

Rapport de Projet

Application de Suivi de Santé Mentale

Licence MIASHS – Parcours MIAGE
Université Paris Nanterre

Réalisé par :

Marwa HAFSATI
Melissa AKLI
Assia FERRADJI
Hanna DOUER

Lien GitHub du projet :

<https://github.com/Melissa-akli25/SanteMentale.git>

Table des matières

1	Introduction	2
1.1	Objectifs et problématique	2
1.2	Méthodologie de travail	3
2	Analyse des Besoins	5
2.1	Besoins fonctionnels	5
2.2	Besoins non fonctionnels	5
2.3	Contraintes techniques	6
3	Conception de la Base de Données	7
3.1	Tables principales	7
3.2	Relations entre les tables	8
4	Architecture Technique	9
4.1	Technologies utilisées	9
4.2	Structure du code	9
4.3	4.3 Organisation des fichiers	11
4.4	4.4 Sécurité	12
5	Fonctionnalités Implémentées	13
5.1	Module d'Authentification	13
5.2	Module Mood Tracking	15
5.3	Module Ressources	16
5.4	5.4 Problèmes techniques rencontrés	19
5.5	5.5 Solutions mises en place	19
6	Améliorations Futures	21
6.1	6.1 Fonctionnalités à développer	21
7	Conclusion	22
7.1	7.1 Synthèse du projet	22
7.2	7.2 Compétences acquises	22
7.3	7.3 Perspectives	22

1 Introduction

Pour ce projet, nous avons choisi de développer une application web dédiée au bien-être mental : un espace digital bienveillant dont l’objectif est d’aider chacun à mieux comprendre, suivre et améliorer son état émotionnel au quotidien.

L’idée est née d’un constat simple : aujourd’hui, beaucoup de personnes expriment le besoin d’être plus attentives à leurs émotions, de suivre leurs habitudes de vie, ou encore d’accéder facilement à des ressources fiables pour se recentrer et se sentir mieux.

Nous avons donc imaginé une plateforme apaisante, intuitive et accessible à tous, qu’il s’agisse d’utilisateurs souhaitant simplement suivre leur humeur ou de personnes cherchant à instaurer des routines de bien-être plus équilibrées.

Nous avons accordé une attention particulière à l’importance d’une interface rassurante, d’un ton doux et d’un design pastel spécialement conçu pour réduire la charge mentale de l’utilisateur.

1.1 Objectifs et problématique

Le développement de cette application s’inscrit dans une démarche visant à structurer et digitaliser la prise en charge personnelle du bien-être mental.

Objectifs du projet

Les objectifs ont été définis pour l’utilisateur pour permettre un usage simplifié :

- offrir un outil structuré de suivi émotionnel permettant d’observer l’évolution de l’humeur au fil du temps
- fournir un graphique facilitant l’analyse des variations émotionnelles
- accompagner l’utilisateur dans l’amélioration de ses habitudes de vie (sommeil, hydratation, objectifs personnels)
- proposer des ressources fiables et accessibles telles que des méditations, exercices de respiration et articles
- garantir une interface rassurante pensée pour réduire la charge mentale de l’utilisateur.

Problématique

La question à laquelle notre projet souhaite répondre est la suivante :

Comment concevoir une plateforme numérique capable d’aider l’utilisateur à comprendre ses émotions, à suivre ses habitudes de vie et à adopter des routines de bien-être, tout en restant accessible, agréable et facile à utiliser ?

1.2 Méthodologie de travail

Pour mener à bien le développement de notre application dédiée au bien-être mental, nous avons adopté une démarche structurée. Le travail a été organisé en plusieurs phases, permettant une progression rapide, une répartition efficace des tâches et une bonne communication au sein de l'équipe.

Phase d'analyse

Cette première étape a consisté à définir précisément les besoins fonctionnels et non fonctionnels de l'application. Nous avons identifié les attentes des futurs utilisateurs, dans un contexte où la santé mentale représente un enjeu majeur aujourd'hui. De nombreuses personnes, et plus particulièrement les jeunes, expriment un mal-être, souvent marqué par une augmentation du stress, de l'anxiété et des difficultés à gérer leurs émotions au quotidien.

Ce sujet nous tenait particulièrement à cœur. Il nous semblait essentiel de concevoir une application capable d'apporter un soutien pour un maximum de personnes, en proposant un environnement bienveillant et des ressources permettant aux utilisateurs de mieux comprendre et réguler leur état émotionnel, tout en préservant leur anonymat et en évitant tout sentiment de jugement.

Phase de conception

Une fois les besoins identifiés, nous avons élaboré le modèle conceptuel de données (MCD) et structuré la base de données. Cette étape a également inclus la définition de l'architecture générale du projet Django, l'organisation des différentes applications internes, ainsi que la planification de la navigation entre les pages. Nous avons porté une attention particulière à la cohérence visuelle de l'interface et à l'esthétique pastel censée réduire la charge mentale de l'utilisateur.

Phase de développement

Le développement s'est déroulé en plusieurs étapes, afin de permettre une construction progressive et cohérente de l'application. Notre objectif était de garantir une expérience utilisateur fluide tout en assurant une architecture robuste côté serveur.

Backend

Le développement backend a été réalisé en Python. Nous avons utilisé le framework Django, qui offre une structure robuste, modulaire et parfaitement adaptée au développement d'applications web complètes.

Django nous a permis d'intégrer rapidement :

- un système d'authentification sécurisé
- la gestion des profils utilisateurs
- les modèles dédiés au suivi de l'humeur (Mood Tracking)
- les modules de bien-être (sommeil, hydratation, objectifs)
- l'accès aux ressources (méditations, exercices de respiration, articles)

Grâce à son ORM intégré, nous avons pu manipuler la base de données de manière fluide sans rédiger manuellement de requêtes SQL complexes. Django garantit également une séparation claire entre la logique métier, les vues et les templates, ce qui a facilité l'organisation du code, la maintenance et l'évolution du projet.

Frontend

Pour la partie frontend, nous avons utilisé les templates HTML de Django combinés avec le framework Tailwind CSS. Ce choix nous a permis de concevoir une interface moderne et adaptée aux attentes d'une application centrée sur le bien-être mental.

Tailwind a facilité :

- l'application d'une charte graphique pastel et apaisante
- la mise en place d'un design responsive adapté à tous les appareils

Grâce à Tailwind, nous avons pu concevoir rapidement des composants visuels harmonieux, en accord avec l'ambiance douce et rassurante recherchée. JavaScript a également été utilisé pour enrichir l'interactivité, notamment pour l'animation de cohérence cardiaque via la Web Animation API.

Base de données

Nous avons choisi MySQL comme système de gestion de base de données, en utilisant le langage SQL pour stocker, interroger et manipuler les données. La base contient l'ensemble des informations essentielles au fonctionnement de l'application, notamment :

- les profils utilisateurs
- les humeurs enregistrées au quotidien
- les données liées au sommeil et à l'hydratation
- les ressources proposées (méditations, respirations guidées, articles)

Phase de test

Une série de tests a été réalisée afin de vérifier le bon fonctionnement de chaque module. Cela inclut des tests de navigation, des tests d'intégration entre les différentes applications Django, ainsi que la vérification de la cohérence des données stockées. Nous avons également effectué des tests d'affichage sur différents appareils (ordinateur, tablette) pour garantir une expérience fluide et responsive.

Phase de finalisation

La dernière phase a consisté à corriger les derniers bugs identifiés, harmoniser l'interface graphique, optimiser les contenus et rédiger la documentation du projet. Nous avons également utilisé l'intelligence artificielle comme outil d'assistance pour améliorer le code.

Nous avons par la suite rédigé un rapport du projet sur latex

Tout au long du projet, la communication interne a joué un rôle essentiel : un Google Doc partagé a servi de support centralisé pour les idées, décisions et suivis de tâches, tandis que GitHub a permis une collaboration technique structurée et efficace.

2 Analyse des Besoins

2.1 Besoins fonctionnels

Les besoins fonctionnels définissent l'ensemble des actions que l'utilisateur doit pouvoir réaliser au sein de l'application

Gestion des utilisateurs

La gestion des utilisateurs représente un module fondamental de l'application. Elle permet à chaque individu d'avoir un compte unique et sécurisé

- Inscription, connexion et déconnexion
- Gestion de session sécurisée pour protéger les données sensibles
- Accès personnalisé aux fonctionnalités selon son profil

Mood Tracking

Le suivi de l'humeur offre à l'utilisateur la possibilité d'enregistrer ses émotions chaque jour et d'en suivre la progression au fil du temps.

- Saisie quotidienne de l'humeur via des emojis
- Message facultatif pour apporter de la motivation
- Visualisation d'un calendrier des émotions

Bien-être

Ce module regroupe plusieurs fonctionnalités complémentaires visant à améliorer les habitudes de vie des utilisateurs. Chaque outil a été pensé pour accompagner l'utilisateur sans pression, dans une démarche bienveillante.

- Suivi des heures de sommeil enregistrées
- Suivi de l'hydratation quotidienne
- Fixer des objectifs personnels (ex : faire du sport demain, prendre un café avec mes amis)

2.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent la qualité globale du système et les exigences nécessaires pour garantir une expérience utilisateur optimale. Ils concernent principalement la sécurité, la performance, l'ergonomie et la maintenabilité de l'application.

Sécurité

La sécurité représente un enjeu essentiel, notamment dans une application manipulant des données personnelles et émotionnelles. Plusieurs mesures ont été intégrées afin d'assurer la protection des utilisateurs :

- Hachage des mots de passe via le système d'authentification Django
- Protection CSRF intégrée pour empêcher les attaques par falsification de requêtes

Performance

La fluidité d'utilisation est un critère essentiel pour encourager une utilisation régulière de l'application :

- Temps de chargement réduit pour un accès rapide aux informations.
- Ressources optimisées afin de ne pas surcharger le navigateur ou le serveur

Maintenabilité

L'application doit pouvoir évoluer facilement et être simple à maintenir :

- Architecture Django modulaire permettant d'ajouter ou modifier des fonctionnalités.
- Structure de code claire, documentée et organisée pour faciliter les mises à jour.

2.3 Contraintes techniques

Plusieurs contraintes techniques ont orienté la conception et le développement de l'application :

- **Backend** : Framework Django utilisant Python 3
- **Frontend** : HTML, CSS, JavaScript
- **Animations** : Web Animation API pour la respiration guidée
- **Base de données** : MySQL pour le développement et la production

3 Conception de la Base de Données

La base de données du projet *a* été conçue de manière simple et efficace afin de regrouper toutes les informations nécessaires au suivi quotidien de l'utilisateur. Elle est composée de trois tables principales : **utilisateur**, **tracking** et **resolutions**. Les relations établies permettent d'assurer la cohérence des données.

3.1 Tables principales

Table utilisateur

Cette table contient les informations personnelles de chaque utilisateur inscrit :

- **id_utilisateur** : identifiant unique (clé primaire)
- **prenom et nom** : informations d'identité
- **adresse_mail** : email de connexion
- **mdp** : mot de passe haché

C'est la table de référence à laquelle les autres tables sont connectées

Table tracking

Cette table enregistre les données quotidiennes saisies par l'utilisateur

- **date_mood** : date de saisie
- **hydratation** : quantité d'eau consommée
- **activite** : niveau ou temps d'activité physique
- **sommeil** : durée du sommeil en heures
- **humeur** : humeur du jour

Chaque utilisateur a son propre tracking

- **id_utilisateur** : clé étrangère vers **utilisateur**.

Table resolutions

Elle enregistre les objectifs ou les résolutions fixés par l'utilisateur. Une résolution contient :

- **intitule** : nom ou objectif formulé
- **checked** : état d'avancement (accompli ou non)
- **date_fixee** : date à laquelle l'objectif a été défini

Elle est également reliée à un utilisateur grâce à la clé étrangère :

- **id_utilisateur**

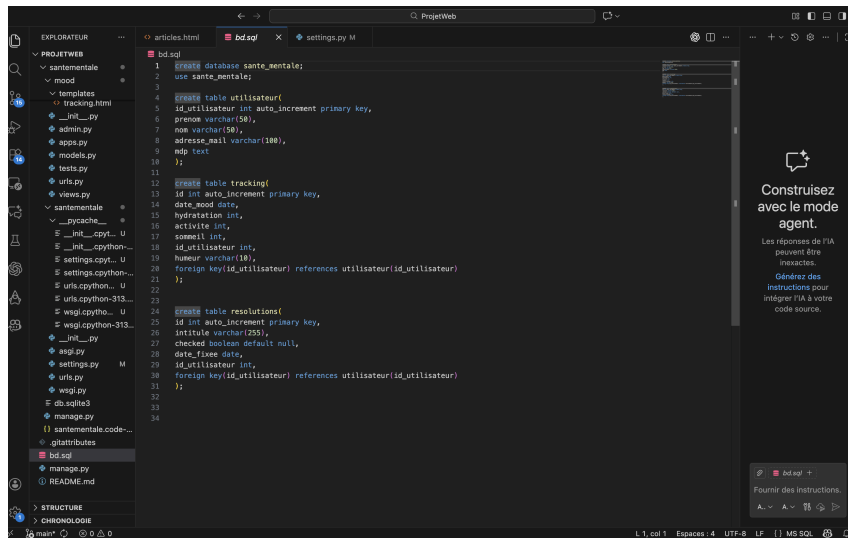


FIGURE 3.1 – Base de donnée

3.2 Relations entre les tables

Les relations de la base sont basées sur l'utilisateur :

- Un utilisateur peut posséder plusieurs enregistrements de suivi dans **tracking** (one to many)
- Un utilisateur peut définir autant de résolutions qu'il souhaite **resolutions** (one to many)

4 Architecture Technique

Voici l’architecture technique de notre application, nous avons utilisé Django comme framework au sein de ce projet

4.1 Technologies utilisées

Backend

Le backend repose sur le framework Django (version 4+). Les technologies principales sont :

- Django 4+
- Django ORM pour la gestion des données
- système d’authentification natif
- système de migrations permettant de modifier la base de données en fonction des réponses de l’utilisateur

Frontend

L’interface utilisateur est développée à l’aide de technologies web classiques, inclut par des animations visuelles et des fichiers multimédias pour les exercices de respiration :

- HTML5
- CSS3 (avec un fichier `tailwind.css`)
- JavaScript
- Web Animation API
- fichiers audio intégrés dans `/static/assets` pour les méditations.

4.2 Structure du code

Le projet suit l’architecture MVC (Model–View–Controller) de Django. Cependant, le fonctionnement général reste très proche du modèle MVC classique :

- **Model** : gestion des données et interaction avec la base
- **View** : renvoie les fichiers HTML
- **Controller** : représenté par le système d’URLs de Django, qui redirige vers les vues

Models

Les modèles correspondent directement aux tables de notre base SQL. Ils définissent la structure des données et gèrent la communication avec la base.

Dans notre projet, nous avons choisi d'utiliser **notre propre base de données MySQL**. Pour cela, nous avons défini un attribut **Meta** dans chaque modèle, afin de préciser :

- le nom exact de la table SQL ;
- le schéma déjà existant.

Cela permet à Django d'utiliser nos tables telles que créées dans notre fichier SQL initial.

Les principaux modèles sont :

- **Utilisateur**
- **Tracking**
- **Resolutions**

Ainsi, le modèle sert d'intermédiaire entre notre code Python et la base de données : **il nous permet de manipuler les données sans écrire directement de requêtes SQL**

Views

Les vues constituent la partie *logique* du projet. Elle est prise en charge par le fichier `views.py`. Elles remplissent plusieurs fonctions essentielles :

- gérer les formulaires (connexion, inscription, saisie du suivi quotidien) ;
- valider les données envoyées par l'utilisateur
- appliquer la logique métier (création d'un objectif, enregistrement du tracking, authentification)
- récupérer les informations depuis la base grâce au modèle
- renvoyer les templates HTML correspondants

La **view fait le lien entre l'utilisateur, la base de données, et l'affichage HTML**.

Templates

Les templates constituent la partie "Vue" au sens de l'interface utilisateur. Ils se trouvent dans `/mood/templates/` et représentent toutes les pages affichées.

Exemples :

- `home.html` : tableau de bord
- `tracking.html` : formulaire quotidien
- `resolutions.html` : gestion des objectifs
- `articles.html`, `exercices.html`, `numeros.html` : pages de ressources
- `login.html`, `register.html` : authentification

Ces fichiers utilisent le moteur de template Django pour afficher dynamiquement les données récupérées par les vues.

Controller

Le controller est pris en charge par le fichier `urls.py`. Il permet de définir quelle fonction de la vue va-t-on appeler selon l'url définie. En d'autres termes, il consiste à créer des chemins et définir pour chaque chemin la fonction de rendu dans le fichiers `views.py`. Exemple : `www.santementale.fr/connexion` va rendre la fonction `connexion` dans `views`.

Static

Le dossier `/static/assets/` regroupe tous les fichiers destinés à styliser l'interface utilisateur :

- images (icônes, illustrations) ;
- fichiers audio `.mp3` pour les méditations et bruits relaxants ;

Rôle du contrôleur dans Django

Ce fichier redirige chaque route vers la vue correspondante :

URL → View → Template

Ainsi :

- l'URL joue le rôle de contrôleur (Controller en MVC)
- la vue exécute la logique
- le template affiche le résultat

4.3 4.3 Organisation des fichiers

Voici l'organisation générale telle qu'observée dans le projet :

- `/mood/` : application principale regroupant les modèles, vues, templates et fichiers statiques
- `/mood/migrations/` : historique des migrations de la base
- `/templates/` : rendu HTML global (hérité par pages spécifiques)
- `/static/assets/` : images, sons, fichiers CSS
- `settings.py` : configuration du projet (base de données, middleware, templates)
- `urls.py` : définition des routes
- `db.sqlite3` : base locale utilisée par Django durant le développement.

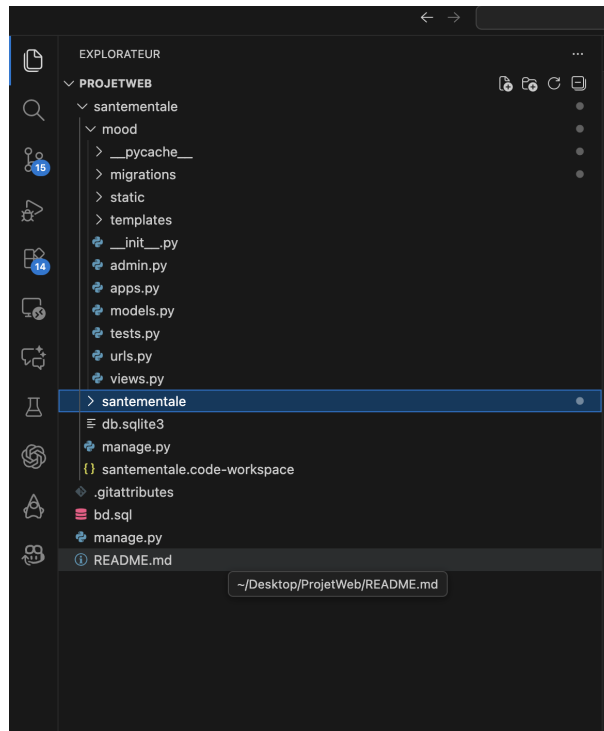


FIGURE 4.1 – Arborescence du projet

4.4 4.4 Sécurité

La sécurité constitue un aspect essentiel du projet. Django fournit plusieurs mécanismes intégrés utilisés dans l'application :

- hachage des mots de passe via le système d'authentification Django
- protection CSRF automatique sur tous les formulaires
- gestion stricte des sessions utilisateurs
- structure des vues empêchant l'accès aux pages sans authentification

```
<!-- Contenu principal -->
<main class="px-6 pb-6 max-w-[63rem] w-full m-auto flex-1">
  <!-- Formulaire d'ajout -->
  <form method="POST" action="{% url 'save_resolution' %}" class="flex gap-4 items-center mb-6">
    {% csrf_token %}
    <input type="text" id="resolutionInput" name="resolutionInput"
      placeholder="Take time for myself..."
      class="flex-1 p-3 rounded-md border border-gray-300" />
    <input type="date" id="resolutionDate" name="resolutionDate"
      class="p-3 rounded-md border border-gray-300" />
    <button type="submit" class="bg-card px-4 py-2 rounded-md hover:bg-gray-400 transition">
      Ajouter
    </button>
  </form>
```

FIGURE 4.2 – Sécurisation des formulaires via le token CSRF Django

5 Fonctionnalités Implémentées

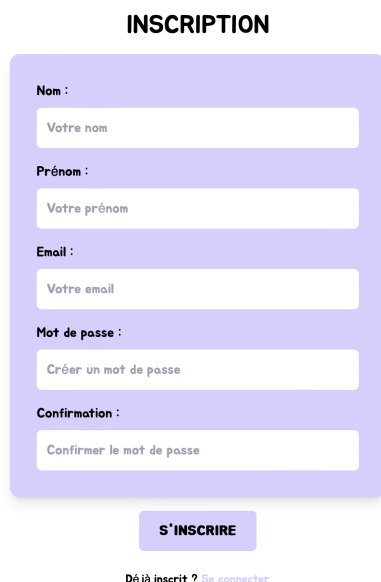
Cette section présente l'ensemble des fonctionnalités développées dans le cadre de l'application. Elles couvrent les aspects essentiels du suivi du bien-être, de l'authentification jusqu'au tracking quotidien.

5.1 Module d'Authentification

Le module d'authentification repose sur le système intégré de Django. Il assure une gestion sécurisée des utilisateurs ainsi qu'un contrôle strict des sessions.

Fonctionnalités

- **Création de compte** : l'utilisateur peut s'inscrire via un formulaire dédié. Les données sont validées, puis enregistrées dans la base. Les mots de passe sont automatiquement hachés par Django.



The image shows a registration form titled "INSCRIPTION". The form is contained within a light purple rounded rectangle. It has five input fields, each with a label above it: "Nom :", "Prénom :", "Email :", "Mot de passe :", and "Confirmation :". The placeholder text for the first four fields is "Votre nom", "Votre prénom", "Votre email", and "Créer un mot de passe" respectively. The placeholder for the confirmation field is "Confirmer le mot de passe". Below the form is a purple button with the text "S'INSCRIRE" in white. At the bottom, there is a link that says "Déjà inscrit ? [Se connecter](#)".

FIGURE 5.1 – Interface d'Inscription

- **Connexion** : une fois inscrit, l'utilisateur peut se connecter grâce au module d'authentification. La vérification des identifiants est réalisée côté serveur.

CONNEXION

The image shows a login interface titled "CONNEXION". It features a light purple rounded rectangle containing two input fields. The first field is labeled "Mail :" and contains the placeholder text "Votre mail" with a small key icon on the right. The second field is labeled "Mdp :" and contains the placeholder text "Votre mot de passe". Below these fields is a purple button with the text "SE CONNECTER" in white. At the bottom, there is a link that says "Pas Encore Inscrit ? Créer un compte".

Pas Encore Inscrit ? [Créer un compte](#)

FIGURE 5.2 – Interface de connexion

- **Déconnexion** : l'utilisateur peut se déconnecter à tout moment.
- **Sécurité des sessions** : Django protège automatiquement les sessions contre les attaques (CSRF, etc.).

```
<!-- Contenu principal -->
<main class="px-6 pb-6 max-w-[63rem] w-full m-auto flex-1">
  <!-- Formulaire d'ajout -->
  <form method="POST" action="{% url 'save_resolution' %}" class="flex gap-4 items-center mb-6">
    {% csrf_token %}
    <input type="text" id="resolutionInput" name="resolutionInput"
      placeholder="Take time for myself..."
      class="flex-1 p-3 rounded-md border border-gray-300" />
    <input type="date" id="resolutionDate" name="resolutionDate"
      class="p-3 rounded-md border border-gray-300" />
    <button type="submit" class="bg-card px-4 py-2 rounded-md hover:bg-gray-400 transition">
      Ajouter
    </button>
  </form>
```

FIGURE 5.3 –

Une fois connecté, l'utilisateur est redirigé vers la page d'accueil, qui affiche son prénom et propose un accès direct au suivi quotidien, aux objectifs, aux ressources et au profil.

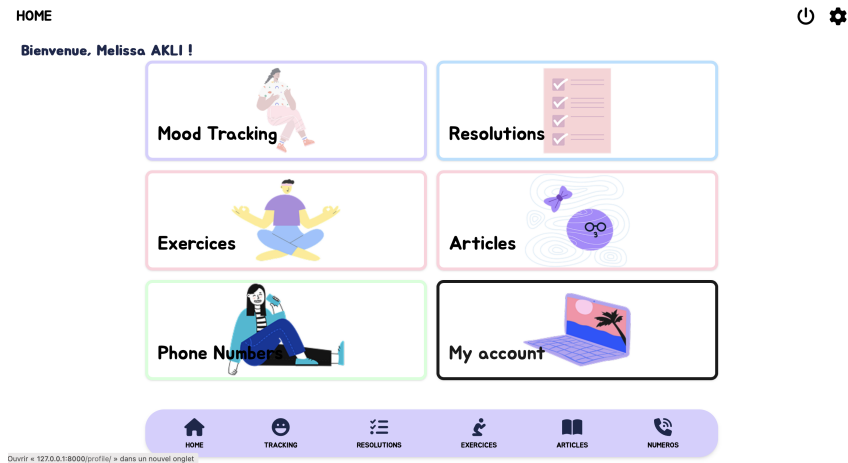


FIGURE 5.4 – Accueil

5.2 Module Mood Tracking

Le suivi quotidien constitue le cœur fonctionnel de l'application. Il permet à l'utilisateur de consigner son humeur et plusieurs indicateurs de bien-être.

Fonctionnalités

- **Saisie d'humeur** : sélection d'un emoji représentant l'émotion du jour,
- **Hydratation** : saisie du nombre de verres d'eau consommés.
- **Sommeil** : enregistrement du nombre d'heures dormies.
- **Activité physique** : évaluation simple de l'activité quotidienne.

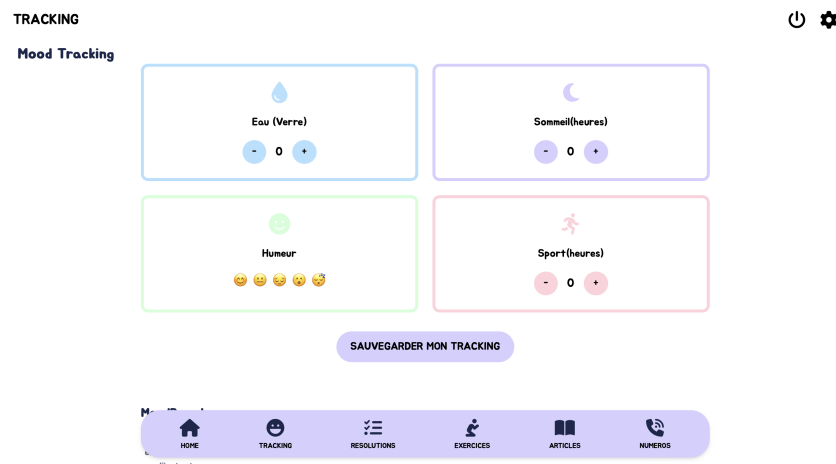


FIGURE 5.5 – Interface de tracking

- **Calendrier** : l'utilisateur peut visualiser son historique de saisies sous forme de calendrier interactif.



FIGURE 5.6 – Diagramme du jour

- **Statistiques** : l'application génère des graphiques ou tendances montrant l'évolution de l'humeur, du sommeil ou de l'hydratation.
- **Messages personnalisés** : selon l'humeur enregistrée, des messages d'encouragement ou de soutien s'affichent.

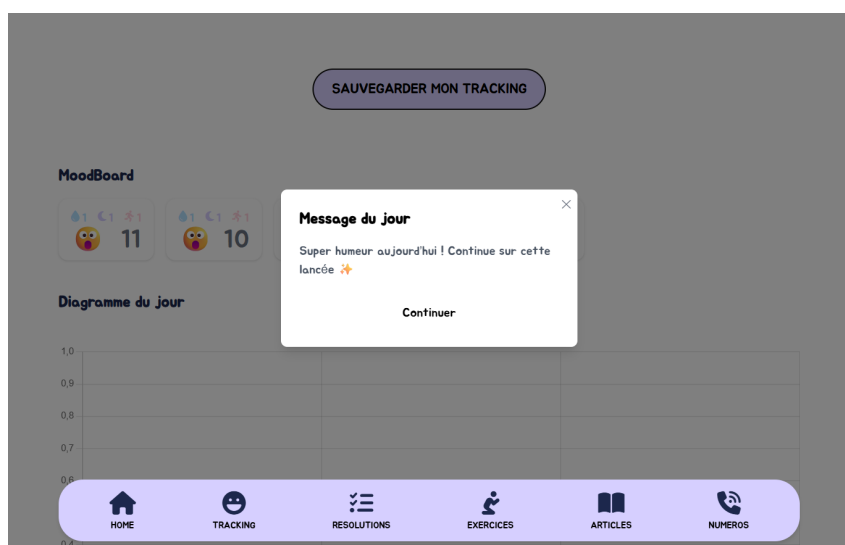


FIGURE 5.7 – Message en fonction de l'humeur

Toutes ces données sont stockées dans la table `tracking`, liée à la table `utilisateur`.

5.3 Module Ressources

Ce module propose un espace d'aide et d'accompagnement pour favoriser le bien-être mental de l'utilisateur. Il regroupe quatre types de contenus.

Méditation

Sessions audio ou vidéos visant à réduire le stress et à favoriser la détente. Elles sont intégrées directement dans l'interface via le dossier `/static/assets/`.

Respiration guidée

Une animation développée en JavaScript grâce à la *Web Animation API*. Elle guide l'utilisateur dans un exercice de cohérence cardiaque.

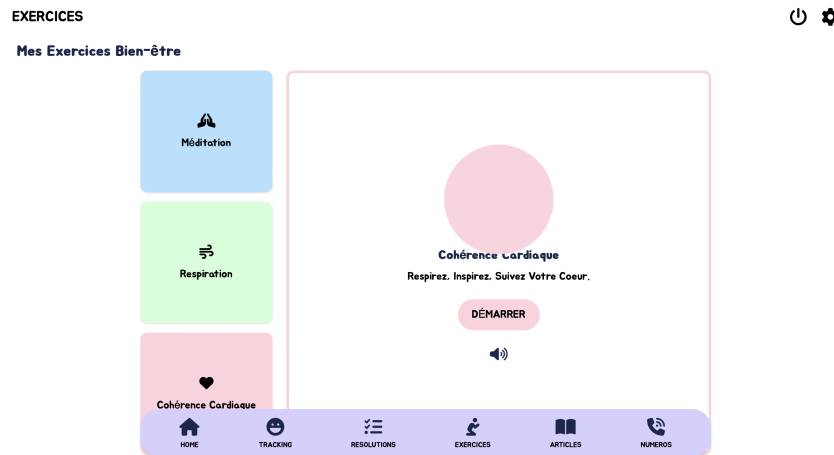


FIGURE 5.8 – Interface des exercices bien-être

Articles

Des contenus informatifs et validés scientifiquement couvrent :

- les émotions ;
- l'anxiété ;
- le sommeil ;
- la gestion du stress.



FIGURE 5.9 – Interface des articles

Urgence

Un accès rapide aux numéros d'aide essentiels :

- 3114 Prévention suicide
- SAMU
- Pompiers

Ces numéros restent accessibles depuis toutes les pages



FIGURE 5.10 – Interface des numéros d'urgence

Mon profil

On peut apporter des modifications à ses informations :

- modifier son nom, prénom
- changer son mot de passe
- supprimer son compte

Cette page reste accessible tout au long de la navigation.

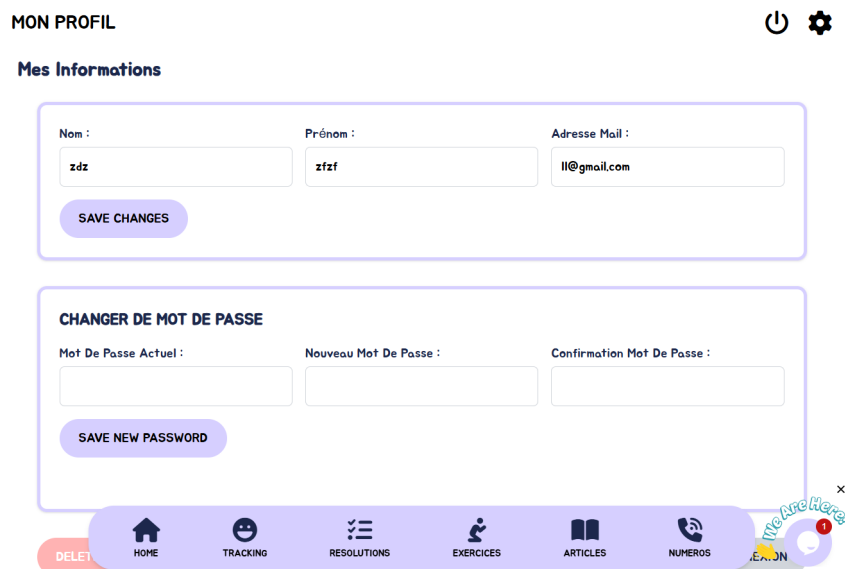


FIGURE 5.11 – Interface du profil

5.4 5.4 Problèmes techniques rencontrés

Au cours du développement de l'application, plusieurs difficultés techniques ont été rencontrées. Elles sont liées aussi bien à l'architecture Django qu'à l'intégration des différentes fonctionnalités.

Principaux problèmes

- **Gestion des migrations Django** : certaines modifications de modèles ont entraîné des conflits entre les migrations, notamment lorsque la base avait été modifiée manuellement via SQL. Des erreurs de dépendances et de synchronisation ont nécessité une régénération partielle des migrations.
- **Organisation des applications** : la structuration en modules distincts (mood, wellbeing, ressources) a demandé une réflexion particulière pour éviter les dépendances circulaires entre modèles, vues et URL.
- **Bug dans le code Javascript et la liaison à la base de données** : le code présentait des bugs dans le fonctionnement du DOM, avec la disparition de certains éléments, ou la coordination des actions avec Django. Nous avons pu utiliser l'intelligence artificielle pour réviser nos bugs et faciliter la cohérence des actions en front et en back.
- **Optimisation du responsive design** : plusieurs pages ne s'affichaient pas correctement sur mobile. L'intégration des fichiers Tailwind CSS et l'adaptation des composants ont nécessité des ajustements.
- **Synchronisation des modules** : les données issues du tracking (humeur, hydratation, sommeil) devaient être cohérentes entre les vues, le modèle et les statistiques. L'un des défis a été d'assurer une mise à jour correcte et instantanée des informations affichées.

5.5 5.5 Solutions mises en place

Afin de résoudre ces difficultés, plusieurs mesures techniques et méthodologiques ont été appliquées au cours du projet.

Méthodes et solutions

- **Modularisation du code** : chaque fonctionnalité a été isolée dans son application Django (mood, wellbeing, ressources), ce qui a amélioré la lisibilité, la maintenabilité et l'évolutivité du code.
- **Tests réguliers** : des tests manuels ont été effectués à chaque ajout de fonctionnalité pour vérifier la cohérence des données, le bon fonctionnement des vues et des redirections, ainsi que la stabilité de l'interface.
- **Validation utilisateur** : chaque formulaire (connexion, création de compte, saisie quotidienne) a été accompagné de messages d'erreur ou de confirmation pour garantir une interaction fluide et intuitive.
- **Documentation précise** : une documentation interne a été créée afin de suivre l'évolution du projet, décrire les modèles, répertorier les chemins internes (URL), et faciliter la résolution des bugs ou conflits de migration.

Ces solutions ont permis de stabiliser l'application, de simplifier les futures évolutions et d'améliorer l'expérience utilisateur.

6 Améliorations Futures

Bien que l'application *Santé Mentale* propose déjà un ensemble complet de fonctionnalités, plusieurs évolutions sont envisagées afin de renforcer l'accompagnement émotionnel et d'améliorer l'expérience utilisateur.

6.1 6.1 Fonctionnalités à développer

- **IA de recommandations émotionnelles** : un module d'intelligence artificielle pourrait analyser l'historique des humeurs, du sommeil et de l'hydratation afin de proposer des recommandations personnalisées : exercices de respiration, méditations ciblées, conseils adaptés, messages positifs, etc.
- **Notifications de bien-être** : l'intégration de notifications pousserait l'utilisateur à maintenir ses habitudes (boire de l'eau, respirer profondément, enregistrer son humeur, dormir suffisamment). Ces rappels contribueraient à la régularité du suivi.
- **Synchronisation multi-appareils** : une synchronisation via un compte cloud permettrait d'utiliser l'application sur plusieurs appareils (ordinateur, tablette, smartphone) tout en conservant un suivi cohérent et centralisé.
- **Application mobile dédiée** : la création d'une version mobile native (iOS/Android), ou hybride via React Native ou Flutter, offrirait une utilisation plus fluide, un accès permanent et une intégration plus naturelle des notifications.

Ces améliorations visent à enrichir l'application, à renforcer son caractère personnalisé et à offrir une expérience plus immersive et continue.

7 Conclusion

7.1 7.1 Synthèse du projet

L'application *Santé Mentale* constitue une plateforme complète dédiée au suivi du bien-être émotionnel. Elle permet à l'utilisateur d'analyser ses habitudes quotidiennes, de suivre l'évolution de son humeur, de consulter des ressources d'aide et de fixer des objectifs personnels. Grâce à une architecture robuste basée sur Django et à une interface orientée utilisateur, le projet offre une solution fiable, intuitive et adaptée aux besoins du quotidien.

Le travail réalisé démontre la capacité de combiner des aspects techniques (développement, modélisation, sécurité) et des dimensions humaines telles que l'accompagnement émotionnel et la sensibilisation au bien-être.

7.2 7.2 Compétences acquises

Au cours de la conception et du développement du projet, plusieurs compétences majeures ont été consolidées :

- **Développement Django** : maîtrise du framework, gestion des vues, modèles, système d'authentification, ORM et migrations.
- **Modélisation de données** : conception de tables SQL, gestion des relations, optimisation des accès aux données via l'ORM.
- **Conception UX/UI** : réflexion sur l'expérience utilisateur, mise en page intuitive, design responsive et intégration d'animations visuelles.
- **Gestion de projet** : planification, résolution de problèmes, documentation et organisation modulaire du code.

Ces compétences témoignent d'une progression significative dans la maîtrise des outils de développement web et dans la réalisation d'un projet complet, de la conception à la mise en œuvre.

7.3 7.3 Perspectives

Le projet possède un fort potentiel d'évolution. Plusieurs axes d'amélioration ont été identifiés, notamment :

- **Intégration d'une IA de recommandations émotionnelles** : cette fonctionnalité permettrait d'adapter les contenus proposés (méditations, messages, exercices) selon l'état émotionnel de l'utilisateur.
- **Mise en place de notifications intelligentes** : rappels d'hydratation, suivi de l'humeur, exercices de respiration ou conseils de sommeil.

- **Synchronisation multi-appareils** : permettre une continuité d'usage entre smart-phone, tablette et ordinateur grâce à un stockage centralisé.
- **Développement d'une application mobile dédiée** : offrir une utilisation plus fluide, accessible en permanence, et mieux intégrée aux habitudes des utilisateurs.

À terme, l'application pourrait devenir un véritable assistant émotionnel intelligent, capable d'accompagner l'utilisateur au quotidien et d'améliorer durablement son bien-être de manière personnalisée et évolutive.