

Déploiement de la Solution

Application de Collecte Audio avec Déploiement via Docker S4-DACS-01-1

1 Introduction

Ce document spécifie les exigences pour le développement d'une application web destinée à collecter des enregistrements audio d'utilisateurs lisant des phrases prédéfinies. L'application doit permettre la collecte de données vocales de manière contrôlée et conviviale, tout en garantissant l'intégrité des données et la confidentialité des utilisateurs. La solution doit être conteneurisée à l'aide de Docker et orchestrée via Docker Compose afin d'assurer la cohérence entre différents environnements et de faciliter le déploiement. Le code source et la documentation seront gérés via GitHub, avec une extension facultative pour déployer l'application sur une plateforme cloud.

2 Exigences Fonctionnelles

L'application doit présenter une page d'accueil servant de point d'entrée pour les utilisateurs. Ensuite, les utilisateurs devront fournir des données démographiques se limitant exclusivement à l'âge et au genre, accompagnées d'une confirmation explicite de consentement. Aucune information personnellement identifiable, telle que les noms, ne doit être collectée, garantissant ainsi l'anonymat des utilisateurs.

Une fois ces informations soumises, l'application affichera successivement des phrases que l'utilisateur devra lire à voix haute. L'interface doit offrir des contrôles permettant à l'utilisateur d'enregistrer sa voix, de réenregistrer si nécessaire, d'enregistrer définitivement l'enregistrement actuel, et de passer à la phrase suivante. Le nombre de phrases à présenter doit pouvoir être configuré par l'utilisateur au début de la session d'enregistrement.

Le système doit permettre à l'utilisateur de mettre fin à la session de manière anticipée. Dans ce cas, tous les enregistrements effectués jusqu'à ce moment doivent être sauvegardés de manière fiable, sans perte de données. À la fin de la session ou en cas de sortie anticipée, l'application doit rediriger l'utilisateur vers la page d'accueil, prête pour une session ultérieure.

3 Spécifications Techniques

L'ensemble de l'application doit être conteneurisé à l'aide de Docker afin d'encapsuler l'environnement d'exécution, les dépendances et la configuration, garantissant l'indépendance vis-à-vis de la plateforme et la reproductibilité. Si l'architecture comprend plusieurs services — par exemple, des composants backend et frontend distincts — Docker Compose devra être utilisé pour gérer et orchestrer ces conteneurs de manière efficace. Le contrôle de version sera maintenu via GitHub. Le dépôt devra inclure tout le code source, les fichiers de configuration Docker, ainsi qu'un fichier README complet. Ce fichier devra fournir des instructions claires pour construire, exécuter et tester l'application à la fois dans un environnement local et dans les conteneurs Docker.

4 Extension Optionnelle : Déploiement dans le Cloud

Une amélioration facultative consiste à déployer l'application conteneurisée sur une infrastructure cloud au choix, comme Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, Heroku ou toute autre plateforme équivalente. Ce déploiement doit démontrer la capacité à faire évoluer et gérer l'application dans un environnement similaire à la production. Des instructions de déploiement détaillées, ainsi que toute configuration spécifique à la plateforme, devront être documentées rigoureusement dans le dépôt GitHub.

5 Livrables

Le livrable final est une application web entièrement fonctionnelle répondant aux exigences fonctionnelles et techniques décrites ci-dessus. Le projet sera évalué sur la base des fonctionnalités de l'application, de la qualité du code, de la complétude et de la clarté de la documentation, ainsi que du respect des bonnes pratiques en matière de conteneurisation. La démonstration du déploiement cloud et la documentation associée seront prises en compte pour une reconnaissance supplémentaire.