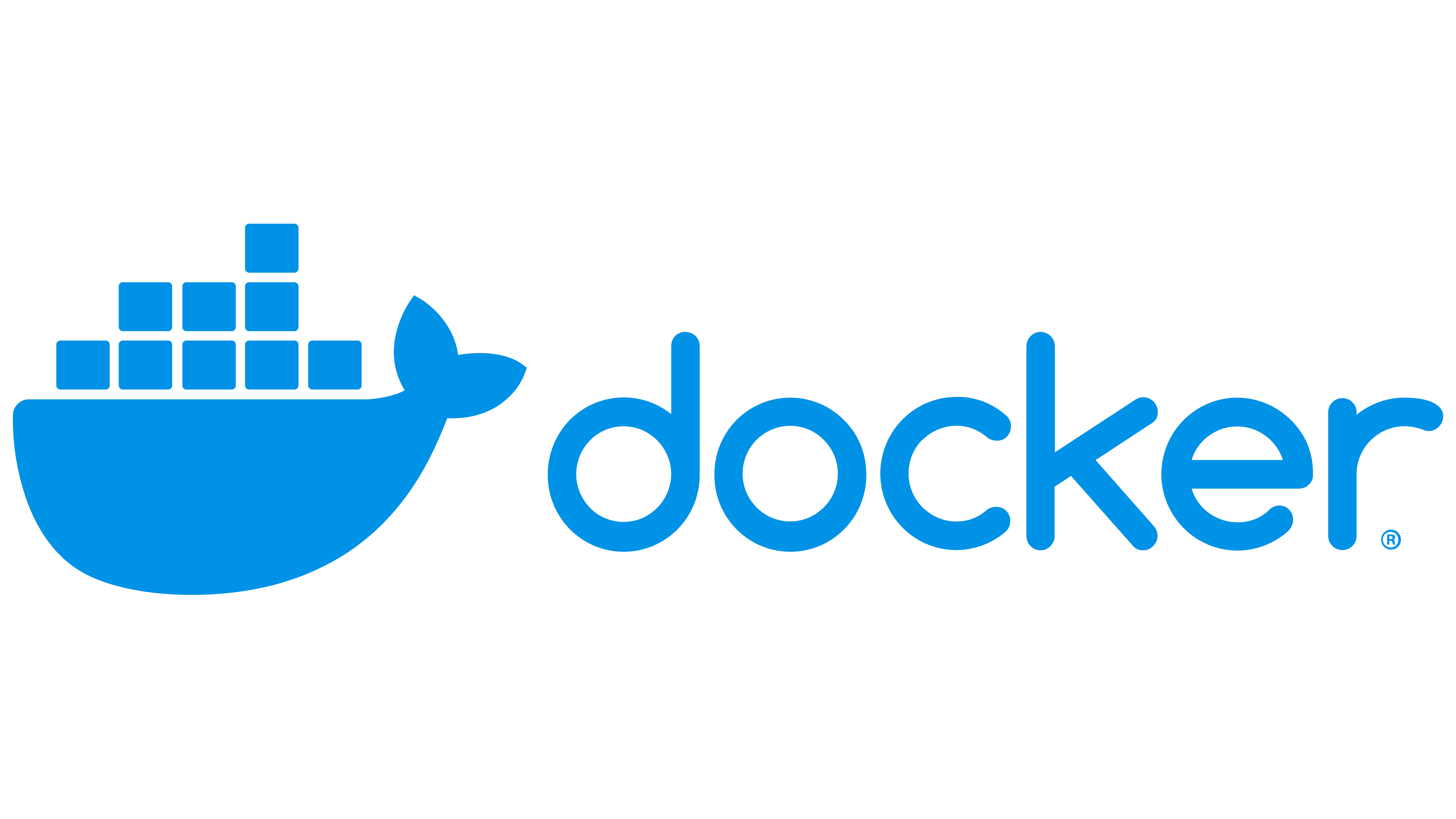


Figure 20:Logo de docker

**Docker** est une plateforme de conteneurisation permettant de développer, expédier et exécuter des applications de manière isolée et reproductible.

* **Isolation des Applications** : Exécution des applications dans des conteneurs isolés, garantissant qu'elles fonctionnent de manière cohérente sur différents environnements.
* **Portabilité** : Conteneurs qui peuvent être exécutés sur n'importe quel système prenant en charge Docker, assurant une portabilité maximale des applications.
* **Gestion des Dépendances** : Conteneurisation des applications avec toutes leurs dépendances, éliminant les problèmes de compatibilité.
* **Déploiement Rapide** : Accélération du cycle de déploiement grâce à des images de conteneurs légères et préconfigurés.
* **Scalabilité** : Facilité de mise à l'échelle des applications conteneurisées pour répondre aux variations de la charge de travail.

Une image contenant Graphique, Police, symbole, conception

Description générée automatiquement

Figure 21:Logo de Git

**Git** est un système de contrôle de version distribué, essentiel pour la gestion de projets de développement logiciel.

* **Versioning Distribué** : Chaque développeur possède une copie complète de l'historique du projet, facilitant le travail hors ligne et la sauvegarde des données.
* **Branches et Fusions** : Gestion avancée des branches permettant le développement parallèle, l'expérimentation et la fusion de différentes branches de code.
* **Traçabilité des Modifications** : Historique complet des modifications du code, avec des commits détaillés et la possibilité de revenir à des versions antérieures.
* **Collaboration** : Facilite la collaboration entre plusieurs développeurs, avec des outils pour résoudre les conflits de fusion et coordonner les contributions.
* **Performance et Efficacité** : Conçu pour être rapide et efficace, même pour les projets de grande taille.

****

Figure 22:Logo de Termius

**Termius** est un client SSH moderne pour gérer et sécuriser les connexions à des serveurs distants.

* **Gestion des Connexions** : Stockage et organisation des connexions SSH, facilitant l'accès rapide aux serveurs fréquemment utilisés.
* **Synchronisation Multi-appareils** : Synchronisation des connexions et des configurations entre différents appareils pour une accessibilité sans interruption.
* **Gestion des Clés SSH** : Génération, stockage et utilisation des clés SSH pour des connexions sécurisées.
* **Interface Utilisateur Conviviale** : Interface intuitive avec options de personnalisation pour améliorer l'expérience utilisateur.
* **Sécurité Avancée** : Chiffrement des connexions et des informations d'authentification pour garantir la sécurité des communications.

Une image contenant capture d’écran, Graphique, Caractère coloré, graphisme

Description générée automatiquement

Figure 23:Logo de Slack

**Slack** est une plateforme de messagerie collaborative conçue pour améliorer la communication et la productivité des équipes.

* **Canaux de Communication** : Organisation des discussions par projet, équipe ou sujet, permettant une gestion claire et structurée des conversations.
* **Partage de Fichiers** : Facilité de partage de documents, images, et autres fichiers directement dans les conversations.
* **Intégrations Tierces** : Intégration avec une multitude d'outils externes tels que Google Drive, Trello, GitHub, et bien plus, pour centraliser les flux de travail.
* **Recherche Avancée** : Fonctionnalités de recherche puissante pour retrouver rapidement des messages, fichiers et discussions antérieures.
* **Appels Audio et Vidéo** : Possibilité de passer des appels audio et vidéo, ainsi que d'organiser des réunions en ligne avec partage d'écran.
* **Bots et Automatisation** : Utilisation de bots et de scripts pour automatiser les tâches répétitives et améliorer l'efficacité.

Une image contenant Police, logo, Graphique, symbole

Description générée automatiquement

Figure 24:Logo de Jira

**Jira** est un outil de gestion de projets et de suivi des problèmes développé par Atlassian, largement utilisé pour la gestion agile des projets logiciels.

* **Suivi des Tâches** : Permet de créer, d'assigner et de suivre les tâches et les tickets de développement, garantissant une gestion efficace des projets.
* **Gestion Agile** : Supporte les méthodologies agiles comme Scrum et Kanban, avec des tableaux de bord interactifs, des sprints, et des backlogs.
* **Rapports et Analyses** : Génère des rapports détaillés sur les progrès, les performances de l'équipe, les délais et les charges de travail, facilitant la prise de décision basée sur les données.
* **Personnalisation** : Offre des options de personnalisation flexibles pour adapter les workflows, les champs de tickets et les permissions aux besoins spécifiques de l'équipe.
* **Intégrations** : S'intègre avec une multitude d'outils tiers tels que Confluence, Bitbucket, GitHub, Slack, et plus encore, pour centraliser les flux de travail.
* **Automatisation** : Dispose de fonctionnalités d'automatisation pour les tâches répétitives, réduisant le travail manuel et augmentant l'efficacité.
* **Gestion des Versions et des Requêtes** : Facilite la gestion des versions logicielles et le suivi des requêtes, des bogues et des améliorations.
* **Sécurité et Permissions** : Offre des contrôles de sécurité avancés et des paramètres de permission granulaires pour protéger les données et gérer les accès.



Figure 25:Logo de Streamlit

**Streamlit** est un Framework open-source qui permet de créer facilement des applications web interactives en utilisant Python.

* **Simplicité d'utilisation :** Streamlit offre une syntaxe simple et intuitive qui permet aux développeurs d'écrire des applications web en quelques lignes de code Python.
* **Rapidité de développement :** Grâce à son approche orientée développeur, Streamlit permet de créer rapidement des prototypes fonctionnels et des démonstrations interactives sans nécessiter de compétences avancées en développement web.
* **Widgets et composants interactifs :** Streamlit facilite l'intégration de widgets interactifs tels que des sliders, des boutons et des champs de texte, permettant aux utilisateurs d'interagir dynamiquement avec les données.
* **Personnalisation :** Il offre des options de personnalisation pour ajuster l'apparence et le comportement des applications selon les besoins spécifiques du projet.
* **Déploiement facile :** Les applications Streamlit peuvent être déployées facilement sur diverses plateformes cloud ou sur des serveurs locaux, facilitant ainsi leur accessibilité.

### 1.4 Gestion de Version

La gestion de versions est un élément crucial de notre processus de développement, nous permettant de suivre les modifications apportées au code source et de garantir la cohérence de notre projet. Nous avons utilisé GitLab comme plateforme principale pour la gestion de versions de notre projet, en utilisant des fichiers de spécification des dépendances pour assurer la reproductibilité de l'environnement de développement.

**GitLab**

GitLab nous a fourni une plateforme complète pour gérer notre code source, collaborer efficacement et suivre les problèmes.

* **Plateforme** : GitLab
* **Dépôts Git** : Nous avons hébergé nos dépôts Git sur GitLab.
* **Merge Requests** : Utilisation des merge requests pour fusionner les modifications dans la branche principale après examen par les pairs.
* **Suivi des Problèmes** : Gestion des tâches, des problèmes et des fonctionnalités directement depuis GitLab.

**Gestion des Dépendances Python**

Nous avons utilisé des fichiers requirements.txt pour spécifier les dépendances Python et garantir la cohérence de l'environnement de développement.

* **Python** : Utilisation d'un fichier requirements.txt pour spécifier les dépendances Python.

**Documentation**

La documentation de notre projet, rédigée en Markdown, a été versionnée dans le même référentiel GitLab que le code source.

* **Format** : Utilisation de Markdown pour la documentation.
* **Stockage** : Documentation versionnée dans le même référentiel GitLab.

## 2. Implémentation de la solution

Le développement de notre application de synthèse vocale s'est basé sur l'utilisation de Coqui TTS v2 et de Flask pour créer une API qui traite le texte et génère un fichier audio. Cette application permet de corriger les erreurs de texte et de remplacer les abréviations avant de produire une synthèse vocale, dont l'URL est ensuite intégrée dans les articles du site web du journal "Le Matin". Ce chapitre détaille l'implémentation de cette solution, y compris les bonnes pratiques suivies, les étapes de développement, et les tests réalisés.

### 2.1 Contraintes

Lors de la conception et du développement de toute application, il est crucial de prendre en compte un certain nombre de contraintes qui peuvent influencer le déroulement du projet et son succès final. Ces contraintes peuvent être de nature technique, ergonomique, ou liées à d'autres aspects du projet. Dans cette section, nous examinerons les principales contraintes rencontrées dans le développement de notre application de synthèse vocale. La compréhension et la gestion de ces contraintes sont essentielles pour assurer la robustesse, la performance, et l'utilisabilité de notre application.

#### 2.1.1 Contraintes techniques

* **Performance du Serveur** :
* Le serveur doit être capable de gérer un grand nombre de requêtes simultanées sans dégradation des performances.
* Nécessité d'avoir un matériel performant, comme un processeur Intel Xeon et une mémoire suffisante pour supporter la charge.
* **Latence** :
* La génération de synthèse vocale doit être rapide pour ne pas frustrer les utilisateurs. Les temps de réponse doivent être optimisés.
* **Sécurité** :
* Mise en place de protocoles de sécurité pour protéger les données transmises, notamment en utilisant HTTPS pour les communications.
* Authentification des utilisateurs pour accéder à l'API afin de prévenir les accès non autorisés.
* **Disponibilité** :
* Le service doit être hautement disponible avec un temps d'arrêt minimal. Cela peut nécessiter l'utilisation de serveurs de secours et de mécanismes de failover.
* **Compatibilité** :
* Assurer la compatibilité avec les différentes plateformes et navigateurs pour que l'audio puisse être lu correctement sur tous les appareils des utilisateurs.
* **Scalabilité** :
* Le système doit pouvoir évoluer en fonction de la demande croissante. Cela peut nécessiter l'utilisation de solutions de mise à l'échelle horizontale et verticale.

#### 2.1.2 Contraintes ergonomiques

* **Simplicité d'Utilisation** :
  + L'interface utilisateur du site web du journal "Le Matin" doit être intuitive et facile à utiliser, même pour les utilisateurs non techniques.
  + Les interactions avec l'API via Postman doivent être simples et bien documentées.
* **Retour Utilisateur** :
  + Fournir des messages clairs et concis pour les actions effectuées par l'utilisateur, notamment en cas d'erreurs ou de succès des opérations de synthèse vocale.
* **Consistance de l'Interface** :
  + Assurer une consistance visuelle et fonctionnelle à travers les différentes parties de l'application pour offrir une expérience utilisateur fluide.
* **Temps de Réponse** :
  + Les utilisateurs doivent recevoir une réponse rapide lors de l'utilisation de l'application, avec des temps de chargement minimaux pour améliorer l'expérience utilisateur.
* **Design Adaptatif (Responsive Design)** :
  + Le site web doit être optimisé pour une utilisation sur différents appareils, y compris les ordinateurs de bureau, les tablettes et les smartphones.

### 2.2 Les bonnes pratiques

Pour assurer un code de qualité, nous avons respecté plusieurs bonnes pratiques de développement. En complément, j'ai fait une auto-formation sur les bonnes pratiques de codage en Python. Voici quelques-unes des règles suivies :

* **Ré-usinage de code (Refactoring)** :

Réorganisation et amélioration du code source sans ajouter de nouvelles fonctionnalités ni corriger des bogues, afin de le rendre plus lisible et maintenable.

* **Écrire des fonctions pures** :

Les fonctions pures dépendent uniquement de leurs paramètres d'entrée et n'ont pas d'effets de bord, facilitant ainsi les tests et la compréhension du code.

* **Différenciation des valeurs et des variables** :

Utilisation de valeurs plutôt que de variables autant que possible, et maintien de l'immutabilité des objets en évitant les setters, favorisant ainsi la création de nouveaux objets pour chaque modification de données.

* **Nommer les éléments correctement** :

Utilisation de noms clairs et descriptifs pour les variables, fonctions et classes afin d'améliorer la lisibilité du code.

* **Éviter les fuites de mémoire** :

Gestion prudente des ressources et nettoyage approprié des objets pour éviter les fuites de mémoire.

* **Petits composants** :

Création de petites fonctions et modules pour faciliter la maintenance et la compréhension du code.

### 2.3 Présentation de l'application

#### 2.3.1 Description générale

L'application TTS (Text-to-Speech) développée pour le groupe Le Matin vise à faciliter la consommation de contenu audio par les utilisateurs en utilisant le modèle Coqui TTS de Hugging Face. Cette application convertit automatiquement du texte en audio de haute qualité, répondant à la demande croissante de contenus audio.

#### 2.3.2 Fonctionnalités principales

* **Conversion de texte en audio :** La fonctionnalité principale de l'application est la conversion de texte en discours audio à l'aide du modèle Coqui TTS. Les utilisateurs soumettent divers contenus textuels à l'application, qui génère ensuite un fichier audio correspondant (voir la figure 26).
* **Correction automatique des fautes et abréviations** : Le texte est prétraité pour corriger les fautes et remplacer les abréviations courantes.

Une image contenant texte, logiciel, Page web, Icône d’ordinateur

Description générée automatiquement

Figure 26:Interface de test pour l’entrée du texte

* **Choix de la voix et clonage :** L'application permet de sélectionner parmi plusieurs voix intégrées et offre des options de clonage vocal, permettant de personnaliser davantage l'expérience utilisateur (voir la figure 27).

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Figure 27:Interface de test montrant le choix de l’audio pour le clonage

* **Génération de l’audio :** Après la conversion, un fichier audio est généré, et les utilisateurs peuvent l'écouter ou le télécharger directement depuis l'interface de test (voir la figure 28).

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Figure 28:Audio généré sur l’interface de test

### 2.4 Utilisation de l'API

L'API joue un rôle central dans le processus de conversion de texte en audio, avec une gestion spécifique du traitement du texte et de la génération du fichier audio.

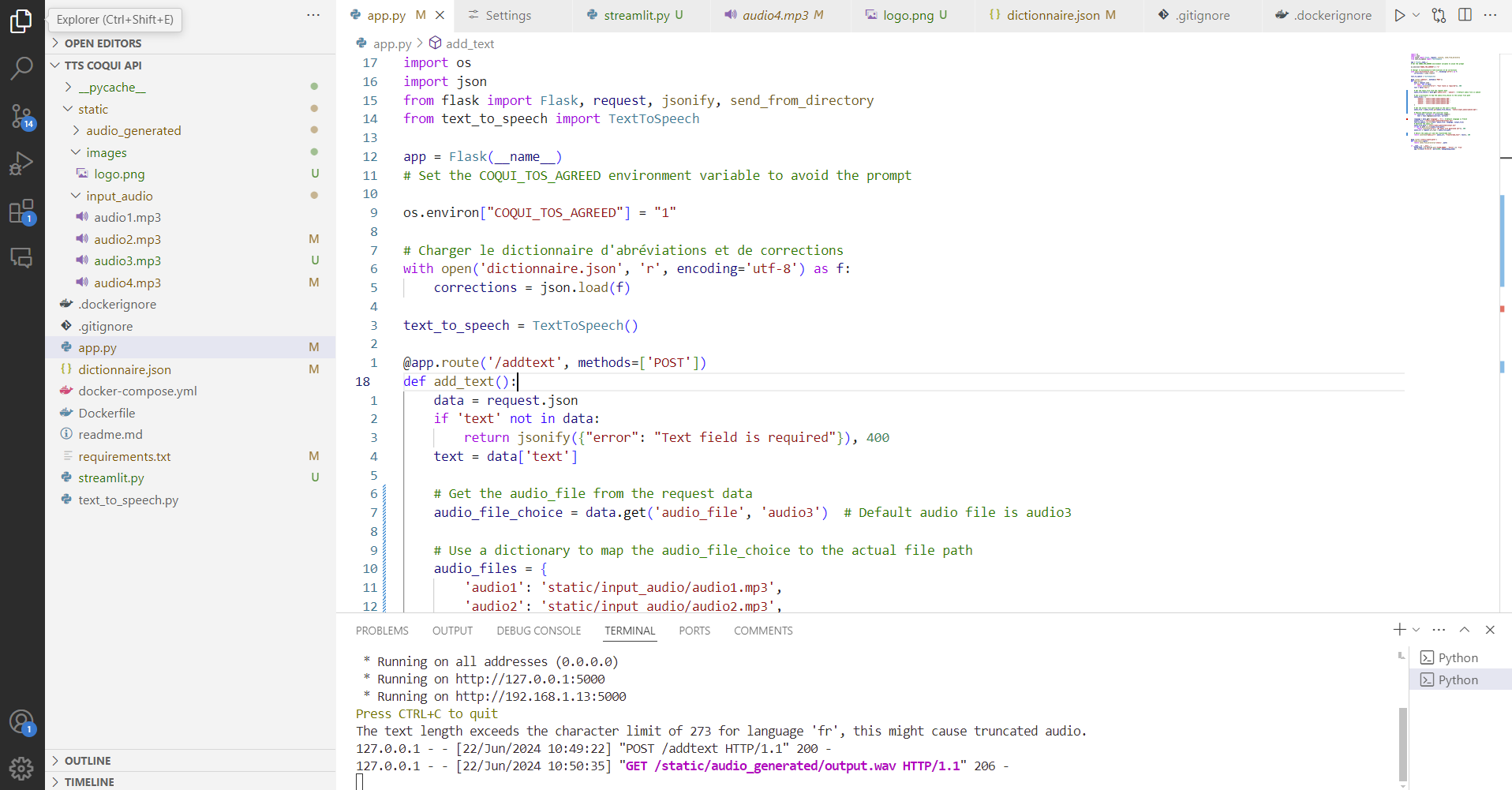


Figure 29:Démarrage de l’API en local

Une image contenant texte, logiciel, Icône d’ordinateur, Page web

Description générée automatiquement

Figure 30:Démarrage de l’API sur le serveur

Cette capture confirme le démarrage de l’API sur les ports 127.0.0.1:5000 (local) ainsi que 192.168.18.121:5000 (Serveur)

#### Fonctionnement détaillé

Lorsqu'un article est soumis à l'API pour la conversion en audio, voici les étapes clés :

* **Pré-traitement du texte :** Le texte de l'article est envoyé à l'API.

**Correction automatique des fautes :** Utilisation d'un dictionnaire dédié pour corriger les erreurs typographiques.

**Remplacement des abréviations :** Transformation des abréviations courantes en leur forme complète pour une meilleure compréhension vocale.

* **Conversion en audio :** Une fois le texte traité, l'API utilise le modèle Coqui TTS pour générer un fichier audio correspondant.
* **Génération du lien :** À la fin du processus, l'API renvoie un lien vers le fichier audio généré.

Une image contenant texte, nombre, Police, logiciel

Description générée automatiquement

Figure 31:L’envoie du texte en local

Capture d'écran montrant l'envoi d'une requête POST via Postman en local (port 5000) vers <http://localhost:5000/addtext>.

Une image contenant texte, logiciel, nombre, Police

Description générée automatiquement

Figure 32:L’envoie du texte cote serveur

Capture d'écran montrant l'envoi d'une requête POST via Postman en local (port 5000) vers <http://192.168.18.121:5000/addtext>.

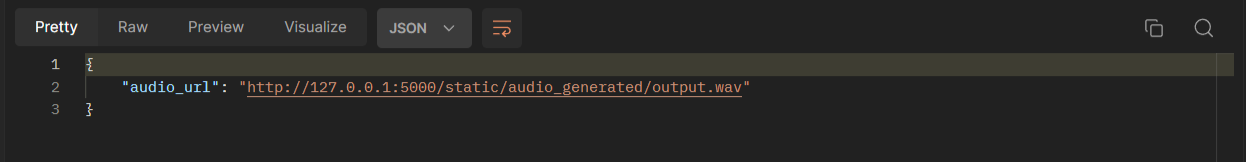


Figure 33:reponse sous forme de lien de l’audio généré(local)

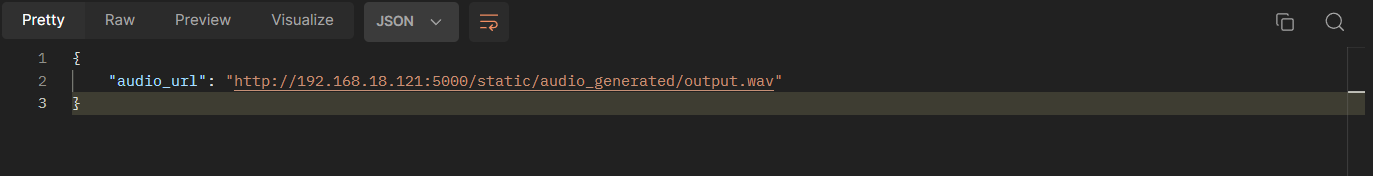


Figure 34:reponse sous forme de lien de l’audio généré (Serveur)

Les deux captures d'écran illustrent la réponse de l'API après avoir converti avec succès le texte en synthèse vocale. La première capture montre le résultat de la transformation réalisée localement, tandis que la seconde capture montre le résultat obtenu après traitement sur le serveur.

# Conclusion

En décrivant ainsi le fonctionnement de l'API dans votre application, vous assurez une compréhension claire du flux de travail technique impliqué dans la conversion de texte en audio. Bien que vous n'ayez pas d'exemple d'utilisation spécifique à montrer, vous pouvez expliquer comment ce processus serait intégré dans le flux de publication d'articles sur le site web du journal Le Matin après la validation des tests nécessaires. Assurez-vous de structurer votre présentation de manière à fournir une vue d'ensemble complète et détaillée de votre application de synthèse vocale, en mettant en avant ses bénéfices et ses fonctionnalités principales.

# **Conclusion Générale et Perspectives**

Ce projet de développement d'une application de synthèse vocale a marqué une avancée significative dans l'évolution technologique du Groupe Le Matin, visant à rendre ses contenus numériques plus accessibles et à enrichir l'expérience utilisateur. À travers ce rapport, nous avons exploré en détail chaque étape de ce parcours, depuis la conceptualisation initiale jusqu'à la réalisation technique avancée.

Nous avons atteint nos objectifs initiaux en répondant à la demande croissante pour des contenus accessibles grâce à une technologie de synthèse vocale moderne et performante. Notre application robuste convertit efficacement le texte en audio, en intégrant le modèle Coqui TTS reconnu pour ses performances exceptionnelles.

Notre approche a été guidée par une étude approfondie des techniques Text-to-Speech, nous orientant vers des méthodes modernes basées sur l'apprentissage profond. Cela nous a permis d'offrir une qualité audio supérieure et une personnalisation accrue grâce au clonage de voix, surpassant ainsi les méthodes traditionnelles pour répondre aux besoins spécifiques de notre contexte.

Pendant la mise en œuvre, nous avons surmonté plusieurs défis techniques et ergonomiques avec succès. La gestion rigoureuse des contraintes techniques et l'application de bonnes pratiques de développement ont assuré la stabilité et la performance optimale de notre application. L'intégration sécurisée et efficace de l'API pour la conversion et la distribution des fichiers audio a également été une étape critique dans la validation de notre solution.

Pour l'avenir, nous envisageons d'optimiser continuellement notre application, d'explorer de nouvelles fonctionnalités et de nous adapter aux évolutions technologiques dans le domaine de la synthèse vocale. Nous cherchons également à étendre l'utilisation de cette technologie à d'autres secteurs au sein du Groupe Le Matin, renforçant ainsi notre position en tant que leader de l'accessibilité numérique et de l'innovation dans le domaine médiatique.

Sur un plan personnel, ce projet a été enrichissant à bien des égards, renforcé par une collaboration étroite au sein de l'équipe de développement et un soutien précieux de nos mentors. Je suis reconnaissant pour cette opportunité de mettre en pratique mes compétences et de contribuer à un projet qui impacte positivement le paysage numérique d'aujourd'hui.

En conclusion, ce rapport démontre notre engagement envers l'innovation technologique et notre capacité à transformer les défis en opportunités concrètes. Je suis convaincu que notre application de synthèse vocale continuera à servir efficacement les stratégies globales du Groupe Le Matin, en améliorant significativement l'expérience des utilisateurs et en ouvrant de nouvelles perspectives pour l'avenir.

# **Bibliographie**

Une image contenant texte, capture d’écran, Police, graphisme

Description générée automatiquement

**Publisher‏: ‎**Springer; 1st ed. 2023 edition (May 30, 2023)

**Autor:** [Xu Tan](https://www.amazon.com/Xu-Tan/e/B0C6L325VY/ref=aufs_dp_fta_an_dsk)

**Language‏: ‎**English

**Hardcover‏: ‎**226 pages

**ISBN-10‏: ‎**9819908264

**Editorial:**Editions ENI; No.: 5 edition (March 11, 2020)

**Autor:** [Laurent Debrauwer](https://www.amazon.es/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=Laurent+Debrauwer&search-alias=stripbooks), [Fien Van der Heyde](https://www.amazon.es/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=Fien+Van+der+Heyde&search-alias=stripbooks)

**Language :**French

**Soft cover :**341 pages

**ISBN-10 :**2409024084



# **Webographie**

https://welovedevs.com/fr/articles/postman/

https://www.python.org/doc/

https://flask.palletsprojects.com/en/3.0.x/

https://code.visualstudio.com/docs

https://termius.com/about

https://slack.com/about

https://huggingface.co/docs

https://github.com/coqui-ai

https://about.gitlab.com/company/

https://docs.docker.com/

https://en.wikipedia.org/wiki/Jira\_(software)

# **Annexes**

Cette annexe fournit les spécifications détaillées de l'API utilisée pour la conversion du texte en audio. L'API reçoit un texte en entrée et renvoie un URL pointant vers le fichier audio généré.

## Annexe 1 : Code de l'API

**Capture d'écran 1 : Endpoint de soumission de texte**

Ce code montre l'endpoint de l'API qui reçoit le texte soumis par l'utilisateur. L'endpoint prend un texte en entrée via une requête POST.

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Figure 35:Soumission du texte à l’API

**Capture d'écran 2 : Traitement du texte**

Cette partie du code détaille le processus de correction des fautes et de remplacement des abréviations avant la conversion en audio.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 36:Chargeement le dictionnaire d'abréviations et de corrections

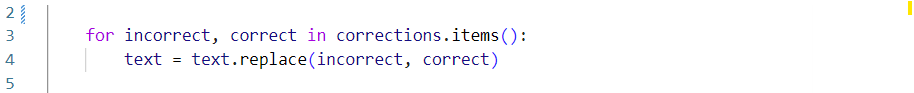


Figure 37:Remplacement des mots incorrectes et les abréviations

**Capture d'écran 3 : Génération de lien audio**

Ce code illustre comment le fichier audio généré est stocké et comment un lien vers ce fichier est renvoyé à l'utilisateur.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 38:La génération et le renvoie de l'URL de l'audio

## Annexe 2 : Code de l'Interface Streamlit

**Capture d'écran 1 : Interface utilisateur**

Description : Cette capture montre le code de l'interface utilisateur développée avec Streamlit, où les utilisateurs peuvent entrer du texte à convertir en audio.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 39:Code de l’interface de Streamlit

**Capture d'écran 2 : Intégration de l'API**

Description : Ce code montre comment l'interface Streamlit communique avec l'API pour envoyer le texte à convertir et recevoir le lien vers le fichier audio généré.



Figure 40:Liaison de Streamlit avec l'API

## Annexes 3 : Code du Modèle de Transformation de Texte en Audio

**Capture d'écran 1 : Chargement du modèle Coqui TTS**

Description : Cette capture montre comment le modèle Coqui TTS est chargé et initialisé pour être utilisé dans le processus de conversion.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Figure 41:Initialisation du modèle de synthèse vocale (Coqui)

**Capture d'écran 2 : Conversion du texte en audio**

* Description : Ce code illustre la méthode principale utilisée pour convertir le texte traité en un fichier audio, en utilisant le modèle de synthèse vocale.

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

Figure 42:La fonction responsable a transformé le texte en audio