

# ***DHIKR AI***

---

**Released by:**  
**DOUAE JELLAL**  
**MOHAMED YASSINE BENMEZIANE**

**Professor: Mr. Hakim Hafidi**  
**Project Evaluator: Mr. Ayoub Rabeh**

## ***Academic Year: 2025–2026***

### ***1. Introduction générale***

Dans un monde marqué par une accélération du rythme de vie, le stress, l'anxiété et les fluctuations émotionnelles sont devenus des problématiques courantes. Parallèlement, les technologies numériques, et plus particulièrement l'intelligence artificielle (IA), offrent aujourd'hui de nouvelles opportunités pour améliorer le bien-être mental et émotionnel des individus.

Dans le cadre de ce projet, nous proposons le développement d'une application mobile innovante nommée **Emotional State Detection & Adkar App**. Cette application a pour objectif de combiner **la reconnaissance émotionnelle par intelligence artificielle** avec **la spiritualité islamique**, afin de fournir une guidance spirituelle personnalisée et adaptée à l'état émotionnel de l'utilisateur.

L'idée centrale du projet est de détecter automatiquement l'émotion de l'utilisateur à partir de son visage, sans intervention manuelle, puis de lui recommander des **adkar** ou **douaa** pertinents, contribuant ainsi à l'apaisement émotionnel et au renforcement spirituel.

### ***2.Objectif principal :***

Développer une application mobile intelligente capable de détecter automatiquement l'état émotionnel de l'utilisateur et de lui proposer des adkar et versets coraniques adaptés de manière éthique et personnalisée.

#### ***Objectifs spécifiques :***

- Détecter les émotions faciales avec une précision  $\geq 80\%$
- Proposer un contenu religieux fiable et validé
- Offrir une expérience utilisateur simple et apaisante
- Garantir une séparation stricte entre IA et textes sacrés

### 3. Contexte et justification du projet

De nombreuses applications religieuses existent aujourd'hui (Coran, adkar, rappels), mais elles restent généralement **passives** et **non personnalisées**. L'utilisateur doit souvent chercher manuellement les invocations ou sélectionner son humeur, ce qui limite l'efficacité et l'engagement.

Le projet s'inscrit donc dans une logique d'innovation en apportant :

Une **automatisation complète** de la détection émotionnelle

Une **personnalisation intelligente** des recommandations spirituelles

Une **interaction naturelle**, rapide et intuitive

Ainsi, l'application répond à un besoin réel et actuel, à la croisée de la technologie et du bien-être spirituel.

### 4. Cas d'utilisation principal :

1. L'utilisateur ouvre l'application
2. Il autorise l'accès à la caméra
3. L'image du visage est capturée
4. Le backend détecte l'émotion
5. Une douaa et un verset sont sélectionnés
6. Une explication contextuelle est affichée
7. L'utilisateur reçoit un soutien spirituel adapté

## 5. Analyse des utilisateurs cibles

### 5.1 Profil des utilisateurs

Les utilisateurs visés par l'application sont :

- Des **musulmans** souhaitant renforcer leur connexion spirituelle au quotidien
- Des personnes traversant des périodes de **stress, d'anxiété, de tristesse ou de fatigue émotionnelle**

- Des utilisateurs de smartphones ayant une utilisation basique des applications mobiles
- Des individus recherchant une solution simple, rapide et discrète pour se recentrer

## ***5.2 Environnement d'utilisation***

L'application est conçue pour être utilisée :

- À domicile
- Dans des moments de solitude ou de réflexion
- Lors de périodes de stress (travail, études, examens, problèmes personnels)

Elle doit donc être **rapide, fiable et non intrusive**.

## ***6. Analyse détaillée des besoins utilisateurs***

Les besoins identifiés sont multiples :

### **1.Compréhension de l'état émotionnel**

Beaucoup d'utilisateurs ont du mal à identifier précisément ce qu'ils ressentent (stress, anxiété, tristesse).

### **2.Guidance spirituelle adaptée**

Les utilisateurs souhaitent recevoir des adkar ou douaa correspondant exactement à leur état émotionnel.

### **3.Accessibilité permanente**

L'application doit être disponible à tout moment, sans contrainte de lieu ou de temps.

### **4.Simplicité d'utilisation**

L'expérience utilisateur doit être fluide, sans configuration complexe ni actions manuelles répétitives.

## ***7. Problèmes rencontrés (Pain Points)***

Malgré les solutions existantes, plusieurs obstacles persistent :

- Difficulté à reconnaître son émotion réelle
- Oubli des adkar appropriés à certaines situations

- Manque de personnalisation dans les applications actuelles
- Applications trop statiques ou nécessitant une sélection manuelle

Ces problèmes réduisent l'impact spirituel et émotionnel des solutions existantes.

## **8. Rôle et apport de l'Intelligence Artificielle**

L'intelligence artificielle constitue le **cœur du projet** et permet de dépasser les limites des solutions traditionnelles.

### **8.1 Détection automatique des émotions**

Grâce à la caméra frontale et à des modèles de reconnaissance faciale, l'application est capable d'identifier plusieurs émotions, telles que :

- Joie
- Tristesse
- Stress
- Anxiété
- Calme

Cette détection est réalisée sans intervention de l'utilisateur.

### **8.2 Recommandation intelligente**

Une fois l'émotion détectée :

- Le système associe l'émotion à une catégorie d'adkar ou de douaa
- Les recommandations sont adaptées au contexte émotionnel
- L'expérience devient personnalisée et pertinente

## **9. Présentation générale du système**

L'application proposée repose sur une **architecture client–serveur** avec une **séparation claire des responsabilités** entre le frontend mobile, le backend intelligent et la couche de données.

Cette conception modulaire permet d'améliorer significativement la **scalabilité**, la **maintenabilité** et la **sécurité** du système.

Le système est composé de trois couches principales

- Application mobile (Flutter)
- Backend et services d'intelligence artificielle (FastAPI – Python)

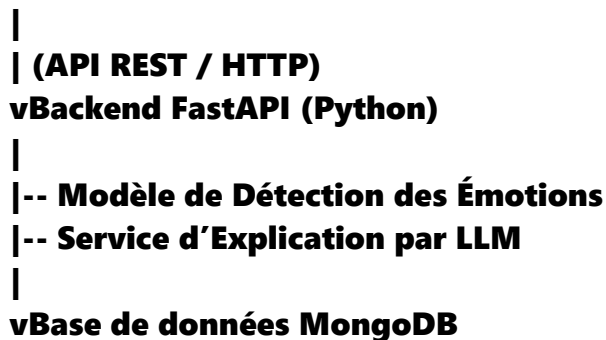
- Couche de données (MongoDB)

Cette architecture garantit une évolution indépendante de chaque composant tout en assurant une forte cohérence globale.

## 10. Architecture globale du système

Le fonctionnement général de l'application est illustré par le schéma suivant :

### Application Mobile Flutter



L'application mobile agit comme **client**, tandis que le backend expose des **services REST** responsables du traitement IA, de la logique métier et de l'accès aux données.

## 11. Architecture Frontend (Flutter)

### 11.1 Rôle du Frontend

Le frontend est responsable de l'ensemble des **interactions avec l'utilisateur**. Son objectif principal est d'offrir une expérience **calme, intuitive et accessible**, tout en masquant la complexité technique du système.

Le frontend assure :

- L'interaction utilisateur
- L'accès à la caméra frontale
- L'affichage des résultats (émotion, score de confiance, explication, douaa, ayah)
- La navigation et l'authentification (optionnelle)

## 11.2 Technologies utilisées

- **Flutter**  
Framework mobile multiplateforme (Android et iOS) à partir d'un seul code source.
- **API Caméra**  
Capture d'images du visage via la caméra frontale.
- **Client HTTP**  
Communication avec le backend via des requêtes REST.
- **Gestion d'état**  
Gestion des résultats de détection émotionnelle et des mises à jour de l'interface.
- **Support RTL (Right-to-Left)**  
Assure l'affichage correct et respectueux des textes arabes (douaa et versets coraniques).

## 11.3 Responsabilités du Frontend

L'application Flutter est chargée de :

- Capturer l'image du visage de l'utilisateur
- Envoyer l'image au backend pour analyse

Afficher :

- L'émotion détectée
- Le score de confiance
- Une explication contextuelle
- Une douaa et un verset coranique pertinents
- Proposer une interface utilisateur apaisante et ergonomique
- Gérer la navigation et l'authentification utilisateur

## 12. Architecture Backend (FastAPI – Python)

### 12.1 Rôle du Backend

Le backend constitue le **cœur intelligent du système**.

Il orchestre l'ensemble des traitements IA, la sélection de contenu religieux et l'accès sécurisé aux données.

## **12.2 Technologies utilisées**

- **FastAPI**  
Framework Python haute performance pour le développement d'API REST.
- **Uvicorn**  
Serveur ASGI pour l'exécution asynchrone et rapide des requêtes.
- **Python**  
Langage principal, parfaitement adapté au traitement IA.
- **Vision Transformer (vit-face-expression from Hugging Face)** : Bibliothèque utilisée pour l'intégration de modèles de langage (LLM) destinés uniquement à la génération d'explications contextuelles. Les textes religieux (douaa et versets coraniques) ne sont jamais générés par l'IA et sont exclusivement récupérés depuis la base de données.
- **PyTorch / TensorFlow**  
Frameworks utilisés pour l'inférence du modèle de détection émotionnelle.
- **OpenRouter API->Mistral 7B(free trier)** : Modèle de langage pré-entraîné utilisé via la bibliothèque Hugging Face pour la génération contrôlée d'explications textuelles non religieuses.

## **12.3 Services principaux du Backend**

### **12.3.1 Service de Détection des Émotions**

- Reçoit les images faciales depuis le frontend
- Analyse les expressions faciales via un modèle de deep learning

**Produit :**

- Une étiquette d'émotion (stressé, triste, calme, etc.)
- Un score de confiance .

### **12.3.2 Service de Sélection de Contenu**



- ◆ Récupère les douaa et versets coraniques depuis MongoDB
- ◆ Associe chaque émotion à un contenu religieux approprié
- ◆ Garantit l'authenticité et la fiabilité des textes
- ◆ Aucun texte sacré n'est généré par l'IA

### **12.3.3 Service d'Explication par LLM**

- ◆ Utilise un **modèle de langage de type Transformer**
- ◆ Génère de courtes explications contextuelles et apaisantes
- ◆ Fonctionne avec :
  - ✓ Des prompts strictement contrôlés
  - ✓ Des mécanismes de validation
  - ✓ Ne génère jamais de versets ou de douaa

### **12.3.4 Couche de Sécurité et de Validation**

- ✓ Validation des entrées (format, taille des images)
- ✓ Filtrage et vérification des sorties
- ✓ Mécanisme de secours (fallback) en cas d'échec de l'IA
- ✓ Respect de la confidentialité des données sensibles

## **13. Couche de Données (MongoDB)**

### **13.1 Rôle de la base de données**

La base de données constitue la **source officielle de contenu religieux et émotionnel** utilisée par l'application.

### **13.2 Technologies utilisées**

#### **MongoDB**

Base de données NoSQL orientée documents, adaptée aux contenus textuels et multilingues.

### **13.3 Données stockées**

La base de données contient :

- ✓ Textes des douaa
- ✓ Versets coraniques avec références authentiques
- ✓ Catégories émotionnelles
- ✓ Métadonnées (langue, date, contexte)

### **13.4 Justification du choix de MongoDB**

MongoDB a été choisi pour :

- ✓ Sa flexibilité de schéma
- ✓ Sa bonne intégration avec Python
- ✓ Son efficacité pour le stockage de texte
- ✓ Sa capacité d'évolution (historique utilisateur, personnalisation)

## **14. Modèles d'Intelligence Artificielle**

### **14.1 Détection des émotions**

- ✓ Modèle de deep learning basé sur les expressions faciales
- ✓ Analyse des traits du visage
- ✓ Sortie : émotion + score de confiance
- ✓ Objectif de précision :  $\geq 80\%$

### **14.2 Modèle de Langage (LLM)**

- ✓ Modèle Transformer
- ✓ Utilisé uniquement pour l'explication contextuelle
- ✓ Contrôlé par des règles strictes
- ✓ Aucun contenu religieux sacré généré

## **15. Conception Éthique de l'IA**

Le projet respecte des principes éthiques fondamentaux :

- ✓ Séparation claire entre données sacrées et contenu généré par l'IA

- ✓ Aucune hallucination de textes religieux
- ✓ Explications non autoritaires et centrées sur l'humain
- ✓ Mécanismes de transparence et de sécurité
- ✓ L'application ne remplace en aucun cas une autorité religieuse et se limite à proposer des contenus de soutien spirituel à titre indicatif.

## **16. Protocole de communication**

- ✓ API REST
- ✓ Échanges au format JSON
- ✓ Communication sans état (stateless)
- ✓ Sécurisation via HTTPS

## **17. Scalabilité et extensibilité**

L'architecture permet d'ajouter facilement :

- ✓ De nouvelles émotions
- ✓ Le support multilingue
- ✓ La récitation audio (Text-to-Speech)
- ✓ L'historique émotionnel des utilisateurs
- ✓ Des mécanismes avancés de personnalisation

## **18. Conclusion**

L'architecture proposée combine efficacement :

- Le développement mobile moderne (Flutter)
- Des services backend performants (FastAPI)
- Une intégration responsable de l'intelligence artificielle
- Une gestion sécurisée des données (MongoDB)

Ce projet constitue une solution **techniquement solide, éthiquement responsable et académiquement pertinente**, adaptée à une utilisation réelle et évolutive.

---

