

# SYSTEME DE SURVEILLANCE INTEGREE POUR EXAMENS

Plateforme de Prévention de la Triche en  
Milieu Éducatif

## Abstract

Notre plateforme utilise smartphones, montres connectées et caméras d'ordinateur pour surveiller les examens. Elle détecte la triche via biométrie, analyse d'images et surveillance audio. L'architecture centralisée assure une surveillance continue, avec alertes en cas de fraude.

# Table de matière

1.	Introduction .....	3
2.	Objectifs du Projet .....	3
3.	: Etat de l'Art.....	4
a.	Introduction .....	4
b.	Quelques exemples de logiciels de surveillance des examens .....	4
i.	Surveillance d'examen en télésurveillance avec Proctorio.....	4
ii.	Avantages .....	5
iii.	Inconvénients .....	5
c.	Surveillance d'examen en télésurveillance avec TestWe .....	5
d.	Surveillance d'examen en télésurveillance avec Metttl .....	6
e.	Surveillance d'examen en télésurveillance avec Proctorio .....	6
f.	Démonstration avec Proctoredu .....	7
g.	Le marché du e-learning .....	12
4.	Équipe de développement.....	13
5.	Technologies Utilisées.....	13
h.	Interface Utilisateur (UI).....	13
i.	Backend (Python) .....	13
j.	Périphériques de Surveillance.....	14
k.	Protocoles de Communication .....	14
l.	Modules d'Analyse et de Détection .....	14
m.	Stockage des Données .....	14
6.	Architecture Système .....	15
7.	Description des Périphéries.....	18
8.	Diagramme de cas d'utilisation de notre système de surveillance .....	19
9.	Protocole gRPC.....	21
10.	Caméra Frontale / Caméra de l'Ordinateur .....	<b>Erreur ! Signet non défini.</b>
11.	Reconnaissance vocale.....	25
11.1.1.	Objectifs .....	<b>Erreur ! Signet non défini.</b>
11.2.	Cadre Théorique et Technologique.....	<b>Erreur ! Signet non défini.</b>
11.2.1.	Cadre Théorique et Technologique .....	<b>Erreur ! Signet non défini.</b>

11.2.2. Utilisation pour la Surveillance.....	<b>Erreur ! Signet non défini.</b>
11.3. Bibliothèques et Modules Utilisés .....	25
11.4. Implémentation .....	26
11.4.1. Configuration et Enregistrement de l'Audio .....	26
11.4.2. Transcription Automatique.....	26
11.4.3. Traitement du Texte.....	26
11.4.4. Enregistrement des Résultats .....	26
11.5. Résultats .....	<b>Erreur ! Signet non défini.</b>
12. Résultats .....	27
13. Références .....	27

# 1. Introduction

Dans le contexte actuel de l'enseignement à distance et de l'évaluation en ligne, le besoin de surveiller de manière efficace et équitable les étudiants lors des examens et des évaluations est devenu primordial. Pour répondre à cette exigence, notre projet propose une solution innovante basée sur la technologie pour surveiller les étudiants pendant leurs sessions d'examen en ligne.

Cette solution repose sur l'utilisation d'une combinaison de périphériques technologiques, notamment les smartphones des étudiants, les montres connectées et les caméras frontales des ordinateurs, afin de capturer et d'analyser divers paramètres tels que les mouvements, les signes vitaux et les images. En utilisant des techniques avancées telles que la détection de triche et l'analyse des données biométriques, notre système vise à assurer l'intégrité des évaluations en ligne tout en garantissant une surveillance efficace et équitable des étudiants.

Cette introduction fournit un aperçu général de notre projet, mettant en évidence son importance dans le contexte de l'enseignement à distance et de l'évaluation en ligne, ainsi que les objectifs principaux que nous cherchons à atteindre à travers cette solution technologique.

## 2. Objectifs du Projet

L'objectif principal de ce projet est de concevoir et développer un système de surveillance innovant pour les examens en ligne, utilisant des technologies telles que les smartphones des étudiants, les montres connectées et les caméras frontales des ordinateurs, afin de garantir l'intégrité des évaluations tout en assurant une surveillance efficace et équitable des étudiants.

# Chapitre 1 : Etat de l'Art et Technologies

## 3. : Etat de l'Art

### a. Introduction

Faire passer des examens traditionnels dans un environnement en ligne sécurisé. Le logiciel de surveillance est un outil avantageux tant pour les établissements d'enseignement que pour les étudiants. Examinons quelques-uns de ses principaux avantages:

- Un environnement académique honnête: le logiciel de contrôle est une solution efficace pour maintenir l'intégrité des cours en ligne. En plus d'aider les établissements d'enseignement à protéger leur réputation, il offre aux étudiants un environnement d'apprentissage équitable.
- Flexibilité: les examens surveillés en ligne peuvent être passés n'importe où et n'importe quand. Ainsi, les enseignants et les étudiants peuvent gérer leur emploi du temps de manière plus efficace.
- Réduction du stress: les étudiants se sentent plus détendus lorsqu'ils passent des examens depuis leur domicile ou d'autres lieux familiers. De plus, dans le cas des examens enregistrés et surveillés par l'IA (Intelligence Artificielle), ils n'ont pas besoin de communiquer avec des examinateurs humains.
- Rentabilité: le contrôle en ligne offre aux établissements d'enseignement un avantage économique significatif. Ils n'ont pas besoin de payer des frais administratifs ou d'embaucher autant d'enseignants et de personnel administratif que pour les évaluations hors ligne.
- Valeur: le contrôle en ligne permet aux établissements d'enseignement d'asseoir leur réputation et, par conséquent, d'accroître la valeur de leurs certificats et diplômes.
- Evolutivité: le contrôle en ligne permet aux établissements d'enseignement d'élargir la couverture des étudiants tout en maintenant un haut niveau d'intégrité académique.

Les étudiants qui vont passer un examen surveillé en ligne sont généralement tenus d'avoir: un ordinateur ou un ordinateur portable qui répond aux exigences du logiciel d'examen et qui est équipé d'une caméra web et d'un microphone, un navigateur approprié et à jour, une connexion internet robuste (avec une vitesse généralement supérieure à 256 kbps), un autre appareil doté d'une caméra (téléphone ou tablette) pour mieux voir le lieu de l'examen. Dans le cas de notre projet, nous ajoutons un outil accroché sur le bras de l'étudiant (smartphone dans notre cas) et qui va relever ses paramètres vitaux.

### b. Quelques exemples de logiciels de surveillance des examens

#### i. Surveillance d'examen en télésurveillance avec Proctorio



Proctorio est un logiciel qui permet d'effectuer la télésurveillance des étudiants lors d'une évaluation en ligne. Il permet de filmer l'activité des étudiants, de restreindre la consultation de ressources Web et de générer des rapports automatisés selon un algorithme d'intelligence artificielle.

Selon le scénario d'utilisation envisagé, Proctorio est combiné à l'outil Questionnaire de monPortail ou à un logiciel de visioconférence dans le but de faciliter la reconnaissance de certains comportements suspects. Ce logiciel de télésurveillance s'avère être un outil dissuasif.

## ii. Avantages

Il y a 3 niveaux de surveillance. Le niveau le plus sévère est de limiter l'ensemble des fonctionnalités du poste des étudiants. À ce propos, 8 comportements sont proposés aux fins d'analyse :

- Changement de fenêtre ;
- Concordance entre les frappes du clavier et les types de réponses du questionnaire ;
- Utilisation de la fonction « copier-coller » à l'aide de la souris ou des raccourcis du clavier ;
- Modification de la dimension de la page du navigateur ;
- Changements sonores ;
- Mouvement de la tête ou du regard ;
- Présence d'une autre personne ;
- Sortie de la fenêtre de l'examen, du cadre de la caméra ou de la pièce.

## iii. Inconvénients

- Toute action suspecte détectée par Proctorio ne garantira pas être un cas de tricherie, car il existe une diversité des contextes dans lesquels un étudiant peut être amené à réaliser une évaluation en ligne ;
- Disponible via un seul navigateur : Google Chrome ;
- Il n'y a pas la possibilité de savoir s'il y a une autre personne dans la pièce pendant l'examen (absence de caméra frontale) ;
- L'activité cardiaque de l'élève n'est pas enregistrée (changement de pouls en cas d'envie de tricherie).

Proctorio génère un rapport de surveillance. Celui-ci présente la liste de tous les étudiants ayant effectués l'examen. Un pourcentage et des indicateurs de suspicion sont associés à chaque personne selon les indicateurs retenus par l'enseignant. En cliquant sur le nom d'un étudiant, on peut consulter les détails de l'analyse du déroulement vidéo de son évaluation en ligne et de ce fait simultanément visionner : l'enregistrement de sa caméra, l'enregistrement de la fenêtre de son navigateur (son questionnaire en ligne), Une trame chronologique permettant d'analyser ses actions et ses comportements pour la durée totale de l'examen.

## c. Surveillance d'examen en télésurveillance avec TestWe



L'offre comprend une plateforme logicielle et un support qui sont modulables. Les responsables administratifs et pédagogiques peuvent utiliser la plateforme pour entrer tout type de sujets et tout type de documents associés, pour identifier les candidats, les surveiller en direct ou en différé, obtenir des corrections automatiques ou personnalisées, accéder à des statistiques et des outils d'aide pédagogiques. TestWe est certifiée ISO 9001 et a été construite sur le fondement des normes RGPD et CNIL, et répond aux impératifs européens de souveraineté numérique.

L'équipe de R&D a mis au point une solution qu'elle est la seule à avoir et qui permet de s'affranchir des problèmes de connexion WiFi. Le candidat charge les épreuves cryptées sur son ordinateur avant le début de l'examen et sa copie est renvoyée automatiquement après la fin de l'examen. Pendant l'examen, il n'a pas besoin de connexion. Dans un contexte où l'égalité des chances entre les candidats est fondamentale et peut faire l'objet de recours juridiques, c'est clé. Dans le contexte, à juste titre contraignant, de la réglementation européenne, ce système anti-

triche prend en compte le consentement aux règles par le candidat, la surveillance humaine et le stockage des preuves.

La plateforme TestWe propose des produits tels que TestWe-classic, ProctorWe et TestWe IA.

- **TestWe-classic:** son système anti-triche propose les fonctionnalités suivantes : l'authentification du candidat par la prise de photo, la surveillance en synchrone ou asynchrone, l'usage de la webcam et des fonctions audio pour la surveillance et le verrouillage des fonctions de l'ordinateur pendant l'examen. Malheureusement, l'on note l'absence d'une caméra frontale qui scanne l'environnement direct de l'étudiant avant et pendant l'examen et son activité cardiaque n'est pas surveillée (celle-ci permettrait de juger d'une éventuelle envie ou d'un éventuel cas de tricherie).
- **ProctorWe:** ce produit quant à lui intègre les fonctions suivantes : l'authentification du candidat par la prise de photo, l'usage de la webcam et des fonctions audio pour la surveillance, le verrouillage des fonctions de l'ordinateur pendant l'examen et La surveillance en synchrone ou asynchrone et des rapports de surveillance générés par une équipe d'experts TestWe. On note ici également l'absence d'information quant à l'environnement dans sa globalité (bien que partiel), mais aussi les changements des paramètres vitaux de l'utilisateur, en l'occurrence l'étudiant.
- **TestWe IA:** outre les fonctionnalités citées plus haut, cette solution renforce la surveillance des examens et aide à détecter toute tentative de triche avec ChatGPT ou autres IA génératives. Il permet d'assurer l'intégrité et l'équité de des examens avec le système intelligent.

#### d. Surveillance d'examen en télésurveillance avec Metttl



Dans le cas de Metttl de Mercer, nous avons les fonctionnalités suivantes:

- **Authentification des étudiants:** technologie d'authentification multifactorielle pour écarter le risque d'usurpation d'identité des étudiants ;
- **Technologie de surveillance:** une technologie complète de surveillance humaine et basée sur l'IA prenant en charge la surveillance en direct, la surveillance automatique et la surveillance à double caméra ;
- **Navigateur d'examens sécurisé Metttl:** la technologie de verrouillage de navigateur sécurisé qui assainit les ordinateurs des étudiants en désactivant des onglets, navigateurs, ports externes supplémentaires etc.

#### e. Surveillance d'examen en télésurveillance avec Proctorio



Les logiciels de surveillance en ligne sont généralement dotés de certaines fonctions de base, telles que:

La surveillance audio et vidéo qui permet à un surveillant à distance ou à une technologie basée sur

l'intelligence artificielle de surveiller le processus d'examen via le microphone, l'écran et la caméra Web des étudiants.

Un verrouillage du navigateur qui empêche les étudiants de rechercher des informations sur Internet.

Le contrôle d'identité permet de vérifier l'identité des étudiants en comparant le scan de leurs documents d'authentification et l'image de la webcam.

Outre ces fonctionnalités, les solutions de contrôle à distance aident les établissements d'enseignement à éliminer la tricherie en intégrant des fonctions de contrôle avancées, telles que:

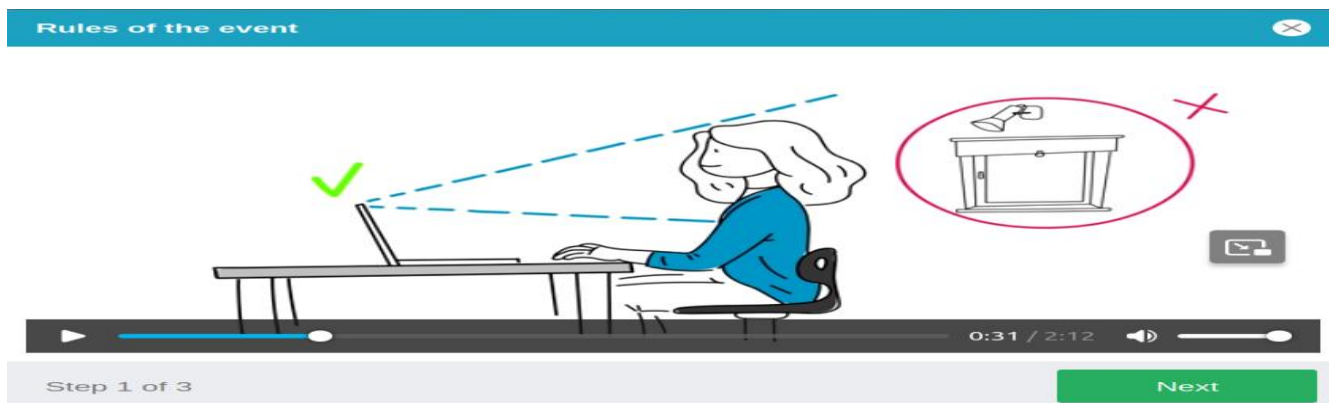
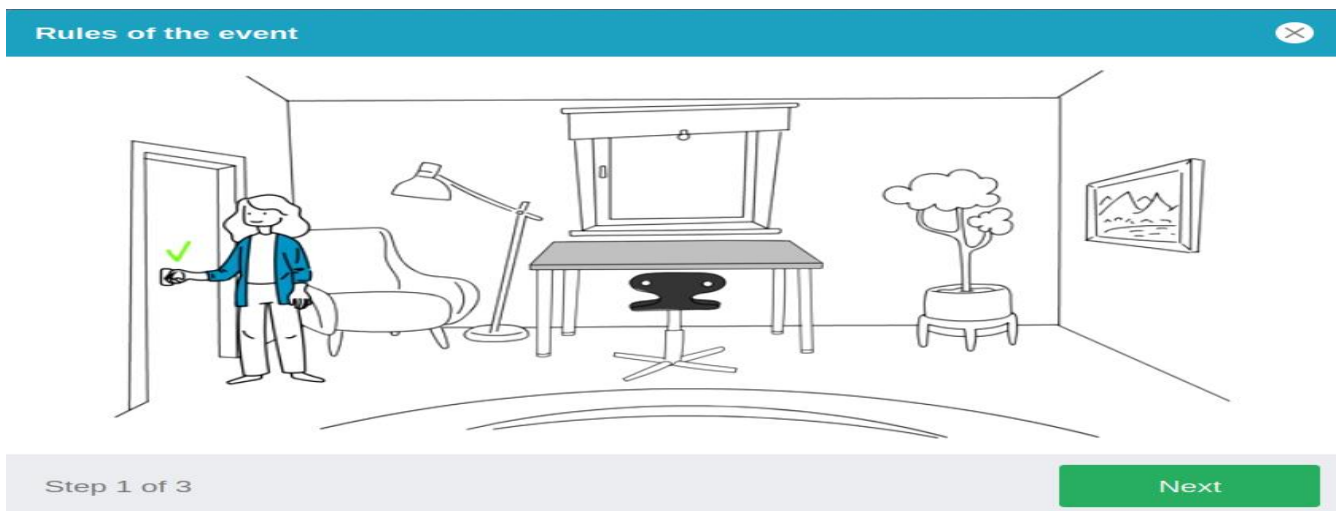
- La détection des visages ;
- La détection du bruit ;
- La prévention de l'utilisation d'appareils mobiles ;
- La protection du contenu.

Les options de livraison les plus appropriées sont les suivantes :

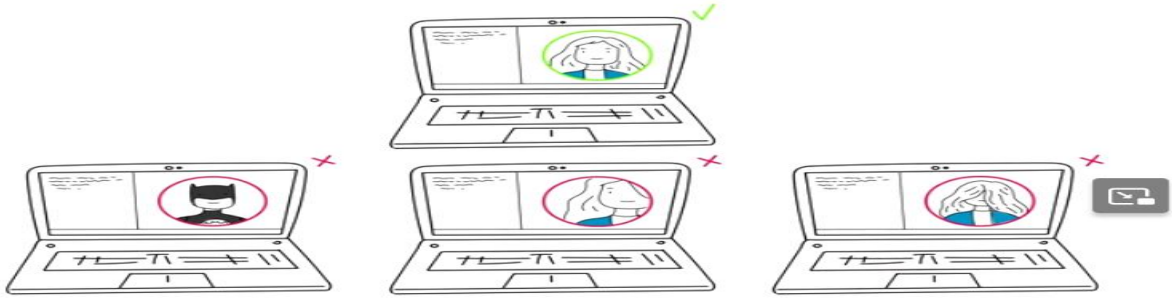
Cloud : Le système fonctionne dans le nuage avec un accès via l'Internet. La mise à jour, la maintenance et toute la complexité technique sont cachées, et l'on ne reçoit que le service requis.

Sur site : Le système est déployé sur votre serveur au sein de votre infrastructure. Il permet un contrôle total des données et peut fonctionner au sein d'un réseau local sans accès à l'internet.

## f. Démonstration avec ProctoredU







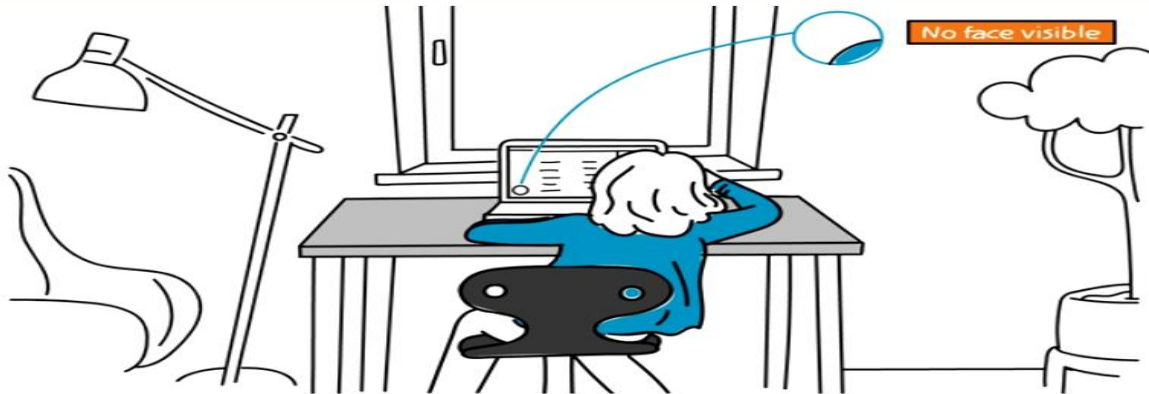
Step 1 of 3

Next



Step 1 of 3

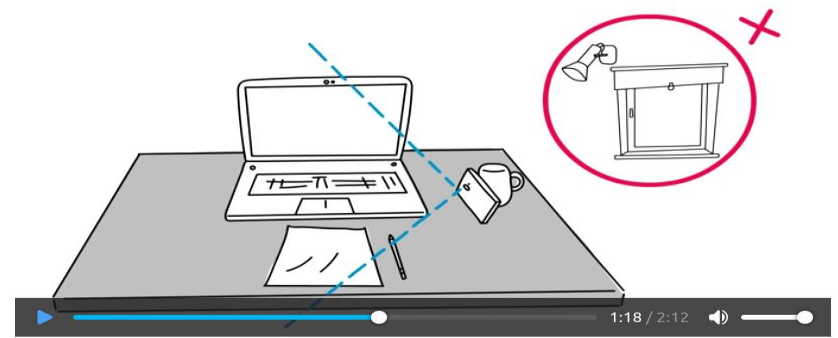
Next



Step 1 of 3

Next

Rules of the event



Step 1 of 3 Next

Equipment check

Please wait while the system checks your computer and the network so that possible technical issues do not interfere with the exam.

- Browser check
- Webcam check
- Microphone check
- Screen check
- Network check
- WebRTC check

Step 2 of 3 Next

Rules of the event

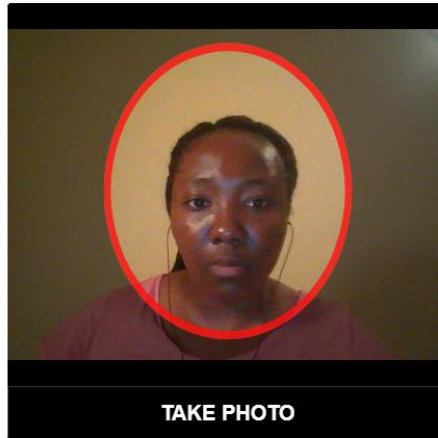


Step 1 of 3 Next

### Taking a face photo



Take a photo with your face fitting into the oval frame on the screen. Make sure that there is enough light in the room. If the photo is not verified, please take a new photo.



Step 3 of 3

Next

### Connection to the event



Everything is ready, now you can start the event. If the event could not start, try again later or select another event.

Full name  
student\_1716925575092

Event title  
Autoproctor — How It Works Demo and  
Equipment Check

Start

demo.proctoring.app/#/report?id=665634881d99e6e4827b285a

SURVEILLANCE 4.12.0-59dd5f1

Autoproctor – How It Works Demo and Equipment Check student\_1716925575092 PDF Taux Discussion

student\_1716925575092

28 Mai 21:52	28 Mai 21:53	28 Mai 21:54	28 Mai 21:55	28 Mai 21:56	28 Mai 21:57	28 Mai 21:58
0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45
70	71	75	90	83	48	70

Lecture x 1.00 Toutes les métriques

10 6 b1 b2 b3 c1 c2 c3

SURVEILLANCE 4.12.0-59dd5f1

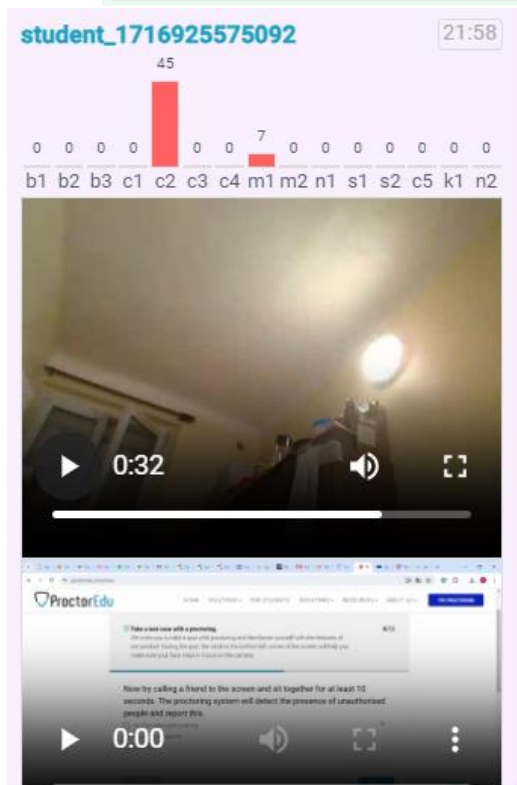
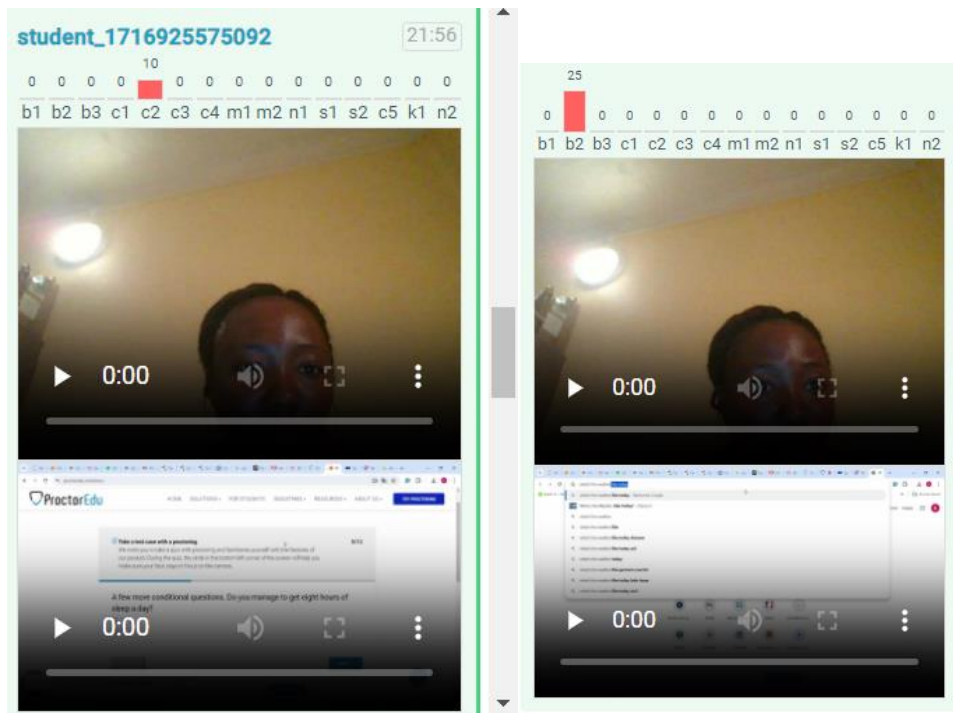
Autoproctor – How It Works Demo and Equipment Check student\_1716925575092 PDF Taux Discussion

student\_1716925575092

28 Mai 21:52	28 Mai 21:53	28 Mai 21:54	28 Mai 21:55	28 Mai 21:56	28 Mai 21:57	28 Mai 21:58
0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45	0 15 30 45
70	71	75	90	83	48	70

Lecture x 1.00 Toutes les métriques

10 6 b1 b2 b3 c1 c2 c3



Toutes les solutions citées plus haut ne permettent pas d'automatiser le jugement d'un éventuel cas de tricherie afin de réduire l'effort humain, ce qui est l'un des objectifs de notre projet.

## g. Le marché du e-learning

Depuis les années 2000, le secteur du e-learning (apprentissage digital) a connu une croissance fulgurante. La part de marché du e-learning dans le monde atteint aujourd'hui plus de 190 milliards de dollars (selon l'étude statistique de "l'entrepreneur en vous" datant de fin

décembre 2022) avec une croissance exponentielle en Asie, le nombre d'apprenants indiens ayant sextuplé depuis 2016. La pandémie Covid-19 a bouleversé les modalités d'enseignement et a ouvert la voie au distanciel à l'échelle mondiale. En effet, les prévisions de l'apprentissage en ligne pour les années à venir laissent entrevoir une croissance impressionnante et des tendances clés avec le développement de la réalité virtuelle et de l'IA.

## 4. Équipe de développement

L'équipe de développement du projet est constituée de professionnels qualifiés et expérimentés, dirigés par un architecte système. Cette équipe multidisciplinaire comprend des développeurs frontend, des ingénieurs logiciels, des analystes en cybersécurité et des gestionnaires de projet. Leur collaboration vise à garantir le succès du projet en mettant l'accent sur l'innovation, la qualité et la satisfaction du client.

Mohammad Raheel	Architecte système
Anissa BEN BOUDAUD	Développeur frontend
Nassim Laghoub	Ingénieur logiciel
FOMO DAFANG Rykiel Galina	Analyste en cybersécurité
Aïcha El Jili	Gestionnaire de projet

## 5. Technologies Utilisées

### h. Interface Utilisateur (UI)

L'interface utilisateur à travers laquelle les utilisateurs interagissent avec le système.

Fonctionnalités :

- Affichage des informations de surveillance.
- Activation / désactivation de la surveillance.
- Réception des alertes de triche.

### i. Backend (Python)

Le cœur du système qui gère la logique métier, les communications et le traitement des données.

Fonctionnalités :

- Gestion des connexions entre les périphériques.
- Traitement des données biométriques et des images.
- Analyse des données pour détecter la triche.
- Envoi d'alertes en cas de détection de triche.

## j. Périphériques de Surveillance

Les dispositifs utilisés pour surveiller l'étudiant pendant l'examen.

Comprend :

- Smartphone de l'étudiant : pour collecter les données biométriques et audio.
- Caméra frontale et caméra de l'ordinateur : pour capturer les images de l'étudiant et de l'environnement de l'examen.
- Oxymètre : Détection de pouls

## k. Protocoles de Communication

Protocoles utilisés pour la communication entre les différents composants du système.

Comprend :

- GRPC : pour la communication entre l'ordinateur hôte et le smartphone de l'étudiant.
- API standard (RESTful par exemple) : pour les autres communications entre les composants.

## l. Modules d'Analyse et de Détection

Les modules logiciels responsables de l'analyse des données collectées et de la détection de la triche.

Comprend :

- Algorithmes de machine Learning : pour l'analyse des données biométriques et des images.
- Algorithmes de traitement du signal : pour l'analyse des données audio.

## m. Stockage des Données

Le stockage des données collectées pour référence future et analyse.

Comprend :

- Base de données : pour stocker les données biométriques, les images capturées et les alertes de triche.
- Technologie utilisée : PostgreSQL/SQL
- Sécurité des données : mesures de sécurité robustes pour protéger les données sensibles, incluant le chiffrement des données en transit et au repos, et des contrôles d'accès stricts.

# Chapitre 2 : Conception

## 6. Architecture Système

### Microservices principaux :

- Service d'authentification et de gestion des utilisateurs
- Service de reconnaissance faciale
- Service de surveillance vidéo
- Service de surveillance audio
- Service de gestion des sessions
- Service de notification et d'alertes
- Service de gestion des examens
- Service de supervision

### Infrastructure et Communication :

- API Gateway
- Service de gestion de configuration
- Service de découverte
- Message Broker (pour la communication asynchrone)

### Base de données :

- Utilisation de bases de données spécialisées pour chaque service

### 2. Détails de l'Architecture

#### 1. Service d'Authentification et de Gestion des Utilisateurs

- **Responsabilités :**
  - Gérer les utilisateurs (inscription, connexion, déconnexion)
  - Stocker les informations des utilisateurs
- **Technologies possibles :** Node.js, Express, JWT pour les tokens, PostgreSQL ou MongoDB
- **Endpoints :**
  - POST /signup
  - POST /login
  - POST /logout
  - GET /user/:id

#### 2. Service de Reconnaissance Faciale

- **Responsabilités :**
  - Gérer la reconnaissance faciale pour l'authentification
- **Technologies possibles :** Python, Flask, OpenCV, TensorFlow, PostgreSQL ou MongoDB
- **Endpoints :**
  - POST /recognize (envoi l'image pour vérification)

#### 3. Service de Surveillance Vidéo

- **Responsabilités :**
  - Analyser les flux vidéo pour détecter des comportements de triche
- **Technologies possibles :** Python, OpenCV, TensorFlow, PyTorch
- **Endpoints :**
  - POST /analyze-video (reçoit et analyse un segment vidéo)



- **Communication interne :**  
Utilisation de RabbitMQ ou Kafka pour les messages asynchrones

#### 4. Service de Surveillance Audio

- **Responsabilités :**  
Analyser les flux audio pour détecter des sons ou des voix non autorisées
- **Technologies possibles :** Python, PyAudio, SpeechRecognition, TensorFlow, Kaldi
- **Endpoints :**  
POST /analyze-audio (reçoit et analyse un segment audio)
- **Communication interne :**  
Utilisation de RabbitMQ ou Kafka pour les messages asynchrones

#### 5. Service de Gestion des Sessions

- **Responsabilités :**  
Gérer les sessions d'examen des utilisateurs  
Suivre l'état de la surveillance (caméra, micro, etc.)
- **Technologies possibles :** Node.js, Express, PostgreSQL ou MongoDB
- **Endpoints :**  
POST /start-session  
POST /end-session  
GET /session/:id

#### 6. Service de Notification et d'Alertes

- **Responsabilités :**  
Gérer l'envoi des notifications et des alertes en cas de détection de triche
- **Technologies possibles :** Node.js, Express, RabbitMQ ou Kafka
- **Endpoints :**  
POST /notify (envoie une notification ou une alerte)
- **Communication interne :**  
Consomme les messages des services de surveillance

#### 7. Service de Gestion des Examens

- **Responsabilités :**  
Gérer les examens (création, gestion des questions, résultats)
- **Technologies possibles :** Node.js, Express, PostgreSQL ou MongoDB
- **Endpoints :**  
POST /create-exam  
GET /exam/:id  
POST /submit-exam

#### 8. Service de Supervision

- **Responsabilités :**  
Permettre aux superviseurs de visualiser l'état de surveillance de chaque élève en temps réel
- **Technologies possibles :** React.js, Node.js, WebSocket
- **Endpoints :**  
GET /supervision

### 3. Infrastructure et Communication

#### API Gateway

- **Responsabilités :**  
Centraliser les points d'entrée pour les microservices  
Gérer l'authentification, la journalisation et le routage
- **Technologies possibles :** Kong, NGINX, Zuul

#### **Service de Gestion de Configuration**

- **Responsabilités :**  
Centraliser la gestion des configurations des microservices
- **Technologies possibles :** Spring Cloud Config, Consul

#### **Service de Découverte**

- **Responsabilités :**  
Permettre aux microservices de découvrir dynamiquement les autres services
- **Technologies possibles :** Eureka, Consul, Zookeeper

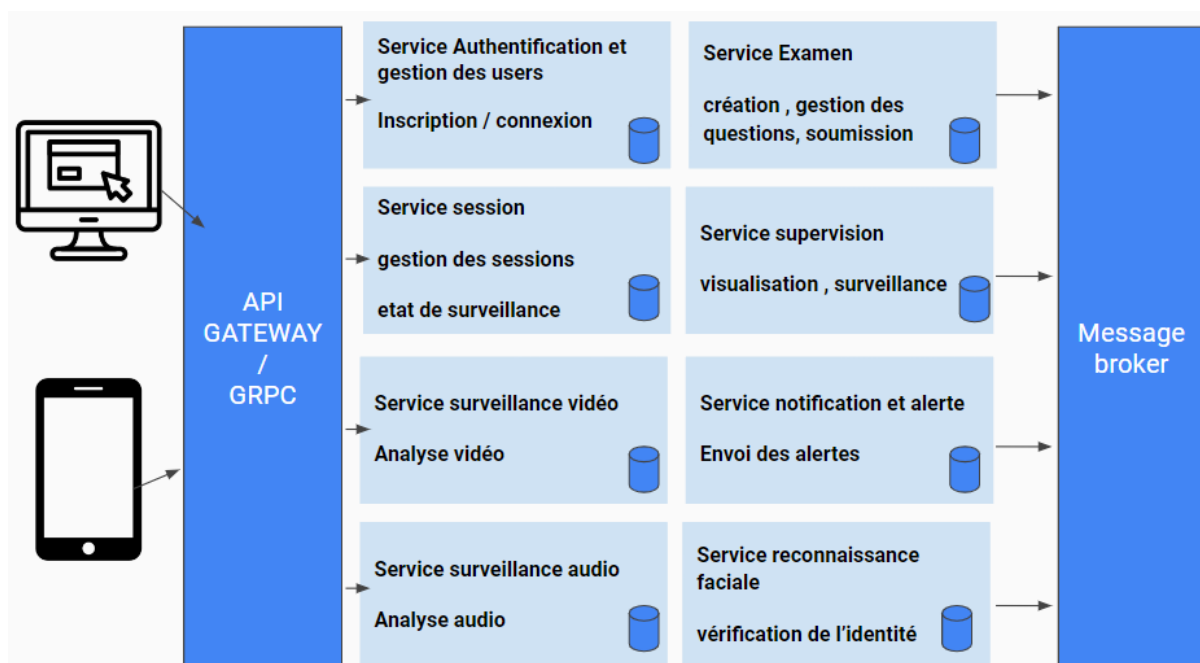
#### **Message Broker**

- **Responsabilités :**  
Faciliter la communication asynchrone entre les microservices
- **Technologies possibles :** RabbitMQ, Kafka

### **4. Base de Données**

#### **Utilisation de Bases de Données Spécialisées**

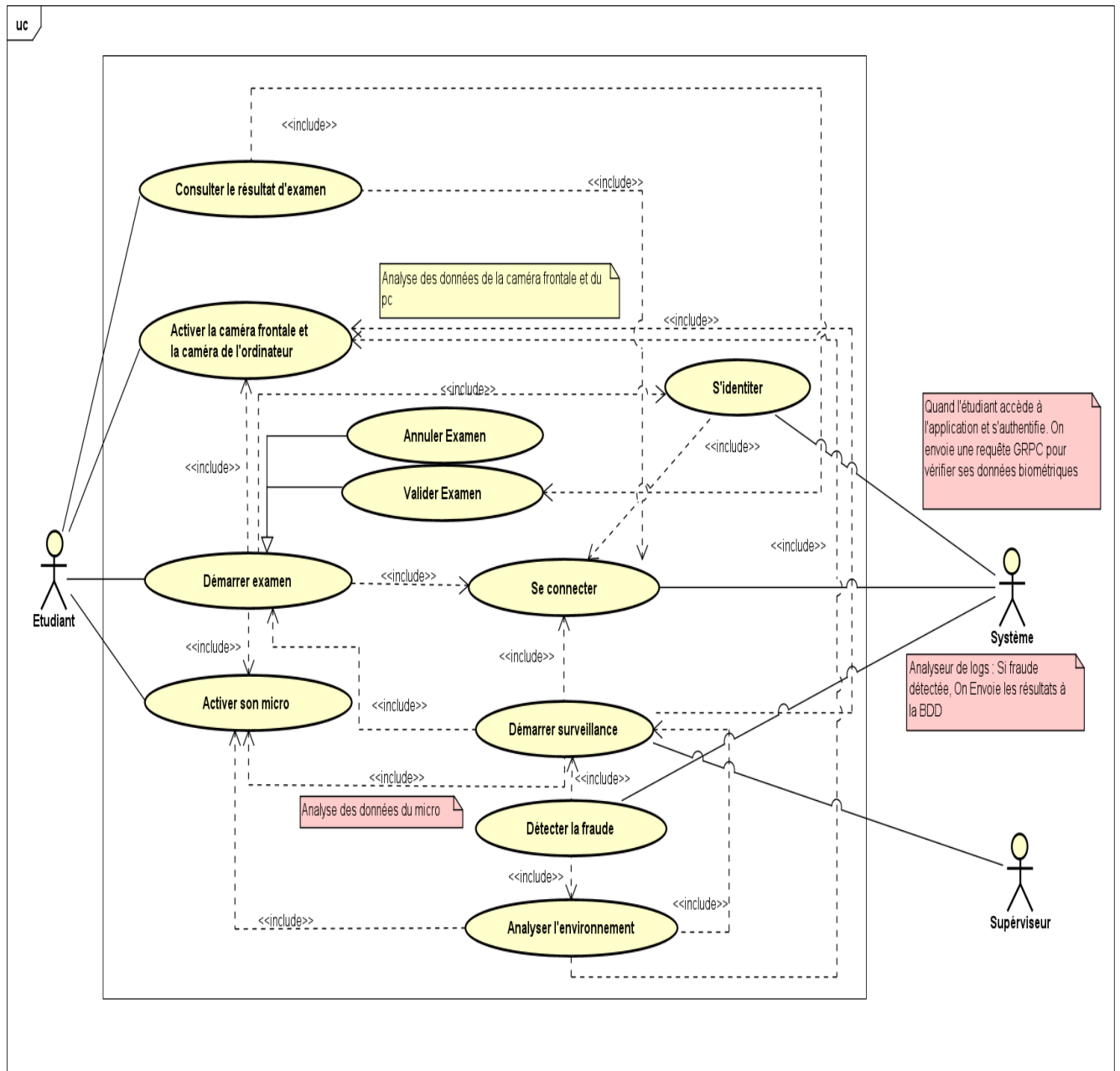
- **Service d'Authentification et de Gestion des Utilisateurs :** PostgreSQL ou MongoDB
- **Service de Reconnaissance Faciale :** PostgreSQL ou MongoDB
- **Service de Surveillance Vidéo :** Stockage de vidéos dans un système de fichiers distribué (par exemple, Amazon S3) et métadonnées dans PostgreSQL ou MongoDB
- **Service de Surveillance Audio :** Stockage d'audio dans un système de fichiers distribué et métadonnées dans PostgreSQL ou MongoDB
- **Service de Gestion des Sessions :** PostgreSQL ou MongoDB
- **Service de Notification et d'Alertes :** PostgreSQL ou MongoDB
- **Service de Gestion des Examens :** PostgreSQL ou MongoDB



## 7. Description des Périphéries

Périphériques	Technologies suivantes associées
Ordinateur hôte	GRPC pour la communication avec le smartphone de l'étudiant, peut-être des bibliothèques Python telles que OpenCV pour l'analyse d'images, et d'autres bibliothèques de machine Learning pour la détection de la triche.
Smartphone de l'étudiant :	Utilisation de GRPC pour la communication avec l'ordinateur hôte, et des fonctionnalités intégrées du smartphone telles que le microphone pour la surveillance audio.
Oxymetre	Utilisation de l'oxymètre pour la surveillance des pouls de l'étudiant, notre choix s'est porté sur Android Sensor Framework.
Caméra frontale / Caméra de l'ordinateur	Utilisation de technologies de traitement d'image OpenCV en Python ou de vision par ordinateur pour l'analyse des images capturées.

## 8. Diagramme de cas d'utilisation de notre système de surveillance



Les participants: Étudiant, Smartphone de l'étudiant, Caméra frontale / Caméra de l'ordinateur, Ordinateur hôte, Analyseur central des données

Étudiant->PC de l'étudiant :

Démarre la session et se connecte

Smartphone de l'étudiant-> Ordinateur hôte :

Envoie des requêtes GRPC pour les données biométriques (reconnaissance faciale, identifiants ...)

Ordinateur hôte-> Analyseur central des données (BDD des étudiants) :	Requête GRPC pour les données biométriques (reconnaissance faciale, identifiants)
Analyseur central des données (Système)-> Ordinateur hôte :	Identification réussie
Ordinateur hôte -> Smartphone de l'étudiant :	Requête de démarrage de la surveillance continue
Ordinateur hôte -> Caméra frontale du smartphone / Caméra de l'ordinateur hôte et le micro du smartphone :	Capture des images/son dès que l'examen démarre
Caméra frontale du smartphone / Caméra de l'ordinateur hôte et le micro du smartphone -> Ordinateur hôte :	Analyse de l'environnement pendant l'examen
Ordinateur hôte -> Analyseur de logs :	Envoie des données enregistrées s'il y a triche
Analyseur de logs -> ordinateur hôte :	Envoie des données enregistrées s'il y a triche
Ordinateur hôte -> BDD de l'étudiant	Envoie les informations sur la BDD de l'étudiant si une triche est détectée

# Chapitre 3 : Réalisation

## 9. Protocole gRPC

Le travail a consisté à mettre en place l'environnement de développement nécessaire pour la création et l'exécution d'un service gRPC. Pour commencer, j'ai téléchargé et installé Python 3.12, puis j'ai utilisé pip pour installer les bibliothèques gRPC (grpcio et grpcio-tools). En parallèle, j'ai également installé le JDK depuis le site officiel d'Oracle et Android Studio pour le développement sur appareil mobile. Une fois ces installations terminées, j'ai configuré et installé le compilateur protocbuf en ajoutant son chemin d'accès à la variable d'environnement PATH. Bien que l'installation de GIT fût optionnelle, elle s'est avérée utile pour le débogage, en particulier si le plugin grpc\_python\_plugin.exe était manquant. Ensuite, j'ai démarré le serveur gRPC et j'ai programmé les fichiers Python nécessaires, notamment Server.py sur l'ordinateur et Client.py sur le téléphone. L'exécution de ces scripts a automatiquement généré les fichiers mon\_service\_pb2.py et mon\_service\_pb2\_grpc.py, contenant les classes Python requises pour le service gRPC à partir du fichier proto.

Ci-dessous les étapes détaillées :

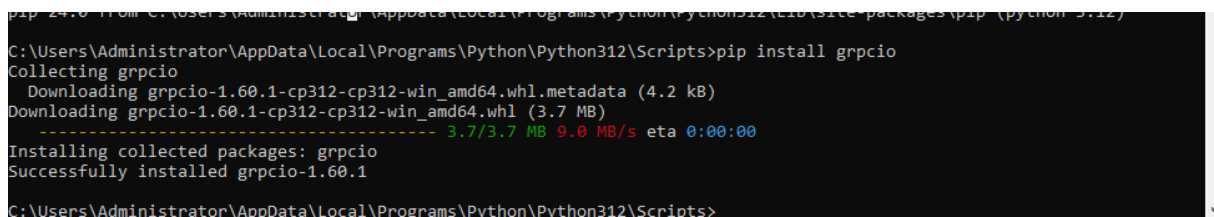
Téléchargement et Installation de Python 3.12

Installation de gRPC :

Ouvrez une fenêtre de commande.

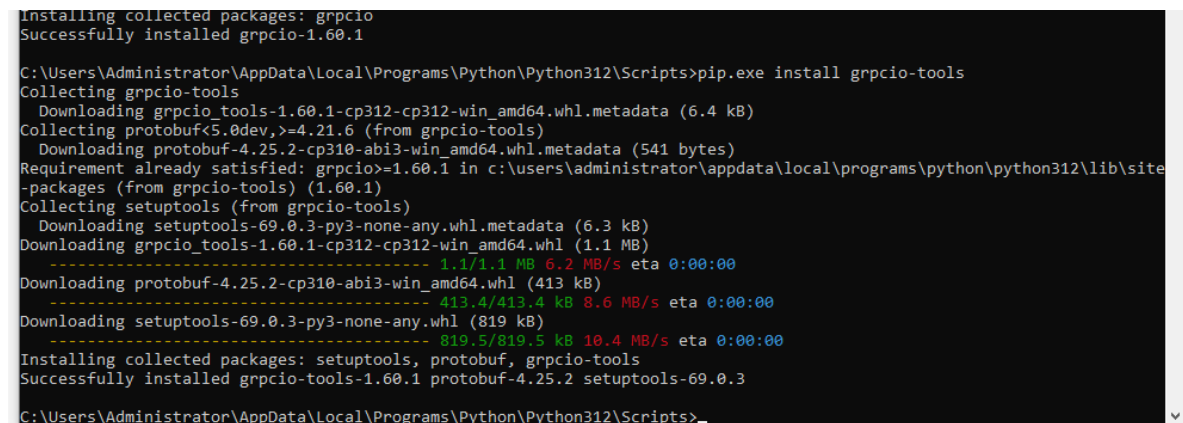
Utilisez pip pour installer les bibliothèques gRPC en exécutant les commandes suivantes :

pip install grpcio



```
C:\Users\Administrator\AppData\Local\Programs\Python\Python312\Scripts>pip install grpcio
Collecting grpcio
  Downloading grpcio-1.60.1-cp312-cp312-win_amd64.whl.metadata (4.2 kB)
  Downloading grpcio-1.60.1-cp312-cp312-win_amd64.whl (3.7 MB)
----- 3.7/3.7 MB 9.0 MB/s eta 0:00:00
Installing collected packages: grpcio
Successfully installed grpcio-1.60.1
C:\Users\Administrator\AppData\Local\Programs\Python\Python312\Scripts>
```

pip install grpcio-tools



```
C:\Users\Administrator\AppData\Local\Programs\Python\Python312\Scripts>pip.exe install grpcio-tools
Collecting grpcio-tools
  Downloading grpcio_tools-1.60.1-cp312-cp312-win_amd64.whl.metadata (6.4 kB)
Collecting protobuf<5.0dev,>=4.21.6 (from grpcio-tools)
  Downloading protobuf-4.25.2-cp310-abi3-win_amd64.whl.metadata (541 bytes)
Requirement already satisfied: grpcio>=1.60.1 in c:\users\administrator\appdata\local\programs\python\python312\lib\site-packages (from grpcio-tools) (1.60.1)
Collecting setuptools (from grpcio-tools)
  Downloading setuptools-69.0.3-py3-none-any.whl.metadata (6.3 kB)
  Downloading grpcio_tools-1.60.1-cp312-cp312-win_amd64.whl (1.1 MB)
----- 1.1/1.1 MB 6.2 MB/s eta 0:00:00
  Downloading protobuf-4.25.2-cp310-abi3-win_amd64.whl (413 kB)
----- 413.4/413.4 kB 8.6 MB/s eta 0:00:00
  Downloading setuptools-69.0.3-py3-none-any.whl (819 kB)
----- 819.5/819.5 kB 10.4 MB/s eta 0:00:00
Installing collected packages: setuptools, protobuf, grpcio-tools
Successfully installed grpcio-tools-1.60.1 protobuf-4.25.2 setuptools-69.0.3
C:\Users\Administrator\AppData\Local\Programs\Python\Python312\Scripts>
```

Installation de JDK sur le site officiel d'Oracle.

Installation d'Android Studiosur le site officiel.

Configuration et Installation du compilateur protocbuf :

Téléchargez le compilateur protocbuf depuis le site officiel.

Ajoutez le chemin d'accès du compilateur protocbuf à la variable d'environnement PATH de votre système.

Installation de GIT (optionnel : Si nécessaire, installez GIT pour le débogage, car le plugin grpc\_python\_plugin.exe peut être manquant.

### Serveur gRPC Démarré

```
Configuration IP de Windows
Yester Carte Ethernet Ethernet :
Déte Statut du média. . . . . : Média déconnecté
      Suffixe DNS propre à la connexion. . . : eu.dior.fashion
Previ Carte réseau sans fil Connexion au réseau local* 9 :
GRPC Statut du média. . . . . : Média déconnecté
      Suffixe DNS propre à la connexion. . . :
Singu Carte réseau sans fil Connexion au réseau local* 10 :
Explic Statut du média. . . . . : Média déconnecté
      Suffixe DNS propre à la connexion. . . :
Comp Carte réseau sans fil Wi-Fi :
Expor Suffixe DNS propre à la connexion. . . :
      Adresse IPv6 de liaison locale. . . . : fe80::f3e4:e44b:46f9:ea67%14
Missio Adresse IPv4. . . . . : 192.168.1.151
      Masque de sous-réseau. . . . . : 255.255.255.0
      Passerelle par défaut. . . . . : 192.168.1.1
Previ Carte Ethernet Connexion réseau Bluetooth :
Cours Statut du média. . . . . : Média déconnecté
      Suffixe DNS propre à la connexion. . . :
Install C:\Users\rmohammad_ext\AppData\Local\Programs\protoc-25.2-win64\bin>python server.py
Rende Server started. Listening on port 50051...
```

### Programmation Python

Server.py (sur l'ordinateur) :

```
server.py - C:\Users\rmohammad_ext\AppData\Local\Programs\protoc-25.2-win64\bin\serv...
File Edit Format Run Options Window Help
import grpc
import mon_service_pb2
import mon_service_pb2_grpc
import concurrent.futures

class MonServiceServicer(mon_service_pb2_grpc.MonServiceServicer):
    def DireBonjour(self, request, context):
        # Implement your RPC method here
        return mon_service_pb2.BonjourResponse(message="Response from server")

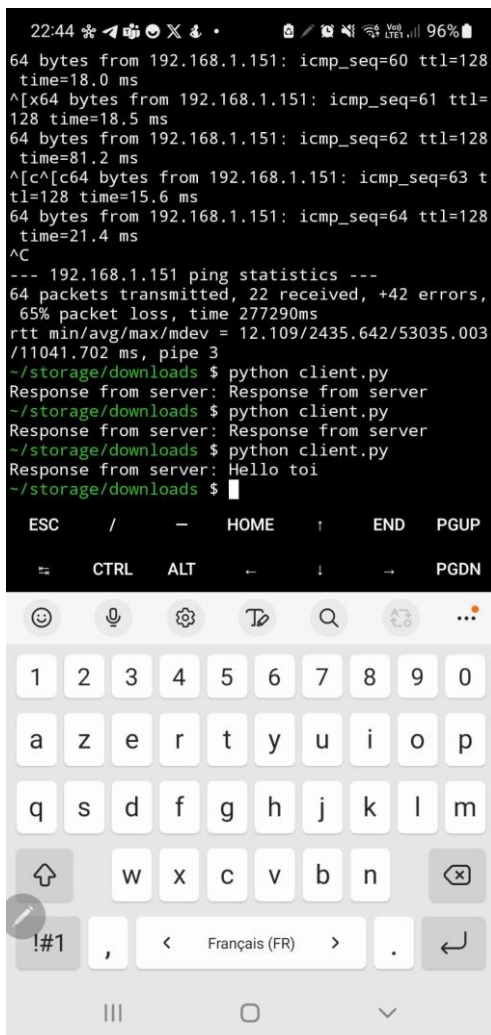
def serve():
    server = grpc.server(concurrent.futures.ThreadPoolExecutor())
    mon_service_pb2_grpc.add_MonServiceServicer_to_server(MonServiceServicer(),
    server.add_insecure_port('192.168.1.151:50051')
    server.start()
    print("Server started. Listening on port 50051...")
    server.wait_for_termination()

if __name__ == '__main__':
    serve()
|
```

Client.py (sur le téléphone :

```
client.py - C:\Users\rmohammad_ext\OneDrive\Sorbonne\IoT projet\Connexion smartphone\client.py (3.12.2)
File Edit Format Run Options Window Help
import grpc
import mon_service_pb2
import mon_service_pb2_grpc
def run():
    with grpc.insecure_channel('192.168.73.37:50051') as channel:
        stub = mon_service_pb2_grpc.MonServiceStub(channel)
        response = stub.YourRpcMethodName(mon_service_pb2.YourRequest(request_data="Request from client"))
        print("Response from server:", response.response_data)
if __name__ == '__main__':
    run()
```





Une fois exécuté, les fichiers suivants seront générés automatiquement :

**mon\_service\_pb2.py\*\***: Fichier contenant les classes Python générées à partir du fichier proto.

```

mon_service_pb2.py - C:\Users\mohammad_ext\AppData\Local\Programs\protoc-25.2-win64\bin\mon_service_pb2.py (3.12.2)
File Edit Format Run Options Window Help
# -*- coding: utf-8 -*-
# Generated by the protocol buffer compiler. DO NOT EDIT!
# source: mon_service.proto
# Protobuf Python Version: 4.25.0
"""Generated protocol buffer code."""
from google.protobuf import descriptor as _descriptor
from google.protobuf import descriptor_pool as _descriptor_pool
from google.protobuf import symbol_database as _symbol_database
from google.protobuf.internal import builder as _builder
# @@protoc_insertion_point(imports)

_sym_db = _symbol_database.Default()

DESCRIPTOR = _descriptor_pool.Default().AddSerializedFile(b'\n\x11mon_service.proto\x12\nmonservice\x1d\n\x0e\x12onjourRequest\x12\x0b\n\x03nom\x18\x01 \x01(\t^\n\x0f\x12onjourResponse\x12\x0f\n\x07mes:

_globals = globals()
_builder.BuildMessageAndEnumDescriptors(DESCRIPTOR, _globals)
_builder.BuildToDescriptorKindMessages(DESCRIPTOR, 'mon_service_pb2', _globals)
if _descriptor._USE_C_DESCRIPTORS == False:
    DESCRIPTOR._options = None
    _globals['_BONJOURREQUEST']._serialized_start=33
    _globals['_BONJOURREQUEST']._serialized_end=62
    _globals['_BONJOURRESPONSE']._serialized_start=64
    _globals['_BONJOURRESPONSE']._serialized_end=98
    _globals['_MONSERVICE']._serialized_start=100
    _globals['_MONSERVICE']._serialized_end=184
# @@protoc_insertion_point(module_scope)

```

**mon\_service\_pb2\_grpc.py** : Fichier contenant les classes Python générées pour le service gRPC à partir du fichier proto.

```

mon_service_pb2_grpc.py - C:\Users\mohammad_ext\AppData\Local\Programs\Python\Python312-64\bin\mon_service_pb2_grpc.py (3.12.2)
File Edit Format Run Options Window Help
# Generated by the gRPC Python protocol compiler plugin. DO NOT EDIT!
"""Client and server classes corresponding to protobuf-defined services."""
import grpc

import mon_service_pb2 as mon_service_pb2

class MonServiceStub(object):
    """Missing associated documentation comment in .proto file."""

    def __init__(self, channel):
        """Constructor.

        Args:
            channel: A grpc.Channel.
        """
        self.DireBonjour = channel.unary_unary(
            '/monservice.MonService/DireBonjour',
            request_serializer=mon_service_pb2.BonjourRequest.SerializeToString,
            response_deserializer=mon_service_pb2.BonjourResponse.FromString,
        )

class MonServiceServicer(object):
    """Missing associated documentation comment in .proto file."""

    def DireBonjour(self, request, context):
        """Missing associated documentation comment in .proto file."""
        context.set_code(grpc.StatusCode.UNIMPLEMENTED)
        context.set_details('Method not implemented!')
        raise NotImplementedError('Method not implemented!')

def add_MonServiceServicer_to_server(servicer, server):
    rpc_method_handlers = {
        'DireBonjour': grpc.unary_unary_rpc_method_handler(
            servicer.DireBonjour,
            request_deserializer=mon_service_pb2.BonjourRequest.FromString,
            response_serializer=mon_service_pb2.BonjourResponse.SerializeToString,
        ),
    }
    generic_handler = grpc.method_handlers_generic_handler(
        'monservice.MonService', rpc_method_handlers)
    server.add_generic_rpc_handlers((generic_handler,))

# This class is part of an EXPERIMENTAL API.
class MonService(object):
    """Missing associated documentation comment in .proto file."""

    @staticmethod
    def DireBonjour(request,
                    target,
                    options=(),
                    channel_credentials=None,
                    call_credentials=None,
                    insecure=False,
                    compression=None,
                    wait_for_ready=None,

```

## 10. Reconnaissance vocale

### 10.1. Bibliothèques et Modules Utilisés

- PyAudio : Utilisé pour capturer l'audio en direct depuis le microphone. Il permet de configurer les paramètres d'enregistrement tels que le format, les canaux et le taux d'échantillonnage. PyAudio est crucial pour la capture précise et en temps réel des conversations pendant les examens, assurant que l'audio est de haute qualité pour la transcription et l'analyse.
- Wave : Utilisé pour lire et écrire des fichiers audio au format WAV. Elle facilite la sauvegarde de l'audio capturé pour une utilisation ultérieure. Cette bibliothèque permet également de gérer les propriétés audio comme le nombre de canaux et le taux d'échantillonnage, garantissant que l'audio est correctement formaté pour la transcription.
- Whisper : Utilisé pour transcrire l'audio en texte. Ce modèle de reconnaissance vocale offre une transcription précise et est essentiel pour convertir l'audio en une forme analysable. Whisper utilise des réseaux de neurones profonds pour comprendre et transcrire le langage parlé avec une grande précision, même dans des conditions de bruit ou avec des accents variés.
- NLTK : Utilisé pour le traitement du langage naturel. Elle permet de tokeniser et lemmatiser le texte transcrit, facilitant ainsi la détection de mots interdits. NLTK offre une

gamme d'outils pour l'analyse linguistique, y compris la tokenisation, la lemmatisation et l'analyse syntaxique, qui sont essentiels pour comprendre et analyser le texte transcrit.

- os : Utilisé pour les opérations sur les fichiers, facilitant la gestion et datetime : Utilisé pour gérer les horodatages, permettant de suivre les occurrences de mots interdits avec précision.

## 10.2. Implémentation

### 10.2.1. Configuration et Enregistrement de l'Audio

L'audio est enregistré en utilisant des paramètres spécifiques pour assurer une capture de haute qualité. Les paramètres incluent la taille des trames (CHUNK = 1024), le format audio (paInt16), le nombre de canaux (stéréo, c'est-à-dire CHANNELS = 2), le taux d'échantillonnage (44100 Hz) et la durée de l'enregistrement (RECORD\_SECONDS = 15 secondes). Une fois les paramètres configurés, un flux audio est ouvert pour capturer les données en temps réel. Les trames audio sont lues dans des blocs, puis stockées dans une liste. Après la capture, les trames sont combinées et enregistrées dans un fichier WAV à l'aide de la bibliothèque wave. Ce fichier WAV servira de base pour la transcription ultérieure.

### 10.2.2. Transcription Automatique

Pour transcrire l'audio capturé en texte, le modèle Whisper est utilisé. Whisper est un modèle avancé de reconnaissance vocale basé sur des réseaux de neurones profonds. Il est chargé et utilisé pour transcrire le fichier audio en texte lisible. Cette étape est cruciale pour convertir les données audio en une forme analysable. La transcription est ensuite enregistrée dans un fichier texte, où chaque ligne représente une nouvelle entrée de transcription. Ce processus permet d'assurer que les données vocales sont prêtes pour l'analyse textuelle et la détection de triche.

### 10.2.3. Traitement du Texte

Le texte transcrit est ensuite traité à l'aide de NLTK (Natural Language Toolkit). Le traitement du texte inclut plusieurs étapes importantes, telles que la tokenisation et la lemmatisation. La tokenisation divise le texte en mots individuels, tandis que la lemmatisation normalise ces mots en leurs formes de base, facilitant ainsi l'analyse. Une liste de mots interdits, associés à des comportements de triche, est comparée aux mots lemmatisés. Si des mots interdits sont détectés, ils sont marqués pour une vérification ultérieure. Ce traitement permet de préparer le texte pour l'enregistrement des résultats et d'assurer que les mots interdits sont détectés avec précision.

### 10.2.4. Enregistrement des Résultats

Les occurrences de mots interdits détectées dans le texte transcrit sont enregistrées dans un fichier log. Chaque occurrence est horodatée, fournissant un enregistrement précis du moment où le mot interdit a été détecté. Cela permet de suivre les comportements suspects et d'effectuer des analyses ultérieures. Le fichier log contient des entrées détaillées, indiquant le mot détecté et le moment de sa détection. Ce processus assure une traçabilité complète des événements et fournit des preuves claires pour toute action disciplinaire nécessaire.

# 11. Résultats

# 12. Références

- Python Software Foundation. (2023). \*Documentation de Python 3.12\*. Récupéré sur <https://docs.python.org/3.12/>.
- Docstring : [Comment installer et configurer Python sur Windows](#)
- gRPC. (2023). \*Documentation gRPC\*. Récupéré sur <https://grpc.io/docs/> et <https://cloud.google.com/run/docs/triggering/grpc> .
- Développeurs OpenCV. (2023). \*Documentation OpenCV\*. Récupéré sur <https://docs.opencv.org/> .
- Google. (2023). \*Documentation Android Studio\*. Récupéré sur <https://developer.android.com/studio>.
- Oracle. (2023). \*Documentation Java SE\*. Récupéré sur <https://docs.oracle.com/en/java/javase/>.
- Git. (2023). \*Documentation Git\*. Récupéré sur (<https://git-scm.com/doc>).
- OpenAI. (2022). Modèle de langage développé par OpenAI. Disponible sur : <https://chat.openai.com/>
- TensorFlow. (2023). \*Documentation TensorFlow\*. Récupéré sur [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs) .
- Apache. (2023). \*Documentation Apache Kafka\*. Récupéré sur <https://kafka.apache.org/documentation/> .
- Termux. (2023). Documentation Termux. Récupéré sur <https://termux.com/docs/>.
- <https://www.cnil.fr/fr/surveillance-des-examens-en-ligne-les-rappels-et-conseils-de-la-cnil>
- <https://www.enseigner.ulaval.ca/ressources-pedagogiques/proctorio>
- [https://www.enseigner.ulaval.ca/sites/default/files/proctorio\\_-\\_guide\\_de\\_lenseignant\\_vf\\_ul\\_2\\_0.pdf](https://www.enseigner.ulaval.ca/sites/default/files/proctorio_-_guide_de_lenseignant_vf_ul_2_0.pdf)
- <https://testwe.eu/fr/>
- <https://mettl.com/fr-fr/online-exam-software-system/#proctoringtools>
- <https://proctoredu.com/>
- <https://demo.proctoring.app/#!/report?id=665634881d99e6e4827b285a>
- <https://www.callimedia.fr/le-learning-lab/blog/le-marche-de-le-learning-en-2023-en-france-et-dans-le-monde/>

