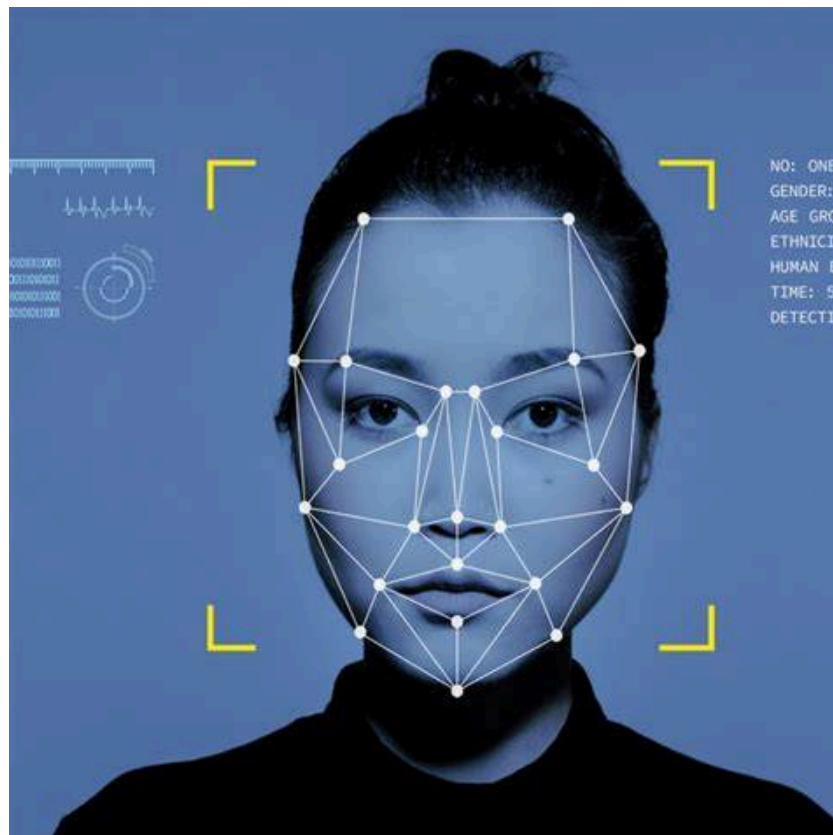

PLAN PROJET RECONNAISSANCE FACIALE POUR LA GESTION DE PRESENCE



***MISKAR AMINA
SABROU HAFSA
ZAHIR OUMAIMA***

PROCESSUS DU DÉVELOPPEMENT DE L'APPLICATION :

1. Définition des objectifs et des exigences :

- Identifier les objectifs de l'application : améliorer la gestion de la présence des étudiants, automatiser le processus de vérification de présence, améliorer la sécurité.
- Spécifier les exigences fonctionnelles et non fonctionnelles : reconnaissance faciale précise, interface utilisateur conviviale, sécurité des données, performances élevées, compatibilité avec les navigateurs web courants, etc.
- Rédiger un cahier des charges professionnel détaillé incluant tous les aspects fonctionnels et techniques de l'application.

2. Planification de gestion de projet :

- Adopter la méthode agile de Scrum pour le développement de l'application.
- Nommer le Product Owner, qui sera généralement un représentant de l'école ou un responsable des systèmes d'information.
- Nommer le Scrum Manager (Scrum Master) chargé de faciliter le processus de développement et de veiller à ce que l'équipe Scrum respecte les pratiques et les valeurs de Scrum.
- Créer le backlog produit contenant toutes les fonctionnalités et les tâches à réaliser pour l'application, priorisées en fonction de la valeur métier.

3. Conception de l'architecture et de l'interface utilisateur :

- Concevoir l'architecture de l'application, y compris le frontend et le backend.
- Créer des wireframes et des maquettes pour visualiser le design de l'interface utilisateur.
- Définir la structure de la base de données pour stocker les données des étudiants, des cours et des présences.
- Spécifier les interactions entre les différentes composantes de l'application.

4. Développement :

- Backend :
 - Utiliser Laravel (avec PHP) comme framework backend pour sa facilité de développement et ses nombreuses fonctionnalités.
 - Utiliser Python avec OpenCV pour la partie reconnaissance faciale.
 - Implémenter les fonctionnalités de gestion de présence, d'authentification des utilisateurs, de gestion des cours, etc.

- *Frontend :*
 - *Utiliser React.js comme framework frontend pour sa modularité et sa réactivité.*
 - *Créer des composants réutilisables pour l'interface utilisateur.*
 - *Intégrer la reconnaissance faciale dans l'interface utilisateur pour la vérification de présence.*
- *Base de données :*
 - *Utiliser SQL (MySQL, PostgreSQL, etc.) pour la base de données afin de stocker les informations sur les étudiants et les présences.*

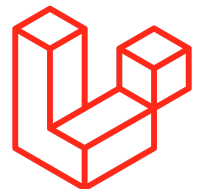
Exploration des technologies et choix technique

Une application web est un logiciel accessible via un navigateur web qui permet aux utilisateurs d'accomplir diverses tâches ou d'interagir avec des services en ligne. Contrairement aux logiciels traditionnels qui doivent être installés sur un appareil local, une application web fonctionne entièrement à travers Internet.

1. Interface Utilisateur (UI) : L'interface utilisateur est ce que les utilisateurs voient et avec quoi ils interagissent lorsqu'ils utilisent l'application. Elle comprend des éléments tels que les menus, les boutons, les formulaires et les pages web.
2. Backend (côté serveur) : Le backend d'une application web gère les opérations côté serveur, telles que le traitement des requêtes des utilisateurs, l'accès aux bases de données, la logique métier et la génération de contenu dynamique. Il est généralement écrit dans des langages de programmation tels que PHP, Python, Ruby, Java, etc., et est hébergé sur des serveurs distants.
3. Frontend (côté client) : Le frontend d'une application web s'occupe de l'interface utilisateur et de l'interaction avec l'utilisateur côté client. Il est généralement développé en langages web tels que HTML, CSS et JavaScript, et est exécuté dans le navigateur web de l'utilisateur.
4. Base de données : La base de données stocke les données nécessaires au fonctionnement de l'application. Elle peut s'agir de données utilisateur, de contenu dynamique, de paramètres de configuration, etc. Les informations sont stockées de manière organisée pour permettre un accès rapide et efficace.
5. Communication client-serveur : Les applications web communiquent entre le frontend et le backend via des requêtes HTTP. Le frontend envoie des requêtes au backend pour récupérer des données ou effectuer des actions, et le backend répond à ces requêtes en renvoyant les informations demandées.
6. Sécurité : Les applications web doivent mettre en place des mesures de sécurité pour protéger les données des utilisateurs et prévenir les attaques malveillantes. Cela inclut la gestion des sessions utilisateur, l'authentification, l'autorisation, la validation des entrées, la protection contre les attaques CSRF (Cross-Site Request Forgery), XSS (Cross-Site Scripting), etc.

LARAVEL :

Laravel est un framework PHP moderne et puissant qui offre une syntaxe élégante et expressive. Il inclut de nombreuses fonctionnalités telles que l'authentification, la gestion des



sessions, la migration de bases de données, ainsi que des outils pour le développement d'applications web robustes et évolutives.

PHP :



Le PHP est un langage **de script côté serveur** largement utilisé pour le développement web. Il permet de créer des pages web dynamiques en intégrant du **code PHP dans du HTML**. Grâce à ses fonctionnalités avancées, telles que la gestion des formulaires et l'accès aux bases de données, le PHP est adapté à la création de sites web interactifs, de systèmes de gestion de contenu, et bien plus encore. Son extensibilité et sa grande communauté en font un choix populaire parmi les développeurs web.

HTML :



HTML, ou HyperText Markup Language, est le langage standard utilisé pour structurer et organiser le contenu des pages web. Il repose sur l'utilisation de balises pour définir différents éléments tels que les titres, les paragraphes, les liens et les images. Ces balises permettent de créer une hiérarchie logique de contenu, facilitant ainsi la compréhension et la navigation sur la page pour les utilisateurs et les moteurs de recherche. Bien qu'aucun framework spécifique ne soit associé à HTML, des outils comme **Bootstrap** et Foundation sont largement utilisés pour fournir des composants HTML pré-stylisés, des grilles de mise en page et d'autres fonctionnalités pour accélérer le processus de développement.

CSS :

CSS, ou Cascading Style Sheets, est un langage de feuilles de style utilisé pour définir l'apparence et la présentation des éléments HTML sur une page web. Il permet de contrôler des aspects tels que la couleur, la taille, la police, la disposition et les effets visuels, offrant ainsi un contrôle précis sur l'apparence visuelle d'un site web. Les feuilles de style CSS peuvent être liées aux documents HTML pour appliquer des styles de manière cohérente sur l'ensemble du site.

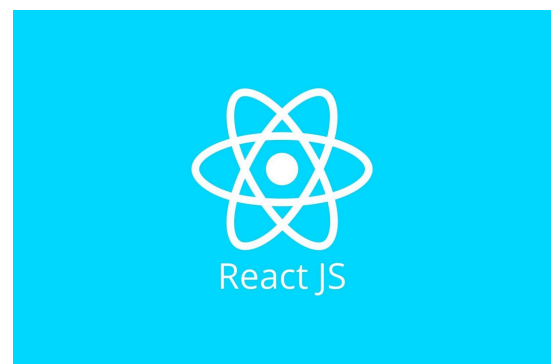
Bootstrap est l'un des frameworks CSS les plus populaires, offrant une vaste



gamme de composants et de styles prêts à l'emploi pour créer des interfaces web réactives et esthétiques. Il fournit également des outils de personnalisation pour ajuster les styles selon les besoins spécifiques du projet.

JAVASCRIPT :

JavaScript est un langage de programmation côté client largement utilisé pour rendre les pages web *interactives et dynamiques*. Il permet d'ajouter des fonctionnalités telles que la manipulation du contenu HTML, la réponse aux événements utilisateur, la validation des formulaires et les animations. En outre, JavaScript peut être utilisé pour interagir avec les API côté client, envoyer des requêtes asynchrones au serveur et manipuler des données JSON. Plusieurs frameworks JavaScript populaires sont utilisés pour développer des applications web modernes et réactives. Parmi eux, Vue.js, *React.js* et Angular offrent des fonctionnalités avancées pour la création d'interfaces utilisateur dynamiques, la gestion de l'état de l'application et le routage. Ces frameworks simplifient le développement en fournissant une structure organisée, des composants réutilisables et des outils de développement avancés.



PostgreSQL:

PostgreSQL est un système de gestion de base de données relationnelle open source, souvent abrégé en "Postgres".

Caractéristiques principales :

Fiabilité : PostgreSQL est reconnu pour sa fiabilité et sa stabilité, avec un système de sauvegarde et de récupération robuste qui assure la disponibilité des données.

Conformité SQL : PostgreSQL est conforme aux standards SQL, ce qui le rend compatible avec de nombreuses applications et outils.

Extensibilité : Il offre une architecture extensible qui permet aux utilisateurs de créer leurs propres types de données, fonctions et langages de programmation.

Performances : PostgreSQL est conçu pour gérer des charges de travail élevées, avec des fonctionnalités avancées telles que la gestion des transactions, la réplication et l'optimisation des requêtes.

Utilisations courantes :

PostgreSQL est largement utilisé dans une variété de domaines, notamment :

- *Développement d'applications web et mobiles*
- *Systèmes de gestion de contenu (CMS)*
- *Analyse de données et entrepôts de données*
- *Applications financières et bancaires*
- *Applications géospatiales et de cartographie*